Industrial Perception of Legacy Software System and their Modernization

B. V. Batlajery

R. Khadka

A. M. Saeidi

S. Jansen

J. Hage

Technical Report UU-CS-2014-004
Feburary 2014

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl
Industrial Perception of Legacy Software Systems and Their Modernization

Belfrit V. Batlajery, Ravi Khadka, Amir M. Saeidi, Slinger Jansen, and Jurrian Hage
Utrecht University, Utrecht, The Netherlands
{b.v.batlajery, r.khadka, a.m.saeidi, slinger.jansen, j.hage}@uu.nl

Abstract

Research in legacy system modernization has traditionally focused on its technical challenges, and takes the standpoint that legacy systems are obsolete systems, yet they are crucial for an organization's operation. Nonetheless, it remains unclear whether practitioners in the industry also share this perception. This paper describes the outcome of an exploratory study in which 26 practitioners from the industry are interviewed on what makes a software system a legacy system, what the main drivers are leading to the modernization of such systems, and what challenges are faced during the modernization process. The findings of the interviews have been validated by means of a survey with 198 respondents. As it turns out, practitioners value their legacy systems highly, the challenges they face are not just technical, but also include business and organizational aspects.

1. Introduction

After three decades of legacy modernization research, it is surprising to find that legacy systems are still in daily operation. A legacy system is any software system that significantly resists modification but are business critical, and hence, their failure can have serious impact on the business [1]. Most of these systems were developed years ago, and they are continually evolving. With changing requirements frequent, modifications within these legacy systems have resulted in unstructured source code, which is difficult and costly to maintain. Furthermore, the knowledge about those legacy systems are scarce as the original programmers leave the company or retire, and up-to-date documentation is usually lacking [2]. These issues have been recognized by the software engineering community and a plethora of legacy modernization approaches have been proposed, primarily, to reduce maintenance costs, and to increase flexibility. Despite the issues of legacy systems and acclaimed benefits of legacy modernization, technology consulting firms estimate that 180-200 billion lines of legacy code are still in active use for daily business [2]. This discrepancy has triggered us to investigate legacy systems and their modernization from an industrial perspective. We aim at identifying what are good characteristics of legacy systems, what bad characteristics of legacy systems are the drivers for legacy modernization and what challenges are faced in industry while modernizing legacy systems.

We set up our research as an explorative study with an aim at discovering new perspectives and insights about legacy systems in industry, instead of starting out with predefined hypotheses and testing them. Therefore, we used grounded theory [3], which is increasingly popular in software engineering research [4], to conduct and analyze semi-structured interviews with 26 practitioners from industry with an experience in legacy systems and legacy modernization. The findings of the interviews were validated through a separate structured survey with 198 respondents.

This report has the following contributions:
1. We document the industrial perception of legacy systems and their modernization.
2. We identify the perceived benefits of the legacy systems, drivers of modernization, and challenges that the industry faces during modernization.
3. We report the perceptual differences of and about legacy systems between the industry and academia.
The paper is structured as follows. In Section 2, we present the related work; in Section 3, the research method is detailed, in particular the GT approach; in Section 4, research design is explained; in Section 5, the research execution is presented; in Section 6 the research findings are explained; in Section 7, the validation using the results of the survey is presented with some statistical details used. Finally, the report concludes in Section 9.

2. Related Work

Brodie & Stonebraker [5] in their book describe legacy systems as ``any systems that cannot be modified to adapt to constantly changing business requirements and their failure can have a serious impact on business.'' and reported various characteristics of the legacy systems such as mission critical, hard to maintain, inflexible and brittle. In [6], Bennett described legacy system as ``large software system that we don't know how to cope to cope with but that are vital to our organization''. Bisbal et al. [1] enlisted several problems of legacy systems such as legacy systems run on obsolete hardware, expensive to maintain, lack of documentation and understanding of system, difficult to extend and integrate with other systems. From the aforementioned definitions and characteristics of legacy systems, it is evident that legacy systems are presented as a serious problem in academia. The current study, in contrast, takes different approach to identify the characteristics that still keep them operational in industry. Nevertheless, the study also explores various issues of the legacy systems that complement the issues identified in the academia. In fact, we explicitly identified overlaps between the issues of legacy systems in academia and in industry such as high maintenance cost, lack of resources, achieve flexibility. In this research, we have presented those issues as drivers for legacy modernization.

Within 2 decades, a plethora of legacy modernization approaches have been reported. We refer to systematic literature studies [7-9] for further details. With respect to legacy modernization challenges, Brodie [10] enlisted various technical challenges of legacy modernization that are more influenced by academic research experiences. Van Deursen et al. [11] presented an overview of techniques to facilitate legacy modernization and the issues of modernization, particularly aiming at identifying objects. The legacy modernization research and the challenges identified in the academia are largely technology oriented. They provide different techniques/methods to facilitate legacy modernization and point out various challenges faced in the course of applying those techniques/methods. Our research not only identified various business issues, but also confirmed the technical observations made by researchers.

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. The result will be compared with academia perspective. Two research methods are applied in this research, being interview and survey. 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.

The overall research model is depicted in Figure 1.
In this research, we aim at exploring new perspectives and insights about legacy systems. Thus, this research adopts exploratory research using Grounded Theory (GT) as qualitative research through the process of interview. The 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. We construct multiple questions based on his knowledge, literature study, survey’s fact, and input from experts in order to get better quality of information from the 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 interviewees beforehand. In the interview protocol, the research motivation is introduced briefly and then question regarding the information regarding the interviewee is stated. Some representative questions about legacy system and legacy modernization are also stated in the protocol.

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. By using Grounded theory, we try to get the new perspective and insight by exploring the data received from interview to grow his insight.

First step in GT is conducting the interview and record it. 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(s) and the interviewee(s) will have a conversation in English for about 60-90 minutes depends on information flowing between them.

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 processes are not only the process of labeling, but also the process of leading the raw data to the idea and link them together. From coding the author can also identify the
pattern such as similarity, difference, frequency, sequence, correspondence, or causation. The codes are organized hierarchically into concepts and at the end will be grouped into categories.

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. We transcribed the recorded interviews and read the transcribed documents carefully on what our participants said and try to understand what does it means. After that we try to present our idea on why does that matter in our research. 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. Upon reaching the saturation stage - a condition that there is no additional information is being found with new interviews.

We used an online survey to triangulate the finding of the interview, to which 198 responded. 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.

Our survey contains 17 questions that capture the demographic background of the participants, questions about legacy systems and also about legacy modernization. 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. All of them can be completed in approximately 10 minutes and was designed accordingly.

4. Research Design

In total, there were 23 interviews with 26 participants who can provide relevant information on legacy systems and legacy modernization from various organizations across the Netherlands. We started the current research with a series of interviews conducted with 26 practitioners (identified as P1-P26 in this paper), each lasting 1-2 hours. The informants were selected based on two criteria: (i) have experience with legacy systems, and (ii) have experience with legacy modernization projects. The informants were identified opportunistically, via industrial collaborators, followed by snowball samplingkitchenham2002principles, in which the first generation informants help to identify other informants fulfilling the criteria. In total, 23 interview sessions were performed. In three of the interview sessions, the interview was conducted with two informants from the same organization, and additionally, two practitioners were from a same company, reducing the sample size of the participating organizations to 22. The sample is arguably broad to represent the professional software engineering, including diversity among the domain of the industries and variations among the informant’s roles and experiences. Table 1 depicts the details of the informants with the domain of their company. The company sizes ranged from consulting firms to global corporations such as IBM, Deloitte, Capgemini. The variation among the informant’s roles is also broad, ranging from software developers, to system analysts, consultants, software architects, business architects, research and development managers, and Chief Information Officers (CIOs). The experiences of the informants range from 5 years to 43 years, with >19 years as an average experience of the sample and cumulatively, the informants have >490 years of experiences in information technology. Additionally, the data totaled 25 hours of recorded data.
We conducted semi-structured interviews in person in English; these were recorded. Prior to the interview session, informants were provided with an interview protocol with sample questions to be discussed in the interview sessions. The interview session consists of three categories of questions (i) about characteristics of legacy systems, (ii) drivers for legacy modernization, and (iii) challenges faced during modernization. Afterwards, the recorded interviews were transcribed and each interview transcript was analyzed through coding: a process of breaking up the interviews into smaller coherent units, and adding codes to these units. Subsequently, a process of writing down narratives that explain the ideas of the evolving theory, known as memoing, was used to develop the coding. These coherent units represent key characteristics of the interview being analyzed. Later, the codes were organized into concepts, which in turn were grouped into categories. When the interviews being analyzed progressively provided answers similar to the earlier ones, a saturation stage was observed. To confirm the saturation stage, we conducted two more interviews and found that the analysis resulted in similar responses to the earlier ones. We used Nvivo 10 as an instrumentation tool to facilitate the interview analysis process.

The second and final phase of this research aimed at validating the results. We adopted a structured survey as a data triangulation process to validate the findings of the interview results. A data triangulation process, using more than one data source or collecting the same data at different occasions, is typically used to increase/decrease confidence in a finding by confirming or contradicting evidence and helps to improve

---

1 www.qsrinternational.com/
validity of the findings of an empirical study such as this. The survey was announced via mailing lists, social media such as Twitter, LinkedIn, Facebook, and via personal referrals. Listing 1 depicts the groups that we used in LinkedIn to spread our survey.

Listing 1. LinkedIn groups used for distributing survey.

<table>
<thead>
<tr>
<th>Group Name</th>
<th>Group Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-SCAM</td>
<td>Software Improvement Group</td>
</tr>
<tr>
<td>Legacy migration</td>
<td>Centrum Wiskunde &amp; Informatica - Employees &amp; Alumni of CWI</td>
</tr>
<tr>
<td>Legacy</td>
<td>TCS - Tata Consultancy Services</td>
</tr>
<tr>
<td>modernization</td>
<td>and Mathematisch Centrum</td>
</tr>
<tr>
<td>TOGAF</td>
<td>Application Modernization and</td>
</tr>
<tr>
<td>Mainframe Migration</td>
<td>IBM Enterprise Modernization Professionals</td>
</tr>
<tr>
<td>Amadeus alumni</td>
<td>IBM Mainframe Migration</td>
</tr>
<tr>
<td>Shell Australia IT</td>
<td>Centrum Wiskunde &amp; Informatica - Employees &amp; Alumni of CWI</td>
</tr>
<tr>
<td>Network</td>
<td>Employees &amp; Alumni of CWI</td>
</tr>
<tr>
<td></td>
<td>and Mathematisch Centrum</td>
</tr>
</tbody>
</table>

5. Research Execution

In the following section, we detail the execution of the GT method that we have used in this study. Following the GT process, we started with transcribing the recorded interview and then iteratively started coding. These coding were then grouped into subcategories and categories. In total we have 3 categories comprising of 45 codes. In the following section, we provide the detail of the resulting codes with short description.

5.1 Resulting collections of codes

The process of coding was executed after the first interview has been transcribed. Quotes and codes are built up freely during the time depends on the comments given by participants in their interview sessions. The coding system in this research comprises 3 top-level categories, 25 subordinate concepts and 1-5 basic codes per concepts giving a total 44 codes. Each code has its own name and short description for further explanation about the code. The details of the research codes are presented in Appendix A and the respective code with quotes of the interviewees are presented in Appendix B.

6. Research Finding

The primary findings included the followings:

6.1 Legacy systems

The interview sessions usually started by asking a definition of a legacy system from the informants. Most of the informants agreed that legacy systems are “old” systems. Despite being old systems, the informants pointed out that legacy systems are “core” systems that have been proven to work correctly in a production environment for decades. P1 provided his opinion about “old” and “core” system as: “Most of the legacy systems are older than 20-30 years...[]. Most of the systems of the legacy systems are the core system.” P11 agreed with P1 by stating: “It [Legacy system] is an old system; ... a lot of legacy system is the core system.”

Interestingly, most of the informants related legacy systems as systems which that do not fit with the future IT strategy of the organization. P19 expressed this as: “My definition of a legacy system is systems and technologies that do not belong to your strategic technology goals.” To investigate further, the informants were asked if programming language is a determining factor for a system being legacy, we obtained a mixed opinion. More than half of the informants do not agree that the programming language is a determining factor for a system to be legacy, while the rest were in agreement. Such a

---

2 http://servicifi.wordpress.com/legacy-survey/
mixed opinion is also observed from the results of the survey. Around 50% of the respondents agreed that there programming languages do determine if a system is legacy.

6.2 Perceived benefits of legacy systems
The perceived benefits of the legacy systems included (i) business critical, (ii) proven technology, (iii) reliable system, and (iv) performance.

6.3 Drivers for modernization
The drivers of the modernization included (i) high maintenance costs, (ii) lack of knowledge, (iii) to remain agile to change, and (iv) prone to failures.

6.4 Challenges of the modernization
The following challenges were identified: (i) complex system architecture, (ii) difficult to extract and prioritize business logic, (iii) lack of knowledge, (iv) data migration, (v) resistance from organization, (vi) time constraints to finish modernization projects and (vii) addressing soft factors of modernization.

7. Validation
The results of the survey\(^3\) are presented in diagrammatic representation below. Note that out of 198 respondents, only 176 have indicated that they have experiences with legacy systems. Thus, the result represents the views of 176 valid respondents.

7.2.1. Demographics data about the participants

![Respondents per country](image)

Figure 2. Demographic information about the respondents

7.2.2. Profession of the respondents

---

\(^3\) The survey is available at [http://servicifi.wordpress.com/legacy-survey/](http://servicifi.wordpress.com/legacy-survey/)
7.2.3 Roles of the respondents

![Roles of the respondents](image)

Figure 4. Roles of the respondents

7.2.4 Perceived benefits of the legacy systems

![Perceived benefits of the legacy systems](image)

Figure 5. Perceive benefits of the legacy systems as per the respondents
7.2.4 Drivers of the legacy modernization project

Legends: Flex.: - Become flexible to change; FTTM: - Faster time-to-market; Fail.: - Prone to failure; Opport.: - Create business opportunities via mergers/acquisitions; Maint.: - High cost of maintenance; Exp./Doc.: - Lack of experts/documentation; Supp.: - Lack of suppliers/vendors

Figure 6. Drivers of the modernization indicated by the respondents

7.2.5 Challenges of the legacy modernization

Legends: TC: - Time constraint to finish modernization; PROI: - Predicting ROI; DM: - Data Migration; FLM: - Funding modernization project; LK: - Lack of knowledge; DT: - Difficult to test; ResS.: - Resistance from staffs; DBL: - Difficult to extract business logic; Narch.: - Non-evolvable system architecture; DCC: - Difficult to communicate the consequences; DPF: - Difficult to prioritize the functionality; CR: - Cultural resistance from organization
Figure 7. Challenges faced by the respondents in legacy modernization

7.2.6. Programming languages

![Pie chart showing distribution of programming languages](image)

Figure 8. Legacy language as perceived by the respondents

7.2.7 Statistical analysis

The chi-square test was conducted to see if there is any associated between job type and the respondents who indicated programming language determines if a system is legacy. However, The analysis with ($\rho = 2.9$), which is far less than the critical value ($\alpha = 12.59$), revealed that such an association does not exist.

### Crosstabs

<table>
<thead>
<tr>
<th>Case Processing Summary</th>
<th>Cases</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Valid</td>
<td>Missing</td>
<td>Total</td>
<td>N</td>
<td>Percent</td>
<td>N</td>
<td>Percent</td>
<td>N</td>
<td>Percent</td>
</tr>
<tr>
<td>Job Type * Do you consider the programming language as one of the factors to decide if a system is legacy?</td>
<td>161</td>
<td>15</td>
<td>176</td>
<td>176</td>
<td>100.0%</td>
<td>176</td>
<td>100.0%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Job Type * Do you consider the programming language as one of the factors to decide if a system is legacy?**

<table>
<thead>
<tr>
<th>Job Type</th>
<th>Do you consider the programming language as one of the factors to decide if a system is legacy?</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Business Analyst</strong></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>26.5</td>
<td>22.5</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>4.3</td>
<td>3.7</td>
</tr>
<tr>
<td><strong>CIO</strong></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>21.6</td>
<td>18.4</td>
</tr>
<tr>
<td><strong>Developer</strong></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>10.8</td>
<td>9.2</td>
</tr>
<tr>
<td><strong>IT Manager</strong></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Expected Count</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Expected Count</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------</td>
<td>---------------</td>
</tr>
<tr>
<td>Software Architect Count</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Expected Count</td>
<td>.5</td>
<td>.5</td>
</tr>
<tr>
<td>Count</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Expected Count</td>
<td>9.2</td>
<td>7.8</td>
</tr>
<tr>
<td>System Analyst Total</td>
<td>87</td>
<td>74</td>
</tr>
<tr>
<td>Expected Count</td>
<td>87.0</td>
<td>74.0</td>
</tr>
</tbody>
</table>

chi-square tests

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>df</th>
<th>Asymp. Sig. (2-sided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson Chi-Square</td>
<td>4.073</td>
<td>7</td>
<td>.771</td>
</tr>
<tr>
<td>Likelihood Ratio</td>
<td>4.529</td>
<td>7</td>
<td>.717</td>
</tr>
<tr>
<td>N of Valid Cases</td>
<td>161</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4 cells (25.0%) have expected count less than 5. The minimum expected count is .46.

8. Conclusion

Although legacy systems and their modernization have been extensively researched, this paper attempts to address the relative absence of empirical studies of industrial perception of legacy systems and their modernization. To the best of our knowledge, this is the first attempt to empirically investigate the perceived benefits of legacy systems, problems associated with legacy system that initiate modernization, and the challenges faced during modernization from an industrial perspective. With the current status of the research some findings complement the academic literature, while some are new and surprising to the previous observations made.

1. We document the industrial perception of legacy systems and their modernization.
2. We identify the perceived benefits of the legacy systems, drivers of modernization, and challenges that the industry faces during modernization.
3. We report the perceptual differences of and about legacy systems between the industry and academia.

As of future work, we aim to validate the findings considering a real world legacy modernization projects.
References

Appendix A

Interview Protocol

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

The interview takes about one hour and some of the sample questions are presented below. The interview will be recorded. Furthermore, we assure you complete confidentiality and if indicated, a report of our findings will be provided.

Participant Background Information

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent organization</td>
<td></td>
</tr>
<tr>
<td>Current Position</td>
<td></td>
</tr>
<tr>
<td>Primary Function</td>
<td></td>
</tr>
<tr>
<td>Work Experience (yrs)</td>
<td></td>
</tr>
<tr>
<td>Expertise</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td></td>
</tr>
</tbody>
</table>

Interview Questions

The information you provide in this interview will be used mainly to construct a new insight from an industrial perspective about legacy systems and modernization for research purposes. Below you will find some of the representative questions that will be discussed.

Part I: Legacy System
1. What is your definition of legacy system?
2. What are the important characteristics of legacy systems?
3. What are the problems with legacy systems?
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. How do you HANDLE the problems caused by Legacy Systems?
7. From all your applications, which Legacy System would you consider Modernizing first?
8. 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 drive Modernization?
3. What are the CHALLENGES of doing Modernization? Can you rank them from (1 being not challenging; 5 being extremely challenging)
4. Please describe STRATEGIES or TACTICS they had employed to overcome the OBSTACLES / CHALLENGES?
5. During the Modernization, what FACTOR can determine SUCCESS / 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?
7. To what extent, the process of modernization is automated?

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.

<table>
<thead>
<tr>
<th>Participant’s feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please describe any unusual circumstances and/or events that had any bearing on the interview such as interruptions, language difficulty, etc.:</td>
</tr>
<tr>
<td>What do you think need to be improved in order to make the interview become better?</td>
</tr>
<tr>
<td>Additional comments:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interviewer Qualifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐ Knowledgeable</td>
</tr>
<tr>
<td>☐ Structuring</td>
</tr>
<tr>
<td>☐ Clear</td>
</tr>
<tr>
<td>☐ Gentle</td>
</tr>
<tr>
<td>☐ Sensitive</td>
</tr>
<tr>
<td>☐ Open</td>
</tr>
<tr>
<td>☐ Steering</td>
</tr>
<tr>
<td>☐ Critical</td>
</tr>
<tr>
<td>☐ remembering</td>
</tr>
<tr>
<td>☐ Interpreting</td>
</tr>
</tbody>
</table>

Date:

________________________
Researcher / Interviewer

________________________
Participant / Interviewee
Appendix B

Category 1. Characteristics

Characteristic of the legacy systems

Concept 1.1. Maintainability.

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

| Concept 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.

| Concept 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. |
| Concept 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.

| Concept 1.3.1 | User Interface Aesthetics | Degree to which a user interface enables pleasing and satisfying interaction for the user. |
| Concept 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).

| Concept 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. |
| Concept 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.

| Concept 1.5.1 | Functional Completeness | Degree to which the set of functions covers all the specified tasks and user objectives. |
| Concept 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.

| Concept 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. |
| Concept 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.

| Concept 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.

| Concept 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.

- **Old system**: Systems which are older than 10 year and was built with an outdated technology.
- **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.

- **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.

- **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 measures the volume of the system.
- **Database issue**: Problem with database in legacy system. Including Unstructured and inappropriate use of data in legacy system.
- **Complicated system**: System in which contain of a lot and complex rules. Complexity determine structural characteristics by examine how object are interrelated.
- **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.

- **Knowledge become scarce**: Condition in which there is not sufficient knowledge available around the system including lack of documentation and expert (programmer, analyst, etc.) in legacy system.

Legacy system in regards to business strategy of organization.

- **Doesn’t fit with future strategy**: Inability of legacy system to support business functionality. It is not a part of the organization’s strategy because it is getting more difficult to in-line with business strategy.
- **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. The systems also bring benefit or profit to organization.

Legacy system from financial point of view.

- **Expensive in maintenance**: The cost for maintenance and operating the legacy systems is perceived high.

Category 2. Challenges
The difficulties that arise in legacy modernization project.

Concept 2.1 Data Migration.
The process of transferring the data from the legacy system into the new system.

- **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.

Concept 2.2 Lack of knowledge.
Limited knowledge to assist the process of legacy modernization.

- **Lack of resources**: Lack of reference (e.g. documentation) and limited manpower to involve in legacy modernization.

Concept 2.3 Non evolvable Architecture.
Obsolete architectural design of the system.

- **Poor system architecture or infrastructure**: Old fashion of building the system or obsolete IT landscape (E.g. monolith and, hardcoded, spaghetti architecture).
- **Difficult to test**: Difficult to complete test case. Test case is 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 2.4 Time constraints.
**Time restriction in the process of legacy modernization.**

| [2.4.1] | Time constraints to finish legacy modernization | Limited time available to finish legacy modernization, including time management in every step of legacy modernization. |

**Concept 2.5 Difficult to extract and prioritize business logic.**

*The challenge in govern the process of legacy modernization.*

| [2.5.1] | Difficult to extract business rules or knowledge | Difficult to extract business rules or knowledge from legacy system. |
| [2.5.2] | 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.). |

**Concept 2.6 Resistance from Organization.**

*The acceptance from the organization and the users of the legacy system.*

| [2.6.1] | 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. |
| [2.6.2] | Resistance from the current users or maintainers in the organization | Uncoooperative manner from current users or maintainers in an organization or behave of certain people which are oppose the process of legacy modernization |
| [2.6.3] | Communicate the reasons or consequences of modernization | Difficult to make people, who involve in legacy modernization, to accept the legacy modernization together with its consequences. |

**Concept 2.7 Funding.**

*The difficulty in getting the business sponsor in the process of legacy modernization.*

| [2.7.1] | 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. |
Category 3. Drivers

Factors that provide impulse or motivation to do legacy modernization.

Concept 3.1. Become Flexible.

Desire to get the flexible system.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3.1.1]</td>
<td>Agile to business requirements</td>
</tr>
<tr>
<td>[3.1.2]</td>
<td>Create new business opportunity</td>
</tr>
<tr>
<td>[3.1.3]</td>
<td>Faster time to market</td>
</tr>
</tbody>
</table>

Concept 3.2. High maintenance cost.

Desire to reduce the cost of maintenance and operational of the legacy system.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3.2.1]</td>
<td>Reduce cost of maintenance and operation</td>
</tr>
</tbody>
</table>

Concept 3.3. Prone to failure

Vulnerable system.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3.3.1]</td>
<td>Prone to failure</td>
</tr>
</tbody>
</table>

Concept 3.4. Lack of knowledge.

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3.4.1]</td>
<td>Lack of supplier / vendor</td>
</tr>
<tr>
<td>[3.4.2]</td>
<td>Lack of documentation / experts</td>
</tr>
</tbody>
</table>

Appendix C

Quotations by the respondents

The process of coding is basically based on the result from transcribing the interviews. Below, we present the quotes and respective mapping to the respective codes from our participants during their interview sessions.

Participant 1

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1.3.2]</td>
<td>Learnability</td>
</tr>
<tr>
<td>[1.5.1]</td>
<td>Functional completeness</td>
</tr>
<tr>
<td>[1.5.2]</td>
<td>Functional Appropriateness</td>
</tr>
<tr>
<td>[1.9.1]</td>
<td>Old system</td>
</tr>
<tr>
<td>[1.2.2]</td>
<td>Stable system</td>
</tr>
<tr>
<td>[1.4.2]</td>
<td>General Performance is good</td>
</tr>
<tr>
<td>[1.6.1]</td>
<td>Interoperability</td>
</tr>
<tr>
<td>[1.6.2]</td>
<td>Adaptable</td>
</tr>
<tr>
<td>[1.11.4]</td>
<td>Poor quality of design and ecosystem</td>
</tr>
<tr>
<td>[1.12.1]</td>
<td>Knowledge becomes scarce</td>
</tr>
<tr>
<td>[3.4.2]</td>
<td>Lack of documentation / experts</td>
</tr>
<tr>
<td>[1.1.1]</td>
<td>Modifiability</td>
</tr>
<tr>
<td>[2.6.2]</td>
<td>Resistance from the current users or maintainers in the organization</td>
</tr>
<tr>
<td>[1.13.2]</td>
<td>Business Critical</td>
</tr>
<tr>
<td>[1.13.1]</td>
<td>Large system</td>
</tr>
</tbody>
</table>
1.3.1 Availability
- The system has been around for a long time and has been tuned to stability, robustness, availability and so on. So they're well performing and stable. Functionalities (quality attributes) that count are stability, robustness, reliability and availability of this system.

1.10.1 Risk of running legacy system
- Risk is one of the other factors.
- Money is not a least problem, risk is the biggest problem. You know, they can afford these systems to failed only for 1 hour.
- It currently works but they worry if it can continue long enough into the future?
- Do the second level you might say that people start beginning to worry if it can continue this way.

2.7.1 Funding the legacy modernization project
- Now you see with this project, the funding is guaranteed: so, business only offers 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.
- You 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 to migrate? It's a tough question.

1.8.1 Not flexible
- So, we are able to more quickly adapt the certain components on our own.
- Infrastructure flexibility.

3.3.1 Prone to failure
- The either main objective is to get smaller, less complex components. So disintegrate the logic and create more self-contained components.
- Because most of them were monolithic... 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 influences more on design and how to optimize things.
- We are revising the central role system who work tidily couple, so more or less monolithic, decomposing them in more... more service oriented way.

3.4.1 Lack of supplier / vendor
- Also continuity, as the driver behind starting current with this technology that you want to have continuity of your core system.

1.14.1 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 programmed, the way they were designed... ok... because most of them were monolithic.
- It can still be a problem of the functionalities or the requirements specification. And of course if you have legacy system ill designed or old fashioned designed as a monolith that might still be the reason for you to change.
- Because most of them were monolithic... 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 influences more on design and how to optimize things.

2.3.1 Poor system architecture or infrastructure
- Making the right decision in how to complement your system.

2.5.1 Difficulty 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.

2.4.1 Time constraints to finish legacy modernization
- They run out of budget, they run out of time: [...] that's mainly to do with scarcity of people on the legacy system.

2.6.2 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 stay 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.

1.11.3 Doesn't fit with future strategy
- If it still suit their needs in the future.

2.6.3 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 vitamins to your environment... a lot of stuff.

**Participant 2**

1.1.1 Knowledge become scarce
- In production it is legacy. My experience is and... 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 in production, you get the ad hoc changes on it and those kind of things, and what you see is that within... for some companies 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.

1.1.5 Poor quality of design and ecosystem
- If it is forced to come up with the quick and dirty solutions often and don't have time afterward to adjust it, to make it well suited in the design of the application itself.

2.4.1 Time constraints to finish legacy modernization
- 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.

3.4.2 Lack of documentation / experts
- Knowledge erosion is the biggest problem there.

1.1.2 Knowledge become scarce
- 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.

3.4.3 Lack of documentation / experts
- Stable system

1.1.3 Back Office
- Legacy system is what you find in back office. Because back office processes are more stable than mid-office and front-office. So you have less business requirements on that paper.

1.1.6 Poor quality to design and ecosystem
- You also see that is because they are very knowledgeable people, the application itself doesn't suffer from many defects anymore. Over the years all the defects has been solved.

1.4.2 General Performance is good
- (Q) Already there...so performance is still good in legacy system. (A) Yeah.

1.1.7 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 the moderns or even in the area of mid-office and front-office. And don't reach 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 the not the question of quality of service of the application.

1.1.11 Multiple system
- A lot of banks and insurance end up with 3 or 4 systems overlapping functionality. Because they cannot decommission the old one.

1.1.12 Business Critical
- So, one of the nice things about that is we also a lot of work at customers decommission really old application.

1.7.1 Unsupported supplier
- I mean what you trigger or other things like for instance a hardware... I have a custom one running the systems on the hardware that run out of support.
- If something is broken down they have to look on EBay to find spare parts.
- So when your environment [legacy systems environment] runs out of support then it is really dying and if that's true then you are already late.

1.14.1 Expensive in maintenance
- Yeah...cost reduction.
- Cost of mainframe is perceived as high. And they want to migrate users to migrate away to mid-range systems.

1.3.2 Create new business opportunity
- It's very difficult for instance in a new old client-server environment, power builder... for instance... yeah... That is just a disaster to work on that, because it is very... a lot of logic is concentrated on the client. And what you also see is that one application is developed that is uses also the database of another one. And it's not through a normal interface but it's a back door to get some data over there.
- They have a lot of legacy applications and it is handcoded often in the code.

1.11.5 Poor quality of design and ecosystem
- Very difficult...if for instance in a new old client-server environment, power builder... for instance... yeah... That is just a disaster to work on that, because it is very... a lot of logic is concentrated on the client.
- And what you also see is that one application is developed that uses also the database of another one. And it's not through a normal interface but it's a back door to get some data over there.

2.2.1 Lack of resource (e.g. documentation, expertise, etc)
- You need C level support for that and you have to involve business already. 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 is to get C level support for this kind of program.

2.6.3 Communicate the reasons or consequences of modernization
- Get really commitment from that level... from that level to really set up the whole period. And not well... OK next year... well... we have less budget... well we stop and we quit.

2.5.2 Difficult to effectively prioritize the functionality for modernization
- If the company or that customer doesn't governs the modernization as one overall project. When you look at large modernization it means that the business will need to be involved.

2.5.2 Difficult to effectively prioritize the functionality for modernization
- If the company or that customer doesn't governs the modernization as one overall project. When you look at large modernization it means that the business will need to be involved.

2.7.1 Funding legacy modernization
- And what is the alternative? Yeah... is to rebuild the system? I can tell you, that is more expensive than restructuring it and making it
Participant 3

3.2.1 Reduce cost of maintenance and operation
- [XXX] looked and said it cost a lot of money, a lot of danger. So last 10 years [XXX] migrated 5 big systems.

1.1.4 Modifiability
- So the technical drivers are old, difficult to maintain, the knowledge was not easy to get because PL/I programmer is not easy to find.

1.3.1 Old system
- From business side, it becomes expensive to technically fulfill the business reason.

1.1.4.1 Expensive in maintenance
- [Q] Is that means...Is that means decisions to start COBOL programming? [A] Yeah... it is. And even though it is not expensive, it is hard to find like skilled people or people who want to do it and who know that all this knowledge is in their head.

1.1.5 Lack of documentation / experts
- [Q] Is it difficult to interconnect with...interconnect the application with your legacy system. [A] Yeah... it is difficult. We try to standardize the interface.

Participant 4 and Participant 5

3.4.1 Lack of supplier/vendor
- Legacy system is a system which service life has expired.

1.1.4.1 Expensive in maintenance
- Maintenance is very expensive...too expensive.
- The maintenance is cost higher.

1.1.1 Modifiability
- Not maintained that means not functional updates or no security updates.

1.3.3 Old system
- That's obsolete technology.

1.3.2 Learnability
- People are used to the system. They know how it works.

1.3.2 Availability
- Proven technology is often the reason they are still in use.

1.1.1 Knowledge become scarce
- But the documentation of the old system is a primary.

1.3.2.2 Lack of documentation / experts
- I think in government, we have influence of politics.

2.2.2 Resistance from the current users or maintainers in the organization
- [Q] If the organization wants to modernize their legacy system, there is certain people who try to not cooperate because they're an expert in legacy system. By the time when legacy is modernized, they might lose their job. Do you see kind of phenomena in organization? [A] Yes.

1.6.4 Interoperability
- [Q] Is it difficult to interconnect with...interconnect the application with your legacy system. [A] Yeah... it is difficult. We try to standardize the interface.

Participant 6

1.1.2 Business Critical
- I think it is a core system, most of the time.
- Like the core transaction.

1.8.1 Not flexible
- They also have a lot of limitations on flexibility.
- Very rigid. Let say if you want to change it like data structure, it is not clear for me what exactly the data structure is.
- Yeah... it is not flexible. It's a lot of hassle.
- But actually, like most companies now say but we want to be flexible.

1.1.5 Four quality of design and ecosystem
- And I could see people 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, little bit there.

2.6.2 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.

1.8.3 Functional completeness
- But for me is all about if you would define, it is about limitation. You see they have street level which is limited to certain characters.
- But it is not process thinking.
- It is only simple logic, this is simple operations.
- There is no a lot...most of the time, there's a lot of complex logic in it.
- So it is just you can like update, maybe the address of the client.

1.1.1 Knowledge become scarce
- I think that's a lot of knowledge which is only in their head.

1.1.4.1 Expensive in maintenance
- Like if you're young like you and me who decides to start 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 designers if you like end of 20th, you just graduated, and nobody decides to start COBOL.

1.3.2.1 Lack of documentation / experts
- [Q] Is that means...Is that means that the maintenance of legacy system also expensive or not? [A] Yeah... it is... so 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.

1.2.1 Robust
- Yeah... just like what I said before, it is robust.
- I think they all UNIX systems. They do not crash often.

1.9.2 Availability
- Like the system they are online, they always available and because there is no a lot of...most of the time, there's 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 totally integrated into the system.

1.3.3 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.

1.3.1 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 Agile to 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.

1.1.5 Poor quality of design and...Also the business logic...let say you want to determine what is the premium? that is all hard coded. That's especially...
[1.1.1.2] Database issue  
- Data garbages  
- So the quality of the data.

[2.1.1] Funding legacy modernization projects  
- Should also have budget right like if he says we are going to 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.

[2.1.1.1] 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 I see even our own, if we started after their own databases tables, and work like that, the project would be quickest so much.  
- 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 merge their data at the time.

[2.6.2] Resistance from the current users or maintainers in the organization  
- Because what is our need if we have a new system, which is working not with COBOL. Who is gonna [going to] need me anymore, so they ditch me after it [modernization] is done. So, why should I cooperate?  
- People all have kind of different motive not to work along.  
- Make people should cooperate but how do you get them there for.

[2.6.3] Cultural resistance in organization not to adopt 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.

[1.1.1.1] Old system  
- The older system.  
- It's like if they choose the system, they're going to use it for a like 10 years.

[3.2.1] Reduce cost of maintenance and operation  
- They wanna have rather like one person having just one user interface.

[2.2.1] Lack of resource (e.g. documentation, expertise, etc.)  
- You need people, skilled people to make services.

[2.6.3] 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.

[2.4.1] Time constraints to finish legacy modernization  
- Just time... so if you want to change something we say like can you change it. O yeah... and then... it also thing they say like yeah... we have limited capacity, because they don't have guys, old pays doing it.  
- First you have to change processes, you have to implement it. I think that takes some time. If you choose who is going to implement it?  
- Who's going to test it? How you are going to migrate the data to the new system... blu, blu, blu, and then it's also maintenance about the time I said... implement it you want... there is be maintained by... there is also going to like some minor changes during... but if they chose, they're going to use it for the coming 5 or 10 years.

[2.6.3] Communicate the reasons or consequences of modernization  
- Like also problem this kind of guy that's not really good in communication.

[1.1.1.1] Old system  
- So legacy system for me is an older system.

[1.1.1] Large system  
- On average is a large system because small systems don't impose big problem.

[1.1.1.4] 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 something with technical complexity, a lot of components, a lot of layers, lot of programs and interfaces interacting with each other.

[1.1.2] Stable system  
- And that basic functionality is stable for very long time because that all for whole year. It is working that way.  
- The functionality is stable... just let it stay as is it.

[1.4.2] General Performance is good  
- On the functionality on average is OK.

[1.1.3] Muddifiability  
- But the problems is with the technical quality, the maintainability is for several reasons had.  
- 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.

[1.1.5] Poor quality of design and ecosystem  
- You made adaptation on and adaptation on... and then 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.

[1.1.2] Adaptability  
- 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 going away to other job.

[3.4.2] Lack of documentation / experts  
- Or I do not have the personal anymore who can do it. Or I don't have documentation and I do not 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 is going... what new effect... still being executed by the program and the coding. That can be the main challenge.

[2.6.3] Communicate the reasons or consequences of modernization  
- Someone like... sometimes you are 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.

[1.1.1.1] Old system  
- The older system.  
- It's like if they choose the system, they're going to use it for a like 10 years.

[2.2.1] Lack of resource (e.g. documentation, expertise, etc.)  
- You need people, skilled people to make services.

[2.6.3] 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.

[2.4.1] Time constraints to finish legacy modernization  
- Just time... so if you want to change something we say like can you change it. O yeah... and then... it also thing they say like yeah... we have limited capacity, because they don't have guys, old pays doing it.  
- First you have to change processes, you have to implement it. I think that takes some time. If you choose who is going to implement it?  
- Who's going to test it? How you are going to migrate the data to the new system... blu, blu, blu, and then it's also maintenance about the time I said... implement it you want... there is be maintained by... there is also going to like some minor changes during... but if they chose, they're going to use it for the coming 5 or 10 years.

[2.6.3] Communicate the reasons or consequences of modernization  
- Like also problem this kind of guy that's not really good in communication.

[1.9.1] System  
- The older system.  
- It's like if they choose the system, they're going to use it for a like 10 years.

[3.2.1] Reduce cost of maintenance and operation  
- They wanna have rather like one person having just one user interface.

[2.2.1] Lack of resource (e.g. documentation, expertise, etc.)  
- You need people, skilled people to make services.

[2.6.3] 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.

[2.4.1] Time constraints to finish legacy modernization  
- Just time... so if you want to change something we say like can you change it. O yeah... and then... it also thing they say like yeah... we have limited capacity, because they don't have guys, old pays doing it.  
- First you have to change processes, you have to implement it. I think that takes some time. If you choose who is going to implement it?  
- Who's going to test it? How you are going to migrate the data to the new system... blu, blu, blu, and then it's also maintenance about the time I said... implement it you want... there is be maintained by... there is also going to like some minor changes during... but if they chose, they're going to use it for the coming 5 or 10 years.

[2.6.3] Communicate the reasons or consequences of modernization  
- Like also problem this kind of guy that's not really good in communication.
Participant 9

[1.6.2] Adaptability
- [Q] So you can you categorize UIMS as a legacy system or not, in your opinion? [A] Usually... 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 technology isn't legacy right? So the technology is still used in 4G networks well.

[1.11.5] Poor quality of design and ecosystem
- Systems that are where the development environment is outdated.
- Not only the development part, we have to think about the architecture of the system also. Because if people are bumping against architecture, changing the earlier architecture.

[1.8.3] Old system
- The same receive concerning the levels of the applications that endured it. I think 30 years old or so.

[1.4.4] Interoperability
- That communicates with, via X.25 connection. So that's all old stuff and then you have to have a legacy system because nothing else can talk that staff again.
- If you integrate application to another system, if you link application... You can do it on different ways. You can say well... I only have low-level API and I just present data by myself. Or maybe the data is already presented by legacy system or whatever system.
- A lot of application running around it the thing to nefioll it ahd that kind of thing, but it is OK. But if you look at harmony, this application is called harmony from [ORTECH] and we have a lot of problem with Harmony.

[1.5.8] Functional completeness
- So the problem you are having at that point is that... that you are being limited in functionality by the legacy system.
- It perfomance is very poor.

[1.1.1] Modifiability
- The effort to build the modifiability is so high that it 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 does not support that functionality anymore... sorry... you have to... we have to start modernization.

[1.13.2] Business Critical
- Legacy system in the core of business.

[1.14.1] Expensive in maintenance
- The amount of people who know that legacy system 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 maybe be platform OS or those kind of things.
- But reason they don't... that is a lot of money, because their framework does not support... they do something strange with enabling the web part of that application. And change that is a lot of money.

[2.7.1] Funding legacy modernization projects
- It is very expensive to change it... still running fine so that's not a problem.

[3.1.1] Agile to business requirements
- And the screening at the gate and the screening at the gate just schipol the only airport that does that...it is needed because the passenger flows mixed. So arriving passengers and departing passengers meet. And that's the problem is schipol is going to...

[1.13.3] Doesn't fit with future strategy
- Legacy system does not facilitate them doing their business.

[2.7.2] Predicting Return of Investment (ROI) of modernization
- For filling the business case.

- Legacy system also tends to hide complexities at business process.

[2.6.1] Lack of resource (e.g. documentation, expertise, etc)
- You do not want COBOL programmer, but what do you want somebody who understands COBOL and can extract functionality in logical was from the legacy system.

[2.5.2] Difficult to effectively prioritize the functionality for modernization
- There is data on how much ICT project is always over budget and takes to much time right? so there is some problems in our management concern in ICT project.
- [Q] when you do modernization do this kind of phenomena. Like people in your...maybe in your team, they don't really cooperate with the modernization process because they are the experts in the legacy system? [A] Yah that happens... So, the most important 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 applications whatever.

[2.7.2] Predicting Return of Investment (ROI) of modernization
- And the amount of change is going through the legacy system was not really foreseeable at the moment we built it, we build the application. So there's some limitations in the application itself.

[2.7.2] Large system
- We have a big system running.

[3.8.1] Not flexible
- And if... it's a legacy system behind it then I can imagine, that you are less flexible presenting the data then newer systems.

[2.4.3] Knowledge become scarce
- They get a head and that kind of thing.

[2.6.3] Communicate the reasons or consequences of modernization
- I do not 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 that's not a case.

Participant 10

[1.13.2] 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 do not fit into the way people want to work and how infrastructure work of today.

[1.1.1] Doesn't fit with future strategy
- They trouble maintaining it, adjusting it.
- I think it will be harder and harder to change the system.

[3.1.2] Create new business opportunity
- [XX] is improving on the customer experience.
- Customers do not like it anymore. So the human experience, the user interface which is old fashion.

[1.11.2] Knowledge become scarce
- Then you see 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.
- Nobody knows all the rules anymore which are in the system.
- Like Japanese, the high linguistic language, the core language, very few people understand it. And so that 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 cannot 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.

[1.1.1] Expensive in maintenance
- 1 Billion for maintenance.

[2.7.2] Predicting Return of Investment (ROI) of modernization
- It is a lot of money... it is enormous of money. But would it be cheaper to make new system?

[2.1.1] Database issue
- The database issue. They have same issue with database management and they want to renew that.

[2.5.2] Difficult to effectively prioritize the functionality for modernization
- We should do it as quickly as 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... Well maybe you should paint it another color.
- [Q] Yeah... you change the decision in the middle of process... [A] Yah.
- The big puzzle is how to do modernization of this system while keeping it maintained.

[2.1.1] Data migration
- 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.
- 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 new business in the new system, and old system.

[1.6.3] Interoperability
- People are still have trouble making the interface of SOA, simple and transparent that are services is easily used. Yeah... it is a big challenge.

[2.6.3] Communicate the reasons or consequences of modernization
- The effort management doesn't understand the issue and they don't give budget for it [legacy modernization].
Participant 11

1.9.1 Old system
- There is a lot of legacy systems that are 20-30 years old already.
- A legacy system is an old system.

1.9.2 Business Critical
- It is an old system, but it is an obsolete system in a business system, because it is not a system of the core system.
- No good enough to run the legacy systems. Because then they can increase their profit.

1.9.3 Availability
- Normally, because there are not many changes in legacy system availability, they are very available, and there are more less 24/7 up and running.
- Time behavior
- It is also characteristic about response time, it is very fast because normally it is very simple to send messages, so there is not that a lot of overhead.

1.9.4 Knowledge becomes scarce
- There are several knowledge of that system in organization because the people are knowledgeable about that kind of system are already gone; they are already left building you might say. And documentation, so the characteristic also from old system that is the lack of documentation, because it was never documented in those days.
- We have 100 thousand people working for [XXX] in India. That is the biggest part of [XXX] in India. But there is a lot of knowledge around. So where we built up education around legacy transformation in India itself.

1.9.5 Modifiability
- It is hard to maintain because there is no real updated software anymore and it is not also supported by a lot of other organizations.

1.9.6 Unsupportable supplier

1.9.7 Interoperability
- The second one is a system integration. Because old data system has most of more or less have old application program interface. So sometimes there is no or less a lack of knowledge about integrating legacy systems by itself.

1.9.8 Unsuccessful implementation

1.9.9 Lacks of documentation or/and expertise

1.9.10 Difficult to effectively prioritize modifiability for modernization
- Sometimes people do not like changes. Not only in the business organizations, but also in IT organization. So you need to persuade them for the need of transformation [modernization].
- Sometimes they start the transformation, then forget to educate the people.

1.9.11 Interoperability
- The second one is a system integration. Because old data system has most of more or less have old application program interface. So sometimes there is no or less a lack of knowledge about integrating legacy systems by itself.

1.9.12 Data migration
- The main risk in modernization is that the data migration, which cannot be done perfectly. Errors are made and you have some risk that your new system is disturbed after modernization.

1.9.13 Cultural resistance
- People do not like changes. So you need to do something for their acceptance.

1.9.14 Poor quality of design and ecosystem
- That has everything to do with it, I think the architectural technical complexity.

Participant 12

1.1.1 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.

1.1.2 Modifiability
- It needs constraint by technology, for instance it is built on technology that is no longer maintained.

1.1.3 Lack of documentation or/and expertise
- Or it may rely on specific people because lack of documentation or complexity of the system.

1.1.4 Knowledge becomes scarce

1.1.5 Lacks of documentation or/and expertise

1.1.6 Funding legacy modernization projects
- The third one is a reverse transformation plan itself. What steps will you do you need to prepare to govern the whole transformation plan by itself. So you need to very good project manager.

1.1.7 Business Critical
- Core is the system where is very difficult to find out what are exactly other rules that are implemented.

1.1.8 Functional completeness
- Sometimes people do not like changes. Not only in the business organizations, but also in IT organization. So you need to persuade them for the need of transformation [modernization].

1.1.9 Expensive in maintenance
- If you modernize it now they have to completely rework.

1.1.10 Cultural resistance in organization not to adapt new system
- Sometimes they do not want to do that so they choose to completely rework, it will be major investment. Many man years project

1.1.11 Difficult to test
- It is usually testing. Because that difficult also difficult to get your test case complete.

1.1.12 Difficult to extract business rules or knowledge
- It is more challenge for...to get requirements very clear and complete.

1.1.13 Communication 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.

1.1.14 Cultural resistance in organization not to adapt new system
- Sometimes it can be it difficult and to let that go and start working on new technology and new system.
- Sometimes they see legacy systems as their baby and they tend to know every aspect of it. Sometimes it is difficult to work with them while modernizing because they might not share their knowledge.

1.1.15 Difficult to test
- It is usually testing. Because that difficult also difficult to get your test case complete.
- It can be difficult to extract all the use cases to test it. It also requires a lot of work to compare functionality from legacy system to the new application. It can be time consuming and difficult work.

1.1.16 Difficult to extract business rules or knowledge
- It is more challenge for...to get requirements very clear and complete.
[2.1.1] 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 those improve on the quality you may end up with the quality it is...can very...well...the large project on its own.

[1.1.2] Database issue
- Because the system, they have been running for many years, they tend to be some dirty data inside.

[2.4.2] Resistance from the current users or maintainers in the organization
- They can be satisfy with the system, but they can be very dissatisfied with the risk building up after the years.

[1.1.0] 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 the risks, they see it as an operational risk when staying with the current.

[2.4.3] Time constraints to finish legacy modernization
- And it will take many months or sometimes even more than one or two years for the project of modernization to complete.

[2.5.2] 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 deliver, you always deliver late.
- Because you deliver on specification which 1.5 year old.
- 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.

[1.3.0] Not flexible
- They expect more flexibility when they want to change their business process.
- They expect more cheaper and faster ways of implementing those changes.

Participant 13

[1.9.4] Old system
- Legacy system for me is a combination of hardware and software which was built long time ago.

[1.1.2] Knowledge become scarce
- It is usually built or installed or configured by someone or somebody who is not around anymore, so lacking knowledge.

[3.4.2] Lack of documentation / expertise
- (1) Some knowledge scarcity is a problem in the future. (2) Years, years...
- The issue is that there is less knowledgeable people [experts] so they are not available in the organization because the knowledgeable people of the system are already gone [left job]. The other characteristic of legacy systems is the lack of documentation.

[2.7.1] Funding legacy modernization projects
- Usually a lot of time and energy and money and research involved to get rid of it.
- So it is a lot of invest. Cost them a lot of time.
- Basically a 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.

[3.1.2] Create new business opportunity
- 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.
- Natural moment. So basically if something else happen then which forces you to invest in the application anyway, then migrate it to a new platform.

[1.1.4] Expensive in maintenance
- So simply the skill is not available, the only way to by pass that is to hire quite expensive, dedicated consultants.
- It is getting more and more expensive because risk are getting bigger and also knowledge is getting scarce.
- You have to hire somebody quite expensive to make your legacy world SEPA comply.
- Maintenance cost are getting higher because knowledge is getting scarce and risk of failure are growing.

[2.2.4] Lack of resource, e.g. documentation, experts
- You do involve the business, the users, the owners, to test, to monitor, to see if the application is working.

[3.2.4] Reduce cost of maintenance and operation
- We want to do more toward a standardized way of working in the entire company.
- So standardize from process point of view.

[2.4.1] Time constraints to finish legacy modernization
- The business impact is always the bigger ones. An it usually just simple time and money.
- Your biggest problem is an availability of resources [documentation and expertise] and availability of money and [some] extend availability of time.

[2.6.2] Resistance from the current users or maintainers in the organization
- Usually, the people who use the system are satisfy with the system.

[2.5.2] Difficult to effectively prioritize the functionality for modernization
- Basically, what you need to have is you need to have strong governance model. You need to have one clear set of rules, which make a decisions and what the impact and magnitude the decision can be.
- Yeah...so you always have to make sure from ok... if I do a migration always have to do a lot of check and balances. I have to do yes...I have to do comply check... I have to do assessment. I have to do...I have to make sure that everythings works, including the people.

[1.1.0] Risk of running legacy system
- Eventually the risks could be the legacy system falls down, our production process is a stopped.
- Continuity. That the biggest problem. Because it is old it will fall down.

[1.1.1] Modifiability
- Because it is old and the knowledge is not available, no body can do the adjustment something like that.

[1.8.3] Not flexible
- Implement a changes coming from your environment like the SEPA project or couple of years ago the implementation of euro with staff 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 half of a lot more cost.

[3.3.1] Prone to failure
- From the cost avoided point of view to have your environment up to date and integrated and standardize as much as possible. In the end that is a cheaper way run your IT.
- We have an old ERP system, old almost 10 years old. And it drives the production in the plant and also the logistic and warehous and also the order towards the customers. If that system stops, the plant stops, the warehouse stops. Which is a lot of business opportunity.

[2.6.3] Communicate the reasons or consequences of modernization
- Somebody quite high at the top, somebody who can enforce decisions where they need to be enforce. And somebody who can also get workers to change their behavour.

[2.5.2] Difficult to test
- So there’s always human work and also always room for human error.
- No...well...testing sometimes. But that is it. For now and then we use every known. And we had use a script to for instance to deploy an image an applications on a virtual server and stuff like that.

Participant 14

[1.9.3] Old system
- Let’s call it a system which is older.

[1.1.2] 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.

[1.3.3] 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.

[1.1.2] 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.

[1.1.8] 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.

[1.1.4] Expensive in maintenance
- If you look at the maintenance cost, I’m quite sure I can run in similar environment against lower cost. If I would use a standardized product, let’s say state of the art ERP environment...because I don’t need somebody to maintain.
- So I’m quite sure I can make it cheaper but that means you have three of 1 or 1 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.

[2.7.2] Predicting Return on investment (ROI) of modernization
- Are you willing and then you have to calculate the risk of the hoor if it can happen. And then you make a decision.
### Participant 15

1. **Not flexible**
   - Is it flexible enough to be changed in order to support the future requirements? So that's flexibility.

2. **Poor quality of design and ecosystem**
   - Is the technical architecture of these applications still up to date and still supported for the near future?

2. **Data migration**
   - The larger system though, the larger back office systems, take some time, because you need to convert data.

2. **Funding legacy modernization projects**
   - This is time consuming, and costly.

2. **Time constraints to finish legacy modernization**
   - Transformation of old system to a new system is costly since you need to test it very carefully. And testing is costly.

3. **Difficult to test**

4. **Interoperability**
   - Legacy system typically are old in a integration with the internet. They do not support integration with the internet.

5. **Functional completeness**
   - Do you expect your target system when you do modernization would be more beneficial than your legacy system? [A] Yes, absolutely.

6. **Lack of resource (e.g. documentation, expertise, etc.)**
   - Really need people who understand the old system in order to transform the data to the new system. And those people are usually quite scarce.

6. **Communicate the reasons or consequences of modernization**
   - Data transformation is not a very sexy subject.

6. **Communicate the reasons or consequences of modernization**
   - Really need people who understand the old system in order to transform the data to the new system. And those people are usually quite scarce.

6. **Frame 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 change to the future requirements.

6. **Agile to business requirements**
   - If you have application landscape which is too complicated, you need to simplify it, otherwise you are not agile to the future.

6. **Complicated system**
   - We need to implement as an insurance company, quite a few legal requirements. If you don’t have enough budget.

6. **Knowledge become scarce**
   - So that's a reason why people are a little bit less reluctant.

6. **Reduce cost of maintenance and operation**
   - Idealy you have standard interface between each of these compartment of each functionality. And SOA helps with that.

6. **Frame to failure**
   - If you have separated functional areas, you can also separate maintenance. This is the way of managing complexities.

### Participant 16

1. **Lack of supplier / vendor**
   - End supported or end of life be the supplier.

1. **Adaptability**
   - It’s not compatible in our IT environment.

1. **Multiple system**
   - The fact that lot of time we replace application but the old application is still running.

2. **Funding legacy modernization projects**
   - Not enough budget.

1. **High risk of running legacy system**
   - Big risk because you don't have any support of your supplier.

1. **Stable system**
   - On legacy, we don’t do any change anymore so sometimes it is stable.

1. **Knowledge become scarce**
   - We still have people with knowledge of legacy system. But what we see is that the knowledge is walking out because people are moving to other job.

1. **Reduce new business opportunity**
   - But it is not good for our customers Resistance from the current users or maintainers in the organization.

1. **Not flexible**
   - On the other things why we want to get rid of the legacy because it is very... it is not flexible.

1. **Large system**
   - Because legacy is rigid, it big.

1. **Expensive in maintenance**
   - We now migrate it to ditoEKE and one licence fee is deep for something like half.
Participant 17

- **Old system**: It is old.
- **Unsupported supplier**: There’s no patches. If suppliers stop their product, organization needs to find another way to keep supporting their systems.
- **Lack of sufficient experience** (e.g., documentation, expertise, etc.): Sometimes we are the first example in the world.
- **Availability**: It is proven technology most of the time.
- **Expensive in maintenance**: Yeah, actually higher, because it is not supported anymore.
- **Data migration**: So biggest problem was how to update these 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 below.
- **Create new business opportunity**: 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 consequences.
- **Interoperability**: There’s no patches. If suppliers stop their product, organization needs to find another way to keep supporting their systems.
- **Lack of resource (e.g., documentation, expertise, etc.)**: If you are the first, you think you will have a risk. Biggest risk is you go to the new system that has not any proven technology it or is really new.
- **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 we will take 3 years to get that kind of transformation.
- **Communicate the reasons or consequences of modernization**: The technical difficult was all the sub-systems of the legacy system. So there was something of printing, there was something for reporting, there was some external connection for getting all the data, there were so many system and connectivity around the legacy system.
- **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 have a plan, we start to do more training, do small implementations. No they’re not satisfied. But I think in IT you do not really have satisfied end users because if the system does not work for 5 minutes, they are already frustrating because they need to do something in those 5 minutes.
- **Predicting Return of Investment (ROI) of modernization**: I think maybe the acceptance of the business people that they won’t accept the newer system or hard to change.

Participant 18

- **Knowledge become scarce**: Nobody really knows about it, but see there’s reason for not to touch it.
- **Lack of documentation / expertise**: They do not know what the hidden features are behind or what the business rules.
- **Expensive in maintenance**: The performance is not really good.
- **Adaptable**: Yes because it doesn’t work in windows. It works on windows 8 but not really… doesn’t really support all technology.
- **Unsupported supplier**: Hardware that are not supported anymore. Environment.
- **Old system**: Legacy system don’t support very good databases or does not have relation database model.
- **Value in a new solution**: I think maybe the acceptance of the business people that they won’t accept the newer system or hard to change.
[2.6.2] 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.

[1.1.4] Complicated system
- The one I have is pretty complex. The one I use for the IT is a terminal.

[1.6.4] Interoperability
- But they have an old legacy system which is very hard to connect to that bag.

[1.3.3] User interface aesthetics
- Interface is pretty clear that’s just a message like XML but in the old version.
- Clean and clear interfaces.

[2.6.3] Communicate the reasons or consequences of modernization
- They have cargo, they have to transport cargo from there to there. That is 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.

[1.1.1] Agile in business requirements
- If something happen and then there is... They need 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 it’s to promote the traffic by the river. And then off by the road, promote traffic jam etc.
- For the government, change in regulation.

[1.1.5] Poor quality of design and ecosystem
- Because the program is really hard-coded, it is not comparable that makes modernization difficult.

[1.1.5] Poor quality of design and ecosystem
- Because the program is really hard-coded, it is not comparable that makes modernization difficult.

[2.6.3] 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 have read a book about SOA and we... it should be SOA.
- So there is a lot of confidence and emotion.

[2.7.1] 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 do not think return of investment of 1 year can...
- Big financial given by the governments to make some modernization.

Participant 19

[1.1.3] Doesn’t fit with future strategy
- My definition of a legacy system is systems and technologies that do not belong to your strategic technology goals.

[1.6.4] Adaptability
- Very hard to keep it up to date, to make it more forward to newer versions.
- Every old system is stuck in the technology that you dont wanna carry forward.

[1.1.4] Modifiability
- Now that is the legacy. You will not find anyone who can maintain that.
- It is always new rules, regulations, common labour agreement, and thing that changed between taxes, so that’s always something changes. So there, legacy is really painful.

[1.1.12] Knowledge becomes scarce
- It’s a product that we still maintain as we go forward but for as it is very hard to find new UNIFACE developer anymore.

[3.4.2] Lack of documentation / experts
- 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 kind of things. That’s very limited group.
- Undocumented feature or knowledge, things in the product that no one knows about or specific feature that works forever and that has been maintained recently and the one that invented it has long left the company.

[1.5.2] Functional appropriateness
- It works as expected. Even if that bugs, people know the bugs, people have all worked around.
- Because there expectation are completely inline with what they are getting.

[1.1.2] Stable system
- Then good things about a product, a program that you dont maintain anymore it’s 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 bugs so that know 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 is expected.

[1.4.4] Time behavior
- Because hardware, mono-hardware always get faster.
- Old code went as amazing speed on new hardware.

[1.1.4] Learnability
- Yeah, they 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 des. On if you press wrong just do this step and then you back on track.

[2.8.2] Resistance from the current users or maintainers in the organization
- And most people very happy.

[2.1.4] 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 dialog, how you store them on your databases, so that’s kind all specific in the data migration that is hard.

[2.6.1] Cultural resistance in organization not to adopt 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.

[1.3.1] Agile to business requirements
- If they come up with new kind of rules, of regulation and you have to come up with additional test then it has the different way of calculation, you have to go in and build new type of rule.
- Say everyone has to pay this amount of test except for single mothers with children under 5, single retired people over the age of 50 or 70, people like this, people like that.

[1.7.4] Unsupported suppliers
- If the technology supplier is not support it anymore, then you also have a risk.
- So, even [XXX] stop supporting it, then you are stuck because then you get a new hardware, you dont have a driver anymore for the new equipment or, so you have to figure out how can install the platform and install the application in the platform.

[1.4.1] Expenses in maintenance
- So prices goes up, because they have, basically they have so much capacity and for maintaining it, it cost them those money.

[3.4.4] Lack of supplier / render
- Where the product was at the end of the product life cycle.
- Continuity. Continuity is the biggest issue that I have because if... and it can be, because you can find the developers anymore. you are stuck. If the technologies is not support anymore by a provider.

[2.5.5] Poor quality of design and ecosystem
- But usually systems are so closely integrated and complex.

[1.6.1] Interoperability
- Technology incomparability. So you have rack in mainframe and you have rack in windows, and still rack script that one on mainframe just not always run on windows, because implementation, slightly different.

Participant 20 and Participant 21

[1.1.3] Business Critical
- And we love our customer legacy system. Because as a company, we are successful because of all differences between legacy systems. Because all legacy system talk and communicate with the same protocol, then we would not exist.
- So the difficulties of supporting legacy is one of the cores of our business.

[1.6.1] Interoperability
- Because normally these systems are not talk each other.

[1.4.1] Modifiability
- Can we do maintenance on the system? And if it become 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
[1.1.1.1] Risk of running legacy system  
- You are always about risk right? You have to balance risks. If it’s big risk to maintain in current state...well risk could be economical risk or technical risk. If the risk become so big, then you should consider it as a legacy.  
- We also have legacy but we do not get rid of it, because it has no big risk yet. But we still consider it as legacy.

[1.1.1.1] Doesn’t fit with future strategy  
- Some of the engine need consider it is legacy, but we still use them. But we not enhance them anymore or add new feature to that.

[1.1.1.1] Multiplex 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.

[1.1.1.1] Complicated system  
- There is so much rule and information in that engine.

[2.5.1] Difficulty to extract business rules or knowledge  
- To deal with all kinds of exception. To work around and cooperate with other system.
- To extract all the rules and details in there [legacy systems] is really difficult.

[3.1.1.1] Knowledge become scarce  
- One of the factors for maintanability is also documentation. So it is 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 is going on in the old system and the risk to migrate is bigger.

[3.1.1.1] Unsuppored supplier  
- Whenever the platform the software ran on is outdated, so we cannot transfer what is there. It is much more serious level are very high.

[2.5.1] Difficulty to extract business rules or knowledge  
- Who likes further than current version and who look at new techniques and new frameworks on new things like SQRS or big data or who looks further than current system and spot the technique that are useful to migrate to
- They [Top management] are always looking for a short term Return of Investment. Once you put the money in, they want to earn it back.

[2.6.1] Communication 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 in a longer time to get the experience.

[2.5.2] Difficult to test  
- Regression test that cover enough all the contingency.

[2.6.2] Difficulty to extract business rules or knowledge  
- liffe cycle, so we have to acquire the necessary to maintain it. But we do not enhance it anymore or add new feature to that.

[3.1.1.1] Agile to business requirements  
- There is a government body called NCHIT, who decide the standard for modernizing. That is one driver if they upgrade the standard, then we have to migrate to.
- Other point is that my customer wants flexibility, and a short time-to-market, then you have to get rid of your legacy. Because legacy is rigid, and it is not flexible.

[1.1.1.1] Knowledge become scarce  
- One of the factors for maintainability is also documentation. So it is lacking, then it is much more difficult for new developers to learn how things work, a huge factor.

[2.6.3] Communication 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 in a longer time to get the experience.

[2.5.2] Difficult to test  
- Regression test that cover enough all the contingency.

[1.1.1.1] Doesn’t fit with future strategy  
- Legacy is what I would get rid of. Legacy is kind of what hindering me.

[1.1.1.1] Not flexible  
- So it is not flexible enough to do the things you wanna do.
- They cannot response to change request.

[2.7.1] 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.

[2.5.2] Difficulty to effectively prioritize the modernization  
- But that doesn’t work. Because every time the attention goes away from modernizing to functionality.
- 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.

[1.1.1.1] Agile to business requirements  
- The pension world is very much movements now. so it is very rapidly changing.

[3.1.1.1] Knowledge become scarce  
- We cannot find the people to keep it going.

[2.5.1] Difficulty to extract business rules or knowledge  
- 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.

[1.1.1.1] Complicated system  
- There is so much rule and information in that engine.

[3.1.1.1] Lack of documentation / experts  
- One of the factors for maintanability is also documentation. So it is lacking, then it is much more difficult for new developers to learn how things work, a huge factor.

[1.1.1.1] Not flexible  
- So it is not flexible enough to do the things you wanna do.
- They cannot response to change request.

[3.1.1.1] Faster time-to-market product  
- They have too high time to market.
- We need a faster time to market, and we are not able to do that in COBOL environment.

[2.6.1] Communication 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 in a longer time to get the experience.

[2.5.2] Difficult to test  
- Regression test that cover enough all the contingency.

[1.1.1.1] Lack of documentation / experts  
- One of the factors for maintanability is also documentation. So it is lacking, then it is much more difficult for new developers to learn how things work, a huge factor.

[2.6.1] Communication 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 in a longer time to get the experience.

[2.5.2] Difficult to test  
- Regression test that cover enough all the contingency.

[1.1.1.1] Knowledge become scarce  
- One of the factors for maintainability is also documentation. So it is lacking, then it is much more difficult for new developers to learn how things work, a huge factor.

[2.6.3] Communication 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 in a longer time to get the experience.

[2.5.2] Difficult to test  
- Regression test that cover enough all the contingency.
Participant 24

<table>
<thead>
<tr>
<th>[1.1.1]</th>
<th>Expensive in maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1.1.1]</td>
<td>Multiple systems</td>
</tr>
<tr>
<td>[1.1.1]</td>
<td>Complicated system</td>
</tr>
<tr>
<td>[1.1.2]</td>
<td>Business Critical</td>
</tr>
<tr>
<td>[3.1.1]</td>
<td>Knowledge become scarce</td>
</tr>
<tr>
<td>[3.4.2]</td>
<td>Lack of documentation / experts</td>
</tr>
<tr>
<td>[1.4.2]</td>
<td>General Performance is good</td>
</tr>
<tr>
<td>[3.2.3-4]</td>
<td>It's business, not IT decision</td>
</tr>
<tr>
<td>[2.0.3]</td>
<td>Resistance from the current users or maintainers in the organization</td>
</tr>
<tr>
<td>[1.2.1]</td>
<td>Data migration</td>
</tr>
<tr>
<td>[1.2.1]</td>
<td>Lack of resource (e.g. documentation, expertise, etc.)</td>
</tr>
<tr>
<td>[1.18.1-2]</td>
<td>Risk of running legacy system</td>
</tr>
<tr>
<td>[2.0.3]</td>
<td>Communicate the reasons or consequences of modernization</td>
</tr>
<tr>
<td>[2.0.3]</td>
<td>Time constraints to finish legacy modernization projects</td>
</tr>
<tr>
<td>[1.5.2]</td>
<td>Functional appropriateness</td>
</tr>
<tr>
<td>[1.5.2]</td>
<td>Availability</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Not flexible</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Functional appropriateness</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Functional completeness</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Modifiability</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Create new business opportunity</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Create new business opportunity</td>
</tr>
</tbody>
</table>

Participant 25

<table>
<thead>
<tr>
<th>[1.8.1]</th>
<th>Not flexible</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1.11.5]</td>
<td>Poor quality of design and ecosystem</td>
</tr>
<tr>
<td>[1.11.3]</td>
<td>Business Critical</td>
</tr>
<tr>
<td>[1.11.2]</td>
<td>Unsupported supplier</td>
</tr>
<tr>
<td>[1.11.1]</td>
<td>Risk of running legacy system</td>
</tr>
<tr>
<td>[1.11.4]</td>
<td>Agile to business requirements</td>
</tr>
<tr>
<td>[1.11.4]</td>
<td>Expensive in maintenance</td>
</tr>
<tr>
<td>[3.1.2]</td>
<td>Create new business opportunity</td>
</tr>
<tr>
<td>[3.1.2]</td>
<td>Create new business opportunity</td>
</tr>
</tbody>
</table>

Participant 26

<table>
<thead>
<tr>
<th>[1.9.3]</th>
<th>Old system</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1.8.1]</td>
<td>Not flexible</td>
</tr>
<tr>
<td>[1.12.1]</td>
<td>Knowledge become scarce</td>
</tr>
<tr>
<td>[3.4.2]</td>
<td>Lack of documentation / experts</td>
</tr>
<tr>
<td>[1.1.1]</td>
<td>Modifiability</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Functional appropriateness</td>
</tr>
<tr>
<td>[1.3.2]</td>
<td>Learnability</td>
</tr>
<tr>
<td>[1.6.2]</td>
<td>Adaptability</td>
</tr>
<tr>
<td>[1.3.1]</td>
<td>Agile to business requirements</td>
</tr>
<tr>
<td>[2.0.3]</td>
<td>Resistance from the current users or maintainers in the organization</td>
</tr>
<tr>
<td>[Q]</td>
<td>Do you think that the maintenance of legacy system could be cheaper if you move legacy system to the new environment?</td>
</tr>
<tr>
<td>[1.4.2]</td>
<td>General Performance is good</td>
</tr>
<tr>
<td>[3.1.1]</td>
<td>Agile to business requirements</td>
</tr>
</tbody>
</table>
| [Q] | So everyone from old system not cooperate? | [A] Yeah...
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.

Cultural resistance in organization not to adopt 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.

Communicate the reasons or consequences of modernization

- Yes...when they support it, they have to give time and money to go to education.

Difficult to effectively prioritize the functionality for modernization

- We are very optimistic. That's one of the lesson learnt to take your time.

Index of Codes to 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

[1.8.1] Not flexible

[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

[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.

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

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

[1.14.1] 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

[3.1.2] Create new business opportunity

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

[3.2.1] 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.

[3.3.1] Prone to failure

[3.4.1] Lack of supplier / vendor
P1, P7 and P8, P9, P16, P17, P19, P22, P23, P18, P4 and P5.

[3.4.2] Lack of documentation / experts
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] Data migration

[2.2.1] Lack of resources
P2, P9, P11, P13, P15, P24, P6, P26, P16, P17.

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

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

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

[2.5.1] Difficult to extract business rules or knowledge
P1, P12, P19, P20 and P21, P22, P18, P14.
[2.5.2] Difficult to effectively prioritize the functionality for modernization
P2, P7 and P8, P9, P11, P12, P13, P17, P20 and P21, P22, P14, P26, P10.

[2.6.1] Cultural resistance in organization not to adapt new system

[2.6.2] 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.

[2.6.3] 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.

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

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

In this section we present the online survey that is publicly available at esurv.org⁴.

1. Personal Information

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.

Please add your personal information.

* Country of work

* Experience (No. of years in IT)

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.

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:

What is your role in the organization?

☐ Researcher  ☐ Developer

☐ System Analyst  ☐ Business Analyst

☐ IT manager

☐ Chief Information Officer

Other (specify)

⁴ http://esurv.org/online-survey.php?surveyID=OCMKGL_857c0d3b
2. Characteristics of legacy systems

**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.

What do you think are defining characteristics of legacy systems?

☐ Reliable system
☐ High performance
☐ Proven technology
☐ Business critical

Others:

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.

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of experienced manpower</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limited suppliers/vendors to support and maintain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unable to adequately support, maintain, or enhance inhouse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incompatible with current and/or future technological environments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Too rigid to comply with new business requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monolithic architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High risk of failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor user interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Too costly to maintain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Do you consider the programming language as one of the factors to decide if a system is legacy?

☐ Yes
☐ No
Can you give some examples of programming languages that you consider a legacy or are being used as a legacy in your organization?

3. Legacy Modernization Projects

Definition:
Legacy Modernization is a process of migrating / evolving a software system to a new target system.

Have you ever been involved in legacy modernization?

- [ ] Yes
- [x] No

Please rank which of the following driver(s) are the motivating factors to modernize legacy systems.

- 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/documentation around legacy systems
- Prone to failures
- Faster time-to-market of product

Very weak  Weak  Strong  Very Strong

How challenging are the following obstacles in a legacy system migration projects?

- 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
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

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:

Do you use any academic resources while performing legacy modernization projects? (for example: research articles, tools, etc)

- [ ] Yes
4. Legacy Modernization Project

What resources do you use from academia?

☐ Research articles
☐ Research software tools
☐ Research collaboration with academics
☐ Internship students
Other (specify)

5. Legacy Modernization Project

If your answer to Question 13 (Do you use any academic resources while performing legacy modernization projects) is No, please specify some reasons.

How can academics play a larger role in industrial legacy modernization projects?

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.)

This research is a part of the ServiciFi research project, funded by NWO/JACQUARD research program and is carried out by B.V. Batlajery, drs. R. Khadka, and Dr. S. Jansen on behalf of Utrecht University.

Thank you very much for your participation in this survey. If you have any questions or concerns please do not hesitate to email us at legisystemstudy@gmail.com.

This research is being carried out by B. V. Batlajery; drs. R. Khadka; Dr. S. Jansen on behalf of Utrecht University.
Appendix E

We presented the overall result of the survey from the esurv tool\(^5\). Note that these results represent the findings of all respondents (i.e., this result also includes responses that were discarded). Hence, the results presented by esurv differ from the results of Section 7.

\(^5\)www.esurv.org
Results for: Revisiting legacy systems and legacy modernization from industrial perspective

Page: 1/7

Personal Information

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.

<table>
<thead>
<tr>
<th>Country of work</th>
<th>Experience (No. of years in IT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netherlands</td>
<td>5</td>
</tr>
<tr>
<td>Portugal</td>
<td>25</td>
</tr>
<tr>
<td>United States</td>
<td>40</td>
</tr>
<tr>
<td>usa</td>
<td>30</td>
</tr>
<tr>
<td>India</td>
<td>13</td>
</tr>
<tr>
<td>Canada</td>
<td>32</td>
</tr>
<tr>
<td>Costa Rica</td>
<td>20</td>
</tr>
<tr>
<td>United States</td>
<td>35</td>
</tr>
<tr>
<td>Netherlands</td>
<td>23</td>
</tr>
<tr>
<td>france</td>
<td>25</td>
</tr>
<tr>
<td>usa</td>
<td>30</td>
</tr>
<tr>
<td>netherlands</td>
<td>27</td>
</tr>
<tr>
<td>Holland</td>
<td>25</td>
</tr>
<tr>
<td>Netherlands</td>
<td>29</td>
</tr>
<tr>
<td>Netherlands</td>
<td>19</td>
</tr>
</tbody>
</table>

2) Do you have experiences with systems that are considered legacy within the organization?

<table>
<thead>
<tr>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>89.90</td>
</tr>
<tr>
<td>No</td>
<td>10.10</td>
</tr>
</tbody>
</table>

Answered Question: 209
Skipped Question: 2

If Yes, please indicate your experience with legacy systems in years.

- 10
- 30
- 15
- 1
- 20
- 25
- 10
- 24
- 15
- 4

3) Please specify the domain of your organization.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic Institution</td>
<td>9.35</td>
<td>26</td>
</tr>
</tbody>
</table>
I checked multiple options because I worked for many different companies during my 20+ years. The bulk of my experience with legacy systems was in the telecom and financial sectors.

4) What is your role in the organization?

<table>
<thead>
<tr>
<th>Role</th>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher</td>
<td>18.43</td>
<td>40</td>
</tr>
<tr>
<td>Developer</td>
<td>26.27</td>
<td>57</td>
</tr>
<tr>
<td>System Analyst</td>
<td>17.51</td>
<td>38</td>
</tr>
<tr>
<td>Business Analyst</td>
<td>12.90</td>
<td>28</td>
</tr>
<tr>
<td>IT manager</td>
<td>16.13</td>
<td>35</td>
</tr>
<tr>
<td>Chief Information Officer</td>
<td>8.76</td>
<td>19</td>
</tr>
</tbody>
</table>

Other (specify)

- Sales & Marketing
- Technical lead legacy data migration
- Domain Architect
- Partner
- IT Architect
- IT consultant
- business architect
- Project manager
- Marketing
- Director

Page: 2/7

**Characteristics of legacy systems**

**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/functionality and their failure can result in serious financial & legal problems, damages and other penalties.

5) What do you think are defining characteristics of legacy systems?

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable system</td>
<td>25.60</td>
<td>107</td>
</tr>
<tr>
<td>High performance</td>
<td>11.72</td>
<td>49</td>
</tr>
<tr>
<td>Proven technology</td>
<td>26.32</td>
<td>110</td>
</tr>
<tr>
<td>Business critical</td>
<td>36.36</td>
<td>152</td>
</tr>
</tbody>
</table>

**Others**:

- New systems and technologies are invariably superior to legacy
- No longer under active development except for minor feature upgrades and bug fixes
- Not flexible (hard and expensive to change)
- Mature applications, strong integration with business processes
- Poorly documented, limited skilled staff available
- People don’t want to change them; if there’s a bug, fix it without changing the code (or change it as little as possible)
- Frequently underinvested in; based on vendor hardware or software that is out of support; hard to find trained resources
- Mostly a perfect fit for the organization.
- End of life hardware, hard to replace, hard to find spare parts
- Resistance to evolution and maintenance

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.

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Responses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of documentation</td>
<td>9.29%</td>
<td>11.48%</td>
<td>24.04%</td>
<td>26.23%</td>
<td>28.96%</td>
<td>183</td>
<td>10%</td>
</tr>
<tr>
<td>Lack of experienced manpower</td>
<td>2.72%</td>
<td>16.85%</td>
<td>28.26%</td>
<td>32.07%</td>
<td>20.11%</td>
<td>184</td>
<td>10%</td>
</tr>
<tr>
<td>Limited suppliers/vendors to support and maintain</td>
<td>10.56%</td>
<td>17.78%</td>
<td>26.67%</td>
<td>27.78%</td>
<td>17.22%</td>
<td>180</td>
<td>10%</td>
</tr>
<tr>
<td>Unable to adequately support, maintain, or enhance inhouse</td>
<td>7.82%</td>
<td>20.11%</td>
<td>27.93%</td>
<td>26.26%</td>
<td>17.88%</td>
<td>179</td>
<td>10%</td>
</tr>
<tr>
<td>Incompatible with current and/or future technological environments</td>
<td>10.99%</td>
<td>12.64%</td>
<td>25.82%</td>
<td>27.47%</td>
<td>23.08%</td>
<td>182</td>
<td>10%</td>
</tr>
<tr>
<td>Too rigid to comply with new business requirements</td>
<td>10.50%</td>
<td>14.92%</td>
<td>20.44%</td>
<td>32.04%</td>
<td>22.10%</td>
<td>181</td>
<td>10%</td>
</tr>
<tr>
<td>Monolithic architecture</td>
<td>11.05%</td>
<td>22.10%</td>
<td>24.31%</td>
<td>28.18%</td>
<td>14.36%</td>
<td>181</td>
<td>10%</td>
</tr>
<tr>
<td>High risk of failure</td>
<td>30.56%</td>
<td>32.22%</td>
<td>17.22%</td>
<td>12.78%</td>
<td>7.22%</td>
<td>180</td>
<td>10%</td>
</tr>
<tr>
<td>Poor user interface</td>
<td>11.36%</td>
<td>28.41%</td>
<td>27.27%</td>
<td>23.30%</td>
<td>9.66%</td>
<td>176</td>
<td>10%</td>
</tr>
<tr>
<td>Too costly to maintain</td>
<td>9.34%</td>
<td>13.19%</td>
<td>26.37%</td>
<td>25.82%</td>
<td>25.27%</td>
<td>182</td>
<td>10%</td>
</tr>
</tbody>
</table>

7) Do you consider the programming language as one of the factors to decide if a system is legacy?

<table>
<thead>
<tr>
<th>Response (%)</th>
<th>Responses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>54.05</td>
<td>100</td>
</tr>
<tr>
<td>No</td>
<td>45.95</td>
<td>85</td>
</tr>
</tbody>
</table>

8) Can you give some examples of programming languages that you consider a legacy or are being used as a legacy in your organization?
**Legacy Modernization Projects**

**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?

<table>
<thead>
<tr>
<th></th>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>89.47%</td>
<td>153</td>
</tr>
<tr>
<td>No</td>
<td>10.53%</td>
<td>18</td>
</tr>
</tbody>
</table>

Answered Question 171, Skipped Question 40

10) Please rank which of the following driver(s) are the motivating factors to modernize legacy systems.

<table>
<thead>
<tr>
<th>Motivating Factor</th>
<th>Very weak</th>
<th>Weak</th>
<th>Strong</th>
<th>Very Strong</th>
<th>Responses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Become flexible to support changing business requirements</td>
<td>4.82%</td>
<td>11.45%</td>
<td>39.16%</td>
<td>44.58%</td>
<td>166</td>
<td>14%</td>
</tr>
<tr>
<td>Create new business opportunities via mergers &amp; acquisitions</td>
<td>16.56%</td>
<td>38.65%</td>
<td>28.83%</td>
<td>15.95%</td>
<td>163</td>
<td>14%</td>
</tr>
<tr>
<td>Reduce the cost of maintenance &amp; operations</td>
<td>5.39%</td>
<td>16.17%</td>
<td>41.92%</td>
<td>36.53%</td>
<td>167</td>
<td>14%</td>
</tr>
<tr>
<td>Limited suppliers/vendors to support legacy systems</td>
<td>7.32%</td>
<td>32.93%</td>
<td>44.51%</td>
<td>15.24%</td>
<td>164</td>
<td>14%</td>
</tr>
<tr>
<td>Lack of experts/documentation around legacy systems</td>
<td>6.63%</td>
<td>25.90%</td>
<td>48.80%</td>
<td>18.67%</td>
<td>166</td>
<td>14%</td>
</tr>
<tr>
<td>Prone to failures</td>
<td>32.34%</td>
<td>42.51%</td>
<td>18.56%</td>
<td>6.59%</td>
<td>167</td>
<td>14%</td>
</tr>
<tr>
<td>Faster time-to-market of product</td>
<td>9.52%</td>
<td>20.83%</td>
<td>39.29%</td>
<td>30.36%</td>
<td>168</td>
<td>14%</td>
</tr>
</tbody>
</table>

Others:
Enhanced capabilities (which is not the same as "become flexible")

For the most part, it is the PERCEPTION that Windows systems are cheaper that drives most legacy migrations. In actuality, they are NOT cheaper to support in the long run.

- Failing to adequately integrate with the ever changing systems/software landscape - failing to comply with new crosscutting standards such as security mechanisms within a SOA (very strong)

Use of modern development tooling (IDE's) Deployment modernization (DevOps) Increase scalability of processes (parallelisation) Agile development/deployment

Fix bugs. Just because a product is legacy does not mean it's bug-free!

Availability of hardware components (e.g. end of life, dead-end technology)

Hardware end of life, spare parts hard to find

Security vulnerabilities due to out of support platforms; lack of regulatory compliance due to out of support platforms (ie Visual Basic, Windows XP)
after April 8, 2014); requirements for hardware or operating systems that are no longer available (ex: DOS).

An important one is compliance. If the underlying technology is not supported by the original vendor, the main driver becomes compliance.

Politics

11) How challenging are the following obstacles in a legacy system migration projects?

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>Not Challenging</th>
<th>Less Challenging</th>
<th>Quite Challenging</th>
<th>Challenging</th>
<th>Very Challenging</th>
<th>I don't know</th>
<th>Responses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Migration</td>
<td>7.23%</td>
<td>19.88%</td>
<td>22.89%</td>
<td>25.90%</td>
<td>23.49%</td>
<td>0.60%</td>
<td>166</td>
<td>8%</td>
</tr>
<tr>
<td>Lack of resources (e.g. documentation, experts)</td>
<td>1.23%</td>
<td>19.63%</td>
<td>17.18%</td>
<td>30.67%</td>
<td>30.06%</td>
<td>1.23%</td>
<td>163</td>
<td>8%</td>
</tr>
<tr>
<td>Poor system architecture or infrastructure (e.g. monolith, hardcoded, spaghetti architecture)</td>
<td>4.85%</td>
<td>15.15%</td>
<td>20.61%</td>
<td>27.88%</td>
<td>30.91%</td>
<td>0.61%</td>
<td>165</td>
<td>8%</td>
</tr>
<tr>
<td>Difficult to extract business rules/knowledge</td>
<td>5.39%</td>
<td>13.77%</td>
<td>17.37%</td>
<td>31.74%</td>
<td>30.54%</td>
<td>1.20%</td>
<td>167</td>
<td>8%</td>
</tr>
<tr>
<td>Difficult to test</td>
<td>6.59%</td>
<td>22.16%</td>
<td>27.54%</td>
<td>25.15%</td>
<td>17.96%</td>
<td>0.60%</td>
<td>167</td>
<td>8%</td>
</tr>
<tr>
<td>Resistance from the current users/maintainers in the organization</td>
<td>4.76%</td>
<td>27.38%</td>
<td>22.02%</td>
<td>19.64%</td>
<td>26.19%</td>
<td>0%</td>
<td>168</td>
<td>8%</td>
</tr>
<tr>
<td>Cultural resistance in organization not to adapt new system</td>
<td>10.71%</td>
<td>18.45%</td>
<td>26.19%</td>
<td>22.62%</td>
<td>20.83%</td>
<td>1.19%</td>
<td>168</td>
<td>8%</td>
</tr>
<tr>
<td>Difficult to effectively prioritize the functionality for modernization</td>
<td>3.59%</td>
<td>20.36%</td>
<td>30.54%</td>
<td>25.75%</td>
<td>14.37%</td>
<td>5.39%</td>
<td>167</td>
<td>8%</td>
</tr>
<tr>
<td>Difficult to communicate the reasons/consequences of modernization</td>
<td>10.24%</td>
<td>27.11%</td>
<td>22.89%</td>
<td>23.49%</td>
<td>11.45%</td>
<td>4.82%</td>
<td>166</td>
<td>8%</td>
</tr>
<tr>
<td>Funding legacy modernization projects</td>
<td>2.98%</td>
<td>14.88%</td>
<td>22.02%</td>
<td>28.57%</td>
<td>27.38%</td>
<td>4.17%</td>
<td>168</td>
<td>8%</td>
</tr>
<tr>
<td>Time constraints to finish legacy modernization</td>
<td>2.99%</td>
<td>13.17%</td>
<td>17.96%</td>
<td>34.13%</td>
<td>30.54%</td>
<td>1.20%</td>
<td>167</td>
<td>8%</td>
</tr>
<tr>
<td>Predicting Return of Investment (ROI) of modernization</td>
<td>4.76%</td>
<td>10.12%</td>
<td>19.64%</td>
<td>27.98%</td>
<td>31.55%</td>
<td>5.95%</td>
<td>168</td>
<td>8%</td>
</tr>
</tbody>
</table>

Other:

Retraining existing IT-personnel

Suboptimal code quality due to offshore code migrations (e.g. copy&paste vs. reuse of proprietary frameworks)

Selecting the correct migration approach according to the needs of the stakeholders (it involves deciding whether the system is suitable for modernization or replacement): Challenging

Multiple wrap-ups are built up over time and it’s hard to understand what would happen or what other systems would be affected (andso what stakeholders), if modernization projects is implemneted.

From the legacy migration projects I've been involved in, the ROI is negative. Yes, negative.

Just a motivation: data incompatibility is the primary problem. Software functions (methods, api interface definitions) are tightly coupled with the datamodels. As time progresses the legacy system will be expanded upon without much consideration for the integrity of the primary datamodels. ("things have allways been done so"). As such the functions will recieve additional, oftentimes anonymous, side-effected behavior.

A lot depends on the approach for your modernization. The approach I have used mitigates a lot of the challenges/risks that are usually part of the choices made...

I have trouble with your column "quite challenging". To me, "quite challenging" would be more challenging than "challenging", yet this column is located in a less-challenging area. Perhaps you meant to say "somewhat challenging"?

Handle expectations of end-users. The new system must at least contain all the existing functionality.

It is important to evaluate the impact of deploying the modernized system both from a technical and user perspective.

12) Please indicate which of these factors are responsible for postponing legacy modernization. (Please check all that apply)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huge investment</td>
<td>24.73%</td>
<td>138</td>
</tr>
<tr>
<td>Less changes in core business process</td>
<td>8.06%</td>
<td>45</td>
</tr>
<tr>
<td>Difficult to find preferable / customize software</td>
<td>6.09%</td>
<td>34</td>
</tr>
<tr>
<td>No profit made during modernization project</td>
<td>10.22%</td>
<td>57</td>
</tr>
<tr>
<td>New system might behave differently</td>
<td>11.65%</td>
<td>65</td>
</tr>
<tr>
<td>Others:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lack of resources (experts) - they should cover both legacy and modernized systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not easy to convince organisation that the costs are earned back on the longer term view by less testing, less field reports, less bugs, faster extension of features. Going for short term wins</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clients should never get impacted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up to now the most new systems offer fancy devices and features but do not provide proven stability, security, and reliability.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not clear cost-benefit ratio.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time it takes to make the transition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main reason is a combination of budget and unforeseen behavior COMBINED. As the system is expanded upon all manners of uncontrollable behavior is added into the system, also there will be many non-standard organisation-specific functionality hacks within the system. There is just no way that some off-the-shelf system will come even remotely close to the datastructures and software functions of the original. Refactoring efforts are underestimated with a false sense of &quot;we know what we need&quot;. These projects are VERY prone to fail as they will not be setup and run with the required vigor and vision. Refactoring projects require a strong controlled vision/architecture without the interference of the business. RFCs can be added later, but not during the &quot;technical&quot; decomposition and reconstruction of the technical legacy application.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Again, the approach is very important. With the approach I have used, a lot of these items present no issues for the migration.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The initial system took a long time and cost lots of money due to poor management of the project and lack of developer skills. Ever since then there has been a fear of undertaking another rewrite of the application. Ironically newer versions (4 to date) have taken as long as a rewrite just to update to newer user interfaces/environments because of the lack of the complexity in the application. One investment in time/money to rewrite the application would have alleviated the 3 updates that took at least 2 years each to bring the application into current technology/web etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Those who have created the legacy product have moved on to new positions (or possibly even different companies).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Page: 4/7**

**13)** Do you use any academic resources while performing legacy modernization projects? (for example: research articles, tools, etc)  

<table>
<thead>
<tr>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>46.82</td>
</tr>
<tr>
<td>No</td>
<td>53.18</td>
</tr>
</tbody>
</table>

**Page: 5/7**

**Legacy Modernization Project**

**14)** What resources do you use from academia?  

<table>
<thead>
<tr>
<th>Response (%)</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research articles</td>
<td>37.75</td>
</tr>
<tr>
<td>Research software tools</td>
<td>27.81</td>
</tr>
<tr>
<td>Research collaboration with academics</td>
<td>19.87</td>
</tr>
<tr>
<td>Internship students</td>
<td>14.57</td>
</tr>
</tbody>
</table>

**Other (specify)**  
Anything I can find to support the argument to modernise.  
We develop tools ourselves (we are tool vendor) and these are developed in cooperation with University of Leipzig  
I wrote a thesis on data migration before effectively working on a data migration platform. I’ve read a lot of articles on the subject, however it is no longer part of my day to day. Our department uses interns to research various topics that are of value to our modernization efforts.  
Experiences with other organisations
many many articles about computation and complexity. Mainly information related to functional programming (Hurray for Jeroen Fokker and Eric Meijer)

Gartner, etc. to justify approaches and vendors with whom to work

---

**Page: 6/7**

**Legacy Modernization Project**

15) If your answer to Question 13 (Do you use any academic resources while performing legacy modernization projects?) is No, please specify some reasons.

<table>
<thead>
<tr>
<th>Text Answers (15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of adequate offering from University/Research to modernization challenges. To much focus on technological/theoretical techniques. Modernization is seen as a technical problem, not a socio-business one.</td>
</tr>
<tr>
<td>Our company develops Modernization technologies and we use them to help our customers to rescue their current investments.</td>
</tr>
<tr>
<td>We use internally developed tools; some of these originally came from academia 20 years ago.</td>
</tr>
<tr>
<td>No articles known.</td>
</tr>
<tr>
<td>don't need them</td>
</tr>
<tr>
<td>Don't really know what is meant by this question.</td>
</tr>
<tr>
<td>I'm not sure if the tools/products we use is in its origin academic? I think some must be but I can't say which.</td>
</tr>
<tr>
<td>I always work with the applications which are build to replace the legacy systems.</td>
</tr>
<tr>
<td>Never had in my mind</td>
</tr>
<tr>
<td>Never thought of.</td>
</tr>
<tr>
<td>The modernization process, for our company M Corp, is a structured and method driven approach. Academic organizations don't add specific value to the modernization challenge other than addressing why modernization projects fail or generate less value than anticipated. Those studies/analysis are what fuels the consideration of how to modernize.</td>
</tr>
<tr>
<td>not available at the time !</td>
</tr>
<tr>
<td>No need</td>
</tr>
<tr>
<td>Never thought of it...</td>
</tr>
<tr>
<td>First of all, how do you define ‘legacy modernization’? To me, that means keeping the legacy product, but making minor changes to improve it. It’s not a huge project to modernize it; just fixing bugs in existing systems.</td>
</tr>
</tbody>
</table>

---

**Page: 7/7**

16) How can academics play a larger role in industrial legacy modernization projects?
Listen to markets, as you are doing with this survey

Join my fledgling Legacy Guild at www.legacyguild.org. A newly created site dedicated to the support, maintenance, currency, enhancement, preservation, and transformation of Legacy Systems, with emphasis on modernization with Service Oriented Architecture (SOA).

They should work more with organization like ours who perform modernization projects every day. There is not a good academy-industry relationship.

Academia could help by clarifying the real costs and risks of various approaches to software modernization, including: 1. Calculating the financial risks of running legacy or obsolete systems 2. Understanding the real risks and costs of different mitigation strategies, including: a) Rewrite from scratch b) Migrate to modern code base with automated tools c) Run on modern platform using virtualization or runtime libraries d) Replace with commercial off-the-shelf software (COTS) and customize e) Piecemeal component modernization and/or wrapping.

Make articles with direct applicable solutions, which can be used asap.

teach cobol

Educate young people in the world of the legacy systems and modernisation.

Smart solutions to data migration, reuse 'legacy' test base for 'as is' rebuilt of the legacy system in different environment. Conversion tooling for program languages?

If organisations now what the have to do. such as look what you want the system to do, and not what is the system doing. You need with legacy systems a different approach then with new applications. What are the steps to get ride of the legacy systems in steps.

In the first instance, communicating their experiences more easily.

Address details on why projects suffer in the modernization efforts

creating and supporting of frameworks and standards \ governance bodies for adoption !

Not sure

a) overall thought leadership - what are viable approaches and what are not b) practice research - in depth studies of modernization efforts and the outcomes (we know that 50% of new dev projects fail; what is it for modernization) c) modernization framework (like TGAF or CMM) that allows CIO’s to convey where a company is on the “legacy” curve; d) Translate “legacy risk” into “operational risk” - case studies and databases of operational losses resulting from legacy software that can be used to justify the investment in modernization

Does academics have experience in this area? How many legacy systems have they migrated or use?

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.)

<table>
<thead>
<tr>
<th>Text Answers (109)</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:leen.blom@centric.eu">leen.blom@centric.eu</a></td>
</tr>
<tr>
<td><a href="mailto:mleurs@deloitte.nl">mleurs@deloitte.nl</a></td>
</tr>
<tr>
<td><a href="mailto:erik.hegeman@gmail.com">erik.hegeman@gmail.com</a></td>
</tr>
<tr>
<td><a href="mailto:hjige@synthesis-mc.de">hjige@synthesis-mc.de</a></td>
</tr>
<tr>
<td><a href="mailto:victor.duncker@gmail.com">victor.duncker@gmail.com</a></td>
</tr>
<tr>
<td><a href="mailto:g.stoop@noord.amsterdam.nl">g.stoop@noord.amsterdam.nl</a></td>
</tr>
<tr>
<td><a href="mailto:amir.g@mosttechnologies.com">amir.g@mosttechnologies.com</a></td>
</tr>
<tr>
<td><a href="mailto:v.stecher@lijbrandt.nl">v.stecher@lijbrandt.nl</a></td>
</tr>
<tr>
<td><a href="mailto:reg.mulder@delta-software.com">reg.mulder@delta-software.com</a></td>
</tr>
<tr>
<td><a href="mailto:maarten@veger.eu">maarten@veger.eu</a></td>
</tr>
<tr>
<td><a href="mailto:andrei.idu@gmail.com">andrei.idu@gmail.com</a></td>
</tr>
<tr>
<td><a href="mailto:james.roberts@cgi.com">james.roberts@cgi.com</a></td>
</tr>
<tr>
<td><a href="mailto:mitch@lcmg.us">mitch@lcmg.us</a></td>
</tr>
<tr>
<td><a href="mailto:hugo@vmgcompany.nl">hugo@vmgcompany.nl</a></td>
</tr>
<tr>
<td><a href="mailto:yann1560@outlook.com">yann1560@outlook.com</a></td>
</tr>
</tbody>
</table>