Process Activities

4.3. Process activities

- Software specification
- Software design and implementation
- Software validation
- Software evolution

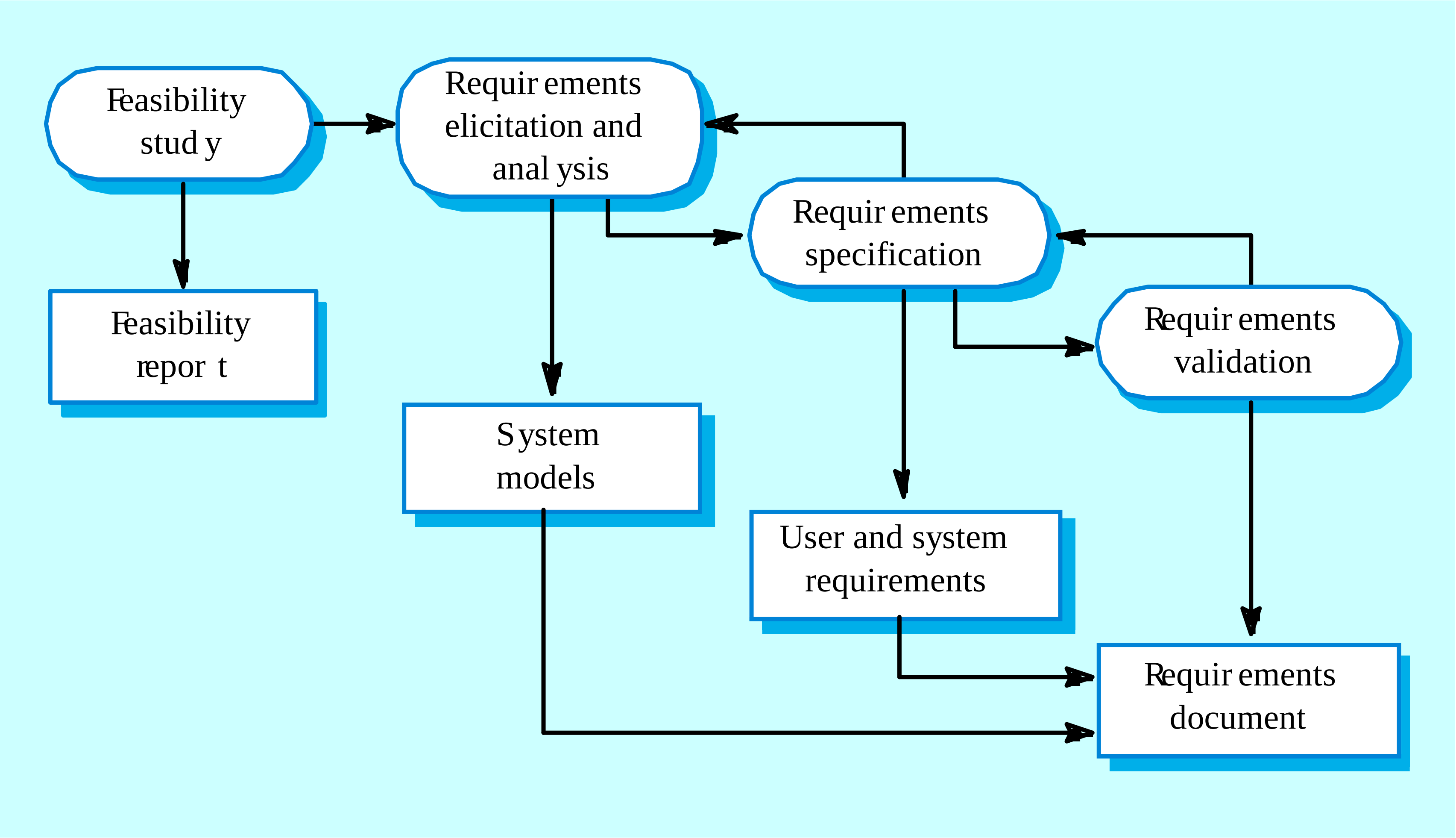
Software specification / RE

- The process of establishing what services are required and the constraints on the system’s operation and development.
- Requirements Engineering (RE) process:
  - Feasibility (technical and otherwise) study
  - Requirements elicitation and analysis
  - Requirements specification (documentation)
  - Requirements validation

(cont'd)

- Feasibility study;
  - User needs satisfied with current software and hardware,
  - Proposed cost effective
  - Cheap and quick
  - Result->detailed design
- Requirements elicitation and analysis;
  - Observation of existing systems
  - Develop one or more prototype and system models
- Requirements specification;
  - Translating the analysis activities into documentation
  - User requirement- abstract statement of the system requirement for the customer
  - System requirement- more detailed description of the functionality
- Requirements validation.
  - Check the requirements for realism consistency and completeness
  - Error in document is discovered(eliminate)

The requirements engineering process



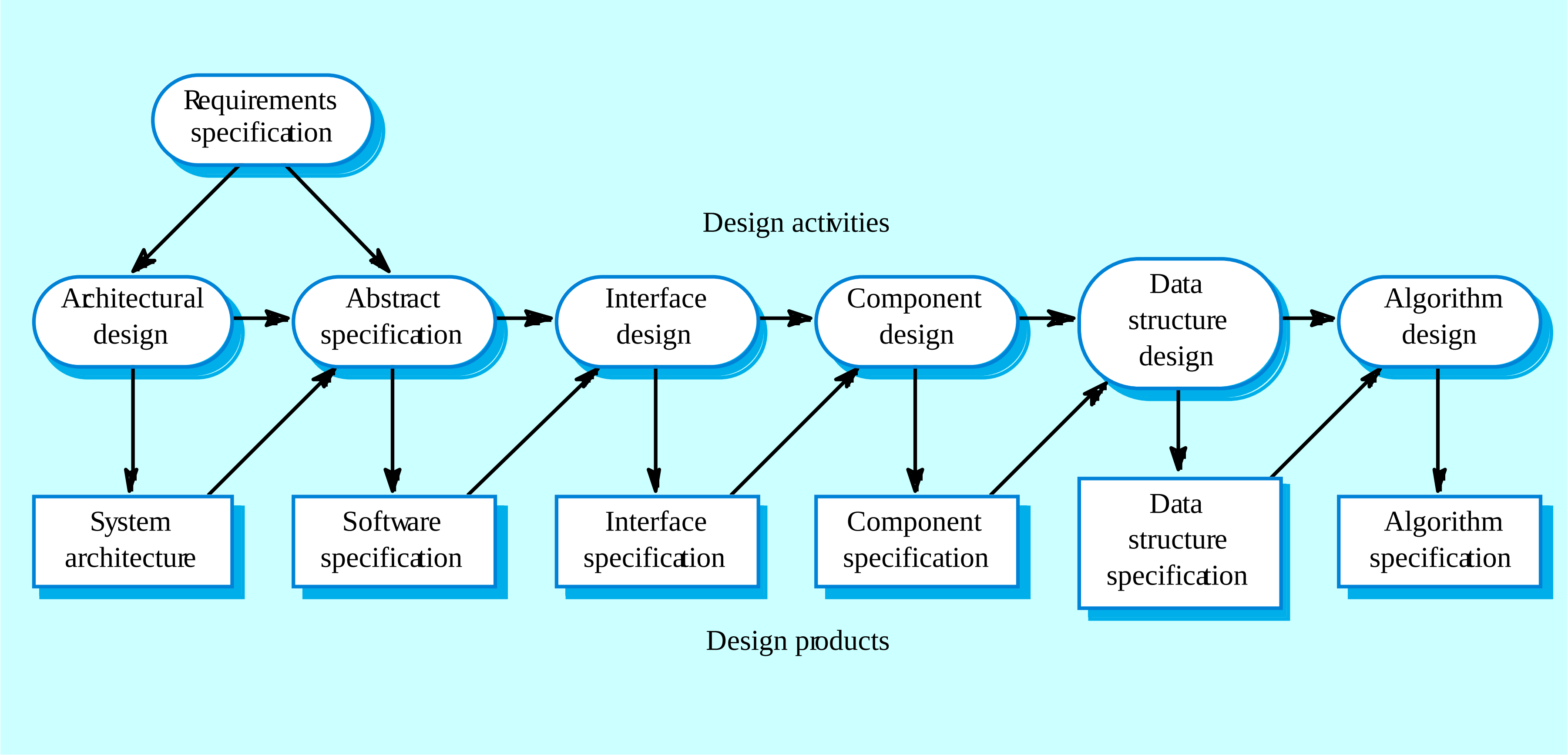
Software design and implementation

- The process of converting the system specification into an executable system.
- Software design
  - Design a software structure that realises the specification;
- Implementation
  - Translate this structure into an executable program;
- The activities of design and implementation are closely related and may be inter-leaved.

Design process activities

- Architectural design
  - Overall structure of the system, sub system, relationship
- Interface design
  - Interface between system component
  - component can be used without other components
  - the components can be designed and developed concurrently.
- Component design
  - where you take each system component and design
  - expected functionality to be implemented
- Database design
  - The work here depends on whether an existing database is to be reused or a new database is to be created.

The software design process



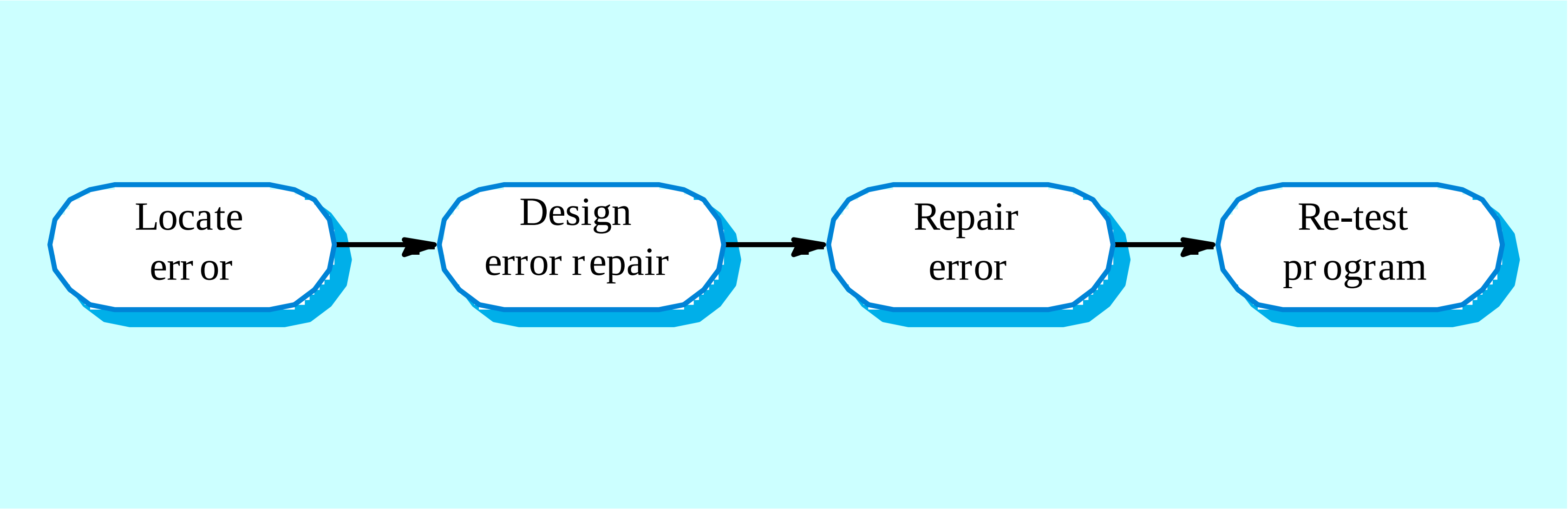
Structured methods

- Systematic approaches to developing a software design.
- The design is usually documented as a set of graphical models.
- Possible models
  - Object model;
  - Sequence model;
  - State transition model;
  - Structural model;
  - Data-flow model.

Programming and debugging

- Translating a design into a program and removing errors from that program.
- Programming is a personal activity - there is no generic programming process.
- Programmers carry out some program testing to discover faults in the program and remove these faults in the debugging process.

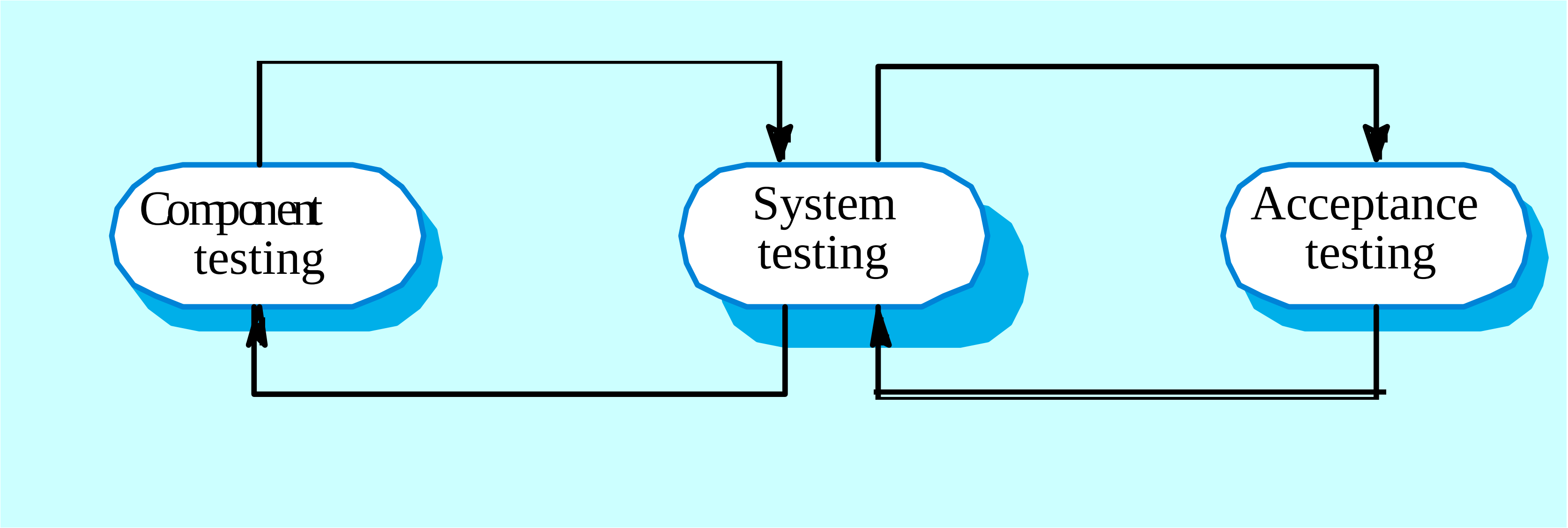
The debugging process



Software validation

- Verification and validation (V & V) is intended to show that a system conforms to its specification and meets the requirements of the system customer.
- Involves checking and review processes and system testing.
- System testing involves executing the system with test cases that are derived from the specification of the real data to be processed by the system.

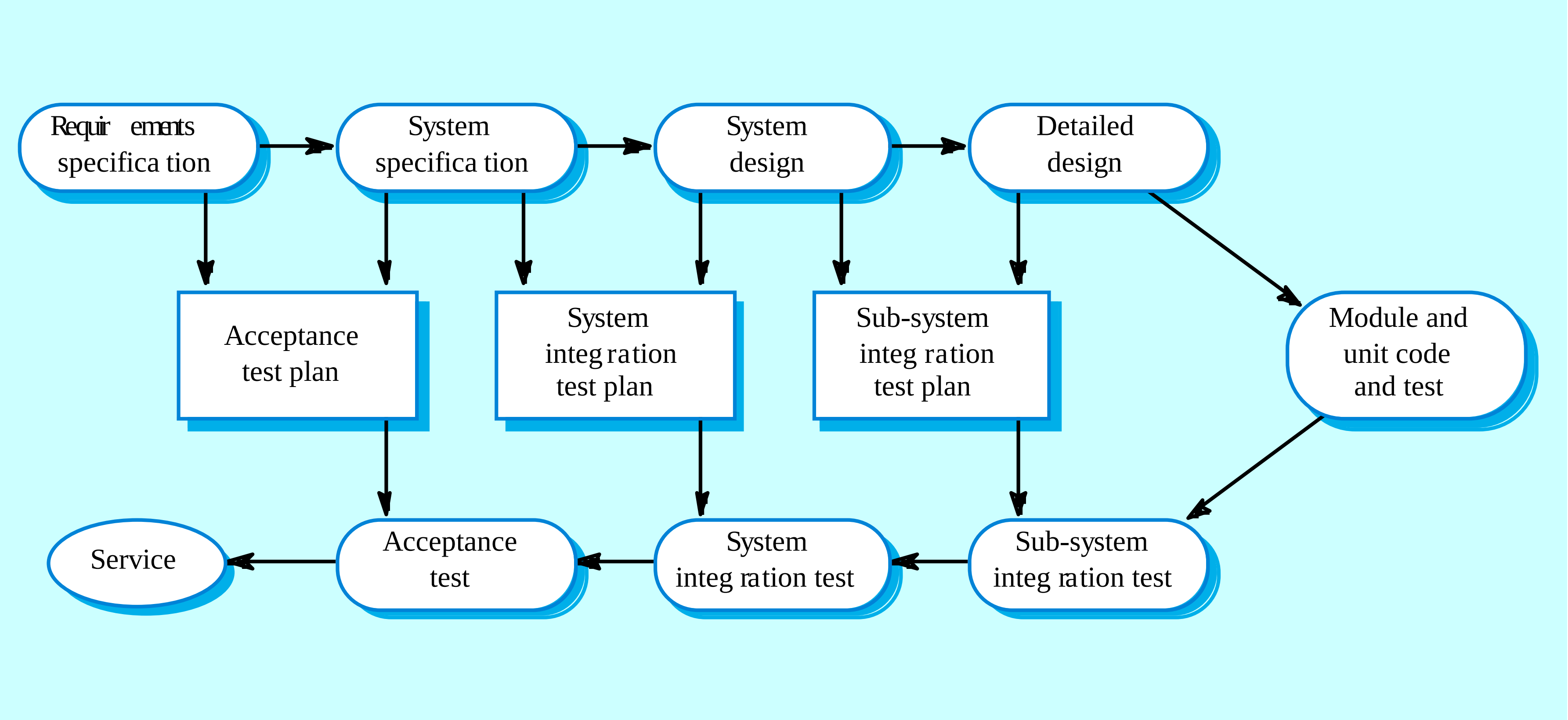
The testing process



Testing stages

- Component or unit testing
  - Individual components are tested independently;
  - Components may be functions or objects or coherent groupings of these entities.
- System testing
  - Testing of the system as a whole. Testing of emergent properties is particularly important.
- Acceptance testing (alpha testing)
  - Testing with customer data to check that the system meets the customer’s needs.

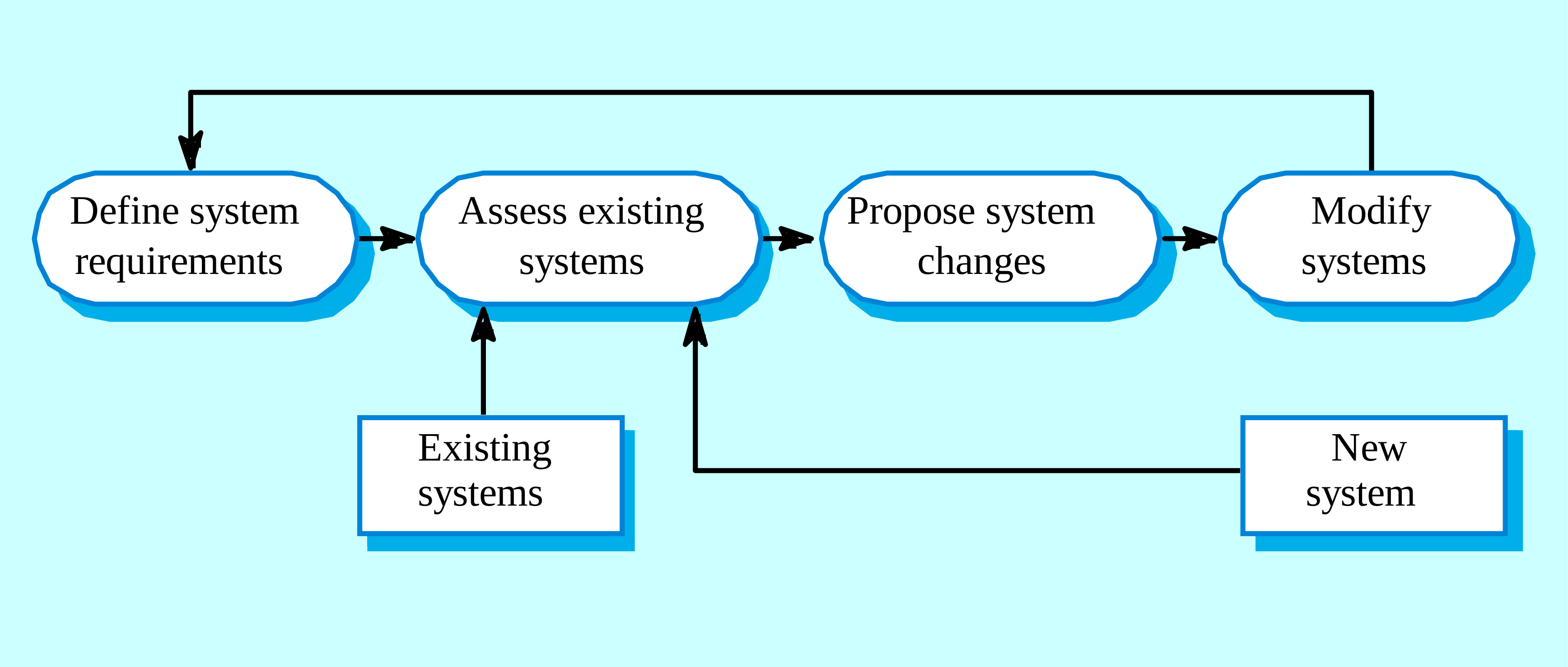
Testing phases


Requir ements specifica tion
System specifica tion
System design
Detailed design
Acceptance test plan
System integ ration test plan
Sub-system integ ration test plan
Module and unit code and test
Service
Acceptance test
System integ ration test
Sub-system integ ration test

Software evolution

- Software is inherently flexible and can change.
- As requirements change through changing business circumstances, the software that supports the business must also evolve and change.
- Although there has been a demarcation between development and evolution (maintenance) this is increasingly irrelevant as fewer and fewer systems are completely new.

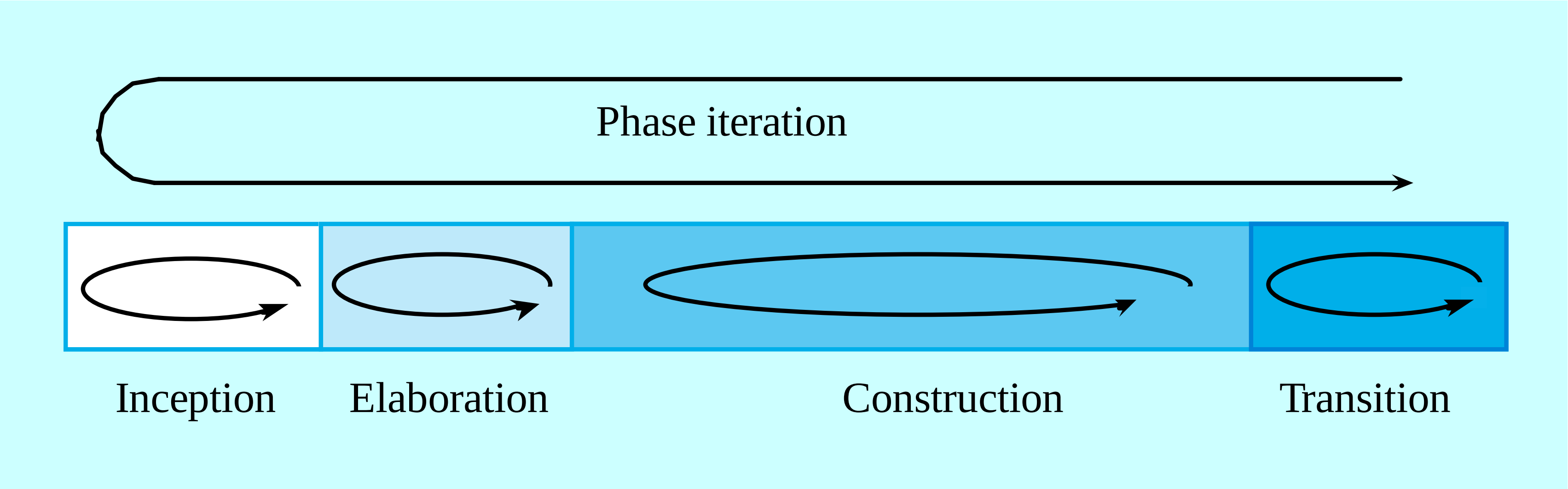
System evolution



4.4. The Rational Unified Process

- A modern process model derived from the work on the UML (Unified Modeling Language) and associated process.
- Normally described from 3 perspectives
  - A dynamic perspective that shows phases over time;
  - A static perspective that shows process activities;
  - A practive perspective that suggests good practice.

RUP phase model



RUP phases

- Inception
  - Establish the business case for the system.
- Elaboration
  - Develop an understanding of the problem domain and the system architecture.
- Construction
  - System design, programming and testing.
- Transition
  - Deploy the system in its operating environment.

Key points

- Requirements engineering is the process of developing a software specification.
- Design and implementation processes transform the specification to an executable program.
- Validation involves checking that the system meets to its specification and user needs.
- Evolution is concerned with modifying the system after it is in use.
- The Rational Unified Process is a generic process model that separates activities from phases.
- CASE technology supports software process activities.