**Individual coursework.**

**Submission Date: 14th/7/2022**

1. a) With examples discuss the Lehman’s laws of software evolution
2. Discuss the importance of reverse and forward engineering as a tool for re-engineering
3. a) Explain the various ways of evaluating legancy systems
4. What are the importance of legancy systems in software evolution?

Engineering is a very versatile industry that is always getting creative. Using creativity and innovation, engineers create never-before-seen products that benefit their communities. They play a key role in expanding local economy and sparking business exchanges. However, one question remains: how do engineers innovate in such a fast-paced environment?

The answer is the reverse, literally. Reverse engineering plays a huge role in sparking innovative and productive minds that produce necessities across every industry. The reverse engineering process includes taking apart worn down products to examine how individual parts work, and then incorporating past inventions into new ones. There are plenty of benefits to the reverse engineering process that will only expand our knowledge and capabilities.

1. Exploring existing designs and maneuvers  
   Reverse engineering allows us to see what already exists. This includes any parts, structures, or processes that could benefit communities in other ways. Examining current products leads to innovation and discovery, all thanks to reverse engineering.
2. Reconstructing a product that is outdated  
   A key part of redesigning an existing product is understanding the product itself. Reverse engineering provides the visual to work out outdated kinks in an older system. Quality is the most important aspect of this process.
3. Discovering any product vulnerabilities  
   Similar to the previous step, reverse engineering supports finding faults in the product. This is to ensure the safety and well-being of the product’s users. It is best for an issue to arise during the research phase rather than the distribution phase.
4. Bringing less expensive & more efficient products to the market  
   Reverse engineering’s main goal is to lead engineers on a path towards innovation and success. Succeeding includes lowering manufacturing costs and raising product effectiveness as much as possible.
5. Creating a reliable CAD model for future reference  
   Most reverse engineering processes include a full-working CAD file for future references. A CAD file is created so the part can be examined digitally if future issues arise. This form of technology has enhanced engineering productivity and product expression.
6. Inspiring creative minds with old ideas  
   Lastly, reverse engineering gives way for innovative design. During the process, an engineer might discover a system that could be useful for a completely different project. This shows how engineering connects projects with previous knowledge.

**Difference between Forward Engineering and Reverse Engineering**

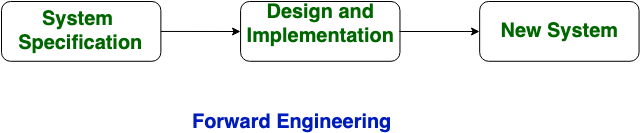
* Last Updated : 14 Jun, 2022

**Forward Engineering:**

Forward Engineering is a method of creating or making an application with the help of the given requirements. Forward engineering is also known as Renovation and Reclamation. Forward engineering  requires high proficiency skills. It takes more time to construct or develop an application. Forward engineering is a technique of creating high-level models or designs to make in complexities and low-level information. Therefore this kind of engineering has completely different principles in numerous package and information processes. Forward Engineering applies of all the software engineering process which contains SDLC to recreate associate existing application. It is near to full fill new needs of the users into re-engineering.

**Characteristics of forward engineering:**

1. Forward engineering is a variety of engineering that has different principles in numerous package and information processes.
2. Forward engineering is vital in IT as a result of it represents the ‘normal’ development process.
3. Forward engineering deals with the conversion of business processes, services, and functions into applications.
4. In this method business model is developed first. Then, a top-to-down approach is followed to urge the package from the model developed.
5. Forward engineering tools are accustomed move from implementation styles and logic to the event of supply code.
6. It essentially permits the user to develop a business model which may then be translated into data system components.
7. These tools basically follow the top-to down approach. System creator and visual Analyst is a forward engineering CASE tool.



**Reverse Engineering:**

Reverse Engineering is also known as backward engineering, is the process of forward engineering in reverse. In this, the information is collected from the given or existing application. It takes less time than forward engineering to develop an application. In reverse engineering, the application is broken to extract knowledge or its architecture.



**Difference between Forward Engineering and Reverse Engineering:**

|  |  |  |
| --- | --- | --- |
| S.NO | Forward Engineering | Reverse Engineering |
| 1. | In forward engineering, the application are developed with the given requirements. | In reverse engineering or backward engineering, the information are collected from the given application. |
| 2. | Forward Engineering is a high proficiency skill. | Reverse Engineering or backward engineering is a low proficiency skill. |
| 3. | Forward Engineering takes more time to develop an application. | While Reverse Engineering or backward engineering takes less time to develop an application. |
| 4. | The nature of forward engineering is Prescriptive. | The nature of reverse engineering or backward engineering is Adaptive. |
| 5. | In forward engineering, production is started with given requirements. | In reverse engineering, production is started by taking the products existing products. |
| 6. | The example of forward engineering is the construction of electronic kit, construction of DC MOTOR , etc. | An example of backward engineering is research on Instruments etc. |

# Forward Engineering

## **What Does Forward Engineering Mean?**

Forward engineering is the process of building from a high-level model or concept to build in complexities and lower-level details. This type of engineering has different principles in various software and database processes.

Generally, forward engineering is important in IT because it represents the 'normal’ development process. For example, building from a model into an implementation language. This will often result in loss of semantics, if models are more semantically detailed, or levels of abstraction.

Forward engineering is thus related to the term 'reverse engineering,’ where there is an effort to build backward, from a coded set to a model, or to unravel the process of how something was put together.

References

<https://plg.uwaterloo.ca/~migod/papers/2013/lehmanPaper.pdf>

<https://www.cedarvilleeng.com/news-and-insights/blog/6-benefits-of-reverse-engineering/>

<https://link.springer.com/chapter/10.1007/978-3-319-14130-5_4?noAccess=true>

<https://astrixinc.com/evaluating-legacy-lims-software/>

<https://www.sciencedirect.com/science/article/abs/pii/S0950584908000694>

Five steps to legacy system nirvana? It can be done. Check out our tips!

Conventional wisdom says that changing a legacy system can be complicated and costly. But what about the opportunity cost when you outgrow your systems and start losing profits? You need to know if your system is reaching its end of life—before it gets there. Here’s our guide…

## Step 1 - Complete a Post-it Note Analysis

**Identify the top three issues and the top three benefits of your system. Write them down on sticky notes.**And keep in mind the following factors:

* **Stability and Reliability**– Does the system stay live consistently or does your team experience downtime too frequently?
* **Maintainability** – Is the system easy to maintain? Can maintenance be done internally? Is the company that built the software still thriving? Is it offering support as your partner, or are they just another vendor? and willing to help?
* **Compatibility** - Are you able to integrate new software to continually improve your system? Is it easy or difficult to connect with other systems?
* **Outdated Function** – Does the software still do everything you need it to do? Is your team creating and using “hidden” manual workarounds?

## Step 2 - Estimate the Severity

Legacy system drag has real business costs—both direct and indirect. You need to put numbers on both. Follow this assessment:

* If the current system is often experiencing downtime, what opportunities are missed during that downtime?
* If the system provider no longer supports the technology, are there other vendors that can?
* How much time is being wasted?
* How much money is associated with that lost time or lost opportunity?
* How deeply is poor user experience contributing to staffing or customer attrition rates?

## Step 3 - Identify a Task Force

Put together a team of internal and external users and stakeholders who would benefit most from legacy system optimization—or have the most to lose from not taking action. Start by identifying user types, then choose a representative from each group. Involve customers and/or partners and vendors where needed—you can do that with a quick, simple email survey to assess ask what’s working, and what’s not.

## Step 4 - Complete a Deeper Assessment (SWOT)

Complete a [SWOT analysis](https://www.wordstream.com/blog/ws/2017/12/20/swot-analysis). Each user type representative can work through a SWOT analysis for the parts of the system they use most. Below are a few questions from each category to aid in your system analysis.

* **Strengths** – What is currently working well in your system? What do you like about the workflow that makes tasks easier? Does the system process things quickly?
* **Weaknesses** – Are there times when you need to enter data into two or more different systems that don’t sync with one another? Does the current workflow leave too much room for human error by needing too many manual entries? Is there a lack of support from your current vendor?
* **Opportunities** – Can you process more orders? Can you improve the customer experience? Can you save money by consolidating or integrating existing workflows?
* **Threats** – Will you fall behind the competition? Does your system get in the way of exceptional customer service? Are employee onboarding and retention stressed by software or organizational shortcomings?

## Step 5 - Compare Your Options to Make Informed Decisions

Now that you’ve outlined the issues, complied a team, and identified areas that require priority attention, you’re ready to weigh your options for action.

**What are the options?**

* Hire a consultant for a deeper analysis
* Make updates to the current system with your incumbent provider
* Implement integrated solutions to keep current functionality and improve productivity
* Start devising a plan to replace your system with a brand-new solution
  + Review both off-the-shelf and custom solutions

**Questions to consider when determining the right decision for your company…**

* Do you have or know a trustworthy development partner who understands the software?
* What is the cost difference between making updates and buying new? Consider integrations to mitigate costs and risks!
* Is the plan for updating going to tackle a symptom of an outdated system or the root cause?
* Do you need to hire a consultant or get a business analysis to get a deeper assessment?

Buildable has helped several hundred companies assess their software needs. We can help you, too.

[**Let's talk.**](https://www.buildableworks.com/contact)

We present two controlled experiments conducted with master students and practitioners and a case study conducted with practitioners to evaluate the use of MELIS (Migration Environment for Legacy Information Systems) for the migration of legacy COBOL programs to the web. MELIS has been developed as an Eclipse plug-in within a technology transfer project conducted with a small software company [[16]](https://www.sciencedirect.com/science/article/abs/pii/S0950584908000694" \l "bib16). The partner company has developed and marketed in the last 30 years several COBOL systems that need to be migrated to the web, due to the increasing requests of the customers. The goal of the technology transfer project was to define a systematic migration strategy and the supporting tools to migrate these COBOL systems to the web and make the partner company an owner of the developed technology. The goal of the controlled experiments and case study was to evaluate the effectiveness of introducing MELIS in the partner company and compare it with traditional software development environments. The results of the overall experimentation show that the use of MELIS increases the productivity and reduces the gap between novice and expert software engineers.

## Introduction

Legacy systems typically form the backbone of the information flow within organizations and are the main driver to consolidate information on their business. In case one of these systems stops working, the business might be dramatically influenced. In the literature different solutions have been proposed to replace such systems [9], [32], [40]. Typical solutions include: discarding the legacy system and building a replacement system, freezing the system and using it as a component of a new larger system, and modifying the system. Changes may range from a simplification of the system through a reduction of size and complexity, to preventive maintenance operations such as redocumentation, restructuring, and reengineering, to an adaptive maintenance process entailing interface modification, wrapping, and migration [32]. These alternatives are not mutually exclusive and the decision of the approach to use is generally based on an assessment of the quality and business value of the system [4]. Often other non-technical factors influence this decision, as for an example the need of software enterprises to move their legacy systems to a modern infrastructure in order to remain competitive in the global market [3]. In this case, system migration is the only viable alternative [3], since the risk of replacing a legacy system might be unsustainable [39].

The migration of a legacy system is a complex task, which is influenced by several concerns (e.g. its decomposability, budget, technical and time constraints, etc…). In order to preserve the past investments, and reduce risks and development costs the encapsulation of legacy systems through wrapping technologies is a viable alternative [8], [14], [23], [39], [41]. Wrapping technologies are now mature and also include commercial solutions (e.g., WebSphere Studio Enterprise Developer [27] offered by IBM). Unfortunately, even when based on wrapping technologies the legacy system migration often requires ad hoc solutions that needs to be assessed.

To determine whether a new technology is successful, we should look at whether it solves a problem and it is commercially viable and at how long the innovation takes to become accepted in practice. Furthermore, it should be also useful to understand how to increase the odds that the technology choice is the right one [33]. Redwine and Riddle [36] have observed that the time needed for a software technology to mature to the point that it can be diffused to the practitioners of a given company ranges between 15 and 20 years. Nevertheless, this is a too long time in the software market, where the time-to-market is essential to keep competitive [33]. A consequence is that innovative technologies are often adopted before a clear evidence of their advantages. To overcome these difficulties experimentation is necessary [4], [25] and should be performed with researchers, practitioners, and users to assess innovative technologies and tools. Experimentation requires a research plan that will take place over the years. Generally, experiments are formal, rigorous, and controlled investigations normally conducted in laboratory environment to provide a high level of control. In case it is not possible to randomly assign subjects to different treatments [11], [44] quasi-experiments can be alternatively performed. Experimentation should be used to enable the identification of technologies ready to be transferred to particular organizations and market segments, so an evaluation of these technologies in typical situations or pilot projects is also required. Generally, such an evaluation cannot be conducted in a controlled way, so case studies have to be performed, where the activities concerning the use of the technology are observed through the collection and the analysis of relevant data [4], [44].

In [16], we have presented a process and a supporting environment, named MELIS (Migration Environment for Legacy Information Systems), for the migration of ACUCOBOL-GT legacy systems to the web. The migration process and tool were developed within a technology transfer project conducted in cooperation with a small software company. The goal of the company was to become an owner of the technology in order to migrate their legacy systems according to the new technology and business change requirements of their customers. In that paper, we have also presented the results of a preliminary case study conducted on a legacy system of the partner company to validate the approach and to get feedbacks about the tool.

In this paper, we evaluate the effectiveness of using MELIS during the migration of COBOL legacy systems to the web, in particular concerning the productivity improvement. To this aim, we have conducted user studies to compare the support given to the migration process defined in [16] by MELIS against the support given by the COBOL development environment used within the partner company, namely ACUBENCH [1], combined with the web development environment integrated in MELIS, namely Lomboz [26]. It is worth noting that while the company has a long experience of COBOL development (in particular with ACUBENCH), web development is still in the future plans, together with the migration of the marketed COBOL legacy systems to the web. This means that there was not an established web development environment available in the company that we could compare with MELIS in the user studies and for this reason we decided to use Lomboz. In particular, we present here the results of two controlled experiments and a case study. The first controlled experiment was conducted using master students in Computer Science, while the replicated experiment was conducted within the laboratory of the partner company and involved professional programmers. The goal was to evaluate how much the use of MELIS improves the productivity of software engineers and reduces the gap between expert and novice programmers. As the nature of controlled experiments requires that migration tasks are completed in a few hours, we had to necessarily select for these tasks small COBOL programs. To evaluate the benefits of MELIS in a real environment we also conducted a case study, where the key practitioners of the partner company were asked to migrate two subsystems of a relevant system with the MELIS tool and the development environments ACUBENCH and Lomboz.

The remainder of the paper is organized as follows: Section 2 presents related work. The developed migration technology in terms of strategy, migration process, and supporting tool is highlighted in Section 3. The controlled experiments and the cases studies together with the achieved results are discussed in Sections 4 and 5, respectively. Final remarks and future work conclude the paper.

## Section snippets

## **Related work**

Internet is an extremely vital technology that is changing the way in which organizations conduct their business and interact with the partners and the customers [3]. To take advantage of the Internet open architecture, most companies discard the existing software systems and then develop new systems that meet the new needs of the company business. However, economic and technical constraints make the development of new software systems impossible in most cases. Therefore, the migration

## **Industrial context and migration strategy**

In this section, we briefly describe the technology transfer project in which the user studies described in this paper have been conducted. In particular, we describe the software development context of the partner company, the results of the assessment of the legacy systems that the company needs to migrate, the defined migration strategy and process and the tool that we have developed to support this process. More details can be found in [16].

## **Controlled experiments**

In this section, we report on the results of two controlled experiments, which have been conducted with master students in Computer Science at the University of Salerno and professional programmers of our partners company, respectively. These experiments aimed at assessing the effectiveness of using MELIS in the migration of ACUCOBOL-GT programs compared to ACUBENCH, the software environment used within the partner company for COBOL development, and the Lomboz Eclipse plug-in for web

## **Case study**

To evaluate how much MELIS improves the productivity of software engineers with respect to traditional development tools in a real migration context, we performed a case study. In particular, we selected the most business and critical legacy system of the partner company. This system is the oldest legacy system of the partner company, developed, evolved, and marketed over the last 30 years and it is used by many important customers of the partner company to support payroll, tax, and social

## **Conclusion**

In this paper, we have presented the results of two controlled experiments and a case study to evaluate the use of MELIS (Migration Environment for Legacy Information Systems) for the migration of legacy COBOL programs to the web. MELIS has been developed as an Eclipse plug-in within a technology transfer project conducted with a small software company [16]. The partner company has developed and marketed in the last 30 years several COBOL systems that need to be migrated to the web, due to the

## Acknowledgement

The authors thank Vincenzo Venezia, Giovanni Vildacci, as well as the programmers of MTSys s.r.l., our partner company, for the stimulating discussion and the precious suggestions. Special thanks are also due to the practitioners of the partner company, the academic researchers, and the students who, in one way or another, contributed to carry out the experiments presented in this paper.

The first laboratory information management systems ([LIMS](https://en.wikipedia.org/wiki/Laboratory_information_management_system)) were designed as simple tracking tools that enabled systematic control of workflows in regulated laboratories. More recently, LIMS software has evolved into something more akin to an enterprise resource planning tool for the lab that has the ability to manage many different aspects of operations across the full data lifecycle – resource management/scheduling, assay data management, data mining, data analysis, case-centric clinical data, and electronic laboratory notebook (ELN) integration.

Because of today’s rapidly changing business environment, many scientific and R&D-oriented companies find themselves in a situation where their legacy LIMS software struggles to meet business needs. Whether due to increases in data volume, regulatory constraints, M&As, globalization, outsourcing, or a myriad of other reasons, the reality is that many legacy systems are becoming extremely costly to manage and are not able to adequately address changing business requirements. The improved functionality and flexibility inherent in modern COTS systems creants a powerful incentive for these companies to explore the possibility of replacing a legacy system with a new commercial LIMS.

Replacing an aging or legacy LIMS software is no small matter, however. Oftentimes, companies have years of historical data and knowledge stored in their existing LIMS that must be migrated over to the new system. Such a project requires large investments of money, resources and time – costing anywhere from hundreds of thousands to millions of dollars and requiring hundreds of person-days to implement.

The important question to answer before embarking on a LIMS migration project is: Do you really need to do it? Is a LIMS migration your best option, or would simply reworking your current system provide the best risk/reward ratio? Before initiating a LIMS migration, it is important to conduct a thorough analysis of the options. In this blog (part 1 of a 2-part series), we’ll explore some of the steps involved in re-architecting/re-engineering your current system, and the benefits that may be captured from doing so.

**Re-engineering/Re-architecting Your Current System**

The reward of migrating to a new LIMS is not always worth the risks, especially when the legacy system is complex and highly-customized. If your organization contains skilled IT professionals who have created a highly customized system to meet your unique business needs, for example, the benefits of switching to a new system may not be worth the cost and time required to accomplish the migration. In this case, the best approach is to re-engineer your current system to improve its ability to meet your business needs. The steps involved in this re-engineering process should include the following:

**Modify and enhance workflow.**Before starting in on re-engineering your system, it’s best to perform a [workflow analysis](https://astrixinc.com/business-process-analysis/) to help optimize your workflow. The future-state requirements that come out of this analysis will then be used to guide the re-engineering process of the legacy system.

**Implement new features.**Sometimes, the reason the legacy system is not meeting business needs is because it was not properly implemented in the first place. The project team should [include specialists](https://astrixinc.com/professional-services/scientific-technology-platforms/) who understand all the configurable functionality of the legacy system, and are able to use this knowledge to configure as much of the future-state workflows into the system as possible.

**Eliminate and/or Reduce Customization.**Extensive customization to satisfy requirements can make a project significantly more expensive and time-consuming, as well as dramatically increase the complexity of future maintenance/migration/validation activities. Customization can also increase system errors. The goal for your re-engineering project should be to implement as much out-of-the-box functionality and eliminate as much customization as possible.

**Integrate other systems & instruments.**A system re-engineering project provides the perfect opportunity to [design an IT architecture](https://astrixinc.com/practical-enterprise-architecture-for-the-laboratory/) that provides for a more integrated laboratory environment to meet business needs. What instruments can be integrated with your system that currently require manual data transcription processes? What other systems can be integrated with your legacy LIMS to provide for better [digital continuity](https://astrixinc.com/driving-innovation-and-efficiency-in-the-lab-through-digital-continuity/) to drive innovation?

**Leverage new technology.**The secure storage of data files in a wide variety of formats with associated metadata is paramount for an organization. Newer cloud-based options for data storage (e.g., AWS) can be set up with relative ease, and provide for a level of organizational agility not possible with proprietary databases stored on the company servers. A legacy LIMS re-engineering project provides a perfect opportunity to reduce costs and improve scalability by leveraging cloud-technology for data storage.

**Benefits of Reworking Your Current System**

A number of important benefits can be realized when you rework your current system as opposed to migrating to a new system:

**Cost Savings.**The scope of a project involving re-engineering/re-architecting your current system will typically be less than the cost of migrating to a new system. In addition, this option allows you to avoid the purchase of another software license and the loss of your original dollar investment.

**Easier data migration.**Data migration from a legacy system, and validating the migrated data, can be a challenging and time-consuming effort. Reworking your current system allows you to avoid this process entirely, or at most migrate the data to the cloud, which is typically much less involved than migrating to another LIMS system.

**Minimal impact on business continuity.**Transitioning from one LIMS software to another inevitably affects other systems and can result in downtime and delays in data flow across systems. This issue is significantly reduced when you are simply re-architecting/re-engineering your current system.

**Reduction in resources, training and validation.**Since company personnel are already familiar with the system, a LIMS re-engineering project will typically require less personnel and person-days to accomplish, ultimately saving costs. Additionally, personnel will require less training on the new system, and system validation will be less involved, when reworking your current system as opposed to a full LIMS migration project.

**Happy lab!**Users are generally happy when they don’t have to learn a new system, and the system they have been using is dramatically improved in terms of its ability to facilitate efficient, optimized and upgraded workflows.

**Conclusion**

When a legacy LIMS is no longer meeting business needs, re-architecting/re-configuring your current system is sometimes the best way to solve the problem. In order to determine when it is best to go this route over migrating your legacy LIMS to a new system, it is wise to enlist the support of a quality informatics consultant like [Astrix](https://astrixinc.com/).

Our professionals can perform a [LIMS Migration Assessment™](https://astrixinc.com/the-astrix-lims-migration-assessment/) that will analyze the risk/reward ratio on all the different possibilities available (version upgrade, LIMS migration or legacy system rework) to improve your legacy LIMS. Once you have determined which option maximizes business value for your organization, Astrix can help. Our professionals have the skills and expertise necessary to architect, implement, integrate, validate and support best in class solutions for your organization’s laboratory environment.

If you would like to learn more about our **LIMS Software Migration Assessment™**, or if you would like to have an initial, no obligations consultation with an Astrix informatics expert to explore how to optimize your laboratory informatics strategy, please feel free to [contact us](https://astrixinc.com/contact-us/).

In part 2 of this blog series, we will cover some of the benefits and challenges involved with migrating your legacy LIMS to a new system.

Legacy systems are crucial for organizations. Their impact, however, is somewhat negative. How so? Since legacy systems were created for a specific and immediate use back in the days, they are now holding many companies back, being unable to adapt to today’s business environment. Yet, most companies we speak with still rely on on-premises solutions to run their business. So what exactly is a legacy system, and why is it still around?

## What is a legacy system?

The definition of a legacy system is an obsolete computer system, programming language, software application, process, or technology that is no longer can be maintained, replaced, or easily updated.

It does not mean that the legacy system is unusable. Many organizations or companies still find these systems essential to their daily work. It depends upon you to either upgrade or replaces it.

When evaluating your legacy tools, you need to logically identify issues with your existing solution that affecting your business.

## Here are five common issues to consider



### Does not run on modern hardware

Some legacy tools can’t work well with more modern computers, so you are stuck managing the old system that sometimes way underperformed. Having to maintain old hardware and operating system can be pricey. Yes, virtualization and emulation solutions do exist, but it comes with a cost.

### Lack of sufficient skillsets to maintain

As time passes, people with knowledge about how to manage or enhance the application moves on. Getting someone that can work with an old system can be challenging. You need to re-train your staff on how the legacy system works, which can be increasing your company operating costs. A lot of time and effort is required just to keep the system operational, let alone enhance it.

### Compatibility and security vulnerability

Obsolete technology often not plays well with others, and it is hard to integrate or stay compatible with modern computers. Security breaches are also the main concern with the older system. A lack of security in your legacy hardware and other systems poses a significant vulnerability that difficult to solve.

### Amount of data and users limitation

The amount of data and users continuously increases as your company grows each day. Your legacy application may not have the capacity to store the amount of information you have captured over time. Furthermore, it also may not be efficient enough to handle new users that need access.

### Poor user interface

How we interact with software has changed drastically. Your legacy system may not be able to meet the expectation of the user experience and usability of the software. Poor user experience can reduce productivity, although the core tools behind the interface are great and working correctly.

If any of these problems exist with your company’s legacy software, it’s probably time to finding a way to upgrade or replace the system.

## Why is legacy software still around?

Before you start to replace or to upgrade your legacy system, take a step back, and understand the positive impact that your legacy system can have. Knowing the contribution legacy system makes so that when you want to replace the system, the new solution must deliver the same or more enhanced solution as the old one.

## Here are five reasons why the legacy system still used in some companies

### Work well enough

Many legacy systems are doing their job just fine. The proverb “if it ain’t broke, don’t fix it” applied even when you need to run your legacy application on a mainframe instead of cloud computing.

### Converting existing legacy data to a new system isn’t easy

The way data is stored and retrieved is changing. An older system may have a different algorithm, and some are no longer common. To convert all of the data into an entirely different form to be stored in a modern computing system are quite challenging on the internal IT team.

### Recertifying and revalidating requires considerable effort and cost

In some industries, software that can impact safety must be validated and certified for compliance. This software is critical to the company or customer. Recertifying and validating the new system may take a long time and a high cost.

### Switching the tool is too disruptive to the organization

The downtime, while transitioning to new software, is considered an escalating cost to pay because of a decrease in productivity when the new software is deployed and learned.

### The cost of replacing is higher than the benefits of a new system

There is monetary value in replacing something old with something new. The reduction in cost and increase in business should be significant than the total cost of switching.

Like many things in business, the reasons that companies don’t switch are complicated. But it typically comes down to three things:

1. People are set in their ways.
2. Change is expensive.
3. Implementing new software is difficult and time-consuming.

Here’s a deeper look at each point.

**#1 People are used to the existing system**

Legacy software may seem unnecessarily outdated and complex to outsiders.

But the people who use the tools every day are experts at using them. And in many cases, these systems still perform the function they were bought for.

Take the employees who deal with a legacy call center system every day. These people know exactly how it all works and what to do in different situations.

Now imagine the tech team wants to introduce a new tool.

Even if it is objectively easier to use, employees still have to undergo a lengthy adjustment period.

During this time, many employees will find working with the new tool more complicated than using the old one.

### #2 Switching to a new system is costly and time-consuming

Switching to new systems can be expensive.

The software itself is usually a significant investment. But you also have to factor in the cost of getting people up to speed.

Introducing new technology means you have to put on extensive training for the people who will use it. Everyone from agents to managers will have to learn the new system. Even your IT team will have to spend time learning the new tools.

You’ll also have to build new processes, create training plans, and potentially hire new people.

And any bumps in the road will affect your ability to offer great support.

|  |
| --- |
| **Not making the switch can be costly too** |
| According to [Cybera](https://retailtouchpoints.com/resources/retailers-spend-58-of-their-it-budget-on-legacy-system-maintenance), the average retailer spends 58% of its budget maintaining old systems.  Meanwhile, decision-makers surveyed by [Avanade](https://www.avanade.com/-/media/asset/white-paper/avanade-it-modernization-whitepaper.pdf) believe that modernization will cut costs and increase revenue. |

### #3 Moving existing data to a new system is challenging

Your legacy tools contain years’ worth of data; moving this to a new system is never easy.

(Especially if there is no way to integrate the existing tool with the new one.)

Imagine a legacy CRM filled with customer data. If you want to update to a modern CRM, you need to get all the information out of the legacy system and into the new software.

This typically means hiring a costly developer to make the transfer.

Fail to do this properly, and you’ll lose access to years of essential customer data – affecting your ability to provide great support.

## Why are legacy systems still used?

In this era of technology, many establishments are now adapting to modern systems to meet their business objectives. However, for some, it tends to stick around despite the disadvantages of legacy systems.

Often, companies now want to replace assets that are not broken. In addition, some companies heavily depend on this system and could not risk downtime for it to be upgraded. Replacing something that they use every day could be risky and might cause a problem.

Aside from that, another reason is that most employees are already comfortable with the legacy system. Teaching these employees could take an adequate amount of time and could cause a delay in their outputs.

Likewise, businesses also tend to be reserved in upgrading their legacy system due to replacing it. Some companies cannot afford to switch to a newer version. Some of these expenses might include replaying recent hardware or components to accommodate the latest software.