Revisiting legacy systems and legacy modernization from the industrial perspective

Master thesis Business Informatics,
Belfrit Victor Batlajery

University of Utrecht

Revisiting legacy systems and legacy modernization from the industrial perspective

Belfrit Victor Batlajery
University of Utrecht, Utrecht, the Netherlands
August 22, 2013

Supervisors

Drs. Ravi Khadka, University of Utrecht
Dr. Slinger Jansen, University of Utrecht

Preface

The master research is a final deliverable of the Business Informatics program in Utrecht University. It has been carried out under supervision of staff of Utrecht University and in the context of Servicifi project (http://servicifi.org/). The research explores how legacy systems and their modernization are perceived in academia and in industry. Investigating the differences or similarities between the academic and industrial perspective is the aim of this research. The outcome of this research can contribute to both the academia and professional in the industry with regards to the legacy systems and legacy modernization project.

The academic can use the result as a reference to understand what the needs of professionals in industry are. Moreover, the products of academia can be widely used in the industry area. For professionals in the industry, the value of this research is the awareness of the existing of the legacy systems and how the organizations worldwide can deal with legacy systems and legacy modernization.

Acknowledgement

The successful completion of this 8-month research project depended on the guidance of many people. On top of all, I would like to thank Ravi Khadka (Utrecht University) and Dr. Slinger Jansen (Utrecht University) for their guidance and support throughout this project. I would also like to express my appreciation for their patience during my difficult and challenging period.

Secondly, I would like to thank my colleagues in Enable-U for helping me to find some participants for the interview sessions. Very special thanks to people in Information and Computing Science (ICS) department for all their valuable suggestions and comments on any aspects of this research.

In addition, I would like to thank my MBI (Master Business Informatics) friends for their valuable feedback and suggestion on improving the survey quality.

Finally, for all participants, respondents and all friends who helped me to spread the survey in order to validate the finding of this research, thank you.

Abstract

Throughout many years of systems engineering development, a plethora of research has been conducted in an academia regarding legacy systems and legacy modernization. Their works have been resulted in papers, journals, and other products. According to some authors, legacy systems can be defined as a complex and critical system that work well, although it was developed with an outdated technology (software and hardware). They resist of modification and evolution, difficult to understand, and there is scarcity of experts/knowledge, and are inflexible towards new business requirements.

The reports from the academic field clearly indicate the legacy systems as the systems that bring difficulties. The problems such as difficult to maintain, limited supplier/vendor, lack of experts/knowledge and integration issues are common in legacy systems. In addition, the old system also has their lifetime and at some point they cannot be expanded anymore. Hence, there is momentum to modernize legacy systems in order to support organizations' business requirements.

From industrial perspective, business requirements are also evolving and they need more flexible, robust, and agile systems. Organizations cannot depend on their old systems any longer since they are difficult to maintain and the knowledge around it are diminishing. The problems mentioned above are having serious impact on the organization and hence, contributing towards higher maintenance costs.

However, many multinational organizations now are still running their business in their legacy systems for so many reasons. A survey in 2008 in the United States revealed that more than 50% of their IT systems are classified as legacy systems. Furthermore, Gartner Group in 1997 reported that 80% of the world's business ran on COBOL with over 200 billion lines of code in existence and with an estimated 5 billion lines of new code annually. In addition, the TIOBE index also reports that COBOL as one of the most popular languages ever used.

Based on this fact, it is clear that there is a different way of perceiving the legacy systems in academia and in industry. Therefore, this research aims at finding the different perception of legacy systems between academia and industry. The *Grounded Theory* method has been used to interview legacy experts from the industry and the results were validated through survey with 104 participants through online surveys during 3,5 weeks.

The study revealed that the legacy systems are not merely about IT, but also involve business and organization aspect. Academies tend to see the legacy systems from a technical point of view which leads to bad impression of the systems. However, professional in the industry see the legacy system more from the business value of the system. Consequently, problems from the technical side of the legacy systems are not really the problems for professionals in industry unless the problems disturb their business process.

Contents

1. Introduction

1.1. Problem area

The past decades have seen the rapid development of information technology. Despite its agility, flexibility and robustness, the exponential development of information technology creates complexity within software system. On the other hand, if we look in business environment nowadays, it is characterized by rapid changes in the business models, mergers and acquisitions, laws and regulations, and changes in the supporting information technology (IT) infrastructures. Such as situation makes organization to keep updating their systems in order to cope their business processes. The evolution of their systems is continuing again and again.

However, the evolution at some points has reached in a situation where modification, maintenance and development are complicated and difficult particularly for the systems which were built more than three decades ago. Those kinds of systems could be considered as legacy systems (Brodie & Stonebraker, 1995). Applications that were built by using old programming languages such as COBOL, FORTRAN or C long time ago are normally considered as legacy systems. Thus, most studies in regard of legacy systems have associated legacy systems with age. For instance, Sneed (2000) defines a legacy system as a jobs, programs, modules, or procedures within existing application systems which are more than five years old. These systems are usually monolithic system, single tier architectural designs and lack of clean interface because they were built by using the old programming languages (Bisbal et al., 1999). These characteristics make legacy systems hard to change, support interoperability among other systems and difficult to be integrated within current technology environment. A simple update or change may result in huge cost for updating the systems. Therefore, it is difficult for organizations to cope their new business requirements if they are still using their legacy systems.

On the other hand, today’s global requirements require advanced technology in order to stay competitive in market. Organization’s systems are required to be more agile, flexible, high integration, and scalable to stay to cope with today’s business requirement. However, legacy systems are struggling to keep in pace with today’s dynamic business. Organizations are induced to provide their customers with their products and services as quickly as possible. What they need is the systems with flexibility, standards and technologies that provide better abstractions for code, and new models which support interoperability. These requirements are missing in the legacy systems which make organizations hard to cope their business requirements. Moreover, replacing their legacy systems with the one which can comply today’s global business requirements is also difficult and costly.

The ideal solution to overcome problems caused by legacy system is legacy modernization. Legacy modernization helps organization to respond to the changing demand in the business world. Ulrich (2004) defines legacy modernization as an attempt to evolve a legacy system, when conventional practices, such as maintenance and enhancement, can no longer achieve desired system properties. The process should be resulting in a new system where one can overcome the problems caused by legacy system. In addition, Wu et al. (1997) emphasize that the new system produced from legacy modernization should also be easily maintained and adapted to meet future business requirement.

However, modernizing legacy systems is not easy since there is a lot of functionality which are hard coded in billion dollar investment systems (Aversano & Tortorella, 2004). Many challenges and issues are confronting organizations when they want to modernize their legacy systems. Some of them originate from the system itself such as monolith design or hardcoded way of programming and the others come from external factor like law and regulation. Many things have to be considered thoroughly before the modernization process can be started. The preparation before legacy modernization is crucial because modernization is not only about organizations deploying software, but also organizations evaluating their business models, analyzing their business process, and evaluating their partner or customer or supplier relationships (Papazoglou & van Den Heuvel, 2006). Having said that, a proper plan and suitable framework for modernization is needed (Khadka et al., 2013a; Khadka et al., 2013b). As a result, not all organizations which currently run legacy systems are willing to modernize their legacy systems.

Now, it is clear that legacy systems bring more difficulties than its benefit for organizations (Bisbal et al., 1999). Many researches in academia have been discussing about legacy systems and how do they create problems for the organizations in the real world. Problems, difficulties and many negative impressions about legacy systems are dominating the journals, papers, and any other reports in academic field.

In reality, many organizations still operate their business process by using their old legacy systems. COBOL for instance, which always has been seen as a legacy language in academia, is still being used by many multinational organizations in the world. Survey by Gartner Group reported that 80% of the world's business ran on COBOL with over 200 billion lines of code in existence. Kizior, Carr, & Halpern (2000) also support that report and states that there are estimated 5 billion lines of new COBOL code annually being added. With this, we can see that many multinational organizations still depend on their legacy systems. In addition, the TIOBE index also reports that COBOL as one of the most popular languages ever used (TIOBE, 2013).

One question that needs to be asked, however, is why professionals in industry area are preserving their legacy systems? Such as phenomenon indicates that there is a discrepancy in academia and in industry with respect to how the legacy systems are perceived. Hence, this research attempts to explore this discrepancy in detail using an explorative study to investigate how the practitioners perceive legacy systems.

1.2. Research Questions

A plethora of research has been done regarding legacy systems in academia. All the studies reviewed so far, however, suffer from the fact that professionals in industry area are still using their legacy systems to run their business. The existence gap on how those two actors (academia and professional in industry) leads this research to answer the main research question:

Main Research Question (MRQ):
What is the gap in the perception of legacy systems between experts from industry and from academia?

The MRQ is further divided into four sub-research questions:

1.2.1. Scientific literature

Sub Question 1 (SQ1):
How do academics perceive legacy systems?

The aim of this study is to evaluate and validate the gap between academia and professional in industry area regarding legacy systems. Therefore, the first sub-research question is answered by thoroughly investigating on current academic literature. From sub-question 1 (SQ1), the characteristics and problems of legacy systems will be identified from academic point of view. At the end SQ1 is representing the perception on legacy systems from academia.

1.2.2. Experts interview in industry

Sub Question 2 (SQ2):
How do practitioners perceive legacy systems?

Sub Question 3 (SQ3):
What are the drivers for legacy modernization?

Sub Question 4 (SQ4):
What challenges are still identified in legacy modernization by practitioners?

In order to know how **practitioners in** industry perceive the legacy systems, sub-question 2 (SQ2), sub-question 3 (SQ3), and sub-question 4 (SQ4) are introduced. These sub-research questions concern the experience and opinions of the experts in industry and have been answered through intensive semi-structured interview with them. From SQ2, the author can identify the main characteristics of legacy systems based on the opinion of industrial experts. SQ3 and SQ4 are more focused on legacy modernization. Subsequently, questions about legacy modernization are asked to identify and discover the main drivers and the main challenges of such as project.

At the end of this research, the author compares the perception between academia and practitioners.

1.3. Research Method

The objective of this research is to explore how legacy system is perceived in industry and revisit legacy modernization from industrial perspective. In this research, the author aims at exploring new perspectives and insights about legacy systems. Thus, this research is adopts exploratory research using *Grounded Theory* (GT) as qualitative research through the process of interview. By doing interview, the author can understand the meaning or nature of experience of persons with legacy systems and legacy modernization. It helps the author to find out what practitioners think about legacy systems and how do they perform legacy modernization. On the other hand, survey is used as a triangulation method to validate the findings derived from interview process.

Two research methods are applied in this research, being interview and survey. The main reason is the blend of technical and human behavioral aspects are needed to answer the research questions. In software development and evolution, there is a number of management and organizational issues, or people problems that need to be addressed in order for the field to progress (Seaman, 1999). Therefore, combining qualitative and quantitative method can help to overcome this complexity in the area of software engineering. When research is relying on human as a subject of interest, mixed method (qualitative and quantitative) is suitable and seems fit in this research. Although mixed method is used in this research, but this research is relying primary on qualitative part and use quantitative part as the way to validate the finding resulted from qualitative part.

The benefit of doing qualitative research is that it forces the researcher to delve into the complexity of the problem rather than abstract it away. Moreover, the nature of the research problem is also the main reason to conduct qualitative research. It allows the author to understand the meaning of experience of person with problem of legacy systems. Therefore, the findings will be richer and more informative (Seaman, 1999). Strauss and Corbin (1998) argue that there are 3 components in qualitative research, namely data that will be analyzed, the procedure of interpret and organize the data, and the reports to represent the findings.

The data from qualitative part involve words, description, pictures, and diagrams and they can be from various sources (data triangulation) such as interview observations, documents, records, and movie. The procedure of interpret and organize the data is sometimes refers to coding in which the author will conceptualize, reduce, elaborate categories and their properties, and relate one proposition statement with the other proposition statements. The way of categorizing, conceptualizing, collaborating and sorting is often refers to what so called coding in *Grounded Theory*.

Meanwhile, quantitative data involve number and classes were analyzed using statistic approach. In this research quantitative method will be done by using online survey. The benefit of survey is it can include open question and closed questions to collect different type of data. The main idea of using survey in this research is to validate the finding which is derived from qualitative method. Thus, survey can also be seen as validation tools in this research.

1.3.1. Research Model

The main focus of this research will be stressed in qualitative method (interview) with the combination of limited quantitative method (survey) for validation purposes. Figure 1 depicts the research model that is followed.

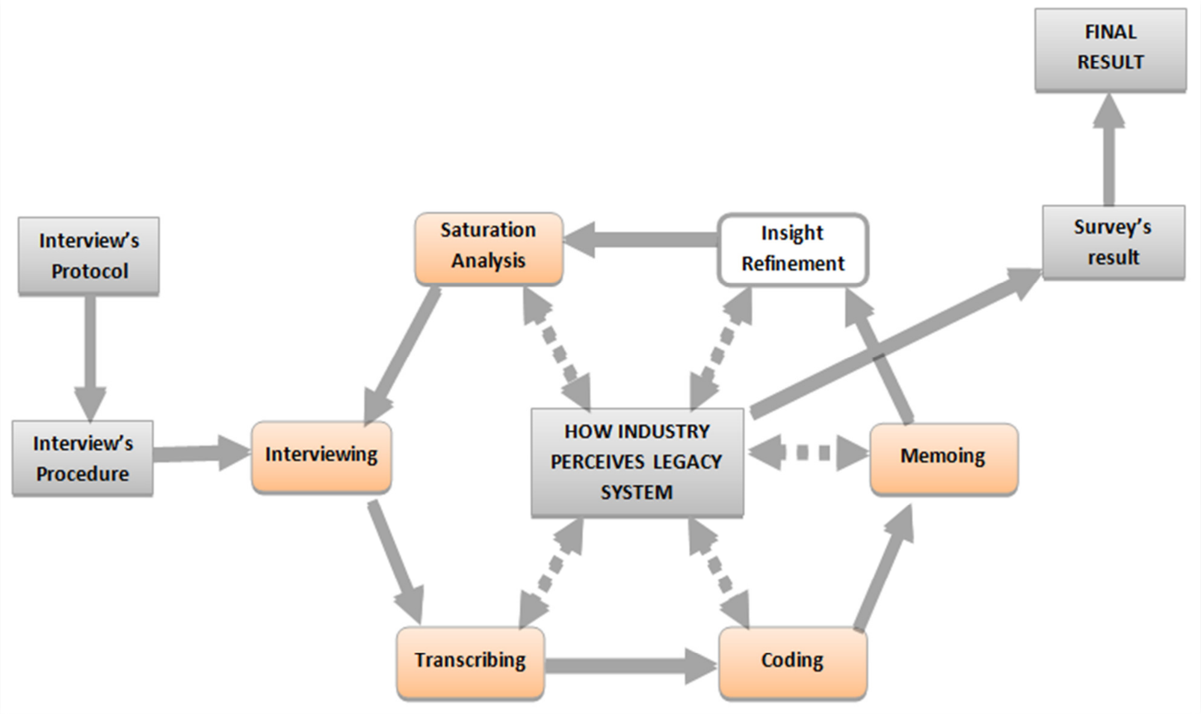


Figure 1. Research Model.

Prior an interview, the author needs to do some preparations such as create a procedure and protocol for the interview. Protocol is a step by step descriptive guideline to achieve completion of a task. Protocol is to be followed in paper and can be applied in almost all circumstances. The procedure, on the other hand, is a description of how to go about performing a task. Procedure can be continuously adjusted to the specific situation or condition but still follows the guidelines in protocol.

Some information covers in the protocol are:

- Names of researchers and contact information
- Purpose of the research
- Procedures used during the interview, i.e. steps the researcher will carry out during these activities

The role of protocol is needed before conducting an interview. It helps author as a guide when conducting the data collection and in that way preventing him from missing to collect data that were planned to be collected. The protocol also helps the author decides what questions to ask which lead to lower the risk of missing relevant data sources. Lastly, it can serve as a log or diary where all conducted data collection and analysis is recoded (Roneson & Host, 2008).

To apply *Grounded Theory* method, data received from interview were processed into several steps: (i) transcribing, (ii) coding, (iii) memoing, (iv) insight refinement, and (v) saturation analysis (Adolph, Hall, & Kruchten, 2011). By using *Grounded theory*, the author tries to get the new perspective and insight by exploring the data received from interview to grow his insight.

First step in this model is conducting the interview and record it. After the interview is performed, the recorded voice was transcribed word by word. The transcription then was cut into smaller units (in sentences or paragraphs) and was categorized based on key characteristic. The process is the called coding. The coding process is not only the process of labeling, but also the process of leading the raw data to the idea and link them together (Richards & Morse, 2007). From coding the author can also identify the pattern such as similarity, difference, frequency, sequence, correspondence, or causation. After the process of coding, the process of memoing is performed. Memoing is the process of writing down narratives explaining the ideas of the evolving theory. The objective of memoing is to demonstrate to the author himself, provide his interpretation, express his position, and explain his decision during the process of research. If the interviewees provide the similar answer with the previous interviewee, then saturation analysis is performed. Saturation state means the condition that there is no additional information is being found. The process of transcribing (written down the text), coding (categorize the text into theme) and memoing (capture researcher's thought) were executed repeatedly. Each time the process is finished, the result will be checked against the preposition conclusion in order to growing it during the process of analysis. These processes were executed until saturation state is achieved.

In this research survey plays a role as supporting information in validating the finding resulted from the interview. Therefore, in order to build strong and solid conclusion, this research is using multiple data sources. The use of multiple data collection is known as data triangulation. The aim of using data triangulation is to prevent an interpretation from one single data source, thus minimize the biases.

1.4. Scientific and Industry contribution

The scientific contribution of this research is to explore the industrial perspective of the legacy systems. The result can be compared with scientific view about the legacy systems, to see the difference between academic views and industrial views. By doing this, it helps academia to understand the relationship between different perspectives of legacy systems. Moreover, the research can also be a source of information for future research and help to develop legacy modernization research agenda in collaboration with industry.

The research also provides some suggestions to academia about how the problems of legacy systems can be better responded. For instance, one of the solutions to the knowledge scarcity problem within legacy systems can be anticipated by planning and introducing courses related with legacy systems in college and university curriculum. It could be the way to make the relation between academic and industry stronger.

On industrial side, this research aims to give a better insight of how are legacy systems are perceived in industry. It implies that the result could enrich the knowledge about the concept of modernization based on real case. Organization can get the benefit by knowing the common characteristics among people in industry, and the common strategy to deal with it. They can also have a broad view about reasons and drivers behind modernization and success and failure factors of legacy modernization.

In addition, the finding can give an input to decision-making regarding legacy modernization and can have a better prediction on what it will be in the future regarding legacy system.

1.5. Outline of the study

The thesis is structured as follows:

Chapter 2 will use a literature study to answer sub-question 1 (SQ1). It is divided into 3 parts. The first subchapter will explain about what a legacy system is from academic point of view such as characteristics and problems it holds. The second part of second chapter is about legacy modernization and how academia contributes to legacy modernization. The last part is discussing about ISO / IEC International Software Standard as an international standard of Information technology, software product evaluation, and quality characteristics of software.

Chapter 3 introduces the *Grounded theory* and its importance in this research. Within this chapter, the author describes how the grounded theory research is executed by explaining the transcribing, coding, memoing and saturation stages.

Chapter 4 explains about the process of interview with experts from industry. How the semi-structure interview is performed including its preparation is be described in this chapter. It consists of four subchapters, namely *Preparation before interview session*, *Interview session*, *Domain of participants*, and *Interview data analysis*. The fourth subchapter covers all the opinions and knowledge about legacy system and legacy modernization from industry point of view, such as *characteristics*, *problems of legacy systems, drivers* and *challenges of legacy modernization*.

Chapter 5 provides the result from the interview with experts. In this chapter, explanation on how certain phenomenon happens in industry is presented. From the grounded theory process, we derive the characteristics of legacy systems, motivating factors behind legacy modernization and finally, challenges associated with legacy modernization. Various causes and effect relationships among the factors identified are also presented.

Chapter 6 describes the validation process of our finding from the interview results. The result from interview part is validated using an online survey. On the other words, the conclusion from experts in industry will be put against the result from survey. The data from the total of 104 respondents is presented after being online for approximately 3,5 weeks. The demographic data such as the origin of respondents, the domain of the organizations and the position in their organization is presented in this chapter.

Chapter 7 presents our discussion. The result from interview session and the result from survey are put against each other to see if the results agree. Later on the result from this discussion is the perception about legacy system and legacy modernization from industrial perspective. In this chapter, the research questions are answered according to the findings. Threat to validity of this research is also explained in this chapter.

Finally, Chapter 8 concludes our research and also provides future.

2. Literature study

Document analysis approach is adopted in this research to collect data. Document analysis can be defined as an activities in which documents (journals, papers, etc.) are interpreted by the researcher as a data source to give the meaning around the interesting phenomenon. The documents can be from any sources such as journal, paper, magazine, whitepaper, annual report, handbook, training materials, slideshow, newspaper, etc. Thus, the author sees this as the same as literature study.

Study of the previous literatures will give the basic idea and information of legacy systems and about legacy modernization. Papers, journals, conference and workshop proceedings are studied to build strong comprehension about the topic of this research. Those sources are used to gather information about legacy systems from academic perspective. The topic that will be studied is the concept on legacy system and how to modernize the legacy system. In order to find the related journals and papers for academic resources, the author used Google scholar and Omega.

The author used several key words to derive certain documents from each search engine. Those keywords are listed below:

- legacy systems
- legacy systems transformation
- legacy system evolution
- software engineering
- migration software
- legacy systems modernization
- software development
- legacy software
- redevelopment software
- legacy systems migration
- system engineering
- legacy application
- wrapping

Besides getting the basic idea and build strong comprehension around assessment topic, the aim of documents analysis is to get better understanding about how legacy systems and legacy modernization are perceived from academic point of view. It is one of the information sources from academic worlds in regards to the topic selected. By doing document analysis, the author was able to answer the sub-research question 1.

2.1. Legacy systems

2.1.1. Definition of legacy systems

It seems that different authors have their own definition of legacy systems. In general, a legacy system could be defined as a large and vital system in organization, but nobody knows how to cope it (Bennett, 1995). Similar definition is provided by Wu et al. (1997); Meng, Qu, & Guo (2013); van Deursen, Klint, & Verhoef (1999); and Rahgozar & Oroumchian (2002) as they define legacy system as large number of existing computers or systems which have been used several decades and resulting some sort of complex and complicated system. The complexity is resulted from many different object models. Those models were popular in the past and the integration between those models creating the complexity in legacy systems environment.

Bennet (1995) describes legacy systems as the systems which are supported by technology which is outdated but the systems are still useful to work. Sneed (2006) supported this argument and refers legacy systems as an outdated technology. The other authors define legacy systems as the systems which not only about the age of the system (Stehle et al., 2008; Khadka et al., 2013; Rahgozar & Oroumchian, 2002; Nowakowski, 2012), but also concern with supportability (software / hardware), risk, staffing and agility (NASCIO, 2008; Sneed, 2006; Weiderman, 1997).

Some authors focus more on data implementation such as indexed file system and non–relational database product (e.g., IMS, VSAM, Adabas, DataComm, CODASYL, CA-IDMS) as a way to define legacy system (Lin, 2008; Rahgozar & Oroumchian, 2002). Similarly, Rahgozar & Oroumchian (2002) mention about the structure of the legacy data which mostly navigational or hierarchical and the logic of the legacy programs is built around this structure.

Last but not least, some authors stated that the programming language is also a way to consider system as a legacy (e.g., Assembler, COBOL, JCL, PL/1), because most of the time they run in obsolete mainframe computers (Lin, 2008; Colosimo et al., 2007; Weiderman, 1997; Geetha, 2012). The language in mainframe could be varying from JCL for batch jobs purposes or SQL for querying databases. For constructing business logic, COBOL, PL/I and FORTRAN are the well-known language which can be applied in mainframe. COBOL, which many authors associate it with legacy language, is a popular language in business world nowadays. Report by TIOBE says that COBOL still is the most popular language ever used until 2012 (TIOBE, 2013). By looking on the way they define legacy system, programming language seems to have much influence to determine if a system is legacy

The summary of the definition of legacy system from various authors is shown in table 1 below.

Authors	Definition
Bennett (1995)	Large software systems that we don't know how to cope with but that are vital to our organization". Legacy software was written years ago using outdated techniques, yet it continues to do useful work.
Wu et al. (1997)	The widespread use of computer technology over several decades has resulted in some large, complex systems which have evolved to a state where they significantly resist further modification and evolution.
Weiderman (1997)	Systems that represent a patchwork of mainframe, minicomputer, and desktop applications, both centralized and distributed, under dispersed control. Geography, database incompatibilities, and corporate mergers can fragment them.
Rahgozar & Oroumchian (2002)	• Aging application systems developed during the last three decades. They constitute a large number of existing systems. • Systems that are using either the simple data implementations such as indexed file systems or the old database technologies such as IDMS, CODASYL, NETWORK, etc. • The implementation or structure of the legacy data is mostly navigational or hierarchical and the logic of the legacy programs has been built around this structure.
Sneed (2006)	Programs which have been developed with an outdated technology make-up for the vast majority of programs in many user application environments.
Colosimo et al. (2007)	Systems that are written in some legacy language like COBOL and do not directly interoperate with other applications
Stehle et al. (2008)	Older systems, commonly referred to as legacy software, have great value to the organizations that use them, but create maintenance issues as they age.
Lin (2008)	Systems that utilize a variety of non-relational database products (e.g., IMS, VSAM, Adabas, DataComm, CA-IDMS), are coded in 2nd or 3rd generation languages (e.g., Assembler, COBOL, JCL, PL/1), and often run on obsolete mainframe computers.
NASCIO (2008)	System that is not solely defined by the age of IT systems (e.g. 20 years) as there are many systems that were designed for continued upgrades, but the term also focuses on elements such as supportability, risk, and agility including the availability of software and hardware support, and the ability to acquire either internal or outsourced staffing, equipment or technical support for the system in question.
Khadka et al. (2013)	Business-critical systems that have been developed over the last three decades or more.
Geetha (2012)	Older IT systems—legacy systems such as mainframes and COBOL based software.
Nowakowski (2012)	Software applications that have been in production for years being constantly evolved in order to adapt them to business changes resulting both from changing market needs as well as emerging new technologies providing new business opportunities.
Meng, Qu, & Guo (2013)	Large-scale and complex software system which has run for a long time (more than 20 years).

Table 1. Summary of the definition of legacy systems from various authors.

2.1.2. Characteristics of legacy systems

Sneed (2006) divided legacy programs into three basic categories in regard to the degree of dependence on their environment.

- Programs which are not dependent on their environment,
- Programs which are partially dependent on their environment,
- Programs which are totally dependent on their environment.

He also provides the examples of programming language which fall into each category. The programming languages that fall into first category are Fortran, Cobol and C/C++, while the second category are PL/I, Smalltalk, and Forte. The programming languages in the third category consist of 4th generation language programs, such as ADSOnline, Natural, CSP and Oracle Frames.

Juric et al. (2000) divided legacy system into 2 parts: (i) traditional, and (ii) modern system. Traditional means that the system is mostly mainframe based system, while modern means that system written in modern language such as C, C++, etc. Zhang, Liu, & Yang (2005) also have the same perception about legacy systems. They argued in their paper that legacy system can be characterized as a monolithic, single-tier and mainframe-based application. In addition, van Geet (2011) in his paper mentioned a legacy system is a system in which the processes are mostly in mainframe. The process could be divided into *Batch processing* and *Online Processing*. Online mode implies interaction with users, which typically takes the form of transactional processing supported by infrastructure such as IMS (IBM's Information Management System) or CICS (IBM's Customer Information Control System). Batch mode, on the other hand, implies non-interactive processing used for processing data in bulk. These batch processes usually run at night or during the weekend and/or controlled by scripts.

In their technical report, Weiderman et al. (1997) also argue that legacy systems are likely to be built in mainframe which make the systems are centralized, geography and database incompatibilities, and corporate mergers can easily fragment them. They also support the previous author by categorizing legacy systems as system that runs in obsolescent hardware and suffers from fragileness and brittleness. These suffers are mainly because of large part of architecture of the systems remained unchanged as a result from prolonged maintenance (Seacord et al., 2001).

Another characteristic of legacy systems is the way it was built. Legacy systems tend to have un-integrated (stovepipe) and monolithic architecture that make them unwieldy and rigid. Weiderman et al. (1997) explained in their paper that un-integrated (stovepipe) software assets that are not used for continuous production of additional assets become stale and require increasingly assets to maintain them. Eventually, there may be more cost associated with their continued maintenance than benefit from their continued use.

However, this traditional architecture at some point has reached its limitation. From technical aspect, the technical architecture of legacy system is obsolete (Colosimo et al., 2007). Thus, they are inflexible to the new changes and do not responsive enough to meet new business requirements. For instance, changing in structure of the organizations, such as corporate mergers or acquisition can easily disrupt legacy system (Weiderman et al, 1997). Moreover, the limitation also comes from incompatibility to meet the expectation for the use of modern technologies (e.g. modern user interface) (NASCIO, 2008).

One main point about legacy systems is that they are typically forming the backbone of information flow and cannot simply be discarded (Aversano & Tortorella, 2004). They hold detailed business rules and play a crucial role in organization's business operations for long period of time. Therefore, these systems are typically mission-critical of the organization and failure of the systems can result serious impact on business. It makes makes legacy systems to operate 24 hours a day in order to keep their business process alive (Wu et al., 1997; Colosimo et al., 2007; Bisbal et al., 1999; and Paradauskas & Laurikaitis, 2006).

Last but not the least, legacy systems due to their age have been changed/evolved many times resulting in complicated system. As they were e, the performance is becoming slower and slower (Bisbal et al., 1999). Moreover, the structure and logic of legacy programs is strongly tied to the legacy data access logic makes legacy system more complex (Rahgozar & Oroumchian, 2002). As a consequence, they resist to modification and evolution to meet the new and constantly changing business requirements (Paradauskas & Laurikaitis, 2006).

The summary of the definition of legacy system from various authors is shown in table 2.

Authors	**Characteristic**
Weiderman et al. (1997)	Centralized, geography and database incompatibilities, inflexible.
Bisbal et al. (1999)	Poor performance.
Juric et al. (2000)	Mainframe application, old programming language.
Seacord et al. (2001)	Difficult to maintain.
Rahgozar & Oroumchian (2002)	legacy programs is strongly tied to the legacy data access logic
Aversano & Tortorella (2004)	Backbone of information flow.
Zhang, Liu, & Yang (2005)	Monolithic, single-tier and mainframe-based application.
Paradauskas & Laurikaitis (2006)	Resist to modification and evolution.
Sneed (2006)	Environment dependent.

Colosimo et al. (2007)	Obsolete technology, stovepipe, monolithic architecture.
NASCIO (2008)	Incompatibility with new ecosystem.
van Geet (2011)	Mainframe application.

Table 2. Characteristics of legacy system from various authors.

2.1.3. Problems caused by legacy systems

Some characteristics and facts above can bring troubles for the organizations that still run their business processes on their legacy systems. The most common reasons why the legacy systems are still there is because the legacy systems play critical role in business process as a backbone of information flow (Bisbal et al., 1999). As consequences, to keep those legacy systems running in their organization, CIO's and IT executives are, at least now facing some difficulties caused by legacy systems such as:

1. *High operational cost:* By keeping legacy systems alive, the cost for maintaining, operating and monitoring will be more expensive. According to an informal industry poll about the budget a company spends for maintaining legacy system is fall between 85% - 90% of the total IT budget (Erlikh, 2000). It means that only 10% - 15% remain for innovation and development. A financial situation like this, of course, is not what the most organizations want.
2. *Difficulty of maintain:* Due to scarcity of experts and up-to-date documentation, the maintenance of legacy systems is difficult. Legacy system mostly was developed with early programming language (such as ASM, COBOL or JCL etc.,), and the development framework or environment of legacy software has been outdated. In additional to this, the codes in legacy systems are also difficult to understand due to ad hoc evolution (Lin, 2008). As the years passes, the legacy systems have been maintained in crisis mode with no updates to the higher level documentation (Wiederman et al., 1997). It makes the legacy systems hard to maintain and to evolve now (Meng, Qu, & Guo, 2013; Wu et al., 1997; Paradauskas & Laurikaitis, 2006).
3. *Unsupported hardware and software* and *inability to be updated:* Because out-of-date technology, it is hard to update the legacy systems. Legacy systems which were built many decades ago are becoming slower now because they are running on obsolete hardware (Paradauskas & Laurikaitis, 2006). These old legacy systems are often not supported by their suppliers or vendors anymore. Patches or security updates are often lack or not available in the market anymore which makes the systems are more vulnerable or have a high risk of failure. Updating the hardware is a problem too, since limited number of vendors is providing hardware for supporting the legacy systems.
4. *Hard to find qualified experts:* It is predicted that in the future it will be difficult to regeneration skilled people who are experienced with legacy systems. The reason is because the existing experts are getting old or nearly to their retirement day and not many young people are interested in the old technology. Research by Morello, Kyte, & Gomolski (2007) revealed that many IT college students and IT interested people around 20-ies are more interested in film, music, multimedia, gaming and internet companies. In United States, for instances, the number of students enroll for computer science have dropped 39% since 2002 (Gartner, 2007).
5. *Complex in integration issues* because of *poor and rigid architecture system*: The architecture of legacy system is mainly stovepipe architecture and haphazardly design. Thus, legacy systems is a monolithic and big sized systems (e.g. without clear distinction between user interface, application logic and business model) (Nowakowski, et al., 2012). With this characteristic, a legacy system has difficulty to integrate with other systems because of non-extensibility, incompatibility, and less-openness of the underlying hardware and software of these legacy systems (Bisbal et al., 1999). Moreover, it also difficult to support large scale and geographic deployment because of this architecture limitation. Legacy application also tends to depend on their legacy software infrastructures which make them not portable with another system's environment.
6. *Rigid to changing business requirement:* Legacy systems are well known for their resistance towards modification and evolution. It makes the process of updating the new functionality slower and leads to higher time-to-market.
7. *Not well documented*: The common problem of legacy systems is that systems often lack up-to-date documentation and knowledge about the system is often unsatisfactory because of the retirement of the expert (van Geet, Ebraert, & Demeyer, 2010). So lack of knowledge written down on paper. There may be little or no conceptual integrity of its architecture and design written down on paper

(Weiderman et al., 1997). It can be a problem if error happens, because tracing faults in the systems will demand a lot of time due to the lack of documentation and understanding of the internal workings of the legacy system (Paradauskas & Laurikaitis, 2006).

2.2. Legacy modernization

All those facts, characteristics and problems above are threatening and worrying all IT executives. The concept of Reliability-Availability-Scalability (RAS), which were the parameters for information technology application since many years ago, will work fine if nothing has changed in business model or business needs. However, relying on only those parameters are not sufficient enough to support dynamic business environment.

Today's business functions require more and better scalability, flexibility and manageability which are hard to achieve using the legacy systems. Enterprises cannot keep maintaining their legacy system. Not only because it becomes increasingly expensive, but also if the maintenance is outpaces the new software development, then at the end nothing will be left for new development. At this phase organizations will be trapped in a legacy crisis. When the limitation of legacy systems makes them difficult to maintain or to connect with current technology, then it may be the indication to start thinking about legacy modernization.

Besides ageing of the legacy system, lowering the cost of maintenance and responding quickly to market's demand are some of the drivers to modernize the legacy systems. However, in most cases, legacy modernization is driven by disgruntled lines-of-business, and shrinking resource pools to maintain them. Thus, in order to help an enterprise to react more effectively to changing business requirements and technological innovations, it is necessary to focus even on the legacy system evolution strategy (Aversano & Tortorella, 2004). Another driver that forces organization to modernize their legacy systems is changes in technology. For instances, the advent of universal network availability and distributed systems, standards and technologies. It will provide the current systems with better abstractions for code, and new models of interoperability that attract organizations. As mentioned in their paper (Maurizio et al., 2008) that IT challenge now is to integrate diverse systems into function rich business process.

In their paper, Zhang, Liu, & Yang (2005) suggested that any software artifact that was built using pre-SOA techniques is legacy. Thus, many existing pre-SOA system should be modernizing toward SOA. They investigated that modernization toward SOA might be affected by Service Orientation (SO) trend. Web service technology and Service-Oriented Architectures (SOA) are rapidly developing and widely adopted. What organization seeks the most from SOA is flexibility of the system. System with SOA architecture allows services to be accessed across a network via simple, well-defined interfaces, and (presumably) without concern for side effects resulting from dependencies between services. It makes many existing non-service-oriented software systems fall into the category of legacy systems. To survive from this environment, they proposed service oriented reengineering. However, whether modernization is toward SOA or not, the target system should also easily be maintained and adapted to meet future business requirement (Wu et. al, 1997).

Beside SOA as a target system, many organizations are starting to migrate their systems toward cloud architecture now. Cloud computing has different way of deploying system compared with traditional deployment. Cloud attracts many organizations because of its scalability, agility, high reliability, high fault tolerance, service-oriented, and etc. With all advance technology available, currently IT department are moving their legacy system to cloud computing without too much change in their legacy code and take the advantage of cloud computing. In general, .NET and Java still dominating the technology of cloud computing.

On the other hand, the drivers could also come from non-technical side. Frequently reorganization via mergers and acquisitions can push organizations to continuously optimize their internal organization (Khadka et al., 2011; Khadka et al., 2013). However, this is necessarily tied to the software systems supporting the organization's activities. If their systems cannot be modified or evolved anymore, then it is difficult for organization to move forward towards an agile enterprise (van Geet, 2011).

From a market trend perspective, there is a pressure from market or customer about "hand-held devices". Such situation is stimulating organizations to move towards the systems that are suitable to be used in a wireless network environment, where the network connection might be discontinuous or unreliable, the bandwidth and the available memory/mass storage are limited, and the user interface has a reduced size and limited capabilities compared to desktop computers.

As mentioned before that legacy systems hold crucial business logic, and often are seen as an asset that represents organizations' investment which grows in value as the years go on. Therefore, the ability of current technology to communicate with legacy systems is a success factor in business. The problems with modernizing the legacy systems become critical since the systems are tightly couple with legacy code written to build those legacy applications. Change or modify it will carries expensive expenditure and high risk of failure.

However, in the research performed by consulting companies such as Gartner and Forrester, all of them yield that 1) IT modernization is no longer a choice, it is an imperative, and 2) application modernization projects must align with stated business objectives and be supported by a business case (costs, risk, flexibility, etc.).

Legacy modernization seems to be a good starting point when exiting the current practices that cannot deliver the objective of business requirements. Legacy modernization tries to help legacy system evolve to a new technology/architecture and be able to cope today's business requirements when maintenance of legacy system is no longer able achieve it. But modernize system is not an easy job. It poses huge investment and high level of risk (Seacord, Plakosh & Lewis, 2001; Aversano & Tortorella, 2004; Nowakowski, 2012). The characteristics of legacy systems such as platform, language, architecture, lack original design documents, and the preferable target system may unexpectedly complicate the task.

2.2.1. The challenges of legacy modernization

While identifying the issue of legacy modernization, Brook (1987) in his work identifies 4 dimensions of developing software, namely (i) complexity, (ii) conformity, (iii) changeability and (iv) invisibility. Two of them (complexity and changeability) are typical problems in legacy modernization because of interdependence within the legacy systems and lack of resource (documentation and expertise) around legacy systems.

Geetha (2012), investigated the challenges during the migration of the systems. She divided the challenge into 2 parts, being technical and non-technical. The technical part covering, Usability, Software Development Service and Support, Security, Data Migration, Code Maintenance and Management, Strategy for Developing Migration Process Success. From non-technical side, the challenges are more from human factor such as Fear of the new software, Knowledge is power, Cost of training personnel for the new tools, Reduced productivity of the personnel.

The legacy systems have grown to the extent; altering one system will have an impact to the other, because of high dependency among the systems. Such circumstances make modernization more difficult and could cause disruption to the business environment (Weiderman et al., 1997; Wu et al., 1997). Newcomb & Doblar, (2001) also mention a risk of doing legacy modernization. They argue that legacy system typical have many dependency with its infrastructure which resulting in incompatibility with the new software environment. In addition, most traditional software are handcrafted, thus integration between systems with legacy system is non-trivial and only little systematic assets that can be reused in other systems (Weiderman et al., 1997).

The difficulty of legacy modernization is legacy applications are usually written with old programming languages like FORTRAN, LISP, and C. These languages are not object-oriented, hard to maintain and understand, and re-structure from code level directly. Moreover, because of historical reasons, many legacy systems lack up-to-date documentation, so the only valuable content is the legacy code itself. Such situation becomes an issue if organizations want to extract some valuable information, such as business entities and metadata, from legacy system (Zhang et al., 2009).

All the difficulties and challenges above are some of the factors that make world-wide organizations postpone their modernization project. There is kind of inclination from IT executive to still keep legacy system running in their organizations. Survey held by CIO-Insight in 2002 (CIO Insight Magazine, 2002) unrolled some factors such as:

1. Track record of the legacy system
2. Time and effort of training staff on the new system
3. Companies cannot afford to stop production systems to switch over to a new system
4. Companies cannot afford any potential downtime from a new system
5. Lack of time and/or resources to stop business and switch to new system.

From non-technical perspective, culture and structure of an organization do have an influence in software development. In their investigation, Xia & Lee (2004) try to identify how technology and organizational aspect can impact the result of software development project. Apparently, from their investigation on 541 information system development project of North American organization, they found out that technological aspect of developing software are perceived as most complex, but organizational aspects have a more significant impact on the actual performance and outcome of the project. Therefore, legacy modernization is not only about technology, but also about culture of the organization and the users or current developers/maintainers- issues that are often raised in academia (Khadka et al., 2013a; Khadka et al., 2013b).

2.2.2. General steps of legacy modernization

There have been many researches about legacy modernization in academia. Some of them produced approaches which can accommodate organization for executing legacy modernization. However, applying those approaches is different from one case to another case since the approaches will strongly dependent on the aim of the modernization or what the organization wants to achieve.

Before doing legacy modernization project, organization needs to have mature preparation and program management in order to avoid legacy modernization being failed. Depend on the objective of the organizations; many considerations need to be taken into account including a deep insight of legacy system and target system, methods of modernization and cost analysis as well as Return On Investment (ROI). Therefore, legacy modernization is not solely technology problem but also business problem (Khadka et al., 2013a). Seacord, Plakosh & Lewis (2001) analyzed that the preparation is not only about creating a budget and milestone, but also involves the order in which the functionality is going to be modernized. It should include market forces, business strategies and prudence approach that outline a total project benefit based on cost, benefit, risk and flexibility.

Another thing in legacy modernization is the data migration. It needs particular attention because legacy data objects have played a key role to keep permanent data in legacy system which often is stored in indexed file. The problem with this data is that they often not structured, thus they need more normalization and reformatted in order to suitable in new relational databases (Rahgozar & Oroumchian, 2002). The changes may concern the logical or the physical structure of data items as well as data tables. In their paper, some activities to normalize the data from legacy system are:

- Changing items format
- Adding or Suppressing items
- Splitting, Atomizing or decoding items
- Verticalizing arrays and matrix of items
- Merging, Mixing and Redefining items

- Splitting or merging tables

In general, many techniques start with legacy system understanding and target system understanding. These are two substantial preparations in most of the techniques in legacy system modernization (Khadka et al., 2013a). Understanding legacy system in crucial since there are huge business value residing in the systems. By having good understanding of the business value in legacy system, it will help organization build the new requirements for their new target system. To help organization do that, Aversano & Tortorella (2004), presented in their paper how to identify the business knowledge captured in legacy systems.

Understanding legacy system can be done by exploring the exact picture of information system. So reverse engineering must be accomplished in order to discover and extract as much as possible business knowledge from legacy sources (Paradauskas & Laurikaitis, 2006). Reverse engineering is the process of recovering knowledge from existing systems. As such, it can ease the modification of software systems to help them adapt to changing requirements (Tonella et al., 2007). It can also be seen as going backwards through the development cycle. It is often applied by taking computer program apart and analyzing its working in details.

The main goal of reverse engineering software systems is to gain a basic understanding of a system and its structure (van Geet, 2011). It can be also used in maintenance or to make a new program that does the same thing without using the original program (Meng, Qu, & Guo, 2013). The result of reverse engineering would be the knowledge of legacy system such as legacy source, including entities, relationships, application-specific meanings of the entities and relationships, business rules, data formatting and reporting constraints, etc. They refer this information as business knowledge. Business knowledge is needed in order to effective recover business value that residing in the legacy system (Nowakowski, 2012).

After understanding the legacy system and establish the requirements for the new target system, the various methods or approaches will be executed according to specific problems that each organization has. They could build the new system from scratch again; migrate their legacy system to the new environment; or wrap their legacy system so it can be interacted with other modern technology. Each of those is explained below.

2.2.3. Approaches in legacy modernization

In general, there are two options for organization to perform modernization namely 1) Create new software to take the place of existing systems, or 2) They can migrate current systems into a framework which utilizes current technologies.

The first option is called redevelopment (Almonaies, Cordy, & Dean, 2010; Stehle et al., 2008; and Zhang, Yang, Zhou, & Zhong, 2010). Redevelopment is one way of modernization where target system is built from scratch. The legacy system will completely re-create and replace with the new target system. At the end of the implementation, the new system will have new architecture, so it is easier to maintain, extend and develop the system after modernization. In addition, because the system is built without any extra layer, the performance loss can be reduced.

Nonetheless, build up the new target system from scratch is a lot of work. It also takes a long period of time from preparation through implementation (Weiderman et al., 1997; Newcomb & Doblar, 2001). Juric et al. (2000) add the disadvantages of redevelopment are big investment and the economic benefit is too far downstream. However, without investment in legacy modernization, the ability of organization to operate as a modern organization and serves its customers are at risk. In addition, redevelopment holds high risk to seriously contemplate a redevelopment approach (Newcomb & Doblar, 2001) because of inconsistency, cost overruns, and schedule delays in which makes this type of migration is less popular from business side. There are several approaches that fall in to redevelopment part. CORUM (Common Object-based Re-engineering Unified Model), CORUM II, MARMI-RE and OSET are all the methods belonging to redevelopment.

The second option is by adding the extra layer so the legacy system still can easily accessed or interconnected with current technology. Though effective and faster in implementation, this type of migration still holds the legacy system behind. One popular technique of this option is wrapping. Wrapping is when useful legacy code is glued with wrapper code to incorporate it into a target system. By wrapping the data, programs, application system, or interface are surrounded by extra layer so the legacy systems will have a "new and improved" look or improve the operation. Wrapping can be classified into UI (User Interface) based wrapping, data-based wrapping, and function-based wrapping. UI-based wrapping reuses the UIs of the legacy system in the new system by interface mapping. Data-based wrapping wraps the data structure of the legacy system, so the data of the legacy system can be used in the new system. Function-based wrapping uses component wrapping, object wrapping and gate wrapping etc. to realize the reusing of the service logic. These wrapping methods can reuse legacy systems for a short time, but they will increase difficulties in the maintenance and management of the new system (Meng, Qu, & Guo, 2013).

Another way of dealing with legacy systems is to migrate it to a more flexible environment. In their paper, Bisbal, et al. (1999) and Brodie & Stonebraker (1995) describe legacy modernization as a process when legacy systems are migrated concerned with developing to a target system while still retaining its functionality and data of the original legacy system. It means that even though the system is changed, the function of the new system should be qualified to fulfill the demand from the new requirements as well as what legacy system did before. Similarly Torchiano, et al. (2011) define software migration as an activity that concerns the transformation or adaptation of an existing software system towards a new technological context.

There are some domains that can be migrated such as, language or code in systems, operating system, data, user interface (UI) and architecture migrations. Meng, Qu, & Guo (2013) divide migration into two types which are component-based migration and system-based migration. Component-based migration classifies the legacy system into independent components, and then migrates the components singly. System-based migration integrates the whole legacy system and its data into the new system. There are some methods that are lying in migration's area. Those are Chicken Little, Butterfly, SGF and AGRIP etc. Migration merely suits for small-scale legacy systems, since it is more possible for losing information if the scale of the legacy system is larger.

One extra way to deal with legacy system is by doing refactoring. Refactoring is lying more into maintenance of legacy system. In their paper, Meng, Qu, & Guo (2013) investigated refactoring as a way to reuse legacy software efficiently. Refactoring is a process to reengineering the old software system by component technology. It can be done by following two steps, being reverse engineering and forward engineering. As explained before, reverse engineering is the process of recovering knowledge from existing systems by taking computer program apart and analyzing its working in details. On the other hand, forward engineering is the process similar to conventional development of software, such as requirements analysis, outline design, detailed design, testing and modification.

Bisbal et al. (1999) also give a nice picture of how those 3 ways of legacy modernization are impacting the legacy systems. It is depicted in Figure 2 below.

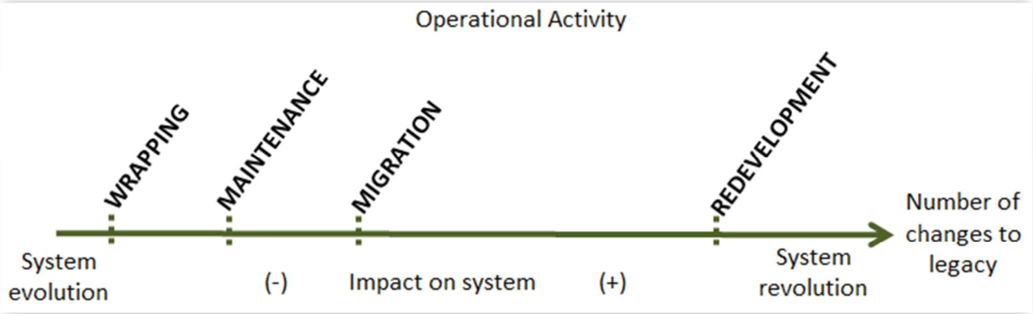


Figure 2. The impact of legacy modernization.

There are different between maintenance and system evolution. Wiederman et al. (1997) define software maintenance as a fine-grained, short term activity focused on localized changes. So the structure of the system does not change much and therefore, the changes only result in few economic and strategic benefits. The example of maintenance would be fix bugs and implement small functional enhancements, which can be accomplished by a dedicated service team, constantly reacting to change requests and bug reports (van Geet, 2011; Meng, Qu, & Guo, 2013).

On the other hand, system evolution is more into coarse grained, high level, structural form of change that makes the systems are qualitatively easier maintained. It complies with broad new requirements and gain whole new capabilities by making changes in architectural level (e.g. Service Oriented Architecture, Cloud computing, etc). Similarly van Geet, (2011) explains that system evolution incorporate large chunk of new or change requirements, which eventually create significantly different version of systems. With this the economic and strategic benefits will be increased dramatically.

Whichever option is chosen by the organization, there will be always challenges that can obstruct the modernization process. Thus, to have a better anticipation of legacy modernization, Venkatraghavan (2008) summarized some considerations to determine the right legacy modernization namely:

1. Cost of migration: all the cost spent for modernization such as, cost for hardware, software and professional service.
2. Effort for migration: all effort given to support modernization successful, such as, support from internal organization (Board, employees, etc.) as well as external (outsource professional service)
3. Elapsed time for migration: The total time started from preparation of modernization until evaluation as a last step of modernization
4. Legacy investment reuse: The choice of which parts of legacy systems can still be used and which parts are not be a part of new system.
5. User experience: The experience of user using new target system compare with their experience using legacy system
6. Business process change: The changes required to optimize the supporting of business process of the application considered modernization
7. Skill set: The skill needed in the future after new target application is implemented
8. Architecture and business functionality: The comparison of legacy application and new target application regarding these two parameters.
9. Dependency on legacy platform: Factor regarding whether or not the new target application still depend on legacy system in any sense
10. Risk mitigation: How company anticipates and deals with the risk during modernization process.

3. Grounded Theory (GT)

This research aims at exploring new insights and perspectives about legacy systems and legacy modernization from the industrial perspectives. Hence, the author follows the Grounded Theory (GT) method proposed by Glaser & Strauss (1967). They use the word *Grounded Theory* because at the end of this method, the new *Theo*ry will derive from data. *Grounded Theory* method consists of systematic inductive guidelines for collecting and analyzing the data to build middle-range theoretical frameworks that explain the collected data (Denzin & Lincoln, 2003). Because *GT* is carved from data, the result offers insight, enhance understanding and provide a meaningful guide to action. In addition, the result statements or propositions are insightful and rich explaining a phenomenon and in most cases, they can be used in future research as hypotheses.

With *GT*, data collection process and analysis data collection can be performed in parallel. These processes are continuously repeated until strong and solid conclusion is derived. In their book about *Designing a Research Project*, Verschuren, Doorewaard, & Mellion (2010) referred *GT* as a method of continuous comparison. The key of *GT* is that the author needs to develop the concept carefully because at the end the new insight will be produced. The result is a proposition that insightfully and richly describes a phenomenon (Seaman, 1999). Thus, the conclusion should be in line with data collection in order to have strong and solid conclusion. Every steps performed during the analysis should be captured in order to build a clear connection between data and finding. It means that sufficient information from each step of the study and every decision taken by the researcher must be presented (Yin, 2009). It is needed to build an understanding of derivation of the results from collected data because if the new insight or conclusion is not strong enough, it can be set aside easily.

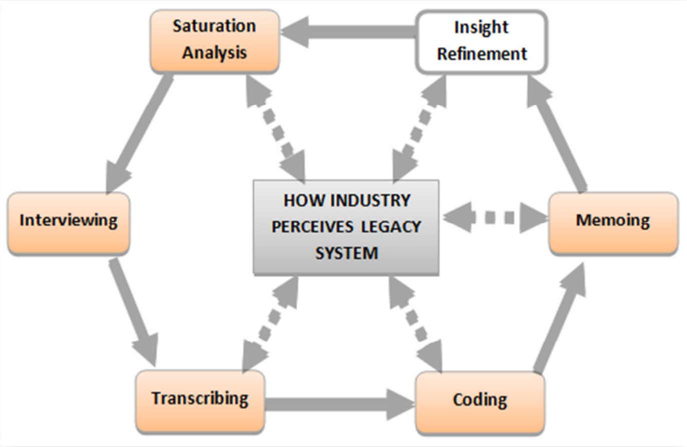


Figure 3. Grounded Theory method

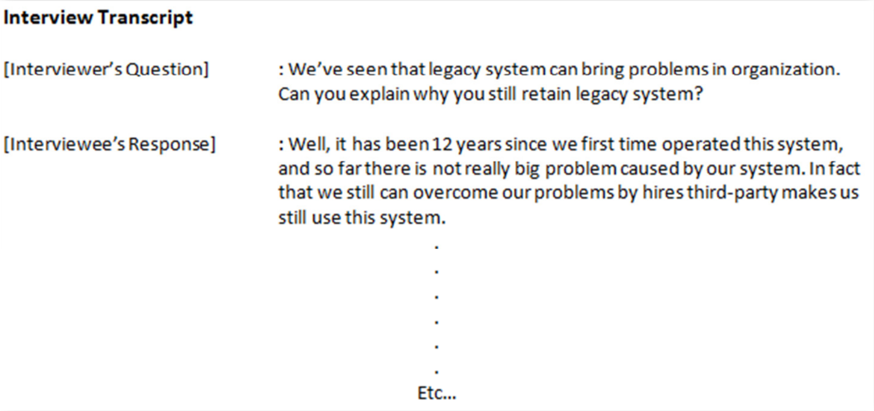
To apply *GT*, data received from interview is processed into several steps: transcribing, coding, memoing, Theory refinement, and saturation analysis (Adolph, Hall, & Kruchten, 2011). These steps will be repeatedly executed until strong and solid conclusions are constructed. But keep in mind that the conclusion cannot be derived directly from incidents happen in industrial area. They will be treated or analyzed as an indicator of phenomena that lead to the conclusion. Therefore, all relevant information (condition, action/interaction, or consequence) should be incorporated into the next interview (Strauss & Corbin, 1998).

During the analysis process, the evidences or relationships are going to be compared with each other and will be put into same concept. It will, at the end, form the basic unit of hypothesis or assumption. By continuously comparing the evidence, it will guard the author against bias by challenging his concepts with the fresh data. Such comparisons also help to achieve greater precision (the grouping of like and only like phenomena) and consistency (always grouping like with like) (Strauss & Corbin, 1998).

It is recommended to start the analysis as soon as the first bit of data is collected because it might be used for the next interview later. So in this research, *GT* is started by transcribing the data. The recorded voice is transcribed into text. Transcribing should be performed intensively because often the new ideas are made during transcribing. Thus, it is strongly recommended that researcher himself should transcribe the data.

3.1. Transcribing

The first step is literally transcribing all the conversation into text so it can be reviewed. Words, sentences or paragraphs will be written down and be arranged in chronological order according to who says what. The text will be consisting of not only the answers provide by participants, but also the question being asked by researcher. All the voices and sounds that come out from the recorder will the written down, so the conversation can be well and proper tracked. Figure 4 shows the excerpt of the transcribing process.



Interview Transcript

[Interviewer's Question] : We've seen that legacy system can bring problems in organization. Can you explain why you still retain legacy system?

[Interviewee's Response] : Well, it has been 12 years since we first time operated this system, and so far there is not really big problem caused by our system. In fact that we still can overcome our problems by hires third-party makes us still use this system.

.
.
.
.
.
.

Etc...

Figure 4. The example of interview transcript.

During this process some things might come up as an initial idea for hypothesis or new insight. Thus, this process is suggested to be done only by researcher himself. Although the initial ideas found in transcribing are immature and only are raw data but researcher can always take a look back of them later to support his finding.

By transcribing the voice into text, it can also help the author to have a helicopter view over different sectors (e.g. Financial, Government, Medical, etc.) in particular theme or concept by looking the most commonly words participant used in certain theme (tag cloud). Researcher can visualize the content of all the conversations and as a starting point to seek the information by looking on tag cloud. In addition tag cloud also helps researcher get familiar with the domain. Figure 5 shows the excerpt of the tag cloud of the legacy system domain.

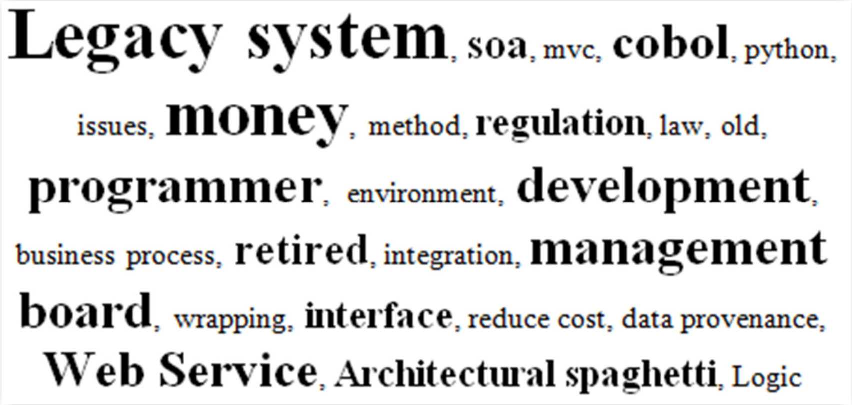

Legacy system, soa, mvc, cobol, python,
issues, money, method, regulation, law, old,
programmer, environment, development,
business process, retired, integration, management
board, wrapping, interface, reduce cost, data provenance,
Web Service, Architectural spaghetti, Logic

Figure 5. Tag cloud of legacy systems domain.

3.2. Coding

The next step after transcribing is coding the interview which has been transcribed. In this step, words, sentences, and even paragraphs will be broken down and assigned to a specific code. It means that one code can consist of many pieces of text from participants. The codes will represent certain theme, area, construct, etc. which are of interest in the study. Richards & Morse (2007) clarify that "*categorizing is how we get 'up' from the diversity of data to the shapes of the data, the sorts of things represented*". So, by coding, it allows researchers to gather all the related information under one theme together and leads to the development of the conclusion.

The idea of coding is to categorize similar information under same concept or theme. Each theme later will be examined to provide the explanations of phenomena. It brings something that real (raw data) into something that more abstract (*Theor*y). Coding allows researcher to gather all the related information under one theme / concept together leads to the solid and strong conclusion. Figure 6 shows the idea of coding until the theory is achieved.

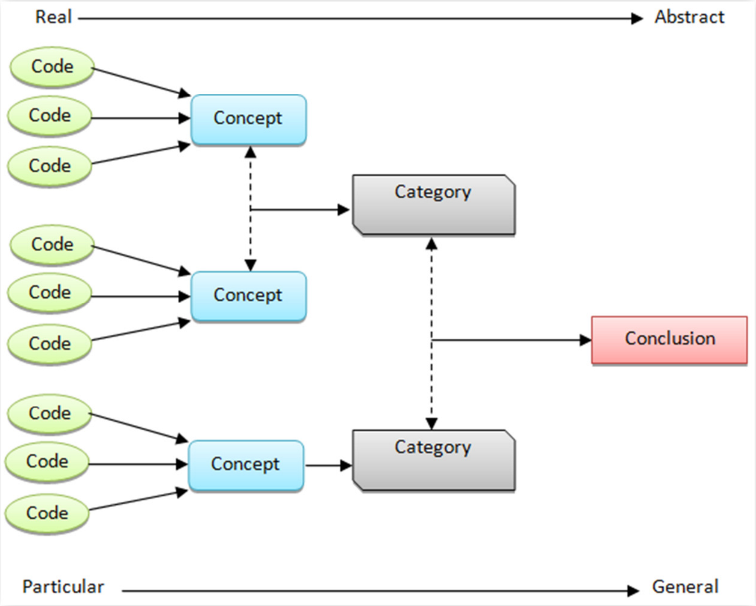


Figure 6. Concept of coding.

The aim of coding is to understand what the data say to the researcher. It can be done by interpreting the meaning of what interviewee have said, putting it under the same theme/concept and see how they related each other. For instance, if participants say "*We change our system because it is too slow*" and "*Our system often crashes during the weekend*" both of which could be coded into "performance". During the process of classifying relevant information, the codes will be added over time result in several themes / concepts. It makes easier for researcher to query or search for one particular theme and analyzes it to support his idea.

With everything in one place, patterns with respect to legacy system (e.g. similarities, causation, etc.) in industrial area can easier be found. Patterns, according to Hatch (2002), can vary from: 1) similarity (things happen the same way), 2) difference (they happen in predictably different ways), 3) frequency (they happen often or seldom), 4) sequence (they happen in a certain order), 5) correspondence (they happen in relation to other activities or events), and 6) causation (one appears to cause another). Later on, from patterns found in the process of coding, it will be interpreted and also can be displayed with chart so the researcher can have a hint of how it the end will looks like.

Coding is divided into 3 types: Open Coding, Axial Coding and Selective Coding. In open coding, phenomena are identified and will be broken down in standard ways of interpreting the phenomena reflected in the data (Strauss & Corbin, 1998). The broken code then will be compared with each other for similarities and differences. The code itself is a representation of certain theme, area, construct etc. It means that one code can consist of many pieces of text. The process of coding is mainly to categorize pieces of text which are relevant or similar to a particular theme that is of interest in the study. Each theme later will be examined to provide the explanations of phenomena.

In axial coding, the relationship between concepts is determined. Sometimes the relationship also exists because of causality between one concept and another concept. All relationships created deductively during axial coding must be considered provisional until verified repeatedly against incoming data. In this research it is called concepts which are how the author gets up to the higher level of codes. They are more abstract and more general from codes.

Lastly, selective coding is when all concepts are unified as a core category and categories that need further explication are filled-in with descriptive detail. The core category is representing the central phenomenon. At this point, the author should be able to answer such as questions, what does all the phenomenon seem to be about? How can I explain the cause of the phenomenon? etc. There might be categories that only stand to the core category as conditions, action/interactional strategies, or consequences. As a condition, the core category will be the characteristics and facts about legacy system (e.g. problems and Positive characteristics of legacy systems). On the other hand, consequences category will be the drivers of legacy modernization and the challenges in the process of legacy modernization.

3.3. Memoing

It is difficult to keep track of all the categories, properties, hypotheses that are evolving from time to time during the analysis. The author might put some comments or notes to the code that he created before. It is known as memoing. The idea of memoing is to capture or record the discoveries quickly before it get lost in the next interesting idea. In the other words, memoing is a way, the author record his thought and filling his idea about the comments from participants. Thus, it is basically a subjective perception from researcher about the data collected. In his book, Strauss (1987) argues that memoing is not just about idea. Memoing is a main key to formulate and revise the Theory during the research process.

The discovery can be anything, even incomplete thought, as long as it can be used to support the findings. By memoing the idea of researcher can be used against the facts that are found from interview and help researcher make a link between concepts.

Memoing also supports the researcher to look back why he makes certain decision and move to certain direction in the process of gaining new insight. Thus, it is suggested that memoing should be started with the first coding sessions and continues until the end of the research. The questions such as why category is separated or why code is separated, etc. could be answered by looking back in memoing. It prevents researcher gets lost of his trial of logic and prevent him guessing for the decision that he has made before. One approach that a researcher may use to capture his ideas is annotation.

The process of memoing is vital in *GT* because without memoing, the researcher will find it difficult to produce new insight which the final result from *GT*. Thus, it is suggested to start memoing whenever any idea come or whenever the researcher has an assumption about a certain comment on transcribed text. In addition, every time the researcher put a memo, it is suggested to write down the date and time for later stage.

When memoing is finished, researcher can start to derive his first assumption. It can be in form a phrases that are similar in different parts of the material, patterns in the data, differences between sub-groups of subjects, etc. The identified assumption could be combines with the next data collection later on since data collection and analysis are performed in parallel. During the iterative process a small set of generalizations can be formulated, eventually resulting in a formalized body of knowledge, which is the final result of the research attempt. The process of memoing will be executed iteratively and they affect each other.

Thoroughly analyze works about legacy systems and legacy modernization from different authors can improve the quality of memoing. It can be seen as the information source for the author to get better understanding about the topic of interesting. The better the author understand the topic, the better the quality of memoing. Therefore, the role of other researchers is necessary in this process to avoid subjective interpretation of the author.

3.4. Saturation stage

Saturation state is reached when no new valuable information is produced or nothing interesting information is found in respect to legacy system and its modernization, then the process of interview is stopped. At this point, the insights that are grounded in the data is produced and become a perspective about legacy systems from the industry side.

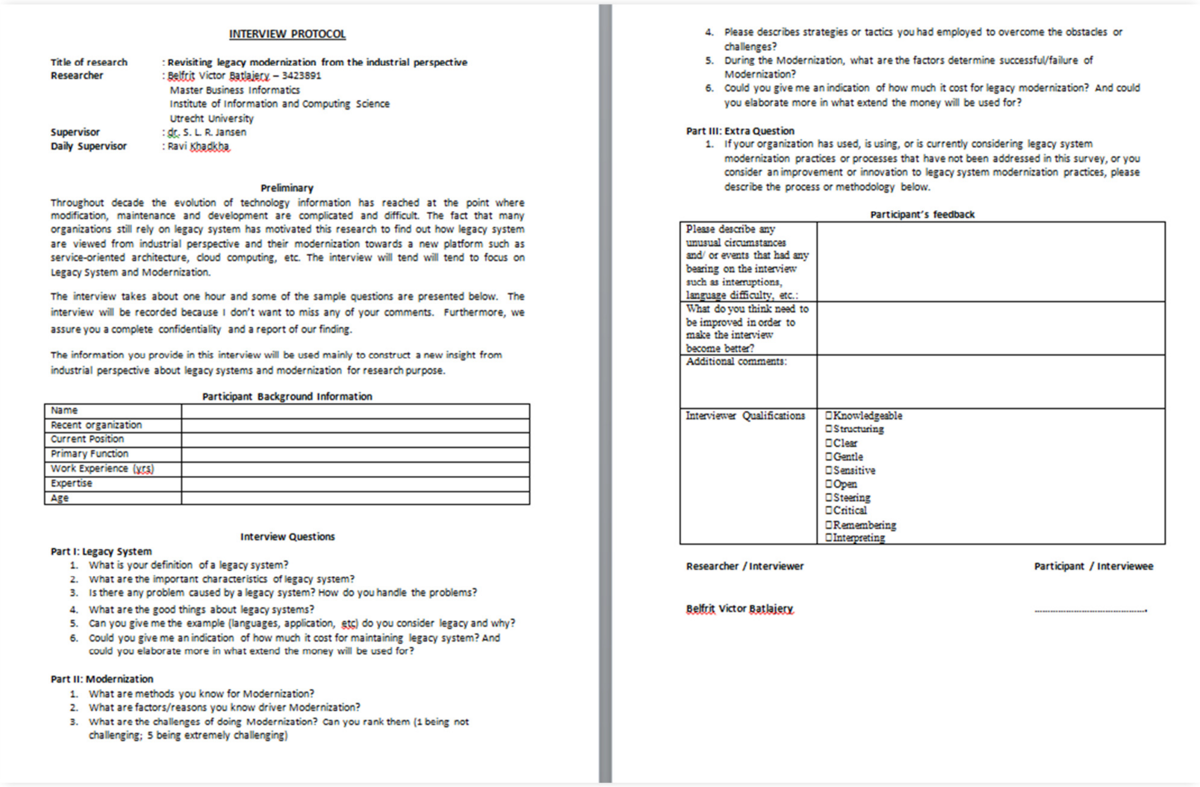
When saturation stage is achieved, the conclusion from *GT* will be validated with survey from broader audience. The objective of survey is to make the conclusion stronger so it cannot be put aside easily. The survey is done by using online survey tools, www.esurv.org.

4. Interview session with the experts

4.1. Preparation before interview session

In this research interview is performed to collect opinions or impressions about legacy systems and legacy modernization from professional in industry. The form of interview session is semi-structured interview. The author will construct multiple questions based on his knowledge, literature study, survey´s fact, and input from supervisor in order to get better quality of information from participants.

Prior to conducting interviews, an interview protocol is developed that provides information about the interview. The interview protocol is then sent to all the interviewee beforehand. Figure 7 shows how the interview protocol looks like.



INTERVIEW PROTOCOL

Title of research : **Revisiting legacy modernization from the industrial perspective**
Researcher : Belfrit Victor Batlajery – 3423891
Master Business Informatics
Institute of Information and Computing Science
Utrecht University
Supervisor : dr. S. L. R. Jansen
Daily Supervisor : Ravi Khadkha

Preliminary

Throughout decade the evolution of technology information has reached at the point where modification, maintenance and development are complicated and difficult. The fact that many organizations still rely on legacy system has motivated this research to find out how legacy system are viewed from industrial perspective and their modernization towards a new platform such as service-oriented architecture, cloud computing, etc. The interview will tend will tend to focus on Legacy System and Modernization.

The interview takes about one hour and some of the sample questions are presented below. The interview will be recorded because I don't want to miss any of your comments. Furthermore, we assure you a complete confidentiality and a report of our finding.

The information you provide in this interview will be used mainly to construct a new insight from industrial perspective about legacy systems and modernization for research purpose.

Participant Background Information

Name	
Recent organization	
Current Position	
Primary Function	
Work Experience (yrs)	
Expertise	
Age	

Interview Questions

Part I: Legacy System

1. What is your definition of a legacy system?
2. What are the important characteristics of legacy system?
3. Is there any problem caused by a legacy system? How do you handle the problems?
4. What are the good things about legacy systems?
5. Can you give me the example (languages, application, etc) do you consider legacy and why?
6. Could you give me an indication of how much it cost for maintaining legacy system? And could you elaborate more in what extend the money will be used for?

Part II: Modernization

1. What are methods you know for Modernization?
2. What are factors/reasons you know driver Modernization?
3. What are the challenges of doing Modernization? Can you rank them (1 being not challenging; 5 being extremely challenging)
4. Please describes strategies or tactics you had employed to overcome the obstacles or challenges?
5. During the Modernization, what are the factors determine successful/failure of Modernization?
6. Could you give me an indication of how much it cost for legacy modernization? And could you elaborate more in what extend the money will be used for?

Part III: Extra Question

1. If your organization has used, is using, or is currently considering legacy system modernization practices or processes that have not been addressed in this survey, or you consider an improvement or innovation to legacy system modernization practices, please describe the process or methodology below.

Participant's feedback

Please describe any unusual circumstances and/ or events that had any bearing on the interview such as interruptions, language difficulty, etc.:	
What do you think need to be improved in order to make the interview become better?	
Additional comments:	
Interviewer Qualifications	□Knowledgeable □Structuring □Clear □Gentle □Sensitive □Open □Steering □Critical □Remembering □Interpreting

Researcher / Interviewer **Participant / Interviewee**

Belfrit Victor Batlajery …………………………………….

Figure 7. Interview protocol before the interview session

In the interview protocol, the research motivation is introduced briefly and then question regarding the information regarding the interviewee is stated. There are also some questions about participants' background that they need to fill in. Some representative questions about legacy system and legacy modernization are also stated in the protocol. However, the interviewees do not have to fill in. To give an impression of the semi-structured interviews some representative questions about legacy system and legacy modernization are provided in the protocol. However, the interviewees do not have to fill in their answer since these representative questions are only the representative questions. These questions are just for illustrative purposes and will be asked in the interview session to get the detail information about legacy system and legacy modernization.

The last part of the interview protocol is the feedback from interviewee. How the interview session is conducted in interviewees' opinion. It is necessary to get their feedback/suggestions/critics in order to improve the quality of the interview for the next interview. So, an interview protocols are to be followed in paper and can be applied in almost all circumstances. It will be sent before an interview session to help interviewee have a better preparation about the topic discussed.

The procedure, on the other hand, is descriptions of how to go about performing a task. It is to be observed and can be altered or modified to suit to the specific situations. For instance, the steps the researcher will carry out during the interview session. Procedure will continuously be adjusted to the specific situation or condition but still follows the guidelines in protocol. If something needs to be updated in preparation, the next interview will use the updated procedure.

The role of interview protocol is vital before conducting an interview. It can help author as a guide when conducting the data collection and in that way preventing him from missing to collect data that were planned to be collected. The protocol also help the author decides what data sources to use and what questions to ask which lead to lower the risk of missing relevant data sources. Lastly, it can serve as a log or diary where all conducted data collection and analysis is recoded (Runeson, & Host, 2008).

After the time and place for interview are agreed by both sides, then interview session can be held. The interviews were performed in face-to-face meeting (direct meeting) at the office of participants or in the some places where both participant (interviewee) and researcher (interviewer) agree with. The interviewer and the interviewee will have a conversation for about 60-90 minutes depends on information flowing between them.

4.2. Interview session

One characteristic of semi-structure interview is open ended and specific question which is allowing and inviting a broad range of answers and issues from the interviewee's subject, or closed offering a limited set of alternative answers. It is also designed to elicit not only the information foreseen, but also unexpected types of information. All the answer will be acquired based on the experience of the interviewees. The questions were first planned but it is not necessary to follow the order they are listed. It allows improvisation and exploration in interview process.

During the interview session, the author will be equipped with interview guide. An interview guide is a non-formal form for help interviewer organizes an interview session. It consists of a list of questions to steer the interview session under different circumstances. In addition, interview is also used to collect historical data from the memories of interviewees about how they experienced with legacy system and legacy modernization.

The structure of interview is divided into several sections. First section is the introduction. In this section, interviewer will introduce himself and try to explain the purposes and the problems of the research. The aim is to avoid uncooperative manner from the interviewees because if interviewees do not understand the goals they are less likely to participate. The author will also ask about the permission to record the interview session and explain why it is needed to record the interview. In addition, the author and participant must agree about the confidentiality of the data recorded and about anonymously in the final report.

After that, a set of introductory questions are asked about the background of the subject. Question asked in this section are simple, just to build the same understanding between interviewer and interviewee. Participants are explaining about themselves and their career path in IT, particularly in regard to legacy system and legacy modernization. Participants are also allowed to ask the author if they have questions regarding the research.

The second section is main conversation. It is the most crucial and contains main questions that address to research questions. The questions will mostly about opinion and value of the interviewee based on his / her knowledge and experience. There are two parts of discussion in the second section, being legacy system and legacy modernization. The questions of legacy system are started with the question about definition of legacy system according to participants' opinion. Based on their definition, the following questions such as characteristics, problems, and example of legacy systems are asked. Furthermore, the questions in regard to maintenance, architecture of the system, etc are also being asked and recorded.

Upon finishing the first part, the interview is focused on legacy modernization. The interviewer has to confirm first whether or not he/she has an experience in legacy modernization. The questions about legacy modernization are mainly based on participants' story about it. For instance, the question about what the system they use with all the problems it has and what system they expect at the end of legacy modernization, the method they used together with the tools they utilized, etc. However, at the end of the conversation, the interviewer has to collect the information covering challenges that are arisen during the legacy modernization project and drivers for such a project as they are a part of the research questions.

Last but not least the conclusion section deals with a summary of the conversation. Remaining questions and answers are also allowed in this session to avoid misunderstanding in analyzing the data and reporting the result. In most cases, participants are asking about how the author will process the data, what tools are being used, what is the current status of the research, when they expect to get the result, etc.

4.3. Domain of participants

In total there were 23 interviews with 26 participants from various organizations across the Netherlands. The participants also came from different position such as Director of information technology (CIO), Manager of research and development, Manager Controlling, Manager of Infrastructure and Application, IT Project manager of legacy modernization, Software Architect, Business architect, and Consultant. The details of each participant are listed below:

Participant	Type of industry	Position	Years of experience
P1	Banking	*Business Architect*	24
P2	Information technology and Services	*Application Innovation services/ Bus App modernization*	33
P3	Airlines / Aviation	*Project Manager*	18
P4 and P5	Government	*ICT Manager and Information Architect*	12 and 15
P6	Consultancy	*SOA/ Integration Consultant*	5
P7 and P8	Software house	*Manager Technology & Quality, and Manager Research and Development*	35 and 15
P9	Security and investigation	*Manager Research and Development*	10
P10	Management consulting	*Management Consultant*	43
P11	Information technology and Services	*Chief Technology Officer Benelux*	27
P12	Information technology and Services	*Software Architect / Consultant*	15
P13	Food and dairy	*Head of Office of the CIO*	17
P14	Poultry	*Director of IT*	20
P15	Insurance	*CIO*	0
P16	Flower auction	*Manager I&A Service, Infrastructure and Applications*	10
P17	Banking	*Business Information Manager*	0
P18	Software house	*ICT & Logistics Consultant*	13
P19	Software house	*CTO*	15
P20 and P21	Information technology and Services	*Manager Business & Product Development and Service architect*	0 and 0
P22	Insurance	*Director Operations*	0
P23	Machinery	*Manager Besturingen (Control)*	18
P24	Consultancy	*Senior manager*	23
P25	Government Administration	*IT Architect Middleware and Integration*	0
P26	Government Administration	*Lead architect*	15

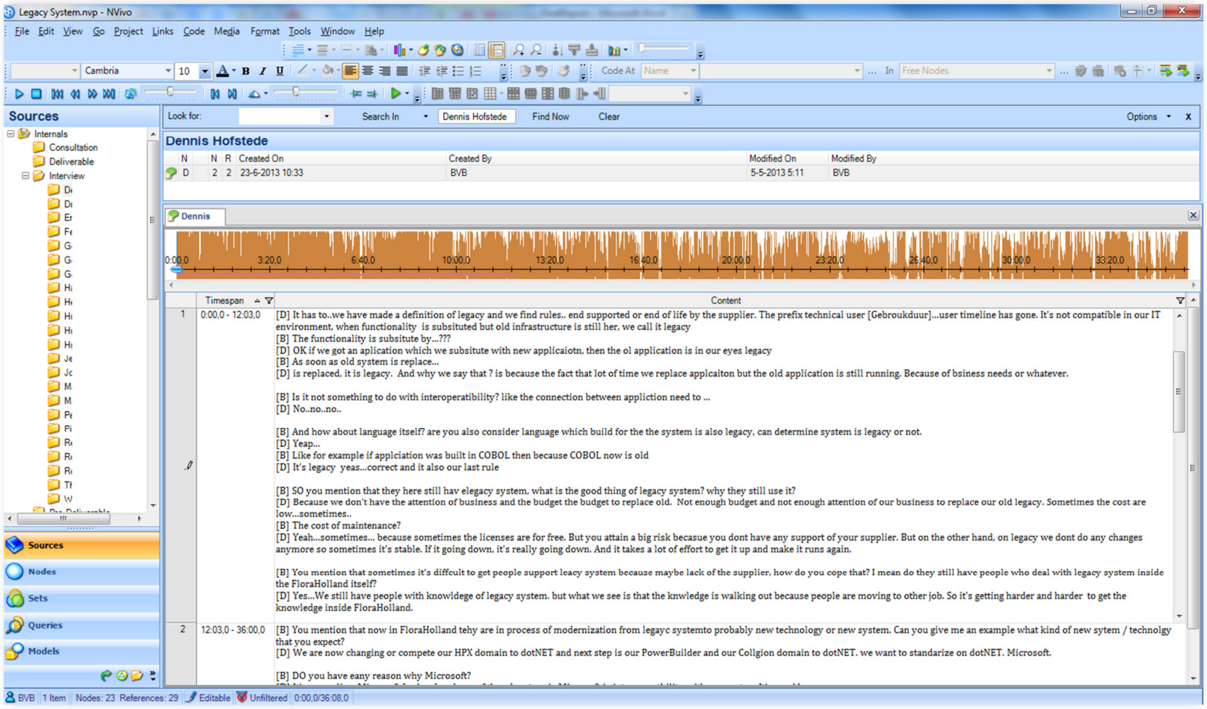
Table 3. List of participants for interview part.

4.4. Interview data analysis

4.4.1. Transcribing

The process of transcribing was started immediately after the first interview had done. The average hour for each interview is approximately 45 – 90 minutes. In total 23 interviews have been transcribed with 26 participants participate in interview session. The tools supported for the process of transcribing are Sony recorder and QSR NVIVO as a software application.

File in format of .m*P3* is recorded and stored in Sony recorder. The file later on was imported into QSR NVIVO 8 for further processes. Figure 8 shows the screen capture of QSR NVIVO.



Dennis Hofstede

N	N	R	Created On	Created By	Modified On	Modified By
D	2	2	23-6-2013 10:33	BVB	5-5-2013 5:11	BVB

	Timespan	Content
1	0:00,0 - 12:03,0	[D] It has to..we have made a definition of legacy and we find rules.. end supported or end of life by the supplier. The prefix technical user [Gebroukduur]...user timeline has gone. It's not compatible in our IT environment, when functionality is subsituted but old infrastructure is still her, we call it legacy [B] The functionality is subsitute by...??? [D] OK if we got an aplication which we subsitute with new applicaiotn, then the ol application is in our eyes legacy [B] As soon as old system is replace... [D] is replaced, it is legacy. And why we say that ? is because the fact that lot of time we replace applcaiton but the old application is still running. Because of bsiness needs or whatever. [B] Is it not something to do with interoperatibility? like the connection between appliction need to ... [D] No..no..no.. [B] And how about language itself? are you also consider language which build for the the system is also legacy, can determine system is legacy or not. [D] Yeap... [B] Like for example if applciation was built in COBOL then because COBOL now is old [D] It's legacy yeas...correct and it also our last rule [B] SO you mention that they here still hav elegacy system, what is the good thing of legacy system? why they still use it? [D] Because we don't have the attention of business and the budget the budget to replace old. Not enough budget and not enough attention of our business to replace our old legacy. Sometimes the cost are low...sometimes.. [B] The cost of maintenance? [D] Yeah...sometimes... because sometimes the licenses are for free. But you attain a big risk becasue you dont have any support of your supplier. But on the other hand, on legacy we dont do any changes anymore so sometimes it's stable. If it going down, it's really going down. And it takes a lot of effort to get it up and make it runs again. [B] You mention that sometimes it's diffcult to get people support leacy system because maybe lack of the supplier, how do you cope that? I mean do they still have people who deal with legacy system inside the FloraHolland itself? [D] Yes...We still have people with knowldege of legacy system. but what we see is that the knwledge is walking out because people are moving to other job. So it's getting harder and harder to get the knowledge inside FloraHolland.
2	12:03,0 - 36:00,0	[B] You mention that now in FloraHolland tehy are in process of modernization from legayc systemto probably new technology or new system. Can you give me an example what kind of new sytem / technolgy that you expect? [D] We are now changing or compete our HPX domain to dotNET and next step is our PowerBuilder and our Collgion domain to dotNET. we want to standarize on dotNET. Microsoft. [B] DO you have eany reason why Microsoft?

Figure 8. The process of transcribing in NVIVO

4.4.2. Codes construction

The process of coding was executed after the first interview has been transcribed. Quotes and codes are built up during the time depends on the comments given by participants in their interview sessions. The coding system in this research comprises 4 top-level categories, 28 subordinate concepts and 1-5 basic codes per concepts giving a total 50 codes. Each code has its own name and short description for further explanation about the code. Table 9 is showing the codes used in this research. The details of code constructions for each participant are presented in Code collection section in Appendix.

Category	Concept	Code
[1] TECHNICAL ASPECT	[1.1] MAINTAINABILITY	[1.1.1] Modifiability
	[1.2] RELIABILITY	[1.2.1] Robust
		[1.2.2] Stable system
	[1.3] USABILITY	[1.3.1] User Interface Aesthetics
		[1.3.2] Learnability
	[1.4] PERFORMANCE EFFICIENCY	[1.4.1] Time Behavior
		[1.4.2] General Performance is good
	[1.5] FUNCTIONAL SUITABILITY	[1.5.1] Functional Completeness
		[1.5.2] Functional Appropriateness
	[1.6] COMPATIBILITY	[1.6.1] Interoperability
		[1.6.2] Adaptability
	[1.7] SUPPORTABILITY	[1.7.1] Unsupported Supplier
	[1.8] FLEXIBILITY	[1.8.1] Not Flexible
	[1.9] PROVEN TECHNOLOGY	[1.9.1] Old system
		[1.9.2] Availability
	[1.10] RISK	[1.10.1] Risk of running legacy system
	[1.11] ARCHITECTURE OF THE SYSTEM	[1.11.1] Large multiple systems
		[1.11.2] Database issue
		[1.11.3] Back Office
		[1.11.4] Complicated system

		[1.11.5] Poor design and ecosystem
	[1.12] KNOWLEDGE AROUND SYSTEM	[1.12.1] Knowledge become scarce
[2] BUSINESS ASPECT	[2.1] BUSINESS STRATEGY	[2.1.1] Doesn't fit with your future strategy
		[2.1.2] Business Critical
	[2.2] FINANCE	[2.2.1] Expensive in maintenance
[3] DRIVER	[3.1] TECHNICAL DRIVER	[3.1.1] Become flexible to support changing business requirements
		[3.1.2] Limited supplier or vendor to support the continuity
		[3.1.3] Lack experts around the systems
		[3.1.4] Prone to failure
		[3.1.5] Become flexible to support changing business requirements
	[3.2] BUSINESS DRIVER	[3.2.1] Create new business opportunity via mergers and acquisitions
		[3.2.2] Reduce cost of maintenance and operation
		[3.2.3] Faster time-to-market product
[4] CHALLENGES	[4.1] TECHNICAL CHALLENGE	[4.1.1] Data Migration
		[4.1.2]Lack of resource (e.g. documentation, expertise, etc.)
		[4.1.3]Poor system architecture or infrastructure (E.g. monolith and, hardcoded, spaghetti architecture)
		[4.1.4] Difficult to extract business rules or knowledge
		[4.1.5] Difficult to test
	[4.2] BUSINESS CHALLENGE	[4.2.1] Funding legacy modernization project
		[4.2.2]Predicting Return of Investment (ROI) of modernization
	[4.3] ORGANIZATIONAL CHALLENGE	[4.3.1]Cultural resistance in organization not to adapt new system
		[4.3.2]Resistance from the current users or maintainers in the organization
		[4.3.3]Communicate the reasons or consequences of modernization
		[4.3.4]Difficult to effectively prioritize the functionality for modernization
		[4.3.5] Time constraints to finish legacy modernization

Figure 9. Categories, Concept and Codes used in the process of coding

4.4.3. Characteristics of legacy systems

Although it is difficult to define legacy system in general, each of professional in industry has their own criteria to define the legacy systems. Those criteria are described below.

Old system

There was no doubt from participants that legacy system is the old systems (16 participants). In general, there is no straight definition about old system. However, most of participants agree that by old system, it means more than 10 year. It is easy to recognize the legacy systems by looking at their age, because most of the legacy systems were built in 1970 or even earlier. At least 5 participants agree that the age of the legacy system is more than 20 years. *P11* explained, "*There's a lot of legacy systems that are 20 - 30 years old already*". *P23* also said, "*Software of almost 20 years, hardware of almost 20 years that's still working*". Some organizations even have a record to preserve the system for 30 years, like what *P26* said "*It has some age. In [XXX] we have system over 30 years now and still using until now*".

When participants refer to the old legacy systems, they are not only old hardware, but also old software as *P13* explained, "*Legacy system for me is a combination of hardware and software which was built long time ago*". They are old systems that run in old technology like *P4* and *P5* mentioned in their interview session "*That's obsolete technology*". Thus, it could be a single piece of software application or multiple systems working together forming an IT ecosystem. Even though they are old in age, but they are performing their functions well. *P7* and *P8* explained "*It's not the question of quality of service of the application their mainframe. It's a date originally from the 70ies or 80ies. But are still running and they are functioning well*".

Knowledge around the system

When participants were asked about the characteristics of legacy system, 24 participants agree that knowledge about the system is becoming scares. It is one of the common characteristics that participants often associate with the legacy systems. The meant of knowledge around the system is the availability of knowledge, information, discourse, and people about the legacy systems. It could be lack of documentation and lack of experts (e.g. developer, programmer, analyst, etc.) of the legacy systems.

Lack of documentation. Documentation of the legacy system is perceived bad in legacy system (*P1*, *P2, P7* and *P8*, *P11*, *P12*, *P19*, *P20* and *P21*, *P14*, *P23*, *P26*, *P4* and *P5*). All of them support the comment from *P26* "*We have no documentation, or it is not updated*". Same with *P7* and *P8* with their comment, "*The documentation is not...if any documentation is not actual. The knowledge of the people who built it is gone because people are gonna away to other job*". In fact, from interview sessions, none of participant ever mentioned that their legacy system is properly documented.

The problem of documentation is often started immediately at the moment when the original developers built the systems. *P2* explained, "*The moment it is going to production...the documentation is outdated*". His argument explains that when the system in the production phase, documentation is slowly not being updated and within 3 years or more, the documentation is becoming obsolete and not actual anymore.

Somehow, people who worked with the system, have often ignored the documentation. Somehow undocumented is a part of legacy system. *P11* said, "*So the characteristic also from old legacy system that is the lack of documentation, because it was never documented in those days*". They are getting older and even more difficult to ask them to documenting their work now. Some experts even do not work in the organization anymore, so create the documentation is much more difficult nowadays as *P10* mentioned "*Then you saw that only few people who could understand how you could do that (documentation)*".

Lack of experienced manpower. Besides lack of documentation, lack of experts around legacy system was also often mentioned in the interview sessions. It was recorded that all the participants said the same comment about the existence of the experts around the legacy systems. As *P11* said in his interview, "*More or less not that many knowledge of that system in organization because the people who are knowledgeable about this kind of system are already gone*". They are gone for many reasons such as retirement or they have another job in other organizations. *P23* mentioned, "*We have couple of programmers around 45 - 50 and one even older 85, where able to support those PASCAL software*". Same voice with *P7* and *P8,* they explained "*But the problem is our COBOL application is that the people works on it are retiring. So they are leaving the company then...So our knowledge is going away about the system*".

Even if there are experts available in market to help organization dealing with legacy systems, not all of them understand specific thing. For instance, if organization wants to find a developer / programmer to build an application, it is difficult to find a person who can understand typical legacy languages and techniques. *P19* with his experience in software development explained in his interview session that, "*But how many you can find that who can understand really cryptographic algorithm where you could do all kind of bit shifting, hashing and those kinds of things*".

In addition, finding new experts for legacy systems and replacing those old experts is becoming increasingly difficult now. New young people such as students are reluctant to learn about legacy systems. *P24* said "*it is difficult to get people from university*". Not only *P24*, but *P22* also shared the same problem, "*We can't find the people to keep it going*".

Monolithic architecture

The interview sessions reveal that legacy systems were mostly constructed in monolithic design. *P1* explained, "*The way they were programmed, the way they were designed…ok…because most of them were monolith*". It makes sense since earliest system had been built with all technology limitation in the past such as large, monolithic, and hardcoded which were often running on old mainframe machine.

With regards to old legacy system, most of the participants associate legacy systems with the old mainframe machine (12 participants). Old mainframe by its characteristic is not only monolith but often hardcoded too. *P18* said that "*The program is really hard-coded*". *P2* with his comment also said the same. He explained, "*They have a lot of legacy applications and it is hard coded often in the code*". *P6* also explained, *"That's especially hard coded, in the logic to find it, to change it, it is hard*".

Besides monolith and hardcoded system, legacy system is also recognized as a big and large system. *P16* pointed out, "*Because legacy is rigid, is big*". It is mostly resulted from multiple systems do the same thing as *P6* explained, "*Within history they first had this system and then a few years later they thought o yaa...now we need this system. So they have multiple systems for policies*". Same comment from *P2*, "*Lot of banks and insurances and ending up with 3 or 4 systems overlapping functionality. Because they cannot decommission the old one*". *P16* agreed by saying, "*The fact that lot of time we replace application but the old application is still running*". Moreover, when looking on the code of the legacy systems, it is often written in one piece contains thousand lines of codes. *P24* also supported it with his experience, "*Something like 30 or 40 years of programming*".

At the end, the system becomes big and complicated. *P7* and *P8* said, "*So there's always something with technical complexity, a lot of components, a lot of layers, lot of programs and interfaces interacting with each other*". In addition, complicated system could be complication in rules and logic as *P20* and *P21* said, "*There is so much rule and information in that engine*", or complicated in connectivity as *P15* said, "*You have application landscape which is to complex… since most of the products were complicated*". *P6* also said, "*They had a legacy system but during time they also have complementary systems like workflow system. And using both and using all top to do little bit here, little bit there*".

Maintainability

Maintainability is a main benchmark for people in industry to assess whether the system is legacy or not. According to ISO/IEC 25010, maintainability is the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. It includes corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications. It is also covering installation of updates and upgrades. The response measure for maintainability is modifiability which is defined as degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality. It includes the labor or clock time required to make, test, and deploy the modification

Maintainability was often mentioned during interview session with all participants. Most of them said that maintaining the legacy system is difficult (*P1, P2, P6, P7 and P8, P9, P10, P11, P12, P13, P16, P17, P19, P20 and P21, P22, P26, P4 and P5*). As *P7 and P8* said "*But the problems is with the technical quality, the maintainability is for several reasons bad… simple change takes so much time because no one knows exactly the system works*". *P10* also said "*They trouble maintaining it, adjusting it*".

The difficulty could be resulted from the architectural technical complexity of the systems or how the systems were designed in the past. In most cases, logic in legacy systems is closely integrated in the system. *P12* mentioned in his comment, "*It may be constrains by technology, for instance it is built on technology that is no longer maintained*". *P6* also said, "*A lot struggle to change that, because all these logic about that is totally integrated into the system*". *P18* also supported it by saying *"Because the program is really hard-coded...it is not configurable that makes it difficult*".

Lack of documentation makes organizations have to depend on certain people that still understand the technology about legacy system. It brings difficulty to maintain the system since those people are becoming scarce as *P1* commented "*You become dependent on scarce set of skills and specific people and that can make them more difficult to maintain*". *P13* also agreed through his comment, "*Because it is old and the knowledge is not available, nobody can do (adjustment) something like that*". *P19* also said the same, "*Now that is the legacy. You won't find anyone who can maintain that*". Not only that, when new people arrive in the organization, it will take long time before they understand the systems. As *P1* commented "*System when you get somebody new in it would take a lot of time when these people, if you can get them at all when these people have to learn how the system works and how they can maintain it*".

In the future, maintainability will be coming even worse than today. *P10* gave his prediction that, "*I think it will be harder and harder to change the system*". *P16* also said the same, "*I think the system is un-repairable in the future because of the lack of the parts of the machinery and because of the lack of the* *knowledge regarding of the software*". And also *F17*, "*Upgrading and getting to the higher level for supporting the organization in your business, your legacy system is getting more difficult*".

Compatibility

Compatibility of the system is determined by two aspects, being interoperability and adaptability. As mentioned before interoperability is degree to which two or more systems, products or components can exchange information and use the information that has been exchanged. On the other hand, adaptability is degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.

When the conversation was about compatibility of the legacy systems, the participants had almost the same answer. Adaptability of the legacy system is considered poor according to 11 participants in their interview sessions (*P1, P2, P7 and P8, P9, P16, P19, P23, P18, P25, P26*). It is started when the environment of the legacy systems is evolving and left the legacy systems behind as *P1* said, "*They operate within an environment. If the systems stay the same, and the system surround the systems, the environment is evolving to new technologies you get a discrepancy and you get less possibility to properly interconnect with the older system*". *P16* also said the same, "*It's not compatible in our IT environment*". It is a problem too as *P26* told his problem in regard to the legacy systems in his organization, "*It's a big problem, because lots of software were created for the platform XP, doesn't work for platform window 7*".

The interoperability in legacy system is also poor. A part from adaptability, it is difficult to connect a legacy system to the other systems and share resources together. *P2* said, "*That is hard to align that with an old system of course*". Legacy systems from the old days were often equipped with the old or obsolete connectors (*P2*P, broadcast or multicast) and old interface. As *P9* said "*So that's all old stuff and then you have to have a legacy system because nothing else can talk that stuff again*". *P11* gave his reason that, "*The second one is system integration. Because old date system has most of more or less have old application program interfaces. So there is more or less a lack of knowledge about integrating legacy systems by itself.*"

With those properties, 15 participants agree that the interoperability of legacy system is perceived bad. It is supported by *P1* saying "*So the problem of legacy system I think are not in the system themself but in the supporting technology*". However, if legacy system needs to be connected to the other systems, one challenge emerges is a conversion as *P11* said, "*Another issue or challenge could be system conversion also from software perspective*". To minimize the compatibility's problem, standardize system is often a solution for the organizations as *P4* and *P5* said, "*Yeap…it is difficult. We try to standardize interface*".

Supportability

Support from vendor is also a consideration to determine the systems being legacy. Supportability for legacy system indicates how good the supplier or vendor support the legacy systems. It includes support for software (e.g. patches, security update, service pack, etc) and hardware. For some organizations, both software and hardware they use are produced by other organizations (suppliers / vendors). They are users of certain product provided by third party, because it is not feasible to build their own software application or hardware by themselves. According to the interview sessions, 14 mentioned about supportability toward legacy system. All of them have the same voice saying the suppliers or vendors of legacy systems are lesser and lesser supports their legacy systems.

When suppliers do not support the software application, it means that no patches, security update, service pack, etc available for the application anymore. *P7* and *P8* explained "*System is built with program languages, library and frameworks that no longer support or the suppler no longer support*". *P26* also shared the same opinion, "*If we continue then we don't get the security update*". As a result, the maintenance becomes difficult as *P11* said, "*It is hard to maintain, because beside there is no real updated software anymore and it is also not supported by a lot of other organizations*".

Not only for the software part, but also for the hardware. *P18* explained, "*System cannot longer be supported by software or hardware… Hardware that are not support anymore...environment*". Thus, *P12* even define the legacy system, "*I would define a legacy system as a system which is constrain by hardware, specific hardware which is no longer fully supported or being scarce*".

When there is no support from supplier, organizations will be in a danger situation. Risk of continuity of the system is growing bigger as *P2* said, "*If the technology supplier is not support it anymore, then you also have a risk*". *P19* also said, *"If the technology supplier is not support it anymore, then you also have a risk*". And *P18* said, "*Because machines that are not supported any longer, is a great risk for business*". *P14* shared his experience, "*We have to ask the vendor please develop that part in your software and make it as a standard feature*".

It is a problem for organizations which still use the systems provided by supplier and really depends on them. As *P25* said "*And at the certain point [XXX] say that they don't support that system anymore. So we picked up the whole operating system and tag it on top of [XXX] and related the whole environment. We've been running that for years. And finally we had to change that system since there's no support anymore*". Therefore, organizations need to take a next step to deal with their legacy systems or it will be too late as *P12* said, "*If the legacy system is depend on specific hardware, then if you cannot stock it up on that hardware, then you completely relying on its functionality. If it is break down, then you don't have a choice and it is too late*".

Flexibility

Flexibility is defined as a degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements. From flexibility point of view, organization will consider their system as a legacy system if the system is less flexible. It is supported by 12 participants. *P9* explained, "*And if...it is legacy system behind it then I can imagine that you are less flexible presenting the data than newer systems*". *P6* also shared the same opinion, "*Yeah... it is not flexible. It's a lot of hassle…they also have a lot of limitations on a flexibility*".

One of the participants, *P26*, gave the real example of inflexibility of the systems in his organization. They have the system which requires its users to login before they use it. If the system is not flexible enough then the users need to login several times separately, mainly because each login's system represent different system. It is not user friendly and users often unsatisfied with it. If the system could be easily interlinking each other and quickly accommodate new requirement, that will create high flexibility of the system.

Inflexibility of the system often appears when a new requirement with complicated business logic is coming. It is really difficult to make legacy system comply with a new requirement. *P19* said "*It's always new rules, regulations, common labor agreement, and thing that changed between taxes, so that's always something changes. So there, legacy is really painful*". Therefore, legacy system is not flexible enough to accommodate the new requirements as *P22* explained, "*No it is not flexible enough to do the things you wanna do*". So, it makes *P15* defines the system is flexible if it that support the future requirement, "*Support the future requirements. So that's flexibility*".

Inflexibility of the legacy system is one of the reasons why organizations want to remove their legacy system. *P16* explained, "*One of the other things why we want to get rid of the legacy because it is …it is not flexible*". One of many solutions that organization prefers is to move their system to Service Oriented Architecture (SOA). *P2* said, "*Like SOA is easy, it is more flexible*". *P14* gave his reason, "C*an make it pluggable like plug and play by having SOA environment available*".

Usability

The usability of legacy system is not so good according to 12 participants (*P1, P2, P6, P7 and P8, P19, P20 and P21, P18, P26, P4 and P5, P6, P18*). Users often not satisfy with the interface of the legacy systems. *P6* explained, "*Interface...user interface is not ...is not modern...you get this back…yeah this old screen with* *some data on it*”. It is quite reasonable for the systems that were built at least 15 years ago which no advance graphical implementation yet. Consequently, some users find uncomfortable with the system as *P26* said, “*Not easy used by the employees*”.

Usability of the application can also be measured by how good the application can be used by specific users to achieve specific goals of learning to use the product or system. However, the legacy application is not self-explained, meaning that it is really difficult to learn how to use it. *P6* said in his interview “*Sometimes people don’t know about it...they have this...I tried to explain it last time to someone who was taking something over from me. It's like, I don’t get it*”. Same with the *P2’s* experience. He said “*New one comes in, he has to learn about it...well...it is not something you can do in just a few months*”. It will take months before the person can familiarized (her)himself to use the systems. *P20 and P21* also said the same thing, “*If you come here as a new developer, it would be difficult to do anything inside the piece of the system*”.

However, some users are familiar already with the system. Thus, despite bad interface of the legacy system, they still can use and operate it well. *P4* and *P5* said, “*People are used to the system. They know how it works*”. And *P19* also support them with his comments saying, “*Yeah...They know that they have to not touch certain key or if you are in that screen don’t touch that button because then your screen will die. Or if it goes wrong just do this step and then you back on tract*”.

Risk of legacy system

Risk in legacy system can be one of the threats for organization. Eight participants (*P1, P12, P16, P20 and P21, P14, P25, P13*) mentioned about high risk is one of the properties of legacy systems. When he was asked about risk of having legacy system, *P14* explained, “*Can mean the risk of higher cost, can mean risk of losing people that go away that you don’t have knowledge*”.

The biggest risk haunting the organization is the risk of continuity as *P13* explained “*Continuity. That the biggest problem. Because it is old it will fall down*”. *P14* also said the same, “*So it is more about continuity of the legacy system where the risk is*”. *P16* also agreed with his comment, “*If it is going down, it is really going down. And it takes a lot of effort to get it up and make it runs again*”.

Also the risks become bigger if supplier does not support legacy system anymore. Old technology tends to have limited amount of suppliers. If there is no support anymore from supplier, it will be a real problem for legacy system to operate again if it failed. *P16* explained, “*Big risk because you don’t have any support of your supplier*”.

In addition, when the legacy system is a core system, maintaining it has high risk of failure too. If inappropriate changes happen in the core system, it can make core system stops operating. *P25* expressed his idea “*We want to change but it is hard since it is the core of our...all the people in the Netherlands are in it. We cannot afford if database corrupt or to make any changes that can affect the continuity*”.

With this risk, *P1* argued that “*Money is not a largest problem...risk is the largest problem...you know, they can afford these systems to failed only for one hour*”. *P13* also argued the same think, “*Basically the risk could be the legacy system falls down, our production process is a stop*”. Therefore, *P12* argued that risk is strong driver through his comment, “*The risk can be a strong driver*”.

Reliability

Reliability of legacy system was perceived well by professional in industry (*P18*, *P20* and *P21*, *P1*, *P6*, *P12*, *P15*, *P2*, *P7* and *P8*, *P16*, *P17*, *P19*, *P14*, *P23*, *P4* and *P5*, *P11*, *P25*, *P26*, and *P24*). *P18* support this statement with his comment “*It’s reliable...people know how to use it for long time...all the problem has been disappeared from it…so technical problems are usually are not there*”. Also *P20 and P21* said “*So the good think is it is still running, it is working, it is petty reliable, pretty good*”. It runs pretty good as *P1* said, “*Let's see small thing that functionality that count is a stability, robustness, reliability and availability of this system*”.

One factor that determines the reliability of the system is robustness. Robustness is an ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc. Legacy system is perceived stable as *P6* said, "*They don't crashed often*". *P12* also agree, "*They have been around for many years and during the period of those years they have been stabilized*".

Not only robust, legacy system in the perception of participants is stable (11 participants). All of them support what *P1* said, "*So they're well performing and they are good secured and they're stable*". *P7 and P8* agreed through his comment, "*The functionality is stable...just let it stay as it is*". The stability could be also because of the legacy systems are often located in back office, as *P16* said, "*On legacy we don't do any changes anymore so sometimes it is stable*". *P23* also argue that most of the time the old technology is more stable in his comment, "*Often the old technology is more stable than the new technology*".

Performance

The performance is measured based on time behavior and functionality of the system. Generally, nine participants agree that performance of legacy system is perceived good. *P24* even stated, "*Performance is never an issue in legacy system, at least I've never seen it*". Not only him, *P26* also support his comment with "*Yes... performance still OK. There's not a real problem*".

The good performance of legacy systems is supported by good functionality they have. Functionality is the ability of the system to do its job for which it was intended. ISO/IEC 25010 and ISO/IEC 9126 define functionality almost in the same sense as capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions. The functionality in the legacy systems is perceived good. Quote by *P14*, "*The performance is enough*". It makes user get what they want from legacy system, as *P19* said, "*So the performance as expected*".

Good performance of legacy system is mostly supported by good functionality it has. *P7* and *P8* explained, "*So the functionality is on average is OK*". It is mostly driven by basic functionality it has, as *P1* said, "*The first involve IT was looking at an automating tasks so it is a normal history of automation at first. Handling all kind of standardizes administrative stuff which can be for easily formalize and automated*". *P12* also agrees through his comment, "S*implicity it has, for instance user interface… simplistic because it keeps out all the unnecessary stuff*". Also *P15* commented the same thing, "*But not supportive in lot of calculations or validation of the data*". So, legacy system completes its task well as *P1* commented, "*A legacy system is a system that works. OK, it is there because it is there and it is old because it works and it works fine*".

Organization also experienced that the response time of legacy system is generally fast. As *P11* mentioned "*It is also characteristic about response time, it is fast because normally it is simple small data messages, so there's not that a lot of overhead*". The response time is measured satisfy because legacy system does automating task without heavy process thinking. It is driven by its design, monolith architecture is good for processing a huge number of data as *P2* said "*But On the other hand, it does the job is efficient, it is fast and that's more less if talk about back office system*". It is particularly for mono-hardware such as mainframe machine, because mainframe is good for I/O intensive system. In addition *P19* also argue that "*old code went as amazing speed on new hardware*".

Availability

The availability of legacy system is perceived good (*P6*, *P11*, *P1*, *P17*, *P14*, *P23*, *P24*, *P4 and P5*). Availability is defined as a degree to which a system, product or component is operational and accessible when required for use. In total, there are 9 participants agreed on it including *P6*, saying "*They always available*". His argument is also supported by *P11* saying that, "*They are available, they are long time online and they are more less 24/7 up and running*". Furthermore, *P1* explained in his comment "*Let's see small thing that functionality that count is a stability, robustness, reliability and availability of this system*".

In addition, old age of legacy systems make them available to support business task for many years. No wonder *P17* said, "*It's proven technology most of the time*". *P14* also said that, "*It has proven, it is stable but it is also has proven*". And *P23* gave his reason that, "*That can stay over a couple of years that way, so you can say that the old technology is proven technology*". In addition, legacy system is often located in back office, as *P7* and *P8* said, *"I think another characteristic of legacy system is that they are mainly back office system*". So it is more stable compare with mid-office and front office system.

Business critical

From the perspective of how crucial the system is for organization, some of those old systems are being used since the organization was born. So, they could be considered as a core system of the company, such as the systems that support critical business processes (administrative business such as payroll/personnel, accounting/budgeting, and purchasing). Core systems means the systems that support critical business function or contain vital features and it was mentioned by nine participants (*P1*, *P9*, *P10*, *P11*, *P12*, *P14*, *P25*, *P4*, *P6*). They are valuable systems and they have great business value. The disruption or malfunction of them will cause a failure in business operation and can have serious consequences such as financial and legal problems, damages and other penalties.

P24 in his interview session mentioned why he thinks that legacy system is a business critical. He explained that, "*rules have been a key for the survival for the company. Because they have been there 30 years, so they really the foundation for the survival of the organization*". *P6* also had the same argument, saying "*I think it is a core system, most of the time…like this core transaction*". So they are core of the business of the organization as *P9* mentioned, "*legacy system in the core of business*".

Such type of system has brought profit for organization as *P11* explained, "*It is an old system, it is an obsolete system maybe but it is useful and has a business impact still and generate a lot of revenue for banking and their clients, because it is still a lot of legacy system in the core system*". However, for software provider or message service provider, the existence of legacy system is their business area. *P7 and P8* said, "*In fact some of our legacy systems are the most profitable of our system*". Their customers still like to use their legacy system because the legacy systems still bring them profit. Also by *P20 and P21*, "*Because as a company, we are successful because of all differences between legacy systems, because if all legacy system talk and communicate with the same protocol, then we wouldn't exist*".

Cost of operation and maintenance

From 26 interviews, 20 participants agree that the cost to maintain legacy system is expensive. Many of participants shared *P11*'s opinion, "*The cost of maintenance, the maintenance is perceived high*" as same as what *P4 and P5* said, "*Maintenance is expensive...too expensive*". In particular, when participant associates legacy system with mainframe, the cost to operate and maintain it is expensive as *P2* said, "*Cost of mainframe is perceived as high. And they want to migrate way to migrate away to mid-range systems*". Same voice with what *P17* said in his interview session, "*But the maintenance cost of the AS400 was sky high so we need to get rid of the system and really turn it out*". One of the reasons is because of hardware to run the system is expensive as *P7 and P8* said, "*The hardware on which the system runs for example mainframe becoming too expensive compare to cheaper hardware nowadays available*".

The old architecture of legacy system with old framework also contributes to the high cost of maintenance. *P9* explained, "*Their framework doesn't support...they do something strange with enabling the web part of that application. And change that is a lot money*". Therefore, organization has problem to keep updating their system. Another reason is because lack of people who can maintain the systems, "*The amount of people who know that legacy system or the environment is build are limited*" by *P9*.

License of legacy system and total cost of ownership can also make the maintenance and operation cost becomes higher. Software licenses and cost of hardware are too high for organization. *P11* explained, "*The cost of maintenance, the maintenance is high. So that will increase the total cost of the ownership of the system*". He shared his experience that "*License cost of the software. That is also expensive*". No wonder, high cost is one of the common drivers to modernize legacy system. Thus, by modernization *P16* said "*We now migrate it to .NET and our license fee is drop for something like half*".

4.4.4. Drivers for legacy modernization project

Inflexible system

Although legacy systems perform their job well, they are observed not flexible. *P5* said, "*It did its job, but less flexible*". In general, all participants who see legacy system has a problem with flexibility tend to use this problem as a driver of doing legacy modernization (*P1*, *P2*, *P9*, *P12*, *P13*, *P16*, *P22*, *P14*, *P25*, *P15*, and *P6*). They shared *P25*'s comment "*So then they need to rebuild the whole system to get more flexibility*". *P1* also agreed with it since by doing legacy modernization, it can improve the system's flexibility to cope the new requirements, "So*, we are able to more quickly adapt the certain components on their own*".

Most of the time, inflexibility comes when there are changes in functionality or when there is new requirements that can force the organizations to do changes in their legacy systems. *P7 and P8* explained, "*Definitely, business requirement changing overtime*". *P19* also had the same opinion, "*If they come up with new kind of rules, of regulation and you have to come up with additional text then it has the deferent way of calculation, you have to go in and build new type of rule*". Therefore, *P12*'s comment, "*They expect more flexibility when they want to change their business process*".

The requirements can come from many aspects, such as economic, law, politics, even just from customers. *P20 and P21* explained, "*There is a government body called NICTIS, who decide the standard for messaging. And that's one driver if they upgrade the standard, then we have to migrate too*". Implementing a new requirement is not difficult if organization have a proper IT infrastructure as *P15* said "*We need to implement as an insurance company, quite a few legal requirements. If you don't...if you have a complex application landscape, you need to implement all legal requirements in all systems. If you have less systems, then it is easier and costless*".

In addition, to achieve flexibility, some organizations look into Service Oriented Architecture (SOA) as a solution. *P2* with his comments said "*Like SOA is easy, it is more flexible*". Organizations see the SOA more or less as Lego as *P14* said "*Can make it pluggable like plug and play by having SOA environment available*". *P12*, who see SOA is suitable for his organization also said "*They expect cheaper and faster ways of implementing those changes*".

Business opportunity

As mentioned above, legacy modernization is not technology decision as *P22* commented "*it is not a technology problem. Because the technology people, do not decide whether or not, we are making the shift or not. That's a business decision*". His argument had been supported by 16 other participants during interview session. *P14* put the same argument, "*I think every IT decision should be business case. So basically every decision I made, is always oriented on business case*". Those arguments are reasonable since legacy modernization must have something to do with money, thus, *P11* also mentioned, "*It's always the financial decision*".

One opportunity is to grow bigger via merger and acquisition. *P13* expressed his opinion, "*OK now it is time to stop that and we going to move to the new world. Like merger or acquisition or reorganization or something like that*". On the other hand, merger and acquisition is also a solution to mitigate the risk. As *P17* said "*Bank is too big, you have to split it because otherwise the risk is too high if it is fall apart and the governance has to carry all the consequence*". However, this phenomenon must be supported by good IT system as well so they can merger with minimum changes in their IT system as possible. So, having good IT system is what organization expects to accommodate them open their business through merger and acquisition.

The opportunity can also come from customer demand. *P2* explained, "*People really want instant correct information. And that might drive some companies to...well...to do modernization...yes...to enable the* *legacy application to cope with those kind of information request*". Same with what *P25* said, "*Higher demand also from the public to the system that [XXX] has. The public want immediately response*". *P15* also gave his argument, "*Customer don't like it anymore. So the human experience, the user interface which is old fashion*".

Furthermore, the development of the new technology is kind of opening the new opportunity for business too. Sometimes, when new technology is introduced, previous system cannot cope with that. So the new feature cannot be implemented in the systems. For instance, *P20* and *P21* said "*New opportunity, if you think the market changes, new message format or new way of connecting*". Eventually, the old systems will be called legacy systems.

High cost of maintenance

One of the most common drivers to modernize the legacy systems is cost reduction. 20 participants, who mentioned that the cost to maintain and operate legacy system is high, use this reason to modernize their legacy system. *P17* explained, "*When you look at the cost of the system and new innovation if a system stop with innovation and the cost getting higher because maintenance getting more expensive, maybe you should think of change*". So do *P7 and P8*, "*So the driver that the maintenance of legacy system causes a lot of money*".

Most likely because this driver is money oriented. *P18* even explain that money is the major aspect in legacy modernization, "*From [XXX] is money. It's all money and to reduce the cost of this thing and maintaining the IT*". Organizations often do not satisfy with the way they spend their money for maintenance and operation of legacy system. Many organizations experience 80% of their total expenditure goes to maintenance of legacy system and only 20% goes to innovation. *P24* with his experience in as a consultant in many organizations confirmed that, "*Most company claim that they spent 80% of their budget on running and 20% of innovation*".

Although, the process of legacy modernization is also not cheap, at one point the cost for maintaining and operating the system for the long term is higher compare with the legacy modernization project itself. One indication for doing legacy modernization is when the cost of maintenance and operation seems over exceeding the cost for modernization itself. As *P7* and *P8* mentioned, "*So one of the main factor is the cost to do changes becoming higher and higher, compare to the change itself*".

It means that there is a trade-off to keep maintaining the legacy systems and to modernize the legacy systems. As *P9* said *"the effort to build the functionality is so high that is not a good business case. So if for instance we have to put in another data field in the system and that takes 4 weeks to program then yeah…you probably say well…sorry the system doesn't support that functionality anymore…sorry…you have to…we have to start modernization again*". When organization arrives at this point, then legacy modernization is the best option.

License is also one factor that can make organizations spend a lot of their money. One way to achieve low cost in license is through standard software. So, sometimes organization uses this reason to modernize their legacy system as *P13* said "*We want to do move toward a standardized way of working in the entire company. So standardize from business process point of view*". The same voice with what *P12* mentioned, "*But if you can move to standard product, then it could be usually an advantage because the maintenance cost for standard product is usually lower*". *P14* also said, "*So if you look at the cost, I'm quite sure I can run on similar environment against lower cost. If I would run a standardize…let say state of the art ERP environment…because I don't need those developers*".

Lack of support

As it is written above, one of the characteristics of legacy systems is having limited supplier. Such characteristic is also a driver for legacy modernization because it could also resulting problems in which will disturb the continuity of the system. *P19* said, "*So, even [XXX] stop supporting it, then you are stuck* *because then you get a new hardware...you don't have a driver anymore for the chipset or, so you have to figure out how can install the platform and install the application in the platform*".

Limited of supplier usually come because the application has been reached the end of its lifecycle. So, as *P1* said to continue product lifecycle of the system "*Also continuity, so the driver behind staying current with this technology is that you want to have continuity of your core system*". Continuing software lifecycle could be done by providing patches to stabilize the situation of legacy system as *P17* said "*There's no patches*". However, as supplier stop their product, organization needs to find another way to keep supporting their system.

If organization needs to change something in their legacy system and they cannot find it through the original supplier, the will try to find it to the other place. As *P2* said, "*I mean what you trigger or also things like for instance a hardware. I've seen customers running systems on hardware that run out of support. And even hardware that if something is broken down they have to look on E-bay to find spare parts*". Such situation is of course not an ideal situation for any organization. Thus, this drive mainly to make sure that the systems still can be supported.

Lack of experts

As we know that lack of the expert is mostly the problems of legacy modernization, this for sure is the driver for legacy modernization. *P1* said, *"They knew they were getting pretty close to the border where you hit a brick wall and trying to get someone with knowledge on the system and to extend them or to amend them*". *P10* said the same, "*I think the big problem that you can't find people to understand them and understand the technology*". Also *P18*, "*The problem with the language like C is that there's no... less knowledge in the market right now*".

Those situation make *P2* also predicts that this scarcity of knowledge is not only today's problem, but will be worse in the future, "*Knowledge erosion is the biggest problem there*". It is supported by some of the participants (*P1*, *P2*, *P7 and P8*, *P12*, *P19*, *P4* and *P5*, *P15*, and *P6*). They share *P25* thought, "*It's going to be a problem for few futures*" and *P26* thought, "*When they are not there we cannot help new employee to work with it*".

Moreover, organization expressed themselves that it is difficult to find new people now. *P22* said "*because you cannot attract people that they have to maintain it or it doesn't cooperate new kind of technology. That's really bothering us here*". Therefore moving the application to the current technology can help organization overcome this difficulty. People on the market now are more familiar with all current technology, not with old technology anymore.

Risks are growing

Risk is one of the drivers to modernize legacy system. For some organizations, risk is even stronger than any other problems. Risk is a huge or strong driver because most organization cannot afford if their system goes down as *P1* mentioned "*Money is not a largest problem...risk is the largest problem*". One example by *P13*, "*We have an old ERP system, old almost 10 years old. And it drives the production in the plant and also the logistic and warehouse and also the order towards the customers. If that system stops, the plant stop, the warehouse stops*".

Risk of failure can have the massive impact on business and that is why it is a strong driver, although not often mentioned in the conversation. Risk can be built from problems such as support from supplier as *P2* said "*So when your environment runs out of support then it is really dying and if that's true then you are already late*". Thus, if suppliers do not support the system anymore, organization cannot do anything if the systems fall down. If the risks keep growing, organization might not sure if their system still can support their business or not as *P1* explained, "*It currently works but they worry if they continue it, can continue it long enough into the future*". Therefore, if it is too late, then organization might go out of market because their business is stopped. *P6* explained, "*I think the company will go out of market*".

However, risk is often can be seen before it becomes bigger. Therefore, if the risk is not big yet, organization might feel comfortable with their legacy system as long as it still benefit them. *P20* and *P21* in their comment *"We also have legacy but we don't get rid of it, because it has no big risk yet*".

Faster time-to-market of product

Another driver is aiming to satisfy the customer or customer demand, such as quicker time to market. With new era of technology now, customers are quite demanding compare with customers from 10 years ago. They assume that with the advance technology as we have now, they should be able to get the response from the system quickly. As *P25* said "*Higher demand also from the public to the system that [XXX] has. The public want immediately response… They wanna have the calculation from [XXX] about the amount that you have to pay or you receive back. Why shouldn't [XXX] response immediately. So demand of that will be higher than before*". Such motivation leads organization which provides information for their customer started to do modernization to satisfy their customer.

Faster time to market can be seen as a driver from business side as *P14* said, "*Business driver is time to market*". Also by *P22*, "*We need a faster time to market, and we are not able to do that in COBOL environment*". In addition, for software house, by doing legacy modernization, they hope to deliver their product to their client faster. As *P7* and *P8* mentioned, "*To launch quicker new product for the customers*".

Although, this driver is not as strong as other drivers, it can be one of the many reason for executing legacy modernization. Because it has to do also with priority this driver is not on the top priority as *P7* and *P8* give an example of back transaction "*do you consider when you do electronic banking transaction that as a legacy system? Well not, because it is a modern user interface. It looks flexi on the mobile device. That's what interests our costumer*".

4.4.5. Challenges in legacy modernization project

If organization has decided to do modernization, there are some challenges that they have to confront.

Data Migration

Some migration issues are common to all software engineering projects and are widely researched and supported. These include target system development, testing, and database model selection. Same with the finding in interview session, among all the challenges in legacy modernization, from technology point of view, data migration is the most challenging one according to 12 participants.

P10 even admit that data migration is more complex than re-write the code. "*You have…if you migrate the data from the legacy system to the new system. Often that's also a big issue...it is even more complex than the writing the code*". It is so complex that can be the project by its own. "*You want to migrate it completely with all its problems or do you want to improve on the quality as well? But if chose improve on the quality you may end up with the whole new project on your hand as well. Because improving data quality is…can ...well...be large project on its own*" by *P12.*

The difficulty could also be come by the nature of old database. *P17* explain "*Normally legacy system does not support the good databases or does not have relation database model*". For instance, *P6* experienced "*So, it could like sometimes there is empty column for the account on a name…I've seen strange thing, it is just missing for maybe like 5% of the record some really vital data …they are just missing*". Also *P10* said "*The database issue. They have same issue with database management and they want to P14w that*".

In addition, the difficulty of data migration could be also emerging because of lack of awa*P14*ss by the organization about it. As *P2* experience "*And at the time they have a new system they want to migrate all the data from the legacy to the new one. And then they find that...oouww...but here we have something...ouww...we haven't had foreseen that*".

All of the difficulties above make *P3* suggested to perform the data migration as soon as possible, "*That's why we started so early with data migration. We did prevent the data migration actually more than 2 times* *a month. So at the moment we thought that we had covered all the data, we did all kind of validation*". And, *P14* suggested in his comment "*Strongly believe if you are doing migration of your legacy, you have to prepare a good data migration strategy*". It is believed to be a successful factor in legacy modernization.

Lack of resources

Another challenge is the lack of resources either the documentation or the experts of the legacy systems. Lack of resources, especially documentation create a big risk in legacy modernization as *P20* and *21* said "*If documentation is lacking then it is a bigger risk to migrate. Because you don't know what's going on in the old system naturally, make the risk to migrate bigger*". Especially, if the project is the first project ever performed. It makes the process of modernization becomes more difficult as *P16* said, *"Sometimes we are the first example in the world for some systems. So there's no reference*".

Therefore in legacy modernization, it is suggested to perform reverse engineering to get the requirements from legacy system, as *P18* said "*Documentation...yeah...for new system but also for the old systems. See can we reverse engineer the old system and document them well so we know what the requirements are and...Because then if you have the requirements, you can say well…we can build new system*".

From human point of view, working with people from different domain is a challenge. Project manager have to unify all of them so they can work together as a team supporting each other. Therefore, commitment and dedicated team is a factor in legacy modernization project as *P2* commented, "*Get really commitment for that level...from that level to really set up the whole period. And not well…OK next year...o well...we have less budget...well we stop and we quit*". However, it is difficult to get dedicated team since people usually come and go away in organization.

In legacy modernization involves variety of experts as *P13* said, "*You do involve the business, the user, the owner, to test, to monitor ,to see if the application is working*". However, sometimes, they have different opinion which can lead to the pro and contra in handling the problem. For instance *P18*, "*That makes it difficult. And the problem is that skip has another view of the [world] than the government*". Failed to facilitate this can cause delay of the project. The problem of different perspective could be a challenge in legacy modernization project.

Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)

Poor IT landscape is often a problem for legacy modernization as *P11* said, "*I think the architectural technical complexity*". At least, 7 participants find the complexity of the structure of the legacy systems is a problem. *P19* also said the same, "*But usually systems are so closely integrated and complex*". For instance, as *P2* said, "*And what you also find is that one application is developed that it uses also the database of another one. And it is not through a normal interface but it is via back door to get some data over there*".

Poor of architecture, as we know, is resulted from previous inappropriate maintenance as *P9* said, "*Not only the development part, but also the architecture I think, in the system as well. Because if people are bumping against architecture changes or misguidance in earlier architecture*". The mistake from the past often make IT landscape of the organization becomes worse. For instance, *P10* experienced, "*Sometimes they think it is impossible, so let's do…let the new system do the new things, and the old system do the old things. Then they have 2 systems overlapping*". He also gave his example that "*[XXX], the insurance company, you know that? They have, I think 4 or 5 different life insurance systems. Because every time they build new life insurance system, modernization of the old, but they find out that they cannot move the data from the old system to the new. So they do the new business in the new system, and old system...and the insurance is still in the old system*".

Moreover, for organizations who want to migrate their system into SOA architecture, they find it difficult to deal with connectivity between the systems. *P7* and *P8* said "*So there's always something with technical complexity, a lot of components, a lot of layers, lot of programs and interfaces interacting with each other*". So *P14* said "So one of the challenges is how do you migrate all your interfaces which are point to point instead of SOA".

Difficult to extract business rules/knowledge

In order to make a new system performs as good as the old system, the knowledge / rules / requirements need to be extracted. It is crucial as *P18* said, "*Yes...you can only do that if you know exactly what the thing does. That is the hard part*". It is seen by 7 participants as one of the challenges in legacy modernization. *P20 and P21* said that, "*To extract all the rules and details in there is really difficult*".

P12 expressed his experience "*The company or the project team has to extract exactly the internal functionality of this legacy application. It can be difficult to extract it and to document it and to implement it properly*". Also by *P9,* he said, "*Legacy system also tends to hide complexity of business process*".

Therefore, *P18* mentioned that this is the main success of legacy modernization, being get the clear requirements (requirement before and after). "*We try to get the functional requirements clearly, what should the new system do*". One way to extract the knowledge in legacy system is by doing re-engineering. That is why *P14* suggested that get a clear requirement is one of the success factors in legacy modernization. He said, "*Having good requirements…Requirement before and requirement after*".

Difficult to test

Testing is one of the difficulties in the process of legacy modernization according to the six participants. For instance, *P20 and P21* said, "*Regression test that cover enough all the continuality*". Also *P13* argued, "*Regression test that cover enough all the continuality*". The difficulty can also result in high cost of testing as *P15* said, "*Transformation of old system to a new system is costly since you need to test it carefully. And testing is costly*".

So, it is challenging to get the test case completed as *P12* said "*But you have to also test it. And it can be difficult to extract all the use cases, the original system has supported, and because that difficult also difficult to get your test case complete. And it can require a lot of work and comparing functionality from legacy system to the new application. It can be time consuming work. And it difficult to get it complete*". Therefore, sometimes organization hires external parties who have all the experiences to test it. *P26* said, "*We hire people form [XXX], consultant, architect. It's too big for our regular IT staff to do it, so we're hiring a lot of experts from companies like [XXX], [XXX] to help design new system and help develop the new system, test it.*".

In addition, *P2* shared his experience regarding testing the systems, "*So you have primary system in current technology and side by side you have a new system the converted system on the new technology. And we tried to do the same load for both systems*".

Funding legacy modernization projects

Legacy modernization is not merely about technology problem, but also business problem. So it does deal with money that will be spent for the process of legacy modernization. Thirteen participants shared their experience regarding this difficulty. However, get the money from top management is not that easy as *P10* said in his interview "*I think top management doesn't understand the issue...no…they don't give budget for it*". Top management usually expect to get their Return On Investment quicker as *P2* reported "*But what you now see is last year I had customer who said I want to it but I want to earn it back in 6 months. If you have talk about a project of at least one year you cannot and never earn it back in 6 months*". If CIO cannot do that, then funding will not be granted.

Besides internal difficulty, there is also external factor. *P2* said, "*Because you see now...because of the economic situation, they don't have the budget to set up the program just to rebuild the functionality.*" As a result, *P15* expressed in his opinion that in past days, it is easier to spend money in innovation of a new system as he said "*I think that in the past, the cost of operation and the cost of IT were not really a problem. So, innovation in a new system was quite easily done with the culture. So that's quite easily money for a new system or a new inoovation*".

Another problem is track record of IT department. One of the participant said, "*If you look at IT in companies, success is not so big. Yeah…They are not so successful for most IT people in the company. They often start project, it takes longer, it cost more, it doesn't work as advertised".* So organizations are not satisfied with their IT department.

Predicting Return of investment (ROI) of modernization

Legacy modernization is a complicated project, thus it is normally taking number of years to be completed. For some participants, predicting the ROI is not easy. One remark from *P2*, "*They are always looking for a short Return on Investment. One you put the money in, they want to earn it back. But what you now see is last year I had customer who said I want to it but I want to earn it back in 6 months. If you have talk about a project of at least one year you cannot and never earn it back in 6 months*". *P18* also said, "*And a lot of companies also say they have their return of investment. They say it can be 1 year and then it should be done. well...I don't think return of investment of 1 year*".

Predicting ROI is crucial since it determines the funding of the project. So, *P2* suggested to have good business case, "*So you always have business case over there. And the moment that business can yeah...so it is not positive and for many companies if the Return on Investment is not received one year and say we don't do it*". It is perceived difficult since it is difficult to predict the IT trends as *P9* said, "*And the amount of changes is going through the legacy system wasn't really foreseen at the moment we built it*". *P14* also shared his difficulty in making the business case, "*Are you willing and then you have to calculate the risk of the how if it can be happen. And then you make a decision*". Thus, *P17* argued, "*I think maybe the acceptance of the business people that they won't accept the newer system or hard to change*".

Cultural resistance in organization not to adapt new system

Culture of organization has to be considered before the project of legacy modernization is started. 10 participants agree that this is one of the most challenging factors. As *P11* said, "*Change that the organization is ready for the change. Is the culture in the organization, do they allow a change for that kind of different information*". *P17* also said the same, "*We also change the culture of people and how they should work and report and everything. I think it will take 3 years to get that kind of transformation*".

In participants' opinion, to adapt with the new technology is difficult. *P12* said, "*Sometimes it can be a bit difficult and to let that go and start working on new technology and new system*". By using new technology, organization also has to change the way they work and not all of them like it. *P26* said, "*Culture yes...people are used to do thing in a way, when you are changing that, well that's you have to think about it, how can we change the behavior about people*". Also *P11* said, "*Sometimes people do not like to change*".

In addition, training about new system is necessary after the process of modernization. *P22* said, "*Yes it is, because then you are discovering that not everybody is going to be able to use the new technology ... you train all kind of people. More than hundred people*".

Resistance from the current users/maintainers in the organization

People, who work with legacy system for long time tend to protect the system. As *P12* said, "*Sometimes they see the legacy system as their baby and they tend to know every aspect of it. And sometimes it is quite difficult because you are in the team coming from the outside to perform project and they consider this as their primary area*". *P19* said "*They like they safety zone and they only use the system because they have to get the job done and get how at 5 o'clock leave the office, go watch football, something like that*". So, this can be the reason why people do not like to move to the new system as *P15* said, "*So that can be a reason why people are a little bit less reluctant*".

The other reason is has to do with job security. Some people only have ability working in legacy system. *P6* explained, "*Because what's the need if we have new system, which is working not with COBOL, who's gonna need me anymore, so they ditch me after it is done. So then they think like hei...why should I cooperate*". So they are not really agreed with the process of modernization. In order to mitigate this risk, *P9* gave his solution for not cooperate user and maintainer "*So the most crucial thing is, I think to make sure that the people you have in your company don't...aren't link to a specific system but have another goal, maybe maintain multiple systems or build some other application whatever*".

Difficult to communicate the reasons/consequences of modernization

Communication is vital element in legacy modernization at least for 11 participants. Communicate with top management about legacy modernization is, for some people, are difficult. *P12* said, "*It's really difficult, but you can only do it when it also part of the assignment. Because sometimes they don't really feel a need to change the culture*". The difficulty might be resulting because top management does not like the subject. They are not interesting in legacy system or legacy modernization conversation. *P15* said, "*Data transformation is not a sexy subject*".

However, it is difficult to convince people on board because lack of trustworthy data. As *P9* said "*I don't have a real trustworthy data for them...OK it is going to crash on 12 of November 2013 and after that date your business is gone. So and that's not a case*". Same with what *P24* said "*There's another priority for tomorrow. And you can repeat that same question for years and years and nothing will happen and that still won't break.so ...you've been saying that it will break within the next 4 months from the last 10 years*".

Therefore, to get in priority, a good business case is needed. As *P11* said "*Are you capable to really describe, ok if we are going to start this transformation within 3 years, it can be create or deliver functionality A,B and C, and it costs 1-2- and 3. You must create one kind of relationship from and more financial perspective. So, creating the business case, that is the second challenge*". *P19* also agree that "*You have to somehow come up with the business case that says what my cost today, what migration cost, what the new total cost of ownership, and that you have to build the Return of Investment*".

Difficult to effectively prioritize the functionality for modernization

Thirteen participants expressed themselves that to govern the process of modernization is difficult. *P13* said, "*Yeah...so you always have to make sure from ok...if I do a migration I always have to do lot of check and balances. I have to do yea…Ii have to do comply check...I have to do assessment, I have to do...I have to make sure that everything works, including the people*".

The difficulties are including changing the requirement during the process as *P17* said, "*Also the business strategy itself change several time during the migration. So we also made some changes during the migration*". Same with *P7* and *P8*, "*Changing requirement during the process...but that has always is the combination with the time you need for the modernization*".

Therefore, *P13* suggested "*Basically, what you need to have is you need to have strong governance model. You need to have one clear set of rules, who make a decisions and what the impact and magnitude the decision can be*". However, the project should be little flexible with new requirement as *P12* said "*Because if it takes that long the world is changes during the project. So you have to adjust to that. If you don't do you deliver, you always deliver late. Because you deliver on specification which 1,5 year old*".

Time constraints to finish legacy modernization

Time is also perceived one of the factors in determining success or failed the process of legacy modernization. Fourteen participants share the same opinion. *P1* explained "*One that I mentioned, they run out of budget… they run slightly out the time... that's mainly to do with scarcity of people on the legacy system*". *P17* also mentioned "*We were in the time squeeze. We got only a few months to do the conversion*". *P7* and *P8* shared their experience "*It last too long...We plan modernization for modernization for 3 years, and after 5 years we stop the whole modernization, and start it over*".

Time often becomes a problem because the availability of the resources. Lack of resources makes time to finish certain steps in legacy modernization are delayed. Furthermore, money can also be a caused of the delayed of the project. *P13* said, "*Your biggest problem is an availability of resources and availability of money. And extend availability of time*".

To mitigate this challenge, *P20* and *P21* suggested their opinion. They argued that with having a better insight about IT trends, organization can do legacy modernization effectively. "*If you are not fluent with this technology, you cannot assess whether the new technology is suitable for you or not. You have to be sure that the path you take is the right path*".

5. Result from the Interview with experts

5.1. Legacy systems in industry

In order to answer our first research question, we start with find out how professional in industry define the legacy systems.

Definition of legacy systems from industrial perspective is difficult to formulate. After interview part, it seems that each participant has their own definition. It is understandable since legacy systems are perceived as a problem by organizations and each organization has its own problem. Therefore, it is difficult to come up with a precise definition of legacy system. All characteristics mentioned during the interviews session are always has pros and cons from participants

However, all participants agree that the main consideration for determining a legacy system is whether or not the system is in line with the organization's business strategy. They start to define the legacy systems if their business cannot run properly anymore because the systems are unable to deliver their business objectives. It is obvious since the role of IT is to support business process. Therefore, as long as the system can fulfill the business requirements, then it will not be perceived as a legacy system by the organizations. On the other words, the definition of legacy systems mostly comes from the difficulty caused by legacy systems and professionals in industry only define legacy system if the problems emerge.

Legacy system is not emerging at one point like big bang, but it emerges as the time passes by. Therefore, it is hard to define when exactly the system becomes legacy. Organizations can only give their estimation about the time when legacy system came at the first place. The variation opinions appear when participants give the estimation years of their legacy system.

Most of the systems which are considered legacy, were built long time ago when the technology was not developed as good as today's technology. So, age might also be a consideration to determine the systems being legacy. The reason why organization considers age as a factor to determine systems being legacy is mainly because the old systems are usually occupied with the old stuff. The fact that old systems (old software and old hardware) usually bring a lot of problems, makes the systems have a big risk and can disturb the continuity of the systems and eventually fall into category of legacy system.

It is interesting to observe that some participants define that all system are legacy system. Such definition is relying on the perception that technology today is the legacy for tomorrow. The systems they are using now (despite the systems are still flexible, still be supported, and still have a lot of knowledge available) are legacy systems. That is why age itself is not a strictly rule because some organizations do not consider age as a factor to determine system being legacy. The fact is that IT always moves forward and produces more advance technology than the previous technology. It creates the perception that legacy systems have nothing to do with age. Thus, as soon as the new technology exists which provide better alternative and could be substituting the existing systems, the old one could be considered as a legacy system. Although, some participants even argue that legacy system can be from derived from new technology, but dominantly, participant agree that legacy systems were built from 20 or 30 year ago.

The type of the systems also play a role in determine system being legacy. In many cases, legacy systems are typically the core system of the organization. They were built long time ago and accompany organization during many years. However, there are some cases that legacy system is not a core system. The reason is because the systems are not so vital in business process anymore. Such way of thinking makes the opposite perception of the legacy systems. The maintenance is not difficult and not expensive, no attention goes in the system anymore, and if the system collapsed, it will not hurt organization at all.

5.2. Problems of legacy systems

5.2.1. Knowledge becomes scarce

The common characteristic of legacy system is about scarcity of documentation. The documentation of the system is lacking, not updated or even does not exist at all. At the moment when the original developers built those systems, it might be well-documented as often been seen in normal software development. However, after the software is in the production phase, documentation is slowly not being updated. It is becoming obsolete. Consequently, when the changes happened, ad-hoc way of maintaining the system or implementation a new functionality are applied.

Such a maintenance without proper documentation has been applied for decades and resulting some damages in structure and data of the legacy systems. For instance, when new functionality needs to be added into legacy systems, often quick and ad hoc solution was implemented. It works or at least fulfills the requirement, but not in a good way. As a result, internal technical structure of the application is not optimal anymore. Not only that, by inappropriate way of changes, it will make the IT landscape of the systems is becoming complicated. It happens if new requirement need to introduce new connectivity.

There is a speculation of why the process of documentation did not work well. It might have been due to lack of supporting tools in old days. As the trend of technology in past days was not as good as the technology's trend today, people were often limited by that to document something. The existence of personal computer was not common in every organization, back on 20 years ago. Thus, the process of documentation was not easy in the past.

Another possible reason is that documentation was not perceived as something essential. People often did not take it seriously or simply forgot documenting their work. Most of the developer /programmer do not prefer documenting their work. They might have known that it is good thing to do, but under pressure, delivering the product has to go first. They only have time to comment in the body of the program itself. Even writing a comment is extra work for programmer, so do documentation is even more extra work for them. Such habit has been running since the system was first introduced.

Sometimes, the documentation (if any) is attached in the body of the codes of the application itself in form of comments. It might create difficultly for people to understand the system, since it is not properly presented or abstracted. It is also not properly structured, no picture and graph, making people easily forget about it. However, changes in the codes during many years of operation and maintenance often make such documentation misplaced from its original place. At the end, it leaves organizations without any documentation at all. It makes the knowledge about the systems is trapped in the head of the developers and most of the time, that is the only resource about legacy systems available for organization.

With this kind of situation, some organizations have the systems in which they say as mysterious systems. They do have a system which does their works properly, but they do not know exactly how the system is working. They cannot understand what the coding of the system supposed to do, because they do not have the documentation of the application. Only original developers know how it works. However, they have often retired or moved to the other organization. Furthermore, the rules of some functionality are often forgotten. Legacy system has been sitting in the organization for the long time and people simply do not take a look of that functionality. It works all the time, so why they should care anymore about it. However, it will eventually create unawareness of the system and bring difficulty when something needs to be changed in the system. There must be some hidden feature or functionality that the organization itself does not aware of.

As a result of incomplete documentation, organization has a problem of depending on certain people. Those people have been working with the system for the long time and they know every aspect of the system. They are needed by organization in order to keep the systems alive. Those people, who understand the logic inside the legacy systems, are becoming scarce for many reasons. The common reasons are those people are old people and nearly their retirement day. Those people have often started their career more than 20 years ago together when legacy systems were still new in the organization. The problem will arise if the original developers have left to another organization and leave the legacy systems with their knowledge. So, there are only limited numbers of people available to maintain the systems.

Even if there are experts available in market to help organizations with their legacy system, not all of them understand specific thing. It is even more difficult with old programming language, such as C, COBOL, PASCAL, etc.

Such situation is becoming worse since students in college or university are not interested in learning the legacy systems. It is reasonable, since not many courses about legacy system or legacy modernization are taught in college or university. As a result, this topic is less attractive for students. So, the knowledge is going away from the system and lesser knowledge about it is available in the market now. At the end, nobody really knows about legacy system and does not know what are the hidden features or business rules behind it. It is a challenge too in legacy modernization, because lack of documentation and lack of expert will lead to difficulty to extract knowledge from legacy system. These characteristics are depicted in the cause-effect relationship in Figure 10.

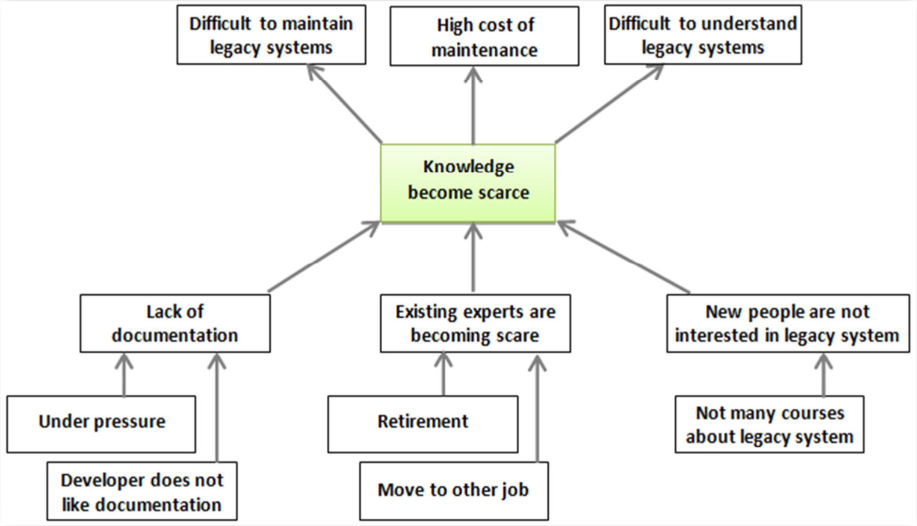

Difficult to maintain legacy systems
High cost of maintenance
Difficult to understand legacy systems
Knowledge become scarce
Lack of documentation
Existing experts are becoming scare
New people are not interested in legacy system
Under pressure
Developer does not like documentation
Retirement
Move to other job
Not many courses about legacy system

Figure 10. Cause-effect relationship of “Knowledge become scarce”.

5.2.2. Poor IT architecture and landscape

In old days, many systems were built in monolith architecture with hardcoded, including the legacy systems. The application is called hardcoded if the variable, value or data are written directly into a program. It makes an application is difficult modify because it is not parameterized or not configurable. It also requires sufficient understanding of the implementation to be sure that the change will not introduce inconsistency and cause the program to fail.

Application in which was built in monolith architecture tend to have user interface and data access are coupled with the logic of the source code. Such type of architecture has limitations which are not modularized and componentized application. Consequently, it is difficult to use only small part of it because the system works as a single piece. Figure 11 shows how monolith application is different with other type of architecture.

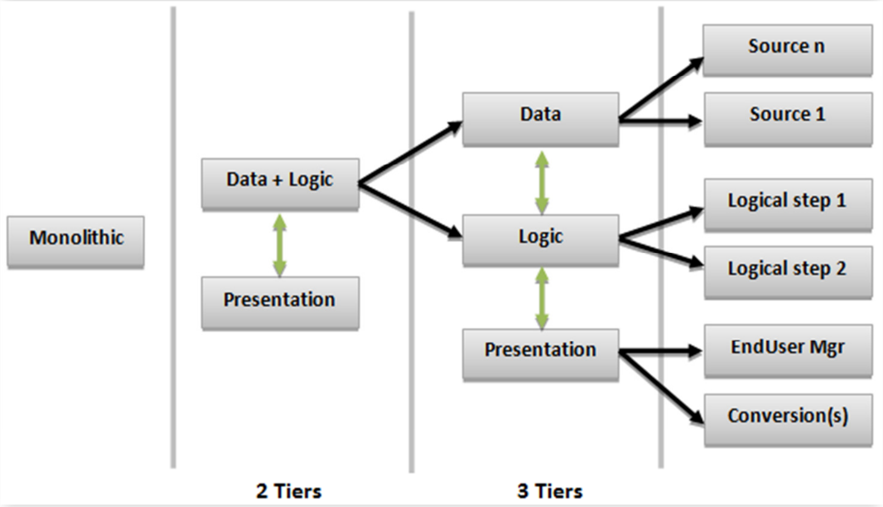


Figure 11. Monolith architect compare with other architects.

As depict in Figure 11, monolith architecture makes data, logic and presentation in one piece. Therefore, change in only one of them will require maintainers has to change the whole application. It is not componentized and is rigid to adapt to the new change. Another difficulty with monolith application is that it is hard or almost impossible to change the control flow of the application. Some architecture were designed and believed to be better that monolithic architecture.

Organizations that have been asked about support of the hardware tend to pin point on mainframe as an example of the legacy system. Clearly, because old mainframe systems are typically designed as monolithic architecture and can only allow specific programming language run on it. Moreover, not many experts are available to work with such are programming languages anymore.

Such way of programming leads to more problems such as:

Limited functionality. The systems which are built long time ago, only have simple operation which make the functionality in legacy system is limited for today's business requirement. Old technology with old type of connector has restriction to interconnect with the other systems. For instance, in insurance industry, customers' information are first recorded on a policy application that can be re-use later on in completing a claim submission and later in a renewal form. However, to integrate the other systems to re-use the data from legacy system is difficult.

Not only for software, for hardware as well. Some modern application cannot always run in old system supported by old hardware (e.g. 16-bit to 32-bit or mainframe to PCs). A limitation like this was not a problem in the old days, but during the development in IT world, it becomes a problem because it is limited the movement (agility) of business nowadays. For commercial organization, if this limitation still continues, they have a risk of go out of the market.

Poor Architecture design. When application is made with monolithic architecture design, the connectivity among the systems, more or less, is point-to-point. If *n* applications are connecting each other, it will produce *n*(*n-1*) number of connection or interface. Consequently, if system "A" need to be connected, it means the organization needs to generate, document, test and maintain *2n* new connection. For instance, if organization has 5 applications with 20 interfaces, adding 6th application will require them add extra 10 new interfaces. It will increase the complexity as more applications are integrating each other. Moreover, an organization needs to modify the code in each of the application and resulting huge cost in maintain the systems.

It is also not easily configurable, when the new requirements are coming and require the developer to modify the code in the system. The developer has to find the line where the code needs to be modified. If developers or programmers are new and not familiar with the code, then it will take long time to change small part of the system. It makes even more difficult since each developer has his style of writing the code which can make other developer has to spend more time to understand the code.

All of those lead to poor architecture of the systems. The complexity of connectivity among systems is typical characteristic of the legacy systems. Sometimes, since legacy system is a core system, the other subsystems need to connect to it. Multiple systems had been added when the new requirements came. Function for printing, reporting and external connections are integrated with the legacy system. Even though it could be linked with other systems, the connection would be nasty and not in appropriate way (again because lack of documentation and expertise).

There is also the case that sometimes organization failed to remove the old system after new system was introduced. So, the new business processes are running in the new system and old business processes are running still in the old system. It creates redundant systems in the IT ecosystems of the organization. The maintenance will be difficult and expensive. However, for some organizations, they sometimes preserve the old system in purpose for compatibility reason in their ecosystem (e.g. services message provider organization). They do that to accommodate their clients who still use the old format of message. Although the cost for maintenance the old engine is perceived high, but as long as their clients are able to afford it, it works fine with them. Consequently, system is becoming big and large and contains complicated business rules as the time past over.

Such architecture often does not fit with the way people want to work and how the infrastructure work today. If the systems are not designed properly, it will run into trouble faster. The problem will be started when changes are made (inappropriate maintenance) and resulting the damage of the structure of the supplication. Figure 12 depicts the cause-effect relationship of factors that causes poor IT architecture and the consequences of having poor IT architecture.

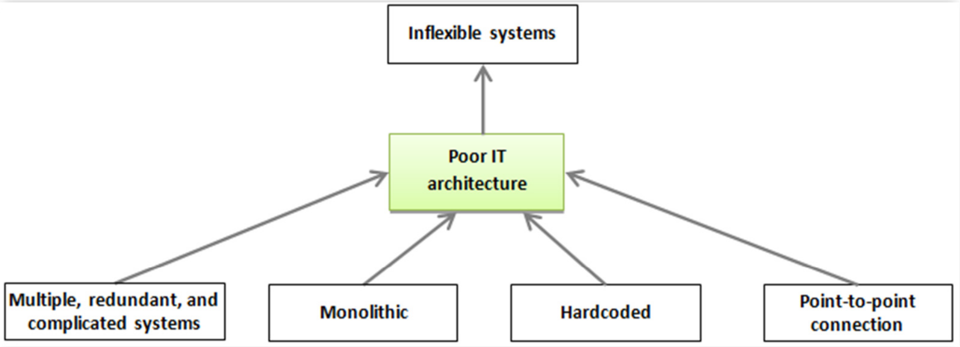

Inflexible systems
Poor IT architecture
Multiple, redundant, and complicated systems
Monolithic
Hardcoded
Point-to-point connection

Figure 12. Cause-effect relationship of "Poor IT architecture".

5.2.3. Unable adequately supported, maintained, or enhanced in-house

In legacy systems, maintenance and adjustment of legacy system is getting more difficult because several reasons being, lack of knowledge around system, poor architecture of the system and no support from provider anymore.

Lack of knowledge is mainly because the old system such as legacy system tends to be built without good documentation. If there is documentation available, it was often attached in the source code of the application itself and no graphical feature. It makes the documentation is difficult to understand and make the documentation of legacy systems only usable for people who are already work with that systems for a long time. However, in many cases, the documentations are often missing or were destroyed accidentally due to reorganization or natural disaster. As a result, no one knows exactly how the system does its job.

Lack of documentation makes organizations have to depend on certain people who still understand technology about legacy system they have. However, the availability of experts are limited makes the adjustment of legacy system becomes harder and harder. Not only that, it makes the change will take so much time before it finish. The cost for maintenance and operation would be high as well, because they know they are rare and those people often have many years' experience with the legacy systems. In addition, when the new developers are coming, they will basically rely on the old developers and their skill on read the source code. It will take some times before new people are able to understand completely the system and to maintain it. Such problem becomes worse since the number of experts is decreasing to assist the new developers makes the new comers need to learn by themselves

The architecture of legacy system in itself also makes the changes within the system difficult. When the system is built in old days, they are monolithic and hardcoded. The coding like that leads to the difficulty of maintenance because change in small part will require developer to open the entire codes and start to read hundred even thousand lines of code. So, maintenance cannot be done effectively and efficiently because simple change can take so much time and resources to be finished.

Moreover, applications which come from the old days tend to have limited support from their supplier now (no updates or security patches, limited in hardware support, etc.). Legacy systems now consist of many components which are no longer available. For instances, when original supplier stop produce the hardware of the legacy system, user will not get any driver for the chipset anymore. So organizations need to figure out how they can install the platform and the application in the platform. Not only for hardware, but also support for the software as well. The maintenance becomes harder since there is no updated software anymore.

The difficulty of maintain the system will get worse if the legacy system is a core system of the company itself. In some cases, this core system (mostly backbone) has many links to the other sub-systems. Changing the core system could impact the exiting functionality and can lead to the failure of the entire systems. Lack of knowledge about system and supportability from supplier lead organization to just let the systems like what it is now. They are afraid if they touch their legacy system, it will go down and they cannot make it up again. In this case, if there is possibility to just modify the mid-office or front-office, then the company will go for it. An often observed phenomenon is “Don't fix it, until it is broken”.

The combination of all problems such as, monolith architecture, hardcoded, lack of documentation and scarcity of knowledge bring difficulty to maintain legacy system and difficulty to extract the knowledge from legacy system. With all these problems, better maintainability (quickly and easily to maintain) / become maintainable is the driver to modernization the legacy systems. All situations above define legacy system as a system in which resist in maintenance. Figure 13 shows the cause-effect relationship of “Difficult to maintain”.

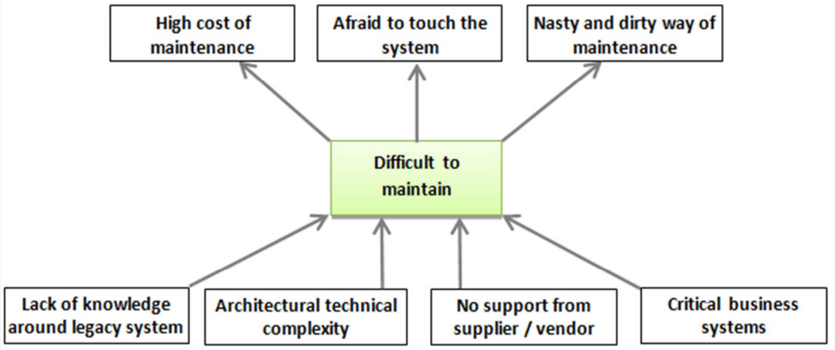

High cost of maintenance
Afraid to touch the system
Nasty and dirty way of maintenance
Difficult to maintain
Lack of knowledge around legacy system
Architectural technical complexity
No support from supplier / vendor
Critical business systems

Figure 13. Cause-effect relationship of "Difficult to maintain".

5.2.4. Incompatibility with current and/or future intended environment

Organizations often see legacy systems in broader context relate it with the ability to interconnect the legacy system to other systems. It is more and less like the chain of information system which connect one application with other applications. The connection is not only connecting the software, but also the hardware.

As we know that environment of legacy system also evolves during the time thus, aligning old legacy system with newer IT environment is difficult. The common example is the development of the Internet. Legacy system was build when the era of Internet was not as advance as today. So they lack or even do not have capability to communicate with outside world by using internet connection. They are also not equipped with modern API (*Application Programming Interface*) like most of the today's systems. It makes a legacy system does not usually communicates and talks with other systems. In addition, because of today's environment more or less requires internet connection, this obstacle could be a driver for legacy modernization.

Therefore, when it comes to interconnection with the other systems, legacy system has a big problem. The interoperability of legacy system with others system is bad and difficult. Normally, legacy systems were created without having information exchange system. With this characteristic, it makes the systems hard to connect and interlink with other systems. Such condition results in difficulty as well if new requirements are coming and need to be implemented in the system. The system becomes inflexible when a new requirement requires connectivity between legacy system and other systems. Change or adaptation of the systems becomes much more difficult. The problem becomes harder since there is limited knowledge on how to integrate legacy system available. By looking on this fact, problem in connectivity and incompatibility are one of the top priorities in many organizations.

Another reason is that the margin between technology in used by legacy systems and the new technology in other system is large. The margin / discrepancy between legacy system and its environment are huge enough that can make legacy system difficult to adapt with other systems in the ecosystem. It is not compatible with organization's IT environment and getting more difficult to keep the legacy system updated or to move it forward to newer version. For instance organization has to renew their old windows XP applications because they cannot run their XP's applications in windows 7 or windows 8 platforms. For the information, in 2014, windows XP will not be supported by Microsoft anymore, so this situation forces organization to immediately modernize their applications.

During decades many connections were built in ad hoc manner and this makes the harmony of the systems are bad. The system becomes much more complex and complicated. Therefore, for some companies who want to implement SOA principle in their IT landscape, they find it difficult to connect their legacy system with their service bus. It is a lot of work to make the new interface so their system can be SOA complies. No wonder organization is looking for the standardize system as a reason to do modernization. Figure 14 shows the cause-effect relationship of "Poor compatibility".

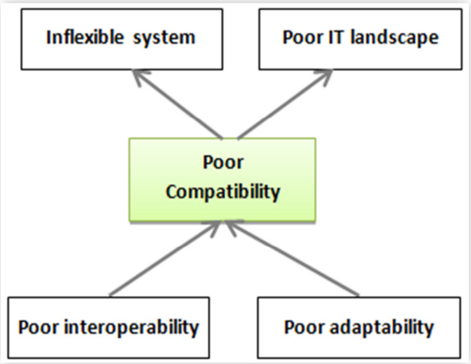


Figure 14. Cause-effect relationship of "Poor Compatibility".

5.2.5. Limited suppliers / vendors

Talking of legacy systems, they usually lack of support from their supplier either for the hardware part or software part according to 14 participants in their interview session. It had been confirmed by six participants from the organization which provide products or services for the others organizations. Suppliers or vendors of legacy systems are lesser and lesser supports their legacy systems because it takes a lot of effort to work around them. For instance, software provider will find it difficult to build an application which requires specific hardware to make it run. It is not-feasible anymore now, since the suppliers of that particular hardware now stop producing their product.

The other reason is because suppliers are more focusing their market to the new technology. Market demand for new technology is perceive higher than the demand for legacy system. Typical characteristic from supplier is they are IT trend follower. They always compare between old system and the new system and see which one will benefit them the most. In many cases, new technology often comes with more advance technology that benefits the supplier in technology and financial side. Thus, the investment to support legacy system is slowly reduced and goes in to the development of the new system. However, for them as an IT follower, it is not easy to predict the IT trends in the upcoming years. It is the typical challenge for them in their organization.

Usually, the support for the hardware can be a big problem if the suppliers decide to stop supporting the hardware they produced. Organizations really depend on the producer of the hardware to run their business processes. They depend on the supplier because it is not common to produce and develop their own hardware and internally, organization does not have enough technical experts to maintain their old hardware. When legacy system is constrained by hardware and organization could not get support from their supplier, then E-bay is sometimes the solution for their problem. It is not ideal to maintain the systems.

The problem is not only for the hardware part, but also for the software part. When supplier stops supporting their product, it means that no patch and no security update are available anymore. It happens when suppliers do not see the benefit of preserve the legacy systems. Therefore, they tend to discontinue support the systems. As the product is approaching the end of its lifecycle, the organizations often ask the supplier to extend the lifecycle of the product. It could be done by internal collaboration between supplier and consumer. The provider will develop part of the software or hardware so does the system has longer lifecycle. With this, it can mitigate the risk of failure of the system. However, special request like this usually cost the organization a lot of money.

It could be also the case that less people now are working around specific programming language. It could be driven by lack of investment on certain programming language. New modern programming languages have a lot of advantage compare with the old ones. No wonder, the applications which are built nowadays have rich features (better UI, easy interaction, and easy user interface). It attracts many developers to build their applications with new programming languages and left the old ones behind. Consequently, the old programming languages are not popular anymore and make investor not to invest in developing the languages. Moreover, new graduate students are less interested in learning old languages make the languages have no future. At the end, software application which was built by using those old languages lacks popularity and experts. Figure 15 shows the cause-effect relationship of “Limited supplier / vendor”.

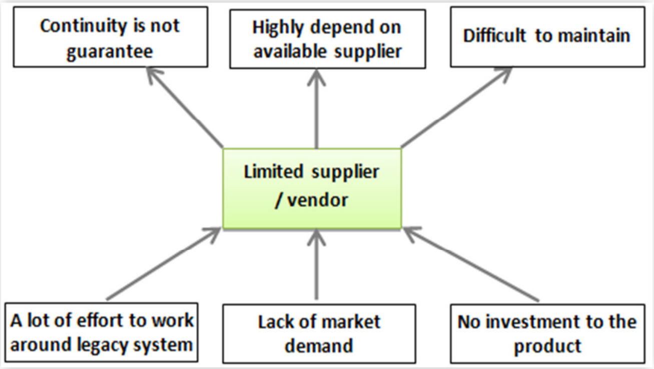

Continuity is not guarantee
Highly depend on available supplier
Difficult to maintain
Limited supplier / vendor
A lot of effort to work around legacy system
Lack of market demand
No investment to the product

Figure 15. Cause-effect relationship of "Limited supplier/vendor".

5.2.6. Too rigid to comply with new business requirements

From flexibility point of view, organization will consider their system as a legacy system if the system is less flexible. One way to determine the flexibility of the system is by looking on the way the system was designed. Legacy systems are mostly rigid because they came from the old days. In the old days most systems was designed in monolithic architecture and hardcoded. Monolith applications are difficult to reuse some functions of the system without using the entire application.

On the other hand, hardcoded means that the software is written with embedded input or configuration. With these types of design, it is hampering the development of system itself especially if the system needs to adapt with new business requirements. It hampers the organizations by the way the system was built. If a new requirement with complicated business logic is to be implemented, it is really difficult to make legacy system comply with a new requirement. It creates inflexibility in legacy system. For example, the organizations which are closely related to the government (tax, insurance, etc.), they have obligation to follow the regulations from government and the regulations are often rapidly changed. At the end, this problem can make the system less flexible and so called legacy system.

Inflexibility in legacy system creates a lot of hassle for organization to cope with the new requirements. Business requirement cannot be implemented quickly enough, so time-to-market of the product increases. Or in worst case legacy system completely cannot support new requirement. The impact of inflexibility would be loss of profit, loss of clients or in worst the organization will out of the market. Figure 16 shows the cause-effect relationship of "Inflexible systems".

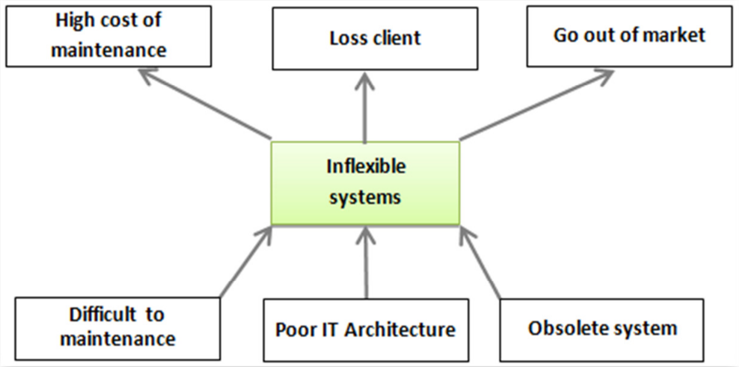


Figure 16. Cause-effect relationship of "Inflexible system".

5.2.7. Poor user interface

The usability is a problem in legacy system. The user interface is not modern in legacy system and can give a bad interpretation in presenting the information. End users often not satisfy with interface of the legacy system. When new employee join it is difficult for him/her to understand how the system works. It needs more time because new employee, especially young people, to become familiar with old screen and old interface that legacy system has. Such a problem often appears in governmental institution. As an end user of the system, their employees only operate the system without knowing the technicalities behind it.

Furthermore, unlike any IT organizations, governmental institution is more or less not an IT trends follower. They are not immediately implementing new technology in their system if ones appear. They tend to stick with their old system as long as their primary processes can be fulfilled. No wonder, old systems are dominating in most of governmental institutions.

On contrary, old users are familiar with the system. They are mostly the end users who work with the legacy system since long time ago. They know every single aspect of legacy system. In fact, if the legacy system is replaced by new system with new and modern interface, they might not satisfy with the system and their productivity will not be as optimal as by using legacy systems.

Another driver would be customer’s satisfaction. Sometimes the demand from customer also is a driver to do modernization, for example when customers do not like the old GUI. It does not look fancy and not rich. For some customers it makes their work not as effective as with new modern GUI.

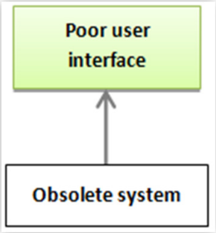


Figure 17. Cause-effect relationship of "Obsolete system".

5.2.8. High risk of failure (High risk of continuity of the system)

Legacy systems as mentioned above bring some problems that can be painful for organization. All the problems are accumulated and become bigger that can threaten the continuity of the system or the organization. When the problems are big enough, the risk of failure of the system will be increasing as well. The problem regarding the risks is more crucial if the systems are handling the core business of the organization. Such system contains valuable business value in it for instance core transaction. If this system fails, it will bring severe to the organization.

The participants who are the architects agree that this is the crucial problem. They know that from IT point of view, the IT landscape of the legacy systems is not suitable with the technology nowadays. If this situation still continues, the risk of failure will be bigger and bigger. The systems are also complicated that can make the recovery time become longer. If the system have problem, the time to make it work again would be longer. It is because inappropriate documentation and complication of the. Besides that, for the back-office systems, they do not have many changes. They are doing their job perfectly, so people tend to just let it like it is. However, if it continuously happens, it can build the feel of unawareness of the system. And at the time when the system is crashed and organization needs to make a change in their back-office, they cannot understand about the rules in the back office anymore.

The risks become increases if supplier does not support legacy system anymore. Old technology tends to have limited amount of suppliers. If they do not support the system anymore, organization will only rely on their internal people and cannot maintain their system properly. It is, for sure, increasing the chance for the system being failed. If there is no support anymore from supplier, it will be a real problem for legacy system to operate again if it failed.

However, change the core system is also has high risk of failure. Core systems usually have many dependencies with the other systems and the connectivity is messy and complicated in legacy system environment. If inappropriate changes happen in the core system and makes core system stops operate, then all the systems connected to the core system are also be affected. It is in line with what *P1* said that legacy modernization is like open heart surgery, thus prudent in approach is strongly recommended.

As a consequence of all problems accumulated, the systems often down or stop operating. Such problem is haunting organization because once the system cannot operate anymore; their business would be stopped as well. Therefore, as long as organizations still have time to migrate the system, it is strongly suggested to migrate the system as soon as possible. Figure 18 shows the cause-effect relationship of “High risk of failure”.

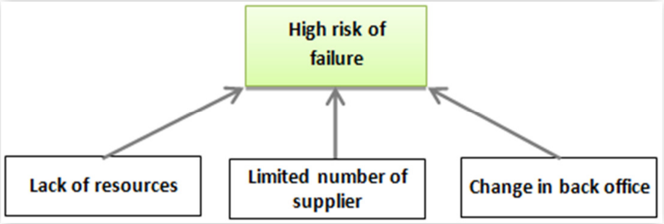


Figure 18. Cause-effect relationship of "High risk of failure".

5.2.9. Too costly to maintain

As discussed earlier, problems of the legacy system become increasingly as the time passes by. The problems accumulate (e.g. difficult to interconnect, scarcity of experts, etc.) and results in higher cost of maintenance. In total many organizations spend 85% of their money in a year for just maintenance legacy system, and only 15% goes to innovation. The problem of higher cost of maintenance is often the first driver for legacy modernization.

Out of 26 participants, 20 participants agree that the cost to maintain legacy system is expensive. The first reason is because the knowledge around the system becomes scare. Less and less people want to learn about old technology with the combination of graying existing expert contributes to the high maintenance cost. For instance, the maintenance of mainframe is expensive because not many people now know how to operate in mainframe now. Only small number of people knows it compared to the new technology. T

In addition, not many students were educated with old technology make them less interested in the area of legacy system. They are often taught about new environment with the future technology and not learn something from the old days. For the organization in which their legacy system holds vital business rules, this is a real problem because they see it more difficult now to find new people (scarcity of knowledge) who can maintain the system. If there is any expert in legacy technology, then the company is really depending on those specific people / supplier to support their legacy system. Those people are limited in capacity and quantity which leads to the higher cost of maintenance. There is no option except hire an expensive and dedicated consultant to maintain their legacy system. Such factor is mainly drives organization to think about how they will replace their legacy systems.

The old architecture of legacy system with old framework also contributes to the high cost of maintenance. It is because the system is not a standard system Furthermore, since legacy systems are not monolithic and hardcoded, it is difficult to connect the system with the mainstream systems (standard system) now. Organization needs to build extra layer or intermediate layer to accommodate the interconnectivity. It is extra work that can increase the cost of maintenance.

License of legacy system and total cost of ownership also contribute to the high cost of maintenance too, particularly for mainframe applications. Software licenses and cost of hardware are too high. Legacy systems are often be seen a cash cow for many suppliers. The customers have new demands which are difficult to build in a legacy environment. Customers are leaving suppliers for that reason.

The other reason of high cost of maintenance is because the maintenance of legacy system is difficult. Old system doesn't support much in today's technology. The framework that was built many years ago was obsolete, so if organization wants to implement new feature in the old framework, it might not applicable. Therefore it might cost the organization extra fee to make it happen. The difficulty of maintenance also comes because the system is hardcoded and lack of documentation. Without proper documentation and difficult modify hardcoded application makes extra work for developer as well. Human resource is expensive and costs money as well.

Moreover, the difficulty can come from the complexity of the IT landscape with a lot of inappropriate connectivity between systems. It is more or less messy a systems infrastructure of legacy systems. The cumulative effect of decades of growth and evolution has produced the complexity the organizations are now dealing with. Many non-standard systems are connecting each other resulting complicated conversions among them. With this complicated architecture, the cost for maintain it become high.

Last but not least, the application portfolios are also grown from time to time. That could be resulted from mergers and acquisitions between two or more companies. Mergers and acquisition are most of the time change the landscape of IT in the organization. As a result, organization may be dealing with redundant applications or applications with their function that cannot easily be reused. It might be a case to that each business unit acts on its own, so many identical functions are implemented in different business unit. Further consequently of this problem would be increasing on both cost and time to market to deploy new functionality, because changes have to be made in each application. Figure 19 shows the cause-effect relationship of "High cost of maintenance".

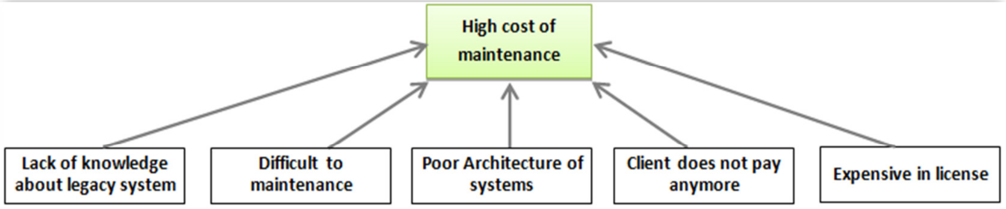

High cost of maintenance
Lack of knowledge about legacy system
Difficult to maintenance
Poor Architecture of systems
Client does not pay anymore
Expensive in license

Figure 19. Cause-effect relationship of "High cost of maintenance".

5.3. Positive characteristics of legacy systems

5.3.1. Reliable (e.g. stable and robust)

Most professional in industry agree that there are two factors that make the legacy system has a good reliability. Those are robustness and availability of the system. Robustness is an ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc. Legacy system has been running for decades and during that period the system has been tuned. So many bugs are fixed and crashes are not happened often. Error handling of legacy system is perceived good and the processes are still executed even though the data are inputted in a wrong way. People know how to use it for long time so that they know how to avoid the error that could happen.

Moreover, legacy systems have been stabilized from time to time resulting in higher availability. As a core system, the systems need to be up and run in order to performing the business processes. Organizations have to really take care of their core system anyway. Looking on the age of legacy system, organization keeps it for many years because legacy system delivers their job well. It has been customized and improved in functional level in which make the system available and able to handle lot of data. In addition, since most of the old legacy systems are core business, organizations really take care of the system well. They try their best to minimize the problem in their core business that can make a system failure.

In addition, most of the time, legacy system is the core of the business in which is located in back office and handle the basic functionality. Naturally, the core of business has not changed that much (less change in core system). It makes legacy system often stay as it is and stable. As a result, no significant problem happens in the legacy system that can make the failure of the system. The system is always up and run 24/7 and perceived stable in term of reliability. It might be a bad code, but it runs without any significant error.

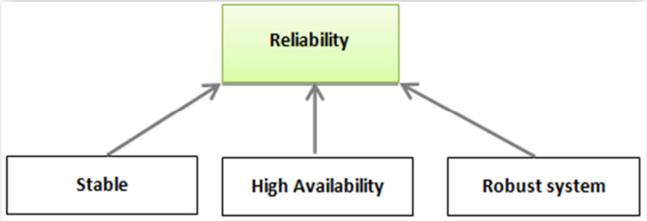


Figure 20. Cause-effect relationship of "Reliability".

5.3.2. High performance

Generally, performance can be defined as the capability of a system to process a given amount of tasks in a determined time interval. The performance could be a measure of latency or throughput of the legacy system. The performance of legacy system is perceived well by experts in industry, especially for the architect. Four architects in interview session agree the performance of legacy system is good. For most of all participants, they see legacy system as a system in which still being used today because it works properly. It is due to the fact that during many years of maintenance, bugs, errors and unexpected behavior in legacy system have been fixed. Even if the bugs are not completely removed, end–user knows the tricks so the error will not happen.

There are several reasons why organizations see the performance of legacy system is good. First, legacy systems originally do the basic functions which sufficient enough to run the organization (e.g. administrative, payroll, etc.). It handles jobs which easily formalize and automated from the first era of automation. Thus, the simplistic it has make this system only has the functionality that organization needs. What organization needs is what it was implemented in the systems. So the performance of legacy system is met the expectation of user and there is no crucial problem with it.

Secondly, because of time response of the legacy systems is perceived fast in general. However, this is the reason comes from participant who are in expert in old mainframe machine. The response time is measured satisfy because for them, legacy system does automating task without process thinking. It is driven by its design, monolith architecture is good for processing a huge number of data. It is particularly for mono-hardware such as mainframe machine, because mainframe is good for I/O intensive system. It makes the system looks stable with high availability to perform business processes. Therefore, no advance logic needed to be implemented into the system which makes the systems can complete its task or job well.

Furthermore, the users also familiar with the old legacy systems because they have been working with the systems for the long time period. So, they know exactly how the systems will behave if the actions have been executed. With this, they can avoid unexpected behavior of legacy systems. They know already from experience that if the error is produced, they simply overcome it by following certain procedure and then they will get back on track again. Or if they know that a particular action will produce error, they will just simple avoid that action so that no error is being produced. It creates awareness for user for not easily make a mistake if they operate the system. Figure 21 shows the cause-effect relationship of "High Performance / Availability".

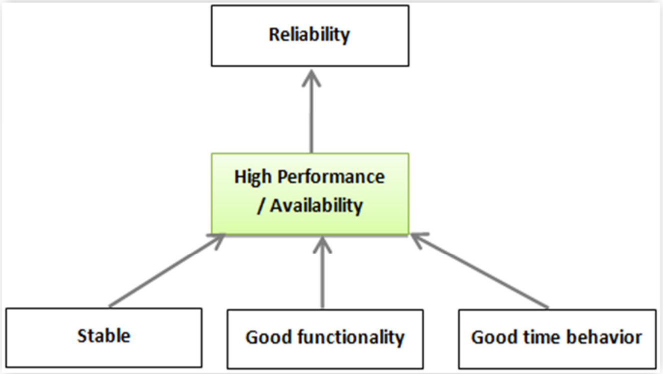


Figure 21. Cause-effect relationship of "High performance / Availability".

5.3.3. Proven technology

Old system is immediately recognized as legacy system. They are old and keep doing their functionality well. Functionality is the ability of the system to do its job for which it was intended. The legacy systems run without any significant failure. They are stable system and do their work for long time period. That is why legacy systems are always on air and do their task until today. It finishes its task well, meaning the expectation of user is in-line with the capability of the system. With this fact, availability of the system is perceived well, because they have been running for such a long time period (20 – 30 years) serve the organizations.

Stability is also come from the characteristic of legacy system in which it is located in back-office. Back office by its characteristic is more stable because not many changes happened there. Back office system often is the core process of the business, so organizations tend to not touch it to avoid fault or error. If they can change the mid-office or the front-office, they prefer to do so. Therefore, not many business requirements changes in back office which makes the system stable.

In addition, legacy system is supporting the initial requirements of the organization, such as administrative task which can be easily formalized for automating task. Although sometimes legacy system deals with a lot of data, there is no smart or intelligent process in it. Therefore it can accomplish its task good for long period of time. Such situation and condition has been running for many years and support organization business since the organization was born. It works fine for many years.

The fact that organization still exist until now means that organization get benefit out of their legacy system. It is in-line with the meaning of proven technology saying that the systems that have been demonstrated or verified without doubt to comply business requirements for long period of time. Therefore, legacy system can be seen as system that is proven technology that generates revenue for the organization for many years. Figure 22 shows the cause-effect relationship of “Proven Technology”.

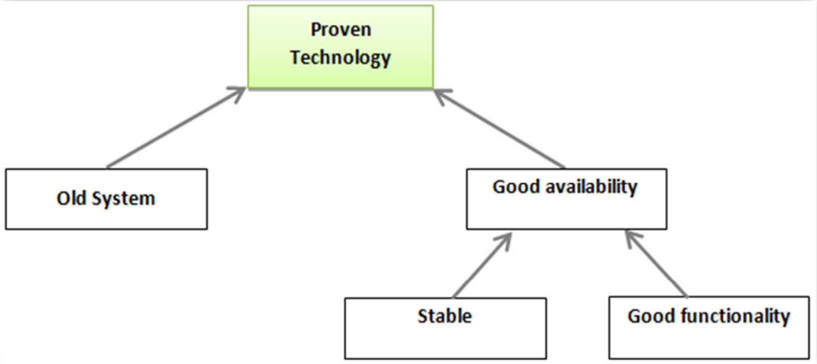


Figure 22. Cause-effect relationship of "Proven technology".

5.3.4. Business critical

Legacy systems were often found as a core system of the organizations. It has also been confirmed by consultants who participate in the research. These systems were built to support the first business processes of the company since many years ago. In some cases, legacy system even was born at the same time as the organization was born. Therefore, it is invaluable system and foundation system for the survival of the company. Such system is usually still has business impact and generate profit for the organization. Inside it, there are a lot of business rules that have been evolved and adjusted to fit into the business strategy of the organization as it develops as well during the time. So for some organizations, their core system contains major algorithm and have complicated business logic inside it, creating large and complicated system. However, those business rules now are not so much different from the original business rules, because they are the foundation rules of the organization.

In addition, the systems have been operating for a long time and bring benefit for the organization. Our participants, who are in managerial position (CIO or IT manager), tend to say that legacy system is a profitable system, especially in multinational organization. The fact that without legacy system, the organization cannot growing as what they are now, make them put legacy system as profitable system as well as business critical system. Organization might not gain profit or even not exist now if there is no legacy system. That is why this system is the key for the organization. It is mainly the reason for them to not change the systems, because by change the system, there is a risk of losing profit even if the technology is a current technology. Such condition makes companies afraid to do changes in their system due to lack of resources, supporter, and complicated system. They cannot afford if by executing legacy modernization, the system will collapse and affect the continuity of their business. For some companies this is the reason for not touching the legacy system and not doing legacy modernization.

In addition, legacy system is beneficial system for the organizations that provide ecosystem to exchange electronic mail among their clients. They see their legacy system is a profitable system because their job is to accommodate the flow of communication between various systems in their client side. The different version and system in their clients side make them has to do a lot of conversion with their engines. Legacy systems in their client side, which holds problem of interconnectivity is their market to connect their ecosystem. Figure 23 shows the cause-effect relationship of "Business Critical".

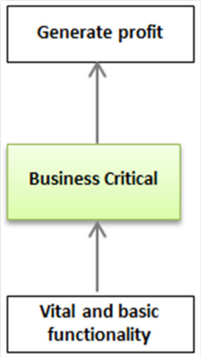


Figure 23. Cause- effect relationship of "Business critical".

5.4. Drivers for legacy modernization project

As explained before, there are some problems caused by legacy systems and mostly, and organizations will start to think about legacy modernization if those problems emerge. Modernization driver from technological view is often about problems such as maintainability, supportability, compatibility, and quickly changes of regulation. Those are the common reasons to modernize legacy systems from technology perspective. For business purpose, the reasons for legacy modernization are so organizations can expand its business, reduce high cost of maintenance, and open a new business opportunity by modernizing their legacy systems. Quicker time to market is one of the business opportunities that the organizations will achieve if they modernize their legacy systems. It is also the driver to modernize legacy system, although not as strong as the other drivers.

The changes in technology are pushing the evolution of systems in several ways. Mobility, apps, social media, and those kinds of thing will be the new generation of application development. All of which work against the legacy systems; therefore, almost all the problems from technology side can actually be mitigated by moving from the old legacy system toward current technology. Current technology such as big data, cloud system, mobile apps and social media do not easily fit with legacy system which make legacy system less flexible. For instance, nowadays a lot of organizations jump into mobile platform. With the old legacy and monolith architecture, it is hard to realize mobility market. So doing technology refreshment by moving their legacy system into new environment is most of the time the solution for mobility strategy.

With current technology the maintainability can be done easier because many resource available and make the cost lower. The system will be more flexible that can make the system easier to be configured if new requirements are coming. And because the technology is closely related to mainstream technology now, it will be equipped too with modern interface that can make interconnectivity between systems a lot easier.

One should keep in mind that modernization is not merely technology problem, but it is also a business problem. It is because legacy systems cannot meet the pace for making the new development. Almost all IT managers who participate in this research say the same. Even if the legacy modernization is driven by technology, it is, at the end, has an impact in reducing cost of maintenance. So, there is always business problem behind legacy modernization.

Also risk of failure and loss of business by having legacy system are triggering organization to do legacy modernization. Business problem even is more vital than technology problem. Thus, legacy modernization without business involvement can lead to failure of the legacy modernization project. No wonder CIO in the participants' list see business case is one of the most difficult challenges in the legacy modernization. Meeting business needs such as lowering cost, reducing cycle times, integrating system internally or externally (e.g.B2B or B2C) and creating adaptive and responsive business model are some of the driver of legacy modernization.

Legacy modernization in essence should be able to facilitate doing business and it is strongly suggested that legacy modernization should always be oriented on business. Therefore, organization will not go for modernization if there is no business value behind it. For instance in electronic banking, the customers will not consider whether the system is old or not. As long as the user interface looks modern and the transaction can be done perfectly, then users do not mind to use the system. It is not only about technology decision, because technology problems can easily be solved. Even though from technical perspective legacy system brings so many problems and has bad impression, legacy modernization can only be executed if there is a significant business interest in it. It is always pressure from business to IT. Thus, it depends on financial decision as well because legacy modernization should bring benefit (e.g. profit) as well.

Wiederman et al. (1997) investigated about the legacy modernization and suggested that there must be enough motivation for the organization to understand business opportunity before they decide to do legacy modernization. Generally, most of the participants said organizations are not reluctant to do modernization as long as there is business motivation behind it. Clearly, it means that there are a lot of motivations now to modernize legacy system toward new system.

5.4.1. Become flexible to support changing business requirements

Inflexibility system restricts the development of organizations because they see legacy system as a rigid system. Become flexible is the strongest driver for the participants who are the architects. They know that in infrastructure of the legacy system is complicated that bring inflexibility to cope with the new requirements, especially when the infrastructure represents monolithic and hard-coded system with point-to-point connectivity. It is also the main driver according to two project manager in legacy modernization. They argued that their organizations are being haunted by inflexibility of the system. Their legacy systems cannot follow the pace of their business activities now.

When a new requirement is coming and if the system is not flexible enough, then new functionality cannot be implemented in their system. In other words, inflexible system cannot facilitate organization doing their business or cannot support their business needs. It makes organizations lose their opportunity to gain profit or expand their business. To overcome this problem, some organizations decided to start the process of modernization in order to have flexible system that is able to make a change faster and more adaptable.

As explained that the inflexibility is often measured as how good the system is to cope with the new requirements. For organizations which provide service for their client (e.g. software house or message service provider), having flexible systems is what they want the most. Their customers always have some requirements that can change the systems and they want to changes happen quickly. So the driver for them is if the new requirements that have big impact in legacy system are coming frequently.

Organizations are also worrying whether legacy system can still continue this way in the future. They are afraid if legacy system cannot facilitate them doing their business and it might disturb their business model. It is mainly because legacy system is getting increasingly difficult to be in-line with organization’s business strategy. Top managements start to feel uncomfortable with the limitations possessed by legacy system. These limitations hamper them with their work and with the technical aspect today. It is getting increasingly far from their future roadmap or their business strategy in the future.

Inflexibility comes when the system cannot quickly implement business requirement or even cannot response to the request at all. Some examples that can lead to legacy modernization are:

1. Those are specified into specific phenomena for example Y2K problem year 2000.
2. In banking industry, Euro implementation in early 2000 and the adaption of SEPA payment standard.
3. Legacy modernization is mostly influenced by law, for instance taxes. Such a change is mainly because of the political situation. For example the rule of in what age someone needs to get retired and get their pension after that.
4. Legal and regulation that cannot be implemented in the system. For instance in insurance organization, each person has his own policy. Thus, the system also has to adapt to the changes.

Inflexibility can be also resulted from capability of the system to analyze the data. In old days, the first era of computers had various constrains such as limited memory, limited processing capacity and so on.

Looking at the fact that old architecture is not flexible enough to cope with business requirements now; organization must develop systems that incorporate heterogeneity as a fundamental part of your IT environment. The reason is so it can accommodate a variety of hardware, operating systems, middleware, languages and data stores that are developing from time to time. One alternative is to move their system into current technology, for example make their system SOA compliance. SOA is expected because it is easy and more flexible and provide them with cheap and faster way implement new requirements. With all these business challenges for IT, no wonder that application integration is one of the tops in priority list of many CIOs.

By making the systems more flexible, implementing requirements is most. Taking SEPA implementation as an example, if organization only have to implement it in one system, there will be no problem. However, do the same thing in 20 – 25 system is exhaustive and cost a lot of money. In addition, by doing legacy modernization in current technology, organization can mitigate incompatibility problem, scarcity of knowledge of legacy systems and make simple IT landscape. And by following IT trends, it can enable new way of operating from just automation to process thinking system or intelligent system.

5.4.2. Create new business opportunity via mergers and acquisitions

Naturally, organization also develops from time to time. Changes within organization and outside organization are forces the application to evolve in order to fit with organization strategy. In many cases, legacy system becomes a problem only if the changes are resulting problems (flexibility, damage in application, etc.) that cannot support today's business requirement.

Natural moment such as acquisition and merger could also be a driver to the modernization of legacy system. Growth by merger and acquisition has become standard fare in many organizations, so entire IT organizations, applications, and infrastructures must be integrated and absorbed. With all the mergers and acquisition, between two or more organizations, some participants define their legacy system as a system which is inheriting from another company. Merger and acquisition are often changing the IT landscape of the organization. Therefore, this is a good driver to do the legacy modernization. Change in the structure of the organization is often seen as the perfect moment or time to do legacy modernization. It is because an organization has to spend their money anyway to facilitate the changes. Such opportunity should be used by any IT departments to come up with the good business case for doing legacy modernization.

When technology is growing to become more advance, it often facilitates opportunity of doing business. For instance, mobile strategy and the era internet nowadays require the system to cope with that environment. Big data, semantic technology, and mobile application are now the trend and this could be an opportunity for organization to gain profit from this situation. However, legacy systems with all its limitation prevent organizations to do their business in those areas. Therefore, modernization is one of the solution for organization to keep compete with the market.

Another driver is utilizing the data in which came from customer. Customers sometimes provide their data that can be used by the organization to increase their profit or expand their market share. Besides the data from customer, organization can also use data available in Internet now via API (*Application Programming Interface*) and beneficial from it. However, legacy system by its characteristics is not smart to processing the extra data that can beneficial the organization. At the end, organizations sometimes just throw away the data they receive from the customer. By doing legacy modernization, organization can enhance the functionality of their legacy system and can process the data and produce something valuable out of it. As a result, they could actually increase the profit or at least make their customer relationship stronger.

In addition, organization that creates the ecosystem to connect all their customers together, they see legacy system as a business opportunity for them. Ecosystem is always growing. Many hardware, software, and service providers see an opportunity to provide value in this market. Poor interoperability between legacy system and other systems create business opportunities for them. In the other words, ecosystem with legacy system is their market.

5.4.3. Reduce the cost of maintenance & operations

Legacy modernization is not only IT decision, but also a business decision. Thus, doing legacy modernization must provide organization with benefit at the end. One of those benefits is money or profit. However, legacy system cannot make that happen because of high cost of maintenance and operation. One reason that creates high cost of maintenance is when the new requirement comes, legacy system is not flexible cope that requirement. As business demands evolve and new requirements are introduced, the cost to enhance and comply with that requirement is perceived high. Lack of support and old framework are also the cause of high cost of maintenance. However, technology refreshment through legacy modernization can help organization reduce their expenditure significantly.

Poor architecture of legacy systems is also contributing to the high cost of maintenance. One characteristic of legacy system is point-to-point IT landscape. It will increase the complexity of the system as well as the cost to operating and maintaining them. To reduce this cost and complexity, organization needs make their IT landscape as simple as possible. They can do that by reducing the number of interfaces for their applications, with only one new interface for each system added. However, it cannot be done if legacy system is still there because legacy system has direct connection. Therefore, legacy modernization is needed to make the IT landscape as effective as possible to avoid high cost of maintenance.

Another example would be high cost in paying the maintainers who can support and maintenance the legacy system. Those people are often old and already have much experience in their career. They are also rare people, therefore organization have to pay them a lot.

Moreover, reducing the number of employee is something that organization did to save their expenditure too. Human resource is expensive. Some functionality that in old days were performed by human can be automated nowadays, so it is reducing the cost of operation. For instance, lack of capability of legacy systems in handling customers made organizations in the past handle that with providing numbers of customers care. However, since the era of the Internet and mobile applications are now common, organizations can reduce their expenditure by migrate their old system into the new system; so that they do not need many customers care anymore. The job in the past in which relying on human work, can be now automated by legacy modernization. Ultimately, save organization from high cost of maintenance and operation.

Last but not least, reducing the cost of maintenance and operation could be done by having standard way of business process. Making the processes standardized will require standardization in application as well. With this reason, organizations are moving the old application toward standard application that can benefit them. Moving to standard software has its own benefit such as having standard interface and interoperability between systems is easier. Also it can reduce the cost of license as well. License cost for mainframe is extremely high in many organizations. So, moving their applications to standard application can save them a lot of money.

In addition, old functionality and unnecessary part from legacy system can be thrown away after the process of modernization which can improve the quality of application. When everything is standardized, it makes maintenance and interconnection among system becomes easier. New requirements can also be filled quickly, which increase the flexibility of the system. Organization will not worry about scarcity of knowledge anymore since many experts and supplier available in market. License cost will also goes down dramatically by using standard software.

5.4.4. Limited suppliers / vendors to support the legacy systems

Supplier will support their client if their product still be able to bring them money. As soon as their product is not profitable anymore, they will stop supporting it. Since the user of legacy system are now moving to current technology, the cost of maintenance the systems now is huge from provider side. It is not a healthy situation in the provider side, thus, they prefer to just ending the lifecycle of the systems.

In the process of lifecycle of software product, software product will go through several phases, being Development, Testing, Acceptance, and Production. After operating in Production phase for several years, at one point, the software needs to be demolished. Usually, suppliers are hesitant to support their product anymore, because the product has been outdated or nearly retirement. It does not have business value anymore. At this point, the suppliers will have a discussion whether or not by continuing support the systems, they can derive the benefit.

However, in many cases, the discussion will end up with not supporting the software product anymore. One factor that makes supplier reluctant to extend the lifecycle of the product as it will take too much effort. A lot of effort for continuing to support the software that is not worthy enough with the profit they will get. Another factor is there might already be a substitute technology that works better than the technology that currently being used in the old software product. And by developing in new technology, supplier does not have to worry about the capacity of developer/programmer.

When the system is nearly in its retirement or the continuity of the system is almost end, and then it is the good time for organization to start thinking about the legacy modernization. Modernization to the current technology for sure can mitigate this problem, because current technology has many suppliers work in it. Therefore, it is suggested to do it as soon as possible. Otherwise, their business process can be disturbed by the limitation of the legacy systems.

5.4.5. Lack of experts around legacy systems

One of the popular reasons to modernize legacy system is a scarcity of knowledge, according to 24 participants. Organizations cannot find people to keep maintain their legacy system anymore. They realize that they are in the border where they cannot find the expert of legacy system easily. When they cannot find the experts to maintain their legacy system, their business is in dangerous because if the system failed, nobody can assist them anymore.

Legacy experts, who are still available now, are typically people who are nearing their retirement day and organization needs to replace these people with the new people, preferably with young people. However, they find it difficult to find competent people. It is difficult to attract young people to study the old legacy systems now. Most of them are not prepared to deal with legacy systems. New people are often being equipped with new technology that they learned from university. Not many courses are being taught in university or college make this topic is less attractive for students. Consequently, limited people are available in market to help organization deal with legacy system. With this situation, organization has no option except starting the discussion about legacy modernization to overcome the scarcity of knowledge. The longer they wait, the higher the risk of the legacy system to collapse.

The existing experts are crucial either to maintain the legacy system or to help in the process of legacy modernization. Thus, before the experts in legacy systems retire and gone with all the knowledge about the legacy system, some organizations take step first to modernize their legacy system. They, for sure, are the key player in legacy modernization, because they are familiar with the systems and know how the system behaves. They can also help with re-documentation, requirement management, testing and education. So that when they are retired, the organizations (hopefully) will have already finished with the legacy modernization project and can ready using their new system.

In addition, to get the experts, it is better to outsource them from the other parties to deal with the legacy system. It is more effective and efficient since train new people understand legacy system completely is time consuming and take a lot of effort. Also, because of rapidly changing in technology, by the time someone become an expert the game probably has already changed. Thus, more likely, they hire people externally who already expert in particular subject, to maintain their legacy systems

5.4.6. Prone to failures

Basically, the main driver to migrate the legacy system is the continuity, either continuity of the system or continuity of business. If the continuity is disturbed which can threat the lifecycle of the business process, then legacy modernization is not an option anymore, it becomes mandatory. One thing that can disturb the continuity is the increasing risk. Risk is one of the most influential drivers for legacy modernization because most organization cannot afford if their system goes down for only short time period. They will be suffering more. Therefore, the organization tries to prevent it before it happens. Such argument is the outcome of our interview analysis and particularly from project manager. They see this driver is crucial for legacy modernization.

Risk of failure of the system can come anytime, but mostly can be seen before the system has actually failed. Therefore, if the organizations reach at the point where they cannot guarantee the continuity of the system, then they should start thinking about legacy modernization. Risk of failure, loss of clients, and risk of over budgeting are driving organizations to initiate legacy modernization. They will lose money and in worst case if the system could not up and run again, their business is over.

5.4.7. Faster time-to-market of product

To satisfy the customer or customer demand, organization needs to improve the quality service of their system. Customers' demand could be delivering correct information instantly, quicker time to market, improving customer experience and GUI.

Those drivers often come from organizations that provide service or product to their clients (E.g. software house or message service provider). They are organizations, which provide software for other organizations. They also are doing the maintenance for their clients and their clients only have to use it and pay for the maintenance cost. The clients even do not know about technology behind the applications they use. All of technology part will be taking care by the software provider.

However, the clients often come with several new requirements in order to run their business process. When the new requirements are coming, software providers need to modify the system so that fulfill the demand of their client. Thus, launch the product as quickly as possible to the market is what they want. Usually this pressure came from sales and marketing department. These organizations admit that it would be difficult to realize this with old programming language and therefore this could be a driver to modernize their system.

In Figure 24, we summarize the cause-effect relationship of drivers of legacy modernization.

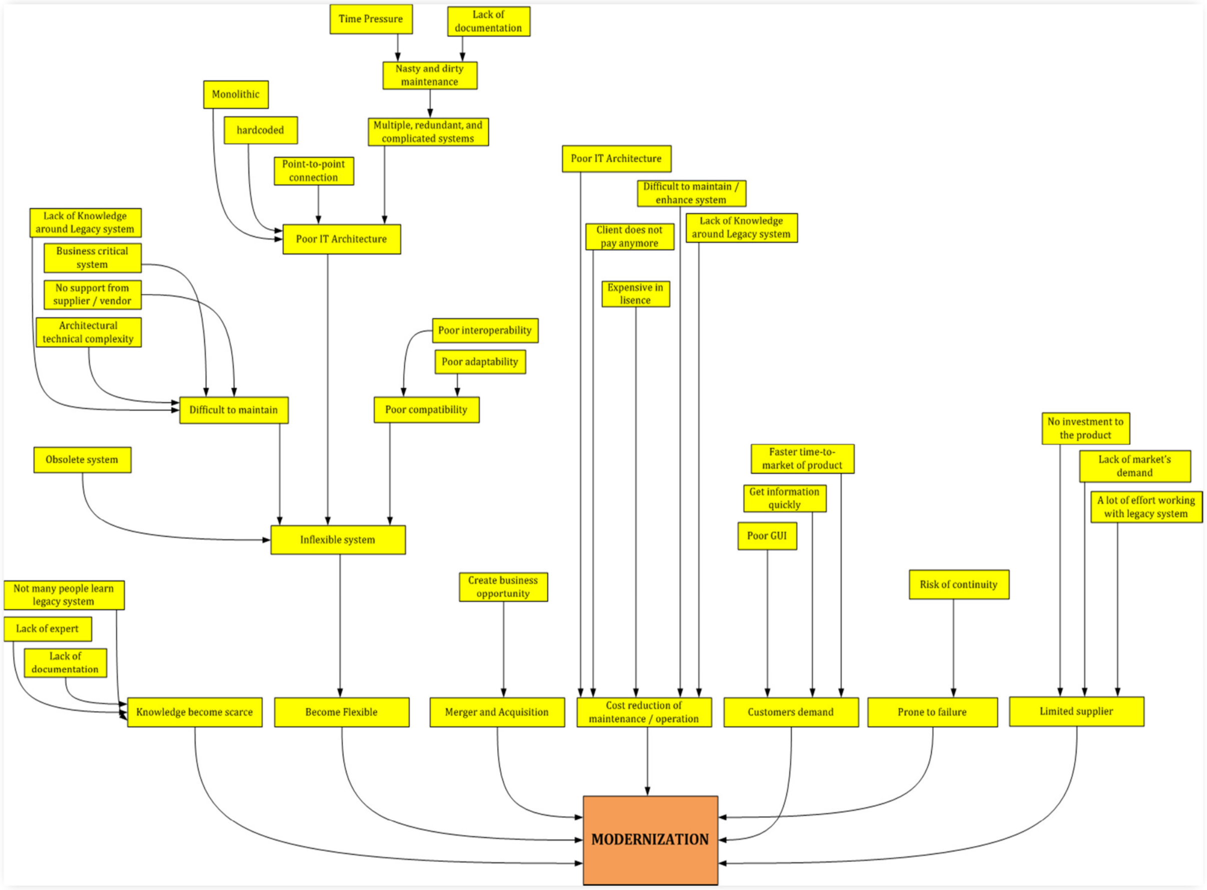

Time Pressure
Lack of documentation
Nasty and dirty maintenance
Monolithic
hardcoded
Multiple, redundant, and complicated systems
Point-to-point connection
Poor IT Architecture
Lack of Knowledge around Legacy system
Business critical system
No support from supplier / vendor
Architectural technical complexity
Difficult to maintain
Poor interoperability
Poor adaptability
Poor compatibility
Obsolete system
Inflexible system
Not many people learn legacy system
Lack of expert
Lack of documentation
Knowledge become scarce
Become Flexible
Create business opportunity
Merger and Acquisition
Poor IT Architecture
Difficult to maintain / enhance system
Client does not pay anymore
Lack of Knowledge around Legacy system
Expensive in lisence
Cost reduction of maintenance / operation
Faster time-to-market of product
Get information quickly
Poor GUI
Customers demand
Risk of continuity
Prone to failure
No investment to the product
Lack of market's demand
A lot of effort working with legacy system
Limited supplier
MODERNIZATION

Figure 24. Cause-effect relationship of "The drivers toward legacy modernization".

5.5. Challenges in legacy modernization project

If organization has decided to do modernization, there are some challenges that they have to confront.

5.5.1. Data Migration

Data migration is the most challenging factor in legacy modernization from technology side. It is so complicated that it can have its own project. However, failure in data migration means failure of the entire legacy modernization project. Therefore, all project managers in the participants' list suggested that good preparation regarding data migration should be taken into account before migrating the data.

The difficulty of data migration originally came from inappropriate maintenance of the system and database problem in legacy system. Since documentation of the legacy systems is not updated, the new developers lack necessary information to maintain the system as well as to implement the new functionality. Ad hoc and quick solutions are often executed in maintenance of the legacy systems. As a result data are saved not in right or correct format or in the wrong order.

Not only that, database schema has evolved over time, and not necessarily in an optimal fashion. During many years of operating, many kinds of changes in data structures have happened and as a consequence, many dirty and unstructured data are lying inside the legacy system. And legacy system tends to have bad database from the old days. It is also a reason to modernization because legacy system tends to have its database contains lots of inconsistencies and redundancies data.

Unstructured data is a main problem in data migration. The old database is originally non-referential database that make the data are unstructured. So, generally, unstructured data could be seen as any data that does not fit into the structured data definition. On the other words, it does not fit well into relational database. On the other hand, in new and modern database, it is mandatory to put the data in correct way that comply parent-child relationship. Thus, there is a discrepancy in the natural characteristic of databases between the legacy system and new target system. However, even in relational database in which usually contains structured data fields, sometimes there are data that are not structured. The fields with unstructured data were usually caused by people typing comments into a dialogue box or by a document file that was uploaded to the database. It often happens in legacy system.

The database problem could also be the result of merger and each of which has its own data format. It is not easy to combine those two data types. Improper use of field in database is danger because can be resulting miss-understanding and miss-interpretation in reporting. Therefore, it is aimed of data migration to improve the representation of data in database in the new environment. In addition, data in which reside in legacy database for long time period is often not. They are old and not actual anymore and often refer to data garbage

Therefore, it is suggested to provide proper attention on cleaning the data before executing data migration to get the good information and right information out of the legacy. Cleaning the data can extend the business value of core data that have been collected over many years. Such factor would be one of the considerations before organization migrate the data. Do the data will be migrated like what they are now? Or do the data need to be converted before the transformation? In one hand transformation of all the data like what they are now is easy and fast, but less realistic to achieve. The data will be transferred together with all their entire problems.

On the other hand, doing conversion is time consuming and needs more effort. Extra layer is needed to convert the data before they can be stored in new environment. It also holds high risks of missing data and incomplete data. Missing data could lead to incorrect or misleading information. At the end, the data are updated and their quality is improved in the new environment. Moreover, Seacord (2001) claimed in his paper that organization can gain some benefit if they improve the quality of data. The idea of improving the quality of the data is eliminating redundancy, improving the performance, reducing storage requirements and reducing the potential for database anomalies.

5.5.2. Lack of resources

Organizations often find it difficult to perform legacy modernization because they cannot find enough people to get the knowledge out of their current legacy system. Organizations do not know about what is going on with the system; what kind of effect if the certain code is being executed by the program. However, this problem is being handled well by building education around legacy transformation in particular country where people with knowledge about legacy system still available there. The idea is to help the producer of legacy system modernize their system that being used by their client.

To overcome lack of documentation, some companies try to do re-documentation in maintenance procedure using reverse engineering techniques. The purpose is to make it easier for the new people to learn how to maintain the systems too. Doing re-documentation also helps organization to get the knowledge reside in the legacy system. Such process is considered crucial later on, in the process of legacy modernization because the requirements from legacy system will be put against requirement from the new system in the testing phase. Organizations do also need people who can understand the old technology to help them understand the logic and help extracting knowledge from legacy system. These experts have a deep understanding and they are better equipped with their prior experience that can answer the right questions and identify the optimal answers. They know all the tricks in the legacy systems, which have been used in the past. However, those experts are rare and mostly expensive because they are external and organization desperately needs them

Another challenge is the need of variety people. Legacy modernization is not only about expert in legacy system, but also expert in database, operating system, middleware, enterprise services bus architecture and also expert from business. Business people should be involved as performing legacy modernization will be impacting business as well. They can also help in extracting business rules or business functionality from legacy system in order to get clear requirements.

5.5.3. Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)

Architecture of the system comprises components and connectors. Components could be tasks, processes, objects, programs, functions, library, etc. On the other hand, connectors are calls, invokes, signals, data flow, etc. Regarding the age of the legacy systems, they have old architecture style which is already obsolete and thus, equipped with old components and obsolete connectors. These components and connectors are not suitable anymore with today’s IT environment. Reusing components of legacy systems is almost impossible since the technology margin between legacy system and current technology is huge.

Regarding the problem of connectivity, sometimes there are systems around legacy system that play as a complementary system. Old legacy systems do not have capability of analytic (no intelligent system). When a system cannot perform analysis, third party analytical subsystems are integrated with the core system to accommodate those functionalities. As the time passes by, many subsystems are connected to the legacy system in order to access the data or function from legacy system. The process creates a chaotic IT with complex interconnection (spaghetti architecture).

Legacy system is also recognized as a large system because of poor in modified and changed the system. Sometimes some features are overlapping with each other due to improper way of modifying the system. Thus, the system becomes larger because during more than 10 years of operation, many features have been made by organization without realizing that they actually could use the old functionality. At the end, the systems become larger and consist of numbers of store procedure. All problems in architectural complexity are the challenges in legacy modernization. No wonder, making the IT landscape simple is also one of the reasons to modernize legacy system.

In addition, they are large systems because most legacy systems deal with huge data and have millions of line of codes. The codes line written in the application is more or less about 30 - 40 years of programming which makes it even difficult to modernize. Larger system together with overlapping features obviously results in complex software ecosystem.

5.5.4. Difficult to extract business rules/knowledge

As explained above, not many people know and/or understand the inner-function of legacy system, so extracting and documenting it are difficult. Two managers from Research and Development department also argued the same. The rules and knowledge in legacy system is becoming complex from time to time due to many requirements added in the past. No wonder legacy system is often seen as mysterious system because no one knows exactly about the systems. Lack of documentation and scarcity of expertise are believed the main reasons for this challenge. Furthermore, for some systems, the processes in legacy systems contain of all kind of exceptions and confused logic that will take some times to extract them properly. However, if this challenge cannot be handled properly, it would be difficult at the end to get clear and complete requirements.

Poor understanding of legacy systems can lead to incorrect target-system requirement specifications and ultimately towards failure of modernization projects. As the old proverb says that "*You can't truly solve a problem until you truly understand the nature of the problem you're solving*". In short, when organization fail to understand the essence of legacy system, they run into the risk of running legacy modernization that simply does not work. Thus, to begin with, organization should have a good understanding of the legacy systems including data, interfaces, and tool that can support the process of modernization. However, sometimes tools available are not sufficient to help the process of modernization the legacy system. Thus, some specialized tools need to be created in order to fit the legacy system and target system.

In addition, extracting knowledge from legacy system is crucial in legacy modernization because it can help the actors in modernization to understand the system in order to redevelop or migrate the legacy system. However, complexity of business logic and business rules in legacy system, bring extra difficulty in extracting the knowledge from legacy system.

5.5.5. Difficult to test

some of the modernization project, legacy data and database are extracted, transformed and populated into the new and modern environment that completely replace the legacy database. It could be an exhausting work since in general, data access layer –the protocol for application to communicate with the database- must be re-engineered, and this require extensive tests. As Bisbal et al. (1999) also expressed that up to 80 percent of a migration engineer's time can quite legitimately be spent testing the target system, because by nature, legacy system holding mission critical of the organization. Therefore, the output should be completely consistent with those in legacy system.

The process of testing becomes more difficult since legacy modernization takes many years that during the period new functionality are introduced. It makes the testing more difficult. And when the testing phase is difficult, it is likely that the cost would also increase. Moreover, the size, complexity, budget, the testers, data volume and the time scope also influence the difficulty of the testing.

In order to complete the testing, organizations usually do parallel testing or shadow testing to test their new system. However, testing becomes more difficult if it involves the whole integration part. Sometimes finding where exactly the cause of the error is difficult and time consuming. Thus, log system during legacy modernization is recommended to keep track all the activities during testing phase.

5.5.6. Funding legacy modernization projects

Funding of course is the major problem in legacy modernization. Experience from the all consultants in interview session said that to get funding for executing legacy modernization from top management is really difficult. Everything on proposal need to be in-line with what organization needs, the business benefit of doing legacy modernization, and the time period that would take before the organization can get its benefit. On the other words, it is a matter of creating the proper business case. Proper business case can definitely determine the success and the failure of proposing the legacy modernization project. For many CIO's the difficulty is to come up with some trustful numbers. There are a lot of considerations in calculation before they can create a proper business case.

5.5.7. Predicting Return of investment (ROI) of modernization

ROI of legacy modernization project is normally long. It's natural that if company spends their money, they expect to get it back as quickly as possible. Some organizations even do not want to do modernization because they know the ROI cannot be met in short time after the modernization. However, these organizations should be careful because as mentioned above, the longer they wait, the higher the risk of failure of the system will be.

Although many organization expect to get their ROI back as soon as possible, but it is not suggested to execute legacy modernization. Legacy modernization by itself is not an easy job and would take years to finish it completely. Making it short will increase the failure rate significantly. So it is suggested to take into account ROI and time length before doing legacy modernization.

To have better prediction of ROI, organization needs to follow the technology trend. Thus it is advised that the ability to foresee the IT movement can determine the successful or failure of modernization process. The right choose of technology can make the new system have the long life cycle and can stand longer.

5.5.8. Cultural resistance in organization to adapt new system

Due to the high percentage of legacy system, companies in general are not reluctant to do modernization. However, in legacy modernization, cultural aspect plays vital role. According to Heskett (2011), culture can account for 20-30% of the differential in corporate performance when compared with 'culturally unremarkable' competitors. Organizational culture in general is structured by members' shared values, beliefs, symbols, and behaviors that can bring everyone on a common platform and being a compass providing direction for organization as a whole.

Many organizations find that this is the most challenging part in legacy modernization, especially for the project manager. The organization need to be ready for changing the way in their business process. Changing the culture of the organization is not only dealing with people who work in the organization, but also the resource available around the organization.

In interview session, most organizations declared that the challenge in legacy modernization project is not about technology, but more into cultural and human aspect. Change cultural adaptability is the most challenging aspect for many organizations. So, it should be the first consideration before migrating the legacy system. People sometimes do not like changes and can significantly delay legacy modernization. There are people who have all kind of different motivation not to work along. To overcome this challenge, people in top management are required to address this problem by persuade them and explaining on how and why modernization is necessary.

More than that, educating people after the modernization is perceived crucial. Not every people are able to use the new technology. So, it is suggested before doing modernization, it is strongly recommended to know whether or not organization is ready for changing.

5.5.9. Resistance from the current users/maintainers in the organization

The resistance from user is divided into two parts, the resistance of end-users and the resistant of maintainers. Both of them have their own motivation to reject the implementation of the new system. From end-users' side, they are already comfortable with the old system. They are familiar on using the old system that they just do not like to change it. Also for these types of people, they do not like changes because they do not want to change the way they work now as. Change the system will mostly change the way the process and change their work habit. It needs adaptation and which all employees may not like.

From developers' or maintainer's side, they are most of the time old people who are maintain the system for long time. Sometimes, they treat their system like their own baby and know everything about the systems. They are satisfied with it and do not want to let it go. When new systems come they try to protect their system. That is their area of expertise. Losing legacy system makes them have to learn new things again in their old age and everyone may not like it, particularly, people who nearly their retirement day. Therefore, they may not cooperate with the legacy modernization process.

5.5.10. Difficult to communicate the reasons/consequences of modernization

The big obstacle in culture of organization is communication among all people who are involved in the modernization process. People inside the project are the people who are involve directly in the legacy modernization process. Working between them, for example with the maintainer is sometimes difficult because they have their own perception on how to handle the system they are familiar with. Therefore, regular meeting and communication between them to find solution, strategy or next steps is the best way to handle the different perception. Also involving business people is needed as legacy modernization is also about business. It is perceived crucial, so the business people know what IT people are doing and can give their opinion from business point of view.

In order to get accepted by top management, CIO has to come up with a good business case. It is a challenge too. The challenge to propose the process of modernization is to explain the current situation and the ideal situation as a solution and make them approved by management board in an approval concept. The approval concept also deals with financial perspective such as what is the revenue? what are the cost and how long it could be to get meet Return On Investment (ROI)? More than that, analysis about risk involve in the process of modernization as need to be presented. It is the key to convince the top management that legacy modernization is an urgent matter.

Risk evaluation is difficult if organization cannot accurately foresee the future of IT trends. So it is recommended to have Research and Development department with ability to see the new opportunity of new technology.

Another difficulty is topic such legacy modernization or data transformation are not an interesting topic to be discussed with top management. Top management rather to discuss about strategy and money involve in the business process. They do not like to talk about technical limitation caused by legacy system. But once top management has decided to support legacy modernization, it can make legacy modernization less difficult.

Regarding communication problem, participants also suggested that people who have initiative to do modernization also have to convince and persuade others to support the modernization too. Such situation, for some IT department is difficult since it needs special skill in human interaction and this is not what people in IT are expert on. Thus, team leader or project manager will have a big influence in the process of modernization.

5.5.11. Difficult to effectively prioritize the functionality for modernization

Another challenge of legacy modernization is to governance the modernization process itself. Finding a good modernization path/process is difficult because normally legacy modernization takes years and during that period something might change the direction of modernization. The project of modernization might lost its focus because changes in requirements during the process of modernization. And sometimes the priority changes quickly. So, no wonder attention goes away from modernization to specific features. Therefore, solid governance of legacy modernization project is required in order to make it successful.

However, some flexibility during the process of legacy modernization is also needed. Otherwise the project will deliver the old requirements. Moreover, migration projects are often expected to add functionality to justify the project's expense and risk. In this case, the legacy system should be migrated first. New functionality can be incorporated into the target system after the initial migration.

The other difficulty is the need of different experts, such as business people, IT people (developers) and project manager. Legacy modernization touches many aspects of the organization, thus collaboration among experts is needed to conduct the project successfully. It is difficult since legacy system tends to have not many experts to work around it. Each expert tends to have their own perspective and their own solution to modernize legacy. Thus, the project needs someone who can manage all of the experts. It can be achieved by having good project manager and good support from top management. The project manager has a significant role in legacy modernization project because it is his/her job to address the causes of problems as well as solve them effectively. However, to get the charismatic leader as a project manager is not easy.

Beside his role to communicate the plan for people who are involved in modernization, project manager also has to govern the process of modernization. By right direction, it means that doing legacy modernization effectively and efficiently. Theoretically, by defining suitable process for the planning, execution and controlling the project can improve the rates of success.

The other challenge is to do modernization while keep maintain the old system. Legacy modernization normally takes years to be finished. During the process of legacy modernization, sometimes organizations still need their original system, so they still have to maintain it. Therefore, there is a period when legacy system and the new system are running together.

5.5.12. Time constraints to finish legacy modernization

Project duration could be a challenge for organization too when knowledge around legacy system is scarce. It can result in insufficient time and over budget to finish it. Since most of the organizations also have to work under the time pressure, so it is mandatory to have good time management.

With good time management, it is also possible to do some researches beforehand in order to find good and proper solutions for the problems and challenges in legacy modernization. It is the main reason of why organization does not look on academic resources before doing legacy modernization. However, research on academic area is not in their top priority. Not only that, by having good time management legacy modernization be done appropriately.

With regards to this problem, organizations experimented to automate the legacy modernization process as much as possible. Try to minimize human intervention is one of the solution in time management. However, it is not easy as it sounds. All participants expressed their problem of automation in legacy modernization. It seems that automatic transformation of legacy system is still far away from ideal result that organization expect. It is also supported by Newcomb (2001) through his research. In case of automation legacy code he said that 60% or less of a system’s legacy code automatically results in extensive amounts of untransformed code in which still has to be addressed manually.

Figure 25 depicts the cause-effect relationship of challenges in legacy modernization.

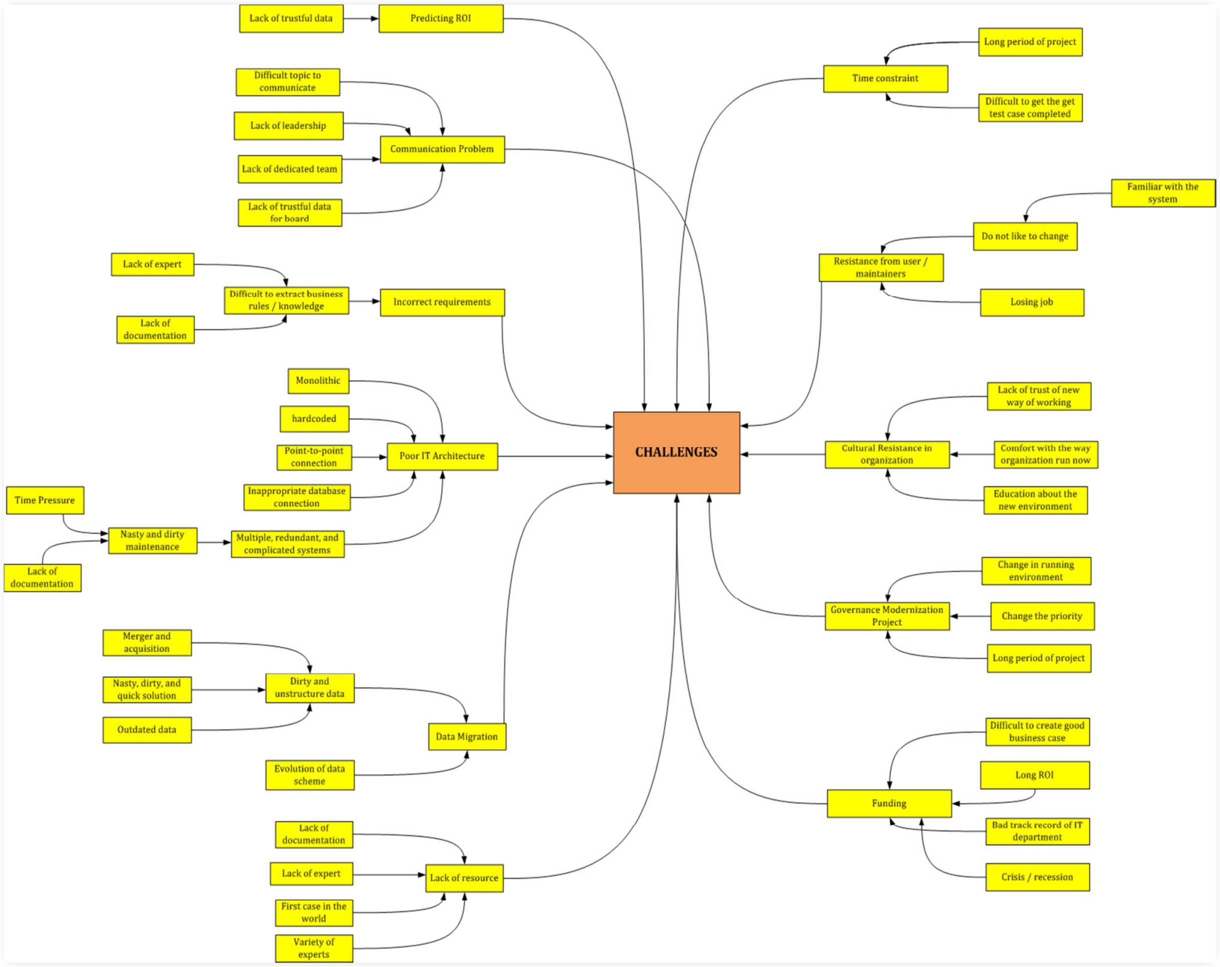

Lack of trustful data
Predicting ROI
Long period of project
Time constraint
Difficult to get the get test case completed
Difficult topic to communicate
Lack of leadership
Communication Problem
Lack of dedicated team
Lack of trustful data for board
Familiar with the system
Do not like to change
Resistance from user / maintainers
Losing job
Lack of expert
Difficult to extract business rules / knowledge
Incorrect requirements
Lack of documentation
Monolithic
hardcoded
Point-to-point connection
Poor IT Architecture
Inappropriate database connection
CHALLENGES
Lack of trust of new way of working
Cultural Resistance in organization
Comfort with the way organization run now
Education about the new environment
Time Pressure
Nasty and dirty maintenance
Multiple, redundant, and complicated systems
Lack of documentation
Change in running environment
Governance Modernization Project
Change the priority
Long period of project
Merger and acquisition
Dirty and unstructure data
Nasty, dirty, and quick solution
Outdated data
Data Migration
Evolution of data scheme
Difficult to create good business case
Long ROI
Funding
Bad track record of IT department
Crisis / recession
Lack of documentation
Lack of expert
Lack of resource
First case in the world
Variety of experts

Figure 25. Cause-effect relationship of "The challenges in legacy modernization".

6. Validation

Triangulation is another technique to validate the findings. Triangulation is a process of taking different angles toward the studied object and thus providing a broader picture. For this research, there is one triangulation applied which is data triangulation. Data triangulation is a way to use more than one data source of collecting the same data at different occasions. It is recommended to use several data sources in order to limit the effects of one interpretation of one single data source. If the same conclusion can be drawn from several sources of information, the conclusion is stronger than a conclusion which just based on a single source. Data to support the findings can be derived from interview or by examining documents such as journals, magazine, papers, etc. for the historical qualitative information.

The basic idea of triangulation is to gather different types of evidence to support a proposition. The evidence might come from different sources, be collected using different methods, be analyzed using different methods, have different forms (interviews, observations, documents etc.), or come from a different study altogether (Seaman, 1999). Therefore, for this research one of data triangulation comes from a survey.

6.1. Survey

A survey is a collection of standardized information from a specific population or some sample from one, usually, but not necessarily by means of a questionnaire or interview (Robson, 2002). The survey of this research will be in the form of online survey and will be spread out to industry participants. The objective of the survey is to solicit information about the legacy systems and the legacy modernization in order to validate the finding derived from the interview sessions.

Before the survey is online, the preparation for survey covers questions was prepared based on the findings from the interview. The questions were made based on previous study with the combination of the author’s knowledge. Once the survey is online, there is no way to change the survey. Therefore, the preparation of survey must be well prepared.

Prior to spreading the survey, the target population has to be determined. In this research, the target population is people who are in expert in legacy modernization. There are several groups on LinkedIn that focus on the legacy systems and the legacy modernization, being:

- IEEE-SCAM
- Legacy migration
- Legacy modernization
- TOGAF
- Mainframe Migration
- Amadeus alumni
- Shell Australia IT Network
- Software Improvement Group
- TCS - Tata Consultancy Services
- Application Modernization and APM
- Enterprise Legacy Modernization (ELM)
- Mainframe Experts Network
- Cognizant Technology Solutions
- Legacy AS400 Iseries IBM i Modernization
- Centrum Wiskunde & Informatica - Employees & Alumni of CWI and Mathematisch Centrum
- Master Business Informatics - Utrecht University
- IBM Enterprise Modernization Professionals
- Shell Information Technology International

The author also used every possibility to spread the survey. Beside groups in LinkedIn, the author also used several groups in Facebook, being Software Engineering Research Community, MBI (Master Business Informatics) Gang, and Business Informatics Alumni. Some of colleagues also helped spread the survey in their own organization.

The survey contains a series of questions that capture the demographic background of the participants, questions about legacy systems and also about legacy modernization. From the first part (background about participants), the author tries to collect demographic information about the respondents and their experience. The aim is to characterize the respondents based on their organization and their position. Specifically, the author asked about their business domain, and the years of experience with the legacy systems.

The questions about legacy systems attempt to capture the respondents’ perception about legacy system together with the problems caused by it. It includes the characteristics (good and bad) associated with the legacy systems. In next part, the questions are trying to capture participants experience about a legacy modernization project such as drivers and challenges. In addition, there were some questions which asked about how the academic product can help the process of modernization.

In total, there are 17 questions (including the question that asks the respondents to provide their email address if they want to receive the result). However, there are some logics in the questions to make the survey simple. These logics will allow respondents to skip the questions which are not applicable to their experience. In order to get as many respondents as possible, a large portion of the survey questions are multiple choices and consist of only three open questions. All of them can be completed in approximately 10 minutes and was designed accordingly.

The author used an online survey using esurv.org (http://esurv.org). The online survey has some benefits compare with paper-based survey of email-based survey. By using online survey, the respondents can answer the questions easier and quicker. Hence, the data entry is easier and more structured. The study also shows that web-based questionnaires have a high return rate (Tomassetti et. al., 2012).

6.2. Survey Result

6.2.1. Data and statistic

The survey received totally 104 respondents from various countries and domains until 18th July 2013. The domains and the positions of participants are listed in table 5 below:

Netherlands (53)	USA (13)	UK (6)	India (5)	Italy (3)
Singapore (3)	Germany (3)	Portugal (2)	Israel (2)	Finland (2)
Spain (1)	Nepal (1)	Malaysia (1)	Belgium (1)	France (1)
Poland (1)	DR Congo (1)	Canada (1)	Greece (1)	

Table 4. Countries of origin of respondents.

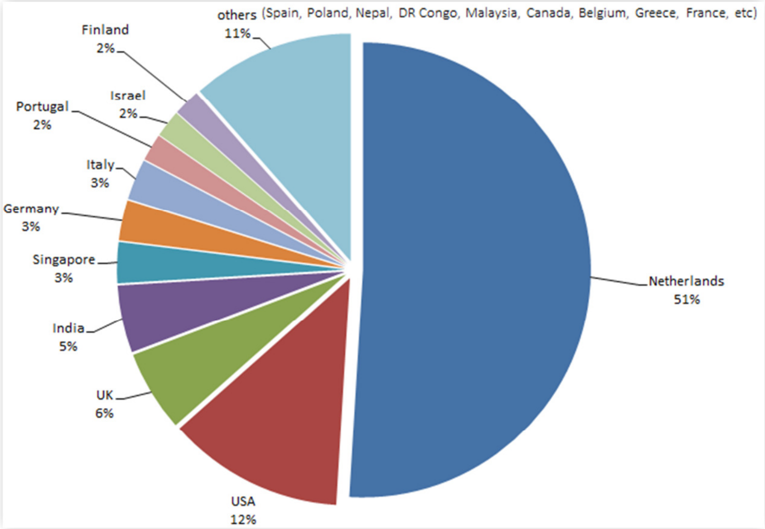


Figure 26. The origin of the respondents.

However, there were 3 respondents which were excluded due to invalid and/or incomplete data. The domains of respondents are also varied. They are presented in table 6.

Software Security (4)	IT Researcher Institution (4)	Academic (8)	Online retailer (1)	Service Provider (20)
Consultant Company (31)	Financial Institution (11)	Industry (1)	Education (1)	Healthcare (2)
Software Development Company (45)	Government institution (9)	Logistics (1)	Public transport (1)	Media (1)
Rail Infrastructure Provider (1)	Telecommunications (1)	United Nations (1)	Bio Tech Org (1)	

Table 5. Domain of the respondents.

The respondents also came from many different positions in their career presented in table 7:

Developer	Software Architect	Architect	Business unit manager	Sr. Consultant
System Analyst	Solutions/data Architect	IT manager	Development Manager	Consultant
Quality assurance	Information Architect	CIO	Product manager	VP Sales Marketing
Support for Application	IT Architect	CTO	Product leader	Sales, Sales Engineer
System support	Chief architect	CEO	Project Management	Marketing
Solution Architect	IT and enterprise architect	Researcher	SAP consultant	Business Analyst

Table 6. Position of the respondents.

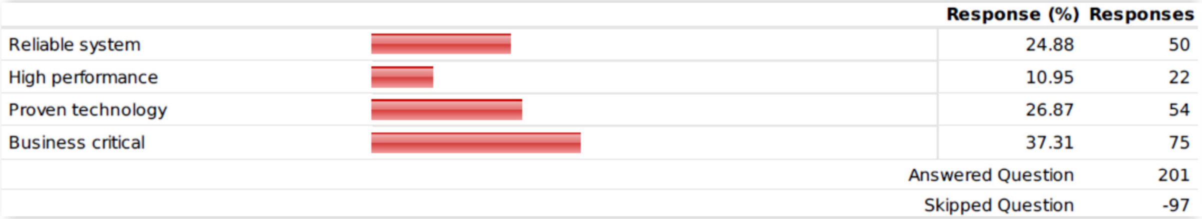
However, all positions have been categories into two groups, being Practitioner and Academic. The practitioner is also divided into 2 parts which are Technical and Business. Table 8 shows all the positions.

Practitioner	**Technical**	Architect (Architect, IT Architect, Chief architect, IT and enterprise architect, Information Architect, Solution Architect, Software Architect, Solution/Data Architect)	14
		Developer	25
		System Analyst	14
		Quality assurance	1
		Support for Application	1
		System support	1
	Business	Marketing (VP Sales Marketing, Sales, Sales Engineer, Marketing)	4
		Top Management (IT Manager, CIO, CTO, CEO)	34
		Manager (Business unit manager, Development Manager, Product manager, Product leader, Project Management)	4
		Business Analyst	16
		Consultant (SAP consultant, Sr. Consultant, Consultant)	4
Academic		Researcher	12
		Total	**130***

*Note: the total number is 130 mainly because one person could have more than one position

Table 7. Position of the respondents

6.2.2. Positive characteristics of the legacy systems



	Response (%)	Responses
Reliable system	24.88	50
High performance	10.95	22
Proven technology	26.87	54
Business critical	37.31	75
	Answered Question	201
	Skipped Question	-97

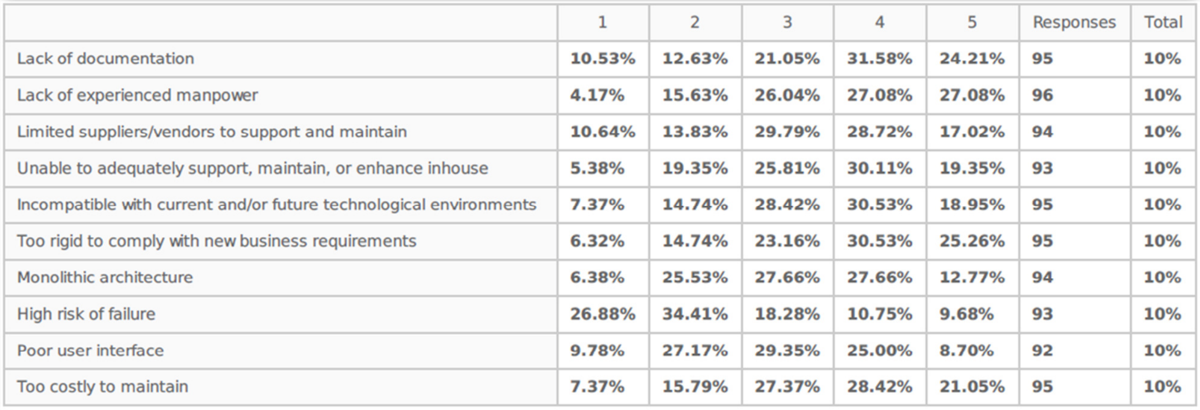
Figure 27. Statistic in positive characteristics of the legacy system.

From the above graph, the positive characteristics of legacy systems are apparently dominating by respondents from the business area. *The business critical system* is the most popular characteristic by 75 respondents, following by *Proven Technology* (54 respondents), *Reliable system* (50 respondents), and *High performance system* (22 respondents). The details of who said what can be seen in the appendix section.

In general, most of the participants agree that legacy systems are *business critical*, *proven technology* and *reliable system*. More respondents support the findings that legacy systems are the core systems and profitable system of the organization. They also agree that legacy systems are old and have good availability that makes them as a proven technology. The same result is acquired when the respondents voted for the reliable system. The respondents voted for the robust and the stable system that are the properties of the systems for being reliable.

It is supporting the findings derived from an interview session. However, the *high performance* of the legacy system has the lowest vote among the others. It may be due to the different type of legacy system the respondents associated with (e.g. mainframe, PC, super computer, etc.). Moreover, the time behaviors of the legacy systems vary from one interviewee with the other interviewees in an interview session. More discussion about this matter is in Chapter 7.1.

6.2.3. Problems of legacy systems



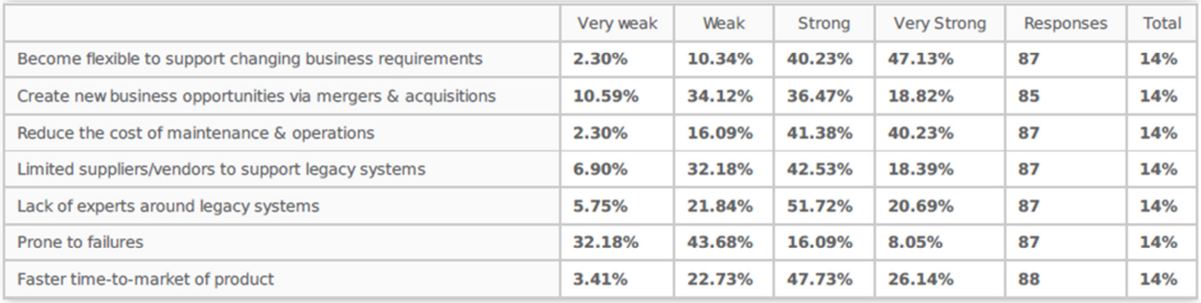
	1	2	3	4	5	Responses	Total
Lack of documentation	10.53%	12.63%	21.05%	31.58%	24.21%	95	10%
Lack of experienced manpower	4.17%	15.63%	26.04%	27.08%	27.08%	96	10%
Limited suppliers/vendors to support and maintain	10.64%	13.83%	29.79%	28.72%	17.02%	94	10%
Unable to adequately support, maintain, or enhance inhouse	5.38%	19.35%	25.81%	30.11%	19.35%	93	10%
Incompatible with current and/or future technological environments	7.37%	14.74%	28.42%	30.53%	18.95%	95	10%
Too rigid to comply with new business requirements	6.32%	14.74%	23.16%	30.53%	25.26%	95	10%
Monolithic architecture	6.38%	25.53%	27.66%	27.66%	12.77%	94	10%
High risk of failure	26.88%	34.41%	18.28%	10.75%	9.68%	93	10%
Poor user interface	9.78%	27.17%	29.35%	25.00%	8.70%	92	10%
Too costly to maintain	7.37%	15.79%	27.37%	28.42%	21.05%	95	10%

Figure 28. Statistic in problem of legacy system

As we can see in the Figure 29, the most critical problem of the legacy systems is a *lack of experience manpower* and the least critical problem is a high risk of failure. It is also in-line with the result from interview session which claimed that the most critical problem of the legacy systems is a scarcity of knowledge. The other problems of the legacy systems also support the result of the interview session. Almost all problems in survey part had been seen as the critical problems as same as the finding derived from interview sessions.

On the other hand, only a high risk of failure is perceived as quite a critical problem in survey part. During the interview sessions, although the participants argued that high risk of failure is a problem of legacy systems, but it was not mentioned often. Organizations do not usually wait until their systems fail before they fix it. Therefore, if they see some indications that the system is going to fail, they then immediately fix the systems and the risk is mitigated. In addition, there is a possibility that legacy systems do not pose high risk of failure yet when the interview was conducted. Chapter 7.2 discusses about this matter and the details about all the problems are shown in appendix section.

6.2.4. Drivers toward legacy modernization project



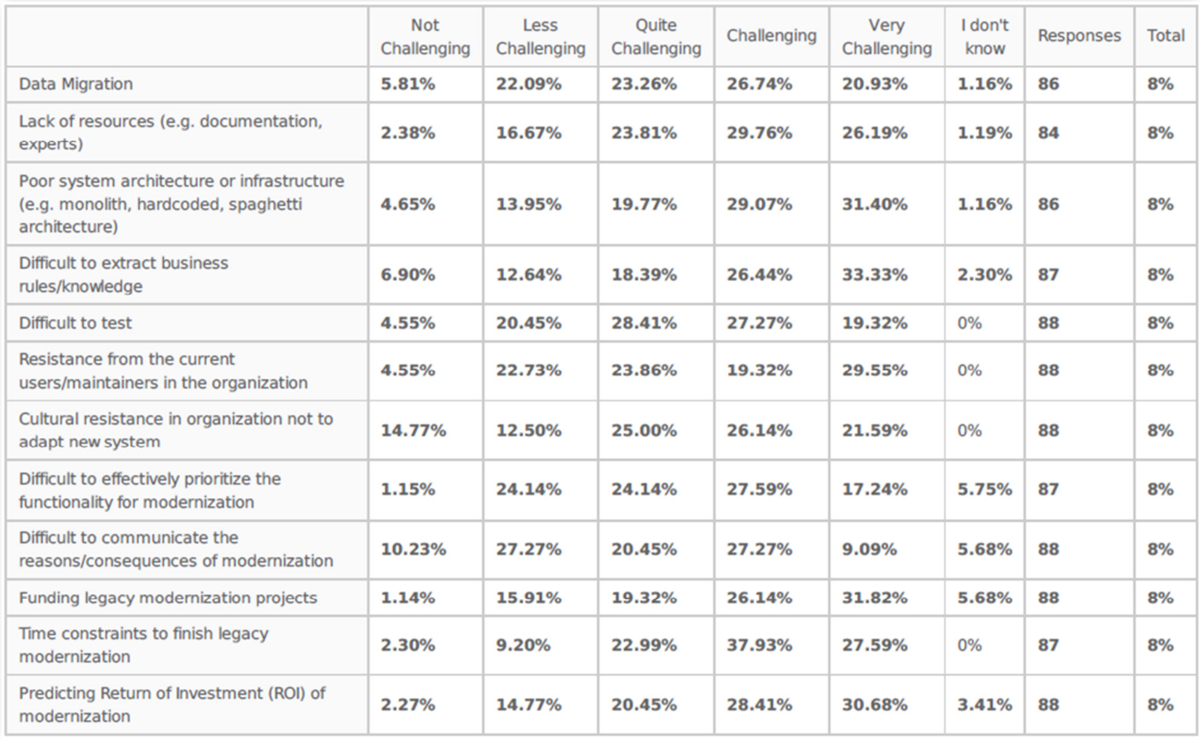
	Very weak	Weak	Strong	Very Strong	Responses	Total
Become flexible to support changing business requirements	2.30%	10.34%	40.23%	47.13%	87	14%
Create new business opportunities via mergers & acquisitions	10.59%	34.12%	36.47%	18.82%	85	14%
Reduce the cost of maintenance & operations	2.30%	16.09%	41.38%	40.23%	87	14%
Limited suppliers/vendors to support legacy systems	6.90%	32.18%	42.53%	18.39%	87	14%
Lack of experts around legacy systems	5.75%	21.84%	51.72%	20.69%	87	14%
Prone to failures	32.18%	43.68%	16.09%	8.05%	87	14%
Faster time-to-market of product	3.41%	22.73%	47.73%	26.14%	88	14%

Figure 29. Statistic in drivers toward legacy modernization.

Some drivers that had been identified in interview session were asked again in survey part. Overall, the respondents from survey part agree with the drivers from participants in an interview session. All the drivers are perceived strong from the participants in interview session and respondents in the survey. It indicates that the legacy modernization is urgent and need to be discussed imminently. Therefore, organizations are actually not reluctant to do legacy modernization.

However, prone to failures of the systems is perceived weak in survey result. It is understandable since the high risk of failure in the problem of legacy systems is also perceived less critical than the other problems. Chapter 7.2 discusses about this matter and the details about all the problems are shown in appendix section. At the end the result from the interview has been proven true by the result from the survey.

6.2.5. Challenges in legacy modernization project



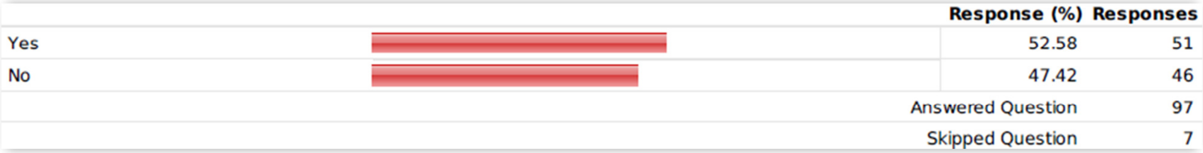
	Not Challenging	Less Challenging	Quite Challenging	Challenging	Very Challenging	I don't know	Responses	Total
Data Migration	5.81%	22.09%	23.26%	26.74%	20.93%	1.16%	86	8%
Lack of resources (e.g. documentation, experts)	2.38%	16.67%	23.81%	29.76%	26.19%	1.19%	84	8%
Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)	4.65%	13.95%	19.77%	29.07%	31.40%	1.16%	86	8%
Difficult to extract business rules/knowledge	6.90%	12.64%	18.39%	26.44%	33.33%	2.30%	87	8%
Difficult to test	4.55%	20.45%	28.41%	27.27%	19.32%	0%	88	8%
Resistance from the current users/maintainers in the organization	4.55%	22.73%	23.86%	19.32%	29.55%	0%	88	8%
Cultural resistance in organization not to adapt new system	14.77%	12.50%	25.00%	26.14%	21.59%	0%	88	8%
Difficult to effectively prioritize the functionality for modernization	1.15%	24.14%	24.14%	27.59%	17.24%	5.75%	87	8%
Difficult to communicate the reasons/consequences of modernization	10.23%	27.27%	20.45%	27.27%	9.09%	5.68%	88	8%
Funding legacy modernization projects	1.14%	15.91%	19.32%	26.14%	31.82%	5.68%	88	8%
Time constraints to finish legacy modernization	2.30%	9.20%	22.99%	37.93%	27.59%	0%	87	8%
Predicting Return of Investment (ROI) of modernization	2.27%	14.77%	20.45%	28.41%	30.68%	3.41%	88	8%

Figure 30. Statistic about the challenges in legacy modernization project.

The challenges in legacy modernization projects are same between the result from interview session and survey part. The challenges can be divided into 3 parts, namely technical, business and organizational challenges. Business challenges include *funding* and *ROI prediction* are perceived challenging in survey part. The technical challenges in legacy systems (*data migration*, *scarcity of knowledge*, *poor architecture*, *knowledge extraction* and *testing*) are mostly perceived challenges. The organization challenges which include *cultural adaptability of the organizations*, *resistance from current users*, *communication*, *functional prioritizing* and *time management*, are also perceived challenges.

It seems that business and organizational challenges are more challenging than the technical challenges. The argument is based on the fact that more participants in interview sessions said that technology challenges can always be solved. However, human aspect and cultural aspect are even more difficult. It is also supported by survey results. Almost all challenges are dominated by business domains respondents than technical respondents. The details are shown in appendix section.

6.2.6. Programming Language consideration



	Response (%)	Responses
Yes	52.58	51
No	47.42	46
	Answered Question	97
	Skipped Question	7

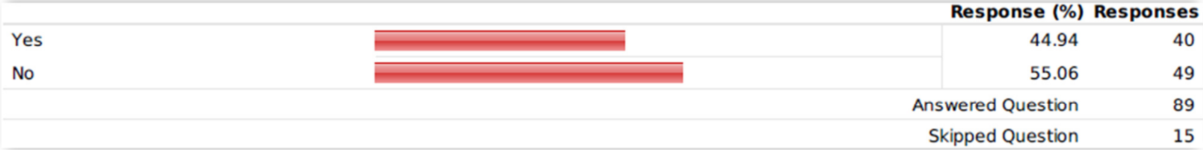
Figure 31. Statistic about programming language in legacy system.

There are slightly different result from interview session and survey in regard to programming language. The meant of programming language is whether or not the programming language can determine the systems for being legacy. The result from interview session is programming languages do not play much in determine the systems for being legacy. However, the result from survey said that programming language determines the systems for being legacy. Thus, the result is slightly different.

Although the result is different, but the different is not significant. The number of respondents who say that programming language determines legacy systems is not much different with respondents who say programming language does not determine legacy systems. More respondents who YES are coming from technical domain while more respondent who said NO are coming from business technical. It is in line with the result from interview session, which participants mainly have business oriented knowledge.

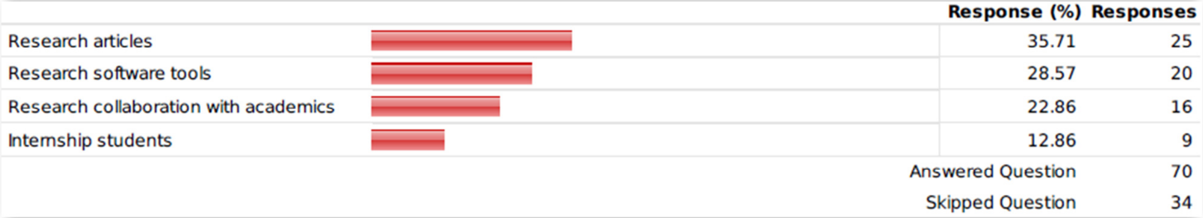
6.2.7. Academic involvement

When respondents were asked about academic involvement, more of them do not use academic resource in their legacy modernization project. However, the different is not significant with the respondents who use academic resource in their legacy modernization project as shown in Figure 33.



	Response (%)	Responses
Yes	44.94	40
No	55.06	49
	Answered Question	89
	Skipped Question	15

Figure 32. Statistic in academic involvement with legacy modernization project.



	Response (%)	Responses
Research articles	35.71	25
Research software tools	28.57	20
Research collaboration with academics	22.86	16
Internship students	12.86	9
	Answered Question	70
	Skipped Question	34

Figure 33. Statistic in resource of academic involvement.

The result from survey is in line with the interview part. Participants in interview parts often do not refer to academic products when they had legacy modernization project. The most common reason it because academic is too technical for them and do not have enough business aspects in it. Therefore, organizations are reluctant to involve academic in their project.

As shown in Figure 34, the most popular resource from academic is research articles (journal, paper, white paper, etc.) with 25 respondents. The reason is because it is the easiest way to get information from academic. The research software tools is the second popular academic's resource with 20 respondents, followed by research collaboration with academic (16 respondents) and Internship students (9 respondents).

Internship students is the least voted option since it is not easy to adjust student's purpose and organization's purpose. Students often follow the protocol from a university or college that is not in line with what organization objective. Therefore, this is the least popular way to use academic product.

7. Discussion

The results from the survey are in line with the findings of the interview part.

7.1. Positive characteristics about legacy systems

The finding from interview session is in line with the result from survey. We found several positive characteristics of legacy systems. They are *BUSINESS CRITICAL, PROVEN TECHNOLOGY, RELIABLE SYSTEM,* and *GOOD PERFOMANCE*. These finding are supported by our respondents in survey.

First of all, legacy systems are business critical systems. From the interview session, most of the participants agree that legacy systems are the core systems in their organization. It is supported by their comments that legacy systems are the core systems and back office systems. Back office systems are most of the time deal with vital functionalities that are not often changed. Moreover, those kinds of systems are usually profitable system because they are accompanying organization for many years until now. That is how the participants in an interview session recognized their legacy systems. The result from the survey shows the same thing. Seventy-five respondents from 101 respondents (73,5%) agreed that the legacy systems are the business critical system. In addition, the *business critical* is the characteristic of the legacy systems that most business-related respondents voted for. Fifty-two percent business-related respondents agree with this characteristic, while only 39% technology-related respondents say the same. Even 20 out of 24 IT managers, 4 consultants, and 3 CEO's voted for this characteristic.

Secondly, legacy systems are also proven technology according to the result from interview session and the survey. Proven technology is supported by the fact that legacy systems are stable, functioning well and they are old systems. According to participants in interview part, the old systems mean that the systems have been running for decades and functioning well. They have been modified and adjusted to be in-line with organizations' expectation in order to fulfill their business process. Similarly, the proven technology was seen as an positive characteristic of the legacy system in the survey part. It can be seen from the fact that this characteristic was voted by 54 respondents (53,4%). However, this is the only characteristic that attracted more technology-related respondents (49%), while only 41% business-related respondents voted for this characteristic. Most developers are voting for this characteristic (60% of all total developers). Same with the Architects (57,1%) and the System Analysts (50%).

The third good characteristic is the *reliable system*. A reliable system is determined by the stability and robustness of the system. Legacy systems do not crash often while operating the business processes. Even though at some point they gave an error, but it was not disturbing the business processes. Thus, the availability of the systems to deliver their functions is perceived well according to the participants in an interview session. That is why they see legacy systems are stable and robust. The result from interview is the same with the result from the survey. At least 50 respondents agreed that legacy systems are reliable system. Moreover, this characteristic is supported by 52% business-related respondents and 42% technology-related respondents. Less technology-related respondents voted in this characteristic. Although half of System Analysts (50%) voted for this characteristic, but only 44% from total Developers and 42,8% from total Architects voted for this characteristic.

Last but not least, is the performance of the legacy systems. In general, the performance of legacy systems in interview session is perceived good. It is supported by the availability of the systems and good functionality of the systems. However, when participants were asked about the time behavior of the legacy systems, the result is almost equal between bad time behavior and better time behavior. It is due to the fact that the legacy systems are mostly associated with the mainframes. Thus, when participants refer to the mainframe, they tend to see that legacy system has a good time behavior because mainframe has a good response when deals with bulk or batch files. In total, only 22 respondents see the performance of a legacy system is good. However, from 22 respondents, there are 17 respondents with experience less than 20 years. It could be an indication that they might not have experience with the old mainframe. Thus, they might not see the good time behavior in the mainframe.

From all the respondents, business-related respondents are more dominating to see the positive characteristics of legacy systems. The completion percentage can be seen in table 9. It is in line with the result from participants in an interview session in which most of them were business oriented people. In addition, with this fact, organizations are not reluctant to do legacy modernization, because most of top management and CEO see legacy systems as a business critical that still have business value in it with all problems surround it.

Reliable ***Business Domain (52%)****; Technology domain (42%); Academic domain (6%)*	High Performance ***Business Domain (51%)****; Technology domain (42%); Academic domain (7%)*
Proven Technology *Business Domain (41%);* ***Technology domain (49%)****; Academic domain (10%)*	Business Critical ***Business Domain (52%)****; Technology domain (39%); Academic domain (9%)*

Table 8. Percentage of the positive characteristics of the legacy systems.

7.2. Problems of the legacy systems

The similarities have also been found in problems of legacy system. The results from survey are supporting the result from interview.

From the interview session, the most common problem from participants is the *knowledge of legacy system*. Knowledge around legacy system includes lack of documentation and expertise. The knowledge in legacy system is becoming scarce from time to time according to 24 participants in an interview session. It is in line with the result from the survey. From the survey result, lack of documentation is critical problem in legacy systems with 31,58% voted. Lack of experience manpower is even the most critical problem in legacy system according to 27,08% respondents in a survey and another 27,08% in the critical problem.

The same result is achieved for the other problems, such as *hard to maintain*, *incompatibility*, and *inflexibility* of legacy systems. Those problems are perceived critical due to the fact that many participants mentioned them during their interview session (19 participants, 20 participants, and 12 participants respectively). The results are matched with the result from the survey. D*ifficult to maintain or enhance in-house*, *incompatible with technology environment*, and *too rigid with the new requirements* are critical problems according to survey's results with 30,11%, 30,53%, and 30,53% respectively. Surprisingly, all those problems got voted mostly by business-related respondents with more than 50% for each of the problems.

The problems above could be the result from poor *architecture or design of legacy systems* (13 participants). Monolith, hardcoded and messy connections between the systems are typical characteristics of legacy systems. These problems were often found in interview session as well. And this is matched to the result of the survey. Monolith architecture is perceived as a critical problem in survey part with 27,66%.

The other problems are *poor user interface* and *high risk of failure*. *Poor user interface* in interview session is not only about graphical user interface in the legacy system, but also about how end users can learn to use the system appropriately (13 participants). Such problem is also perceived as a critical problem from the survey part with 29,35%. In contrast, a *high risk of failure* was not perceived as critical problems. It is in-line with the result from the survey. Not many participants often mentioned *the risk of the failure* during their interview session. The *risk of failure* can usually be predicted beforehand. Therefore, before it becomes bigger, organizations usually anticipate it by taking extra steps which they think necessary to mitigate the risk of failure. As a result, a *high risk of failure* is only perceived as quite critical problems from respondents in the survey. Among those respondents, 52% are coming from business-oriented respondents. Cleary, they see that if the system goes down, it could affect their business process.

The last problem is *the high cost of maintenance* of the legacy systems. It is critical according to the result from the survey. The high cost of maintenance is resulted from all of the problems accumulated and bad characteristics of the legacy systems. The result from survey is matched with the result from interview session. Many of the participants from the interview session (20 participants) mentioned about the high cost of maintenance of the legacy systems.

Lack of documentation *Business Domain (41%); **Technology domain (44%)**; Academic domain (15%)*	Lack of experienced manpower *Business Domain (39%); **Technology domain (50%)**; Academic domain (13%)*
Limited suppliers/vendors to support and maintain ***Business Domain (46%);** Technology domain (40%); Academic domain (14%)*	Unable to adequately support, maintain, or enhance in-house ***Business Domain (52%)**; Technology domain (37%); Academic domain (11%)*
Incompatible with current and/or future technological environments ***Business Domain (56%)**; Technology domain (39%); Academic domain (5%)*	Too rigid to comply with new business requirements ***Business Domain (56%)**; Technology domain (31%); Academic domain (13%)*
Monolithic architecture ***Business Domain (52%)**; Technology domain (34%); Academic domain (14%)*	High risk of failure ***Business Domain (52%)**; Technology domain (40%); Academic domain (8%)*
Poor user interface ***Business Domain (52%)**; Technology domain (7%); Academic domain (41%)*	Too costly to maintain *Business Domain (43%); **Technology domain (46%)**; Academic domain (11%)*

Table 9. Percentage of the problems caused by the legacy systems.

7.3. Driver for legacy modernization

The drivers for legacy modernization that were found in interview session are comprised and has been confirmed by more widely audience.

In general, the result survey seems is the same with the result from interview part. It could be seen from their responses in the survey that most of them agree with the drivers found in the interview sessions. Almost all of the drivers are strong drives toward legacy modernization. Those are (i) *create business opportunity*, (ii) *cost reduction in maintenance*, (iii) *limited supplier*, (iv) *lack of experts in the legacy system*, and (v) faster time-to-market. Even *to get flexible system* is the strongest driver according to the result in the survey. All of those drivers come from the business-related respondents, such as Marketing, Top Management, Manager, Business Analyst, and Consultants.

However, *prone to failure* is perceived as a weak driver. Although most of the business-oriented people are dominating the survey (53%), this driver is still perceived as a weak driver. It is understandable since high risk of failure in the survey is not as critical as other problems.

From the interview part we see that the driver towards legacy modernization must have business value of it. It is in-line with the results derived from the survey part. All the drivers are dominated by business-related respondents. Only one driver is dominated by technology-oriented people, which is *faster time-to-market*. Even though it was dominated by technology-oriented people, the difference is not significant (48% for technology and 46% of business). The list of all the percentage is summarized below.

Become flexible to support changing business requirements ***Business Domain (58%)**; Technology domain (40%); Academic domain (2%)*	Create new business opportunities via mergers & acquisitions ***Business Domain (57%)**; Technology domain (30%); Academic domain (13%)*
Reduce the cost of maintenance & operations ***Business Domain (49%)**; Technology domain (45%); Academic domain (6%)*	Limited suppliers/vendors to support legacy systems ***Business Domain (46%)**; Technology domain (40%); Academic domain (14%)*
Lack of experts around legacy systems ***Business Domain (56%)**; Technology domain (39%); Academic domain (5%)*	Prone to failures ***Business Domain (53%)**; Technology domain (38%); Academic domain (9%)*
Faster time-to-market of product *Business Domain (46%); **Technology domain (48%)**; Academic domain (6%)*	

Table 10. Percentage of drivers toward legacy modernization.

7.4. Challenges in legacy modernization

In regards to challenge in legacy modernization, the results from interview part and from the survey part are supporting each other. Most of the respondents in survey part agree with the challenges that were found in interview part. As predicted, the challenges are mostly coming from the business side and organizational side such as, *hard to extract business rules*, *resistance from current users*, *funding*, and *ROI prediction*. Those challenges are perceived as a “very challenging” in survey part. From the technical side, *only poor system architecture* is perceived as a “very challenging” in the process of modernization. However poor system architecture was voted mostly by business people with (54%).

The other challenges (*time, communication, functions prioritization, cultural organization, resources* and *data migration*) are voted as the challenging. Only *testing* is seen as a quite challenging. It is the same with the result in interview sessions (14 participants, 8 participants, 14 participants, 10 participants, 24 participants, and 12 participants) and only 5 participants mentioned about the difficulty in the testing phase. It is due to the fact that the most of our participants in interview session are business-oriented participants, while testing is perceived as a technical challenge. It is also supported by the fact that 56% people from technology-related respondents voted for difficult to test in survey part.

From the interview session, the participants agree that from a technical point of view, data migration is the biggest challenge in legacy modernization. It is in line with the percentage from technology domain who votes for data migration (56%). Compare to the votes from the business domain in which only 36%.

Data Migration *Business Domain (36%); **Technology domain (56%)**; Academic domain (8%)*	Lack of resources (e.g. documentation, experts) ***Business Domain (55%)**; Technology domain (42%); Academic domain (3%)*
Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti) ***Business Domain (54%)**; Technology domain (38%); Academic domain (8%)*	Difficult to extract business rules/knowledge ***Business Domain (49%)**; Technology domain (45%); Academic domain (6%)*
Difficult to test *Business Domain (38%); **Technology domain (56%)**; Academic domain (6%)*	Resistance from the current users/maintainers in the organization ***Business Domain (59%)**; Technology domain (33%); Academic domain (8%)*
Cultural resistance in organization not to adapt new system ***Business Domain (47%)**; Technology domain (42%); Academic domain (11%)*	Difficult to effectively prioritize the functionality for modernization ***Business Domain (58%)**; Technology domain (39%); Academic domain (3%)*
Difficult to communicate the reasons/consequences of modernization ***Business Domain (52%)**; Technology domain (38%); Academic domain (10%)*	Funding legacy modernization projects ***Business Domain (62%)**; Technology domain (26%); Academic domain (12%)*
Time constraints to finish legacy modernization ***Business Domain (61%)**; Technology domain (34%); Academic domain (5%)*	Predicting Return of Investment (ROI) of modernization ***Business Domain (53%)**; Technology domain (36%); Academic domain (11%)*

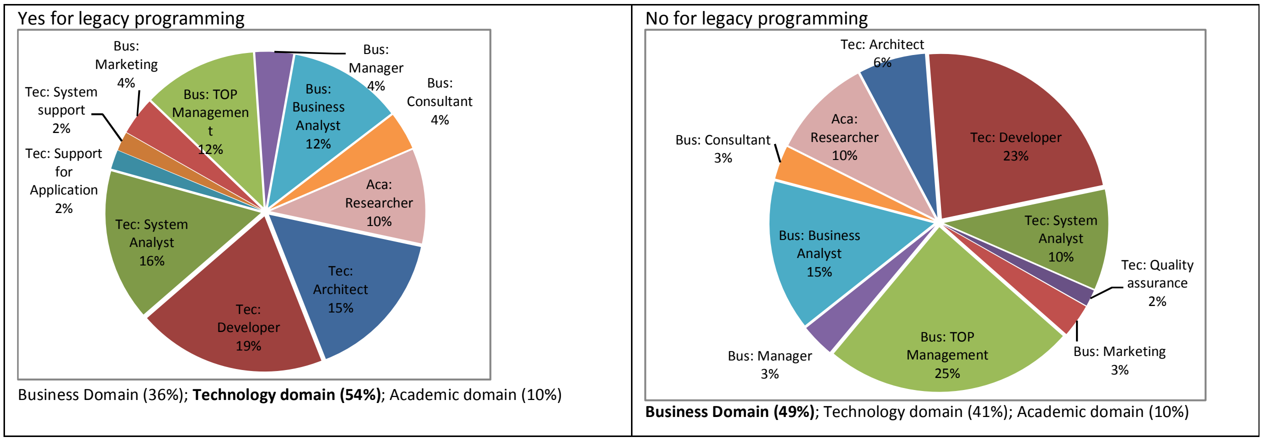
Table 11. Percentage of challenges in legacy modernization.

7.5. Programming language

There is a little difference when the author tried to compare the result of an interview and survey in regard to programming language in legacy system. The result from interview reveals that most participants do not consider programming language as a factor to determine legacy system. Contrary, the result from survey shows that more respondents consider programming language as a one factor to determine legacy system. However, for the respondents who voted that programming language determines the systems for being legacy are mostly coming from the technical domain (54%). Compare with respondents who come from the business domain (36%). Moreover, people from the business domain tend to see programming language has nothing to do with legacy system (49%), compare with people from technical domain (41%).

Most of the participants in the interview session mostly have a business background. Some of them also are in the highest position within the organization. Therefore, most of their opinions have business value in it and less technical value.

Looking at how people in academic see this issue, from 6 people who come from academic arena, 5 of them see the programming language determines the systems for being legacy. Only one respondent disagree with this argument. Thus, we can conclude that people from academic and technical practitioners tend to see that programming language determines a system for being legacy.



7.6. Findings

The main findings in this study are:

1. There is a main discrepancy between academia and industry in regards to legacy system. It has mainly resulted from how those two domains perceived legacy system. Academics tend to see legacy systems from a technical viewpoint, thus little business value is involved in academic product (e.g. Journal, paper, application tools, etc.). In contrast, practitioners in the industry are more looking into all aspects such as business value, organizational, and technology in the legacy systems.
2. There is no valid and general definition about the legacy system. Such term is too broad definition that makes it difficult to formulate a strong and solid definition. It involve anything ranges from technology (hardware, software, ecosystems, programming language, etc.), business (money, rules, regulation, risks, opportunity, etc.), and organization (people, culture, time, etc.).
3. Legacy system is not merely technological problem, but also the problems involving business and organization. In fact, business aspects are more dominating in every angle of legacy system.
4. The organization will do legacy modernization only if there is enough business force toward it. All problems in technical are not motivating organizations to legacy modernization unless there is a problem that can impact their business processes or business continuity. However, the main drivers are cost reduction, flexibility in adapting new requirements, and scarcity of knowledge.
5. The main problem in the legacy modernization is the knowledge that becoming scarce as time goes on. It is also the main characteristic that nobody should be argued about. Problem regarding knowledge around the legacy systems is so crucial that lead to many problems as domino effect such as, difficult to maintain, unsupported supplier, poor IT landscape, and high cost to maintain.
6. The main challenges in legacy modernization are coming from business and organizational aspect, such as funding and resistance from current users within the organization. However, the biggest challenge from technological point of view is data migration.

7.7. Revisiting Research Questions

Sub Question 1 (SQ1):
How do academia perceive legacy systems?

The research reveals that academic tends to define the legacy system from technical point of view. Business aspect is seen as an implication of the bad technology. On the other words, technology comes first to determine a system for being legacy and business aspect as a result of the bad technical aspect in the system. By this way of thinking, it makes the products from academic arena (e.g. journal, paper, etc.) have a lot of stress in technical side of the system, and only little business aspect in it.

Sub Question 2 (SQ2):
How do practitioners perceive legacy systems?

On the other side, professionals in industry always put business value in their definition of the legacy system. They only define legacy system only if their business gets interrupted by the existence of the current system. On the other words, business aspect always come first and determines the system for being legacy. Therefore, as long as their systems satisfy them, there will not be legacy system no matter the system is old and not good anymore from the technology point of view.

Sub Question 3 (SQ3):
What are the drivers for legacy modernization?

After conducting interview session and survey, the list of drivers toward legacy systems are:

1. Become flexible to support changing business requirements
2. Create new business opportunities via mergers & acquisitions
3. Reduce the cost of maintenance & operations

4. Limited suppliers/vendors to support legacy systems
5. Lack of experts around legacy systems
6. Prone to failures
7. Faster time-to-market of product

Sub Question 4 (SQ4):
What challenges are still identified in legacy modernization by practitioners?

The challenges that were identified in this research are listed below:

1. Data Migration
2. Lack of resources (e.g. documentation, experts)
3. Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)
4. Difficult to extract business rules/knowledge
5. Difficult to test
6. Resistance from the current users/maintainers in the organization
7. Cultural resistance in organization not to adapt new system
8. Difficult to effectively prioritize the functionality for modernization
9. Difficult to communicate the reasons/consequences of modernization
10. Funding legacy modernization projects
11. Time constraints to finish legacy modernization
12. Predicting Return of Investment (ROI) of modernization

Main Research Question (MRQ):
What is the gap in the perception of legacy systems between experts from industry and from academia?

There is a gap in how academics perceive legacy system and practitioners in industry perceive legacy system. The technology aspect is dominating in how academics perceive the legacy system, such as *type of system*, *programming language*, and *architecture of the system*. Even though there is business aspect in it, but it is not enough to attract professionals in industry when they have problems with their legacy systems. Practitioners in industry are more attract to the coexistence of the legacy systems with their business and also the organizational condition within their organization.

It has the following implications, such as no academics' product being widely used by practitioners in industry. The academic result is too technical for organizations and no further information on how the result can be applied in real case. Therefore, the methods that have been developed in academia are rarely adopted in the real business case. Moreover, the results from academic research do lack of further guidance. In addition, there is a mind of set that academics only work in upcoming technology. It makes organizations reluctant to find the solution of their old the legacy system.

7.8. Validity of Results

Typically, validity is not directly applicable to *GT* as they are origin for quantitative study. *GT* by its nature do not hold that there is one unified truth in the world, because it is assumed to be rather subjective and largely based on perceptions. Therefore, *GT* is more focus on trustworthiness which is based on credibility, transparency, dependability, and conformability. To establish credibility (internal validity in qualitative research), the author uses the following some measures:

Triangulation (multiple methods, multiple data sources, multiple investigators, or multiple theories). To achieve triangulation, the author conducted online survey which had attracted 104 respondents. The survey was constructed based on the finding from interview session. Therefore, survey can also be seen as one of the validation tool because it helps to verify the main concept and codes to the broader audience.

Although there was a possibility of bias, the author believes the research is an open-minded study which led to findings and we did not expect.

Prolonged communication with expert. Active communication with experts in legacy systems is the best way to get better understanding with the phenomenon. In this research, the author use his supervisor in which also a member of Servicifi project (http://servicifi.org/).

Saturation (adequate amount of data has been examined to support that no additional data would yield different findings). Saturation point is achieved when there is no valuable data anymore. To achieve saturation point, enough interviews are required. In total, this research includes 23 interviews with 26 participants to collect the data from professional in industry. The author believes that the data gathered are sufficient enough to derive the conclusion, since the 21st interview had less new information regarding the topic of interest.

Reflexivity (the use of field journals to capture ideas, connections, methodological notes, etc. related to the understanding of the phenomenon). The author also used journals and papers as his references to improve his understanding in regards to the phenomenon. Journals, papers, whitepapers, presentations, etc. had being used before the author concluded his finding.

To make credibility stronger and mitigate all the risk of mismatch finding with the data, it is suggested to do respondents validation or members checking. However, due to the time restriction, this validation could not be performed until this report is constructed. Such strategy will, hopefully, be performed as the project of Servicifi (http://servicifi.org/) comes along. In addition, to improve transparency, the author also provides 17-page technical report on coding process. In conclusion, this research has been followed every steps in GT carefully and thoroughly.

External validity refers to the degree to which the results of an empirical investigation can be generalized across individuals, settings, and times. It contains *Population validity* and *Ecological validity*. The results in *GT* are usually ecologically valid since they are close to the data that generate the result. They are context-specific, detailed, and tightly connected to the data. Population validity sees how the sample represents the population. The participants from interview session are people from high top managerial. Most of them have built their career for at least 15 years in IT. Therefore, they are more than eligible to discuss about legacy system. Same in survey part, the online survey was spread in the community that discusses more about legacy system and legacy modernization. By doing that, the author hopes that the threat of reliability in survey can be mitigated.

In regards to the external validations, the main idea of this research is seen the gap between academia and professional in industry. Therefore, there is a threat that if the same questions are repeated to the industrial people after the findings is published, they perception might be influenced by academic perception in this report. The other threat would be the different understanding about legacy modernization. Based on the author's experience, some of the people in industry see that change in small parts or enhance the functionality in the system can be seen as a legacy modernization.

8. Conclusion

8.1. General discrepancy between academic and industry

In academic field, legacy systems are often seen as large and complex system which that modification and evolution. Usually, they are backbone system and play as a business critical in the company. Meanwhile, Sneed, (2006) more stresses into outdated technology as a characteristic of legacy system. Stehle (2008) adds some characteristics such as old system with lack of documentation and expert which cause difficulty in maintenance and difficult to understand. Survey report from NASCIO, (2008) defines legacy system not only by age, but also by supportability, risk, agility, availability of software and hardware support and staffing. Moreover, they are also characterized by inability to adequately support business requirement. Bisbal et al. (1999) also put problems such as high risk of failure, obsolete hardware, slow, bad GUI, difficult to integrate and expensive to maintain.

Most of the characteristics and problems of the legacy system from the academia are in-line with the result from industry. On top of all, *scarcity knowledge* is the main characteristic and main problem that is haunting all the organizations. The next problem is *maintenance wise* which is including *difficult to maintain* and *high cost of maintenance*. Next characteristics are *difficult to interconnect* with other systems and *bad adaptability* with modern ecosystems. Both academic and professional agree that legacy systems have these characteristic. *Limited suppliers* to support the systems (hardware or software) and *poor architecture design / IT landscape* are the other bad characteristics that both academia and professional agree upon. In addition, the common characteristics of legacy system are *not flexible* and *end of product lifecycle*.

In regard to the age of the system, both academia and practitioners have no doubt that old system is a one of the characteristics of the legacy system. The reason is because the old systems usually are equipped with obsolete technology that brings many difficulties to organization. However, in academia, the old system is immediately considered as a legacy system due to their technology limitations. It is not what practitioners see about the legacy systems. Even if a system is old and has limitation, but if it is still be able to process their business properly, then the system is not considered as a legacy system.

From business point of view, legacy system is a system which is not a part of their business roadmap (strategy). As soon as the systems hamper the organizations to operate or expand their business, then the systems will be considered as the legacy system. Such mind set stresses business aspect as factor to determine the systems for being legacy. It is different from how academies see the legacy systems. People in academic tend to see the legacy systems from only a technical point of view. Hence, they often see the negative impression of the legacy systems. However, legacy system is not always about bad characteristics. There are some aspects of the legacy systems that are positive too. For instance, legacy systems bring great revenue for organization.

8.2. Programming Language

One different perspective about the legacy systems from academia and practitioners in industry is their view about the role of programming language. From the interview with practitioners in industry, especially for business-oriented people, most of them see programming language has not play too much to determine system for being legacy. They tend to use any language for which it is suitable in their situation, despite the languages are old or not. They do not really care about which languages are used, as long as their business processes can be executed properly by the legacy systems. For example COBOL is an old language, but it is not the reason for the systems for being legacy. If COBOL still brings revenue for company, then they tend to stick with it.

The result from interview is also in-line with the result from survey. Business-oriented respondents, in general, are not considering programming language as a factor that could determine system being legacy.

They will strictly use any programming language that can the best suited to develop the system, no matter how old and how difficult to maintain it as long as their profit can still be increased.

Another reason why organizations do not consider programming language as a factor to determine legacy system is because some organizations only act as an end user of the application. They do not build the applications neither considers about the technical part of the application. They do not really care about which language is being used for the applications or systems. The systems which they use are usually offered by other parties, thus all maintenance and modification will be done by the other parties. They only see the outcome of the application whether or not it can benefit them. All they know is the systems should work as they expect. Therefore, the focus is on the application or software as a whole, not in deep into the language of the system.

However, in term of continuity, all organizations agree that COBOL is in dangerous situation now. Continuity seems to be a future threat for all organizations that still have COBOL in their system. They do realize that COBOL experts become less and less in the future, so it will be a real problem in the future, although it would not happen sooner.

To conclude, people in academic tend to see the legacy systems from technical point of view. That is why they have bad impression of the systems, such as problems and difficulties caused by legacy systems. However, professional in industry see legacy system more into the business value of the system. Consequently, as long as the systems give them profit – no matter how bad the systems are – they will continue operating their legacy systems. Therefore, problems from technical side of legacy system are not really a problem from professional in industry unless the problems disturb their business process. They only consider the legacy systems if the problems hamper their business continuity or disturb their business model.

8.3. Success and failure factors in legacy modernization

The main success factors of doing legacy modernization are *a good business goal* and *a good vision*. To achieve that, *strong governance model* in legacy modernization is required. It comes together with *proper budget calculation* and *time consideration* to finish the process of modernization. Therefore, be strong on you initial requirement (requirements before and requirements after the modernization) is also a crucial factor. It means that, changing the requirements during the process of legacy modernization is not suggested in order to finish the project on time. However, it is suggested to also be a little bit flexible to cope with the small changes. The fact that technology and business requirements are constantly changing, might lead organizations end up with obsolete technology that no longer meets its business needs after a long process of legacy modernization.

With regard to this challenge, the project manager should keep balance on initial goal and new requirement that might come during the process of modernization. Many major changes during the process of migration can lead to over time, over budget, and fail to reach the goal of legacy modernization that eventually make the process of modernization fail completely. On the other hand, he has to maintain current functionality as well.

In the legacy modernization plan, organization also has to think about how the *communication* should be handled between all people involved (directly and indirectly) in legacy modernization. Good vision in why the organization must do modernization is needed or crucial since the process need to be accepted by all stakeholders in the organization. Organization also has to anticipation about *cultural adaptability* in the organization as a whole itself. All of those should be planned carefully. The stronger the governance plan is, the bigger the chance for the project to be success.

To start with the process of modernization, *proper investigation* should be executed first. Extract the knowledge from old system is the key factor during investigation. To get all the necessary knowledge completely, the availability of knowledge (experts and documents) are crucial. Keep in mind that the experts are not merely from IT domain, but also from various domain such as business and human resource domain.

Upon completing the investigation, *vendor or supplier selection* is also vital factor in the process of migration because promising vendor can make the lifecycle of the system longer and mitigate the risk of lacking the experts. Keep in mind that chose the wrong vendor can make the project over time, over budget and make the project failed.

Choosing the *right people* to do the right task or job is the key factor from human point of view. It is including the good project manager and also support from people on board. Project manager can be crucial since he will take care of ICT management, so the process will be on schedule and right decision can be made. It became more difficult task since that should be in line with business value as well. On the other hand, support from people in board is needed to persuade and "order" every person to involve and support the process of migration.

Most organizations do not prefer to execute legacy modernization once at the time (Big bag approach) because in reality, the risk of failure is usually too great. The best way to do legacy modernization is by gradually executing it or step by step. It is more preferable because if they made a mistake, then they can go back to their previous stage without getting big harm. However, method used in legacy modernization is mostly ad hoc or based on experience of the organization. Methods available in academic are often ignored by organization because they argue that they do not see the benefit by doing that. They tend to rely on the experience of someone who have done legacy modernization before successfully.

8.4. General Similarity between academic and industry

There are some similarities regarding legacy modernization between academia and industry people. First, both of them agree that short term solution that leads to maintain obsolete legacy should be avoided, for instance implementation of screen scraping. Change in a GUI of legacy system is still much a short-term solution and does not address many of the serious problems such systems face, including overloading, static functionality, and high maintenance costs. Therefore, legacy modernization is time consuming and takes a lot of effort. So usually it takes many years.

Secondly, in legacy modernization data and program codes are the two keys components. It can be seen that during the interview sessions, data migration and extracting knowledge from legacy system are considered as the successful and failure factor of executing legacy modernization.

Thirdly, change in program codes should be done automatically to avoid human error. People in industry also agree and expect to do as many automation as possible with also double check from expert. However, automation in legacy modernization is still immature. There are always manual works involved in legacy modernization and at the end; manual works always take a large portion in legacy modernization.

Last but not the least, data migration is crucial factor in legacy modernization that needs particular attention. It is mainly because moving the data from the old environment to the new environment holds some high risks, such as missing data or incompatible data structure. For some modernization projects, data migration can only be done once without failure. Thus, prudent approach should be done in migrate the data.

8.5. Suggestion

The fact that professionals in industry rarely use academic resources as their information source indicates the weak relationship between them. There are some causes of the bad relationship between academia and professional in industry captured. Those are:

- Most professional in industry believe that university only equipped their students with modern technology but not with the knowledge about coexistence of the legacy system. Lack of topic in regard to legacy systems or legacy modernization make organizations reluctant to go and find the solution of their problems in academic field. Therefore, it is suggested to place the knowledge about legacy system and legacy modernization in curriculum. However, the knowledge meant is not about how to develop the programming skill with the old language, but more into the understanding of the importance of legacy systems in organization.

- One difficulty in the process of legacy modernization is to extract the knowledge from legacy system. So, improving insight into internal procedure will help organization extracts knowledge from their legacy systems and present it in human readable format. Not in technical way, but understand the content on such as applications. Therefore, it would be helpful for organization if the students are capable to extract the business rules and knowledge out of legacy systems.
- All researches produced in academic are too much focused on technology, and do not talk much about business. However, organizations are more interested in business side of legacy systems or business impact in legacy modernization. Therefore, organizations do not see the benefit by looking to academic resources as their references. If there are a lot of researches in legacy modernization that more involve in business side, then organizations would not mind to use academic works as their references.
- Works that have been done in academia might be related with the problem faced by organizations. However, lacks of guidance makes organizations, find it difficult to apply academic's works in their case. Thus, providing guidance for organizations would be helpful.
- More collaboration between academic and organization in industry through internship program can make the relationship between them stronger. However, it is a little bit difficult to combine student's purpose with what organization wants from student. Thus, continually communication can help organization and academic find a win-win solution for the problem.
- Organizations always have time constrain in every project they have. So, the best solution for them is ready-to-use service. In many cases, when academics are involving in always legacy modernization project, they do research first before provide the solution. However, involving someone from outside organization require organizations to explain everything again. Therefore, ready-to-use service is what the organizations expect from academic.
- Next researches are expected to be more focusing on how to automate some processes in legacy modernization. At the moment now, there are still large portions of manual labor compare with the automation in legacy modernization.

Further research could more focus on non-technical aspect of legacy system, and how it could be incorporated or fit into industry need. It is also help the organizations if research in academic can produce framework/model/check list in which can identify system for being legacy.

References

Adolph, S., Hall, W., & Kruchten, P. (2011). *Using grounded Theory to study the experience of software development. Empirical Software Engineering*, *16*(4), 487-513.

Al-Qutaish, R. E., & Al-Sarayreh, K. (2008). Software Process and Product ISO Standards: A Comprehensive Survey. *European Journal of Scientific Research, 19*(2), 289-303.

Almonaies, A. A., Cordy, J. R., & Dean, T. R. (2010, March). Legacy system evolution towards service-oriented architecture. In *International Workshop on SOA Migration and Evolution* (pp. 53-62).

Aversano, L., & Tortorella, M. (2004). An assessment strategy for identifying legacy system evolution requirements in eBusiness context. *Journal of Software Maintenance and Evolution: Research and Practice, 16*(4-5), 255-276.

Bisbal, J., Lawless, D., Bing, W., & Grimson, J. (1999). *Legacy information systems: issues and directions*. IEEE Software, *16*(5), 103-111.

Bennett, K. (1995). Legacy systems: coping with stress. *Software, IEEE*, *12*(1), 19-23.

Brodie, M.L., & Stonebraker, M. (1995). *Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach*. Michigan: Morgan Kaufmann Publishers

Brooks Jr, F. P. (1987). No silver bullet-essence and accidents of software engineering. *IEEE computer*, *20*(4), 10-19.

CIOinsight. (2002). Legacy Systems 2002: Are Your Older Systems Slowing You Down? - See more at: http://www.cioinsight.com/c/a/Research/Legacy-Systems-2002-Are-Your-Older-Systems-Slowing-You-Down/#sthash.CNG43v4A.dpuf. *CIOinsight*. Retrieved January, 16, 2013, from http://www.cioinsight.com/c/a/Research/Legacy-Systems-2002-Are-Your-Older-Systems-Slowing-You-Down/.

Colosimo, M., De Lucia, A., Francese, R., Scanniello, G., & Tortora, G. (2007). Introducing Legacy System Migration Technologies in a Software Company: a Controlled Experiment.

Denzin, N. K., & Lincoln, Y. S. (2003). *Strategies of Qualitative Inquiry*. London: Sage Publication Inc.

Erlikh, L. (2000). Leveraging legacy system dollars for e-business. *IT professional, 2*(3), 17-23.

Galin, D. (2004). *Software quality assurance: from theory to implementation*. Pearson education.

Gartner, Inc. (2007, December 20). The quest for talent: You ain't seen nothing yet. Gartner Research. Retrieved June, 7, 2013, froim http://www.gartner.com/DisplayDocument?id=569115.

TIOBE, (2013). TIOBE Programming Community Index for January 2013. *TIOBE Software: Tiobe Index.* Retrieved January, 16, 2013, from http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

Geetha, S. (2012). Possible Challenges of Developing Migration Projects. *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY, 3*(3), 463-465.

Glaser, B., & Strauss, A. (1967). *The discovery of Grounded Theory: Strategies for Qualitative Research*. New Jersey: Aldine Transaction.

Hatch, J. A. (2002). *Doing qualitative research in education settings*. SUNY Press.

Heskett, J. (2011). *The culture cycle: How to shape the unseen force that transforms performance*. FT Press.

Juric, M. B., Rozman, I., Hericko, M., & Domajnko, T. (2000). Integrating legacy systems in distributed object architecture. *ACM SIGSOFT Software Engineering Notes, 25*(2), 35-39.

Khadka, R., Saeidi, A., Idu, A., Hage, J., & Jansen, S. (2013). Legacy to SOA Evolution: A Systematic Literature Review. In *In AD Ionita, M. Litoiu, & G. Lewis (Eds.) Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*.

Khadka, R., Reijnder, G., Saeidi, A., Jansen, S., & Hage, J., (2012). A Method Engineering Based Legacy to SOA Migration Method. Proceedings of the 27th IEEE ICSM, Williamsburg, VA, USA. IEEE.

Khadka, R., Saeidi, A., Hage, J., & Jansen, S. (2013a). A Structured Legacy to SOA Migration Process and its Evaluation in Practice. Proceeding of the 7th MESOCA, Eindhoven, The Netherlands, IEEE.

Khadka, R., Saeidi, A., Jansen, S., Hage, J., & Haas, G. P.(2013b*).* Migrating a Large Scale Legacy Application to SOA: Challenges and Lessons Learned. Proceedings of the 20th WCRE, Koblenz, Germany. IEEE.

Khadka, R., Sapkota, B., Pires, L. F., van Sinderen, M., & Jansen, S. (2011). Model-driven development of service compositions for enterprise interoperability. In *Enterprise Interoperability* (pp. 177-190). Springer Berlin Heidelberg.

Kizior, R. J., Carr, D., Halpern, P. (2000). Does COBOL Have a Future?. *Proceedings of the Information Systems Education Conference,* 126, Philadelphia.

Lin, C. Y. (2008). *Migrating to Relational Systems: Problems, Methods, and Strategies*. Contemporary Management Research, *4*(4), 369-380.

Maurizio, A., Sager, J., Corbitt, G., & Girolami, L. (2008, January). Service oriented architecture: challenges for business and academia. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* (pp. 315-315). IEEE

Meng, F., Qu, Z., & Guo, X. (2013). *Refactoring Model of Legacy Software in Smart Grid based on Cloned Codes Detection*. IJCSI International Journal of Computer Science Issues, *10*(1), 296-303.

Morello, D., Kyte, A., & Gomolski, B. (2007). The Quest for Talent: You Ain't Seen Nothing Yet (No. G00153872). *Gartner Inc., Stamford, CT.*

National Association of State Chief Information Officers-NASCIO. (2008). *Digital States at Risk!: Modernizing Legacy Systems*. Lexington: NASCIO publisher.

Newcomb, P., & Doblar, R. (2001). Automated transformation of legacy systems. *CrossTalk, 14*(12), 18-22.

Nowakowski, W., Śmiałek, M., Ambroziewicz, A., Jarzębowski, N., & Straszak, T. (2012). *Recovery and migration of application logic from legacy systems*. Computer Science, 13(4), 53-70.

Papazoglou, M., & Van Den Heuvel, W. J. (2006). Service-oriented design and development methodology. *International Journal of Web Engineering and Technology, 2*(4), 412-442.

Paradauskas, B., & Laurikaitis, A. (2006). *Business knowledge extraction from legacy information systems*, *35*(3), 214-221.

Rahgozar, M., & Oroumchian, F. (2002). A practical approach for modernization of legacy systems. In *EuroAsian Conference on Advances in Information and Communication Technology (ICT 2002)-Workshop on Recent progress in Computers and Comunications*.

Rahgozar, M., & Oroumchian, F. (2002, October). A Transformational Approach for Legacy Systems' evolution. In *Submitted for publication in: 2002 WSEAS International Conference on Applied Mathematics and Computer Science (AMCOS'02), Copacabana, Rio De Janeiro*.

Richards, L., & Morse, J.M. (2007). *Users guide for qualitative methods (2nd Ed.)*. Thousand Oaks, California: Sage Publications Inc.

Robson, C. (2002). *Real world research: A resource for social scientists and practitioner-researchers* (Vol. 2). Oxford: Blackwell.

Runeson, P., & Höst, M. (2008). *Guidelines for conducting and reporting case study research in software engineering*. Empirical Software Engineering, *14*(2), 131-164.

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, *25*(4), 557-572.

Seacord, R. C., Comella-Dorda, S., Lewis, G., Place, P., & Plakosh, D. (2001). *Legacy System Modernization Strategies* (No. CMU/SEI-2001-TR-025). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Sneed, H. M. (2000). Encapsulation of legacy software: A technique for reusing legacy software components. *Annals of Software Engineering*, *9*(1-2), 293-313.

Sneed, H.M. (2006). *Wrapping Legacy Software for Reuse in a SOA*. Multikonferenz Wirtschaftsinformatik, *2*, 345-360.

Stehle, E., Piles, B., Max-Sohmer, J., & Lynch, K. (2008). Migration of Legacy Software to Service Oriented Architecture. *Department of Computer Science Drexel University Philadelphia, PA*, *19104*, 2-5.

Strauss, A. L. (1987). *Qualitative analysis for social scientists*. Cambridge University Press.

Strauss, A., & Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Thousand Oaks, California : Sage Publications Inc.

TIOBE, (2013). TIOBE Programming Community Index for January 2013. *TIOBE Software: Tiobe Index.* Retrieved January, 16, 2013, from http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Tomassetti, F., Torchiano, M., Tiso, A., Ricca, F., & Reggio, G. (2012, May). Maturity of software modelling and model driven engineering: a survey in the italian industry. In *Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on* (pp. 91-100). IET.

Tonella, P., Torchiano, M., Du Bois, B., & Systä, T. (2007). Empirical studies in reverse engineering: state of the art and future trends. *Empirical Software Engineering, 12*(5), 551-571.

Torchiano, M., Di Penta, M., Ricca, F., De Lucia, A., & Lanubile, F. (2011). Migration of information systems in the italian industry: A state of the practice survey. *Information and Software Technology*, *53*(1), 71-86.

Ulrich, W. (2004, September). A status on OMG architecture-driven modernization task force. In *Proceedings EDOC Workshop on Model-Driven Evolution of Legacy Systems (MELS). IEEE Computer Society Digital Library.*

van Deursen, A., Klint, P., & Verhoef, C. (1999). *Research issues in the renovation of legacy systems* (pp. 1-21). Springer Berlin Heidelberg.

van Geet, J. (2011). *Reverse Engineering for Mainframe Enterprise Applications - Patterns and Experiences*.

van Geet, J., Ebraert, P., & Demeyer, D. (2010). *Redocumentation of a Legacy Banking System*. Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), New York, USA, 33-41. doi: 10.1145/1862372.1862382.

Venkatraghavan N. Iyer, (2008). Legacy Modernization Modernize and Scale. Retrieved from Infosys database.

Verschuren, P., Doorewaard, H., & Mellion, M. J. (2010). *Designing a research project*. Eleven International Publishing House.

Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wade, V., O'Sullivan, D., & Richardson, R. (1997). *Legacy Systems Migration: A Method and its Tool-Kit Framework*. Proceedings of the APSEC'97/ICSC'97: Joint 1997 Asia Pacific Software Engineering Conference and International Computer Science Conference, Hong Kong, China, 312-320. doi:10.1109/APSEC.1997.640188

Weiderman, N. H., Bergey, J. K., Smith, D. B., & Tilley, S. R. (1997). *Approaches to Legacy System Evolution* (No. CMU/SEI-97-TR-014). Carnegie-mellon univ pittsburgh pa software engineering inst.

Xia, W., & Lee, G. (2004). Grasping the complexity of IS development projects. *Communications of the ACM*, *47*(5), 68-74.

Yin, R.K. (2009). Case Study Research: *Design and Methods-Fourth Edition*. Washington: Sage Inc.

Zhang, W., Berre, A. J., Roman, D., & Huru, H. A. (2009, October). Migrating legacy applications to the service Cloud. In *14th Conference companion on Object Oriented Programming Systems Languages and Applications (OOPSLA 2009)* (pp. 59-68).

Zhang, Z., Liu, R., & Yang, H. (2005, July). Service Identification and Packaging in Service Oriented Reengineering. In *SEKE* (Vol. 5, pp. 620-625).

Zhang, Z., Yang, H., Zhou, D., & Zhong, S. (2010). *A SOA Based Approach to User-Oriented System Migration*. Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, Washington, USA, 1486-1491.

Appendixes

Code Collection

Category 1. Technical Aspect

Characteristic of the legacy systems from a technical side.

Concept 1.1. Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

[1.1.1]	Modifiability	Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.

Concept 1.2. Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

[1.2.1]	Robust	Ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc.
[1.2.2]	Stable system	Software is stable only if it does not have problems that cause it to stop working. A single instance of loss of availability or a system failure would indicate that the software is not stable.

Concept 1.3. Usability

Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

[1.3.1]	User Interface Aesthetics	Degree to which a user interface enables pleasing and satisfying interaction for the user.
[1.3.2]	Learnability	Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

Concept 1.4. Performance Efficiency

Performance relative to the amount of resources used under stated condition. Resources can include other software products, the software and hardware configuration of the system, and materials (e.g. print paper, storage media).

[1.4.1]	Time Behavior	Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements
[1.4.2]	General Performance is good	Performance can be defined as the capability of a system to process a given amount of tasks in a determined time interval.

Concept 1.5. Functional Suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

[1.5.1]	Functional Completeness	Degree to which the set of functions covers all the specified tasks and user objectives.
[1.5.2]	Functional Appropriateness	Degree to which the functions facilitate the accomplishment of specified tasks and objectives.

Concept 1.6. Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

[1.6.1]	Interoperability	Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.
[1.6.2]	Adaptability	Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.

Concept 1.7. Supportability

Support from supplier / vendor / provider for software or hardware in legacy system.

[1.7.1]	Unsupported Supplier	Suppliers / vendors don't support their product anymore. For example, old versions of hardware / patches / updates are no longer available.

Concept 1.8. Flexibility

Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.

[1.8.1]	Not Flexible	Legacy systems are rigid and not flexible due to the fact that they are difficult and take longer time to implement new technology.

Concept 1.9. Proven Technology

Systems that have been demonstrated or verified without doubt to comply with business requirements for long period of time.

[1.9.1]	Old system	Systems which are older than 10 year and was built with an outdated technology.
[1.9.2]	Availability	Degree to which a system, product or component is operational and accessible when required for use.

Concept 1.10. Risk

Risks identified in legacy system.

[1.10.1]	Risk of running legacy system	Risks are increasing if company is still running legacy system, such as risk of continuity

Concept 1.11. Architecture of the systems

The structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them.

[1.11.1]	Large multiple systems	More than one system does the same functionality in which resulting large systems. Big in size of the system including enormous line of codes and connectivity among applications. It also measure the volume of the system
[1.11.2]	Database issue	Problem with database in legacy system. Including Unstructured and inappropriate use of data in legacy system.
[1.11.3]	Back Office	System that does not interact directly with the customer. It is used to administer operations that are not related to any direct sales effort (such as a salesperson with a customer present) and interfaces that are not seen by consumers.
[1.11.4]	Complicated system	System in which contain of a lot and complex rules. Complexity determine structural characteristics by examine how object are interrelated.
[1.11.5]	Poor design and ecosystem	Legacy system is characterized by damaged structure of application and bad ecosystem

Concept 1.12. Knowledge around the systems

The availability of knowledge around legacy system, such as documentation, programmer / developer, analyst.

[1.12.1]	Knowledge become scarce	Condition which not sufficient knowledge available around the system including lack of documentation and expert (programmer, analyst, etc.) in legacy system.

Category 2. Business Aspect

Characteristic of the legacy systems from a business side.

Concept 2.1. Business Strategy

Legacy system in regards to business strategy of organization.

[2.1.1]	Doesn't fit with your future strategy	Inability of legacy system to support business functionality. It's not a part of your end strategy. Getting more difficult to in-line with business strategy.
[2.1.2]	Business Critical	Systems whose disruption or malfunctioning will cause a failure in business operations. The business critical systems contain vital features/functionalities and their failure can result on serious financial & legal problems, damages and other penalties. System that bring benefit to organization

Concept 2.2. Finance

Legacy system from financial point of view.

[2.2.1]	Expensive in maintenance	The cost for maintenance and operating the legacy systems is perceived high.

Category 3. Drivers

Factors that provide impulse or motivation to do legacy modernization.

Concept 3.1. Technical drivers

Factors that motivate the organizations to do legacy modernization from technical / technology point of view.

[3.1.1]	Become flexible to support changing business requirements	Systems are not flexible enough to be modified or changed when new requirements are coming.
[3.1.2]	Limited supplier or vendor to	Lack of original supplier or vendor to support legacy systems.

	support the continuity	
[3.1.3]	Lack experts around the systems	Difficult to find people who are experts or willing to learn the legacy systems
[3.1.4]	Prone to failure	Systems are vulnerable to be failed and organizations cannot afford if the systems failed

Concept 3.2. Business drivers

Factors that motivate organization to do legacy modernization from business point of view.

[3.2.1]	Create new business opportunity via mergers and acquisitions	Legacy modernization allows organization spread their business more widely.
[3.2.2]	Reduce cost of maintenance and operation	The new systems produced by doing legacy modernization are expected to have low cost of operation and maintenance.
[3.2.3]	Faster time-to-market product	Quickly deliver the product or service to the customers

Category 4. Challenges

The difficulties that arise in legacy modernization project.

Concept 4.1. Technical challenges

The difficulties that arise in legacy modernization project from technical / technology point of view.

[4.1.1]	Data Migration	The difficulties that arise in the process of transferring data between data storage systems, data formats or computer systems, including translating the data from one format into the other formats. The process of moving or transferring data between storage types, formats, or computer system. The data is moved from legacy system to the new environment.
[4.1.2]	Lack of resources	Lack of reference (e.g. documentation) and limited manpower to involve in legacy modernization.
[4.1.3]	Poor system architecture or infrastructure	Old fashion of build the system or obsolete IT landscape (E.g. monolith and, hardcoded, spaghetti architecture)
[4.1.4]	Difficult to extract business rules or knowledge	Difficult to extract business rules or knowledge from legacy system. Getting out the information from legacy system is difficult
[4.1.5]	Difficult to test	Difficult to complete test case. A process of *verifying* and *validating* that a software application or program to meets the business and technical requirements that guided its design and development, and to works as expected.

Concept 4.2. Business challenges

The difficulties that arise in legacy modernization project from business point of view.

[4.2.1]	Funding legacy modernization projects	Difficult to get money to support the legacy modernization project.
[4.2.2]	Predicting Return of Investment (ROI) of modernization	Difficult to create business case as a scenario in which has positive return on investment.

Concept 4.3. Organizational challenges

The difficulties that arise in legacy modernization project from organizational point of view.

[4.3.1]	Cultural resistance in organization not to adapt new system	The ability of organization to adapt to a new culture during or after the process of modernization.
[4.3.2]	Resistance from the current users or maintainers in the organization	Uncooperative manner from current users or maintainers in an organization. Behave of certain people which are oppose the process of legacy modernization
[4.3.3]	Communicate the reasons or consequences of modernization	Interaction between people who involve in legacy modernization including how to deal with top management and Commitment Difficult to make people accept the legacy modernization together with its consequences.
[4.3.4]	Difficult to effectively prioritize the functionality for modernization	The difficulties to govern the process of modernization during long time period of time (e.g. prioritize new functionality, maintain the old and new system, etc.)
[4.3.5]	Time constraints to finish legacy modernization	Limited time available to finish legacy modernization, including time management in every step of legacy modernization.

Key Quotation

1. P1

[1.3.2]	**Learnability**	• System when you get somebody new in it would take a lot of time when these people, if you can get them at all when these people have to learn how the system works and how they can maintain it.
[1.5.1]	**Functional completeness**	• The first evolve IT was looking at an automating tasks so it is a normal history of automation *<u>at first</u>*. Handling all kind of standardizes administrative stuff which can be easily formalize and automated • Like I said before the functionality of the system is basic but old
[1.5.2]	**Functional Appropriateness**	• A legacy system is a system that works. OK, it is there because it is there and it is old because it works and it works fine
[1.9.1]	**Old system**	
[1.2.2]	**Stable system**	• So they're well performing and they are very good secured and they're stable ok...
[1.4.2]	**General Performance is good**	
[1.6.1]	**Interoperability**	• They operate within an environment. If the systems stay the same, and the system surround the systems, the environment is evolving to new technologies you get a discrepancy and you get less possibility to properly interconnect with the older system • It will be become more difficult over the years to connect to your new systems mainly in front-end environment. Yeah… I think those are the bigger problems • If it isn't properly designed, then you will run into trouble faster.
[1.11.5]	**Poor quality of design and ecosystem**	
[1.6.2]	**Adaptability**	• So the problem of legacy system I think are not in the system themselves but in the supporting technology or its only current for a certain period
[1.12.1]	**Knowledge become scarce**	• People who understand technology, the old technology and the way the system has been evolved and sometimes it is not well documented, the knowledge about the system become scarce • You become dependent on *scarce set* of skills and specific people and that can make them more difficult to maintain • A lack of documentation can be a problem. But that's with any language any system. It isn't properly documented. • They knew they were getting pretty close to the border where you hit a brick wall and trying to get someone with knowledge on the system and to extend them or to amend them • It will be very difficult to get knowledgeable people.
[1.1.1]	**Modifiability**	• they knew they were getting pretty close to the border where you hit a brick wall and trying to get someone with knowledge on the system and to extend them or to amend them
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• And they weren't dissatisfy with the systems • I'm satisfied with the way is working right now.
[2.1.2]	**Business Critical**	• Most of the systems of the legacy systems are the core system of the bank
[1.11.1]	**Large system**	• it is basically still the administrative stuff which works in batch mount on large databases • The system has been around for long time and has been tuned to better functionality, robustness, availability and high volume • Let's see small thing that functionality that count is a stability, robustness, reliability and availability of this system
[1.2.1]	**Robust**	
[1.9.2]	**Availability**	
[1.5.2]	**Functional Appropriateness**	
[1.10.1]	**Risk of running legacy system**	• Risk is one of the other factors… • Money is not a largest problem...risk is the largest problem...you know, they can afford these systems to failed only for 1 hour • it currently works but they worry if they continue it, can it continue long enough into the future...ok • on the second level you might say that people start beginning to worry if it can continue this way
[4.2.1]	**Funding the legacy modernization project**	• Now you see with this project the funding is gaited...so, business only gives small amount of money and next year small amount of money...and a little bit more... little bit more... ooo this guy is now achieving something.. • *Y*ou have problems with 10% of the systems, which is used 50% of the time but the other 50% , is 90% of the code and it still works fine. So, how do you judge when is the right moment if it cost a lot in migrate? it is a tough question
[1.8.1]	**Not flexible**	• So, we are able to more quickly adapt the certain components on their own. • Increasing flexibility
[3.1.4]	**Prone to failure**	• The other main objective is to get smaller, less complex components. So disentangle the logic and create more self-contain components • Because most of them were monolith...ok…so, I think if you looking at modernizing is not question of the language that has changed that much for the certain type processing but of course influent more on design and how to optimize things. • We are revising the central role system who work tidily couple, so more or less monolith, decomposing them in more... more service oriented way...
[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• Also continuity, so the driver behind staying current with this technology is that you want to have continuity of your core system
[2.2.1]	**Expensive in maintenance**	• That all depends on how the systems are maintained, have been maintained out of the core of long period. They are becoming more costly
[1.11.5]	**Poor quality of design and ecosystem**	• The way they were programed, the way they were designed…ok.. because most of them were monolith • It can still be a problem of the design or the requirements specification and of course if you have legacy system ill designed or old fashion design as a monolith, that might still be the reason for you to change • Because most of them were monolith...ok…so, I think if you looking at modernizing is not question of the language that has changed that much for the certain type processing but of course influent more on design and how to optimize things.
[4.1.3]	**Poor system architecture or infrastructure**	• Making the right decision in how to componentized the system
[4.1.4]	**Difficult to extract business rules or knowledge**	• [Q] For example like a lot of people might not know what the inner functionality of legacy system because the developers are not there. [A] Yeah…yeah.
[4.3.5]	**Time constraints to finish legacy modernization**	• One that I mentioned, they run out of budget...they run slightly out the time... that's mainly to do with scarcity of people on the legacy system
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• No...not that much... I think the people working on this type of things, they are always some people would like to staying that comfort zone, but I think by now is very clear that if you don't adapt to the environment and people go on without you
[2.1.1]	**Doesn't fit with future strategy**	• If it still suit their needs in the future
[4.3.3]	**Communicate the reasons or consequences of modernization**	• How do you do the whole communication of culture of geographical area, how do you secure things being developed over there, how you check things, how you vitamin to your environment... a lot of stuff.

2. P2

[1.12.1]	**Knowledge become scarce**	• In production it is legacy. My experience is and...that...at the moment the system is built...yeah...it is well documented, it is often well structured. But at the moment it is going to production, you get ad hoc changes on it and those kind of things, and what you see is that within...for some companies is within a year, for others is within 3 years, the documentation is outdated, the structure of the application is damaged...yeah...and the problems start.
[1.11.5]	**Poor quality of design and ecosystem**	
[1.11.5]	**Poor quality of design and ecosystem**	• IT is forced to come up with the quick and dirty solutions often and doesn't have time afterward to adjust it, to make it well suited in the design of the application itself. • Time is sometimes an issue, budget is less an issue. • ...if requirements...new requirements are put on the application, what you then see is that it is often quick and dirty ad hoc and those kind of things. • Business wants something yesterday
[4.3.5]	**Time constraints to finish legacy modernization**	
[1.12.1]	**Knowledge become scarce**	• Knowledge erosion is the most big problem there • If you still have the people who build it and maintain it, then it still works fine. but if they leave the company, then they end up in serious problems. And where you see that is often the pressure between business and IT
[1.2.2]	**Stable system**	• Legacy system is what you find in back office. Because back office processes are more stable than mid-office and front-office. SO you

[1.11.3]	**Back Office**	have less business requirements on that popup • You also see is that because they are very knowledgeable people, the application itself doesn't suffer from many defects anymore. Over the years all the defects have been solved
[1.4.2]	**General Performance is good**	• [Q] Already there...so performance is still good in legacy system. [A] Yeah.
[3.2.3]	**Faster time-to-market product**	• Putting new requirements time to market is long and they wanted nowadays what we find is that within business they come up with an idea, they want to have an initial estimate and time to market estimate within a few weeks and then average implemented in 3 months or something like that
[1.8.1]	**Not Flexible**	• It's more that the business has the requirements which cannot be implemented quickly enough in the legacy.
[1.1.1]	**Modifiability**	• So, what you then often see is that the old legacy system which is the kind of backbone still stays there and that is used in is built in the midterm or more in the area of mid-office and front-office. Just don't touch the legacy application.
[1.5.2]	**Functional Appropriateness**	• it may be a creepy code but it runs and it runs without any failure. • it is not the question of quality of service of the application
[1.11.1]	**Multiple system**	• a lot of banks and insurances and ending up with 3 or 4 systems overlapping functionality. Because they cannot decommission the old one
[2.1.2]	**Business Critical**	• So, one of the nice things about that is we also a lot of work at customers decommission really old application.
[1.11.5]	**Poor quality of design and ecosystem**	• They have a lot of legacy applications and it is hard coded often in the code.
[1.7.1]	**Unsupported supplier**	• I mean what you trigger or also things like for instance a hardware. I've seen customers running systems on hardware that run out of support. And even hardware that if something is broken down they have to look on E-bay to find spare parts. • so when your environment runs out of support then it is really dying and if that's true then you are already late
[2.2.1]	**Expensive in maintenance**	• Yeah...cost reduction. • Cost of mainframe is perceived as high. And they want to migrate way to migrate away to mid-range systems
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• People really want instant correct information. And that might drive some companies to...well...to do modernization...yes...to enable the legacy application to cope with those kind of information request
[1.11.5]	**Poor quality of design and ecosystem**	• Very difficult is for instance in new old client-server environment, power builder for instance...yeah... That is just a disaster to work on it, because it is very easy ...it is a lot of logic is concentrated on the client side or can be. • And what you also find is that one application is developed that it uses also the database of another one. And it is not through a normal interface but it is via back door to get some data over there
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• You need C level support for that and you have to involve business always. Because if you have impact on the business and especially when you look at large modernization it means that the business will need to be involved • The biggest challenge I find always is to get C level support for this kind of program.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• Get really commitment for that level...from that level to really set up the whole period. And not well…OK next year...o well...we have less budget...well we stop and we quit.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• And if the company or that customer doesn't govern the modernization as one overall program, but looks at the application individually
[1.3.2]	**Learnability**	• New one comes in, he has to learn about it...well...it is not something you can do in just a few months and then if changes...
[1.9.1]	**Old system**	• So, if you look at applications in that category what you will find is that system runs for 20years, 25 years
[1.6.2]	**Adaptability**	• That is very hard to align that with a very old system of course
[4.2.1]	**Funding legacy modernization projects**	• And what is the alternative? yeah... is to rebuild the sytem? I can tell you that's more expensive than restructuring it and making it maintainable • So they haven't implemented that, and it is a lot of time money to still implement it, so they keep up running the old one. • So you always have business case over there. And the moment that business can yeah...so it is not positive and for many companies if the Return on a Investment is not receive one year and say we don't do it. • They are always looking for a short Return on Investment. One you put the money in, that they want to earn it back. And...But what you now see is last year I had customer who said I want to it but I want to earn it back in 6 months. If you have talk about a project of at least one year you cannot and never earn it back in 6 months. So, only if we get in more economic better times and we can stretch it a little bit because modernization is not something that you do in a few months. That will takes a year or 3 years or if there is very big program you can find yeah... • Modernization in such a way that it is cost effective to do the job. It's not rebuilding a complete new...of a building...a complete a new system or turning the legacy system upside down or inside out. But just making those kind of changes that is cost effective
[4.1.1]	**Data migration**	• And at the time they have a new system they want to migrate all the data from the legacy to the new one. And then they find that...oouww...but here we have something...ouww...we haven't handen foreseen that...yeah...
[1.5.1]	**Functional completeness**	• it performs basic func...still support basic functionality of the business processes
[1.4.1]	**Time behavior**	• But On the other hand, it does the job is efficient, it is fast and that's more less if talk about back office system, that's where combination are looking for ...yeah...reliable, fast…
[1.4.2]	**General Performance is good**	

3. P3

[3.2.2]	**Reduce cost of maintenance and operation**	• [XXX] looked and said it cost a lot of money, a lot of danger. So last 10 years KL migrated 5 big systems.
[1.1.1]	**Modifiability**	• So the technical driver are old, difficult to maintain, the knowledge was not easy to get because PL1 programmer is not easy to find. From business side, it become expensive to technically fulfill the business reason.
[1.9.1]	**Old system**	
[1.12.1]	**Knowledge become scarce**	
[2.2.1]	**Expensive in maintenance**	
[1.11.5]	**Poor design and ecosystem**	• It's monolith, everything in one source code and in one machine.
[1.4.2]	**General Performance is good**	• The performance of legacy system before we migrate it was enormous quick. But the performance of the new system is more an issue.
[4.1.1]	**Data Migration**	• How we are going to migrate the data into the new system. Because it is operational it has to be real time accuracy. Since it is operational system, you have to do it right, one time. There is no re-tried, because if you start working with the new system and you find out that something is wrong with the database, it is hardly impossible to repair because at that point the whole situation is not up to date anymore, is not maintain anymore so it is not there anymore. It's a really big challenge. And also some of the data is inconvenience, it is not properly migrated.
[1.11.2]	**Database issue**	• Yea…and that's the problem, because it doesn't have a referential data.

4. P4 and P5

[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• legacy system is a system which service life has expired
[2.2.1]	**Expensive in maintenance**	• maintenance is very expensive...too expensive • The maintenance is cost higher.
[1.1.1]	**Modifiability**	• Not maintained that means that no functional updates or no security updates
[1.9.1]	**Old system**	• That's obsolete technology.
[1.2.2]	**Stable system**	• Most of the time it is stable.
[1.3.2]	**Learnability**	• People are used to the system. They know how it works…That's yeah…
[1.9.2]	**Availability**	• Proven technology.
[1.12.1]	**Knowledge become scarce**	• But the documentation of the old system is a primary
[3.1.1]	**Become flexible to support changing business requirements**	• I think in government are, we have influence of politic.
[4.3.2]	**Resistance from the current**	• If the organization want to modernize their legacy system, there's certain people who try to not cooperate because they're an

	users or maintainers in the organization	expert in legacy system. By the time when legacy is modernized, they might lose their job. Do you see kind of phenomena in organization? [H+T] Yes...Yea..
[1.6.1]	**Interoperability**	• [Q] Is it difficult to interconnection with...interconnection application with you legacy system. [A] Yeap.. it is difficult. We try to standardize interface

5. P6

[2.1.2]	**Business Critical**	• I think it is a core system, most of the time • like this core transaction,
[1.8.1]	**Not flexible**	• they also have a lot of limitations on a flexibility • very rigid...let say if you wanna change it like data structure, it is not clear for me what exactly the data structure. • Yeah... it is not flexible. It's a lot of hassle. • But actually, like most companies now say but we wanna be flexible
[1.11.5]	**Poor quality of design and ecosystem**	• And I could see people are using at the...they had a legacy system but during time they also have complementary systems like workflow system. And using both and using all top to do little bit here, little bit there.
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• But still you can see they trust in it , because it is working...some guys said last week, it is working for 20 years now
[1.5.1]	**Functional completeness**	• But for me is all about if you would define, it is about limitation. You see they have street field which is limited to certain characters. • But it is not process thinking • it is only simple logic, this simple operations • there is no a lot of ...most of the time, there's not a lot of complex logic in it • So it is just you can like update, maybe the address of the client but it is not called the address change
[1.12.1]	**Knowledge become scarce**	• I think that's a lot of knowledge which is only in their head. • Like if you're young like you and me who decides to start learning COBOL and working in it. It's also they know that system because they work with it for 10 years. If you want...what kind of guys for designer if you.. like end of 20ies, you just graduated, and nobody decides to start COBOL • [Q] Is that means...Is that means that the maintenance of legacy system also expensive or not? [A] Yeah… it is... so and even though it is not expensive, it is hard...I think it is hard to find like skilled people or people who want to do it and who know that all that things. • there is more IT, they also spend more in IT
[2.2.1]	**Expensive in maintenance**	
[1.2.1]	**Robust**	• Yeah...just like what I said before, it is robust,,, • I think these all UNIX systems. They don't crashed often.
[1.9.2]	**Availability**	• Like if you look what's up time, when did the system start, could be 2 years ago. Like the system they are online, they are...they always available and because there is no a lot of ...most of the time, there's not a lot of complex logic in it
[1.1.1]	**Modifiability**	• changes are hard • A lot struggle to change that, because all these logic about that is totaly integrated into the system.
[1.3.1]	**User interface aesthetics**	• Interface...user interface sucks...user interface is not very...is not modern...you get this back…yeah this old screen with some data on it...
[1.3.2]	**Learnability**	• I tried to explain it last time to someone who was taking something over from me. It's like, I don't get it.
[3.1.1]	**Become flexible to support changing business requirements**	• There was a new regulation, which says they had to make policy to be gender independent. • They had also too much different customization from the policy...so, they're also bringing their policies [pack] to limited...was really complex • That's an important one legal change...
[1.11.5]	**Poor quality of design and ecosystem**	• Also the business logic, so...let say you wanna determine what's the premium, that's all hard coded. That's especially hard coded, in the logic to find it, to change it, it is very hard
[1.11.2]	**Database issue**	• Data garbage. • So the quality of the data
[4.2.1]	**Funding legacy modernization projects**	• And in the process of choosing a new system, they said like, they could see...there is a lot of cost [much] reduction with PEGA, but they say we know this old legacy system it is limiting us to much • Should also have budget right like if he says we are gonna do that, and then you can also put the money that way, because sometimes that's a financial system, financial department involve. • Main thing is always budget, so they should be money...
[4.1.1]	**Data migration**	• I've seen very strange thing, it is just missing for maybe like 5% of the record some really important data it is just missing...hahaha... • Like why don't we use our own, if we started after their own databases tables, and work like that, the project would be quickest so much...you know • Could be like 40 small insurance companies which in the time like 20-30, merge to each other and merge again. Imagine every company have their own data, so they migrate their data at the time
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• Because what's the need if we have new system, which is working not with COBOL, who's gonna need me anymore, so they ditch me after it is done. So then they think like hei...why should i cooperate. • people all have kind of different motive not to work along • Like people should cooperate but how do you get them there for
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• Live close to Rotterdam, i like my job because I can go to my work in 15 minutes, and then at the end of the day 15 minutes. I like my…the way my life is. I like my job, i like the department.
[1.10.1]	**Risk of running legacy system**	• I think the company will go out of market.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• Someone like...sometimes you see like make the decision, but he should be high up in the chain like if they say it…how do you call it...mandate...it should have mandate, the guy should..i think he should be on such a high position.
[1.11.1]	**Multiple system**	• Within history they first had this system and then a few years later they thought o yaa...now we need this system. So they have multiple systems for policies....
[1.9.1]	**Old system**	• the older system • It's like if they choose the system, they're gonna use it for a like 10 years.
[3.2.2]	**Reduce cost of maintenance and operation**	• They wanna have rather like one person having just one user interface.
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• You need people, skilled people to make services
[4.3.3]	**Communicate the reasons or consequences of modernization**	• There's also misunderstanding • You get a lot blaming...sometimes we get the right people together to solve, just solve the problem, it is a big challenges
[4.3.5]	**Time constraints to finish legacy modernization**	• Just time...so if you wanna change something we say like can you change it. O yeaah...and then...it also thing they say like yeah...we only have limited capacity, because they don't have guys, old guys doing it • First you have to change processes, you have to implement it. I think that takes some time. If you choose who gonna implement it, who's gonna test it? How you are gonna migrate the data to the new system...bla..bla..bla..and then It's also maintenance about the time I said...implement it you wanna...there is be maintained by some....There is also gonna be like some minor changes during...but if they chose their gonna use it for the coming 5 or 10 years.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• Like also problem this kind of guy that's not really good in communication

6. P7 and P8

[1.9.1]	**Old system**	• So legacy system for me is an older system
[1.11.1]	**Large system**	• ...on average is a large system because small systems don't impose to big problem.

[1.11.4]	**Complicated system**	• A lot of functionality, lot of business rule, business logic implemented and it is develop over the years to very mature degree. • So there's always something with technical complexity, a lot of components, a lot of layers, lot of programs and interfaces interacting with each other
[1.2.2]	**Stable system**	• And that basic functionality is stable for very long time because that's all for whole year, it is working that way • The functionality is stable...just let it stay as it is
[1.4.2]	**General Performance is good**	• So the functionality is on average is OK
[1.1.1]	**Modifiability**	• But the problems is with the technical quality, the maintainability is for several reasons bad • You have encountered some problems with maintainability • Cannot maintain it efficiently because quite simple change takes so much time because no one knows exactly the system works. • That's a big change for a system like that, and that kind of changes that is difficult for legacy system. So it impacts very much the existing functionality
[1.11.5]	**Poor quality of design and ecosystem**	• You made adaptation on and adaptation on...and the internal technical structure is not very optimal as it was, just after you started developing the system. • So you fix something and something else does not work anymore. So, by doing the changes and changes and changes, it gets the...less...less optimal the system.
[1.6.2]	**Adaptability**	
[1.12.1]	**Knowledge become scarce**	• The documentation is not...if any documentation is not actual. The knowledge of the people who built it is gone because people are gonna away to other job. • Or I don't have the personal anymore who can do it. Or I don't have documentation and I don't even know how it should work the system. • The problem is our COBOL application is that the people works on it are retiring. So they are leaving the company then...So our knowledge is going away about the system • Lack of knowledge about what it is going...what now effect..well being executed by the program and the coding. That can be the main challenge • [Q] Yea...yea....so do you think that to find people with the knowledge in COBOL is difficult now? [A] Not at the moment. But it will be in the future
[4.3.5]	**Time constraints to finish legacy modernization**	• ...in the time pressure. You don't have the time to do it good or to refactor the coding or there is a pressure because you do not have sufficient knowledge about how it should it works. • I last too long...We plan modernization for modernization for 3 years, and after 5 years we stop the whole modernization, and start it over.
[1.12.1]	**Knowledge become scarce**	• [Q] Yea..yea...so do you think that to find people with the knowledge in COBOL is difficult now? [A] Not at the moment. But it will be in the future
[1.11.3]	**Back office**	• I think another characteristic of legacy system is that they are mainly back office system.
[3.1.1]	**Become flexible to support changing business requirements**	• Law, can you give an example of what law maybe. This is from politics the change of the retire...what it is…retirement. • But problems is that legislation of pension fund is changing continuously • Definitely business requirement changing overtime
[3.2.3]	**Faster time-to-market product**	• To launch quicker new product for the customers
[2.2.1]	**Expensive in maintenance**	• So the driver that the maintenance of legacy system causes a lot of money, well it is only a problem when our customer will not pay anymore for it and we still have the obligation to keep it in the air for some years. That's another aspect of legacy system, you have a lot of customers. Together they are paying a lot of money to keep it in the air. • So one of the main factor is the cost to do changes becoming higher and higher, compare to the change itself • the hardware on which the system runs for example mainframe becoming too expensive compare to cheaper hardware nowadays available
[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• Yes...it is strengthens the lifecycle of the system.
[4.2.1]	**Funding legacy modernization projects**	• ...budget we have for that... Time available... yeah...capacity... • And it is not easy to migrate that to another system, to change to another system. It's always, it requires a big efforts to change to another technology or to change the system to another system. So that's the reason why the old system keeps [ins] in use for longer and longer.
[1.7.1]	**Unsupported supplier**	• System is built with program languages, library and frameworks that no longer support or the suppler no longer support
[1.5.1]	**Functional completeness**	• it is not technical automation. It's not embedded software for example
[2.1.2]	**Business Critical**	• and in fact some of our legacy system are the most profitable of our system. Not because the maintenance is done very efficiently but customer pay a lot of it to keep it in the air
[4.3.2]	**Resistance from the current users or maintainers in the organization**	
[1.3.2]	**Learnability**	• What I'm telling is sometimes you do not know anymore what the coding of the system supposed to do.

7. P9

[1.6.2]	**Adaptability**	• [Q] So you..can you categorize UTMS as a legacy system or not, in your opinion? [A] Auuuhh...good question...I think no...but that's more ..well it is now 10 years old, 10 years ago. so could it be legacy but the technolgoy isn't legayc right? So the technolgoy is still used in 4G networks well...
[1.11.5]	**Poor quality of design and ecosystem**	• systems that are where the development environment is out-dated. • not only the development part, but also the architecutre I think in the system as well. Becuase if people are bumping against artichecture changes or misguidance in earlier architecture....
[1.9.1]	**Old system**	• the alarm recceiving center is a the heart of alamr reciveng center is a application that surround, I think 30 years old or so...
[1.6.1]	**Interoperability**	• That communicate with, via X25 connction. So that's all old stuff and then you have to have a legacy ssytem because nothin gelse can talk that stuff again . • ...If you integrate application to another system, if you link applicatoin...You can do it on different ways. YOu can say well...I only have low level API and I just presented data by myself. Or maybe the data is already presented by legacy system or whatever system... • a lot of appoctain runnign aroud it the thing to ofload it adn that kind of thing, but it is OK...but if you look at harmony, this applciation is called harmony from [ORTECH] and we have a lot of probelm with harmony
[1.5.1]	**Functional completeness**	• So the problem you are having at that point is that..that you are being limited in fucntionality by the legacy system.
[1.1.1]	**Modifiability**	• The effort to build the fucntionality is so high that is not a good business case. So if for instance we have to put in another data field in the system and that takes 4 weeks to program then yeah..you probably say well...sorry the system doesnt support that functionalty anymore...sorry...you have to..we have to start modernization again. • Because legacy system tend to have bad maintanable code
[2.1.2]	**Business Critical**	• legacy system in the core of business
[2.2.1]	**Expensive in maintenance**	• The amount of people who know that legacy syste or the environment is build are limited. So that also the reason why cost more money. And probably in time you will have a problem with update of may be platform OS or those kind of things. • but reason they dont ...that's a lot of money, because their framework doesn't support..they do something streange with enablinf the web part of that applcaiton. and change that is a lot money
[3.1.2]	**End of product life cycle**	• Invest in new applion or modernize the application so they can scretch the life cycell of the applciaton agaoin • Out of the life cycle
[4.2.1]	**Funding legacy modernization projects**	• it is very expensive to chnge it and we want to change it...still runnign fine so that's not problem
[3.1.1]	**Become flexible to support changing business requirements**	• And the screening at the gate and the screening at the gate just schipol the only airport taht does taht..it is needed becasue the passeger flows mixed. So arrivng passengers and departing passengersmeet. And that's the problem so schipol is going to...
[2.1.1]	**Doesn't fit with future strategy**	• legacy system doesn't facilitate them doing their business
[4.2.2]	**Business Case**	• For filling the business case.
[1.11.4]	**Complicated system:**	• legacy cystem aslo tend to hide complexity of businesss process

[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• You do not want COBOL programmer, but what do you want is somebdoy who understands COBOL and can extract functionality in logical way from the legacy system
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• There is data on how much ICT project is always overbudget and takes to much time right? so there is some prblem in our management concern in ICT project.
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• [Q] when you do modernization do you this kind of fenomena. LIke a people in your...maybe in your team, they are dont really cooperate with the modrnization process because they are an expert with a modernization process becasue they are an expert in the legacy systme? [A] Yeah..that happens.. • *So,* the most important thing is, I think to make sure that the people you have in your company dont...arent link to a spesific system but have another goal, maybe maintain multiple systems or build some other alpplicaitno whatever
[4.2.2]	**Predicting Return of Investment (ROI) of modernization**	• And the amount of chnagea is going through the legacy system wasnt really foreseen at the moment we built it, we build the application. So there's some limitations in the applicaton itself.
[1.11.1]	**Large system**	• We have a big system running
[1.8.1]	**Not flexible**	• And if..it 's legacy system behind it then I can imagine, that you are less flexible presenting the data than newer systems
[1.12.1]	**Knowledge become scarce**	• they get a beard and that kind of thing...and at some points those people
[4.3.5]	**Time constraints to finish legacy modernization**	• It's a long project
[4.3.3]	**Communicate the reasons or consequences of modernization**	• I dont have a real trustwothty data for them...OK it is going to crash on 12 of November 2013 and after that date your business is gone. So and that's not a case

8. P10

[2.1.2]	**Business Critical**	• I think the major issue about Legacy system is they are valuable • They have valuable business value...great business value
[1.11.5]	**Poor quality of design and ecosystem**	• they don't fit into the way people want to work and how infrastructure work of today
[2.1.1]	**Doesn't fit with future strategy**	
[1.1.1]	**Modifiability**	• *T*hey trouble maintaining it, adjusting it • I think it will be harder and harder to change the system
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• [XXX] on improving on the customer experience • customer don't like it anymore. so the human experience, the user interface which is old fashion.
[1.12.1]	**Knowledge become scarce**	• then you saw that only very few people who could understand how you could do that and...so that is the system is legacy, but still there. • people don't know the rules anymore because they never use them because the systems do the work. So there is no business knowledge anymore in the business. • Nobodies knows all the rules anymore which are in the system • Like Javanese, the high Javanese language, the core language, very few people understand it . And so that's if you're company, you have a system built in such a language which is of course we hard to find people to help you to maintain it and adjust it, then it is a problem • I think the big problem that you can't find people to understand them and understand the technology
[1.11.1]	**Large system**	• This legacy system have enormous amount of very complicated business logic inside.
[2.2.1]	**Expensive in maintenance**	• 200million for maintenance.
[4.2.2]	**Business Case**	• it is a lot of money...it is enormous of money. But would it be cheaper to make new system?
[1.11.2]	**Database issue**	• the database issue. they have same issue with database management and they want to renew that.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• We should do it so quickly that's there are no changes almost. Somehow we have to compromise a little • If people start to change the system or the requirements of the system while you do modernization • Because it looks appealing to do more than only modernization. If you go with your car to the shop and have it fixed, and say... Wel maybe you should paint it another color. • [Q] Yeah...you change the decision in the middle of process.. [A] Yeah • The big puzzle is how to do modernization of this system while keeping it maintained
[4.1.1]	**Data migration**	• if you migrate the data form the legacy system to the new system. Often that's also a big issue...it is even more complex than the writing the code. • They have all the issues about legacy and all the issues about new system at the same time. but that's because data migration is very complicated
[1.11.1]	**Multiple system**	• So they do the new business in the new system, and old system...
[1.6.1]	**Interoperability**	• *P*eople are still have trouble making the interface of soa simple and transparent that are services is easily used. yeah....it is a big challenge.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• I think top management doesn't understand the issue...no..they don't give budget for it...

9. P11

[1.9.1]	**Old system**	• there's a lot of legacy systems that are 20 30 years old already.
[2.1.2]	**Business Critical**	• so it is an old systme, it is an obselete system maybe but it is very usefull and has a business impact still and generate a lot of revenue for banking and tehir clients, becuase it is still a lot of legacy system in the core system. • No...good is good enough let's remain the leacy environments. Because then they can incerase their profit
[1.9.2]	**Availability**	• Normally, because there are not many changes on legacy system availability, they are very available,tehy are long time online and they are more less 24/7 up and running.
[1.4.1]	**Time behavior**	• It is also characteristic about response time, it is very fast because normally it is very simple small data messages, so there's not that a lot of everhead
[1.12.1]	**Knowledge become scarce**	• more or less not that many knowledge of that system in organization because the people who are knowlegdable about this kind of system are already gone. they are already left buildng you might say . and documentation, so the characteristic also from old legacy system that is the lack of documentation, because it was never docuemted in those days • We have 100thousand people working for [XXX] in India. That's the biggest part of [XXX] is in India. But there is a lot of knowledge around there. So we have built up educaiton around legacy transformation in India itself.
[1.1.1]	**Modifiability**	• it is hard to maintain, because beside there is no real updated software anymore and it is also not supported by a lot of other organizsaitons
[1.7.1]	**Unsupported supplier**	
[2.2.1]	**Expensive in maintenance**	• that is maintenances and because the maintenance is not there and if it is then it is the cost of maintenance, the maintainance is very high. So that will increase the total cost of the owneship of the system. • Lisence cost of the software. that is also very expensive.
[1.6.1]	**Interoperability**	• The second one is a system integration. Becuase old date systme has most of more or less have old applicaiton program interfaces. So there is more or less a lack of knowledge about integrating legacy systmes by itself. • An other issue or challange could be system conversion also from software perspective.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• No because the..especially for organiation who are 24/7 in the air. It is very difficult than to have a shodow system besides the normal system because it must be up and running. • Becaue you are changining in a running environment. SO you need somewhere, somehow you need to determine OK..what is the best moment of the change

[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• It is complex, becuse you have a lot of differnet expertise needed yeah... not only need people who undersatnd database environment and operating systems, middleware, things like enterprise services bus of architecture but you also need business who understand business, business functionallity.
[2.1.1]	**Doesn't fit with future strategy**	• another discussion, because then you are not be able to say this legacy system can run for 5 or 10 years so it will...if it is not work properly anymore, it will disrupt business model of the organization.
[1.5.1]	**Functional completeness**	• that the older system cannot deliver what the new system can deliver, because the new system has also thte capability of ananlitic and that could be also predictive analytic, so you have more functionality in the current new environtments around data • So how do you make from data information and how can you make from information derive intelegent and how can you base on that intelegent can ake proper decision.
[4.1.1]	**Data migration**	• Now you always see also see a lot of unstructure data,that is more difficutl becaseu 75% of the generated data of today is unstructred. It's textes, voices, movies, pictures. So that is an adapt...That is more complex to transform into anthoer because it cannot be stored in an hirarical relational database.
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• Sometimes people do not like to change • Sometimes they start the transformatino, they forget to educated the people. • Change that the organizaiton is ready for the change. Is the culture in the organizaiton, do they allow a change for that kind of different information
[4.3.3]	**Communicate the reasons or consequences of modernization**	• So you need to persuade them, the why and the how for the transformation • communicate and infrom the people who are responsiblem for transfroming, so in programitself, but even more important is to communicate and infrom the poeple outside the program
[4.2.2]	**Predicting Return of Investment (ROI) of modernization**	• creating a business case. are you capable to describe in ficancial perspective what are the revenue? what are the cost and are [the show] Return On Investment.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• The third one is of course transformaiton plan itself. What septs will you...do you need to prepare to govern the whole transformation plan by itself. So you need to very good project manager • How do you organize and how do you control all the transformation. also take into acount availability
[1.11.5]	**Poor quality of design and ecosystem**	• That has everything to do with I think the architectural technical complexity

10. P12

[1.7.1]	**Unsupported supplier**	• I would define a legacy system as a system which is constrained by hardware, specific hardware which is no longer fully supported or being very scarce. • if the legacy system is depend on specific hardware, then if you cannot [stock up] on that hardware, then you completely relying on its functionality. If it is break down, then you don't have a choice and it is too late
[1.1.1]	**Modifiability**	• It' maybe constrain by technology, for instance it is built on technology that is no longer maintained.
[1.12.1]	**Knowledge become scarce**	• or it may rely on specific people because lack of documentation or complexity of the system
[1.11.4]	**Complicated system**	
[1.12.1]	**Knowledge become scarce**	• mostly i don't see a lot of documentation on that work • Many companies now are very [relied] on very specific persons and they don't really like tah because you cannot control when the person leave the company. And sometimes they even old enough to nearly retired • if it can help if teach student what is the value of the legacy systems and how to work with that systems.
[4.2.1]	**Funding legacy modernization projects**	• If they modernize it now they have to completely reinvest. • They mostly do not want to do that.so if they choose to completely rebuild, it will be major investment. Many man year project
[2.1.2]	**Business Critical**	• of core in the system where is very difficult to find out what are exactly other rules that are implemented
[1.2.1]	**Robust**	• They have been around for many years and during those years they have been stabilized so a lot of companies tend to see their legacy system as very reliant.
[1.5.1]	**Functional completeness**	• simplicity it has, for instance user interface, very simplistic because it keeps out all the unnecessary stuff • and sometimes they simplicity of the chosen languages or technologies also make it very stable, because it is not possible to add all these kind of feature during the years
[2.2.1]	**Expensive in maintenance**	• [Q] DO you think that the maintenance of the legacy system is expensive? [A] Yes I think so... • the goals to remove those people from the payroll and of course they doesn't really help with their involvement in the project • It can be desire to reduce cost, for instance by moving to standard software. • But if you can move to standard product, then it could be usually an advantage because the maintenance cost for standard product is usually lower
[4.3.3]	**Communicate the reasons or consequences of modernization**	• ...It's really difficult, but you can only do it when it also part of the assignment. because sometimes they don't really feel a need to change the culture • not difficult to find him but sometimes it is quite difficult to work with them.
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• sometimes they see the legacy system as their baby and they tend to know every aspect of it. And sometimes it is quite difficult because you are in the team coming from the outside to perform project and they consider this as their primary area. • sometimes it can be a bit difficult and to let that go and start working on new technology and new system.
[4.1.5]	**Difficult to test**	• ...it is usually testing. • And it can be very difficult to extract all the use cases, the original system has supported, and because that difficult also difficult to get your test case complete
[4.1.4]	**Difficult to extract business rules or knowledge**	• the company or the project team has to extract exactly the internal functionality of this legacy application. It can be difficult to extract it and to document it an to implement it properly • It's more challenge for…to get requirements very clear and complete. • help to improve the insight into internal procedure of this legacy system and not in the very technical way • So I think it can really helpful to provide insight into the internal working of the system, to extract it to a human readable diagrams or documentations
[4.1.1]	**Data migration issue**	• a challenge can be to migrate the data. For instance to..convert from specific encoding or data structures or specific storage types to a new database • during the process of migrating you have all kind of data quality issues • you want to migrate it completely with all its problems or do you want to improve on the quality as well? But if chose improve on the quality you may end up with the whole new project on your hand as well. Because improving data quality is..can very..well...be large project on its own.
[1.11.2]	**Database issue**	• Because this system, they have been running for many years they tend to be some dirty data inside it.
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• they can be satisfy with the system, but they can be very dissatisfy with the risk building up offer the years.
[1.10.1]	**Risk of running legacy system**	
[1.10.1]	**Risk of running legacy system**	• the risk can be a very strong driver. • But for the insurance they really take care of the risk because they can't afford if their system goes down for like one hour • it can be to mitigate risk, they see it as an operational risk when staying with the current
[4.3.5]	**Time constraints to finish legacy modernization**	• And it will take many many months or sometimes even more than one or two years for the project of modernization to complete.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• Because if it takes that long the world is changes during the project. So you have to adjust to that. If you don't do you deliver, you always deliver late. Because you deliver on specification which 1,5 year old and then • And during that period they will still need the old system. Mostly you will see they will keep running their original system, sometimes even keep performing maintenance on the original system because of the project of rebuilt takes that long, they cannot afford to stop on maintenance
[1.8.1]	**Not flexible**	• they expect more flexibility when thye want to change their busness process • they expect more cheaper and faster ways of implementing those changes

11. P13

[1.9.1]	**Old system**	• Legacy system for me is a combinatin of hardware and software whcih was buit long time ago
[1.12.1]	**Knowledge become scarce**	• It is usually built or intalled or configured by someone or somebody who is not around anymore, so lacking knowledge. • [Q] SO knowldger scarscity is a problem in the future. [A] Yeas..yeass..yeas...
[4.2.1]	**Funding legacy modernization projects**	• it usually a lot of time and energy and money and research involve to get rid of it • So it is a lot of invest. It cost them a lot of time. • basically lot of things have already said yeah...so it cost a lot of money and alot of time. basically you can only invest euro once
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• OK now it is time to stop that and we going to move to the new world. Like merger or acuistion or reorganizaiton or something like that • Natural moment. so basically ifsomething else happen then wihich forces you to invest in the application anyway, then migrate it to a new platform.
[2.2.1]	**Expensive in maintenance**	• So simply the skill is not available, the only way to by pass that is that upi hire quite expensive, dedicated consultants. • It is getitng more and more expensive because risk are getting bigger and alos knowledgeis getting scares • You have to hire somebody quite expensive to make your legacy world SEPA comply
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• You do involve the busenesss,the user, the ownes, to test, to moiniteor ,to see if the applicationis working.
[3.2.2]	**Reduce cost of maintenance and operation**	• We want to do move toward a standarized way of working in the entire compnay. So standarize from business process point of view
[4.3.5]	**Time constraints to finish legacy modernization**	• Your biggest prblem is a availability of resources and availability of money. and les extend availability of time. • The business impat is alwats the beggers one. An it is usually just sinple time and money
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• Usually, the people who use th esystem are satisfy with the system
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• Basically, wha tyou need to have is you need to have strong governance model. You need to have one clear set of rules, who make a decisions and what the impact and magnitut the decision can be • Yeah...so you always have to make sure from ok...if i do a migration i always have to do lot of check and balances. I have to d oo yea..i have to do comply check...i have to do assessment, i have to do...i have to make sure that everything works, including the people.
[1.10.1]	**Risk of running legacy system**	• Basically the risk could be the legacy system falls down, our production process is a stop • Continuity. That the biggest problem. because it is old it will fall down.
[1.1.1]	**Modifiability**	• becuase it is old and the knowledge is not availabel, no body can do [adjustment] someting like that
[1.8.1]	**Not Flexible**	• implement a changes coming from your environment like the SEPA project or couple of years ago the implementation of euro with stuff like that and you have one system, you can do it quite fast. if you have to do that same change in 20 or 25 systems, it is a hell of lot more work and therefore hell of a lot more cost
[3.1.4]	**Prone to failure**	• From the cost avoided point of view to have your environment upto date and intergrated and standarize as much as possible. In the end that's a chepest way run your IT
[4.3.3]	**Communicate the reasons or consequences of modernization**	• sombody quite high at the top. somebody who can enforce decision where they need to be enforce. and sombody who can also oversee the impacat of going left - going right
[4.1.5]	**Difficult to test**	• So there's always human work and alos always room for human error. • No...well...testing somtimes. but that's it. for now and then we use every known... and we had use a scripts to for instance to deploy an image an applciaitons on a virtual server and stuff like that

12. P14

[1.9.1]	**Old system**	• let's call it a system which is older
[1.12.1]	**Knowledge become scarce**	• you have less knowledge about • And you notice that not everything is documented properly • in meantime we ask now the people please document your maintenance procedure, please document your recovery procedure
[2.1.1]	**Doesn't fit with future strategy**	• does not fit let say your future roadmap. • but you know you want to get rid of it in the future.
[2.1.2]	**Business Critical**	• Legacy for me is let say business critical
[1.2.2]	**Stable system**	• Legacy by itself it is stable
[1.4.2]	**General Performance is good**	• The performance is enough
[1.10.1]	**Risk of running legacy system**	• So it is more about continuity of the legacy system where the risk is • Can mean the risk of higher cost, can mean risk of losing people that go away that you don't have knowledge
[1.9.2]	**Availability**	• It has proven, it is stable but it is also has proven
[2.2.1]	**Expensive in maintenance**	• So if you look at the cost, I'm quite sure I can run on similar environment against lower cost. If I would run a standardize...let say state of the art ERP environment...because I don't need those developers • So I'm quite sure I can make it cheaper but that means you have through 2 or 3 ERP implementation globally to implement a new ERP system and then take out ...let say... those IT staff. And get rid of the IT staff.
[4.2.2]	**Predicting Return of Investment (ROI) of modernization**	• Are you willing and then you have to calculate the risk of the how if it can be happen. And then you make a decision
[3.2.2]	**Reduce cost of maintenance and operation**	• standardize means no customization anymore, if I want something extra to the system, either it must be configurable or the vendor deliver that needs of new functionality and its part of the new system.
[1.11.5]	**Poor quality of design and ecosystem**	• So one of the challenges is how do you migrate all your interfaces which are point to point instead of SOA
[1.6.1]	**Interoperability**	• Now interfaces last year we are now developing more in the SOA way which means the new interfaces are easier to connect to the new ERP. But the old point to point that would be difficult how that can be solved • we have a huge set up of interfaces. ERP is the core of our company almost every system talk, synchronize or asynchronize with the system
[3.2.3]	**Faster time-to-market product**	• Business driver is time to market.
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• what I see is our business opportunities and our legacy cannot meet the pace for make new development
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• So have a good project... program...how do you do the program management? How do you do the change management in project? • proper governance in those kind of project is crucial in my opinion. And that should be at let say at the right level of the organization
[4.3.3]	**Communicate the reasons or consequences of modernization**	• And there is on one hand a helping with people, explaining to people, we want to different, explain why we want to do it??
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• Because people have really get used to it that they cannot walk... • I want to prevent that I develop something now here that does not gonna work in UK or US. or whatever the location. So the culture here is a people are used of just doing that bilateral
[4.1.1]	**Data migration**	• so if your old system, one of the problems here we have is data quality. • strongly believe if you are doing migration of your legacy, you have to prepare a good data migration strategy • And preferably, as much as automated as possible. • To speed up the process, but also for cost. I think for automatic data migration, especially when you go...
[4.1.4]	**Difficult to extract business rules or knowledge**	• having good requirements. • Requirement before and requirement after
[1.7.1]	**Unsupported supplier**	• we have to ask the vendor please develop that part in your software and make it as a standard feature
[4.3.3]	**Communicate the reasons or**	• like the managerial part also about those bigger legacy system have also impacted

	consequences of modernization	
[1.8.1]	**Not flexible**	• can make it pluggable like plug and play by having SOA environment available

13. P15

[1.8.1]	**Not flexible**	• is it flexibele enough to be changed in order to support the future requirements. So that's flexibility
[1.11.5]	**Poor quality of design and ecosystem**	• Is the technical architecture of this applications still uptodate and still supported for the near future?
[4.1.1]	**Data migration**	• the larger system though, the larger back office systems, take some time, because you need to convert data. • the data transformation cannot be done perfectly . errors are made. so you have some risk that your operation is a little bit disturb after transmormation
[4.2.1]	**Funding legacy modernization projects**	• This is time consuming, and costly. • transformation of old system to a nes system is costly since you need to test it very carefully. And testing is costly
[4.3.5]	**Time constraints to finish legacy modernization**	
[4.1.5]	**Difficult to test**	
[1.6.1]	**Interoperability**	• Legacy system typicaly are old in a integration with the internet. They do not support integration with the internet.
[1.5.1]	**Functional completeness**	• [B]So you expect your target system when you do modernization would be more beneficial than your legacy system?[P] Yes...absolutely. • Storing data in a structured way, but not supportive in lot of calculations or validation of the data
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• really you need people who undrestand the old system in order to transfrom the data to the new system. And those people are usealy quaite scarce.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• data trasnformation is not a sexy subject. • The good news is in this company we have make it joined target. So business and IT, top management have a join target in decreaising the number of applications • i believe but that's asuccess factor if you make it talk of the time by top management, it helps tremendously.
[3.1.4	**Prone to failure**	• So that on point of agility. I think having less applications. Make sure that cost probably run down, and make your company more agile and more easy to be change to the future requirements. • if you have application landscape which is to complex, you need to simplify it, otherwise you are not agile to the future. • We need to implemetn as an insurance company, quite a few legal requirements. If you dont...if you have a very complex application landscape, you need to implement all legal requieremetn in all system. If you have less system, tehn it easier and costless • It should be a driver, and if company are smart, they use it as a driver
[3.1.1]	**Become flexible to support changing business requirements**	
[1.11.4]	**Complicated system**	
[1.12.1]	**Knowledge become scarce**	• So then people start to scarce, but maybe not in the language [is such] but more on the content on that applcaition.
[1.11.4]	**Complicated system**	• since most of the products were very complecated, and tehrefore a quite a few point solutions...
[1.5.2]	**Functional appropriateness**	• is it functional supporting the current requirement?
[1.2.1]	**Robust**	• so it is robust for the future.
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• So that can be a reaosn why peole are a little bit less reluctant
[3.2.2]	**Reduce cost of maintenance and operation**	• ideally you have standard interface between each of these compartment of each functionality. And SOA helps with that
[3.1.4]	**Prone to failure**	• If you have separated functional areas, you can also separate maintenance. This is the way of managing complexity

14. P16

[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• end supported or end of life by the supplier.
[1.6.2]	**Adaptability**	• It's not compatible in our IT environment
[1.11.1]	**Multiple system**	• the fact that lot of time we replace applcaiton but the old application is still running
[4.2.1]	**Funding legacy modernization projects**	• Not enough budget
[1.10.1]	**High risk of running legacy system**	• big risk becasue you dont have any support of your supplier. • No..no...because of the risk and unsupportable. • If it going down, it is really going down. And it takes a lot of effort to get it up and make it runs again.
[1.2.2]	**Stable system**	• on legacy we dont do any changes anymore so sometimes it is stable
[1.12.1]	**Knowledge become scarce**	• We still have people with knowldege of legacy system. but what we see is that the knwledge is walking out because people are moving to other job. So it is getting harder and harder to get the knowledge • Lack of knowledge
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• but it is not good for my customers Resistance from the current users or maintainers in the organization.
[1.8.1]	**Not flexible**	• On of the other things why we want to get rid of the legacy because it is very..it is not flexible • It's regarding interoperability..hmm...it is not flexible... • one other point is that my costumer wants flexibility • legacy it is not flexible and the legacy systems are hidding some secret that we not always know.
[1.11.1]	**Large system**	• Because legcacy is rigid, is big,
[2.2.1]	**Expensive in maintenance**	• We now migrate it to dotNET and our lisence fee is drop for something like half
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• Sometimes we are the first example in the world
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• But also the state of mind of the people • And the awareness with our end users that we have legacy and we have to get rid of it
[4.1.1]	**Data migration**	• Mising data
[1.1.1]	**Modifiability**	• i think the system is unrepairable in the future because of the lack of the parts of the machinery and because of the lack of the knowledge regarding of the software.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• I sometimes think that if you have dedicated team. Dedicated focus on get rid of legacy, maybe it would help
[4.3.5]	**Time constraints to finish legacy modernization**	• the timeframe to test.
[3.2.3]	**Faster time-to-market product**	• short time to market
[1.6.1]	**Interoperability**	• It's regarding interoperability..hmm...it is not flexible...

15. P17

[1.9.1]	**Old system**	• it is old
[1.7.1]	**Unsupported supplier**	• there's not patches
[1.5.1]	**Functional completeness**	• It's not processing any more
[3.1.2]	**Limited supplier or vendor**	• Because you need support, you need update the patches, you need security patches, you need everything to get the stable situation

	to support changing business requirements	
[1.11.1]	**Large system**	• That was a really big
[1.9.2]	**Availability**	• It's proven technology most of the time
[1.2.2]	**Stable system**	• it was stable, it works almost always...hahaha...But it was quite good
[2.2.1]	**Expensive in maintenance**	• Yeah...actually higher, because it is not a standard anymore. • but the maintenance cost of the AS400 was sky high so we need to get rid of the system and really turn it out • That's right now. With legacy system, last year, we spent.. I guess 80% of legacy system just for keeping it running and 20% was for new functionality. So we change that figure. • . When you look at the cost of the system and new innovation if a system stop with innovation and the cost getting higher because maintenance getting more expensive, maybe you should think of change
[4.1.1]	**Data migration**	• so biggest problem was how to update those data? so how you get the good information and get the right information out of the system into the newer system and leave all the old system behind
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• bank is too big, you have to split it because otherwise the risk is too high if it is fall apart and the governance has to carry all the consequence • It was initiated by the change of ING. We have to change then start modernization. So there was in need for changing the modernization, there was a need for changing
[1.6.1]	**Interoperability**	• The technical difficult was all the subsystem of the legacy system. . So there was something for printing, there was something for reporting, there was some external connection for getting all the messages, there were so many system and connectivity around the legacy system.
[4.2.1]	**Lack of resource (e.g. documentation, expertise, etc)**	• if you are the first, I think you will have a risk • biggest risk is if you go to the new system that hasn't any proven technology or it is really new
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• we also change the culture of people and how they should work and report and everything. I think it will take 3 years to get that kind of transformation
[4.3.3]	**Communicate the reasons or consequences of modernization**	• One of them is communication and get business people involve in the project. let them see here and know what's you are doing. So I think a best success factor is you know [K-O-T-T-E-R] • we did a lot of communication. Several time a week we post everything need, what a new strategy was. And that was about communication.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• To convince them and get the migration and accept the new system. I think that's the hard part of it.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• also the business strategy itself change several time during the migration. So we also made some changes during the migration • ...I don't think that you really have some if you have a good plan. We already had a good structure, we already had a new system and we already knew what we are going to do. • I think first of all, you should start with the manager. if you give people much room to complain or do other things and you doesn't support them in how they should work and being an example. I think you have to being an example as a manger how you should work and how can work and if it isn't working, you are not start to complaining, you try to fix it and help
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• Actually one of them didn't apply to a job at my team, because he already saw he didn't have work anymore after the migration of AS400 • we do have a plan, we start but to do more training, do small implementations • No they're not satisfied. But I think in IT you don't really have satisfy end user because if the system doesn't work for 5 minutes, they are already frustrating because they need to do something in those 5 minutes.
[1.1.1]	**Modifiability**	• Upgrading and getting to the higher level for supporting the organization in your business, your legacy system is getting more difficult
[2.1.1]	**Doesn't fit with future strategy**	
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• Like we have a good system, why should you change the system when everyone is happy with it.
[1.11.2]	**Database issue**	• normally legacy system doesn't support very good databases or doesn't has relation database model • ...there was the data in the system, it was old, it was not update
[4.3.5]	**Time constraints to finish legacy modernization**	• we were in the time squeeze. We got only a few months to do the conversion and we didn't have any time to did very good research
[4.2.2]	**Predicting Return of Investment (ROI) of modernization**	• I think maybe the acceptance of the business people that they wont accept the newer system or hard to change.

16. P18

[1.12.1]	**Knowledge become scarce**	• nobody's really knows about it, but see there's reason for not to touch it • they don't know what are the hidden features behind or what the business rules • the problem with the language like C is that there's no... less knowledge in the market right now • It's not documented very well. • Documentation...yeah...for new system but also for the old systems. See can we reverse engineer the old system and document them well so we know what the requirements are and...Because then if you have the requirements, you can say well..we can build new system. • What we trying to do now is, Rijswaterstraat use the document standard the JSTD. It's a...right now it is a IEEE standard, but this is the old version coming from the defense industry but IEEE is a newer standard
[1.6.2]	**Adaptability**	• Yes because it doesn't work in windows. It works on windows 8 but not really... doesn't' really support all technology.
[1.7.1]	**Unsupported supplier**	• hardware that are not support anymore...environment...
[1.9.1]	**Old system**	• legacy system are system that were built long time ago.
[2.2.1]	**Expensive in maintenance**	• one is getting very expensive to maintenance the system • You can just say goodbye to few people. • From ECT is money. it is all money and to reduce the cost of this thing and maintaining the IT
[1.2.1]	**Robust**	• It's reliable.. • People know how to use it for long time...all the problem has been disappeared from it…so technical problems are usually are not there...
[1.5.2]	**Functional appropriateness**	• but it is working so [everyone so well] that's keep it that way because it is doing its things and
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• we have to spend so much time to do the extra things that are not in legacy system • There's we cannot take few people and just keep them there
[1.11.4]	**Complicated system**	• The one I know is pretty complex. The one I use for the IT contain terminal.
[1.6.1]	**Interoperability**	• But they have an old legacy system which is very hard to connect to that bus.
[1.3.1]	**User interface aesthetics**	• interface is pretty clear that's just a messages like XML but in the old version • Clean and clear interfaces
[4.3.3]	**Communicate the reasons or consequences of modernization**	• They have cargo, they have to take cargo from there to there. That's their view. Sometimes you have to drop something off and pick something up. But the government wants to use it for traffic management. So very another way of looking at the data.
[3.1.1]	**Become flexible to support changing business requirements**	• If something happen and then there's…they send new regulations...the one of the things are changing over the time the way they have to handle the danger goods. • And that's calculation is changing every 2 years • Dutch government wants to do is to promote the traffic by the river. And then off by the road, promote traffic jam etc • For the government, change in regulation,
[1.11.5]	**Poor quality of design and ecosystem**	• because the program is really hard-coded...it is not configurable that makes it difficult

[1.3.2]	**Learnability**	• So then usually takes few months for someone anew to learn the system or old system
[4.1.4]	**Difficult to extract business rules or knowledge**	• Yes...you can only do that if you know exactly what the thing does. That's the hard part. • we try to get the functional requirements very clear, what should the new system do
[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• not only because it is old technology
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• to make more use of the information the customers are given. Because in this process in translation, some information also lost. We don't do it, because we don't really know what to do with it, so just throw it away
[4.3.3]	**Communicate the reasons or consequences of modernization**	• so we really assure communication part with the end user • People who think they know better. Because they've read a book about SOA and wooo...it should be SOA • So there's a lot of confidence and emotion.
[4.2.1]	**Funding legacy modernization projects**	• The ROI is too long • And a lot of companies also say they have their return of investment. They say it can be 1 year and then it should be done. well..I don't think return of investment of 1 year can... • big financial given by the governments to make some modernization.

17. P19

[2.1.1]	**Doesn't fit with future strategy**	• My definition of legacy system is sytems an technolgoy that do not belong to your technolgoy strategy...strategic techonology goals
[1.6.2]	**Adaptability**	• very hard to keep it up to date, to make...to move that forward to newer versions • I only call legayc system if you are stuck in the technology that you dont wanna carry forward
[1.1.1]	**Modifiability**	• Now that is the legacy. You won't find anyone who can maintain that. • It's always new rules, regulations, common labour agreement, and thing that changed between taxes [already], so that's alwways somthing changes. So there legacy is really painful
[1.12.1]	**Knowledge become scarce**	• It's a product that we still maintain as we go forward but for us it is very hard to find new UNIFACE developer anymore. • But how many you can find that who can understand really crypthographic algorithm where youcould do all kind of bit shifting, hasing and those kind of things. That's very limited group. • undocumented feature or knowledge, things in the product that no one knows about or spesific feature that works forever and that have been maintained recently and the one that invented it has long left the company. • [Q] SO you think important for the company to educate people in COBOL nowadays? [A] Not yet, but it will be in 10 years. Because then you get [distration] when so much COBOL code stil around and you need people people to maintain it. And to do that forward
[1.5.2]	**Functional appropriateness**	• It works as expected. Even if has bugs, people know the bugs, people have all worked around. • because they they expectation are completely inline with what tehy are getting
[1.2.2]	**Stable system**	• Then good things about a product, a program that you dont maintain anymore is that stable, and it gets faster • So the perception is, it is becoming more stable. It's not actually more stable, because still bugs in it, but everyone knows about the funny things, how to fix that, how to work around it and it's predictable system to everyone.
[1.4.2]	**General Performance is good**	• So the performance as expected.
[1.4.1]	**Time behavior**	• Because hardware, mono-hardware alwasy get faster. • old code went as amazing speed on new hardware
[1.3.2]	**Learnability**	• Yeah..tehy know that they have to not touch certain key or if you are in that screen dont touch that button because then your screen will die. Or if it goes wrong just do this step and then you back on tract
[4.3.2	**Resistance from the current users or maintainers in the organization**	• and most people very happy.
[4.1.1]	**Data migration**	• YOu always have to suffer from migration, incomplete data, lost of data, bug still in the system, not know how it works • Big challenge is always data migration. • Mainframe you have those back decimal, how you store them on your databases, so that's all kind of spesific in the data migration that is hard
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• they like they safety zone and they only use the system because they have to get the job done and get how at 5 o'clock leave the office. go watch football..something like that • I think last one is a the human factors. Developers that dont wanna change
[3.1.1]	**Become flexible to support changing business requirements**	• if they come up wiht new kind of rules, of regulation and you have to come up with additional text then it has the deferent way of calculation, you have to go in and build new type of rule • say everyone has to pay this amount of text except for single mothers with children under 5, single retired people over the age of 50 or 70, people like this, peole like that
[1.7.1]	**Unsupported supplier:**	• if the technolgoy supplier is not support it anymore, then you also have a risk • so, even [XXX] stop suportting it, then you are stuck because then you get a new hardware..you dont have a driver anymore for the chipset or, so you have to figure out how can install the platform and install the application in the platform.
[2.2.1]	**Expensive in maintenance**	• SO prices gooes up, because they have, basically they have so much capasity and for maintaining it, it cos them thier money.
[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• where the product was at the end of the product life cycle • Continuity. Continuity is teh biggest issue that I have becuase if..and it can be..because you can find the developers anymore. you are stuck. if the technolgoy is not support anymore by a provider.
[4.1.4]	**Difficult to extract business rules or knowledge**	• its running code, so you can alwasy use the application and test it, to see what it does. SO from the functional point of view since it is a prototype product, it is not in house system that no one know about, it is just a common product, it is very clear on what it does for many costumers. so that helps
[1.11.5]	**Poor quality of design and ecosystem**	• but usually systems are so closely integrated and complex
[1.6.1]	**Interoperability**	• Technology incompatibility. So you have rack in mainframe and you have rack in windoes, and still rack script that one on mainframe will not always run on windows, because implementation, slightly diffenert

18. P20 and P21

[2.1.2]	**Business Critical**	• And we love our customer legacy system. Because as a company, we are successful because of all differences between legacy systems, because if all legacy system talk and communicate with the same protocol, then we wouldn't exist. • So the difficulty of supporting legacy is one of the cores of our business
[1.6.1]	**Interoperability**	• Because normally these systems are not talk each other
[1.1.1]	**Modifiability**	• Can we do maintenance on the system? And if it became difficult, then it is a legacy • There are people who know about it and interact with it, but not how to modify it. so we are very limited in ability to modify it and to add new functionality and we don't to that anymore
[1.10.1]	**Risk of running legacy system**	• it is always about risk right,? you have to balance risks. If it big risk to maintain in current state...well risk could be economical risk or technical risk...if the risk become so big, then we should consider it as a legacy • We also have legacy but we don't get rid of it, because it has no big risk yet. But we still consider it legacy.
[2.1.1]	**Doesn't fit with future strategy**	• Some of the engine we consider it legacy, but we still use them, we just don't enhance them anymore or add new feature to that
[1.11.1]	**Multiple system**	• We use other engine and they are newer generation engine before that. But we still maintain the old one... • For compatibility reason because in ecosystem around us, some parties is still using the old format
[1.11.4]	**Complicated system**	• There is so much rule and information in that engine
[4.1.4]	**Difficult to extract business rules or knowledge**	• To deal with all kind of exception. To work around and cooperate with other system. To extract all the rules and details in there is really difficult
[1.3.2]	**Learnability**	• If you come here as a new developer, it would be very very difficult to do anything inside the piece of the system
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• And extra handicapped is we are 24/7 highly reliable, up and running, in ecosystem. So it is very difficult in the situation like that to just take a piece out and refactor it and put the piece back in without any down time.

[4.3.5]	**Time constraints to finish legacy modernization**	• The legacy system still there and we are facing it out, but it is gonna take a long time.
[1.2.1]	**Robust**	• SO the good think is it is still running, it is working, it is petty reliable, pretty good • we keeping them there because it is performing well
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• The priority can change quickly. • how to find a good migration path
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• new opportunity, if you think the market changes, new message format or new way of connecting
[3.1.1]	**Become flexible to support changing business requirements**	• There is a government body called NICTIS, who decide the standard for messaging. And that's one driver if they upgrade the standard, then we have to migrate too
[1.12.1]	**Knowledge become scarce**	• one of the factors for maintainability is also documentation. So it that lacking, then it is much more difficult for new developers to learn how things work. a huge factor • if documentation is lacking then it is a bigger risk to migrate. Because you don't know what's going on in the old system naturally, make the risk to migrate bigger.
[1.7.1]	**Unsupported supplier**	• Whenever the platform the software run on is outdated, so we cannot then support what so ever, I mean our server level are very high
[4.2.2]	**Predicting Return of Investment (ROI) of modernization**	• who looks further than current version and who look at new techniques and new frameworks an new things like [SQRS] or big data or who looks further than current system, and spot to the technique that are usefull to migrate to
[4.3.3]	**Communicate the reasons or consequences of modernization**	• can everybody handle it? can we explain good enough? can people learn it? And to use new technology, we have to have skill to use it. we have to invest in it for a longer time to get the experience.
[4.1.5]	**Difficult to test**	• regression test that cover enough all the continuality

19. P22

[2.1.1]	**Doesn't fit with future strategy**	• Legacy is what I would get rid off. Legacy is kind of what hindering me
[1.8.1]	**Not flexible**	• No it is not flexible enough to do the things you wanna do • they cannot reponse to change request
[1.1.1]	**Modifiability**	• because you cannot attract people that they have to maintain it or it doesn't cooperate new kind of technology • I think they have to maintain to many knowladge area, COBOL , Java
[4.2.1]	**Funding legacy modernization projects**	• it is too expensive to replace them • Now I think 5 years later we still are not able to say that we replace the old system. I think we have estimation about 30-40million euros still to spend to get rid of the old crap.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• But that doesnt work. Because evry time the attention goes away from modernizing to functuinality. • Getting it done.. That's really our challenge. • Keeping pace between change demand that come from our environment, our fund, everybody...and innovation. innovation target you have
[3.1.1]	**Become flexible to support changing business requirements**	• The pension world is very much movements now, so it is very rapidly changing
[1.12.1]	**Knowledge become scarce**	• we can't find the people to keep it going
[3.2.3]	**Faster time-to-market product**	• we need a faster time to market, and we are not able to do that in COBOL envirionment • They have too high time to market
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• Yes... you train all kind of people. More than hundred people • Yes it is, beacause then you are decovering that not ebody is going to be able to use the new technolgy. • [Q] So it is difficult to get people to learn people to get use to the new environment [A] Yeap... yeah it is difficult
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• There's always people that cannot move to the new technologyies. But we dont have that problem because it takes so long that everybody in the old technolgy is really on the retirement day period
[3.1.2]	**Limited supplier or vendor to support changing business requirements / system continuity**	• Keep access to your resouerces
[4.3.3]	**Communicate the reasons or consequences of modernization**	• It's a supported from the top number one i think
[2.2.1]	**Expensive in maintenance**	• They are too expensive
[4.1.4]	**Difficult to extract business rules or knowledge**	• [Q] To understand you own system it is really difficult at the begining. [A] Yess...

20. P23

[1.9.1]	**Old system**	• software of almost 20 years, hardware of almost 20 years that is still working • In my opinion, legayc is old hardware, old software thta's still need support.
[1.7.1]	**Unsupported supplier**	• If system cannot longer be supported by software or hardware • Because machine that are not supported any longer is a great risk for business
[1.12.1	**Knowledge become scarce**	• so that's very dificult task, to maintain the knowledge about the software • We have couple of programmers around 45 - 50 and 1 even older 85, 58 where able to support those PASCAL software • [Q] Is that well documented? The old guy with PACAL... [A] Not everything
[1.6.2]	**Adaptability**	• when need to deliver support to old systems then you often confronted the fact that the part that are broken hardware or software, can not be exchange by the same part • and another thing is a it is also difficult task that the old system, the old hardware sytems need still to be supported in new software system
[1.2.2]	**Stable system**	• often the old technology is more stable than the new technology
[1.9.2]	**Availability**	• that can stay over a couple of years that way,so you can say that the old technology is proven technogly
[1.6.1]	**Interoperability**	• because old system doesnt have information exchange system
[3.1.2]	**Limited supplier or vendor to support changing business requirements**	• Other reason is when you have good running product which is well designed and almost at the end of its life cycle
[4.1.1]	**Data migration**	• It's more difficult when a lot of third parties are involved. SO client has machine from us, and warehouse is from supplier A and warehose management system from supllier B, etc... and what you see is that they seldomly at the same level. So one work with data exchange based on small text files, and they are vary each other. So that's always complicated, so always...data exchagne is allways need extra costumization

21. P24

[2.2.1]	**Expensive in maintenance**	• usually the cost of [get rid] of your legacy systme is exteremly prohibited • most company claim that they spent 80% of their budget on running and 20% of innovation.
[1.11.1]	**Multiple systems**	• somthing like 30 or 40 years of programming
[1.11.4]	**Complicated system**	• [Q] A lot of business rule, a lot of business logic inside of legacy system which make it difficult or even impossible to [A] yeah..this is

		so..make it difficult or nearly impossible to get rid of it
[2.1.2]	**Business Critical**	• rules are have been key for the survival for the company. because they have been there 30 years so they really the foundation for the survival of the organization • legacy system contain the key business rules for a business
[1.12.1]	**Knowledge become scarce**	• i think it is very difficult to get people from university and i aske them do you wnat to be a COBOL programmer ?
[1.4.2]	**General Performance is good**	• performance is never an issue in legacy system, at least i've never seen it
[3.2-Special]	**It's business, not IT decision**	• I dont think it make sense from technolgy perspective becuase almost all of it isworking • there's anoher priority for tomorrow. and you can repeat that same question for years and years and nothing will happen and that still wont break.so ...youve been saying that it will break within the next 4 months from the last 10 years. hahaha...
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• it is from those people that have built their career on that system. SO they 're not in the day to day operation.
[4.1.1]	**Data migration**	• somebody use a field that was designed to holds some information was miss-used to some other information
[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• yes...you rally need deep expertise. because they have been so many tricks has used in the past.
[1.10.1]	**Risk of running legacy system**	• they key problem is they go out of fashion.
[4.3.3]	**Communicate the reasons or consequences of modernization**	• [B] so it means that the support from board, the support higher position in the company also determine the successful and the failure of the modernizaiton. [E] Yeah.yeah...absolutely
[4.3.5]	**Time constraints to finish legacy modernization**	• make it as short as possible
[4.2.1]	**Funding legacy modernization projects**	• so to replace a system would probably take at least 10 times than normal maintenace cost. • So it is money but also being very clear that has to be happen
[1.5.2]	**Functional appropriateness**	• so an anything that is working, it is legacy.
[1.9.2]	**Availability**	• So if it working and proven then you could actaully consider it legacy
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• ...if some of the fundamental rules in the business change. and that could either be because they are entering a new business or they merge and they access to another sytem, that could reasons
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	

22. P25

[1.8.1]	**Not flexible**	• So then they need to rebuild the whole system to get more flexibility • we want more flexible and be able to change faster, be more adaptable • But it did its job, but less flexibility
[1.5.2]	**Functional appropriateness**	• But it did its job, but less flexibility • but normally they work • When you do high volume, mainframe is very good in it
[1.5.1]	**Functional completeness**	• there was no portal for citizens. So we need a lot of functionality and we build the new system and throw away the old system
[1.9.1]	**Old system**	• For the system we use, they are old and old technology
[1.6.2]	**Adaptability**	• we want more flexible and be able to change faster, be more adaptable
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• Higher demand also from the public to the system that [XXX] has. The public want immediately response.
[1.4.1]	**Time behavior**	• Higher demand also from the public to the system that [XX] has. The public want immediately response. • it works but it is not quite responsive system. • But if we want I/O intensive system, then we use mainframe since mainframe is really good for that kind of task. I/O in mainframe is really…really fast
[1.11.5]	**Poor quality of design and ecosystem**	• it is step-by-step process, all written in COBOL and they are connected to each other
[2.1.2]	**Business Critical**	• We want to change but it is hard since it is the core of our...all the people in the Netherlands are in it. We cannot afford if database corrupt or to make any changes that can affect the continuity
[1.7.1]	**Unsupported supplier**	• Before they were taken over, we were really worried about continuity. If the company go out of business, we really have a problem
[1.10.1]	**Risk of running legacy system**	• If we arrive at the point that we cannot guarantee the continuity, then we really need to change that system.
[3.1.1]	**Become flexible to support changing business requirements**	• We are government organization and there's also politic involve that in circumstance that we really need to do. And if we don't, then our higher chief, the minister has a problem
[2.2.1]	**Expensive in maintenance**	• We do know that we spend a lot of effort and money on maintenance
[1.12.1]	**Knowledge become scarce**	• I think it is going to be a problem for few futures, but i don't know what the plans are. I think age is a problem in [XXX]
[3.2.1]	**Create new business opportunity via mergers and acquisitions**	• Also creating business opportunity.
[3.2.3]	**Faster time-to-market product**	• It has to do with how fast can we deliver...time to market. if you want to shorten time to market then modern system should be easier and possible to realize it

23. P26

[1.9.1]	**Old system**	• It has some age. In [XXX] we have system over 30 years now and still using until now
[1.8.1]	**Not flexible**	• However, they are inflexible
[1.12.1]	**Knowledge become scarce**	• We have no documentation, or it is not updated • When they are not there we cannot help new employee to work with it. • You have to have people who understand the new technology
[1.1.1]	**Modifiability**	• So we are very afraid to make major changes to it
[1.5.2]	**Functional appropriateness**	• technology still working
[1.3.2]	**Learnability**	• not easy used by the employees
[1.6.2]	**Adaptability**	• It's a big problem, because a lot of software were created for the platform XP, doesn't work for platform window 7
[1.7.1]	**Unsupported supplier**	• If we continue then we don't get the security update.
[3.1.4]	**Prone to failure**	• However, once when are we modernizing it we are using different languages now which are more mainstream.
[2.2.1]	**Expensive in maintenance**	• [Q] Do you think that the maintenance of legacy system could be cheaper if you move legacy system to the new environment? [A] Yes. Now it is a lot higher. We have a budget every year for IT, more than 85% for just maintenance and 15% to do something new.
[1.4.2]	**General Performance is good**	• Yes... performance still OK. There's not a real problem.
[3.1.1]	**Become flexible to support changing business requirements**	• But it is not just migrating the system, but it also major change in legislation. • Yes...Law. So it is influence also by regulation
[4.3.2]	**Resistance from the current users or maintainers in the organization**	• [Q] So even people from old system not cooperate? [A] Yeah...

[4.1.2]	**Lack of resource (e.g. documentation, expertise, etc)**	• We hire people from [XXX], consultant, architect. It's too big for our regular IT staff to do it, so we're hiring a lot of experts from companies like [XXX], [XXX] to help design new system and help develop the new system, test it.
[4.3.1]	**Cultural resistance in organization not to adapt new system**	• Culture yes...people are used to do thing in a way, when you are changing that, well that's you have to think about it, how can we change the behavior about people
[4.3.3]	**Communicate the reasons or consequences of modernization**	• Yes...when they support it, they have to give time and money to go to education.
[4.3.4]	**Difficult to effectively prioritize the functionality for modernization**	• We are very optimistic. That's one of the lesson learnt to take your time.

Index codes of participants

[1.1.1] Modifiability
P1, P2, P6, P7 and P8, P9, P10, P11, P12, P13, P16, P17, P19, P20 and P21, P22, P26, P4 and P5

[1.2.1] Robust
P18, P20 and P21, P1, P6, P12, P15

[1.2.2] Stable system
P1, P2, P7 and P8, P16, P17, P19, P14, P23, P4 and P5

[1.3.1] User interface aesthetics
P6, P18

[1.3.2] Learnability
P1, P2, P6, P7 and P8, P19, P20 and P21, P18, P26, P4 and P5

[1.4.1] Time behavior
P2, P11, P19, P25

[1.4.2] General Performance is good
P1, P2, P7 and P8, P19, P22, P14, P24, P26

[1.5.1] Functional completeness
P1, P2, P6, P7 and P8, P9, P11, P17, P25, P12, P15

[1.5.2] Functional appropriateness
P1, P2, P19, P18, P25, P15, P24, P26

[1.6.1] Interoperability
P1, P9, P11, P17, P19, P20 and P21, P14, P23, P18, P15, P10, P16, P4 and P5

[1.6.2] Adaptability
P1, P2, P7 and P8, P9, P16, P19, P23, P18, P25, P26

[1.7.1] Unsupported supplier
P2, P7 and P8, P12, P17, P19, P20 and P21, P14, P23, P18, P25, P11, P26

[1.8.1] Not flexible
P1, P2, P9, P12, P13, P16, P22, P14, P25, P15, P6, P26

[1.9.1] Old system
P1, P2, P7 and P8, P9, P11, P13, P17, P14, P23, P18, P25, P6, P26, P4 and P5

[1.9.2] Availability
P6, P11, P1, P17, P14, P23, P24, P4 and P5

[1.10.1] Risk of running legacy system
P1, P12, P16, P20 and P21, P14, P25, P13

[1.11.1] Large multiple systems
P2, P10, P16, P20 and P21, P6, P1, P7 and P8, P9, P17, P24

[1.11.2] Database issue

P10, P12, P17, P6

[1.11.3] Back Office
P2, P7 and P8

[1.11.4] Complicated systems
P7 and P8, P9, P12, P20 and P21, P18, P15, P24

[1.11.5] Poor design and ecosystem
P1, P2, P7 and P8, P9, P10, P18, P15, P6, P11, P19, P14, P25

[1.12.1] Knowledge become scarce
P1, P2, P7 and P8, P9, P10, P11, P12, P13, P16, P19, P20 and P21, P22, P14, P23, P18, P25, P15, P24, P6, P26, P4 and P5

[2.1.1] Does not fit with future strategy
P1, P10, P11, P19, P20 and P21, P22, P14, P9

[2.1.2] Business Critical
P1, P9, P10, P11, P12, P14, P25, P24, P6, P2, P7 and P8, P11, P20 and P21

[2.2.1] Expensive in maintenance
P1, P2, P7 and P8, P9, P10, P11, P12, P13, P16, P17, P22, P14, P18, P25, P24, P6, P4 and P5, P26

[3.1.1] Become flexible to support changing business requirements
P1, P2, P9, P12, P13, P16, P22, P14, P25, P15, P6, P26

[3.1.2] Limited supplier or vendor to support the continuity
P1, P7 and P8, P9, P16, P17, P19, P22, P23, P18, P4 and P5

[3.1.3] Lack expert around systems
P1, P2, P7 and P8, P9, P10, P11, P12, P13, P16, P19, P20 and P21, P22, P14, P23, P18, P25, P15, P24, P6, P26, P4 and P5

[3.1.4] Prone to failure
P12, P16, P20 and P21, P14, P25, P7 and P8, P1, P13, P15, P26

[3.2.1] Create new business opportunity via mergers and acquisitions
P2, P10, P16, P18, P25, P13, P17, P24

[3.2.2] Reduce cost of maintenance and operation
P1, P2, P7 and P8, P9, P10, P11, P12, P13, P16, P17, P22, P14, P18, P25, P24, P6, P4 and P5, P26, P15

[3.2.3] Faster time-to-market product
P2, P7 and P8, P16, P22, P14, P25

[4.1.1] Data migration
P2, P10, P11, P12, P16, P17, P19, P14, P23, P15, P24, P6

[4.1.2] Lack of resource
P2, P9, P11, P13, P15, P24, P6, P26, P16, P17

[4.1.3] Poor system architecture or infrastructure
P1, P2, P7 and P8, P9, P10, P18, P15, P6, P11, P19, P14, P25

[4.1.4] Difficult to extract business rules or knowledge
P1, P12, P19, P20 and P21, P22, P18, P14

[4.1.5] Difficult to Test
P12, P13, P15, P20 and P21

[4.2.1] Funding legacy modernization projects
P1, P2, P7 and P8, P9, P12, P13, P16, P22, P18, P15, P24, P6

[4.2.2] Predicting Return on Investment (ROI) of modernization
P9, P20 and P21, P10, P11, P17, P14

[4.3.1] Cultural resistance in organization not to adapt new system
P11, P16, P22, P26, P12, P16, P17, P19, P14, P6

[4.3.2] Resistance from the current users or maintainer in the organization
P1, P9, P17, P22, P15, P24, P6, P26, P4 and P5, P7 and P8, P12, P13, P19, P18

[4.3.3] Communicate the reasons or consequences of modernization
P1, P2, P11, P12, P17, P20 and P21, P18, P6, P9, P10, P13, P22, P14, P15, P24, P26, P16

[4.3.4] Difficult to effectively prioritize the functionality for modernization
P2, P7 and P8, P9, P11, P12, P13, P17, P20 and P21, P22, P14, P26, P10

[4.3.5] Time constraints to finish legacy modernization
P1, P2, P7 and P8, P9, P12, P13, P16, P17, P20 and P21, P15, P24, P6

Interview Protocol

Title of research **: Revisiting legacy modernization from the industrial perspective**
Researcher : Belfrit Victor Batlajery – 3423891
Master Business Informatics
Institute of Information and Computing Science
Utrecht University
Supervisor : dr. S. L. R. Jansen
Daily Supervisor : Ravi Khadkha

Preliminary

Throughout decade the evolution of technology information has reached at the point where modification, maintenance and development are complicated and difficult. The fact that many organizations still rely on legacy system has motivated this research to find out how legacy system are viewed from industrial perspective and their modernization towards a new platform such as service-oriented architecture, cloud computing, etc. The interview will tend will tend to focus on Legacy System and Modernization.

The interview takes about one hour and some of the sample questions are presented below. The interview will be recorded because I don't want to miss any of your comments. Furthermore, we assure you a complete confidentiality and a report of our finding.

The information you provide in this interview will be used mainly to construct a new insight from industrial perspective about legacy systems and modernization for research purpose.

Participant Background Information

Name	
Recent organization	
Current Position	
Primary Function	
Work Experience (yrs)	
Expertise	
Age	

Interview Questions

Part I: Legacy System

1. What is your definition of a legacy system?
2. What are the important characteristics of legacy system?
3. Is there any problem caused by a legacy system? How do you handle the problems?
4. What are the good things about legacy systems?
5. Can you give me the example (languages, application, etc) do you consider legacy and why?
6. Could you give me an indication of how much it cost for maintaining legacy system? And could you elaborate more in what extend the money will be used for?

Part II: Modernization

1. What are methods you know for Modernization?
2. What are factors/reasons you know driver Modernization?
3. What are the challenges of doing Modernization? Can you rank them (1 being not challenging; 5 being extremely challenging)
4. Please describes strategies or tactics you had employed to overcome the obstacles or challenges?
5. During the Modernization, what are the factors determine successful/failure of Modernization?

6. Could you give me an indication of how much it cost for legacy modernization? And could you elaborate more in what extend the money will be used for?

Part III: Extra Question

1. If your organization has used, is using, or is currently considering legacy system modernization practices or processes that have not been addressed in this survey, or you consider an improvement or innovation to legacy system modernization practices, please describe the process or methodology below.

Participant's feedback

Please describe any unusual circumstances and/ or events that had any bearing on the interview such as interruptions, language difficulty, etc.:	
What do you think need to be improved in order to make the interview become better?	
Additional comments:	
Interviewer Qualifications	□Knowledgeable □Structuring □Clear □Gentle □Sensitive □Open □Steering □Critical □Remembering □Interpreting

Researcher / Interviewer **Participant / Interviewee**

Belfrit Victor Batlajery **…………………………………..**

Survey Questions

Legacy Modernization Survey

Thank you for your time to participate in this survey!
Utrecht University is conducting a research on legacy system and legacy modernization aiming at understanding how legacy systems and legacy modernization are perceived in industry. The result is expected to enrich the knowledge about legacy systems from an industrial perspective.

We, therefore, kindly request you to participate in this questionnaire. Completing this questionnaire will approximately take 10-15 minutes. The questionnaire is divided into 3 sections with 17 questions in total: Section 1 includes general questions about you and your organization; Section 2 consists of questions about legacy system; and Section 3 questions about legacy modernization project.

LINK TO THE SURVEY

If you have any questions or concerns please do not hesitate to email us at legacysystemstudy@gmail.com.
We sincerely hope that you will consider participating in this survey.

On behalf of the ServiciFi team,
B. V. Batlajery [b.v.batlajery@students.uu.nl]
Drs. R. Khadka [R.Khadka@uu.nl]
Dr. Slinger. Jansen [slinger.jansen@uu.nl]

I. Personal Information

Universiteit Utrecht

All of your responses will be treated as confidential. The results of the survey will be used for research purposes.

While participating in this survey, please always relate the answers to your experience with legacy systems and/or legacy modernization projects you have taken part in.

1. Please add your personal information.

* Country of work

* Experience (No. of years in IT)

2. Do you have experiences with systems that are considered legacy within the organization?

● Yes ○ No

If Yes, please indicate your experience with legacy systems in years.

3. Please specify the domain of your organization.

□ Academic Institution	□ Telecommunication Institution	□ Software Security
□ Consulting Company	□ Software Development Company	□ Service Provider
□ Health-Care Institution	☑ IT Research Institution	□ Financial Institution
□ Government organization		

Other :

4. What is your role in the organization?

□ Researcher

□ Developer

□ System Analyst

□ Business Analyst

□ IT manager

□ Chief Information Officer

Other (specify)

II. Characteristics of legacy systems

Universiteit Utrecht

Reliable system: Degree to which a system performs specified functions under specified conditions for a specified period of time; includes robustness, stable, high availability sub-characteristics [ISO/IEC 9126].
High performance: The capability of a system to process a given amount of tasks in a determined time interval; includes response time sub-characteristic.
Proven technology: Systems that have been demonstrated or verified without doubt to comply business requirements for long period of time.
Business critical: Systems whose disruption or malfunctioning will cause a failure in business operations. The business critical systems contain vital features/functionalities and their failure can result on serious financial & legal problems, damages and other penalties.

5. What do you think are defining characteristics of legacy systems?

☑ Reliable system

□ High performance

□ Proven technology

□ Business critical

Others :

6. Please rank on a scale of 1-5 (1-being least critical; 5-being most critical) the following problem(s) around legacy systems that you have experienced.

	1	2	3	4	5
Lack of documentation	○	○	○	○	○
Lack of experienced manpower	○	○	○	○	○

Limited suppliers/vendors to support and maintain	□	□	□	□	□
Unable to adequately support, maintain, or enhance inhouse	□	□	□	□	□
Incompatible with current and/or future technological environments	□	□	□	□	□
Too rigid to comply with new business requirements	□	□	□	□	□
Monolithic architecture	□	□	□	□	□
High risk of failure	□	□	□	□	□
Poor user interface	□	□	□	□	□
Too costly to maintain	□	□	□	□	□

7. Do you consider the programming language as one of the factors to decide if a system is legacy?

□ Yes □ No

8. Can you give some examples of programming languages that you consider a legacy or are being used as a legacy in your organization?

III. Legacy modernization projects

Universiteit Utrecht

Definition:
Legacy Modernization is a process of migrating / evolving a software system to a new target system.

9. Have you ever been involved in legacy modernization?

□ Yes □ No

10. Please rank which of the following driver(s) are the motivating factors to modernize legacy systems.

	Very weak	Weak	Strong	Very Strong
Become flexible to support changing business requirements	□	□	□	□
Create new business opportunities via mergers & acquisitions	□	□	□	□
Reduce the cost of maintenance & operations	□	□	□	□
Limited suppliers/vendors to support legacy systems	□	□	□	□

Lack of experts around legacy systems	○	○	○	○
Prone to failures	○	○	○	○
Faster time-to-market of product	○	○	○	○

Others :

11. How challenging are the following obstacles in a legacy system migration projects?

	Not Challenging	**Less Challenging**	**Quite Challenging**	**Challenging**	**Very Challenging**	**I don't know**
Data Migration	○	○	○	○	○	○
Lack of resources (e.g. documentation, experts)	○	○	○	○	○	○
Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)	○	○	○	○	○	○
Difficult to extract business rules/knowledge	○	○	○	○	○	○
Difficult to test	○	○	○	○	○	○
Resistance from the current users/maintainers in the organization	○	○	○	○	○	○
Cultural resistance in organization not to adapt new system	○	○	○	○	○	○
Difficult to effectively prioritize the functionality for modernization	○	○	○	○	○	○
Difficult to communicate the reasons/consequences of modernization	○	○	○	○	○	○
Funding legacy modernization projects	○	○	○	○	○	○
Time constraints to finish legacy modernization	○	○	○	○	○	○
Predicting Return of Investment (ROI) of modernization	○	○	○	○	○	○

Other :

12. Please indicate which of these factors are responsible for postponing legacy modernization. (Please check all that apply)

- □ Huge investment
- □ Less changes in core business process
- □ Difficult to find preferable / customize software
- □ No profit made during modernization project
- □ New system might behave differently
- □ Loss of clients
- □ Resistance from the current users/maintainers in the organization
- □ Downtime of the system during modernization

□ Do not fix a system, until it is broken

Others :

*13. Do you use any academic resources while performing legacy modernization projects?
(for example: research articles, tools, etc)

○ Yes

○ No

14. What resources do you use from academia?

□ Research articles

□ Research software tools

□ Research collaboration with academics

□ Internship students

Other (specify)

15. If your answer to Question 13 (Do you use any academic resources while performing legacy modernization projects?) is No, please specify some reasons.

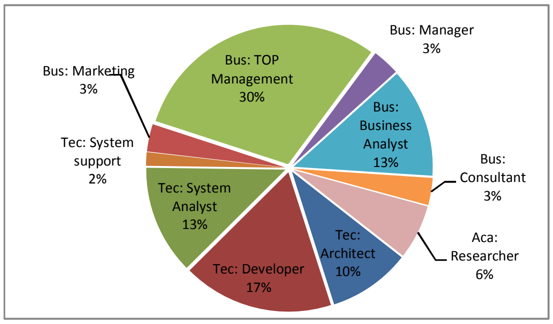
16. How can academics play a larger role in industrial legacy modernization projects?

17. Please provide your email below if you would like to get the result of the survey.
(We will not use your email for advertisement neither will distribute it to third parties.)

Survey data and statistic

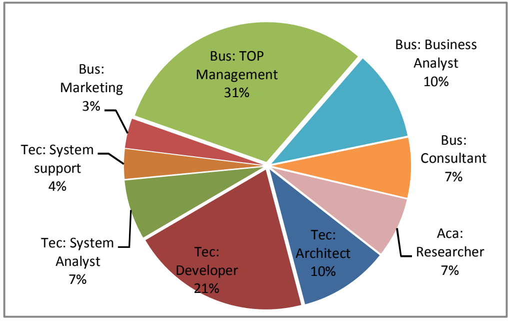
Positive characteristics of legacy systems

Reliable



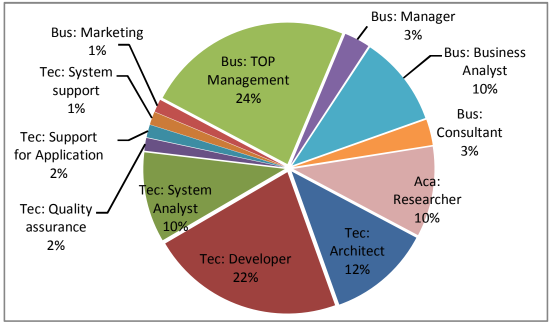
Business Domain (52%); Technology domain (42%); Academic domain (6%)

High Performance



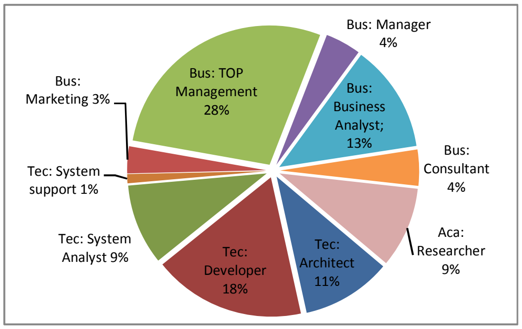
Business Domain (51%); Technology domain (42%); Academic domain (7%)

Proven Technology



Business Domain (41%); **Technology domain (49%)**; Academic domain (10%)

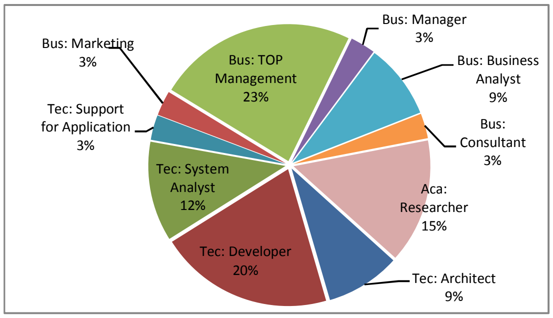
Business Critical



Business Domain (52%); Technology domain (39%); Academic domain (9%)

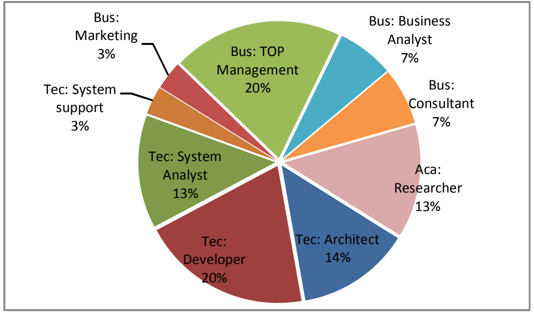
Problems of legacy systems

Lack of documentation



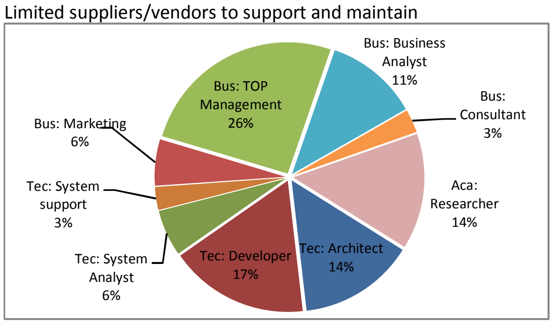
Business Domain (41%); **Technology domain (44%)**; Academic domain (15%)

Lack of experienced manpower



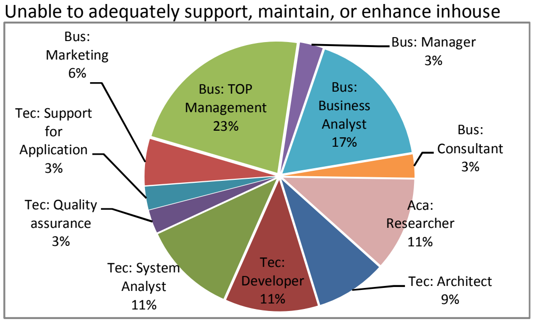
Business Domain (39%); **Technology domain (50%)**; Academic domain (13%)

Limited suppliers/vendors to support and maintain



Business Domain (46%); Technology domain (40%); Academic domain (14%)

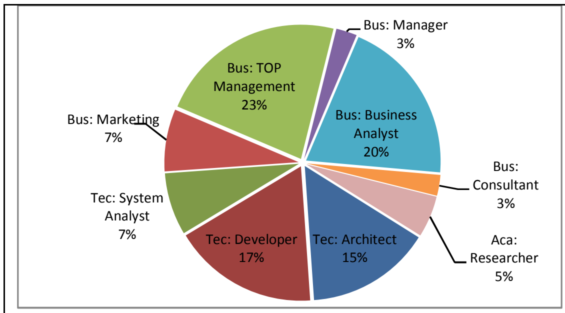
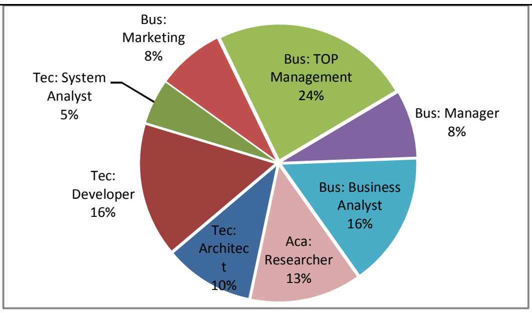
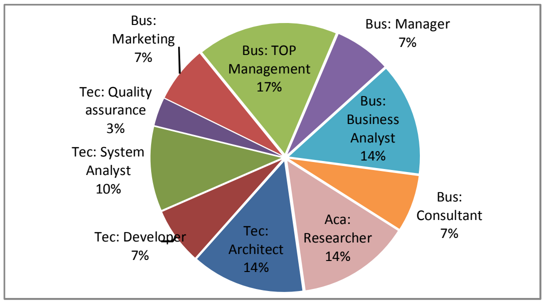
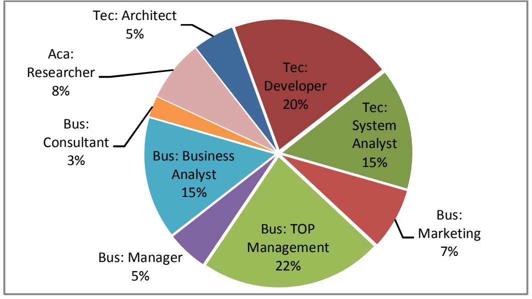
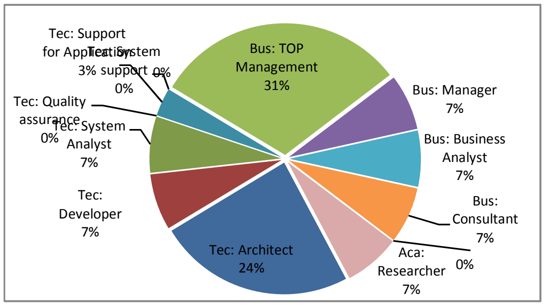
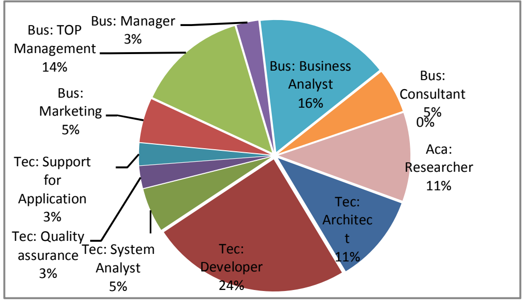
Unable to adequately support, maintain, or enhance inhouse



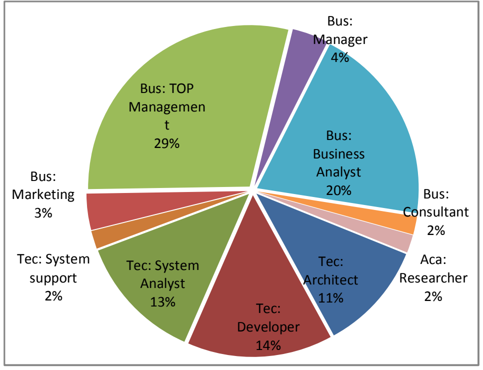
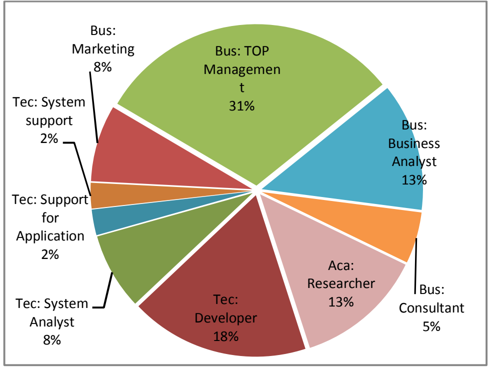
Business Domain (52%); Technology domain (37%); Academic domain (11%)

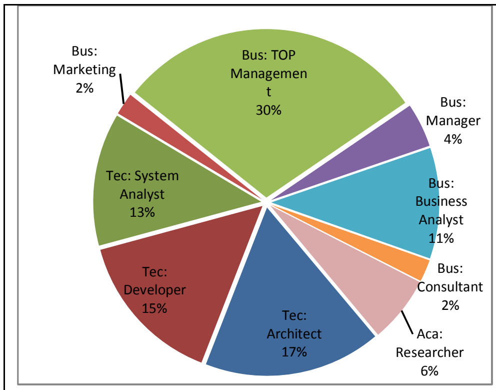
Incompatible with current and/or future technological environments

Too rigid to comply with new business requirements

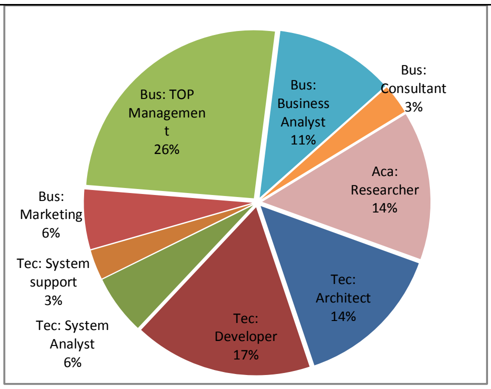
	
Business Domain (56%); Technology domain (39%); Academic domain (5%)	**Business Domain (56%)**; Technology domain (31%); Academic domain (13%)
Monolithic architecture	High risk of failure
	
Business Domain (52%); Technology domain (34%); Academic domain (14%)	**Business Domain (52%)**; Technology domain (40%); Academic domain (8%)
Poor user interface	Too costly to maintain
	
Business Domain (52%); Technology domain (7%); Academic domain (41%)	Business Domain (43%); **Technology domain (46%)**; Academic domain (11%)

Drivers toward legacy modernization project

Become flexible to support changing business requirements	Create new business opportunities via mergers & acquisitions
	
Business Domain (58%); Technology domain (40%); Academic domain (2%)	**Business Domain (57%)**; Technology domain (30%); Academic domain (13%)
Reduce the cost of maintenance & operations	Limited suppliers/vendors to support legacy systems

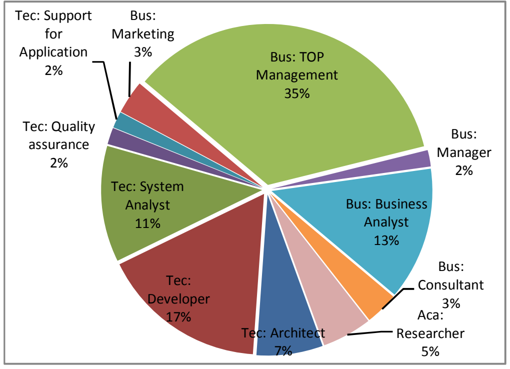


Business Domain (49%); Technology domain (45%); Academic domain (6%)



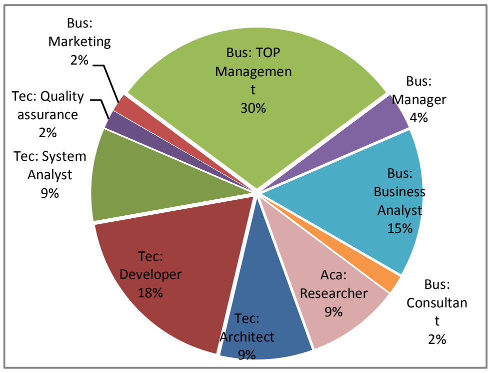
Business Domain (46%); Technology domain (40%); Academic domain (14%)

Lack of experts around legacy systems



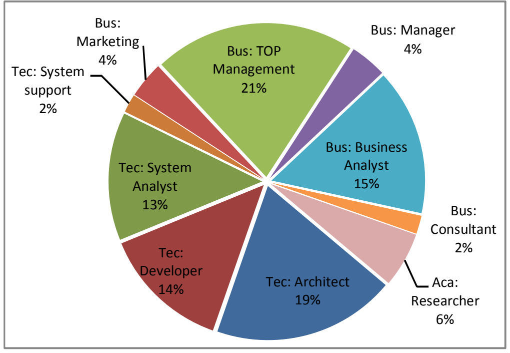
Business Domain (56%); Technology domain (39%); Academic domain (5%)

Prone to failures



Business Domain (53%); Technology domain (38%); Academic domain (9%)

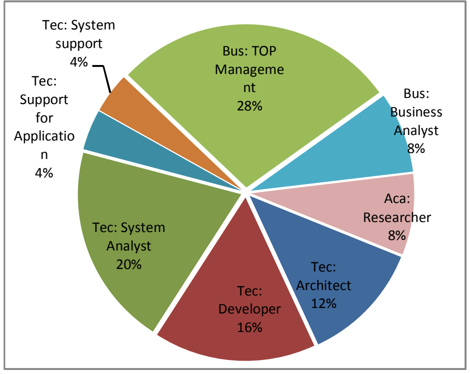
Faster time-to-market of product



Business Domain (46%); **Technology domain (48%)**; Academic domain (6%)

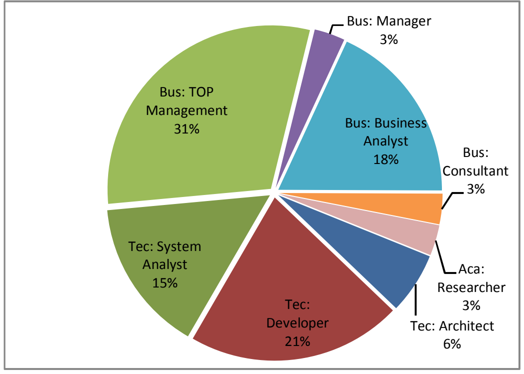
Challenges in legacy modernization project

Data Migration



Business Domain (36%); **Technology domain (56%)**; Academic domain (8%)

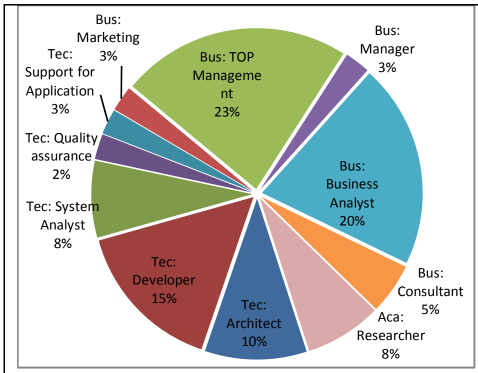
Lack of resources (e.g. documentation, experts)



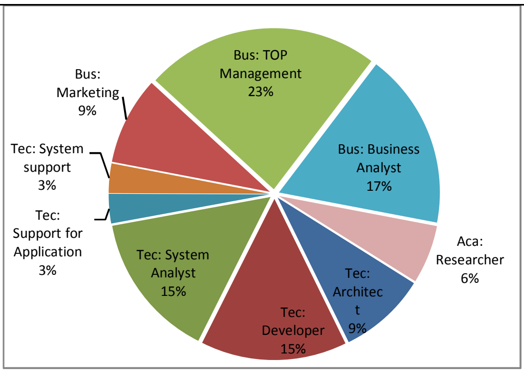
Business Domain (55%); Technology domain (42%); Academic domain (3%)

Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)

Difficult to extract business rules/knowledge



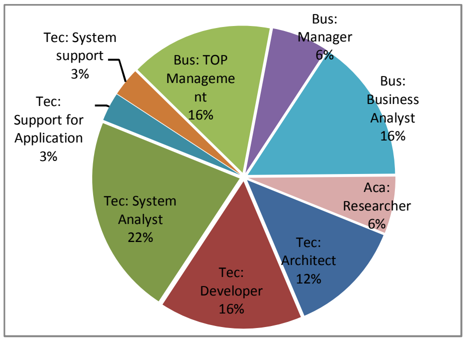
Business Domain (54%); Technology domain (38%); Academic domain (8%)



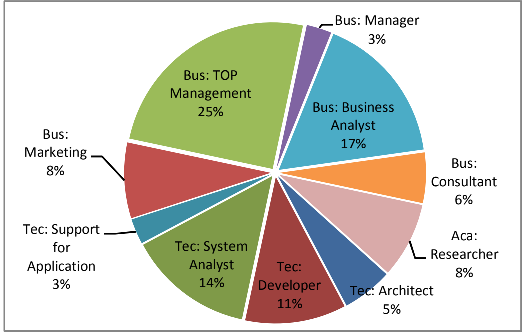
Business Domain (49%); Technology domain (45%); Academic domain (6%)

Difficult to test

Resistance from the current users/maintainers in the organization



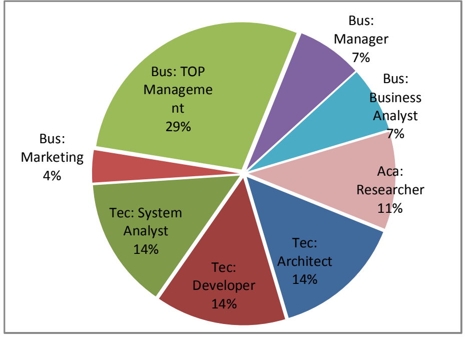
Business Domain (38%); **Technology domain (56%)**; Academic domain (6%)



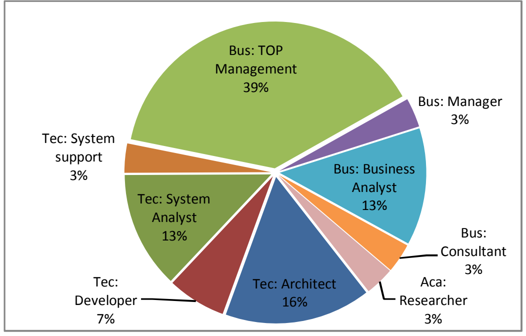
Business Domain (59%); Technology domain (33%); Academic domain (8%)

Cultural resistance in organization not to adapt new system

Difficult to effectively prioritize the functionality for modernization



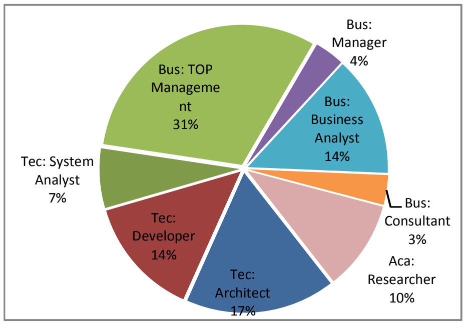
Business Domain (47%); Technology domain (42%); Academic domain (11%)



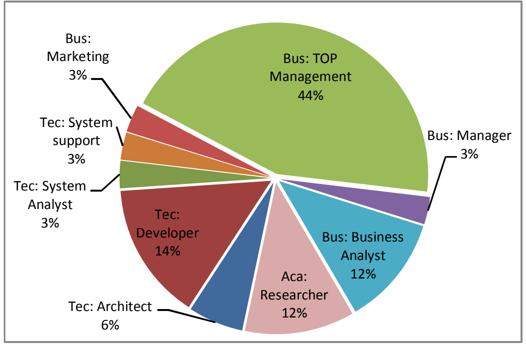
Business Domain (58%); Technology domain (39%); Academic domain (3%)

Difficult to communicate the reasons/consequences of modernization

Funding legacy modernization projects



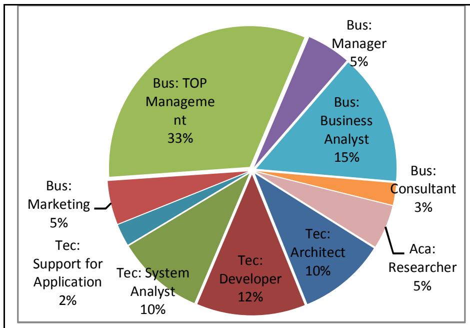
Business Domain (52%); Technology domain (38%); Academic domain (10%)



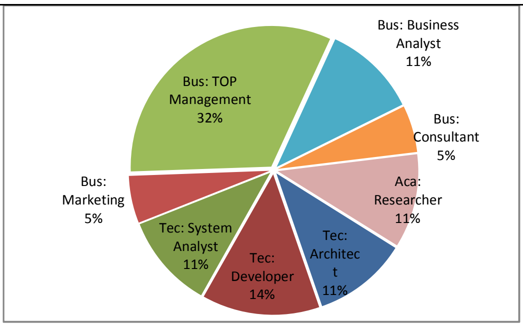
Business Domain (62%); Technology domain (26%); Academic domain (12%)

Time constraints to finish legacy modernization

Predicting Return of Investment (ROI) of modernization



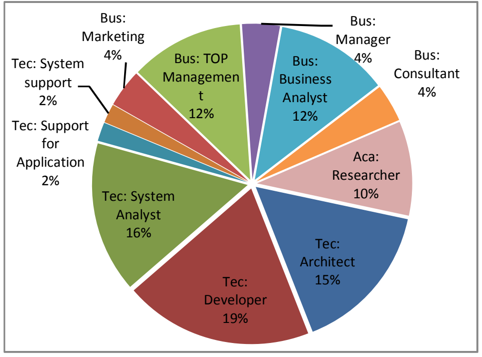
Business Domain (61%); Technology domain (34%); Academic domain (5%)



Business Domain (53%); Technology domain (36%); Academic domain (11%)

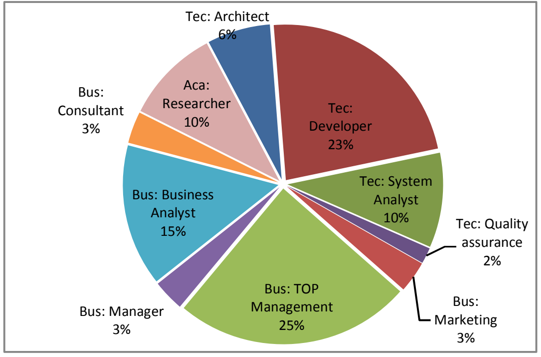
Programing Language consideration

Yes for legacy programming



Business Domain (36%); **Technology domain (54%)**; Academic domain (10%)

No for legacy programming



Business Domain (49%); Technology domain (41%); Academic domain (10%)