

1 INTRODUCTION TO SYSTEMS DEVELOPMENT

James Cadle

CONTENTS OF THIS CHAPTER

This chapter covers the following topics:

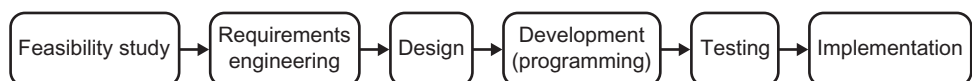
- the purpose of systems development;
- the scope of systems development;
- the typical stages in the systems development process;
- the relationship of systems development to other disciplines, such as project management, business analysis and systems architecture;
- how offshoring and outsourcing influences systems development;
- a synopsis of the remaining chapters of this book.

WHAT IS SYSTEMS DEVELOPMENT?

Systems development is the process of taking a set of business requirements and, through a series of structured stages, translating these into an operational IT system. The stages vary according to the development approach being used – described more fully in Chapter 2, 'Lifecycle types and their rationales' – but typically would include the activities shown in Figure 1.1, including:

- a feasibility study, to see if the project is worthwhile;
- requirements engineering to analyse the business need and specify the users' requirements;
- design of the system to meet the users' needs;
- development of the software needed to meet the requirements;
- testing of the software;
- implementation of the solution.

Figure 1.1 The main stages of systems development



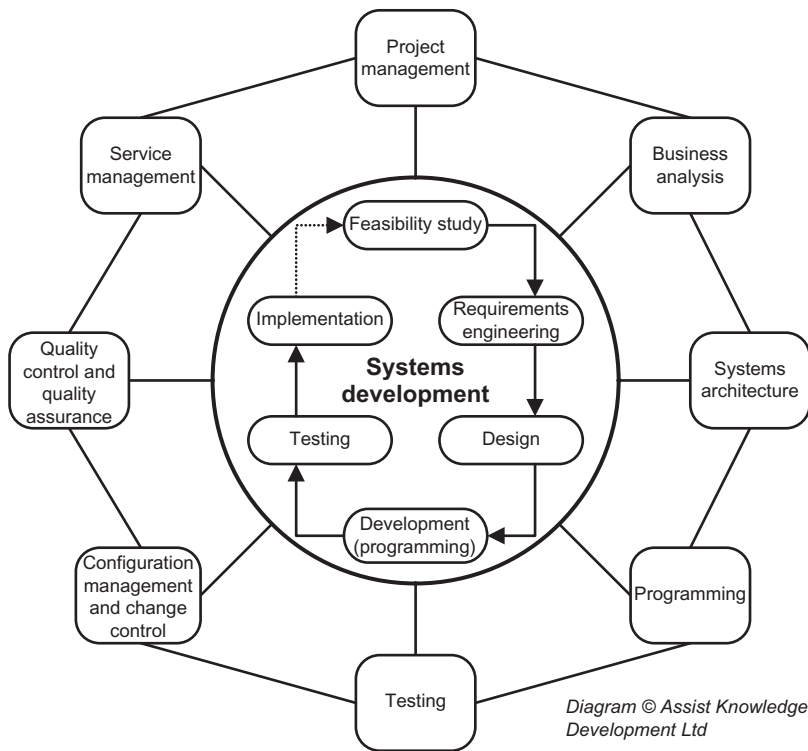
Other activities may also be involved, such as the procurement and installation of the hardware on which the system will operate.

At one time, systems development was undertaken in a rather haphazard, ad hoc way, and the result depended to a large extent on the competence and enthusiasm of the individual developers. Today, the core importance of IT systems within most organisations means that more structured and manageable processes have been introduced, to reduce the unpredictable 'human element' and to make possible the construction of larger and more complex systems.

SYSTEMS DEVELOPMENT AND OTHER DISCIPLINES

Systems development does not take place in isolation; it is part of the intricately connected web of disciplines illustrated in Figure 1.2.

Figure 1.2 Systems development in a wider context



The relationship of systems development to other disciplines may be summarised as follows:

Project management	If a systems development project is to be successful, technical expertise is not enough; effective project management is also required. The project manager plans the undertaking, mobilises the resources required and controls and coordinates the work. The project manager also ensures that the various stakeholders are kept onside and committed to the project's success. Good project management frees the development team to concentrate on the difficult technical task of devising and implementing the solution.
Business analysis	Business analysis is concerned with investigating the business situation and finding out what are the problems to be solved or opportunities to be exploited. It involves developing holistic solutions to business issues, which very often involve the use of IT in some way. Business analysts are also important for eliciting, documenting and managing the requirements for the new or enhanced IT systems and services.
Systems architecture	Systems architects are concerned with developing an architecture for the organisation to support and coordinate its systems and provide a coherent platform for expansion and development.
Programming	Although within the span of systems development, this is a specialist area which calls for high levels of technical expertise, not least in how to exploit to the full the possibilities offered by the hardware and software available.
Testing	The tester's role appears at first to be counter-productive in that he or she is trying to prove that the system does not work. This is an iterative process and, when the tester struggles to identify further defects in any version, it can be stated with some confidence that the system appears to be satisfactory. An important point to realise, though, is that no testing, however thorough, can deliver assurance that the software is one hundred per cent error-free.
Configuration management and change control	As systems have become more complex, it has become even more important to know the latest version of the system, the components it is made up of and how these relate to each other. The discipline of managing these components is known as 'configuration management' and it is related to change control, which is a process for managing changes to a system or product in a controlled way.
Quality control and quality assurance	Quality control consists of the processes – for example, reviews or code inspections – that are employed within a project team to ensure that the delivered products meet their defined quality

criteria. Quality assurance is an external process that ensures that quality control is being exercised; it also puts in place things like standards to assist in quality control.

Service management Service management is concerned with managing and maintaining the services provided by IT systems. It includes, for example, such activities as facilities management – controlling the supporting IT infrastructure – and applications management – supporting and enhancing the applications once they have been delivered initially.

OFFSHORING AND OUTSOURCING OF SYSTEMS DEVELOPMENT

Two changes that have affected many organisations in recent years have been the offshoring and/or outsourcing of systems development work. These two practices are often referred to together but, in fact, they are separate and one does not necessarily imply the other:

- Offshoring involves using development resources in other countries, usually because high-quality resources can be secured for considerably less cost than in the organisation's home country. For example, India has proved to be a popular place for organisations to establish their development centres because the country's education system produces a large number of high quality IT graduates. Leading firms such as Tesco and HSBC have large development centres in India. More recently, development resources have also been found in ex-communist European countries, such as the Ukraine and the Baltic states.
- Outsourcing means handing over the development work to specialist IT services firms and consultancies. An outsourcing contract may cover just development work or, often, the supplier takes complete responsibility for the organisation's IT systems. Cost may be a driver here too, but often the desire to get control of a spiralling IT budget and to transfer responsibility and risk are also important considerations.

Of course, outsourcing and offshoring are sometimes combined, in that the outsourced supplier chooses to perform its development work overseas, for the reasons already mentioned.

In addition to the claimed benefits, there are of course possible downsides to both offshoring and outsourcing, for example:

- **Offshoring:** there can be delays and communication difficulties associated with working with developers who are a long way away, whose first language is not the same as the customer organisation and whose culture is very different. It could be argued, however, that this just reinforces the need for greater precision in the definition of business requirements, which is a good thing.
- **Outsourcing:** one of the chief dangers here is that the customer organisation loses direct control of systems that are critical to its business objectives. In addition, knowledge of these key systems now resides in the supplier organisation rather than being retained in-house.

IN THE REST OF THIS BOOK

The chapters that follow explore the elements – the frameworks and models, processes, procedures and techniques – of systems development in more detail, and here we provide a foretaste of what is to come.

Chapter 2: Lifecycle types and their rationales

A lifecycle provides a framework and structure for undertaking systems development. Over the years, different lifecycles have been developed and employed, ranging from the traditional, linear, step-by-step, 'Waterfall' approach to the currently-popular 'Agile' one. This chapter presents the different lifecycles and assesses their relative strengths and weaknesses.

Chapter 3: Analysing the business need

Before embarking on any system development project, business analysts should examine the real business need and evaluate the options available to meet it. This analysis should also consider the non-IT issues (changes to organisation structures, to business processes and to people's jobs) that will have to be addressed if the system is to be implemented effectively and provide the expected business benefits.

Chapter 4: Making a business case

The business case is – or should be – an examination of the justification for undertaking a systems development project and a rigorous analysis of the costs, benefits, impacts and risks of the courses of action available. Assuming that the case is made initially, the business case should be revisited throughout the project's lifecycle to ensure that it has not been invalidated by, for example, rises in costs or changes in the external environment.

Chapter 5: Requirements engineering

If a system is to be delivered that meets the needs of the organisation, it must be based on well-defined requirements – so that the developers know what is to be produced. Requirements engineering provides a framework and techniques for creating high-quality requirements as a basis for the development work.

Chapter 6: Programming and development approaches

A major decision to be made is whether, having defined the requirements, the solution should be built from scratch or whether a commercial, off-the-shelf (COTS) solution should be produced. Assuming that at least some development work is to be undertaken, this chapter reviews the different programming and development methods that could be employed.

Chapter 7: System modelling techniques

Most engineering disciplines make use of models to assist in the conceptualisation and specification of the solution. In the case of systems development, the product can be

specified in terms of the functional or processing aspects, the data requirements and the 'dynamic' or event-driven view. This chapter presents approaches to modelling from these three perspectives.

Chapters 8 and 9: System design

Design is the stage in the development process where decisions are made about how to meet the defined requirements using the hardware and software available. Both the functions/processing and the data need to be designed, and this often involves making compromises between what would be ideal and what is practical given the technology, time and resources available. These two chapters review the challenges of design and conclude with a discussion of the benefits of using defined design patterns to assist in the process.

Chapter 10: Solution-related architectures

Architecture in IT is similar to architecture in building in that it provides an overall framework and structure for the development of systems. This chapter explains the purpose and approach of architecture, the stakeholders involved and the role of such concepts as Service Oriented Architecture and Service Oriented Development.

Chapter 11: Quality and testing

Systems must not only be developed on time and within budget, but they must also achieve appropriate levels of quality. This chapter defines what is meant by the term 'quality' in the IT context and presents methods that can be used to assure software quality.

Chapter 12: Implementation and changeover

The introduction of systems into service is often a very challenging aspect of systems development involving, as it does, moving from manual or older systems to the new ones, training the staff, data conversion and so forth. This chapter reviews these issues and also considers the different approaches to implementation, for example as a 'big bang' or in a phased manner.

Chapter 13: Maintenance and evaluation

Surveys have shown that, in most cases, the majority of expenditure on IT systems occurs after they have been introduced into service – to fix problems, make enhancements, adapt to changes in other systems and so on. Although live operation follows systems development, this chapter explains the purpose of evaluation and maintenance and shows how decisions made during the development can assist, or hamper, the longevity of the systems.

Chapter 14: Solution development tools

Systems development can benefit hugely from the development team having at their disposal software support tools to help them do their work. These can range from tools to control the vast amount of documentation that is produced, to aids for the developers,

to tools to assist testing. This chapter looks at the pros and cons of software support tools and provides guidance on what to look for in a tool.

FURTHER READING

Cadle, J. and Yeates, D. (2008) *Project management for information systems* (5th edition). Pearson Prentice Hall, Harlow.

Paul, D., Yeates, D. and Cadle, J. (2014) *Business analysis* (3rd edition), BCS, Swindon.

Skidmore, S. and Eva, M. (2004) *Introducing systems development*. Palgrave Macmillan, Basingstoke.

Various authors (2002) *Best practice for application management*. OGC/TSO, London.

Yeates, D. and Wakefield, T. (2004) *Systems analysis and design* (2nd edition). FT Prentice Hall, Harlow.