DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4556

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

IMPLICATIONS OF LEGACY SOFTWARE SYSTEM MODERNIZATION – A SURVEY IN A CHANGED SCENARIO

H. SeetharamaTantry
Department of Mathematics & Computational Sciences
National Institute of Technology Karnataka,
Surathkal, India

Murulidhar N.N.
Department of Mathematics & Computational Sciences
National Institute of Technology Karnataka,
Surathkal, India

K. Chandrasekaran
Department of Computer Science and Engineering
National Institute of Technology Karnataka,
Surathkal, India

Abstract: The problem of a legacy software system is never ending. Today's working system will be tomorrow's legacy. Whatever care a software developer takes to build a state-of-the-art system as per latest available technology today, due to constant improvement/change in technology, the today's working system may become tomorrows' legacy. Legacy System is a computer program, typically a database system, which, although critical to an organization's operations, is in an obsolete format or is installed on an obsolete system. Replacing legacy system or making it work with new requirements is the most time consuming and also economically unviable to the industry. The outdated software system that may still be in use in an organization is because its data cannot be changed to newer or standard format, or its application programs cannot be upgraded. There are many legacy software applications that have either no source codes available or require too much effort to port. One of the severe problems in the evolution of legacy software system is the loss of knowledge about them. Herein a survey of the problems of Legacy Software System is made and arrived at solutions to these problems using the available techniques.

***Keywords*:** Implication, Legacy, Modernization, Migration, Re-engineering, Rehosting, Wrapping.

I. INTRODUCTION

Today's working system will be tomorrow's legacy. It is well known that the hardware and software fields in computing are changing rapidly. Good software developed on a relevant hardware may not work after 3 to 4 years of its development due to change in technology. Hardware may not be useful after 3 to 4 years of its deployment due to non-availability of spares and also the arrival of new hardware architecture. Such hardware or software which are outdated are known as legacy systems. A legacy system is an old method, technology, computer system, or application program "of, relating to, or being a previous or outdated computer system" [2].

Many legacy systems are performing crucial work for their organization: may be billing system in a mall, may be ticket reservation system for a transport system like rail, flight, bus, etc. Changing the legacy system may (i) hamper the work of the organization and other stake holders (ii) affect in losing years of accumulated experience and knowledge (iii) affect in losing of business rules if it is only on the legacy software system which is running successfully till date. It is not possible to discard the legacy software application in such places as these software applications are playing an important role in the business of the organization. These applications may have lots of users, bundles of data, properly set business rules, etc. By just discarding these legacy applications, the organization will be nowhere in no-time.

The fast changing business in today's environment requests state-of-the-art techniques in the area of software, to cope-up with the competition. Competitor organization using a state-of-the-art technology is another reason to modernize the legacy software. Change in hardware technology compared to olden days, change in Operating Systems, change in application software, etc. are also demanding a new software for the business to run smoothly. Increasing the efficiency and scalability of the application system is very important in today's business environment.

So, to survive in the market, each legacy application should be properly taken care to utilize till it is ported to anew platform (hardware or software) and tested vigorously for any difficulty, or till a separate step is taken to replace, in any form, to substitute the legacy application.

Legacy Information System (LIS) consists of 6 components viz., (i) System hardware (ii) Support Software (like OS, etc.) (iii) Application software (iv) Application data (v) Business process (vi) Business rules [4]. Since the System hardware and OS are beyond the scope of this paperand considered only rest of the components.

In the following section, a relook of the Legacy system is presented. Section 3 discusses the necessity of the legacy system modernization. Section 4 shows various ways to modernize the legacy system. This section discusses various approaches and its merits and demerits through a tabular form. Section 5 enumerates the implications of legacy system modernization and challenges to be faced during

© 2015-19, IJARCS All Rights Reserved

legacy modernization. Finally, section 7 concludes this legacy system modernization survey.

II. LEGACY SYSTEM: A RELOOK

Legacy System is "any system that cannot be modified to adapt to constantly changing business requirements and is still valuable to its stakeholder such that its failure can have a serious impact on business"[11].Legacy systems are the backbone of organizations and often considered as a high economic value asset [10]. These systems are mission critical and comprise a lot of business logic which represents many years of knowledge base and data. Organizations cannot simply discard their legacy systems, even though it is heavy to maintain them due to present day's requirements.

A. Behind legacy

A legacy software was written many years ago using techniques which were latest then, which is outdated today, yet it continues to do useful work [5]. The legacy system may be defined informally as "large software system that one who doesn't know how to cope with but that are vital to the organization". Most of them are very old that one will reach the day soon when some 50-year-old software still running successfully. Most of these written in early versions of 3rd generation languages like COBOL, FORTRAN, etc., or written even still older languages like assembly language. Legacy systems are usually made of two kinds of artifacts: source code and databases[18].

Due to the technology available at that time, the clarity and structure were traded off for the sake of speed. Concurrency was achieved using some ad-hocmethods because it was not understood clearly at that time. Frequent software maintenance rapidly degrades the software structure. So, again maintenance becomes a very difficult task. A large number of source code, changed or inserted during maintenance phase hamper the program understanding. So, legacy systems are very difficult to understand and program understanding itself becomes major activity during maintenance.

Legacy problems are made still worse with the management problem like (i) Many software engineers don't like to touch legacy system for maintenance. Many of today's engineers don't know the outdated languages like COBOL, FORTRAN, FoxPro, Visual Basic, Assembly language. Due to its large manual coding, the efficiency is very less. In today's world coding done through some sophisticated utility like SQL, Screens, etc. (ii) User community may not be interested in changing the well running, reliable legacy systems. The replaced new system may be less reliable and requires end-user to learn again to use the new system. This may affect the aged users who are adjusted to the routine legacy system.

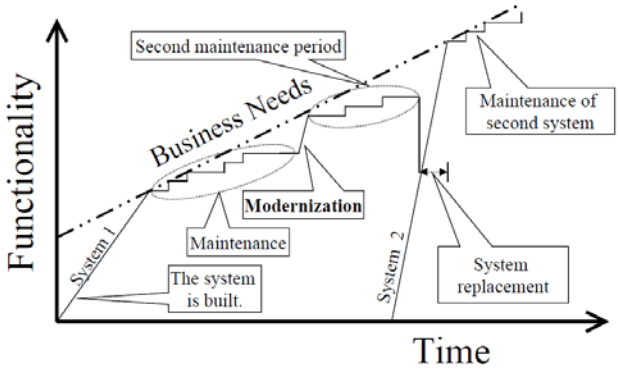

Fig.1: Information system life cycle [7]

Software evolution is an integral part of the development process. The fig. (1) explains how the functionality of the information system changes with respect to time. As the business needs increases, the software requires maintenance and modernization activity, frequently. Reverse engineering and re-engineering should be a continuing process of software development and evolution. Manny Lehman[6] has argued that software must continually change or will become less useful in the real world. In addition, Lehman's second law observes that the structure of evolving software will degrade unless remedial action is taken regularly. The majority of the legacy software will be running without such remedial action. In due course of time of development, the original structure disappears, due to periodic change in requirement. The documents exist, if any, is of no use and maintenance is carried out using source code only, because it is the only reliable source of information about the system. And, the software becomes very difficult to maintain in due course of time.

B. The Legacy system sustaining strategies

The sustaining strategies for the Legacy system include (1) Continue to maintain the Legacy system (2) Discard the system (3) Re-host the system (4) Replace the system (5) Modernize (re-engineering) the system.

1) Continue to maintain the legacy system: Maintenance is an integral part of any software application and is continuous till the existence of the application [9]. This involves small bug correction, adding small routine, but does not involve any major structural change. The major hurdle is getting experts in this outdated technology, which is also very expensive. So, themaintenance costof software application increases with time.

*2) Discard the Legacy system:*Discarding the application system is not a good option even though, but, sometimes there may not be any option other than abandoning the application system. The reason can be afinancial crunch for redeveloping, the legacy hardware fails to support and the proprietary software does not work on any other hardware platform, hardware or software vendor has stopped the productionor support,to name a few.

*3) Replace the Legacy system:*When it is difficult to maintain a legacy software due to many reasons, replacing the legacy system with a new system can be thought of. There are two options. The first option is by replacing with a new, custom made software. The other option is to replace with a commercially available ready to install software.

© 2015-19, IJARCS All Rights Reserved

4) Modernization of the Legacy system: There are many options available for themodernization of the legacy system. A conversion is an option where the legacy codes are converted to new and latest platforms like COBOL code to Java, etc. Re-hosting the legacy system on avariety of other platforms may solve the legacy problem in some case. Remodeling may solve legacy system problem at times. Porting to thecloud is another option in themodern era. Wrapping is an option where legacy system modules are encapsulated as a component in a new implementation. These components may be replaced slowly in future by new software components. Code transformation, functional transformation, re-engineering and reverse engineering are some of the other options available during legacy modernization activity.

III. NEED FOR MODERNIZING LEGACY SOFTWARE SYSTEM

The Legacy system is a software written yesterday. The software system becomes legacy system when it begins to resist change, alterations, and modifications[12]. Working Software requires continuous modifications which fall under 4 categories: i) Perfective, changes made to improve the developed software product. ii) Corrective, repair the defects, shortfalls observed in a working software. iii) Preventive, which improves the software reliability and supports future maintenance. iv) Adaptive, if the working platform gets upgraded to anew platform like hardware, compilers, operating system, DBMS system, etc., then it requires a facelift of the working software, to avoid the software becoming the legacy system.

A. Reasons to go for modernization of legacy systems

A change or upgrade of the working hardware platform due to its obsolescence or frequent failure of the hardware and non-availability of the spares, etc. is one reason for software modernization. The other reason would be like operating system (OS) of the hardware, which gets no support from vendor/developer and is compelled to upgrade it to a new one. Even though the hardware is same and OS is upgraded or vice-versa, it affects the working software system and makes it a legacy system.

As the day passes, the maintenance of the software becomes difficult. No further maintenance is possible due to the architecture design which was carried out for those day's needs which cannot be used for today's requirements. Demands of the marketing departments for alternate, user-friendly, using latest technology pressurizes to go for modernization of existing legacy system. A failure of software in catering the latest demands can lead to far reaching consequences like loss of money, reputation, and trust[13].

Sometimes, commercially available software on the market is cheaper and very flexible in getting advanced options when compared to maintaining a legacy system is another reason for legacy modernization. Moreover, it is available off-the-shelf. It is very easy to get better efficiency and scalability through commercially-off-the-shelf (COTS) software, compared to the legacy system.

B. The Challenge

Modernizing is nothing but replacing the legacy system or making it work with new requirements, is one of the highly challenging, much time and money consuming aspects of the modern organizations. Large expenses are incurred for modernizing the software. Conversion of the huge volume of data is also a time and money consuming activity. Some modernization activity requires running legacy system and newly built system parallel till the desired goal is reached, is a challenge. This attracts more man power and resources. Again, the more monetary burden to the organization. The development team should be well versed with both legacy and new system programming languages.

Computer technology is changing rapidly in such a way that today's software becomes a legacy in just no time. So, the new system should be up in a small time gap, which is a challenging work for the developers. Sometimes, today's software components are developed by various vendors and most of them are distributed on various platforms. Maintaining coherent is a big task here. If not controlled these type of problems, the legacy problem will arise in different type and shape in a much worse form.

C. Merits and Demerits of modernization of legacy systems

There are many benefits of modernization of legacy systems. Latest technologies like web services can be effectively made use of to achieve a high level of usage of the software. Various modules in the modernized software can be made 'talk to each other', which otherwise was not possible with the legacy system. Security at a higher level can be maintained and business partners can be offered secure and automated access to each other's data, using services like web service interface, etc. Stake holders can access their service provider's facility through state-of-the-art web interfaces. The in-house user can access his organization from anywhere in the world.

Client server technology of the legacy system can be replaced by new web technology which cuts costs and improves productivity. Operational cost can be reduced by implementing the modernized system on new platforms like cloud, etc. In olden days dedicated server and clients were used for such type of activity. This can be avoided with new web technologies. Reduced evolution cost during change or re-implement of the applications when those applications rely onbehind web service interface, saves a lot to the business organizations. If the organization makes use of the cloud, it can pay only little for the service it avails. Modern technology like cloud can be used by these modernized systems which save a lot to the organization in the form of the license fee to buy a software, in the form of cost to buy a hardware, storage, etc.

Some of the difficulties faced if modernization of the legacy system carried out are cost and time involved in training the stakeholder on the modernized software system. It will take a long time to come out from old working mood to new mood. Sometimes soft code is the only document available on the legacy system and no separate written document is available on the system due to frequent updates since development. This may result in the fear of losing business rules. Due to this, the new system may not be fully compatible with the legacy. So a parallel system should run

© 2015-19, IJARCS All Rights Reserved

till all the components of the legacy system are modernized to a new one fully. So, a burden of extra hardware, software, man power, etc., which attracts huge money.

IV. WAYS TO MODERNIZING LEGACY SYSTEM

A failure of a software system to deliver the required feature, in an organization, will have far reaching consequences like loss of money, reputation and trust [13]. This may be due to the legacy software which reached its maintenance activity and cannot be maintained further due to its limits. Researchers had proposed various models to tackle the legacy problem.

The following fig.2 gives an overall picture of the various modernization strategies and these have been considered for this study purpose. The major legacy modernizing activity include Rehosting the legacy software on the new hardware platform, Replacing the legacy software with a new software or modernizing legacy software using various reengineering options like wrapping, revamping, etc.

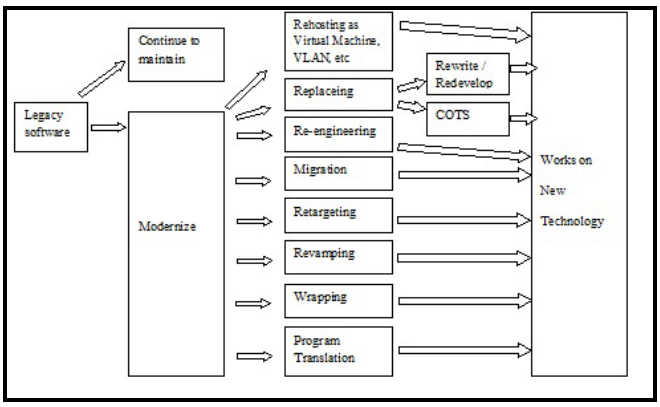

Fig.2: Showing various legacy modernization strategies

A. *Re-Hosting*

Re-hosing is, without many changes, running legacy applications on a different platform. This will reduce the cost of maintenance of legacy hardware. For example, running legacy applications on UNIX platforms which were originally running on Mainframe systems, with full features up. Most of the time this is only a "stop-gap" option, till a completely new application is developed. Some of the frequently used re-hosting techniques are VLANs[8], Hyper-V server, VM (Virtual Machine), MED-V, etc.

B. *Replacing*

Replacement is worth when the legacy system cannot fulfill business needs and when modernization is not worth in terms of cost. When black-box and white-box modernization approach is not possible, then only replacement may be the option available to go for. Replacing legacy system is through either by writing new software from scratch (tailor made) or by replacing it with Commercially-Off-The-Shelf (COTS) package, in whole or part with packages such as ERP, CRM.

*1) Replace with redeveloping:*Discard the legacy system and redevelop using latest state-of-the-art technology. Big bang approach replaces the legacy system in one stretch. Thorough testing for all functionalities will be done before implementing. Whereas, incremental approach replaces the legacy software with a small increment, step by step, or module by module.This requires running of old and new software side by side. Each increment is compared with alegacy system for any discrepancy, then tested and continued. After fully developing, the replaced system works as good as the legacy system. Some more features and options can be included to the newly developed software. This can be further maintained for many more years. Cost wise, redevelopment is expensive, in most of the time. The developer should be well versed with both legacy and new platforms[12]. But normally, people who maintain legacy system only are assigned to do aredevelopment of the legacy system. Most of the case, they may not be familiar with new technologies. This may hinder replacement activity. The legacy system is well tested and may contain lots of business expertise. The new system should be well tested before implementation, for its robustness with thesame functionality as that of the legacy system.

2) Replace with Commercial Off-The-Shelf-System (COTS):If the total business rule changes, or, there is no written document available for the legacy system, or the system is outdated and the modernization activity does not help to improve the situation, then one doesn't have any option other than going for replacement of the software. This replacement can be either from off-the-shelf-package or a custom made software system. The well-proven software works fine on its limits. This may not work as per user requirement as in thecase of redeveloped software. Cost is low compared to redeveloped software in many cases.

C. *Re-engineering*

Reengineering of legacy systems is defined as "examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form" [15]. The re-engineering process includes three phases: forward engineering, reverse engineering, and re-engineering [16]. It is rebuildingof the legacy system in a new technology or platform with equal or more functionality using various techniques. Reengineering is a type of modernization activity which introduces modern technologies to the legacy system, which will help in improving maintainability of the legacy system[12]. It is the analysis and adjustment of an application system in order to represent it in a new form using techniques like reverse engineering, restructuring, redesigning, retargeting, revamping, source code translation, code reduction, and functional transformation. All these re-engineering techniques help in improving the qualities of the legacy system[12]. The SEI developed a definition of reengineering: Reengineering is the systematic transformation of an existing system into a new form to realize quality improvements in operation, system capability, functionality, performance, or evolvability at a lower cost, schedule or risk to the customer [12].

Reengineering is more expensive than maintenance of the legacy system, but, it is a one-time activity. In a technical point of view, replacing the legacy system is more attractive than its re-engineeredcounterpart. The lifetime of legacy system can be extended by way of re-engineering

© 2015-19, IJARCS All Rights Reserved

techniques. Reengineering the legacy system is cheaper and has less risk when compared to replacing the legacy system.

D. Migration

Migrating legacy system to evolvable system requires the help of disciplined re-engineering techniques. To meet new requirements, various engineering principles are applied to the existing legacy system. Migration also represents a chance to modernize the business and consider newer technologies like cloud computing. This is a technique to move the legacy applications to more flexible environments like SOA environments without disturbing the original system's data and rules. The focus is to get a greater return on investment (ROI) than it could be obtained through redevelopment. Meng, Qu, & Guo divide migration into two types which are component-based migration and system-based migration[17]. Component-based migration classifies the legacy system into independent components and then migrates the components singly. System-based migration integrates the whole legacy system and its data into the new system.

E. Retargeting

Migration of legacy system to new hardware platform is retargeting. Expensive maintenance cost forces to migrate the legacy system to a newer and powerful hardware platform. Modern higher performance computer hardware provides an evolvable platform to such other modernization activities. This activity reduces the operational maintenance cost.

F. Revamping

Revamping is replacing the user-interface (UI), which is the most visible part of the software system. This improves the visibility and usability of the legacy system at the end-user level. Web enablement can also be provided with additional hardware and software. A common technique followed here is 'Screen Scrapping', which consists of wrapping old, text-based interfaces with new, graphical user interface. Even then the new system remains difficult to maintain due to its internal legacy nature. Most of the time the revamped software performs slower than the original legacy system[12]. Only visibility improves with this type of modernization activity. The addition of new modules, functions may not be easy even after revamping.

G. Wrapping

Wrapping can be defined as "surrounding the legacy system with a software layer that hides the unwanted complexity of the old system and exports a modern interface"[12]. Wrapping provides a new interface to existing components to make them easily accessible as services to other software components. Wrapping integrates the legacy system to heterogeneous distributed computing environment. Wrapping is cost effective. The deficiencies observed in legacy continues even after wrapping. It is only a stop gap solution[14].

H. Program translation

Program translation is an operation that takes a computer program and generates another program. The transformation may be several types. One among them is source code translation[3]. When the legacy system is moved to new hardware platform due to various reasons, then the current programming language may not be available on the new platform. So, at this time conversion of old source code to a new and more modern language is only the solution. This is called source code translation. Sometimes source code translation may also happen with the same hardware platform, without moving to a new one. It is very difficult to judge whether the whole system converted to a new system. Automatic conversion tools reduce the cost of translation. However, the structure of the code cannot be significantly changed from old to the new language. That remains same at most of the time. For example, a legacy system written in COBOL, when translated to java, looks like COBOL written in Java. Translated code is not easy to maintain.

V. A TABULAR VIEW OF VARIOUS MODERNIZATION STRATEGIES

The following Table.1 explain the advantages and disadvantages of the various Modernization strategies along with success rate of modernization of legacy software system. Some strategies which cost less for modernization activity gives only temporary extended life for the modernized software. Again after some time, it requires modernization activity or re-writing activity, means it is only a stop-gap arrangement. But it doesn't mean that the activity which costs high will survive for a longer time. Everything depends on the type of legacy system and the area where the legacy system is used.

Table.1: A tabular view of comparison of various modernization activities studied

Modernization Strategy	***Cost involved***	***Time required to implement***	***Success rate***	***Advantages***	***Disadvantages***
Re-hosting : As virtual machines[8]	Nil, Any working Desktop is enough	Immediate	High	* No change in the working environment. * Reduce the maintenance cost of Legacy software	*Stop-gap option only. If the software itself is legacy then this option is invalid
Re-hosting : on new hardware[8]	Cost of new and compatible hardware	Immediate	High	* Reduction in the cost of maintenance. * working environment doesn't change * Increase in performance due to new hardware	* Stop-gap option only. If the software itself is legacy then this option is invalid * Software remains still Legacy

© 2015-19, IJARCS All Rights Reserved

Replacement: with re-developing[12]	High	Long time	High	* Uses new technology * New modules can be added	* User training is required * Good testing is a must * Running old & new systems parallel attract more resources
Replacement: with Commercial-of-the-shelf (COTS) products	High	Immediate	High	* Adds new technology to the organizations' software life. * No time wasted.	* Business rule changes. * Customer have to tune the working environment to the pre-developed software
Migration[11]	Medium	Moderate	High	* Same working environment. * Cloud computing technology can be utilized	* Database remains same as that of Legacy.
Re-engineering[12]	Medium	Long time	Medium	* Improvement in the working environment. * Introduces modern Technology	* If the system study is not proper, then failure. * The developer should be well versed with both old & new languages & platforms.
Wrapping[14]	Low	Moderate	High	* Legacy components can be used on web technologies. * Heterogeneous distributed computing environment can be utilized	* Efficiency is not improved in terms of legacy applications. * Only stop-gap option.
Source code translation [3]	Low	Moderate	Medium	Same code can be transformed to new platform with little manual involvement	* System to be tested again for the correctness on the new platform. * Structure of Legacy code remains same in new language also.
Retargeting	Medium	Immediate	High	* Reduces recursive operational maintenance cost. * Legacy hardware maintenance cost reduced	Cost of new hardware may be a burden
Revamping	Low	Moderate	Low	Improves the visibility and usability of system at Front-End	* Internally same Legacy software, so maintenance is difficult. * All the problems of the Legacy system continues.

VI. CHALLENGES IN LEGACY MODERNIZATION

Since the legacy system is critical in operations of many enterprises, a modernization strategy is to be devised so that the system is fully functional during modernization effort. Deploying a modernized system at one stretch is not recommended because of its critical nature. This may attract additional problem. A legacy modernization project should start from a simple component unit first. Then, every time it is to be updated till the goal is reached successfully. So, an incremental method of modernization is recommended. Every modernization replaces part of the legacy system, decreases the legacy percentage. This way legacy system is incrementally modernized. Most of the time the legacy system and modernized system should run parallel. This attracts data duplication, overlapping of the functions, etc.

VII. THE IMPLICATION

After modernization activity, the legacy software can be used for some more time-period. The value of the legacy investment is extended by modernizing it. The various technologies have made the legacy modernization process a cost effective option. The modernized legacy system will support various business requirements and withstand market competitions. The end-user community will be satisfied with the modernized software. The modernized legacy system will be easy to use.

The features of cloud computing shall be utilized effectively while modernizing to cope-up with the present day requirements. The modernized software will engage the user community with its user-friendly features.

An organization adopting a various modernization strategy on the legacy system can reduce costs in terms of maintenance, and also avoid the overhead and management of IT equipment. The modernization activity extends the lifespan of the legacy information systems and the fact that it therefore also helps to improve the return on investment (ROI) [19].

The Legacy modernization helps in reducing cost on going for totally new software. This reduces the financial impact on the business of an organization. The organization also gets a little time to adjust the budget for investing on a totally new software system to fulfill the present day requirements. The modernization activity on the legacy system also saves a lot in giving training to the end user community, if otherwise a must for newly developed software.

© 2015-19, IJARCS All Rights Reserved

Sometimes business rules/procedures exist only in legacy software system, in the form of coding. Due to frequent modification of software over a period of time, these changes might not have been updated in business rule documents. During modernization, one can extract latest updated business rules and processes, and record it for future use.

Ultimately deciding which solution to select will be based on economics. Whether to continue with the legacy system by accepting the cost of maintenance or invest on the new system so that it is easier to maintain, is to be decided. Ranson et al [9] describe an assessment technique to determine to replace, modernize or to continue to maintain the legacy software system.

VIII. CONCLUSION

The computer technology is running in a race. To cope up with the rapidly changing technology, business firms need to use new technology, grow and produce new items. This is necessary to compete in today's business world. The goal is not to change how the business works, but to change the technology used. Legacy systems are the backbone of organizations and often regarded as an organizational asset with a high economic value. Legacy modernization aims to retain and extend the value of the legacy investment. The new technologies like program transformation, wrapping, etc., have made the legacy modernization process a cost effective and accurate way to preserve legacy investments. This helps in reducing cost and business impact of legacy software moving to entirely new software platform. The goal here is to retain the value of the legacy asset on the new platform. Legacy modernization is not a one-size-fits-all strategy. It is quite possible to employ one strategy or a combination of strategies.

In this survey paper, an elaborated study has been carried out on the majority of the works done by various researchers in the area of Legacy System modernization. It has been observed that one best option to solve the legacy problem is migration. Migration involves either moving to a new hardware, new software or both. Migration makes sure that new features are exploited, old settings are intact and current applications continue to work in the new environment. The research is continuing in this direction.

IX. REFERENCES

[1] http://www8.hp.com/h30458/apr/en/sub/Don't-wait-Microsoft-Windows-Server-2003-Support-ends-soon_1433437.html

[2] http://en.wikipedia.org/wiki/Legacy system

[3] http://en.wikipedia.org/wiki/software_modernization

[4] Humairath K M Abu Bakar & Rozilawati Razali, "A preliminary review of legacy information systems evaluation models", 3rd International conference on research and innovation in information systems – 2013 (ICRIIS'13), pp.314-318.

[5] Keith Bennett, "Legacy systems: coping with success", IEEE Software, January 1995, pp.19-22.

[6] M.M.Lehman, "Progress, Life-cycles and the law of program evolution", Proc.IEEE, 1980, pp.1060-1076

[7] Santiago Comella Dorda, Kurt Wallnau, Robert C.Seacord, John Robert, "A survey of legacy system modernization approaches", Technical note, CMU/SEI-2000-TN-003, April 2000.

[8] http://betanews.com/2013/10/14/legacy-apps-holding-you-hostage-7-ways-to-safely-migrate-off-windows-xp/

[9] Ranson, J.; Sommerville, I.; Warren, I. "A method for assessing legacy systems for evolution", Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering (CSMR98, 1998.

[10] Khadka, Ravi. "Revisiting legacy software system modernization". Diss. Utrecht University, 2016.

[11] M. L. Brodie and M. Stonebraker. "Migrating legacy systems: Gateways, Interfaces& the incremental approach". Morgan Kaufmann Publishers Inc., 1995.

[12] Robert C. Seacord, Daniel Plakosh, Grace A Lewis "Modernizing legacy systems: software technologies, engineering processes and business practices", Addison Wesley, Longman Publ. Co., Inc. Boston, MA, USA @ 2003.

[13] A Fumweger, M Auer, S Biffl, ICEIS(1), "Software evolution of legacy system – A case study of soft migration", 2016.

[14] Chia-Chu Chiang and CoskunBayrak, "Legacy software modernization", IEEE international conference on systems, Man, and Cybernetics, October 8-11, 2006, Taipei, Taiwan.

[15] Chikofsky, E., J. Cross II, "Reverse engineering and design recovery: A taxonomy", Software Reengineering, IEEE Computer Society Press, 1992, p.54–58.

[16] Demeyer, S., S. Ducasse, O. Nierstrasz, "Object-oriented re-engineering patterns", Square Bracket Associates, Switzerland, 2009.

[17] Meng, F., Qu, Z., &Guo, X. "Refactoring model of legacy software in smart grid based on cloned codes detection", IJCSI International Journal of Computer Science Issues, 10(1), 296-303, 2013.

[18] Castillo, Guzmán, Piattini, García, "On the use of ADM to contextualize data on legacysource code for software modernization", 16th Working Conference on Reverse Engineering, IEEE, 2009.

[19] Castillo, R.P., MARBLE, "Modernization approach for recovering business processes from legacy information systems", 28th IEEE International Conference on Software Maintenance (ICSM), 2012.

© 2015-19, IJARCS All Rights Reserved