MANCHESTER 1824
The University of Manchester

Evolutionary Design: System Scale

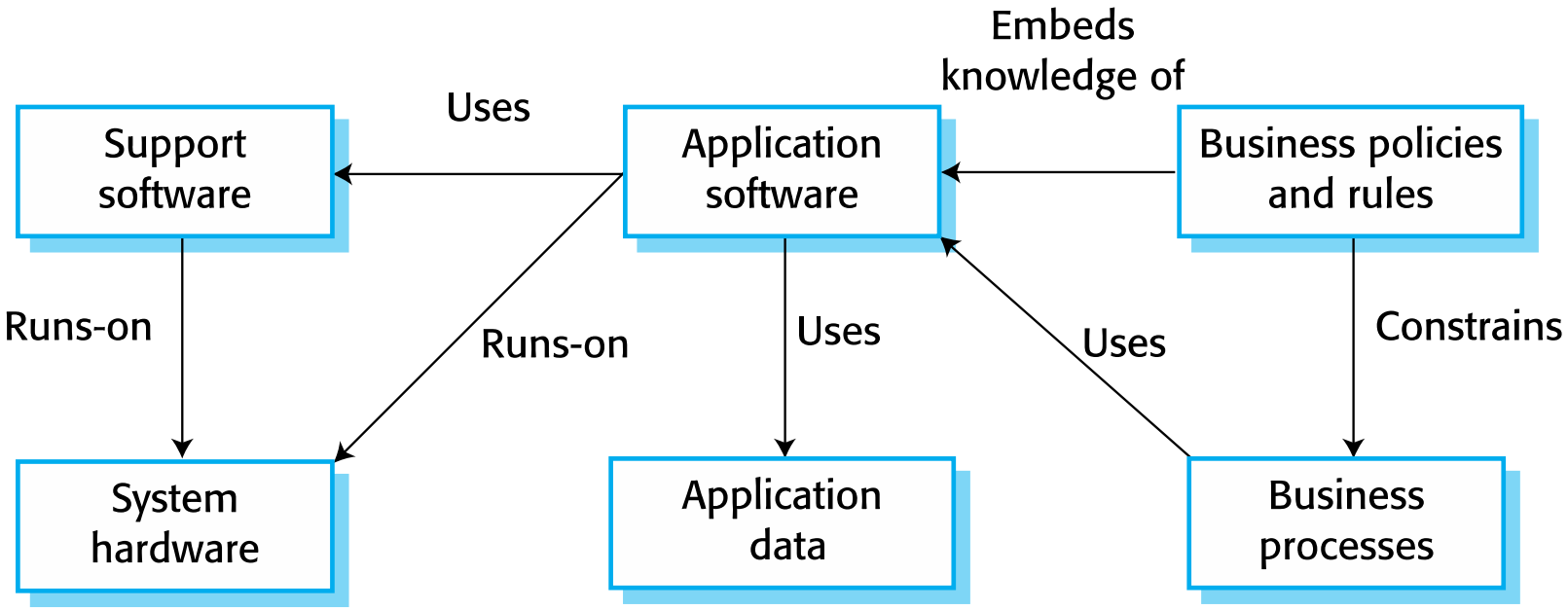
Andy Carpenter

School of Computer Science

(Andy.Carpenter@manchester.ac.uk)

Elements these slides come from Sommmerville, author of “Software Engineering", and are copyright Sommerville

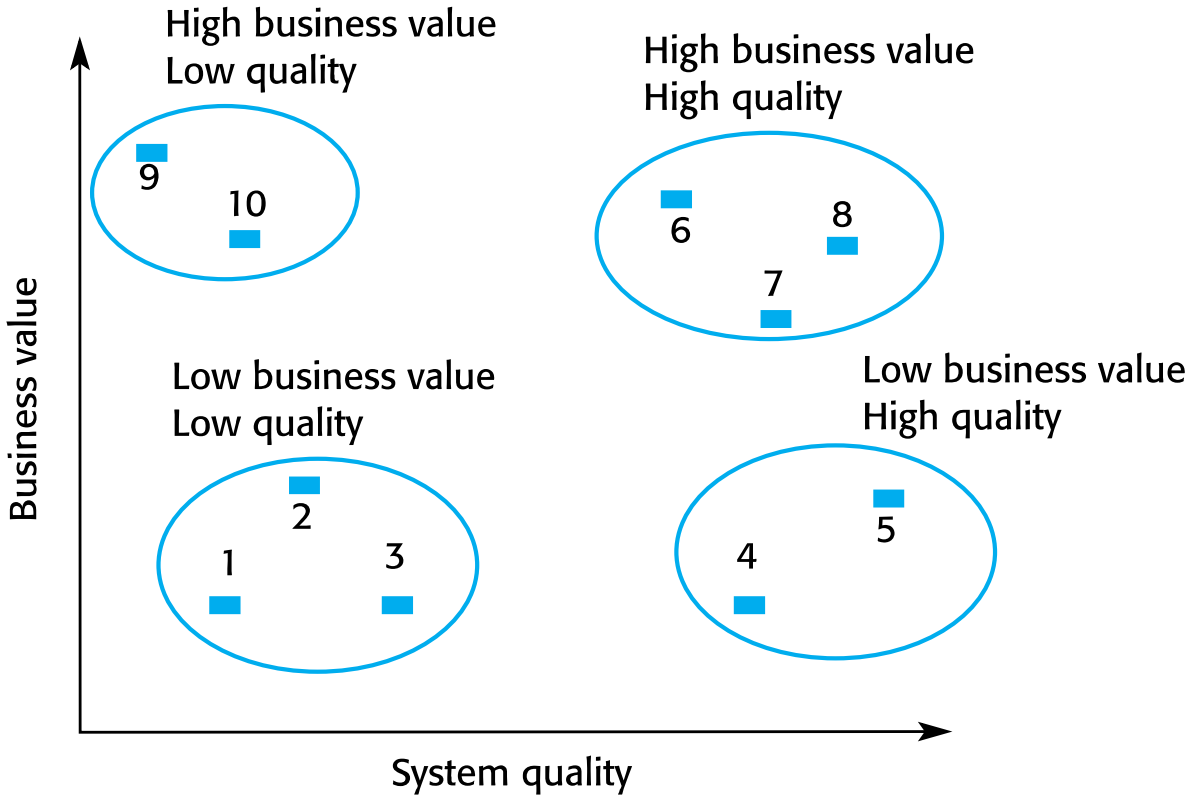
Elements of Legacy System



Legacy system management

- Organisations 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.

Legacy System Assessment



Business Value Assessment

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

Business Value Factors

- The use of the system
  - If systems are only used occasionally or by a small number of people, they may have a low business value.
- The business processes that are supported
  - A system may have a low business value if it forces the use of inefficient business processes.
- System dependability
  - If a system is not dependable and the problems directly affect business customers, the system has a low business value.
- The system outputs
  - If the business depends on system outputs, then the system has a high business value.

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?

MANCHESTER 1824
The University of Manchester

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 organisation 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 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 a more modern system.
Performance	Is the performance of the system adequate? Do performance problems have a significant effect on system users?

MANCHESTER 1824
The University of Manchester

Environment Assessment

Factor	Questions
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, for example, 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 current?
Data	Is there an explicit data model for the system? To what extent is data duplicated across 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

Factor	Questions
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 people available who have experience with the system?

MANCHESTER 1824
The University of Manchester

System Measurement

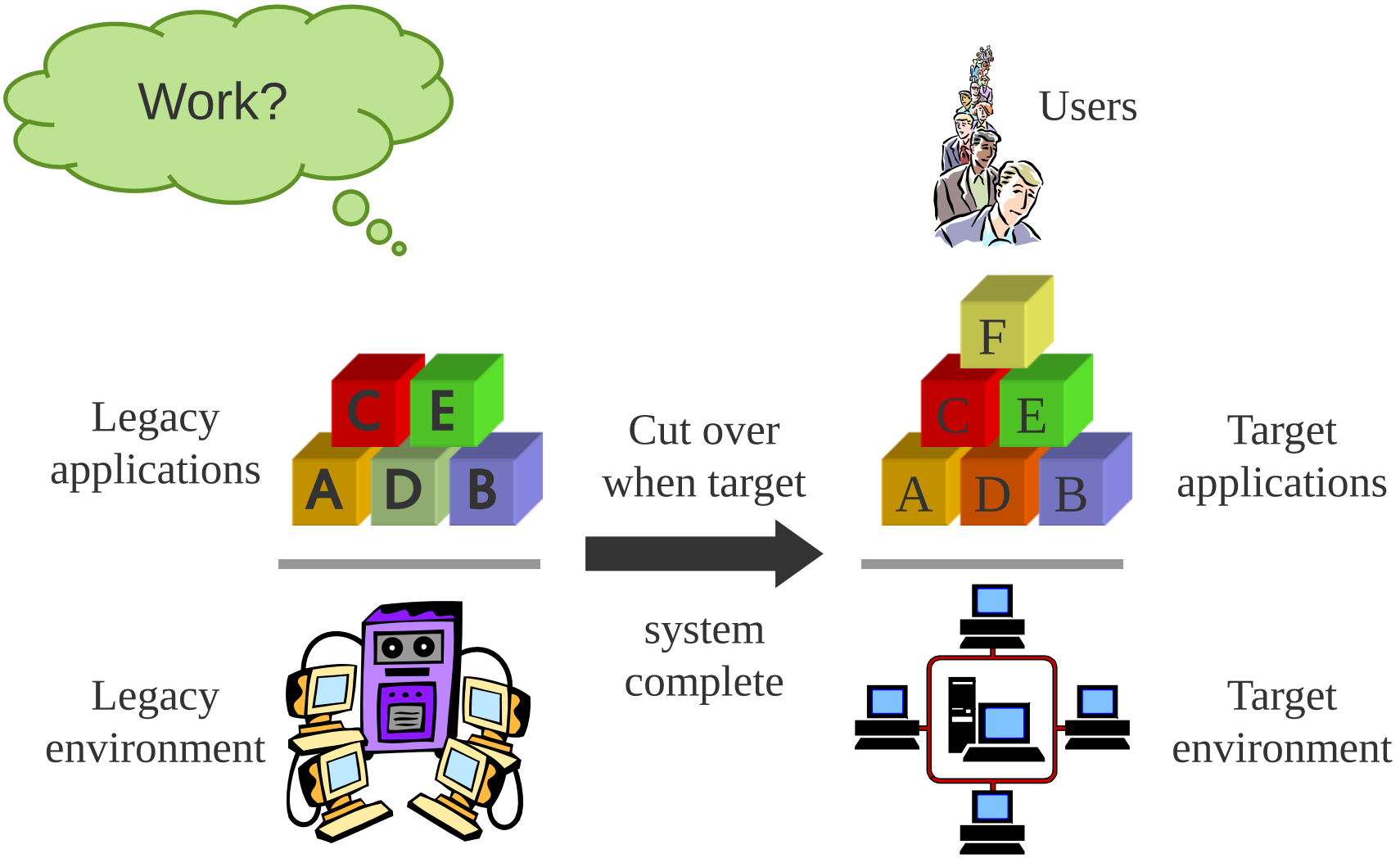
- You may collect quantitative data to make an assessment of the quality of the application system
  - The number of system change requests;
  - The number of different user interfaces used by the system; The more interfaces, the more likely it is that there will be inconsistencies and redundancies in these interfaces.
  - The volume of data used by the system. As the volume of data (number of files, size of database, etc.) processed by the system increases, so too do the inconsistencies and errors in that data.
  - Cleaning up old data is a very expensive and time-consuming process

Evolving Legacy Systems

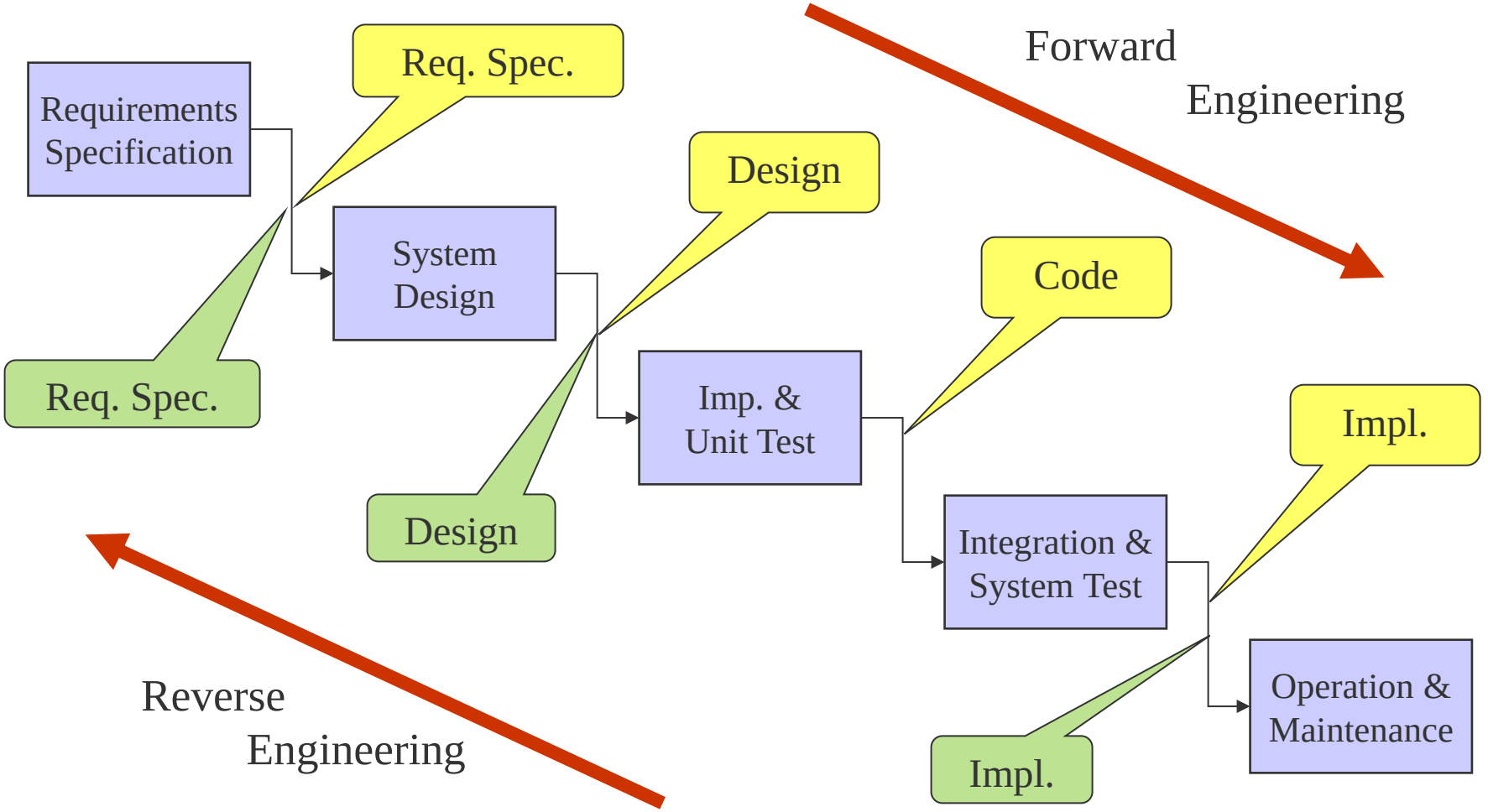
- Perform preventive maintenance to improve maintainability

- Rebuild the system from scratch (as if current system did not exist)

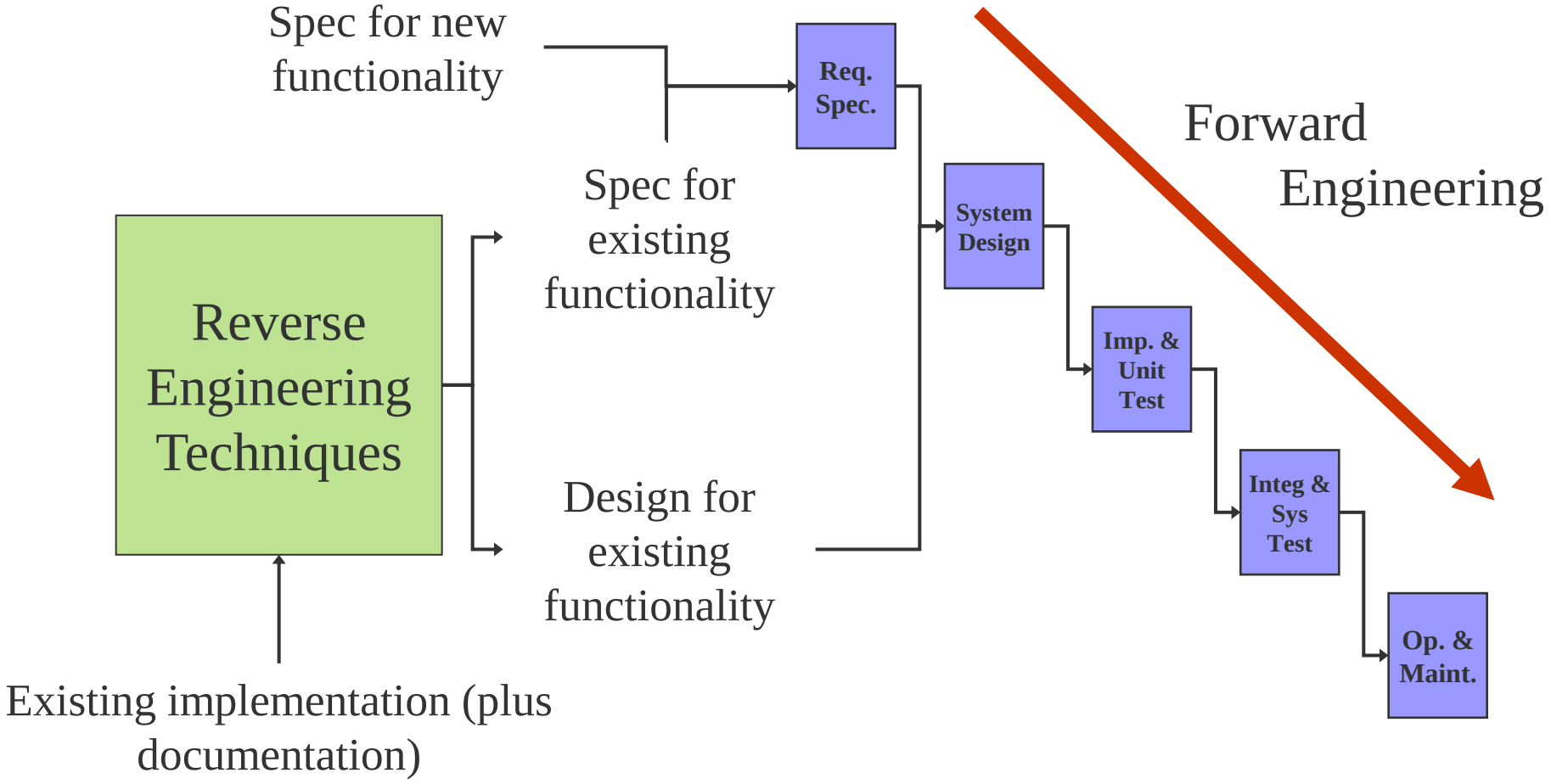
Early Attempts



Forward/Reverse Engineering



Re-engineering



Legacy Replacement Expensive

- No consistent programming style
- Use of obsolete programming languages with few people available with these language skills
- Inadequate system documentation
- System structure degradation
- Program optimizations may make them hard to understand
- Data errors, duplication and inconsistency

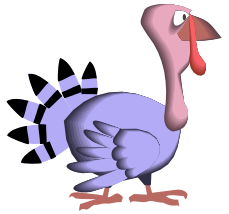
Re-engineering

- Claim:
  - Re-engineering is faster and less risky than a straight rebuild
  - Reverse engineering of specification and design is faster, and less error prone than rewriting them from scratch

- But, many re-engineering projects have failed disastrously
- Why?

Approaches to Migration

- Cold Turkey
  - Experience in general software development shows that all-or-nothing approaches rarely succeed
  - You learn about your biggest mistakes during cut over – this is too late!



- Chicken Little
  - Brodie and Stonebraker suggest that it is better to migrate the legacy system in a piecemeal fashion
  - In general software development, incremental approach is less risky

Evolving Legacy Systems

- Perform preventive maintenance to improve maintainability
- Modernise the system
- Rebuild the system from scratch (as if current system did not exist)

System Modernisation

- What aspects of a system might be modernised?

User Interface		System Interface	
Prog Lang	Business Logic	Architecture	
OS	DBMS	Comms	Web Server
Machine	File store	Network	

Cold Turkey

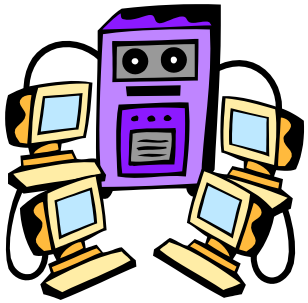
Users

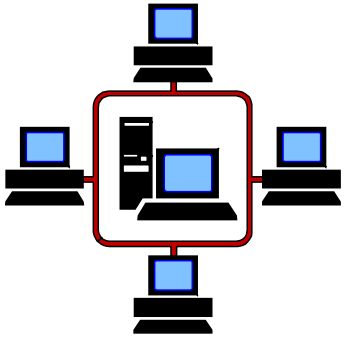
Legacy applications

Cut over when target system complete

Target applications

Legacy environment





Target environment