Legacy system

By Himanshi Liyanage

MSc in CS (Poland), Ph.D. in CS (reading) (Poland)

Legacy System
Higher Maintenance Cost
Data loss
Technical limitation
Lower performance
New System
Lower Maintainance cost
More efficient
Scalable Application
Higher performance

What is a legacy system?

A legacy system is outdated computing software and/or hardware that is still in use.

The system still meets the needs it was originally designed for but doesn’t allow for growth.

What a legacy system does now for the company is all it will ever do.

A legacy system’s older technology won’t allow it to interact with newer systems.

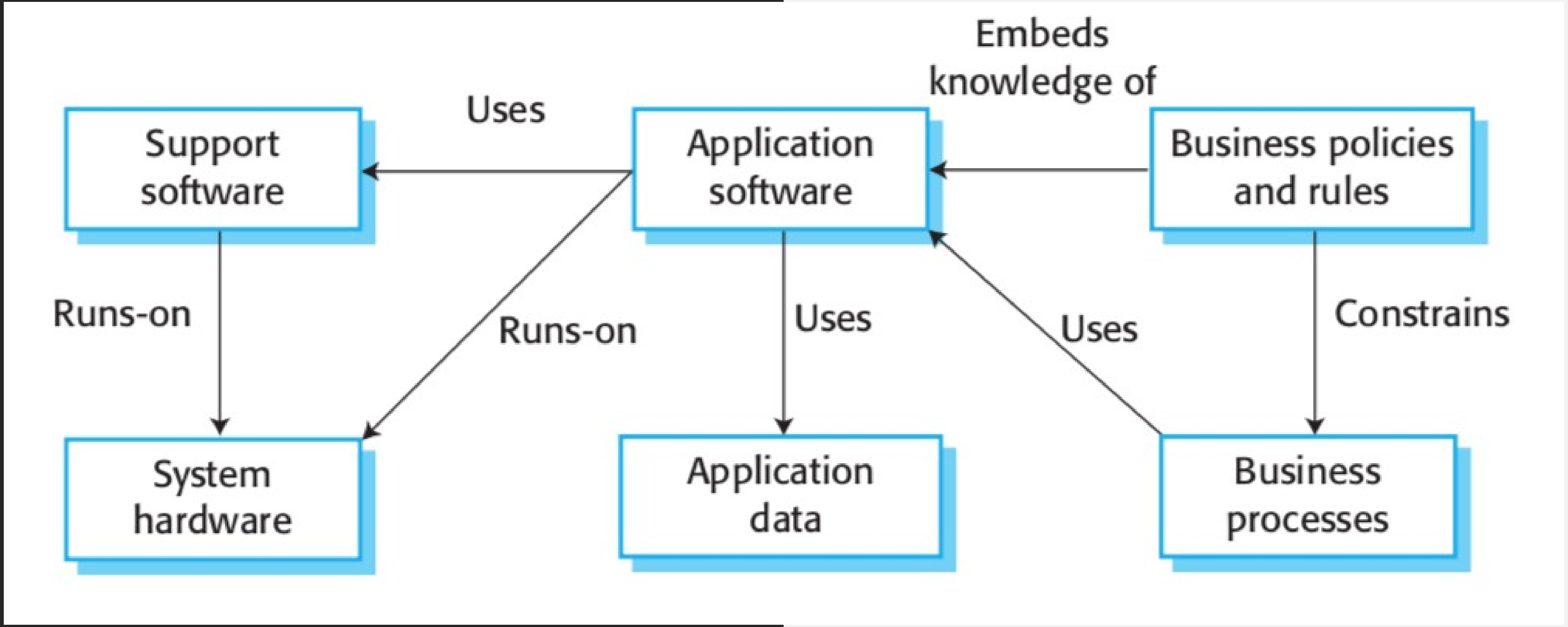
Why company would continue to use a legacy system?

- ***Investment***: Although maintaining a legacy system is expensive over time, upgrading to a new system requires an up-front investment, both in dollars and manpower.
- ***Fear:*** Change is hard and moving a whole company or even a single department to a new system can inspire some internal resistance.
- ***Difficulty:*** The legacy software may be built with an obsolete programming language that makes it hard to find personnel with the skills to make the migration. There may be little documentation about the system and the original developers have left the company. Sometimes simply planning the migration of data from a legacy system and defining the scope of requirements for a new system are overwhelming.

The elements of a legacy system

- Legacy systems are not just software systems but are broader sociotechnical systems that include hardware, software, libraries, and other supporting software and business processes.
- Following diagram shows the logical parts of a legacy system and their relationships.

The elements of a legacy system



logical parts of a legacy system and their relationships

System hardware - Legacy systems may have been written for hardware that is no longer available, that is expensive to maintain, and that may not be compatible with current organizational IT purchasing policies.

Support software - The legacy system may rely on a range of support software from the operating system and utilities provided by the hardware manufacturer through to the compilers used for system development. Again, these may be obsolete and no longer supported by their original providers.

logical parts of a legacy system and their relationships cont.

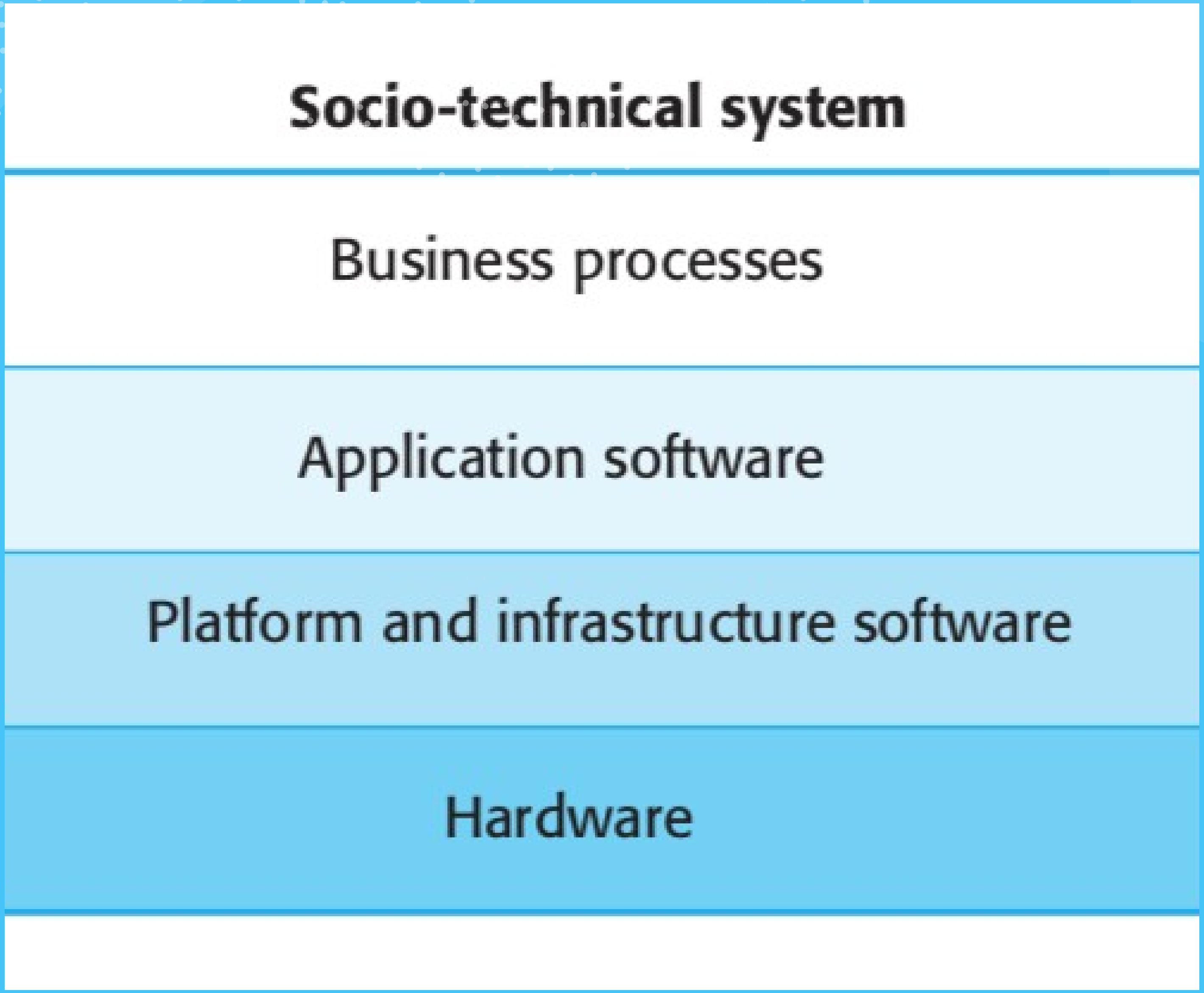
Application software - The application system that provides the business services is usually made up of a number of application programs that have been developed at different times. Some of these programs will also be part of other application software systems.

Application data - These data are processed by the application system. In many legacy systems, a massive volume of data has accumulated over the lifetime of the system. This data may be inconsistent, may be duplicated in several files, and may be spread over a number of different databases.

logical parts of a legacy system and their relationships cont.

- ***Business processes -*** These processes are used in the business to achieve some business objective. An example of a business process in an insurance company would be issuing an insurance policy; in a manufacturing company, a business process would be accepting an order for products and setting up the associated manufacturing process. Business processes may be designed around a legacy system and constrained by the functionality that it provides.
- ***Business policies and rules -*** These are definitions of how the business should be carried out and constraints on the business. Use of the legacy application system may be embedded in these policies and rules.

Legacy system layers



Legacy system layers CONT.

- Each layer depends on the layer immediately below it and interfaces with that layer. If interfaces are maintained, then you should be able to make changes within a layer without affecting either of the adjacent layers. In practice, however, this simple encapsulation is an oversimplification, and changes to one layer of the system may require consequent changes to layers that are both above and below the changed level.

REASONS

- Changing one layer in the system may introduce new facilities, and higher layers in the system may then be changed to take advantage of these facilities. For example, a new database introduced at the support software layer may include facilities to access the data through a web browser, and business processes may be modified to take advantage of this facility.
- Changing the software may slow the system down so that new hardware is needed to improve the system performance. The increase in performance from the new hardware may then mean that further software changes that were previously impractical become possible.

REASONS cont.

- It is often impossible to maintain hardware interfaces, especially if new hardware is introduced. This is a particular problem in embedded systems where there is a tight coupling between software and hardware. Major changes to the application software may be required to make effective use of the new hardware.

Legacy system cont.

- It is difficult to know exactly how much legacy code is still in use, but, as an indicator, industry has estimated that there are more than 200 billion lines of COBOL code in current business systems.
- COBOL is a programming language designed for writing business systems, and it was the main business development language from the 1960s to the 1990s, particularly in the finance industry (Mitchell 2012).
- These programs still work effectively and efficiently, and the companies using them see no need to change them.
- A major problem that they face, however, is a shortage of COBOL programmers as the original developers of the system retire. Universities no longer teach COBOL, and younger software engineers are more interested in programming in modern languages.

Legacy system cont.

Skill shortages are only one of the problems of maintaining business legacy systems.

Other issues include security vulnerabilities because these systems were developed before the widespread use of the Internet and problems in interfacing with systems written in modern programming languages.

The original software tool supplier may be out of business or may no longer maintain the support tools used to develop the system.

The system hardware may be obsolete and so increasingly expensive to maintain.

Legacy system replacement

- Why then do businesses not simply replace these systems with more modern equivalents?
- The simple answer to this question is that it is too expensive and too risky to do so.
- If a legacy system works effectively, the costs of replacement may exceed the savings that come from the reduced support costs of a new system.
- Scrapping legacy systems and replacing them with more modern software open up the possibility of things going wrong and the new system failing to meet the needs of the business.
- Managers try to minimize those risks and therefore do not want to face the uncertainties of new software systems.

Legacy system replacement cont.

- We can see some of the problems of legacy system replacement when analyzing a legacy system replacement project in a large organization. This enterprise used more than 150 legacy systems to run its business. It decided to replace all of these systems with a single, centrally maintained ERP system. For a number of business and technology reasons, the new system development was a failure, and it did not deliver the improvements promised. After spending more than £10 million, only a part of the new system was operational, and it worked less effectively than the systems it replaced. Users continued to use the older systems but could not integrate these with the part of the new system that had been implemented, so additional manual processing was required.

why it is expensive and risky to replace legacy systems with new systems?

- There is rarely a complete specification of the legacy system. The original specification may have been lost. If a specification exists, it is unlikely that it has been updated with all of the system changes that have been made. Therefore, there is no straightforward way of specifying a new system that is functionally identical to the system that is in use.
- Business processes and the ways in which legacy systems operate are often inextricably intertwined. These processes are likely to have evolved to take advantage of the software's services and to work around the software's shortcomings. If the system is replaced, these processes have to change with potentially unpredictable costs and consequences.

why it is expensive and risky to replace legacy systems with new systems cont.?

- Important business rules may be embedded in the software and may not be documented elsewhere. A business rule is a constraint that applies to some business function and breaking that constraint can have unpredictable consequences for the business. For example, an insurance company may have embedded its rules for assessing the risk of a policy application in its software. If these rules are not maintained, the company may accept high-risk policies that could result in expensive future claims.
- New software development is inherently risky, so that there may be unexpected problems with a new system. It may not be delivered on time and for the price expected.

Legacy system change

- Keeping legacy systems in use avoids the risks of replacement but making changes to existing software inevitably becomes more expensive as systems get older. Legacy software systems that are more than a few years old are particularly expensive to change

Legacy system change cont.

- The program style and usage conventions are inconsistent because different people have been responsible for system changes. This problem adds to the difficulty of understanding the system code.
- Part or all of the system may be implemented using obsolete programming languages. It may be difficult to find people who have knowledge of these languages. Expensive outsourcing of system maintenance may therefore be required.
- System documentation is often inadequate and out of date.
- In some cases, the only documentation is the system source code.

Legacy system change cont.

- Many years of maintenance usually degrades the system structure, making it increasingly difficult to understand. New programs may have been added and interfaced with other parts of the system in an ad hoc way.
- The system may have been optimized for space utilization or execution speed so that it runs effectively on older slower hardware. This normally involves using specific machine and language optimizations, and these usually lead to software that is hard to understand. This causes problems for programmers who have learned modern software engineering techniques and who don’t understand the programming tricks that have been used to optimize the software.

Legacy system change cont.

- The data processed by the system may be maintained in different files that have incompatible structures. There may be data duplication, and the data itself may be out of date, inaccurate, and incomplete. Several databases from different suppliers may be used.

Assessing Legacy systems

- Most organizations have a limited budget for maintaining and upgrading their portfolio of legacy systems. They have to decide how to get the best return on their investment. This involves making a realistic assessment of their legacy systems and then deciding on the most appropriate strategy for evolving these systems. There are four strategic options:

Assessing Legacy systems cont.

- ***Scrap the system completely -*** This option should be chosen when the system is not making an effective contribution to business processes. This usually occurs when business processes have changed since the system was installed and are no longer reliant on the legacy system.
- ***Leave the system unchanged and continue with regular maintenance -*** This option should be chosen when the system is still required but is fairly stable and the system users make relatively few change requests.

Assessing Legacy systems cont.

Reengineer the system to improve its maintainability - This option should be chosen when the system quality has been degraded by change and where new change to the system is still being proposed. This process may include developing new interface components so that the original system can work with other, newer systems.

Replace all or part of the system with a new system - This option should be chosen when factors, such as new hardware, mean that the old system cannot continue in operation, or where off-the-shelf systems would allow the new system to be developed at a reasonable cost. In many cases, an evolutionary replacement strategy can be adopted where major system components are replaced by off- the-shelf systems with other components reused wherever possible.

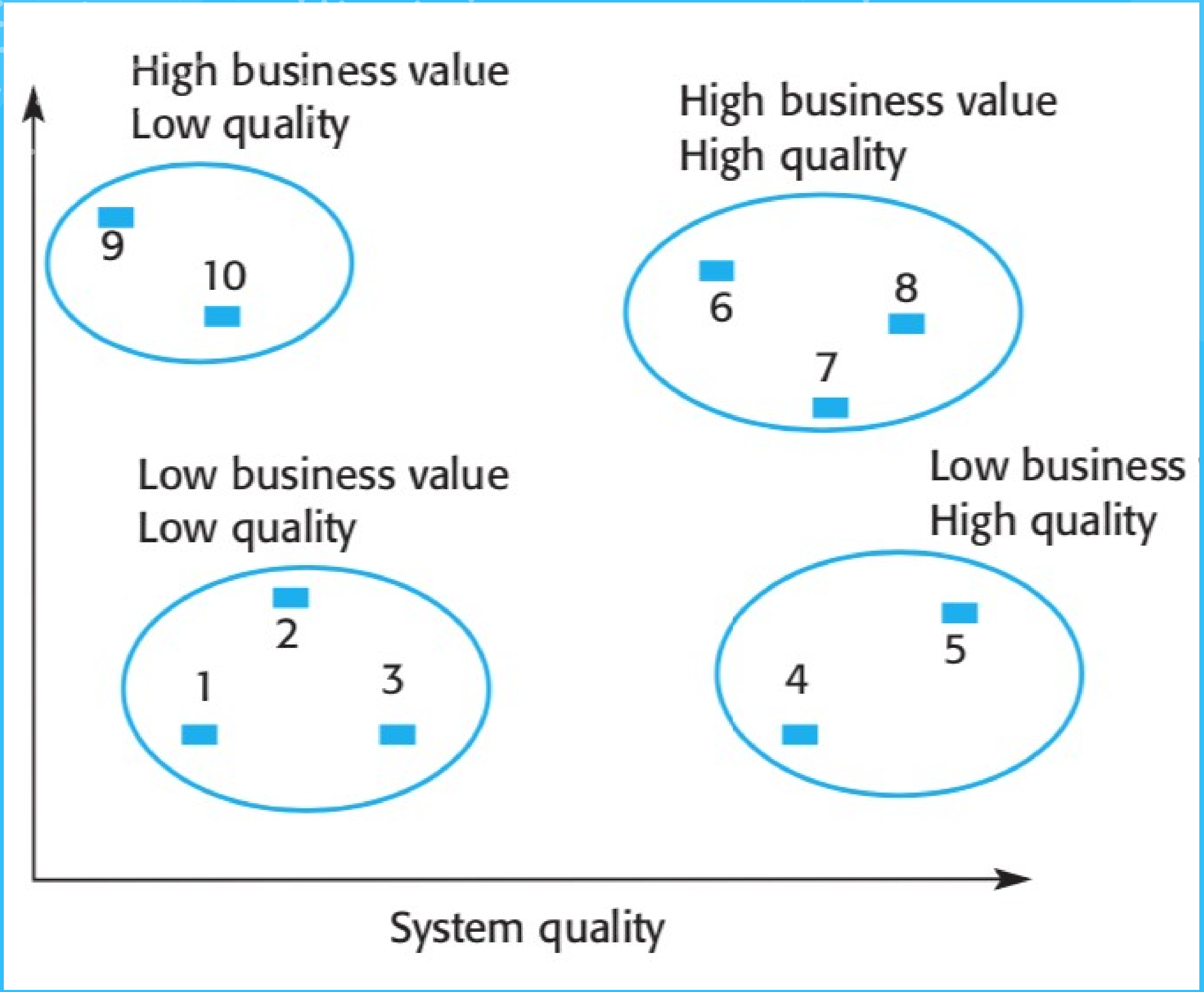
Assessing Legacy systems cont.

- When you are assessing a legacy system, you have to look at it from both a business perspective and a technical perspective (Warren 1998). From a business perspective, you have to decide whether or not the business really needs the system. From a technical perspective, you have to assess the quality of the application software and the system’s support software and hardware. You then use a combination of the business value and the system quality to inform your decision on what to do with the legacy system.

assessing legacy systems cont.

- For example, assume that an organization has 10 legacy systems. You should assess the quality and the business value of each of these systems. You may then create a chart showing relative business value and system quality. In the given diagram, you can see that there are four clusters of systems:

An example of a legacy system assessment



Legacy system categories

- ***<u>Low quality, low business value - </u>*** Keeping these systems in operation will be expensive, and the rate of the return to the business will be fairly small. These systems should be scrapped.
- ***<u>Low quality, high business value - </u>*** These systems are making an important business contribution, so they cannot be scrapped. However, their low quality means that they are expensive to maintain. These systems should be reengineered to improve their quality. They may be replaced, if suitable off-the-shelf systems are available.

Legacy system categories cont.

- ***High quality, low business value -*** These systems don’t contribute much to the business but may not be very expensive to maintain. It is not worth replacing these systems, so normal system maintenance may be continued if expensive changes are not required and the system hardware remains in use. If expensive changes become necessary, the software should be scrapped.
- ***High quality, high business value -***These systems have to be kept in operation. However, their high-quality means that you don’t have to invest in transformation or system replacement. Normal system maintenance should be continued.