
Chapter 28

Software Re-engineering

Software re-engineering

**Reorganising and modifying
existing software systems to make
them more maintainable**

Objectives

To explain why software re-engineering is a **cost-effective** option for system evolution

To describe the **activities** involved in the software re-engineering process

To **distinguish between** software and data re-engineering and to explain the problems of data re-engineering

Topics covered

Source code translation

Reverse engineering

Program structure improvement

Program modularisation

Data re-engineering

System re-engineering and Legacy Systems

Re-structuring or re-writing part or all of a **legacy system** without changing its **functionality**

Applicable where some but not all sub-systems of a larger system require frequent maintenance

Re-engineering involves adding effort to make them **easier to maintain**. The system may be re-structured and **re-documented**

When to re-engineer

When system or subsystem **changes effects** on **system or subsystem** so the system or subsystem should be reengineered

When hardware or software support becomes **obsolete**

When tools to support re-structuring are available

Re-engineering advantages

Reduced risk

- There is a high risk in new software development. There may be development problems, staffing problems and specification problems

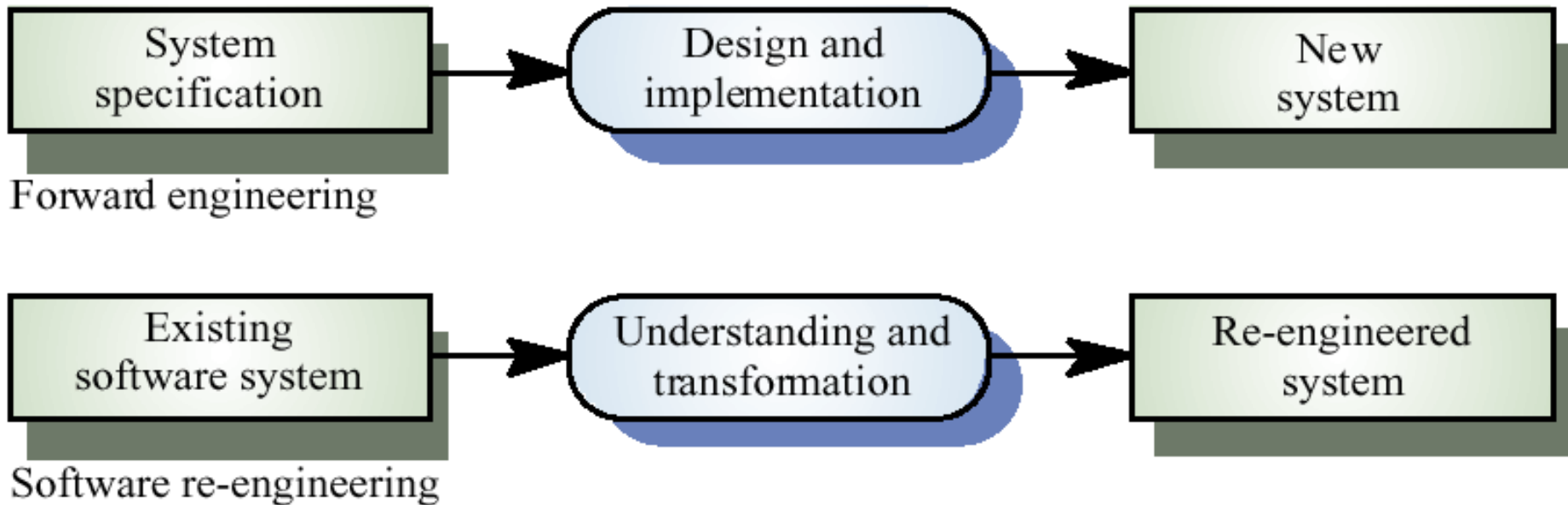
Reduced cost

- The cost of re-engineering is often significantly less than the costs of **developing new software**

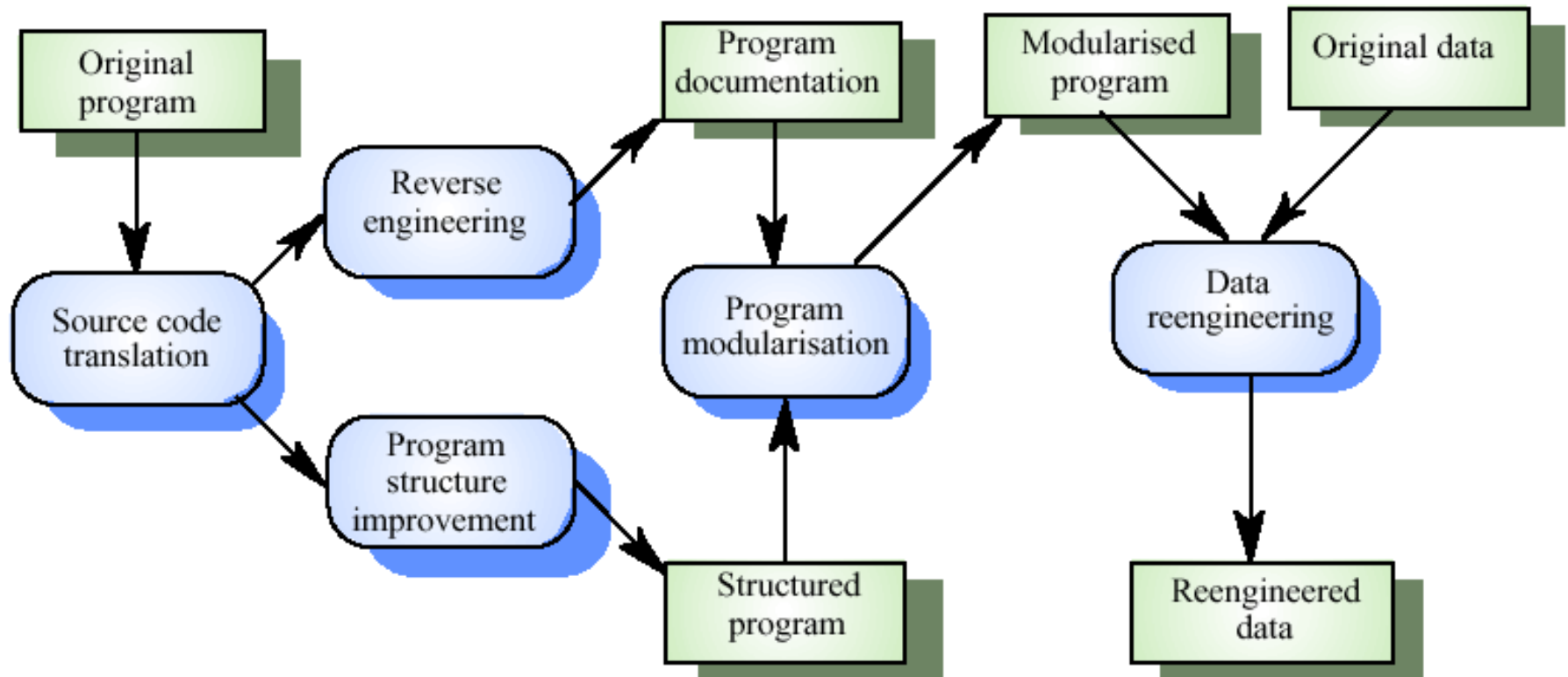
Business process re-engineering

Concerned with **re-designing** business processes to make them more **responsive and more efficient**

Forward engineering and re-engineering



The re-engineering process



Re-engineering cost factors

The quality of the software to be re-engineered

The tool support available for re-engineering

The extent of the data conversion which is required

The availability of expert staff for re-engineering

Source code translation

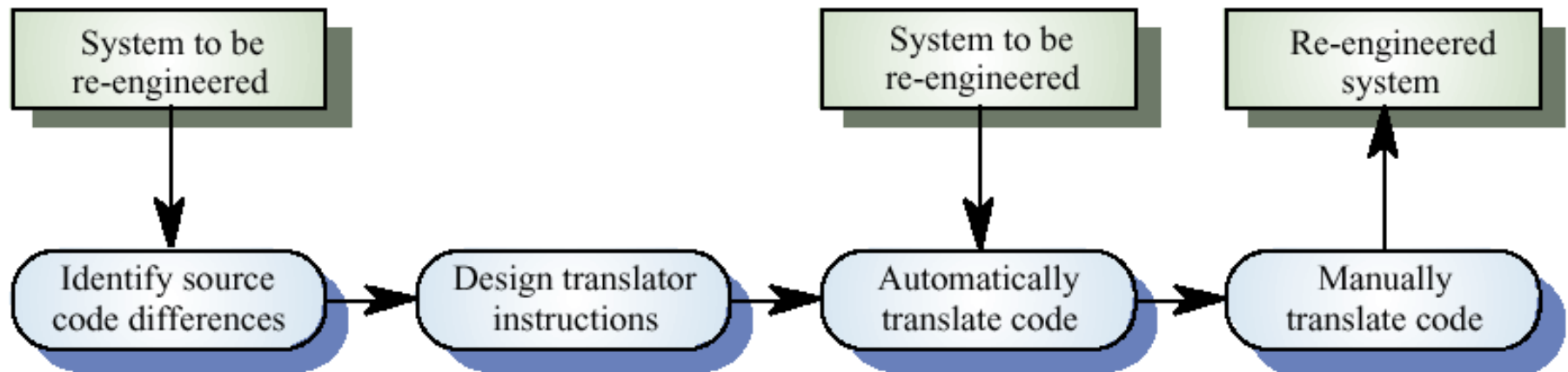
Involves converting the code from one language (or language version) to another e.g. FORTRAN to C

May be necessary because of:

- Hardware platform update
- Staff skill shortages
- Organisational policy changes

Only realistic if an automatic translator is available

The program translation process



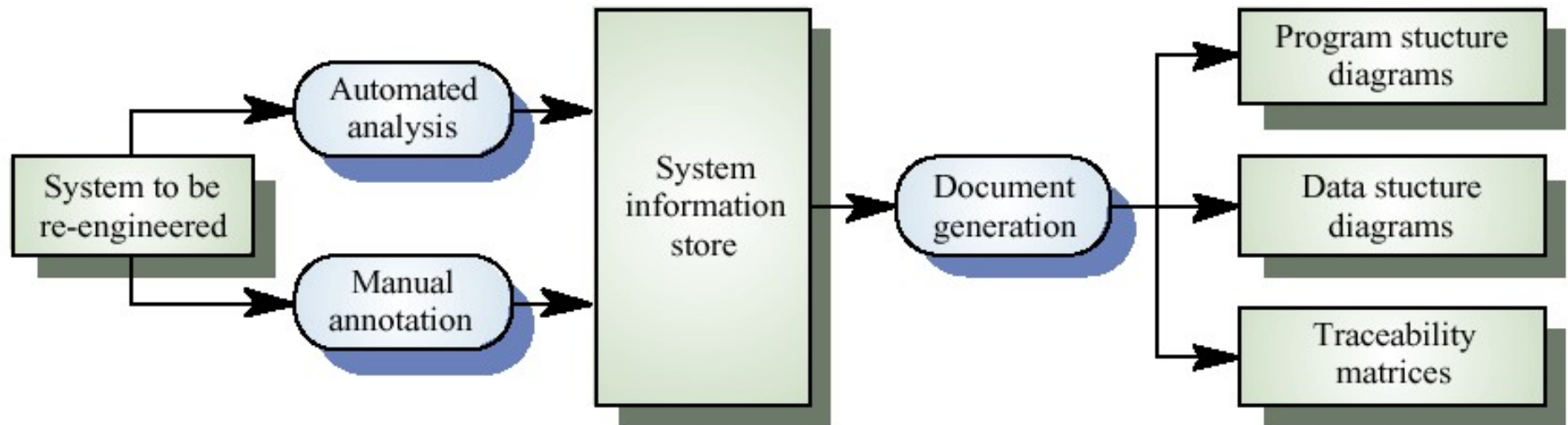
Reverse engineering

Analysing software with a view to understanding its **design and specification**

May be part of a re-engineering process but may also be used to **re-specify** a system for **re-implementation**

Generates information from programs

The reverse engineering process



Program structure improvement

Maintenance tends to corrupt the structure of a program. It becomes **harder and harder to understand**

The program may be automatically restructured to remove **unconditional branches**

Conditions may be **simplified** to make them more **readable**

Spaghetti logic

```
Start:  Get (Time-on, Time-off, Time, Setting, Temp, Switch)
        if Switch = off goto off
        if Switch = on goto on
        goto Cntrld
off:    if Heating-status = on goto Sw-off
        goto loop
on:     if Heating-status = off goto Sw-on
        goto loop
Cntrld: if Time = Time-on goto on
        if Time = Time-off goto off
        if Time < Time-on goto Start
        if Time > Time-off goto Start
        if Temp > Setting then goto off
        if Temp < Setting then goto on
Sw-off: Heating-status := off
        goto Switch
Sw-on:  Heating-status := on
Switch: Switch-heating
loop:   goto Start
```

Structured control logic

```
loop
    -- The Get statement finds values for the given variables from the system's
    -- environment.
    Get (Time-on, Time-off, Time, Setting, Temp, Switch) ;
    case Switch of
        when On => if Heating-status = off then
                        Switch-heating ; Heating-status := on ;
                    end if ;
        when Off => if Heating-status = on then
                        Switch-heating ; Heating-status := off ;
                    end if;
        when Controlled =>
            if Time >= Time-on and Time <= Time-off then
                if Temp > Setting and Heating-status = on then
                    Switch-heating; Heating-status = off;
                elsif Temp < Setting and Heating-status = off then
                    Switch-heating; Heating-status := on ;
                end if;
            end if ;
        end case ;
    end loop ;
```

Condition simplification

-- Complex condition

if not (A > B and (C < D or not (E > F)))...

-- Simplified condition

if (A <= B and (C>= D or E > F)...

Program modularisation

The process of re-organising a program so that related program parts are collected together in a single module

Usually a manual process that is carried out by program inspection and re-organisation

Data re-engineering

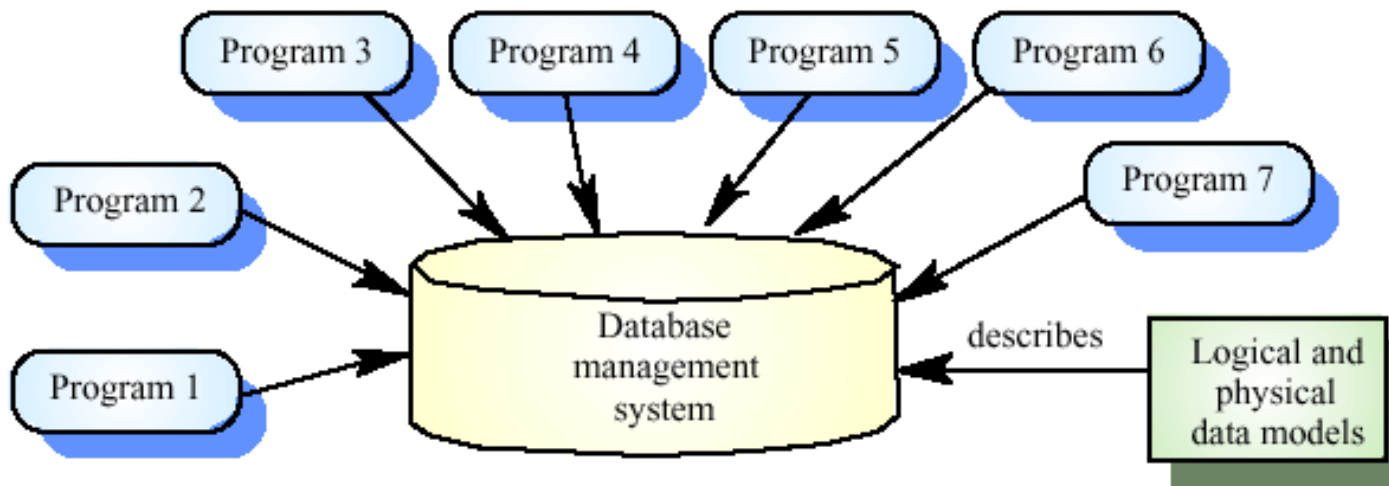
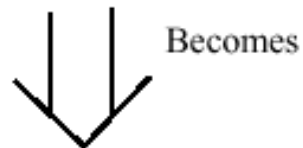
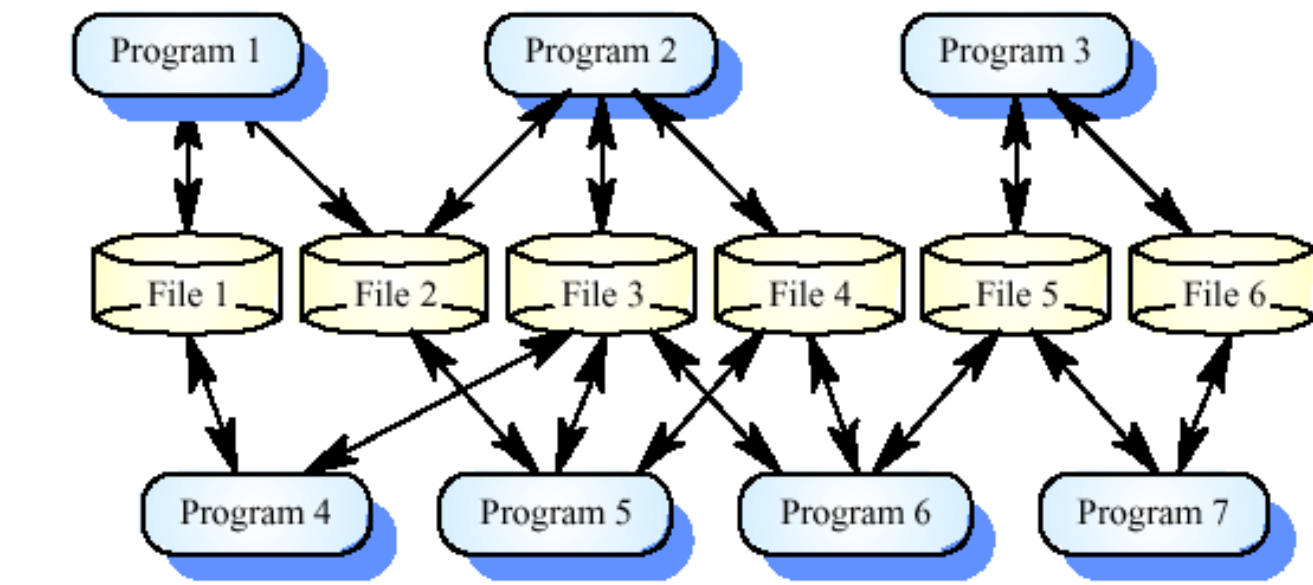
Involves analysing and reorganising the data structures (and sometimes the data values) in a program

May be part of the process of migrating from a file-based system to a DBMS-based system or changing from one DBMS to another

Objective is to create a managed data environment

Approaches to data re-engineering

Approach	Description
Data cleanup	The data records and values are analysed to improve their quality. Duplicates are removed, redundant information is deleted and a consistent format applied to all records. This should not normally require any associated program changes.
Data extension	In this case, the data and associated programs are re-engineered to remove limits on the data processing. This may require changes to programs to increase field lengths, modify upper limits on the tables, etc. The data itself may then have to be rewritten and cleaned up to reflect the program changes.
Data migration	In this case, data is moved into the control of a modern database management system. The data may be stored in separate files or may be managed by an older type of DBMS.



**Data
migration**

Data problems

Data naming problems

- Names may be hard to understand. The same data may have different names in different programs

Field length problems

- The same item may be assigned different lengths in different programs

Record organisation problems

- Records representing the same entity may be organised differently in different programs

Hard-coded literals

No data dictionary

Key points

The objective of re-engineering is to improve the system structure to make it easier to understand and maintain

The re-engineering process involves source code translation, reverse engineering, program structure improvement, program modularisation and data re-engineering

Source code translation is the automatic conversion of of program in one language to another

Key points

Reverse engineering is the process of deriving the system design and specification from its source code

Program structure improvement replaces unstructured control constructs with while loops and simple conditionals

Program modularisation involves reorganisation to group related items

Data re-engineering may be necessary because of inconsistent data management

