How Do Professionals Perceive Legacy Systems and Software Modernization?

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

ABSTRACT

Existing research in legacy system modernization has traditionally focused on technical challenges, and takes the standpoint that legacy systems are obsolete, yet 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 industrial practitioners were interviewed on what makes a software system a legacy system, what the main drivers are that lead 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. The results show that practitioners value their legacy systems highly, the challenges they face are not just technical, but also include business and organizational aspects.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Restructuring, reverse engineering, and reengineering*

General Terms

Experimentation

Keywords

Legacy Systems; Legacy Modernization; Grounded Theory; Empirical Studies

1. INTRODUCTION

A legacy system is any business critical software system that significantly resists modification and their failure can have a serious impact on the business [8, 11]. Software modernization is the process of evolving existing software systems by replacing, re-developing, reusing, or migrating the software components and platforms, when traditional maintenance practices can no longer achieve the desired system properties. The primary aim of software modernization is to reduce maintenance cost and increase flexibility. After three decades of legacy system modernization research, it may come as a surprise that many legacy systems are still in daily operation. Most of these systems were developed years ago, and have continued to evolve. New requirements have led to frequent modifications of these legacy systems resulting in unstructured source code that is difficult to maintain. Furthermore, knowledge about the legacy systems is scarce as original programmers leave the company or retire, and documentation is usually lacking [50].

These issues have been recognized by the software engineering community and a plethora of legacy system modernization approaches have been proposed (cf. Section 5). Despite the problems introduced by legacy systems, and the acclaimed benefits of legacy system modernization, technology consulting firms estimate that 180-200 billion lines of legacy code are still in active use [5, 45, 50]. This fact has triggered us to investigate how legacy systems and their modernization are perceived in industry. Specifically, we aim to identify the perceived benefits of legacy systems, the main drivers for legacy system modernization, and the challenges professionals face with the modernization of legacy systems.

We have set up our research as an exploratory study, aiming to discover new perspectives and insights about legacy systems in the industry. Accordingly, we adopted the grounded theory approach [21], an increasingly popular method to conduct empirical software engineering research [1]. We designed, conducted, and analyzed semi-structured interviews with 26 industrial practitioners. The findings of the interviews were then validated through a separate structured survey, in which 198 professionals expressed (dis)agreement with the results of the interview.

Our study makes 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 differences in perception of legacy systems between the industry and academia.

The paper is structured as follows. In Section 2, we describe our study design. In Section 3 we detail the key findings. Subsequently, in Section 4 we present the confirmation/contradiction of the findings and address threats to the validity. In Section 5 we discuss related work. Finally, in

ICSE'14, May 31 – June 7, 2014, Hyderabad, India
Copyright 2014 ACM 978-1-4503-2756-5/14/05...$15.00
http://dx.doi.org/10.1145/2568225.2568318

Section 6 we conclude our research and propose future research directions.

2. STUDY DESIGN

We employed two different research techniques to conduct our study. We started with semi-structured in-depth interviews, a qualitative technique, to identify how legacy systems and their modernization are perceived in industry. We followed a Grounded Theory (GT) approach [21] to analyze 26 in-depth semi-structured interviews. To further validate the findings of the GT, we used a separate structured survey– a quantitative technique. In an empirical study such as this, the use of multiple research techniques (qualitative and quantitative techniques in our case) increases the confidence that the results are reliable.

GT is an explorative research method that aims at discovering new perspectives and insights, rather than confirming existing ones [21]. We started 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 the two criteria that they have experience with legacy systems, and with legacy system modernization projects. The informants were identified opportunistically via industrial collaborators, followed by snowball sampling [27], in which the first generation informants helped us to identify other informants who fulfilled 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. Furthermore, two practitioners were from the same organization and this reduced the sample size of the participating organizations to 22. The sample is arguably broad enough to well represent the software engineering professionals. Moreover, the sample shows diversity in the industry domain, and the roles and experiences of the participants. Table 1 provides the details regarding the domain of each informant's company. With respect to size, the companies range from small consulting firms to global corporations such as IBM, Deloitte, and Capgemini. The variation among the informants' 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 experience in information technology. Additionally, the interview data totaled more than 25 hours of recorded data.

Table 1: Details of the informants

Participants	Domain
P2, P11, P12, P20, P21	Information Technology Services
P1, P15, P17, P22	Financial Service Providers
P4, P5, P25, P26	Government Organizations
P7, P8, P18, P19	Software Development Company
P6, P10, P24	Consulting Company
P3	Aviation Industry
P9	Manpower (Security) Company
P16	Flower Auction Company
P13	Food & Dairy
P23	Machinery Production
P14	Poultry

We conducted semi-structured face-to-face interviews in English, which were recorded. Prior to the interview session, informants were provided with an interview protocol that consisted of sample questions to be discussed during the interview sessions. The interview session has three categories of questions: about characteristics of legacy systems, drivers for legacy system modernization, and 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*. As the interviews progressively provided answers similar to the earlier ones, a *saturation stage* [1] 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[1] as an instrumentation tool to facilitate the interview analysis process.

In the second and final phase of this research, we adopted a structured survey as a data triangulation process in order to validate the findings of the interview results. A data triangulation process– a method that uses more than one data source, or collects the same data at different occasions– is typically used to increase (decrease) confidence in a finding by providing confirming (contradictory) evidence [20, 39] and helps to improve validity of the findings of an empirical study such as this [23]. The survey was public and announced via mailing lists, social media such as Twitter, Linkedin, Facebook, and via personal referrals. In the end, 198 participants responded to the survey, originating from more than 30 different countries. We performed sampling to exclude responses having no experience with legacy systems. In total, 22 out of 198 were excluded, leaving 176 valid responses. The respondents have an average experience of 13.5 years with legacy systems. Developers formed the largest group of participants (22%), followed by IT Managers (14%) and researchers (12%); they come from various domains such as software development companies (28%), consulting companies (21%), service providers (11%), and financial institutions (9%).

In the subsequent sections, we present the results of our research, categorized over the four themes: legacy systems, perceived benefits of legacy systems, drivers of legacy system modernization, and challenges of legacy system modernization. For each theme, we provide relevant "*quotes*" from the practitioners and the results of the survey.

For further details of the research, we refer to our technical report [6], in which we provide additional data to support our analysis. In the technical report, we provide the coding process, consisting of 44 different codes within 19 categories. For each code, we give a short working description. In addition, we detail key quotes of all the informants (P1–P26) to illustrate how the codes are derived. Furthermore, the technical report also presents the survey with final results along with response counts and percentages. The report also describes statistical analysis, particularly, multiple regression analysis to identify the relationships among the drivers of legacy system modernization and use the results in this pa-

[1] `www.qsrinternational.com/`

per to support our findings. To increase transparency of the data analysis process, we publicly provide anonymous interview transcripts, the Nvivo 10 project file of interview analysis, and the survey data in excel format[2].

3. FINDINGS

In the following subsections, we present the results of the analysis of the interview sessions and the results of the survey. For each interview, we started asking questions regarding their personal information and experiences with legacy systems. Subsequently, the informant was asked to give a definition for a legacy system based on his/her opinion and the findings are discussed in subsection 3.1. The discussion then proceeded by asking questions about perceived benefits of legacy systems; these findings are presented in subsection 3.2. Furthermore, the informants were asked about the issues associated with legacy systems, which are the drivers for legacy system modernization and such drivers are discussed in subsection 3.3. Moreover, the discussion proceeded to investigate the challenges faced during legacy system modernization; these findings are presented in subsection 3.4.

3.1 Legacy Systems

The interview sessions started by asking a definition of a legacy system from the informants. Most of the informants agreed that legacy systems are "old" systems. Additionally, the informants pointed out that legacy systems are "core" systems that have been proven to work correctly in a production environment for decades. P1 said: *"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 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 mixed opinion is also observed from the results of the survey. Around 50% of the respondents agreed that the programming languages do determine if a system is legacy. The top five languages that these informants indicated as legacy are depicted in Figure 1.

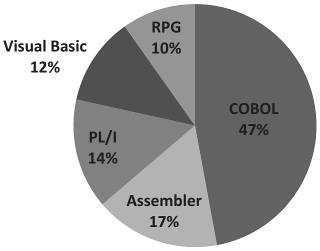

Figure 1: Legacy languages by as per the informants

3.2 Perceived Benefits of Legacy Systems

3.2.1 Business Critical

The interviews reveal that the practitioners view legacy systems as business critical. As per the informants, legacy systems are the core systems of the organizations and their failure can result in serious consequences for daily business. P11 argued that legacy systems are significant to business, taking an example of a financial organization. He expressed his opinion as *"It is very useful and has a business impact still and generates a lot of revenue for banking and their clients"*. P14, a 20 years experienced IT director of a poultry company, simply stated that *"Legacy for me is let's say business critical"*. P24 explained that legacy systems are old and business critical as: *"Because they [Legacy systems] have been there for 30 years, so they really are the foundation for the survival of the organization"*. These opinions clearly indicate that legacy systems support core business processes of a business and their failure can have significant impact on an organization.

The results of the survey also strongly support that legacy systems are business critical. 76.7% (cf. Figure 2) of the respondents indicated that legacy systems are typically business critical. One of the respondents with 28 years of experience gives his strong opinion through an open question of the survey as: "*By definition a legacy system is business critical. A system that is old and obsolete and is not business critical would never reach the status of legacy"*.

3.2.2 Proven Technology

Informants have frequently expressed that legacy systems are old and have been developed, tested and have been in production environment for years. Hence, it is an indication that legacy systems are of a proven technology that still remain as the core systems of many organizations. P17 explained the proven technology characteristics with an example of AS400 as: *"Most of the time it's [a] proven technology. AS400 is stable, it always works [24/7] and is quite good. So, it's proven technology and normally it's stable, [which] is a good thing"*.. In our discussion with P4, he expressed: *"Proven technology is often the reason why they are still in use"*. P11 associated "Availability" with the "Proven technology" and said: *"They are available, and they are more less 24/7 up and running"*.

In the survey, 52.8% (cf. Figure 2) of the respondents indicated that legacy systems are proven technology.

3.2.3 Reliable Systems

The definition of reliability is adopted from ISO/IEC 25010 standard [43] as "degree to which a system, product or component performs specified functions under specified conditions for a specified period of time". This definition was provided to the participants of interview and survey. Based on the interviews, the practitioners indicated that legacy systems are reliable systems, primarily, because they are running in a production environment for decades. P18 supports this statement and said: *"It is reliable...people know how to use it. All the problems have been fixed over the years from it. So technical problems are usually not there"*. Other informants also share P18's opinion as P1 stated: *"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]*

[2] http://www.servicifi.org

that count are stability, robustness, reliability and availability of this system". In general, legacy systems are perceived as reliable systems in industry because of the fact that they have been in production for years and possible bugs and errors are already fixed in the past. P12's opinion is also inline with the others and he said: *"They have been around for many years and during this period they have been stabilized".*

The findings of the survey are in agreement with the opinion of the practitioners regarding legacy systems being reliable systems. 52.3% (cf. Figure 2) of the respondents indicated that legacy systems are reliable systems.

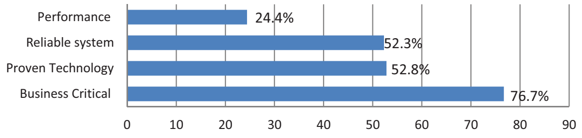

Figure 2: Survey responses for perceived benefits of the legacy systems

3.2.4 *Performance*

With regards to performance, the interview revealed a mixed opinion. Some of the informants (e.g., P3, P10, P24) strongly emphasize on the high performance characteristics of legacy systems. P3 expressed: *"It's [Performance] enormous. Enormous quick. So the old system itself didn't have [performance issues]...I don't think the performance is a problem".* P24 agreed with the opinion of P3 and further added: *"I think ten thousand people are doing airlines booking with this little processing power. So performance is never an issue in legacy system, at least I've never seen it".* In contrast, P9 and P18 perceive that legacy systems do not have high performance. P9, with reference to a real time system that is being used in his Manpower (Security) Company, referred: *"It [performance] is very poor".* P18, a software developer, agreed with P9 *"The performance is not really good".* There are informants (P13, P14, P9, P16, P26) who consider that performance of legacy systems as *"Ok or enough".*

In the survey, 24.4% (cf. Figure 2) of the respondents indicated that legacy systems are of (high) performance, which is comparatively low with respect to other characteristics. One of the plausible reasons could be the fact that legacy systems get their job done and operate at good/enough (satisfactory) speeds in many cases.

3.3 Drivers for Legacy System Modernization

The informants not only expressed their opinion about perceived benefits of the legacy systems, but also explained about issues related to the legacy systems.

3.3.1 *To Remain Agile to Change*

In current dynamic business environment, organizations have to quickly adapt to various changes, including intra-organizational changes, changes in laws and regulations, changes in business collaboration (mergers and acquisitions), and faster time-to-market [49]. Despite the fact that legacy systems are business critical and reliable systems, the informants expressed that legacy systems are inflexible to support changing business requirements. P20 explained how the inflexibility to adapt to new changes has enabled him to modernize legacy systems. He said: *"Other point is that my costumer 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".* P25, an IT architect at the tax office, agreed with P20 and said: *"We had a lot of systems before, and they were built in CICS, COBOL, DB2 and were not flexible. So they needed to be modernized to get more flexibility".* The informants identified faster time-to-market as one of the drivers for legacy system modernization because legacy systems are rigid, which increases the difficulty in promptly addressing the market demands. P22 expressed this as: *"We need a faster time-to-market, and we are not able to do that in COBOL environment".*

The findings of the survey are in-line with the opinions expressed by the interview respondents, in which respondents have strongly expressed that the need of flexibility to comply with the changing business requirements and rapidly evolving future technologies do cause organizations to initiate legacy system modernization projects. For the "Flexibility to change" driver, we depict the contributing drivers (i.e., "Become flexible to change", "Create business opportunities via mergers/acquisitions", and "Faster time-to-market") in Figure 3, as presented in the online survey. It is interesting to observe that 85% (38.1% for "Strong" and 46.9% for "Very Strong") of the respondents of the survey have indicated "Become flexible to change" a major driver. Similar observation can be noticed with "Faster time-to-market", for which 71.1% of the respondents indicated as a driver of legacy system modernization (39.6% for "Strong" and 31.5% for "Very Strong"). But in contrast, the "Create business opportunities via mergers/acquisitions" driver is indicated as a medium driver by 43.5% of the respondents.

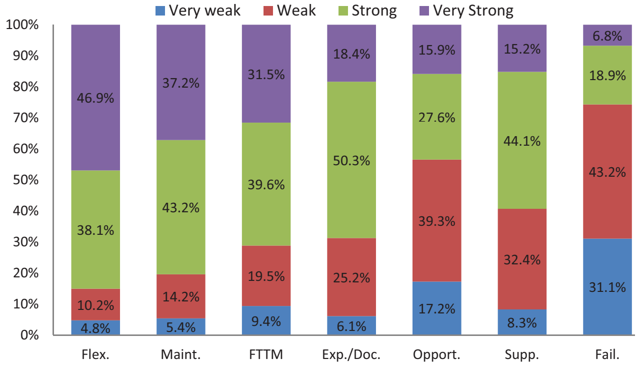

Figure 3: Drivers for Legacy System Modernization (*Legends: Flex.:–Become flexible to change; Maint.:–High cost of maintenance; FTTM:–Faster time-to-market; Exp./Doc.:–Lack of experts/documentation; Opport.:–Create business opportunities via mergers/acquisitions; Supp.:–Lack of suppliers/vendors; Fail.:–Prone to failure*)

3.3.2 *High Maintenance Cost*

The interviews revealed that one of the significant drivers of legacy system modernization is high maintenance cost. More than half of the informants strongly argued that the cost involved in maintaining the legacy systems is high. One of the most common motivations to modernize the legacy systems is to reduce maintenance cost. P17 explained: *"[With legacy systems] the cost is getting higher because maintenance is getting more expensive, [then] maybe you should think of modernization"*. Often participants such as P12, P13, P14 argued that maintenance cost could be lowered if the legacy systems were modernized to standard software products. For instance, P12 mentioned: *"But if you can move to standard product, then it could be usually an advantage because the maintenance cost for standard product is usually lower"*. P14 expressed: *"if you look at the [maintenance] cost, I'm quite sure I can run a 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"*.

From the survey, we observed that respondents (80.4%) strongly indicated (43.2% for "Strong" and 37.2% for "Very Strong") that the high maintenance cost of legacy systems is one of the major drivers behind legacy system modernization (cf. Figure 3).

3.3.3 Lack of Knowledge

As per the participants, one of the most significant drivers of legacy system modernization is a lack of knowledge, particulary scarcity of experts, unavailability of (up-to-date) documentation and limited number of suppliers/vendors of the legacy system. More than 90% (24/26) of the informants pointed out that lack of resources of the legacy systems causes organizations to modernize their legacy systems. P10 argued that: *"I think the big problem is that you can't find people to understand them [legacy systems] and understand the technology"*. P11 not only indicated lack of experts, but also pointed out the outdated documentation and explained: *"The issue is that there is less knowledgeable people [experts] are 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"*. Some informants (P1, P2) expressed a fear that in future the scarcity of the experts of legacy systems will be a severe problem. As to the limited number of suppliers/vendors, 12/26 informants indicated it as a problem. P17 expressed: *"There's no patches. If suppliers stop their product, organization needs to find another way to keep supporting their system"*.

In Figure 3, we depict the "Lack of knowledge" driver comprising of "Lack of experts/documentation", and "Limited suppliers/vendors", as presented in the online survey. Almost 60% (44.1% for "Strong" and 15.2% for "Very Strong") of the respondents indicated that "Lack of suppliers/vendors" is a strong driver for modernization. Furthermore, 68.7% of the respondents indicated "Lack of experts/documentation" as a "Strong" (50.3%) and "Very Strong" (18.4%) driver for modernization (cf. Figure 3).

3.3.4 Prone to Failures

Although "Reliable system" is one of the perceived benefits of the legacy systems, there is a fear shared by most informants that the legacy system might fail due to lack of experts and suppliers/vendors. Informants have identified that "Prone to failure" as a driver of legacy system modernization. Legacy systems are "business critical" and organizations cannot afford their legacy systems to fail. P13 expressed this as: *"We have an old ERP system, old almost 10 years old. And it drives the production in the plant and also the logistic and warehouse and also the order towards the customers. If that system stops, the plant stops, the warehouse stops"*. Often, the informants indicated that risk of failure can be the result of other drivers such as limited suppliers/vendors, and lack of experts. Such opinion is shared by P2 as: *"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"*.

In comparison with other drivers, "Prone to failures" is indicated as the weakest driver by 25.7% of the respondents (cf. Figure 3). One of the plausible reasons for this could be the fact that the "Prone to failure" issue is countered by the "Reliable system" perceived benefit of the legacy systems.

3.4 Challenges of Legacy System Modernization

Upon identifying the drivers of legacy system modernization, the study focused on identifying what challenges are faced by practitioners while modernizing legacy systems. Based on the interviews, the following challenges were identified.

3.4.1 Time Constraints to Finish Modernization

The interviews revealed that finishing any legacy system modernization on time is the biggest challenge. The timing constraint is influenced by other existing issues of legacy systems such as scarcity of resources such as experts and documentation. P1 expressed the challenge as *"They run out of budget.. they run slightly out the time. [...] that's mainly to do with scarcity of people on the legacy system"*. P13 also agreed with P1 and stated that the time constraint to finish modernization project is influenced by lack of resources. He expressed this as: *"Your biggest problem is an availability of resources [documentation and experts] and availability of money and [to some] extent availability of time"*. Due to the long duration of the legacy system modernization projects, many earlier plans regarding the modernization change and this further delays the projects. P7 shared his experience as: *"Legacy modernization project lasts too long. We plan modernization for 3 years, and after 5 years we stop the whole modernization, and start it over"*. Some of the respondents indicated that lack of resources is also one of the factors of not finishing legacy system modernization projects on time.

Interestingly, by far the largest percentage (63.6%) of the respondents indicated that legacy system modernization projects frequently face time constraints to finish (31.8% for "Challenging" and 31.8% for "Very Challenging").

3.4.2 Data Migration

The informants also revealed that data migration in legacy system modernization project is also challenging. P11, who was conducting a modernization project in an insurance company, stressed the importance of the data migration as: *"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"*. Some of the informants indicated that the difficulty of data migration is due to the old databases that the legacy systems use. P17 explained this as: *"Data migration is really difficult because legacy system doesn't support modern databases or doesn't have relation database model"*. To mitigate the risk of legacy system modernization project failure, P14 suggested planning for data migration upfront.

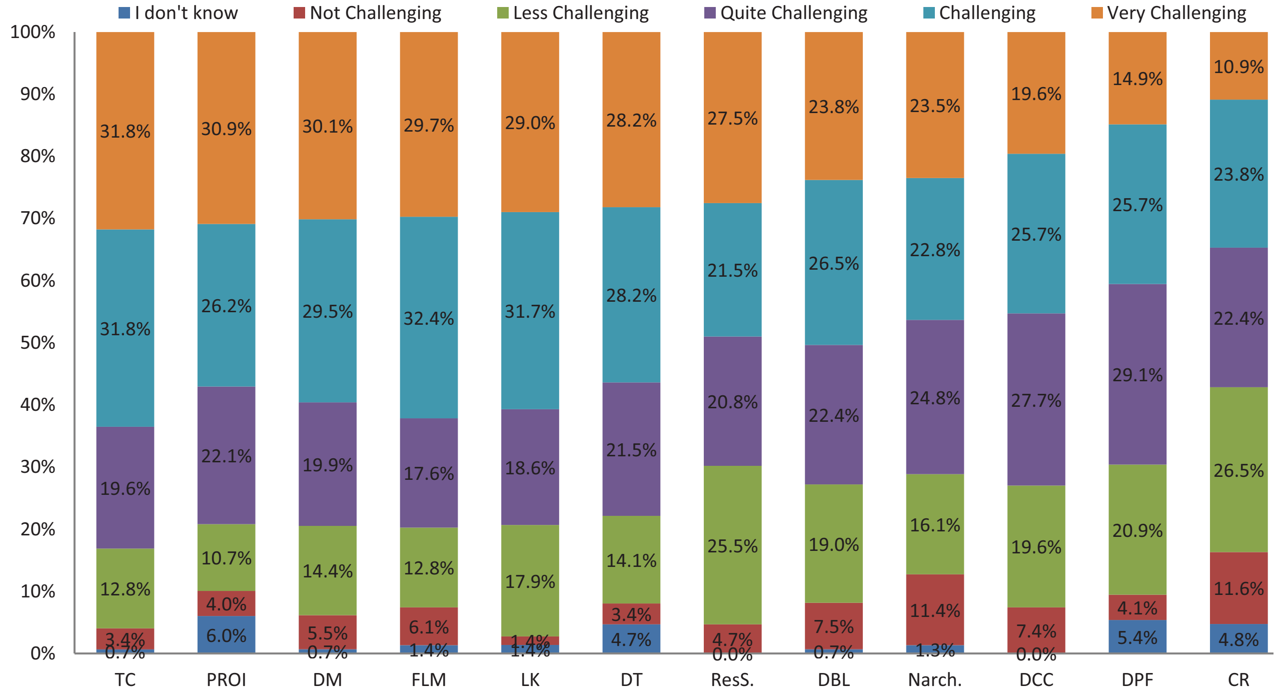

Figure 4: Challenges of Legacy System Modernization (*Legends: TC:–Time constraint to finish modernization; PROI:–Predicting ROI; DM:–Data Migration; FLM:–Funding modernization project; LK:–Lack of knowledge; DT:–Difficult to test; ResS.:–Resistance from staff; DBL:–Difficult to extract business logic; Narch.:–Non-evolvable system architecture; DCC:–Difficult to communicate the consequences; DPF:–Difficult to prioritize the functionality; CR:–Cultural resistance from organization*)

He said: *"If you are doing migration of your legacy, you have to prepare a good data migration strategy"*.

59.6% of the respondents of the survey agreed that data migration is one of the challenges of legacy system modernization projects in industry (cf. Figure 4).

3.4.3 Complex System Architecture

Dealing with the complex system architecture of legacy systems is one of the important challenges that the practitioners face while modernizing legacy systems. In the interviews, the informants used "Poor architecture", "Monolithic architecture" or "Ill–designed" to express complex architecture. P1 expressed his opinion about system architecture as: *"If you have ill-designed legacy system or old fashioned design as a monolith system, that might still be a challenge"*. The informants frequently expressed that the complex architecture is a result of ad-hoc maintenance/upgrade of the past. P9 explained this as: *"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"*. P2 further illustrated the challenge as: *"And what you also find is that one application is developed that it uses also the database of another one. And it's not through a normal interface but it's via back door to get some data over there"*. P18 said: *"Because the program is really hard-coded, it is not configurable that makes modernization difficult"*. Furthermore, the result of such complex architecture has an impact on testing during modernization. P12 argued that testing is a challenging during modernization because: *"It can be difficult to extract all the use cases to test. 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"*.

In Figure 4, we represent the two contributing challenges of "Complex system architecture" (i.e., "Non-evolvable architecture", and "Difficult to test") individually, as presented in the online survey. Cumulatively, "Non-evolvable architecture" is perceived as a serious challenge by 46.3% of the respondents while 56.4% expressed the same for "Difficult to test".

3.4.4 Lack of Knowledge

The informants from the interview perceived lack of knowledge of the legacy systems as one of the important challenges. P11 explained this as: *"It is complex, because you have a lot of different expertise needed to modernize such as people who understand database environment and operating systems, middleware, enterpriser services bus of architecture. You also need people who understand business functionality to transform business functions from legacy systems to another system"*. Additionally, lack of knowledge, particularly documentation can be a big risk in legacy system modernization as P20 said: *"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"*. Nevertheless, P18 indicated the use of reverse engineering techniques to mitigate the lack of (up-to-date) documentation challenge. He explained *"For the old systems,* *we reverse engineer the old system and document them well so we know what the requirements are and can help to build new system".*

As per survey results, 60.7% of the respondents have indicated that lack of knowledge is either "Challenging" or "Very Challenging" for legacy system modernization.

3.4.5 Difficult to Extract & Prioritize Business Logic

As an effect of lack of knowledge, identifying business logic within legacy systems becomes more challenging. The informants revealed that extracting business logic is one of the challenges in legacy system modernization. P20 expressed this as *"To extract all the rules and details in there [legacy systems] is really difficult".* P12 expressed his experience as: *"The company or the project team has to extract exactly the internal functionality of this legacy application. It can be difficult to extract, to document, and to implement it [business logic] properly".* P12 further suggested using automated techniques to initially extract diagrams to better understand the business logic. He explained this as: *"So I think it [business rules] can be really helpful to provide insight into the internal working of the system, to extract it to a human readable diagrams or documentation".* Additionally, P18 said: *"Yes, you can only do that if you know exactly what the thing does. That is [business logic extraction] is the hard part".* Within the critical business processes supported by the legacy systems, prioritizing the extraction of such business logic is also equally important.

Note that, we depict the "Difficult to extract & prioritize business logic" challenge comprising of "Difficult to extract business logic", and "Difficult to prioritize the functionality", as presented in the online survey. As per the results of the survey (cf. Figure 4), 50.3% of the respondents agree that extracting business logic from the legacy systems is an obstacle (26.5% for "Challenging" and "23.8% for "Very Challenging") and 40.6% agree that prioritizing the functionality (25.7% for "Challenging" and "14.9% for "Very Challenging").

3.4.6 Resistance from Organization

Resistance to change from the organization (including users and technical staff of legacy systems) towards a new technology is a well-documented phenomenon as they fear that their expertise and professional experience with legacy systems may become redundant due to modernization. Surprisingly, not only is there resistance from technical staff, but the informants also reported that cultural resistance to adapt to new technology is another challenge in legacy system modernization. P12 explained the resistance from staff as: *"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".* Most of the informants observed that the resistance of the existing technical staff is due to job security. P6 explained this as: *"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?"* Regarding cultural resistance, informants identified that adapting to new technology is difficult for the staff. P11 strongly emphasized that cultural adaption is one of the serious challenges. He stated: *"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]".*

For the "Resistance from organization" challenge, we depict the challenges individually (i.e., "Resistance from staff", and "Cultural resistance of organization") in Figure 4 as presented in the online survey. 49% of the respondents indicated that "Resistance from staff" is a challenge (21.5% for "Challenging" and 27.5% for "Very Challenging"). Similarly, "Cultural resistance from organization" is indicated by 34.7% of the respondents with 23.8% for "Challenging" and 10.9% for "Very challenging".

3.4.7 Addressing Soft Factors of Modernization

In this study, we adopted the concept of soft factors in modernization from Murer et al. [33] that considers non-technological factors such as people, communication, and business values of organization. The practitioners identified three soft factors as challenges. First, communicating the reasons/consequences of legacy system modernization, followed by securing funding from the top management. Often the informants indicated that the earlier challenge has an implication on the latter. P10, a legacy system modernization consultant, expressed his view as: *"I think top management doesn't understand the issue and they don't give budget for it [legacy modernization]".* Additionally, the informants indicated that communicating the consequences of the modernization is also a challenge that is often centered on the return of investment (ROI). P20 explained the difficulty of predicating ROI as: *"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".* An effective way to convince the top management is to identify appropriate business cases and explain the need/consequences of legacy system modernization. P19 explained the need to use business cases as: *"You have to somehow come up with the business case that says what is my current cost, what is the cost of migration, what the new total cost of ownership, and that you have to predict the Return of Investment. A business case can have soft components like improve maintenance or improve performance because they represent business value".*

For the "Addressing soft factors of modernization" challenge, we depict the contributing challenges individually (i.e., "Predicting ROI of modernization", "Difficult to communicate consequences of modernization", and "Funding modernization projects") in Figure 4 as presented in the online survey. 57.1% of the respondents indicated that "Predicting ROI of modernization" is a challenge (26.2% for "Challenging" and 30.9% for "Very Challenging"). Similarly, "Difficult to communicate consequences of modernization" is indicated as a challenge by 45.3% of the respondents with 25.7% for "Challenging" and 19.6% for "Very challenging". 62.1% of the respondents expressed that securing funding for modernization is a challenge (32.4% for "Challenging" and 29.7% for "Very Challenging").

4. DISCUSSION

A number of pervasive findings have emerged from the current research regarding legacy systems and their modernization, many of which are new and surprising while some resonate with the findings of other researchers. This section discusses the confirmation/contradiction of the findings with the academic research and reflects on threats to validity of this research.

4.1 Core System vs. Obsolete System

Evidence from current research indicates that practitioners perceive legacy systems as core systems, rather than obsolete systems– as generally perceived in academia. Practitioners do agree that there exist many problems with legacy systems such as they are inflexible, and costly to maintain. Notwithstanding these issues, legacy systems are crucial to the execution of day-to-day business processes of organizations whatever the business domain they operate in. Legacy systems are often the "back office" systems, which are rigid and inflexible compared to the "front office" systems. For instance, in financial business, COBOL is predominantly used as "back office" systems to process millions of batch transactions per day, which is crucial to any financial institution. The practitioners perceive legacy systems as crucial systems as they are business critical, reliable, and have been running in production for decades. Within the life cycle of the legacy systems, they have been well tested and practically run without errors to execute business processes.

Surprisingly, the main consideration for practitioners to determine if a system is legacy depends on whether the functionalities of the system are still in-line with the business-IT alignment of the organization. This finding contrasts with the observations made in academia, in which factors such as inflexibility [11, 15], expensive to maintain [11, 8], and even use of obsolete programming languages [8, 13] are frequently used to decide if a system is legacy. However, the findings of this study stand in contrast with this. We observed mixed results from the interview (53%) and survey findings (54.1%) who agree that programming language is one of the factors to determine if a system is legacy. Such a mixed observation has led us to further investigate if there is any interesting correlation. Consequently, we performed a Pearson Chi-square test to check if there exists any association between the role of the respondents (7 different categories) and the choice of programming language as a deciding factor for a system being legacy. The analysis with (ρ= 2.9), which is far less than the critical value (α= 12.59), revealed that such an association does not exist. Further, it is interesting to observe that the practitioners could not formulate a concrete definition for a legacy system– a frequently observed issue in academia (e.g., Alderson & Shah [3], Cornelissen et al. [18]).

4.2 Legacy System is a Cash Cow

As identified by Adolph [2], the findings of this study justify the notion of cash cow used for legacy system– systems that have been bringing in revenue for years for organizations. Despite various problems, this study revealed that practitioners perceive legacy systems as critical systems that execute the day-to-day business processes. One of the findings with respect to the "business critical" characteristic conforms with the observation made by Brodie [13]. However, other characteristics such as "reliable system", "proven technology" are merely discussed in academia. In general, legacy systems are presented as a quagmire in academia, often reporting about the problems associated with them and hence, urge for their modernization. But, in industry the legacy systems are still perceived as the "backbone" system of organizations. These systems are developed, tested and have been processing millions of records for decades on a daily basis. Apart from the "business critical" characteristic, the reliability of the legacy systems is the other predominant factor that keeps legacy systems still alive in industry. Frequently, the informants of the interviews indicated that legacy system are cash cows as: "*generate a lot of revenue for banks*"– P11, "*the most profitable systems*"–P7, and "*great business value"*–P10.

4.3 If It Ain't Broke, Don't Fix It

The aphorism "If it ain't broke, don't fix it" suitably captures the belief among practitioners. Not only in the interviews, but the respondents of the survey also stated in the "Others" section of the survey that if the legacy systems are working "well", then legacy system modernization projects are unlikely to be initiated despite various problems. Some of the responses include "*they have been working- why fix it?"* by a developer with 17 years of experience with legacy system, and "*we didn't fix it last year, and survived. Why should this year be different?*" by a chief technical officer of a software development company.

One of the most significant problems raised by the practitioners is scarcity of knowledge, including legacy experts and (up-to-date) documentation of the legacy systems. Evidence in academia shows that lack of knowledge is one of the enablers for legacy system modernization [11, 26, 3, 8]. Apart from issues with documentation, erosion of soft knowledge [33]– existing knowledge in the form of skills and experiences within the technical staff (original developers, maintainer, users)– is also another problem that has triggered legacy system modernization in industry. Over the years, soft knowledge has become scarce resulting in knowledge erosion due to factors such as ageing, retirement, and staff changing jobs. Furthermore, Khadka et al. [26] report on how various mergers and acquisitions of the supplier of Tandem NonStop based COBOL has triggered legacy system modernization in a bank. Additionally, the other significant driver for legacy system modernization identified by practitioners is "high maintenance cost"– a well reported issue of legacy systems in academia [15, 11, 8, 50].

With respect to drivers of the legacy system modernization, the findings of our study are in-line with the problems reported in academia.

4.4 Business vs. Technical Aspects: A Tale of Two Perspectives

Surprisingly, the study findings reveal that the challenges of legacy system modernization in industry largely emerge from the business perspective and also complements most of the technological challenges reported in academia. Challenges such as "Funding modernization projects", "Resistance from organization", "Predicating ROI", "Timing constraints to finish" strongly relate to business perspectives of legacy system modernization. This observation also holds true for the drivers of modernization. In particular, the "Become flexible to change" driver is identified by 85% and the "Faster time-to-market" by 68% of the respondents in the survey reflect the importance of the business aspect of legacy system modernization. Furthermore, the need for a business case to communicate the necessity and consequences of modernization to the top management also reflects the importance of business aspects of modernization. Such evidence clearly indicates that legacy system modernization is not just a technical endeavor, but also a business endeavor. Lately, legacy system modernization research has also focused on considering the business and organization aspects together with the technical aspects (e.g., Murer et al. [33], Nasr et al. [35]).

4.5 Limitations and Threats to Validity

Adolph et al. [1] argues that explorative qualitative research is often viewed with discomfort in software engineering with regards to research validity, and verification as assessing the validity of qualitative research is a challenging task [23]. In the following subsections, the most relevant threats to validity are discussed.

4.5.1 *Credibility*

Credibility refers to the fact that the findings of the research are free from any potential research bias. As to mitigate the internal validity and to strengthen the credibility of the research, various measures have been adopted. To start with, sampling of the informants for the interview and survey was performed in which only the informants having experiences with legacy systems and legacy system modernization were included. Furthermore, the research closely followed the GT guidelines, including careful coding, memoing and categorization. Each coding and categories were cross-validated with the other researchers in the process of developing. Additionally, the results of the interviews were triangulated with an online survey filled in by 198 respondents. To increase the transparency of the research, various artifacts such as interview records, interview transcripts, interview protocol, details of the coding process and the results of the survey are made available[3].

4.5.2 *Generalizability*

Generalizability is concerned with to what extent the findings can be generalized. One of the potential threats to the external validity is the fact that all the interview informants are from the Dutch industry. This choice may somehow bias the results. However, the distribution of the informants includes different company sizes, different level of experience, different roles, and variation in the domain of the companies (refer Table 1) arguably increases the generalizability of the findings. Furthermore, the findings of the research have been validated by respondents from more than 30 different countries in the survey, which also increases the confidence of generalizability.

5. RELATED WORK

This section describes related work aimed at assessing legacy systems and legacy system modernization along with empirical legacy system modernization research.

5.1 Legacy Systems and Their Characteristics

Legacy systems have been researched over the past three decades. The seminal work in this area is the first law of continuing change by Belady & Lehman [7], followed by the concept of preventative maintenance [29]. Both of these works put emphasis on addressing changes to make software systems more maintainable. Following the concepts of software evolution & maintenance, Brodie & Stonebraker 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*" [15]. Brodie [13] mentions various characteristics of legacy systems such as mission critical, hard to maintain, inflexible and brittle. Bisbal et al. [11] listed problems of legacy systems such as legacy systems run on obsolete hardware, are expensive to maintain, suffer from lack of documentation and understanding of system, and are difficult to extend and integrate with other systems. Alderson and Shah [3] describe the issue regarding the lack of legacy experts/resources. From the aforementioned definitions and characteristics of legacy systems, it is evident that legacy systems are perceived as a serious problem in academia. The current study, in contrast, takes a different approach in identifying the characteristics that still keep them operational in the industry. Nevertheless, the study also explores various issues of the legacy systems that are in-line with the issues identified in the academia. In fact, we explicitly identified overlaps between the issues of legacy systems as observed in academia and in industry such as high maintenance cost, lack of resources, achieve flexibility. In this research, we have presented these issues as drivers for legacy system modernization.

5.2 Legacy System Modernization and Challenges

A plethora of legacy system modernization approaches have been reported. Brodie & Stonebraker presented the DARWIN method [14] as an incremental approach for migrating legacy systems. The RENAISSANCE approach [51] delivered a systematic method for legacy system modernization. Wu et al. [54] described the Butterfly method that uses a gateway-free approach unlike Brodie & Stonebraker's gateway approach [15]. Weiderman et al. [53] presented a system modernization approach that leverage middleware and wrapping technology. The use of reverse engineering techniques in legacy system modernization has been reported by Quilici [36] and Weide et al. [52]. Bisbal et al. analyzed the existing legacy system modernization approaches in their survey [12]. Seacord et al., in their book [40], presented a risk-managed approach to legacy system modernization. Various methods/techniques have been used to modernize legacy systems such as architectural pattern languages [22, 24], feature modeling [31, 30], iterative engineering [10], wrappers [46, 53], and aspects [32]. In the last decade, the advancement of web-based technologies has fostered legacy system modernization (e.g. [44, 28, 16, 19]). In particular, the service-oriented architecture (SOA) has been a popular target architecture for legacy system modernization and various legacy to SOA modernization approaches have been reported (e.g., Khadka et al. [25], Razavian & Lago [37], Almonaies et al. [4]).

With respect to legacy system modernization challenges, Brodie [13] listed various technical challenges of legacy system modernization that are more influenced by academic research experiences. Sneed [42] discussed recurring risks such as performance loss, architectural mismatch, testing bottleneck, current staff's rejection that are associated with a reengineering projects. Van Deursen et al. [48] presented an overview of techniques to facilitate legacy system modernization and the issues of modernization, particularly aiming at identifying objects. In a research roadmap, Bennett & Rajlich [9] argued to generalize the legacy system modernization approaches beyond source code, and to include data and objects of the current legacy systems.

[3] www.servicifi.org

The legacy system modernization research and the challenges identified in the academia are largely technology-oriented. They provide different techniques/methods to facilitate legacy system modernization and point out various challenges faced in the course of applying those techniques/methods. Our research not only identified various business and organizational issues, but also confirmed the technical observations made by researchers.

5.3 Empirical System Legacy Modernization Research

Sneed [41] reported a case study of modernization of a commercial application system for distributing books and other publications. Colosimo et al. [17] performed two controlled experiments to evaluate migration of COBOL-based system to a J2EE, web-based system. Murer et al. [33] presented their experience with an evolutionary approach to migrate legacy software of Credit Suisse Bank. Their findings largely complement our results as they focus not only in technical details of the modernization, but also on the business and organizational aspects. Apart from technical issues, the authors in particular identify the importance of knowledge existing within the technical staff, role of cultural values of the organization in the modernization process. Nasr et al. [35] present their experience with realizing two SOA migration case studies in which they identify various business issues such as resistance to change from the organization, time constraints to finish modernization along with technical issues such as lowering the maintenance cost, and increased flexibility. From an industrial perspective, Torchiano et al. [47] surveyed 59 Italian companies to identify the state-of-the-practice in software modernization and explained that the main factor influencing the modernization process is human factor followed by technological factors and issues related with data inconsistency. Razavian & Lago [38] conducted interviews to understand how legacy to SOA migrations are conducted in industry aiming at identifying which modernization activites/processes are prevalent in industry. In a report published by NASCIO [34], a survey of 29 states in the US is reported with the aim to identify the state-of-art of legacy systems, and the challenges faced during their modernization.

The findings of these studies resonate with the results of our study. The findings of this study undoubtedly support the results of these studies, yet we have employed different research methods (qualitative and quantitative methods) to further increase the reliability of the findings. In most of the aforementioned studies such as [37, 34, 47], a single research method is used. With respect to the observational studies [35, 17, 41], we refer to future work to validate our findings.

6. CONCLUDING REMARKS

Although legacy systems and their modernization have been extensively researched, this paper addresses 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 systems that initiate modernization, and the challenges faced during modernization from an industrial perspective. With the current status of our research some of our findings provide empirical evidence for existing hypothesis from the literature while others provide new insights that extend the body of knowledge on software modernization.

What this study adds to the discourse is an evidence that legacy systems are not necessarily a quagmire, but are business critical, reliable and proven systems that effectively execute the day-to-day business of organizations. Such perceived benefits of legacy systems are the factors that keep them alive in industry. Not to mention that the practitioners are largely motivated by the “If it ain’t broken, don’t fix it” aphorism towards legacy system modernization. In addition, the study identifies drivers of and the challenges for legacy system modernization. The drivers of the legacy system modernization in industry confirm the observation made such as lack of knowledge, high maintenance costs, and achieving flexibility. However, the challenges of legacy system modernization faced by the practitioners are not just technical, but also organizational and highly motivated by business considerations.

In summary, this study documents 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 differences in perception of legacy systems between the industry and academia.

The findings reported in this study is the outcome of an empirical research. We, therefore, want to convey that the findings and the conclusions drawn are suggestive, rather than conclusive. We do not claim to have comprehensively addressed all the perceived benefits and the drivers of legacy systems, and the challenges faced during legacy system modernization in the industry. However, the detailed grounded theory approach that we have adopted for analyzing the interview data, followed by the validation of the findings via an online survey, increase our confidence to claim that the findings and conclusions drawn represent a significant view.

The findings have the following implications: with these findings in hand, academics can (re)focus their efforts in legacy system modernization to include, besides the technological scope, the organizational issues, and business drivers for modernization, and in order to benefit from academic research results, practitioners should examine the academic tools, techniques, and methods developed for legacy system modernization, and initiate collaborations.

As to future work, we aim at validating the findings of the current study using observational participating research in a number of real world modernization scenarios. In addition, we believe that this study has highlighted “practitioners’ issues”, which the research community can address in order to facilitate technology and knowledge transfer.

7. ACKNOWLEDGMENTS

This research is sponsored by the Dutch Joint Academic and Commercial Quality Research and Development (JACQUARD) program on Software Engineering Research via contract 638.001.213.

8. REFERENCES

[1] S. Adolph, W. Hall, and P. Kruchten. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4):487–513, 2011.

[2] W. S. Adolph. Cash cow in the tar pit: Reengineering a legacy system. *IEEE Software*, 13(3):41–47, 1996.

[3] A. Alderson and H. Shah. Viewpoints on legacy systems. *Communications of the ACM*, 42(3):115–116, 1999.

[4] A. A. Almonaies, J. R. Cordy, and T. R. Dean. Legacy system evolution towards service-oriented architecture. In *International Workshop on SOA Migration and Evolution*, pages 53–62, 2010.

[5] E. C. Arranga and F. P. Coyle. Cobol: Perception and reality. *IEEE Computer*, 30(3):126–128, 1997.

[6] B. V. Batlajery, R. Khadka, A. M. Saeidi, S. Jansen, and J. Hage. Industrial perception of legacy software system and their modernization. TR UU-CS-2014-004, Utrecht University, 2014.

[7] L. A. Belady and M. M. Lehman. A model of large program development. *IBM Systems Journal*, 15(3):225–252, 1976.

[8] K. Bennett. Legacy systems: Coping with success. *IEEE Software*, 12(1):19–23, 1995.

[9] K. H. Bennett and V. T. Rajlich. Software maintenance and evolution: A roadmap. In *22nd International Conference on Software Engineering*, pages 73–87. ACM, 2000.

[10] A. Bianchi, D. Caivano, V. Marengo, and G. Visaggio. Iterative reengineering of legacy systems. *Transactions on Software Engineering*, 29(3):225–241, 2003.

[11] J. Bisbal, D. Lawless, B. Wu, and J. Grimson. Legacy information systems: Issues and directions. *IEEE Software*, 16(5):103–111, 1999.

[12] J. Bisbal, D. Lawless, B. Wu, J. Grimson, R. Wade, V.and Richardson, and D. OSullivan. A survey of research into legacy system migration. TR TCD-CS-1997-01, Trinity College Dublin, 1997.

[13] M. L. Brodie. The promise of distributed computing and the challenges of legacy systems. In *Advanced Database Systems*, pages 1–28. Springer, 1992.

[14] M. L. Brodie and M. Stonebraker. DARWIN: On the incremental migration of legacy information systems. TR TR-022-10-92-165, GTE Labs Inc, 1993.

[15] M. L. Brodie and M. Stonebraker. *Migrating legacy systems: Gateways, interfaces & the incremental approach*. Morgan Kaufmann Publishers Inc., 1995.

[16] G. Canfora, A. Cimitile, A. De Lucia, and G. A. Di Lucca. Decomposing legacy programs: A first step towards migrating to client–server platforms. *Journal of Systems and Software*, 54(2):99–110, 2000.

[17] M. Colosimo, A. D. Lucia, G. Scanniello, and G. Tortora. Evaluating legacy system migration technologies through empirical studies. *Information and Software Technology*, 51(2):433–447, 2009.

[18] B. Cornelissen, A. Zaidman, A. van Deursen, L. Moonen, and R. Koschke. A systematic survey of program comprehension through dynamic analysis. *Transactions on Software Engineering*, 35(5):684–702, 2009.

[19] A. De Lucia, R. Francese, G. Scanniello, and G. Tortora. Developing legacy system migration methods and tools for technology transfer. *Software: Practice and Experience*, 38(13):1333–1364, 2008.

[20] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2008.

[21] B. G. Glaser and A. L. Strauss. *The discovery of grounded theory: Strategies for qualitative research*. Aldine Transaction, Chicago, Illionios, 1967.

[22] M. Goedicke and U. Zdun. Piecemeal legacy migrating with an architectural pattern language: A case study. *Journal of Software Maintenance and Evolution: Research and Practice*, 14(1):1–30, 2002.

[23] N. Golafshani. Understanding reliability and validity in qualitative research. *The Qualitative Report*, 8(4):597–607, 2003.

[24] W. Hasselbring, R. Reussner, H. Jaekel, J. Schlegelmilch, T. Teschke, and S. Krieghoff. The dublo architecture pattern for smooth migration of business information systems: An experience report. In *26th International Conference on Software Engineering*, pages 117–126. IEEE, 2004.

[25] R. Khadka, A. Saeidi, A. Idu, J. Hage, and S. Jansen. Legacy to SOA evolution: A systematic literature review. In *In AD Ionita, M. Litoiu, & G. Lewis (Eds.) Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*, pages 40–70. IGI Global, 2013.

[26] R. Khadka, A. Saeidi, S. Jansen, J. Hage, and G. Haas. Migrating a large scale legacy application to SOA: Challenges and lessons learned. In *20th Working Conference on Reverse Engineering*, pages 425–432. IEEE, 2013.

[27] B. Kitchenham and S. L. Pfleeger. Principles of survey research: Part 5: Populations and samples. *Software Engineering Notes*, 27(5):17–20, 2002.

[28] J. Lavery, C. Boldyreff, B. Ling, and C. Allison. Modelling the evolution of legacy systems to web-based systems. *Journal of Software Maintenance and Evolution: Research and Practice*, 16(1-2):5–30, 2004.

[29] B. P. Lientz, E. B. Swanson, and G. E. Tompkins. Characteristics of application software maintenance. *Communications of the ACM*, 21(6):466–471, 1978.

[30] J. Liu, D. Batory, and C. Lengauer. Feature-oriented refactoring of legacy applications. In *28th International Conference on Software Engineering*, pages 112–121. ACM, 2006.

[31] A. Mehta and G. T. Heineman. Evolving legacy system features into fine-grained components. In *24th International Conference on Software Engineering*, pages 417–427. ACM, 2002.

[32] M. Mortensen, S. Ghosh, and J. M. Bieman. Aspect-oriented refactoring of legacy applications: An evaluation. *Transactions on Software Engineering*, 38(1):118–140, 2012.

[33] S. Murer, B. Bonati, and F. J. Furrer. *Managed evolution: A strategy for very large information systems*. Springer, 2011.

[34] NASCIO. Digital states at risk modernizing legacy

systems. Technical report, NASCIO, 2008.

[35] K. A. Nasr, H.-G. Gross, and A. van Deursen. Realizing service migration in industry–lessons learned. *Journal of Software: Evolution and Process*, 25(6):639–661, 2013.

[36] A. Quilici. Reverse engineering of legacy systems: A path toward success. In *17th International Conference on Software Engineering*, pages 333–336. ACM, 1995.

[37] M. Razavian and P. Lago. A frame of reference for SOA migration. In *Towards a Service-Based Internet*, pages 150–162. Springer, 2010.

[38] M. Razavian and P. Lago. A survey of SOA migration in industry. In G. Kappel, Z. Maamar, and H. R. Motahari-Nezhad, editors, *Service-Oriented Computing*, volume 7084 of *LNCS*, pages 618–626. Springer, 2011.

[39] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[40] R. C. Seacord, D. Plakosh, and L. A. Grace. *Modernizing legacy systems: Software technologies, engineering processes, and business practices*. Addison-Wesley Professional, 2003.

[41] H. M. Sneed. Software renewal: A case study. *IEEE Software*, 1(3):56–63, 1984.

[42] H. M. Sneed. Risks involved in reengineering projects. In *6th Working Conference on Reverse Engineering*, pages 204–211. IEEE, 1999.

[43] SQuaRE. Systems and software quality requirements and evaluation (SQuaRE)– system and software quality models. Standard ISO/IEC 25010, ISO, 2011.

[44] E. Stroulia, M. El-Ramly, and P. Sorenson. From legacy to web through interaction modeling. In *18th International Conference on Software Maintenance*, pages 320–329. IEEE, 2002.

[45] T. Sucharov and P. Rice. The burden of legacy. Online: http://www.ncc.co.uk/article/?articleid=15665, 2008.

[46] P. Thiran, J.-L. Hainaut, G.-J. Houben, and D. Benslimane. Wrapper-based evolution of legacy information systems. *Transactions on Software Engineering and Methodology*, 15(4):329–359, 2006.

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

[48] A. van Deursen, P. Klint, and C. Verhoef. Research issues in the renovation of legacy systems. In *2nd International Conference on Fundamental Approaches to Software Engineering*, pages 1–21. Springer, 1999.

[49] M. van Sinderen. Challenges and solutions in enterprise computing. *Enterprise Information Systems*, 2(4):341–346, 2008.

[50] N. Veerman. Revitalizing modifiability of legacy assets. *Journal of Software Maintenance and Evolution: Research and Practice*, 16(4-5):219–254, 2004.

[51] I. Warren and D. Avallone. *The renaissance of legacy systems*. Springer, 1999.

[52] B. W. Weide, W. D. Heym, and J. E. Hollingsworth. Reverse engineering of legacy code exposed. In *17th International Conference on Software Engineering*, pages 327–331. ACM, 1995.

[53] N. H. Weiderman, J. K. Bergey, D. B. Smith, and S. R. Tilley. Approaches to legacy system evolution. TR CMU/SEI-97-TR-O14, DTIC Document, 1997.

[54] B. Wu, D. Lawless, J. Bisbal, R. Richardson, J. Grimson, V. Wade, and D. OSullivan. The butterfly methodology: A gateway-free approach for migrating legacy information systems. In *3rd IEEE International Conference on Engineering of Complex Computer Systems*, pages 200–205. IEEE, 1997.