10 THE SYSTEM DEVELOPMENT LIFECYCLE

Chapter Summary

Developing a computer system takes place in stages.
1: The first step is to find out what is required
2: this is followed by a design process that maps the requirements onto the IT.
3: The design is implemented by the programming process, which creates the computer code.
4: Finally, the system needs to be tested before it goes live. These stages can be formalized as a **system development lifecycle**.
The lifecycle just outlined is the **waterfall lifecycle**, an alternative to that is an iterative or evolutionary prototype lifecycle.

Developing a sizable computer system (or implementing a bought-in package) is a lengthy, costly undertaking – and often it does not go to plan. Mapping out the development in terms of a lifecycle and project plan is a good technique in reducing the risks of costly overruns or, worse, failure of the project. The project can be further organized by adopting a methodology that matches up stages with specified activities and techniques. In this chapter we will look at five different methods/methodologies:

- **Structured system analysis and design** (SSADM) – a complete SA&D methodology specifying stages and the techniques to be used at each stage
- **DSDM** – a methodology that formalises an evolutionary prototype lifecycle for rapid application development – techniques are not specified by DSDM
- **Object-oriented analysis and design using UML**. UML is a modelling language for OO design – UML does not specify a lifecycle but this book suggests how these techniques should be applied
- **Agile software development** – a *lightweight* (un-bureaucratic) approach to evolutionary system development;
- **Socio-technical SA&D** – an alternative approach that emphasises the human aspects of the computer system that is to be developed.

Techniques used for SA&D are taught in ***Chapter 11*** and further aspects of implementing a system are examined in ***Chapter 12***.