
Software Evolution

What is a Legacy System

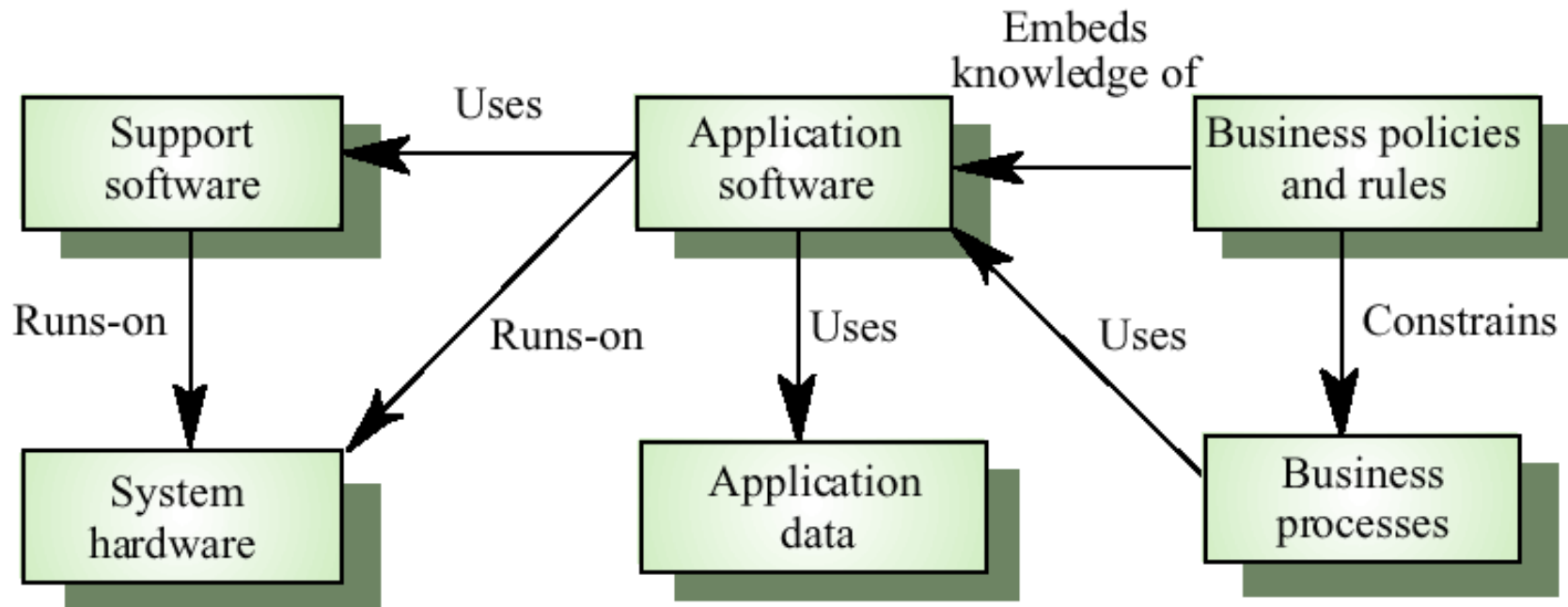
“legacy”

A sum of money, or a specified article, given to another by will; anything handed down by an ancestor or predecessor. — Oxford English Dictionary

- A legacy system is a piece of software that
 - ❑ You have *inherited* and
 - ❑ Is *valuable* to you

 - Typical problems with legacy systems:
 - ❑ original developers *not available*
 - ❑ *outdated* development methods used
 - ❑ extensive patches and *modifications* have been made
 - ❑ *missing* or outdated documentation
-

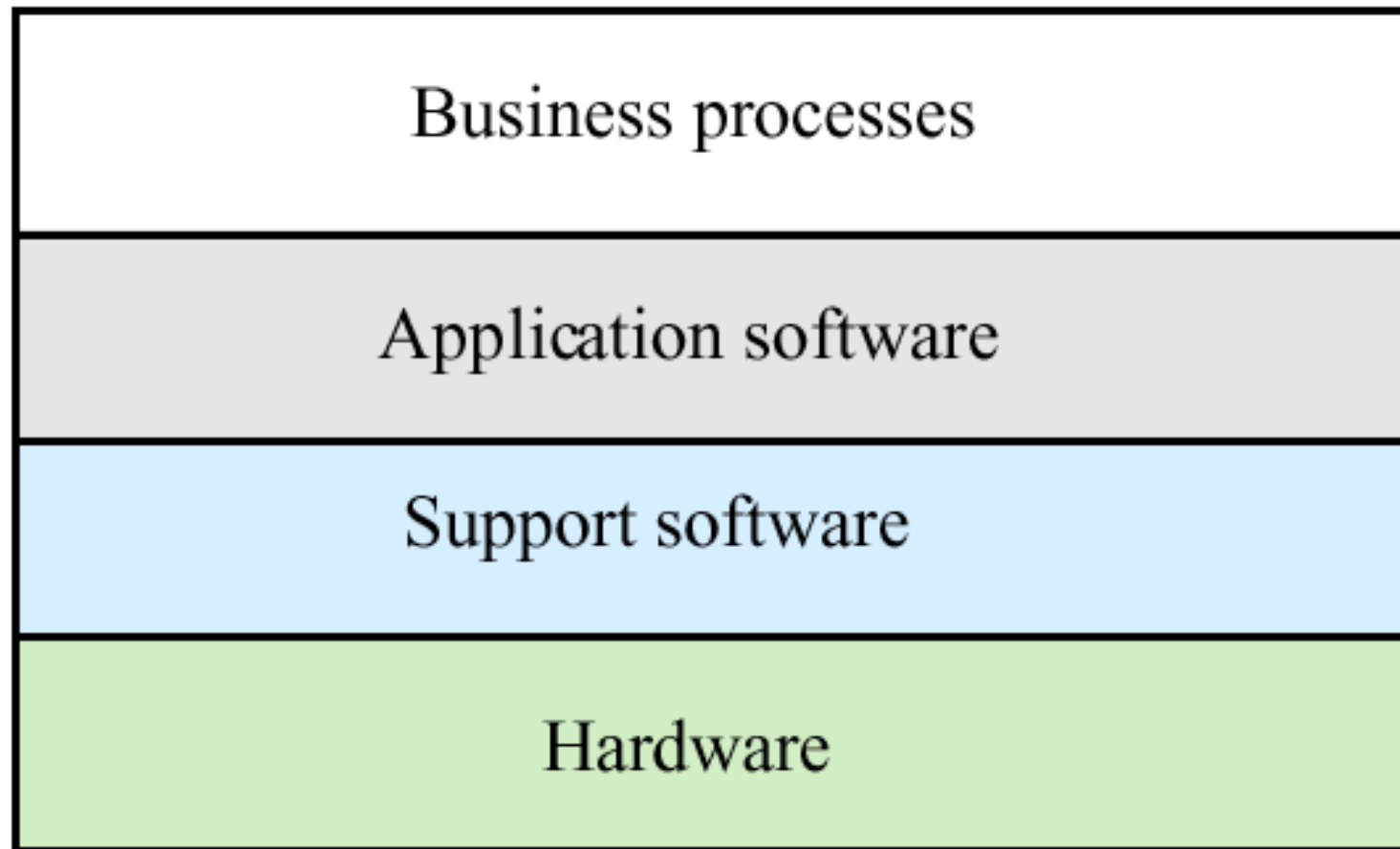
Legacy system structures



- System hardware – Mostly mainframes
- Support software – OS/Compiler support, etc
- Application software – Business services
- Application data – Normally large amount
- Business processes – Processes used in the business
- Business policies and rules – Constraints and rules

Layered model

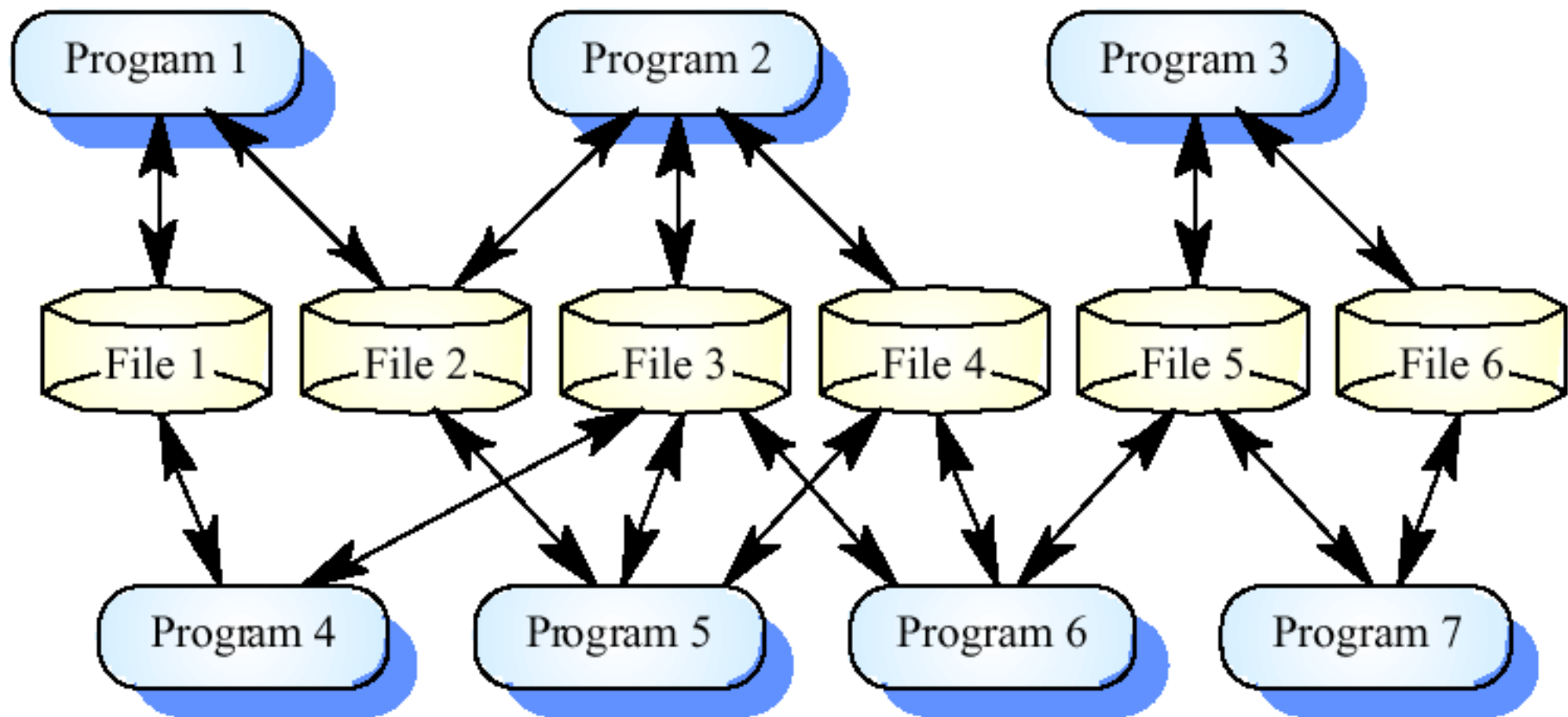
Socio-technical system



System change

- In principle, it should be possible to replace a layer in the system leaving the other layers unchanged
- In practice, this is usually impossible
 - ❑ Changing one layer introduces new facilities and higher level layers must then change to make use of these
 - ❑ Changing the software may slow it down so hardware changes are then required
 - ❑ It is often impossible to maintain hardware interfaces because of the wide gap between mainframes and client-server systems

Legacy application system



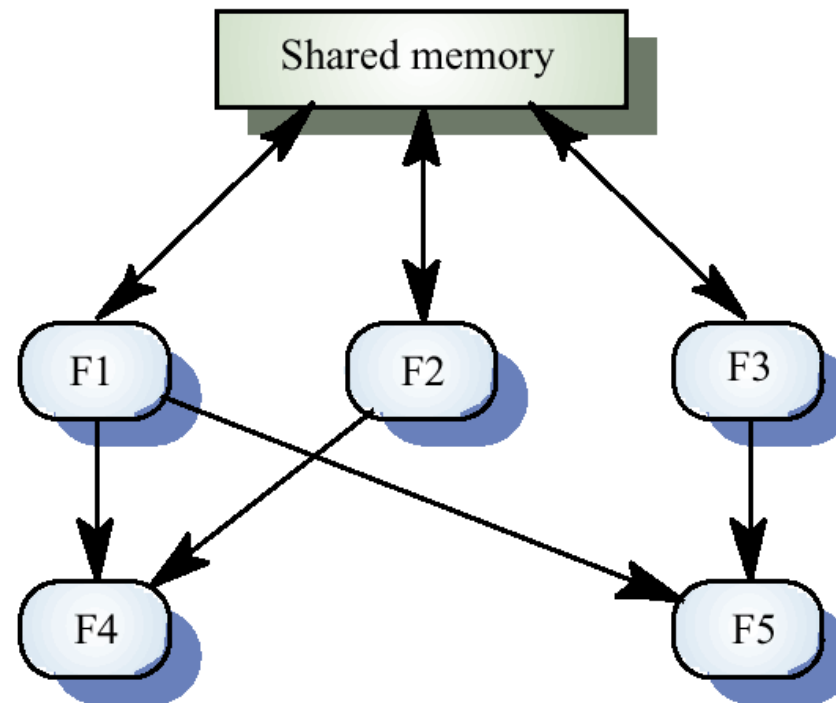
Legacy data

- The system may be file-based with incompatible files. The change required may be to move to a database-management system
 - In legacy systems that use a DBMS the database management system may be obsolete and incompatible with other DBMSs used by the business
 - The teleprocessing monitor may be designed for a particular DB and mainframe. Changing to a new DB may require a new TP monitor
-

Legacy system design

- Most legacy systems were designed before object-oriented development was used
 - Rather than being organised as a set of interacting objects, these systems have been designed using a function-oriented design strategy
 - Several methods and CASE tools are available to support function-oriented design and the approach is still used for many business applications
-

Legacy system design (function-oriented view)



■ Mostly function-oriented design

- ❑ *Batch processing systems* - Data is input and output in batches from a file rather than input and output to a user terminal. Examples of batch processing systems are payroll systems, billing systems, etc.
- ❑ *Transaction processing systems* - Data is input and output as a series of transactions against a database with the transaction generated from a user terminal.

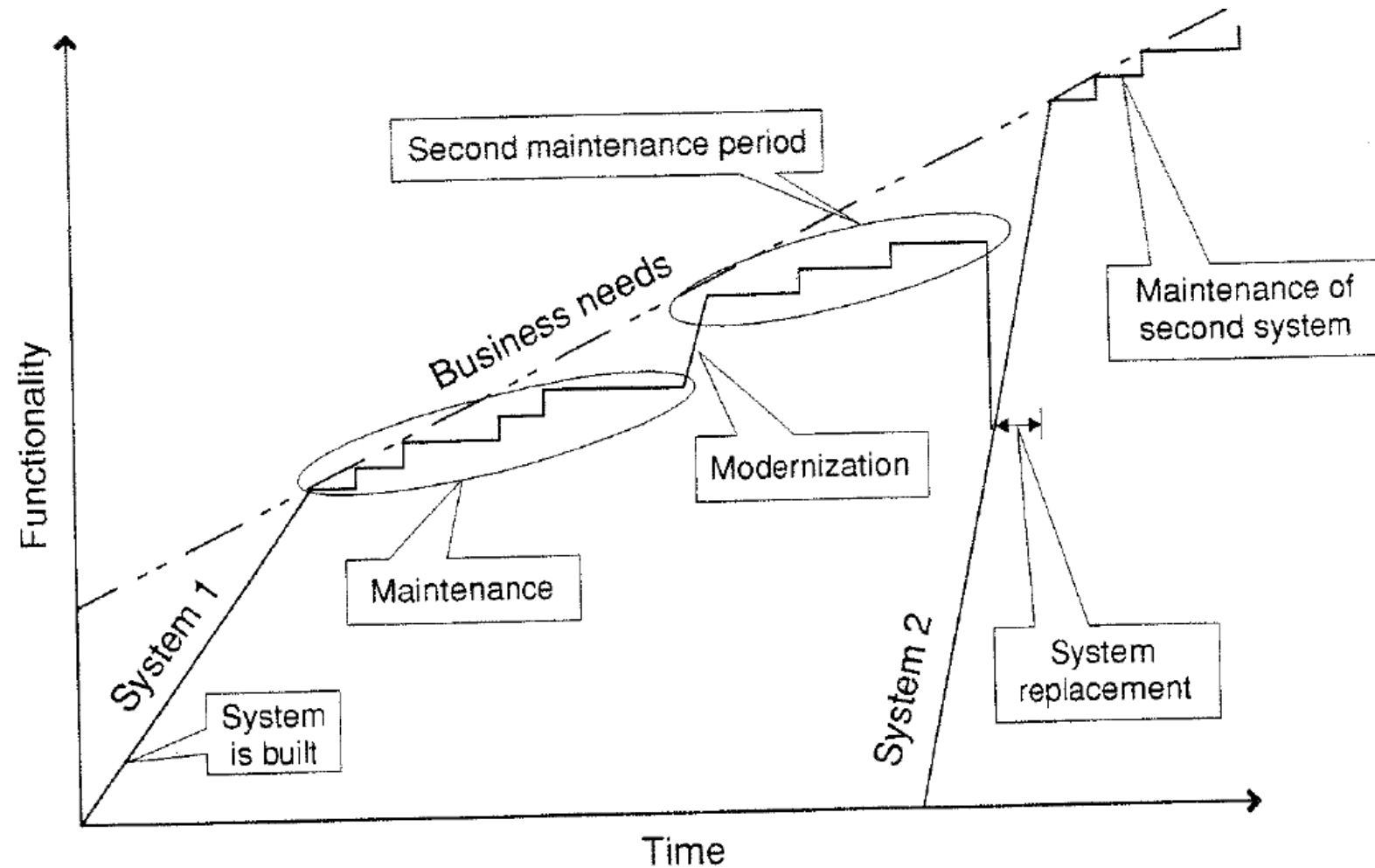
Expensive to change legacy systems

- Different parts implemented by different teams. So, no consistent programming style across the whole system.
- Part or all of the system may be implemented using an obsolete programming language.
 - ❑ difficult to find staff who have knowledge
 - ❑ Expensive outsourcing of system maintenance
- System documentation is often inadequate and out-of-date
 - ❑ Only documentation is the system source code
 - ❑ Sometimes only the executable version is available

Expensive to change legacy systems

- Ad hoc maintenance may have corrupted the system structure
 - The system may have been optimized for space utilization or execution speed
 - Difficulties for programmers who have learned modern software engineering techniques but have not been that have been used.
 - There may be data duplication and the data itself may be out of date, inaccurate and incomplete
-

Evolving Legacy Systems



Maintenance. Modernization. Replacement.

Legacy system evolution

- Organizations that rely on legacy systems must choose a strategy for evolving these systems
 - ❑ Scrap the system completely and modify business processes so that it is no longer required
 - ❑ Continue maintaining the system;
 - ❑ Transform the system by re-engineering to improve its maintainability
 - ❑ Replace the system with a new system
- The strategy chosen should depend on the **system quality** and its **business value**.

Business value assessment

- Assessment should take different viewpoints
 - System end-users
 - Business customers
 - Line managers
 - IT managers
 - Senior managers
 - Interview different stakeholders and collate results
-

System quality assessment

- Business process assessment
 - How well does the business process support the current goals of the business?
 - Environment assessment
 - How effective is the system's environment and how expensive is it to maintain?
 - Application assessment
 - What is the quality of the application software system?
-

Business process assessment

- Use a viewpoint-oriented approach and seek answers from system stakeholders
 - Is there a defined process model and is it followed?
 - Do different parts of the organization use different processes for the same function?
 - How has the process been adapted?
 - What are the relationships with other business processes and are these necessary?
 - Is the process effectively supported by the legacy application software?
 - Example - a travel ordering system may have a low business value because of the widespread use of web-based ordering.
-

Environment assessment

| Factor | Questions |
|--------------------|--|
| Supplier stability | Is the supplier is still in existence? Is the supplier financially stable and likely to continue in existence? If the supplier is no longer in business, does someone else maintain the systems? |
| Failure rate | Does the hardware have a high rate of reported failures? Does the support software crash and force system restarts? |
| Age | How old is the hardware and software? The older the hardware and support software, the more obsolete it will be. It may still function correctly but there could be significant economic and business benefits to moving to more modern systems. |
| Performance | Is the performance of the system adequate? Do performance problems have a significant effect on system users? |

Environment assessment contd...

| | |
|----------------------|--|
| Support requirements | What local support is required by the hardware and software? If there are high costs associated with this support, it may be worth considering system replacement. |
| Maintenance costs | What are the costs of hardware maintenance and support software licences? Older hardware may have higher maintenance costs than modern systems. Support software may have high annual licensing costs. |
| Interoperability | Are there problems interfacing the system to other systems? Can compilers etc. be used with current versions of the operating system? Is hardware emulation required? |

Application assessment

| Factor | Questions |
|-------------------|--|
| Understandability | How difficult is it to understand the source code of the current system? How complex are the control structures that are used? Do variables have meaningful names that reflect their function? |
| Documentation | What system documentation is available? Is the documentation complete, consistent and up-to-date? |
| Data | Is there an explicit data model for the system? To what extent is data duplicated in different files? Is the data used by the system up-to-date and consistent? |
| Performance | Is the performance of the application adequate? Do performance problems have a significant effect on system users? |

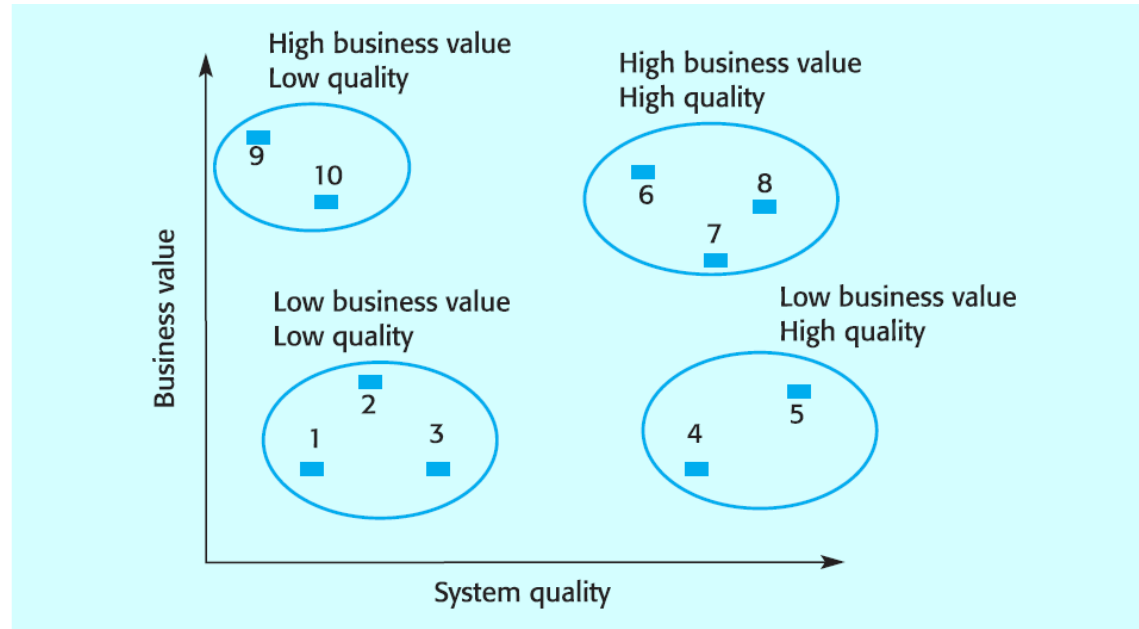
Application assessment contd...

| | |
|--------------------------|---|
| Programming language | Are modern compilers available for the programming language used to develop the system? Is the programming language still used for new system development? |
| Configuration management | Are all versions of all parts of the system managed by a configuration management system? Is there an explicit description of the versions of components that are used in the current system? |
| Test data | Does test data for the system exist? Is there a record of regression tests carried out when new features have been added to the system? |
| Personnel skills | Are there people available who have the skills to maintain the application? Are there only a limited number of people who understand the system? |

Application assessment contd...

- Quantitative data may be collected. For example,
 - Number of system change requests
 - Number of different user interfaces used by the system
 - Volume of data used by the system
-

Legacy system assessment: System quality and business value



- ***Low quality, low business value*** - These are candidates for scrapping.
- ***Low quality, high business value*** - Cannot be scrapped. However, these are candidates for system transformation or replacement if a suitable system is available.
- ***High quality, low business value*** - Normal system maintenance may be continued or they may be scrapped.
- ***High quality, high business value*** - No need to invest in transformation or system replacement. Normal system maintenance should be continued.