Systems Engineering

Eunseok Lee, Prof.
College of Software
Sungkyunkwan University

Objectives

- Understand the concept of a sociotechnical system and understand the difference between a technical computer-based system and a sociotechnical system;
- Understand the concept of emergent system properties, such as reliability, performance, safety, and security;
- Know about the conceptual design, procurement, development, and operation and evolution activities that are involved in the systems engineering process;
- Understand why software *dependability* and *security* should not be considered in isolation and how they are affected by systems issues, such as operator errors.

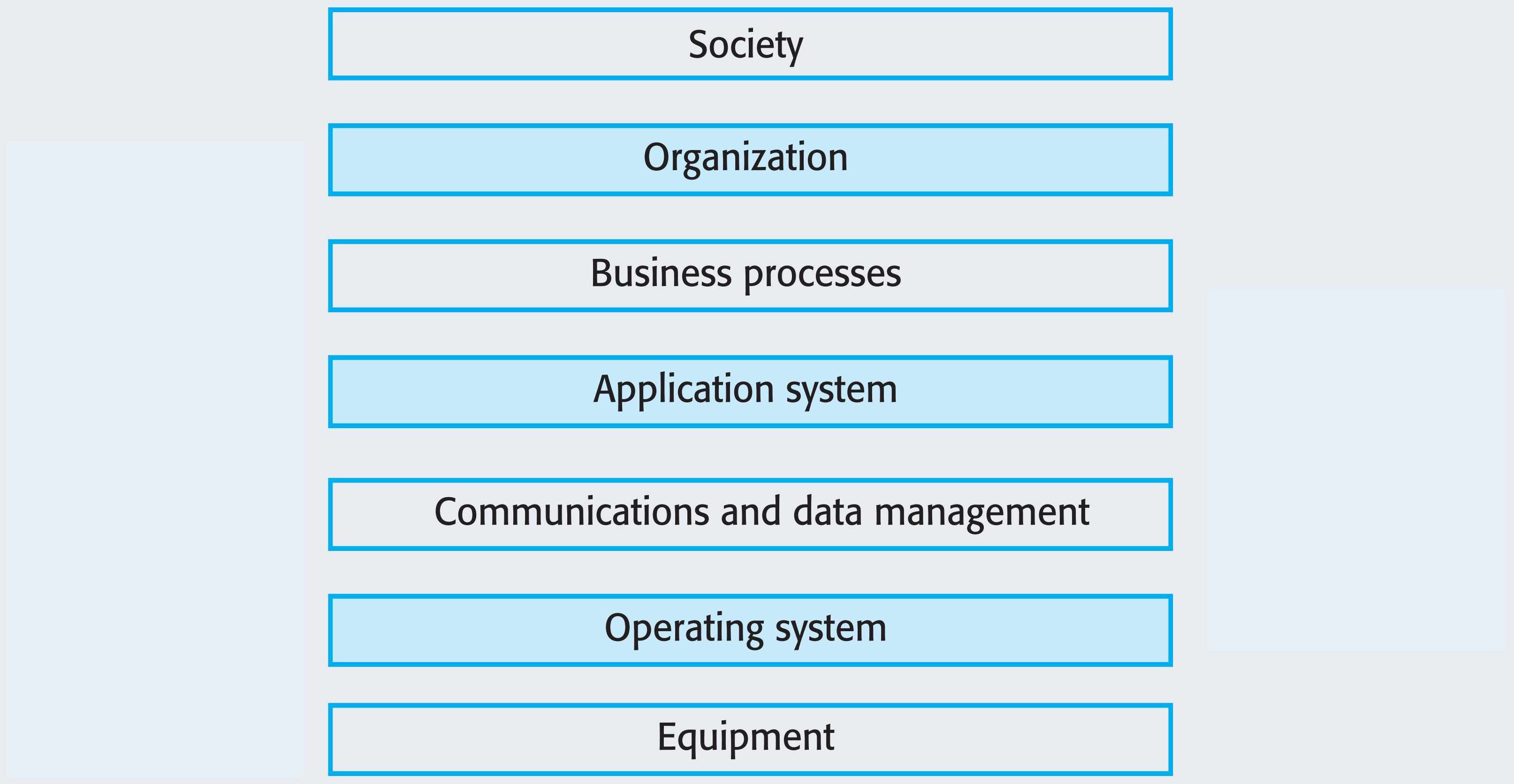
Topics covered

1. Complex systems
2. Systems engineering
3. Conceptual design
4. Systems procurement
5. System development
6. System operation and evolution

Systems

- A system is a purposeful collection of inter-related components working together to achieve some common objective.
- Software engineering is not an isolated activity but is part of a broader systems engineering process.
- Software systems are therefore not isolated systems but are essential components of broader systems that have a *human*, *social* or *organizational* purpose.
- Example
  - Wilderness weather system is part of broader weather recording and forecasting systems
  - These include hardware and software, forecasting processes, system users, the organizations that depend on weather forecasts, etc.

The Sociotechnical Systems Stack



Society

Organization

Business processes

Application system

Communications and data management

Operating system

Equipment

Layers in the STS stack

- **Equipment**
  - Hardware devices, some of which may be computers. Most devices will include an embedded system of some kind.
- **Operating system**
  - Provides a set of common facilities for higher levels in the system.
- **Communications and data management**
  - Middleware that provides access to remote systems and databases.
- **Application systems**
  - Specific functionality to meet some organization requirements.
- **Business processes**
  - A set of processes involving people and computer systems that support the activities of the business.

Layers in the STS stack

- **Organizations**
  - Higher level strategic business activities that affect the operation of the system such as business rules, norms that should be followed.
- **Society**
  - Laws, regulation and culture that affect the operation of the system.
- **There are interactions and dependencies between the layers in a system and changes at one level ripple through the other levels**
  - Example: Change in regulations (society) leads to changes in business processes and application software.

1. Complex Systems

- A system is a purposeful collection of inter-related components working together to achieve some common objective.
- A system may include *software*, mechanical, electrical and electronic *hardware* and be operated by *people*.
- System components are dependent on other system components.
- The properties and behaviour of system components are *inextricably inter-mingled*. This leads to complexity.

System Categories

- **Technical computer-based systems**
  - Systems that include hardware and software but where the operators and operational processes are not normally considered to be part of the system.
  - Example: A word processor used to write a book.
- **Socio-technical systems**
  - Systems that include technical systems but also **operational processes** and **people** who use and interact with the technical system. Socio-technical systems are governed by organizational policies and rules.
  - Example: A publishing system to produce a book.

Organizational Affects

- **Process changes**
  - Systems may require changes to business processes so training may be required. Significant changes may be resisted by users.
- **Job changes**
  - Systems may de-skill users or cause changes to the way they work. The status of individuals in an organization may be affected by the introduction of a new system.
- **Organizational changes**
  - Systems may change the political power structure in an organization. If an organization depends on a system then those that control the system have more power.

Socio-technical System Characteristics

- **Emergent properties**
  - Properties of the system of a whole that depend on the system components and their relationships.
- **Non-deterministic**
  - They do not always produce the same output when presented with the same input because the systems' behaviour is partially dependent on human operators.
- **Complex relationships with organizational objectives**
  - The extent to which the system supports organizational objectives does not just depend on the system itself.

1.1 Emergent Properties

- Properties of the system **as a whole** rather than properties that can be derived from the properties of components of a system
- Emergent properties are **a consequence of the relationships** between system components
- They can therefore only be assessed and measured **once the components have been integrated** into a system

Examples of Emergent Properties

Property	Description
Volume	The volume of a system (the total space occupied) varies depending on how the component assemblies are arranged and connected.
Reliability	System reliability depends on component reliability but unexpected interactions can cause new types of failures and therefore affect the reliability of the system.
Security	The security of the system (its ability to resist attack) is a complex property that cannot be easily measured. Attacks may be devised that were not anticipated by the system designers and so may defeat built-in safeguards.
Repairability	This property reflects how easy it is to fix a problem with the system once it has been discovered. It depends on being able to diagnose the problem, access the components that are faulty, and modify or replace these components.
Usability	This property reflects how easy it is to use the system. It depends on the technical system components, its operators, and its operating environment.

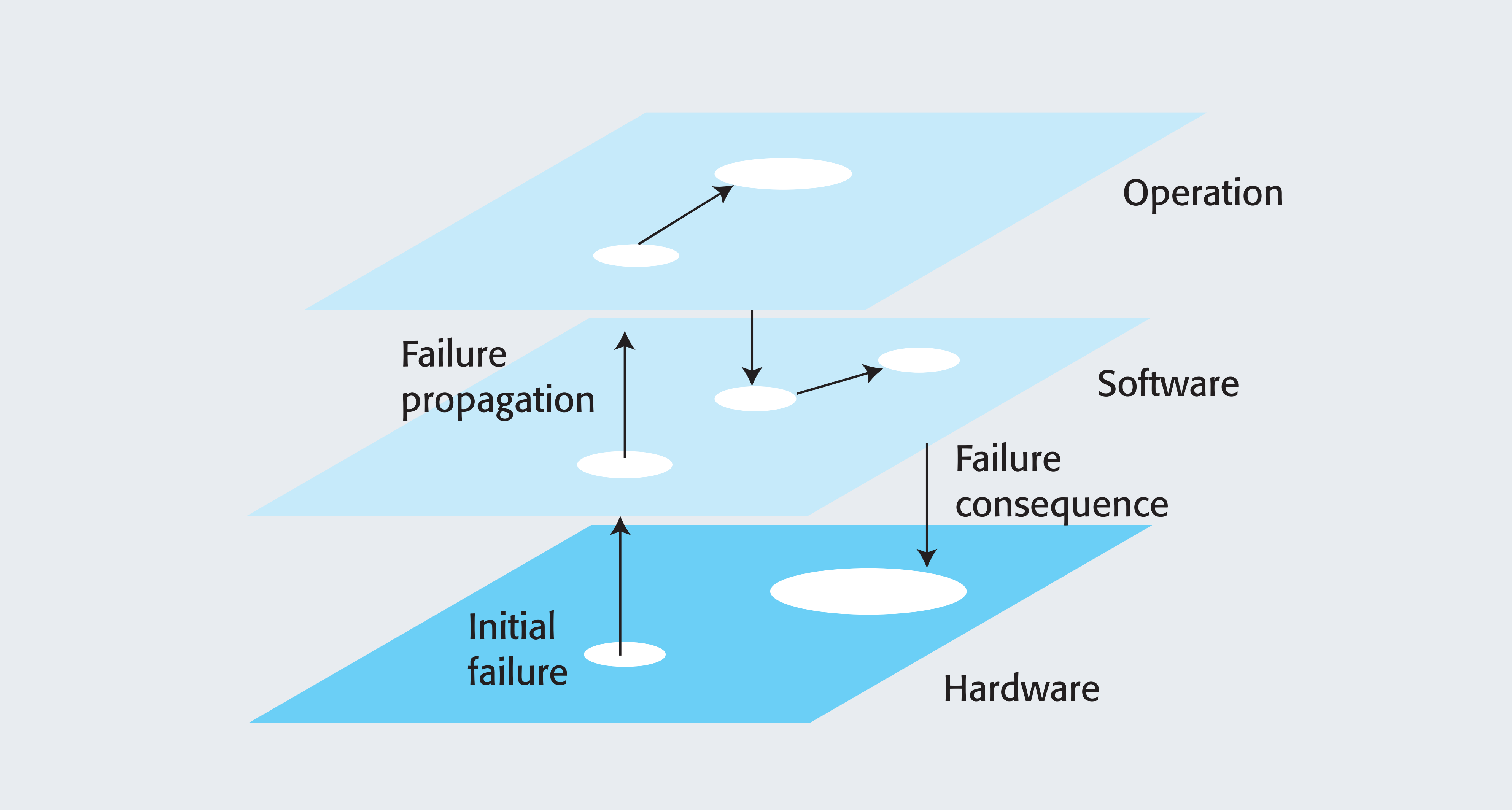
Reliability as an Emergent Property

- Because of component inter-dependencies, faults can **be propagated** through the system.
- System failures often occur because of **unforeseen inter-relationships** between components.
- It is practically **impossible to anticipate** all possible component relationships.
- Software reliability measures may give a false picture of the overall system reliability.

Influences on Reliability

- **Hardware reliability**
  - What is the probability of a hardware component failing and how long does it take to repair that component?
- **Software reliability**
  - How likely is it that a software component will produce an incorrect output. Software failure is usually distinct from hardware failure in that software does not wear out. (not **bathtub curve**)
- **Operator reliability**
  - How likely is it that the operator of a system will make an error?
- **Failures are not independent and they propagate from one level to another.**

Failure Propagation



1.2 Non-determinism

- **A deterministic system is one where a given sequence of inputs will always produce the same sequence of outputs.**
- **Software systems are deterministic; systems that include humans are non-deterministic**
  - A socio-technical system will not always produce the same sequence of outputs from the same input sequence
  - Human elements
    - People do not always behave in the same way
  - System changes
    - System behavior is unpredictable because of frequent changes to hardware, software and data.

1.3 Success Criteria

- Complex systems are developed to address 'wicked problems' – problems where there cannot be a complete specification.
- Different stakeholders see the problem in different ways and each has a partial understanding of the issues affecting the system.
- Consequently, different stakeholders have their own views about whether or not a system is 'successful'
  - Success is a judgment and cannot be objectively measured.
  - Success is judged using the effectiveness of the system when deployed rather than judged against the original reasons for procurement.

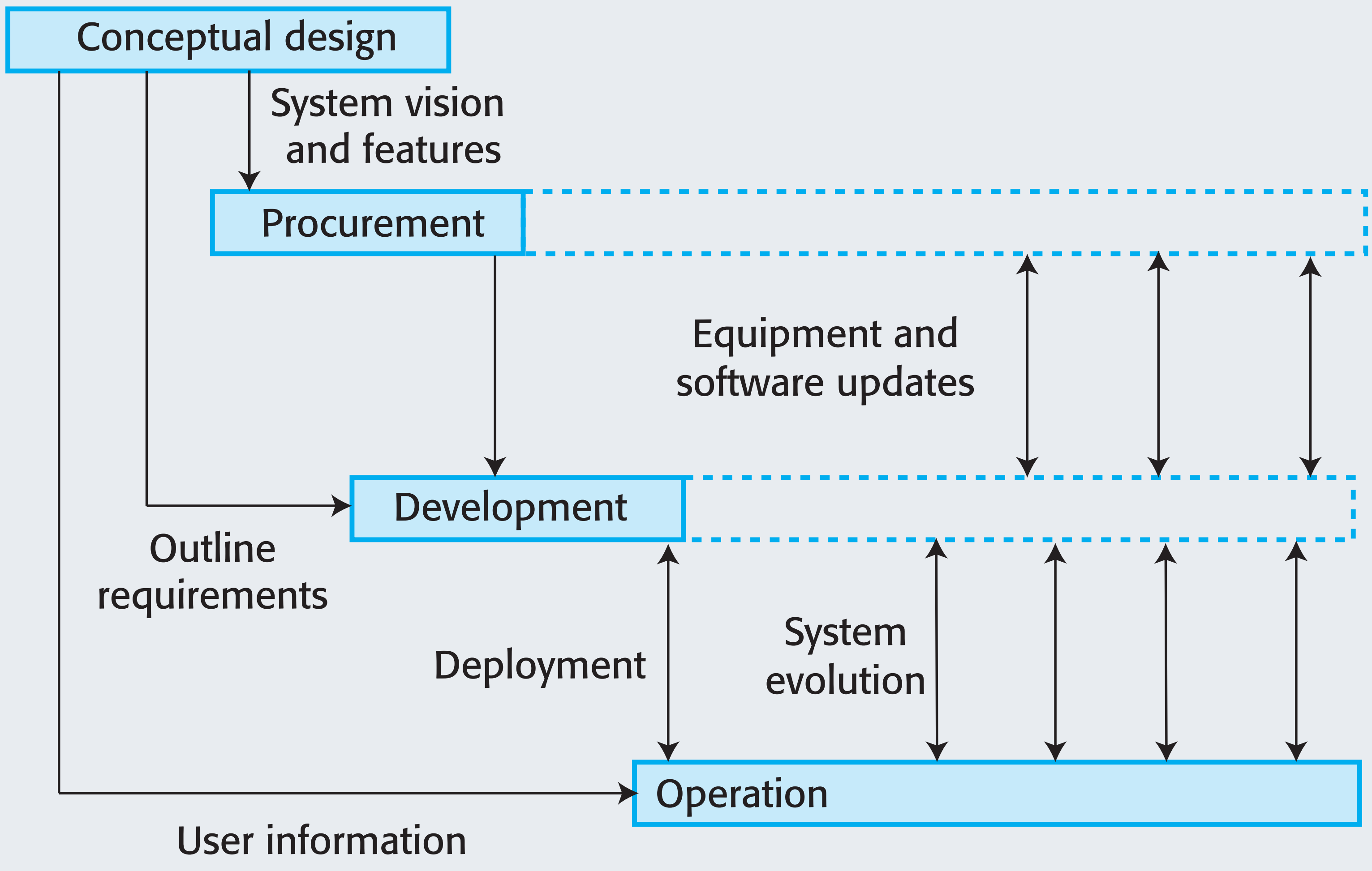
Conflicting Views of Success

- **MHC-PMS designed to support multiple, conflicting goals**
  - Improve quality of care.
  - Provide better information and care costs and so increase revenue.
- **Fundamental conflict**
  - To satisfy *reporting goal*, doctors and nurses had to provide additional information over and above that required for clinical purposes.
  - They had less time to interact with patients, so quality of care reduced. System was not a success.
- **However, managers had better reports**
  - System was a success from a managerial perspective.

2. Systems Engineering

- **Conceptual design, procuring, specifying, designing, implementing, validating, deploying** and **maintaining** socio-technical systems.
- Concerned with the **services** provided by the system, **constraints** on its construction and operation and the **ways** in which it is used to fulfill its purpose or purposes.

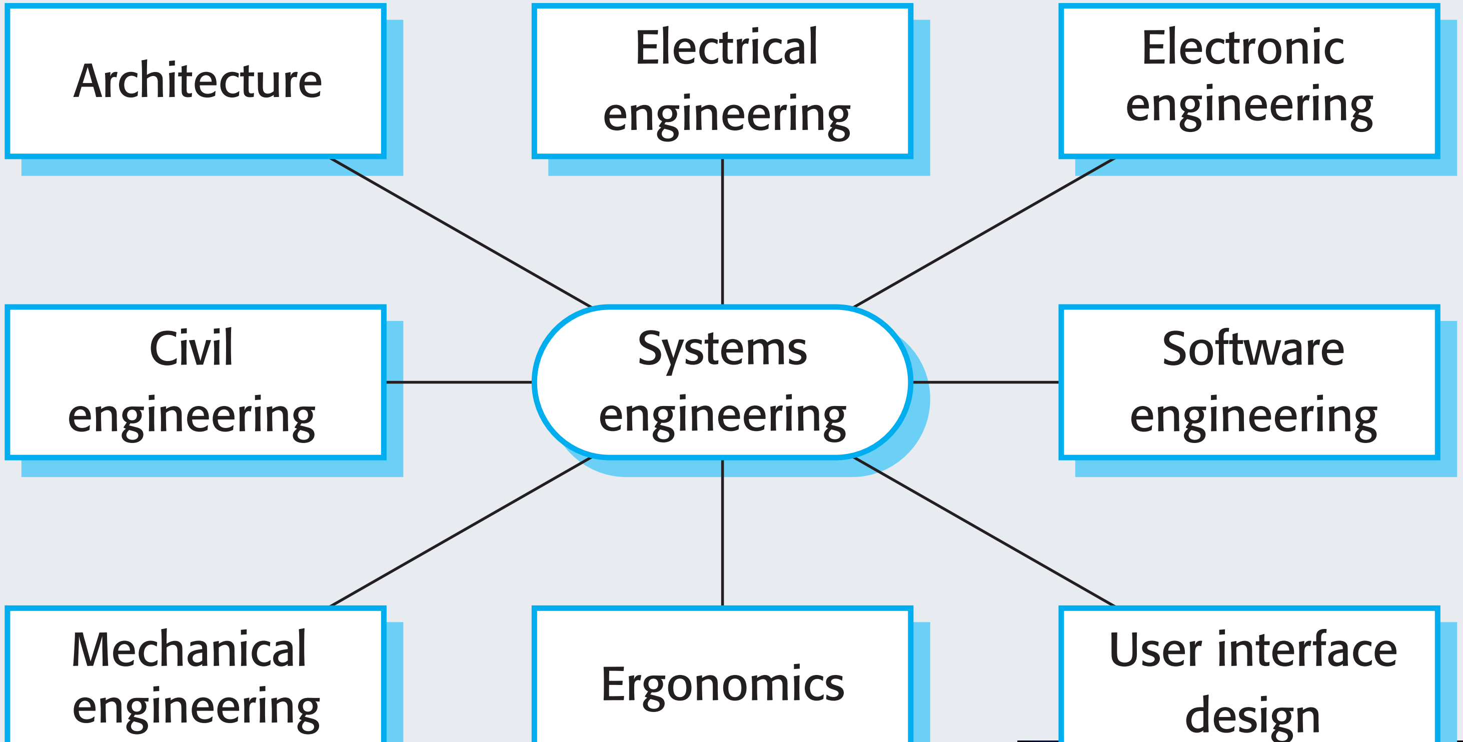
Stages of Systems Engineering



Systems Engineering Stages

- **Conceptual design**
  - Sets out the purpose of the system, why it is needed and the high-level features that users might expect to see in the system
- **Procurement (acquisition)**
  - High-level system requirements are defined, decisions are made on how functionality is distributed and the system components are purchased.
- **Development**
  - The system is developed – requirements are defined in detail, the system is implemented and tested and operational processes are defined and the training courses for system users are designed.
- **Operation**
  - The system is deployed and put into use. Changes are made as new requirements emerge. Eventually, the system is decommissioned.

Professional disciplines involved in systems engineering



Inter-Disciplinary Working

- **Communication difficulties**
  - Different disciplines use the same terminology to mean different things. This can lead to misunderstandings about what will be implemented.
- **Differing assumptions**
  - Each discipline makes assumptions about what can and can't be done by other disciplines.
- **Professional boundaries**
  - Each discipline tries to protect their professional boundaries and expertise and this affects their judgments on the system.

Key points

- Socio-technical systems include computer *hardware*, *software* and *people* and are designed to meet some *business goal*.
- Human and organizational factors, such as the *organizational structure*, have a significant effect on the operation of socio-technical systems.
- Emergent properties are properties that are characteristic of the system as a whole and not its component parts.
- The fundamental stages of systems engineering are *conceptual design*, *procurement*, *development* and *operation*.

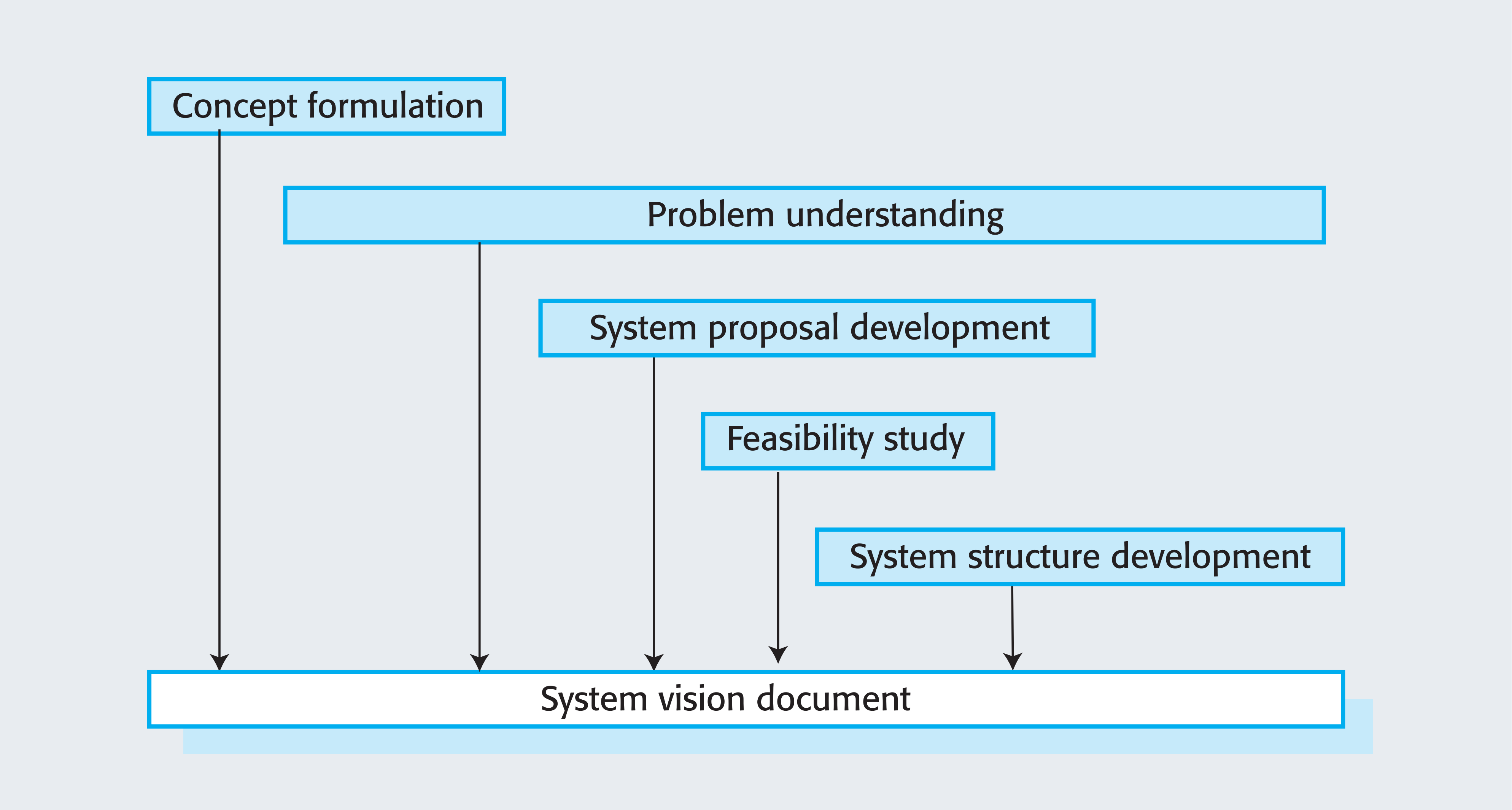
Systems Engineering

Part 2

3. Conceptual Design

- Investigate the feasibility of an idea and develop that idea to create an overall vision of a system.
- Conceptual design precedes and overlaps with requirements engineering
  - May involve discussions with users and other stakeholders and the identification of critical requirements
- The aim of conceptual design is to create a high-level system description that communicates the system purpose to non-technical decision makers.

Conceptual Design Activities



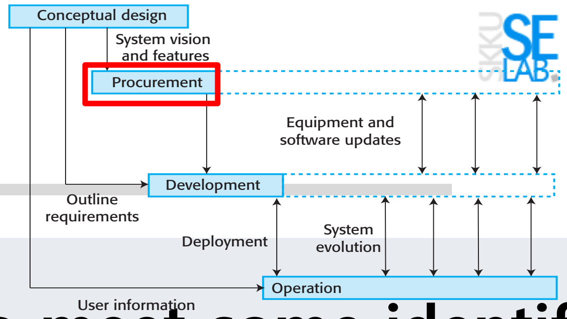
Conceptual Design

- **Concept formulation**
  - Refine an initial statement of needs and work out what type of system is most likely to meet the needs of system stakeholders
- **Problem understanding**
  - Discuss with stakeholders how they do their work, what is and isn’t important to them, what they like and don’t like about existing systems
- **System proposal development**
  - Set out ideas for possible systems (maybe more than one)

Conceptual Design

- **Feasibility study**
  - Look at comparable systems that have been developed elsewhere (if any) and assess whether or not the proposed system could be implemented using current hardware and software technologies
- **System structure development**
  - Develop an outline architecture for the system, identifying (where appropriate) other systems that may be reused
- **System vision document**
  - Document the results of the conceptual design in a readable, non-technical way. Should include a short summary and more detailed appendices.

4. System Procurement



- **Acquiring a system (or systems) to meet some identified organizational need.**
- **Before procurement, decisions are made on:**
  - Scope of the system
  - System budgets and timescales
  - High-level system requirements
- **Based on this information, decisions are made on whether to procure a system, the type of system and the potential system suppliers.**

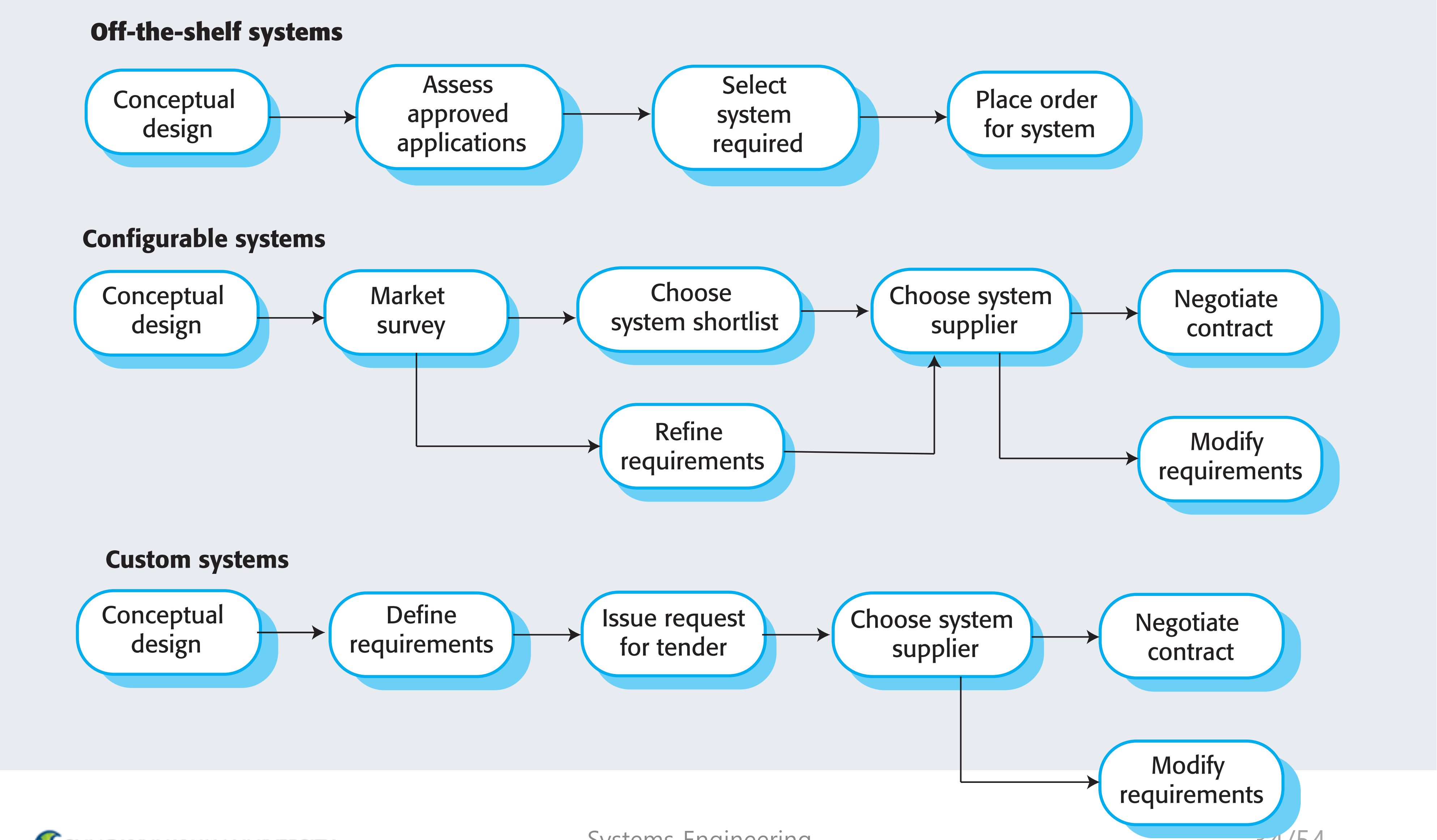
Decision Drivers

- The state of other organizational systems
- The need to comply with external regulations
- External competition
- Business re-organization
- Available budget

Procurement and Development

- **Some *system specification* and *architectural design* is usually necessary before procurement**
  - You need a specification to let a contract for system development
  - The specification may allow you to buy a commercial off-the-shelf (COTS) system. Almost always cheaper than developing a system from scratch
- **Large complex systems** usually consist of a ***mix of* off the shelf** and **specially designed components.** The procurement processes for these different types of component are usually different.

System Procurement Processes



Procurement Issues (1)

- Requirements may have to be modified to match the capabilities of off-the-shelf components.
- The requirements specification may be part of the contract for the development of the system.
- There is usually a contract negotiation period to agree changes after the contractor to build a system has been selected.

Procurement Issues (2)

- When a system is to be built specially, the specification of requirements is part of the contract for the system being acquired.
  - It is therefore a legal as well as a technical document.
  - The requirements document is critical and procurement processes of this type usually take a considerable amount of time.

- For public sector systems especially, there are detailed rules and regulations that affect the procurement of systems.
  - These force the development of detailed requirements and make agile development difficult

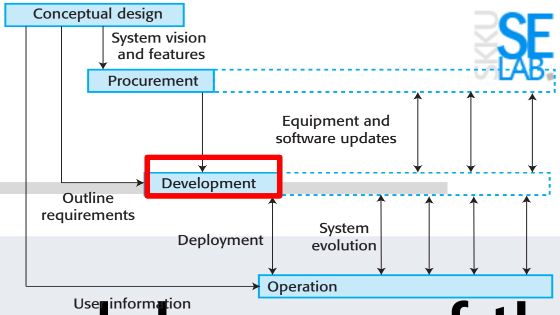
Procurement Issues (3)

- For application systems that require change or for custom systems there is usually a contract negotiation period where the customer and supplier negotiate the terms and conditions for the development of the system.
  - During this process, requirements changes may be agreed to reduce the overall costs and avoid some development problems.

Procurement Decisions

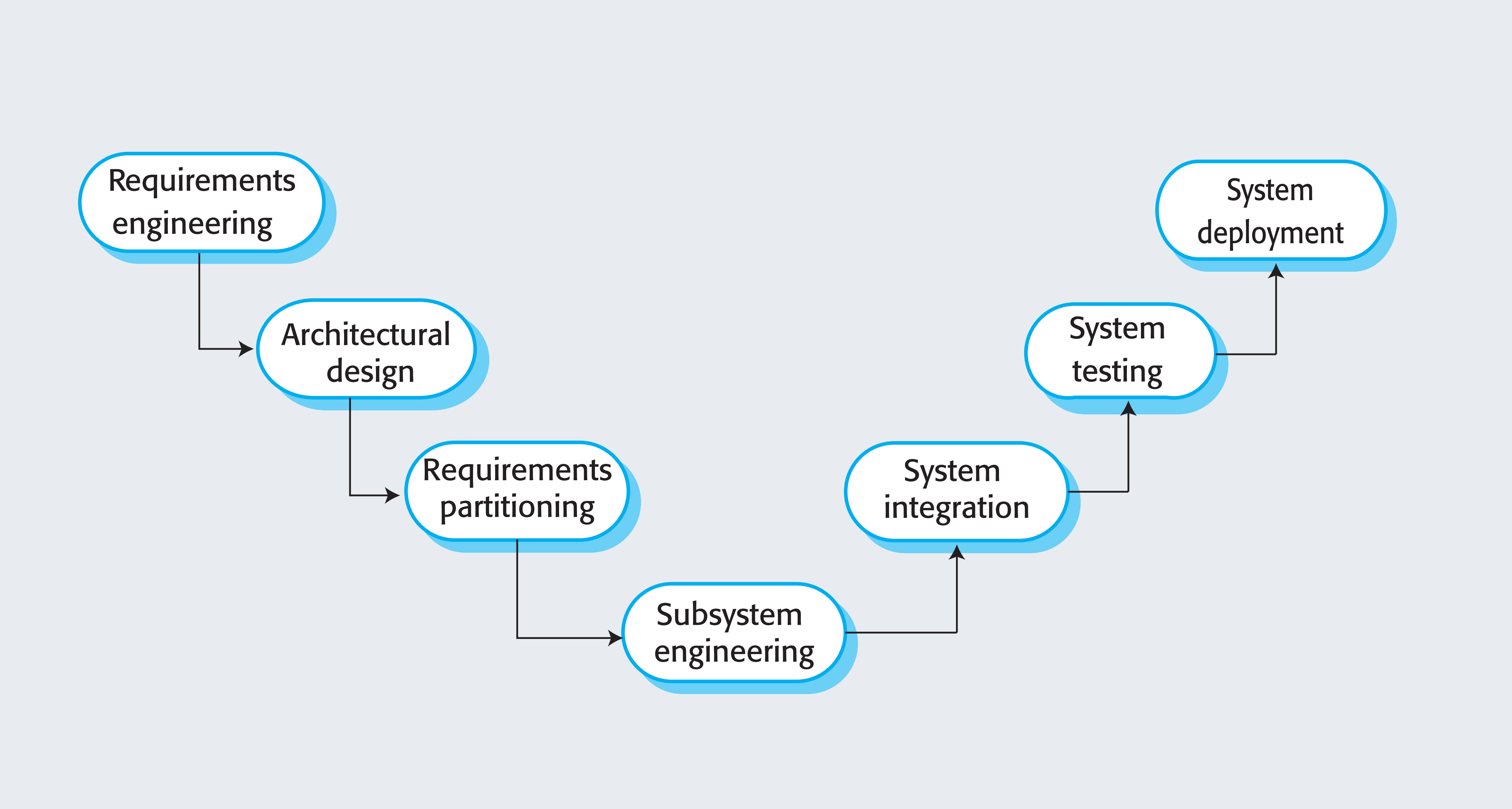
- **Decisions made at the procurement stage of the systems engineering process are critical for later stages in that process.**
  - Poor procurement decisions often lead to problems such as late delivery of a system and the development of systems that are unsuited to their operational environment.
  - If the wrong system or the wrong supplier is chosen then the technical processes of system and software engineering become more complex.

5. System Development



- **Usually follows a plan-driven approach because of the need for *parallel development* of different parts of the system**
  - Little scope for iteration between phases because hardware changes are very expensive. Software may have to compensate for hardware problems.
- **Inevitably involves engineers from different disciplines who must work together**
  - Much scope for misunderstanding here.
  - As explained, different disciplines use a different vocabulary and much negotiation is required. Engineers may have personal agendas to fulfil.

The System Development Process



The System Development Process (1)

- **Requirements engineering**
  - The process of refining, analysing and documenting the high-level and business requirements identified in the conceptual design
- **Architectural design**
  - Establishing the overall architecture of the system, identifying components and their relationships
- **Requirements partitioning**
  - Deciding which subsystems (identified in the system architecture) are responsible for implementing the system requirements

The system Development Process (2)

- **Subsystem engineering**
  - Developing the software components of the system, configuring off-the-shelf hardware and software, defining the operational processes for the system and re-designing business processes
- **System integration**
  - Putting together system elements to create a new system
- **System testing**
  - The whole system is tested to discover problems
- **System deployment**
  - the process of making the system available to its users, transferring data from existing systems and establishing communications with other systems in the environment

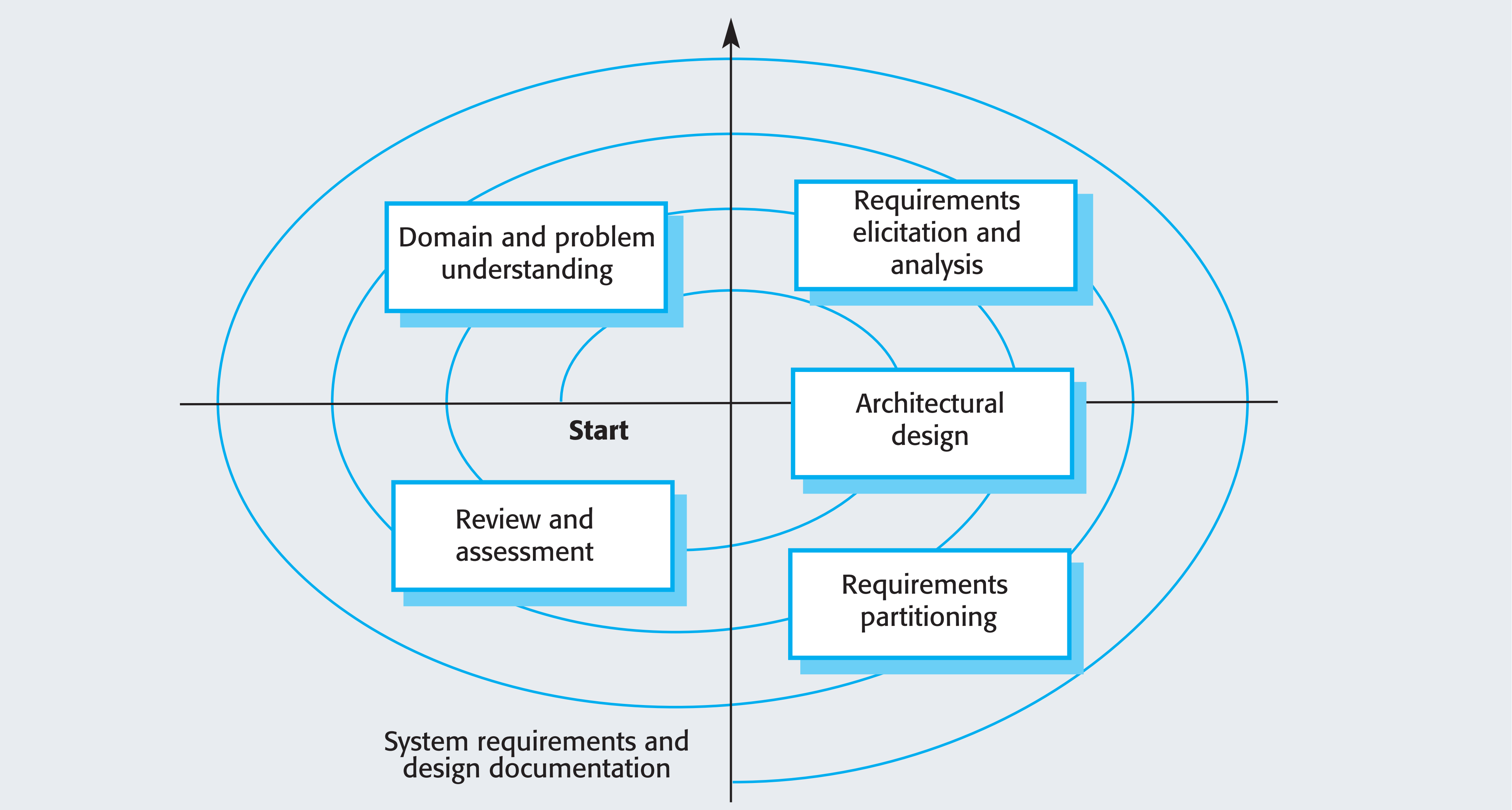
(1)System Requirements Definition

- **Three types of requirement defined at this stage**
  - **Abstract functional requirements**. System functions are defined in an abstract way;
  - **System properties**. Non-functional requirements for the system in general are defined;
  - **Undesirable characteristics**. Unacceptable system behaviour is specified.
- **Should also define overall organizational objectives for the system.**

(2)The Architectural Design Process

- **Partition requirements**
  - Organize requirements into related groups.
- **Identify sub-systems**
  - Identify a set of sub-systems which collectively can meet the system requirements.
- **Assign requirements to sub-systems**
  - Causes particular problems when COTS are integrated.
- **Specify sub-system functionality.**
- **Define sub-system interfaces**
  - Critical activity for parallel sub-system development.

Requirements and Design Spiral


Domain and problem understanding
Requirements elicitation and analysis
Architectural design
Start
Review and assessment
Requirements partitioning
System requirements and design documentation

(3)Sub-system Engineering

- Typically **parallel projects** developing the hardware, software and communications.
- May involve some COTS (Commercial Off-the-Shelf) systems procurement.
- **Lack of communication** across implementation teams can cause problems.
- There may be a **bureaucratic** and **slow mechanism** for proposing system changes, which means that the development schedule may be extended because of the need for rework.

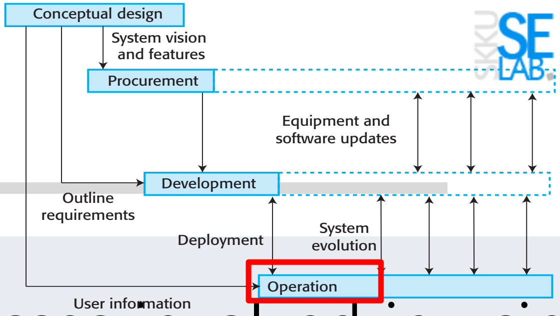
(4)System Integration

- The process of putting *hardware*, *software* and *people* together to make a system.
- Should ideally be tackled incrementally so that sub-systems are integrated one at a time.
- The system is tested as it is integrated.
- **Interface problems** between sub-systems are usually found at this stage.
- May be problems with uncoordinated deliveries of system components.

(5)System Delivery and Deployment

- **After completion, the system has to be installed in the customer's environment**
  - Environmental assumptions may be incorrect;
  - May be human resistance to the introduction of a new system;
  - System may have to coexist with alternative systems for some time;
  - May be physical installation problems (e.g. cabling problems);
  - Data cleanup may be required;
  - Operator training has to be identified.

6. System Operation



- *Operational processes* are the processes involved in using the system for its defined purpose.
- For new systems, these processes may have to be designed and tested and operators trained in the use of the system.
- Operational processes should be flexible to allow operators to cope with problems and periods of fluctuating workload.

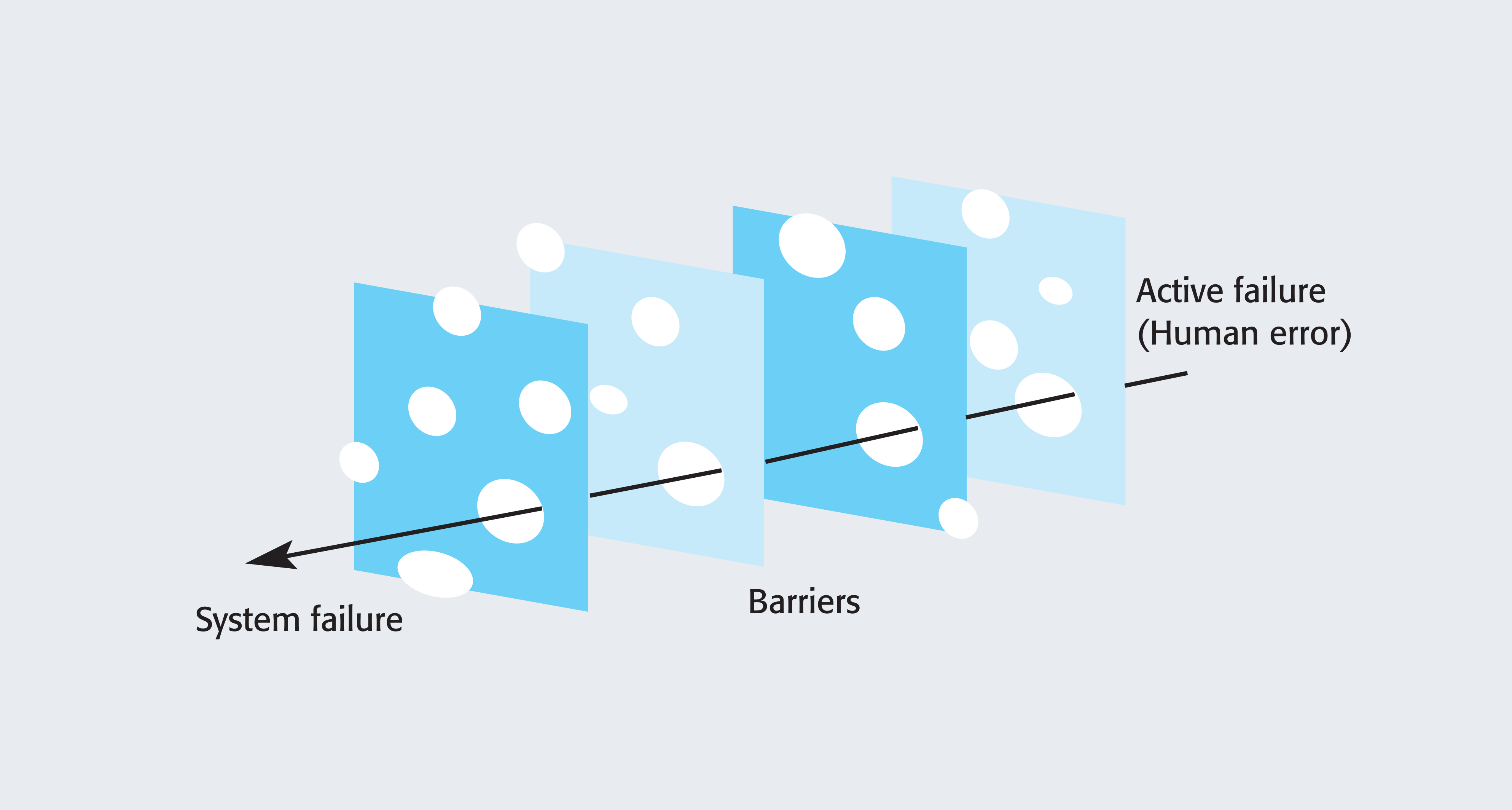
6.1 Human Error

- Human errors occur in operational processes that influence the overall dependability of the system.
- Viewing human errors:
  - The ***person approach*** makes errors the responsibility of the individual and places the blame for error on the operator concerned. Actions to reduce error include threats of punishment, better training, more stringent procedures, etc.
  - The ***systems approach*** assumes that people are fallible and will make mistakes. The system is designed to detect these mistakes before they lead to system failure. When a failure occurs, the aim is not to blame an individual but to understand why the **system defenses** did not trap the error.

System Defenses

- To improve security and dependability, designers should think about the checks for human error that should be included in a system.
- There should be **multiple** **(redundant)** **barriers** which should be different (diverse)
- No single barrier can be perfect.
  - There will be latent conditions in the system that may lead to failure.
- However, with multiple barriers, all have to fail for a system failure to occur.

Reason’s Swiss Cheese Model of System Failure



6.2 System Evolution

- Large systems have a long lifetime. **They must evolve to meet changing requirements.**
- **Evolution is inherently costly**
  - Changes must be analysed from a *technical* and *business* perspective;
  - Sub-systems interact so unanticipated problems can arise;
  - There is rarely a rationale for original design decisions;
  - System structure is corrupted as changes are made to it.
- **Existing systems which must be maintained are sometimes called legacy systems.**

Factors that affect System Lifetimes

Factor	Rationale
Investment cost	The costs of a systems engineering project may be tens or even hundreds of millions of dollars. These costs can only be justified if the system can deliver value to an organization for many years.
Loss of expertise	As businesses change and restructure to focus on their core activities, they often lose engineering expertise. This may mean that they lack the ability to specify the requirements for a new system.
Replacement cost	The cost of replacing a large system is very high. Replacing an existing system can only be justified if this leads to significant cost savings over the existing system.

Factors that affect System Lifetimes

Factor	Rationale
Return on investment	If a fixed budget is available for systems engineering, spending this on new systems in some other area of the business may lead to a higher return on investment than replacing an existing system.
Risks of change	Systems are an inherent part of business operations and the risks of replacing existing systems with new systems cannot be justified. The danger with a new system is that things can go wrong in the hardware, software and operational processes. The potential costs of these problems for the business may be so high that they cannot take the risk of system replacement.
System dependencies	Other systems may depend on a system and making changes to these other systems to accommodate a replacement system may be impractical.

Key Points

- Conceptual design creates a high-level system descriptions.
- System procurement covers all of the activities involved in deciding ***what* system to develop, buy, or outsource** and ***who* should supply that** system.
- System development includes **requirements specification**, **design**, **construction**, **integration** and **testing**.
- System operation is related with using the system for its defined purposes in a customer environment.
- System evolution is evolving the system to cope with change requests.
- Human errors are inevitable and systems should include **barriers** to detect these errors before they lead to system failure.