System development lifecycle – waterfall model

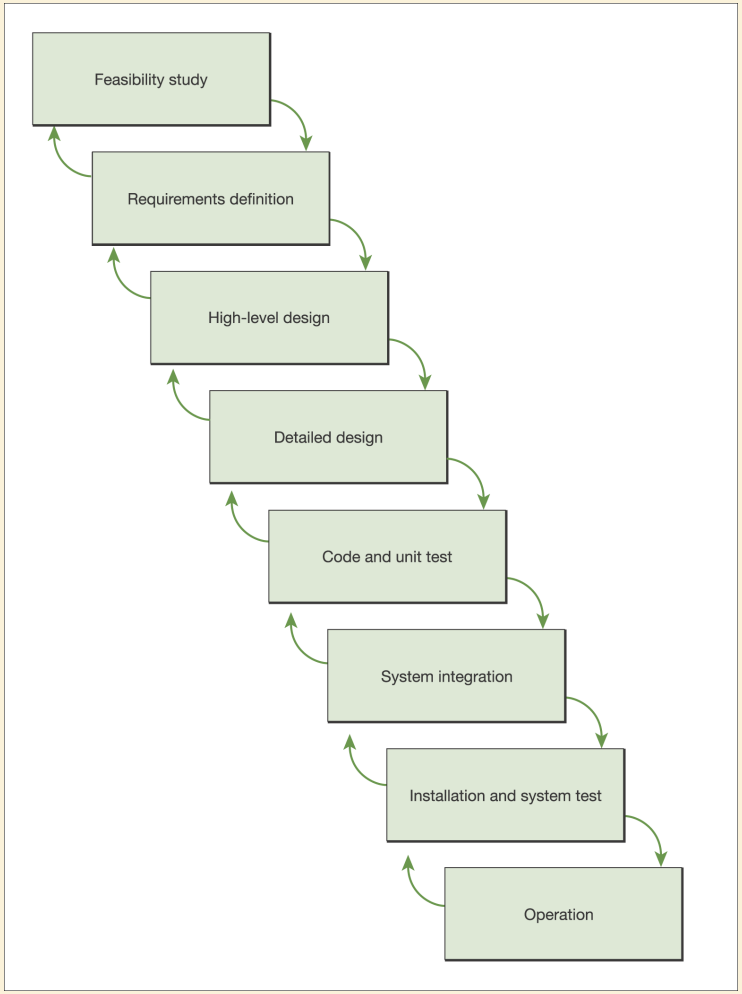


Figure 6.1 The waterfall model of system development lifecycle

The ‘b’ model

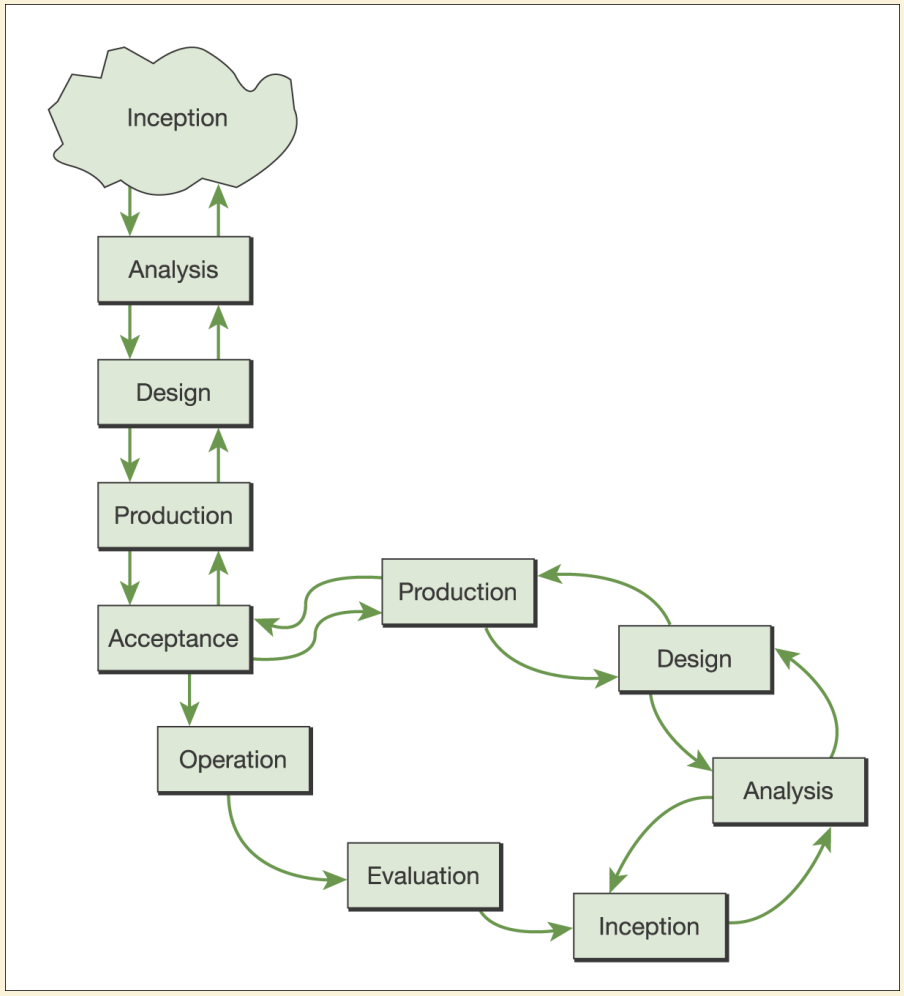


Figure 6.2 The ‘b’ model

Source: N D Birrell and M A Ould, *A Practical Handbook for Software Development*, Cambridge University Press, 1985

The ‘V’ model

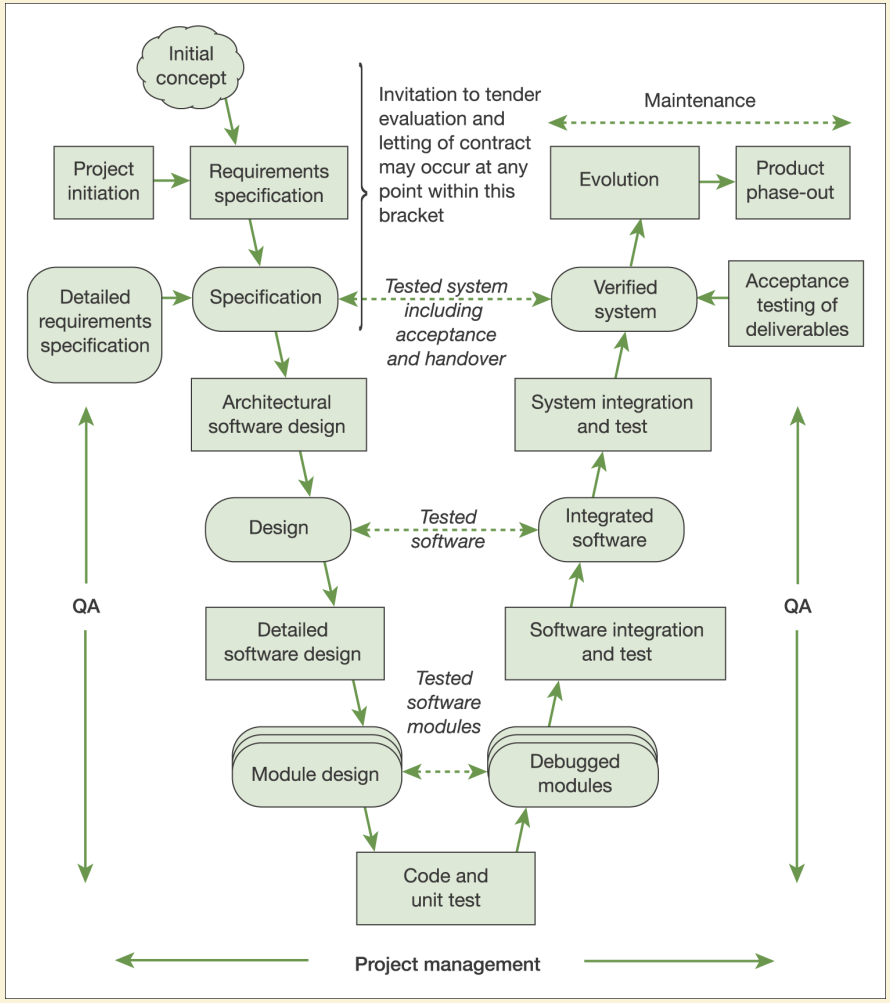


Figure 6.3 The ‘V’ model

Source: Reproduced with permission of the National Computing Centre Limited from the *STARTS Guide*, 1987, which was supported by the Department of Trade and Industry

The incremental approach

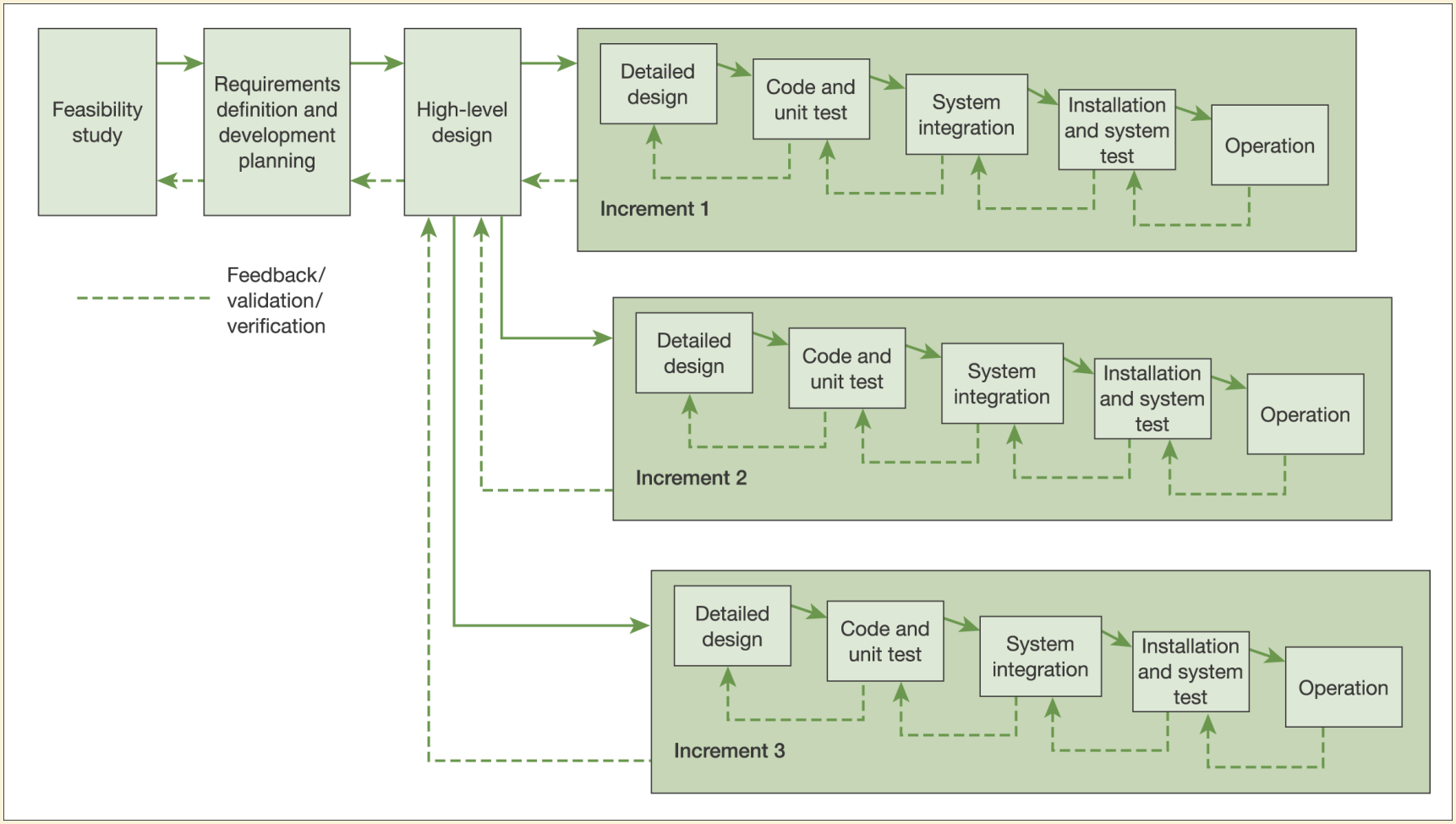


Figure 6.4 The incremental model

The spiral model – evolutionary development

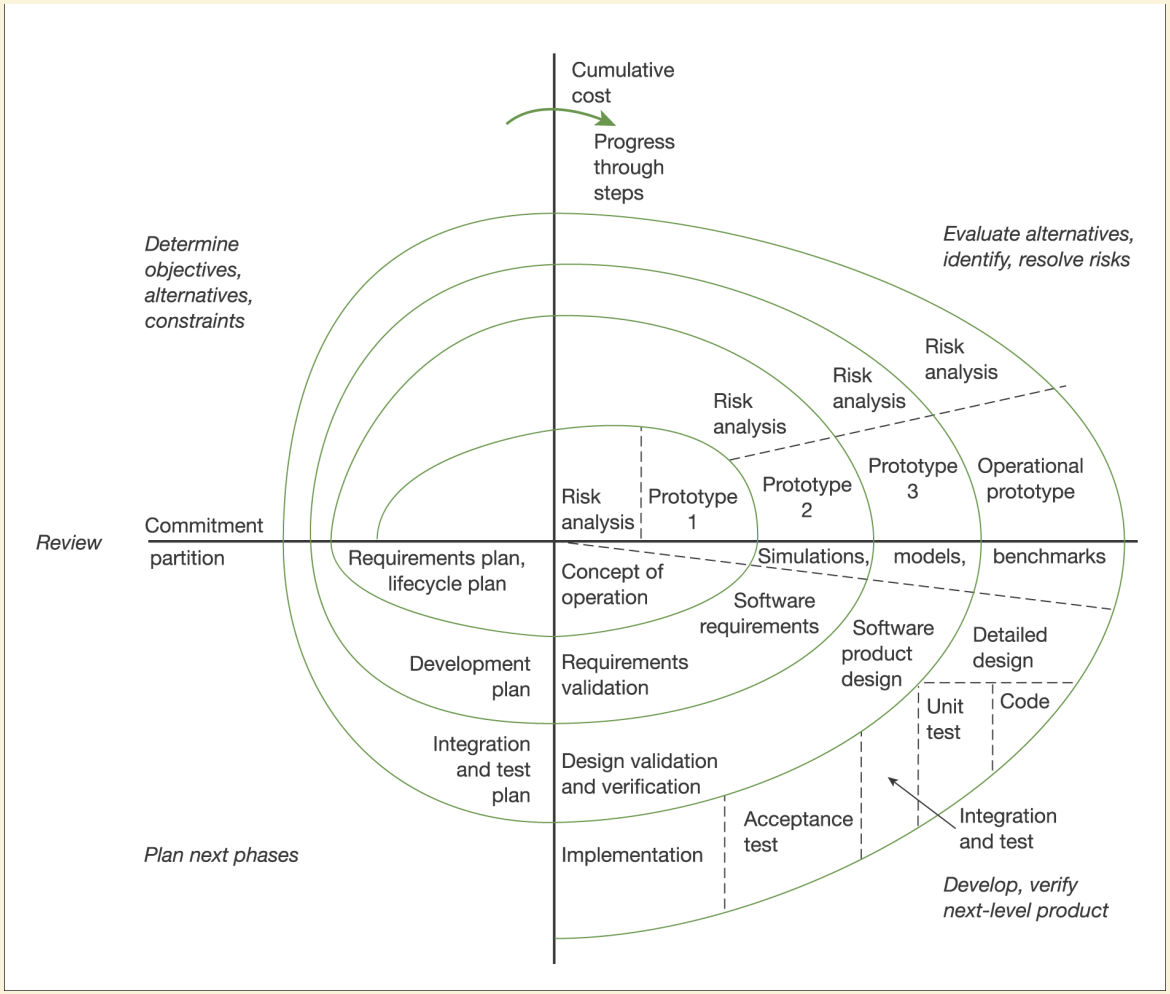
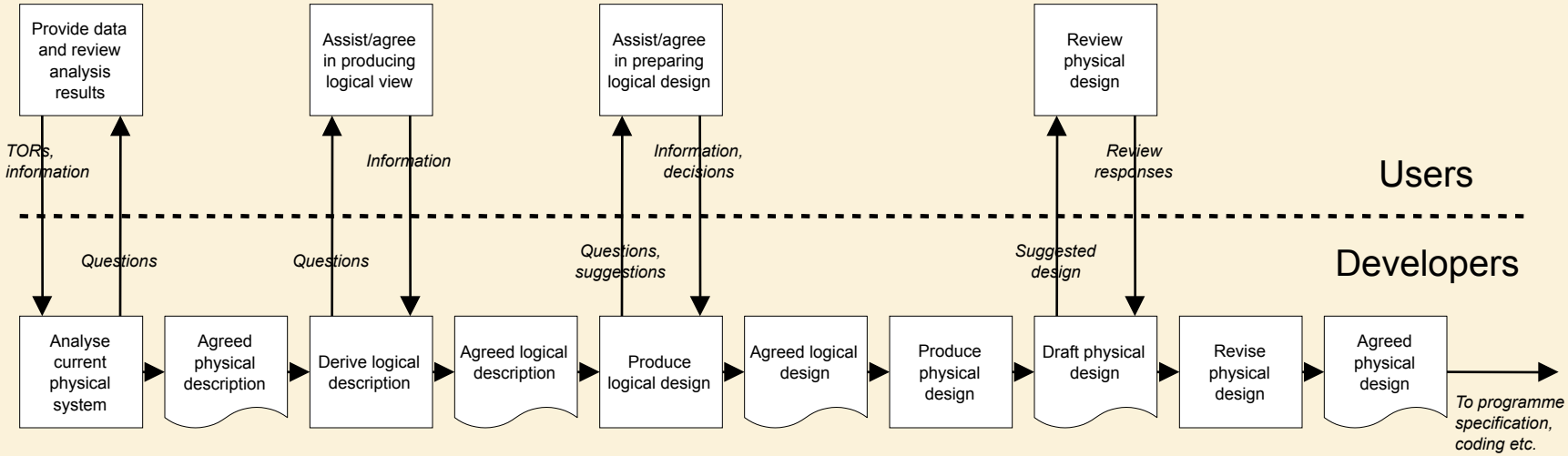


Figure 6.5 Boehm's spiral model

Source: © 1988 IEEE

Structured systems development



Greater user involvement at all stages; driven by users

Agile approaches – origins

- Addresses long-windedness of other approaches
- Prototyping used to:
  - assist users define requirements by demonstrating possibilities
  - investigate novel methods of working
  - test performance implications
  - assist in considering work practices

Agile approaches – features

- Success dependent on empowerment of users and developers
- Deliverables reviewed for business fitness
- Testing integral to iterative lifecycle
- All changes reversible
- Incremental/partial delivery acceptable
- ‘Timeboxing’ used to control timescale (and budget)
- Workshops widely used

Agile methods 1 – Scrum

- Scrum.
- Origins in USA.
- Scrum is daily meeting to review progress and reset priorities.
- Development done in 30-day ‘Sprints’.
- Every aspect of the project time-boxed.
- Project manager is ‘ScrumMaster’ – different from the usual PM role – teams self-governing.

Agile methods 2 - DSDM

- Eight principles:
  1. Focus on business need.
  2. Deliver on time.
  3. Collaborate.
  4. Never compromise quality.
  5. Develop solution iteratively.
  6. Build incrementally from firm foundations.
  7. Communicate continuously and clearly.
  8. Demonstrate control.

Object-oriented development

- Object is a package of software containing:
  - 'variables' (data).
  - 'methods' (processes)
- Objects communicate via messages.
- System is built up from intercommunicating objects.
- Deals with problem of integration of large systems.

UML and Unified Process

- Unified Modeling Language provides visual language for OO projects.
- Unified Process provides process model.
- Approach ‘open’ (non-proprietary) but Rational Corporation offers Rational Unified Process.
- Four phases to process:
  - Inception
  - Elaboration
  - Construction
  - Transition.

Component-based development

- Development of reusable components
- Aim – to create libraries of components that can be combined to build new systems
- Long-term benefits – reduce development costs and produce more reliable systems
- Short-term costs often higher – because of need to consider wider usage of components

Extreme programming

- Created to deal with rapidly changing requirements
- Works best on relatively small projects
- Or on enhancements to existing systems
- Developers work in pairs
- Testing – integral part of process
- Emphasis on frequent releases of small-scale packages of software

Package-based IS projects

- Quicker and cheaper to buy commercial off-the-shelf (COTS) solution than build from scratch.
- Two main types of project:
  - Package-constrained – users adapt to what package can do.
  - Package-based – package tailored to users' exact needs.
- Extensive tailoring probably more expensive than bespoke development.

Soft systems and the Socio-Technical Approach

- Not really an IT development method
- But does consider a wider ‘human activity system’ (business system) of which IT is a part
- Recognizes that real-world problems rarely black and white

Business process re-engineering

- Originated by US consultants Michael Hammer and James Champy.
- Involves going back to first principles and re-designing optimal business systems.
- Leads to radical changes to organizations and fundamental changes to processes.
- High reward – but high risk also.

Functional organization

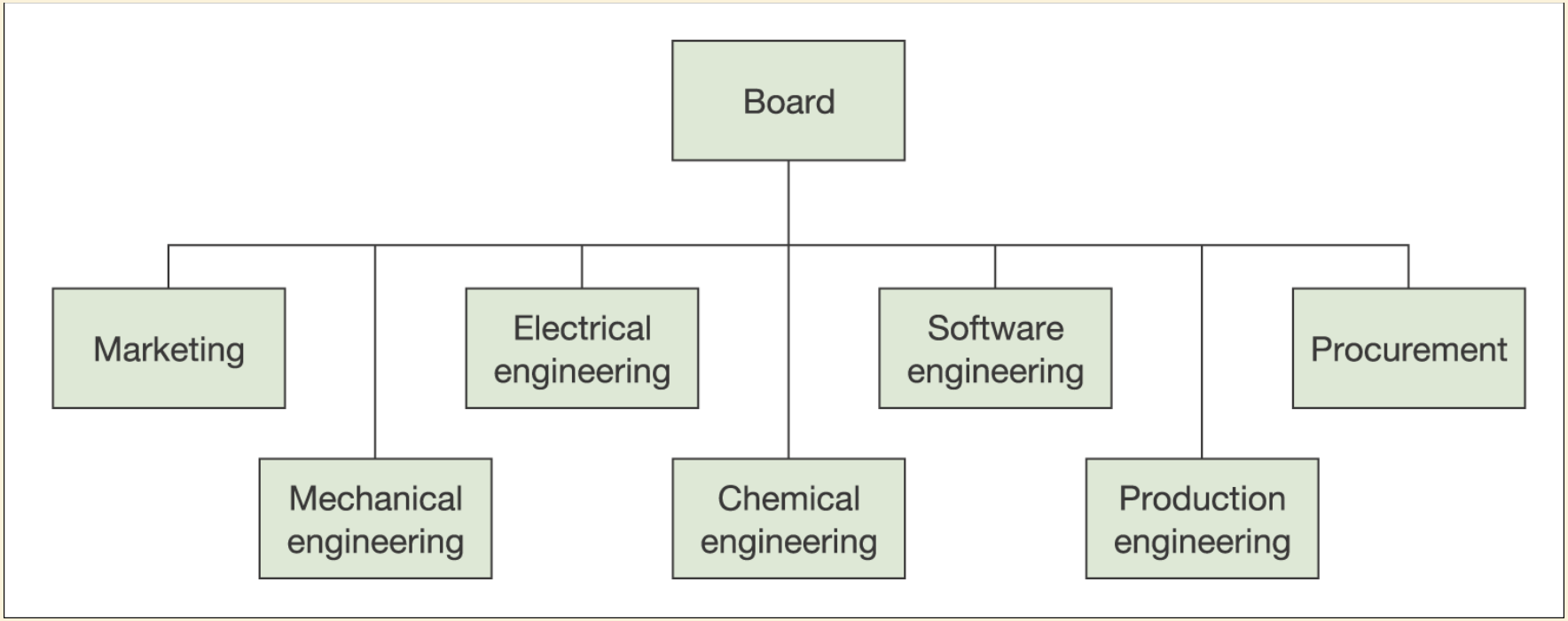


Figure 4.1 Functional organization structure

‘Pure’ project structure

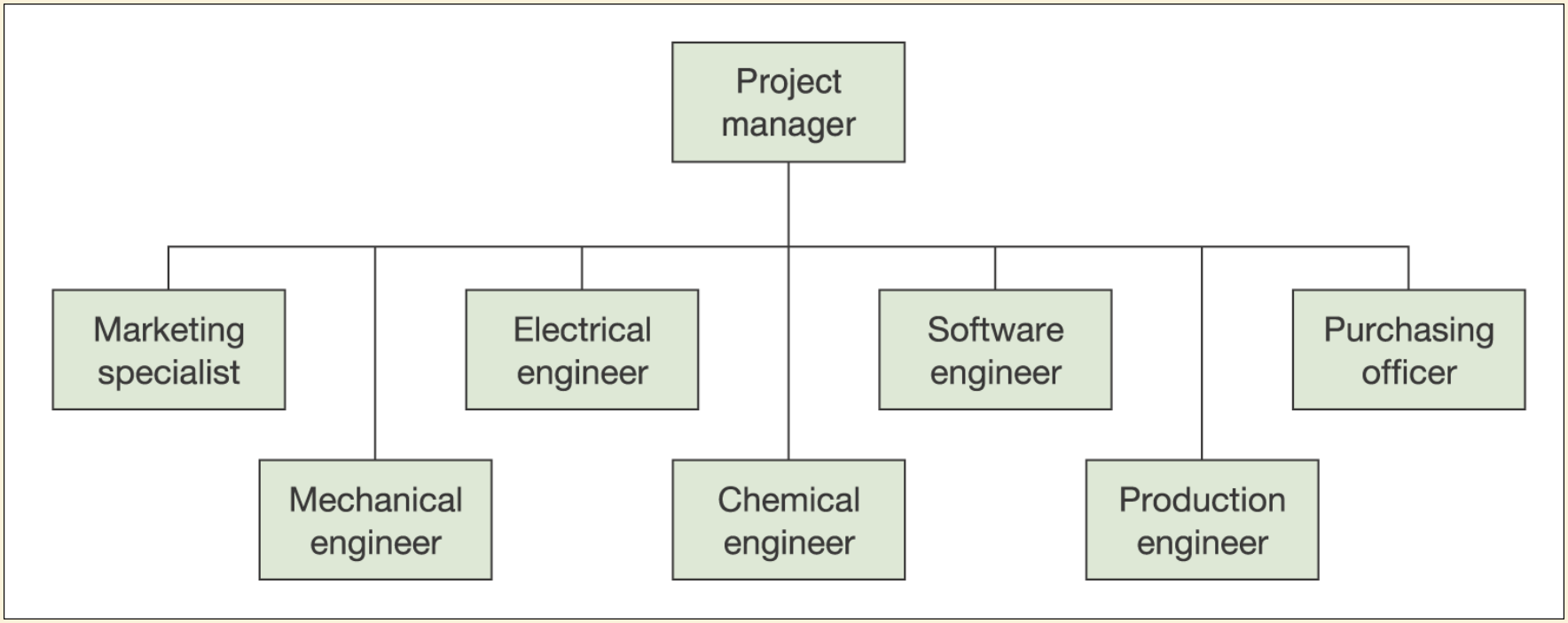


Figure 4.2 ‘Pure’ project structure

Matrix structure

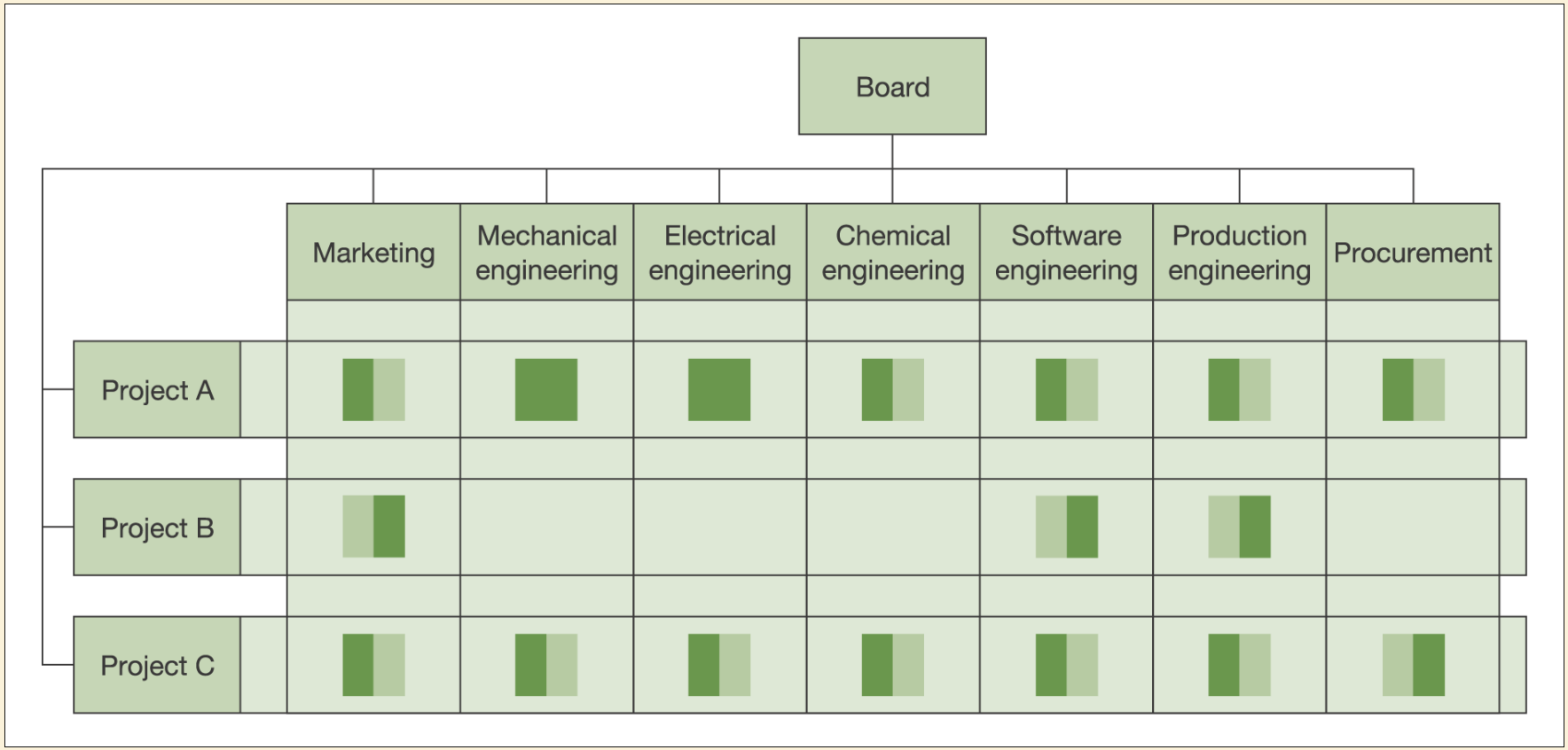


Figure 4.3 Matrix structure

Generic project organization

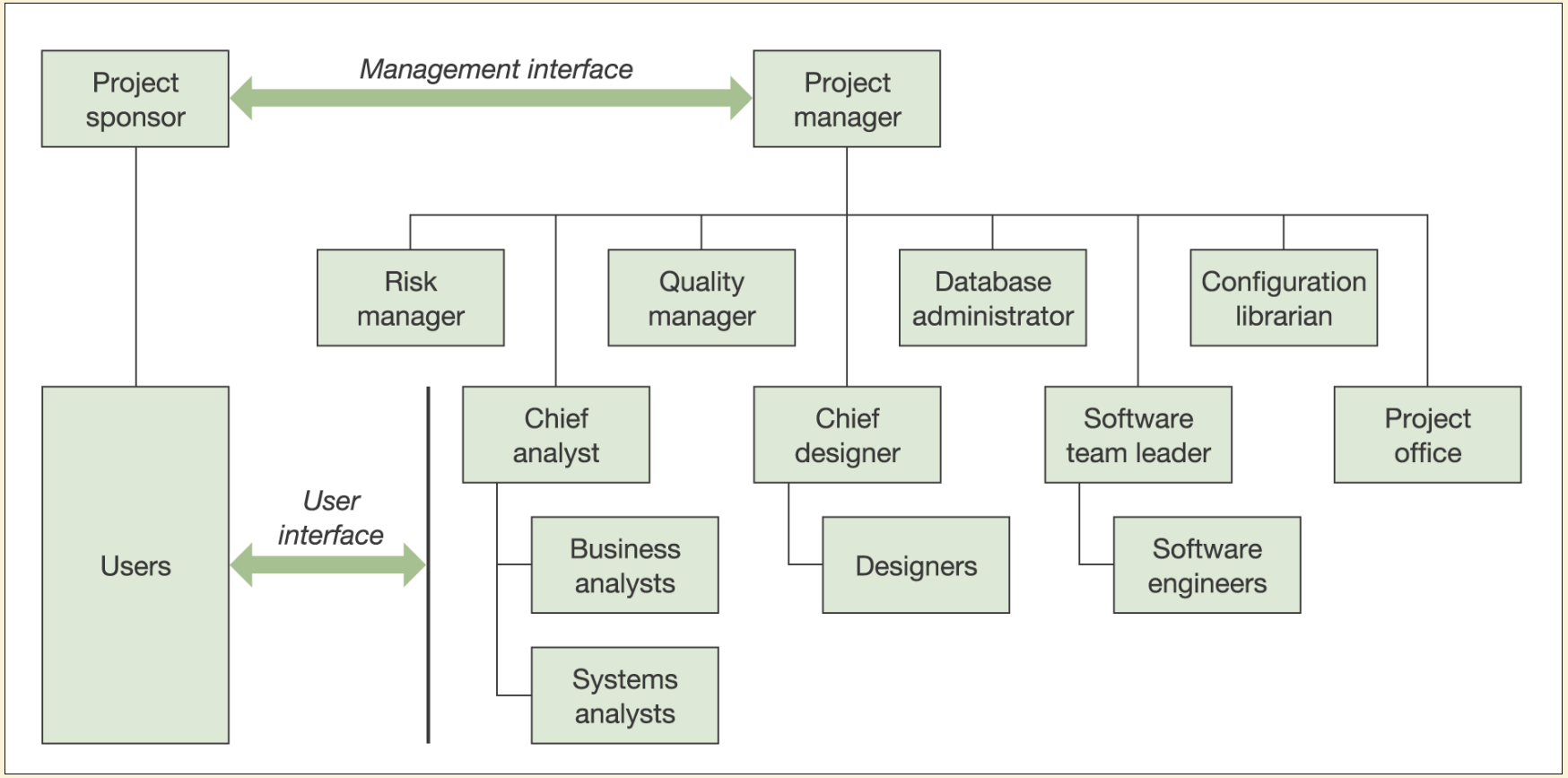


Figure 4.4 Generic project organization and roles

Programme and portfolio management

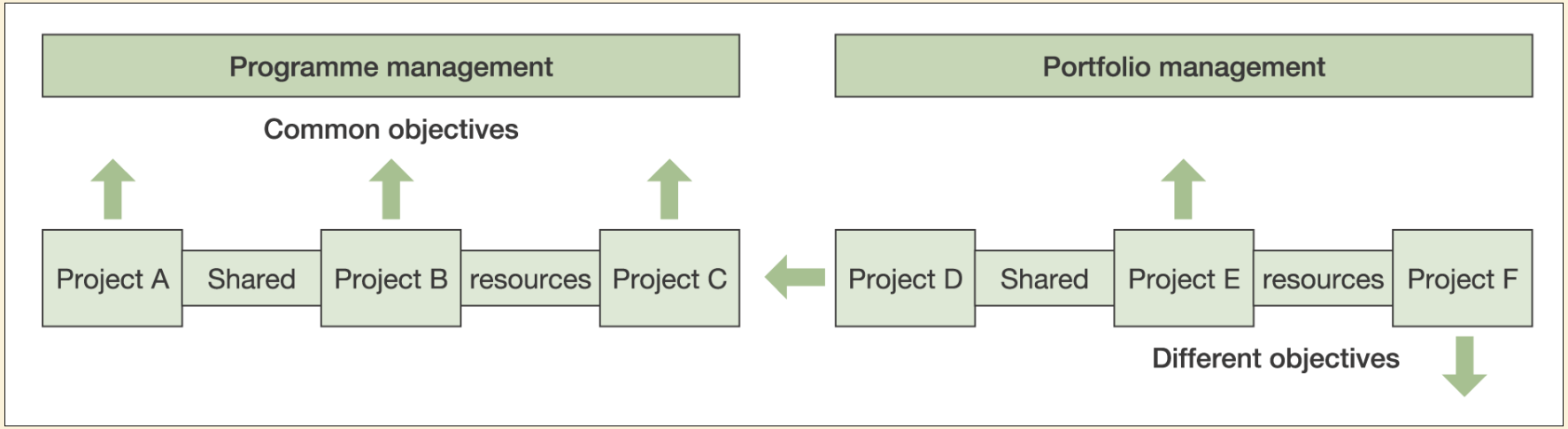


Figure 4.5 Programme and portfolio management

PRINCE2® organization structure

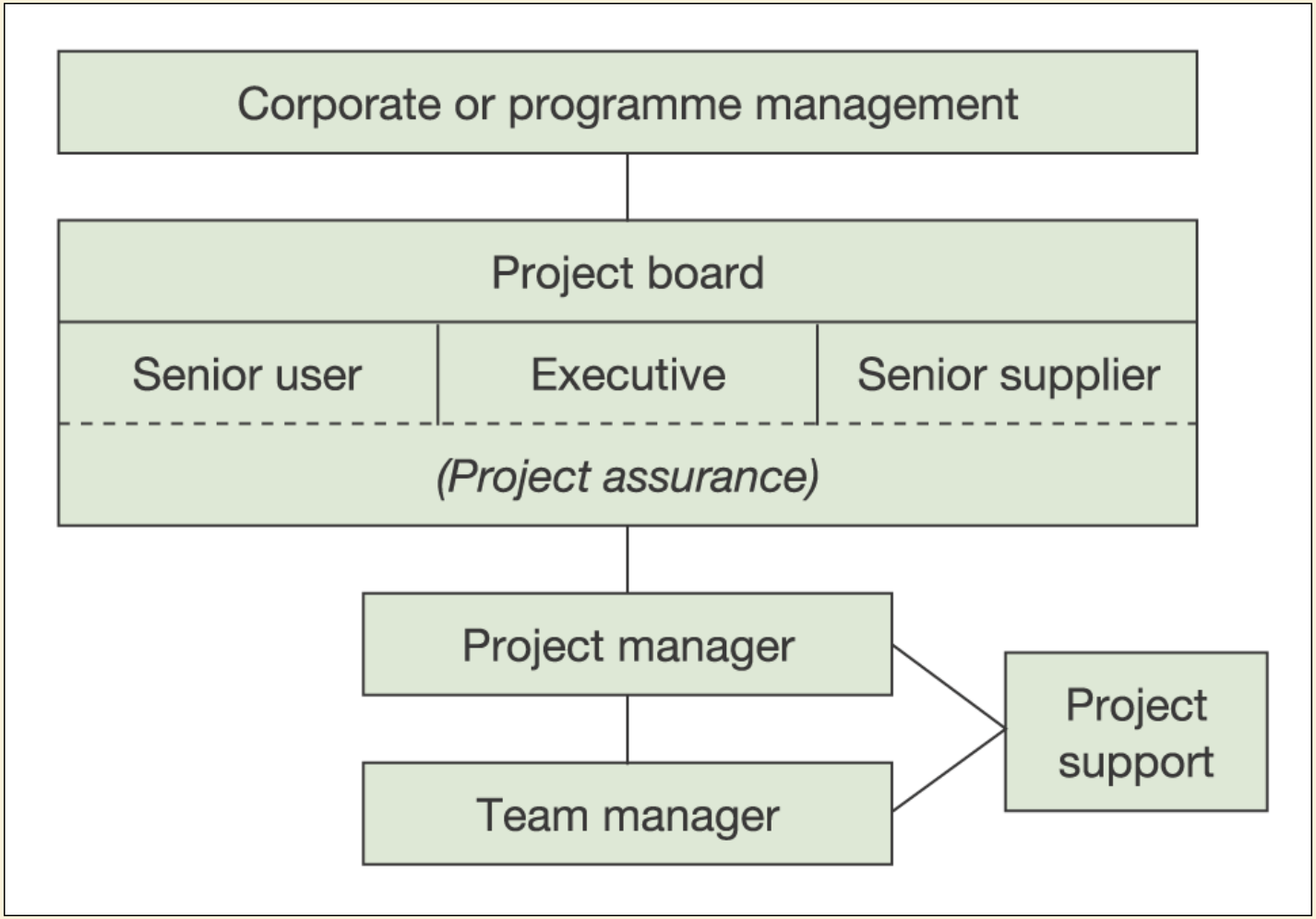


Figure 4.6 PRINCE2® organization structure

Scheduling effort and elapsed time

- Effort = total volume of work.
- Elapsed time depends on effort and also:
  - How many resources are available.
  - What proportion of their time is available to the project.
  - Delays outside the team's control (eg lead times for hardware).
  - Dependencies on others.

Network diagram

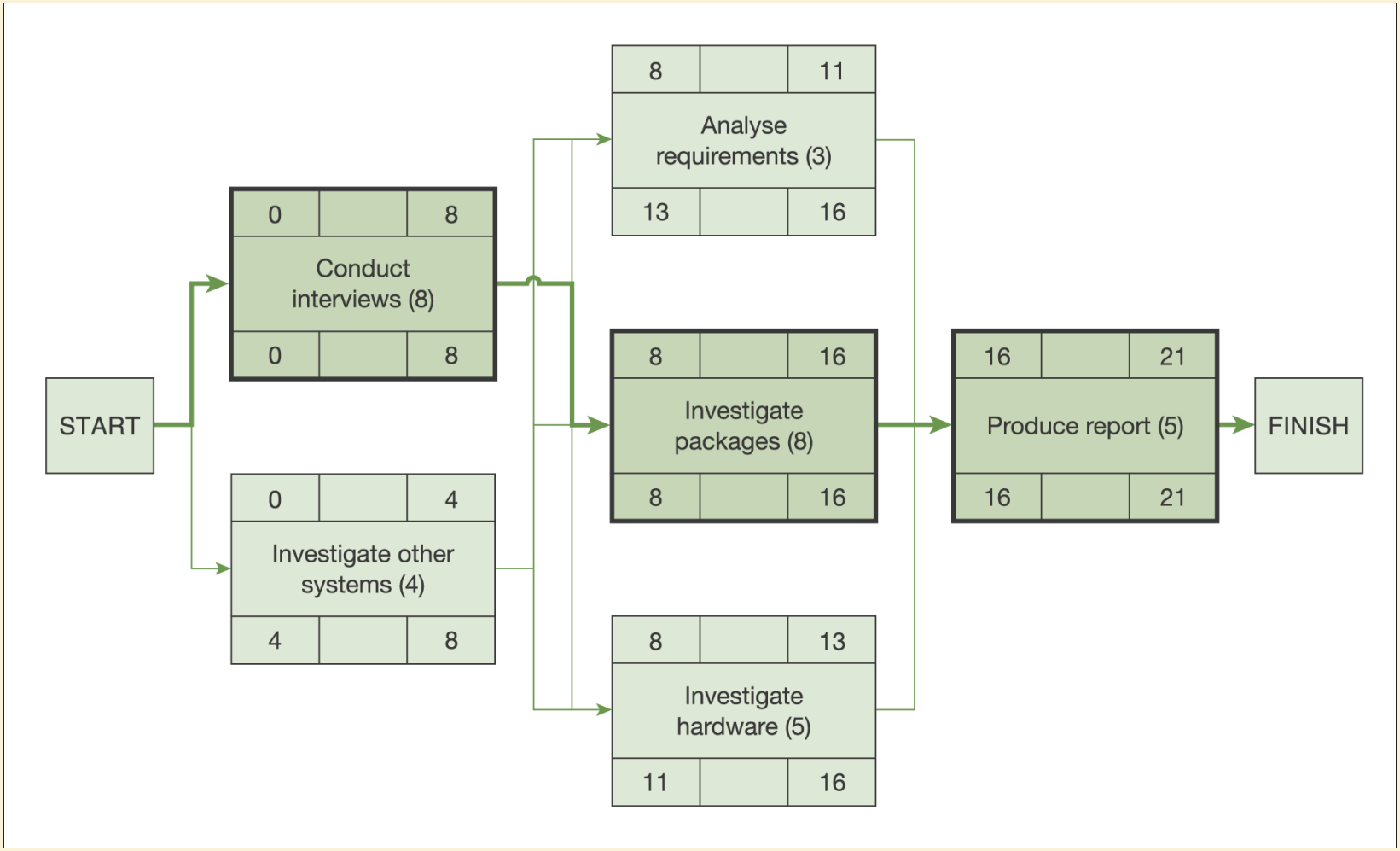


Figure 10.1 Dependency network with activity durations

Bar chart

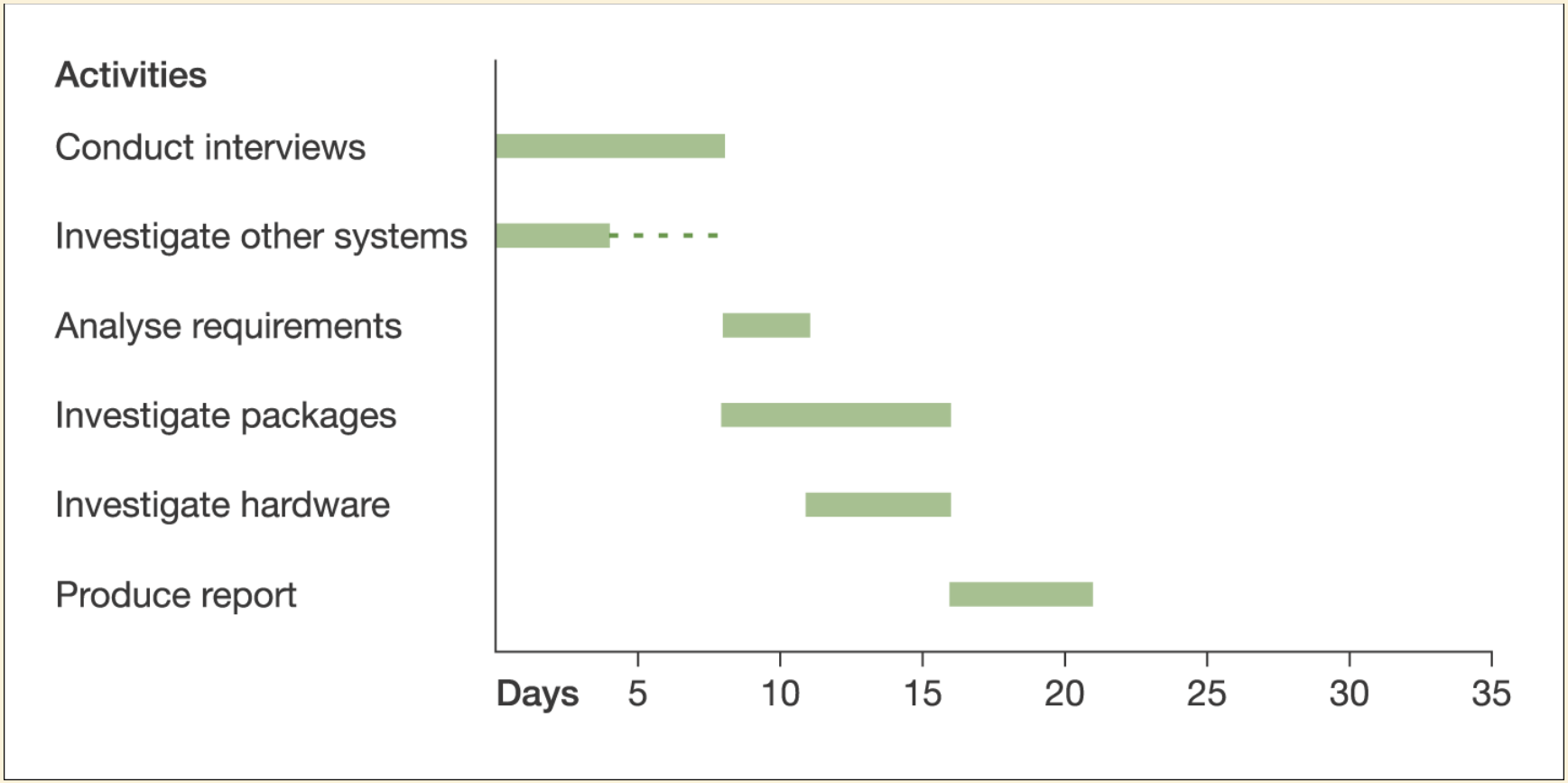


Figure 10.3 Schedule for two-person team showing parallel activities

Bar chart with milestones added

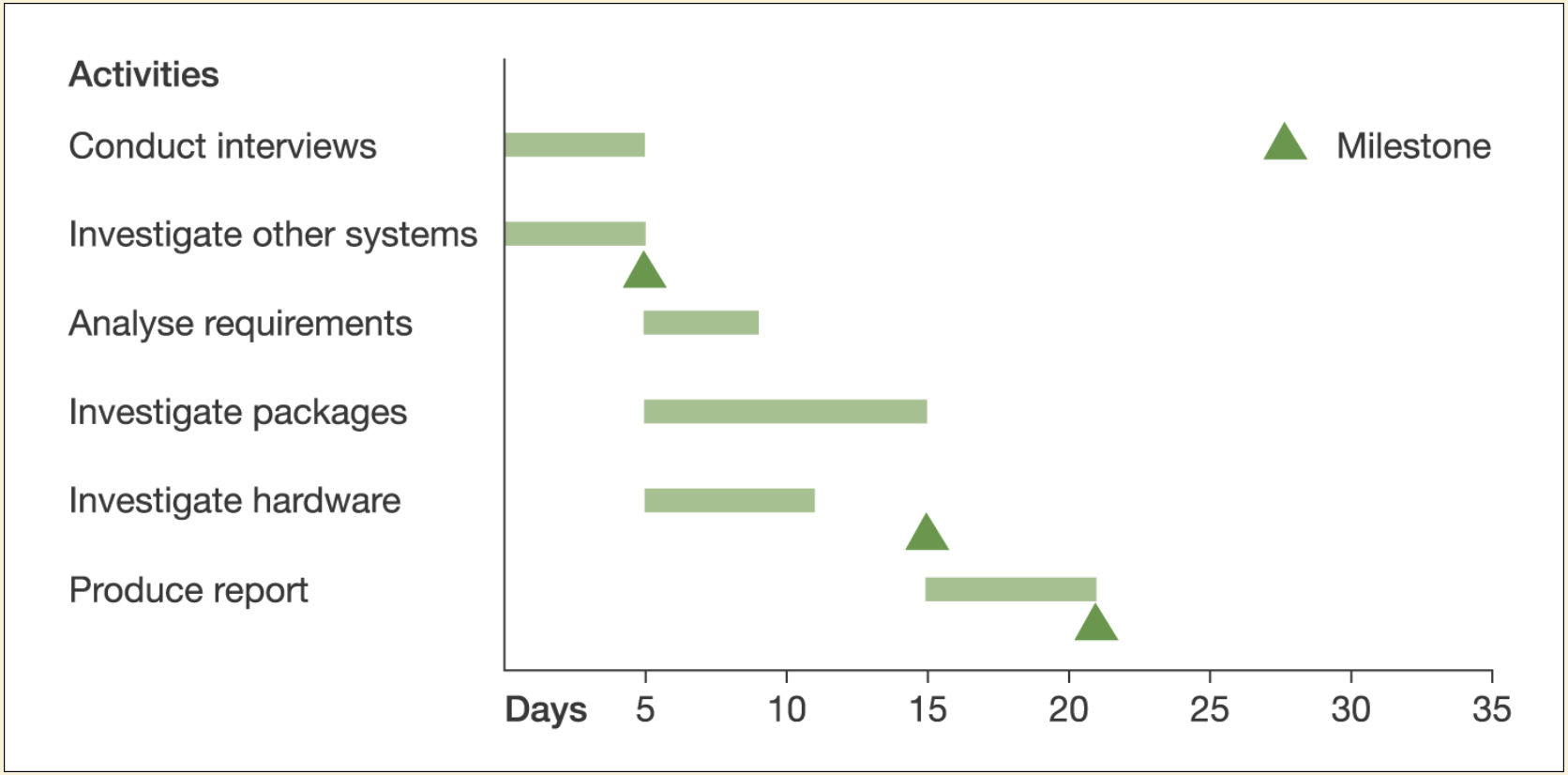


Figure 10.6 Bar chart showing project milestones

Bar chart with ‘overhead’ task added

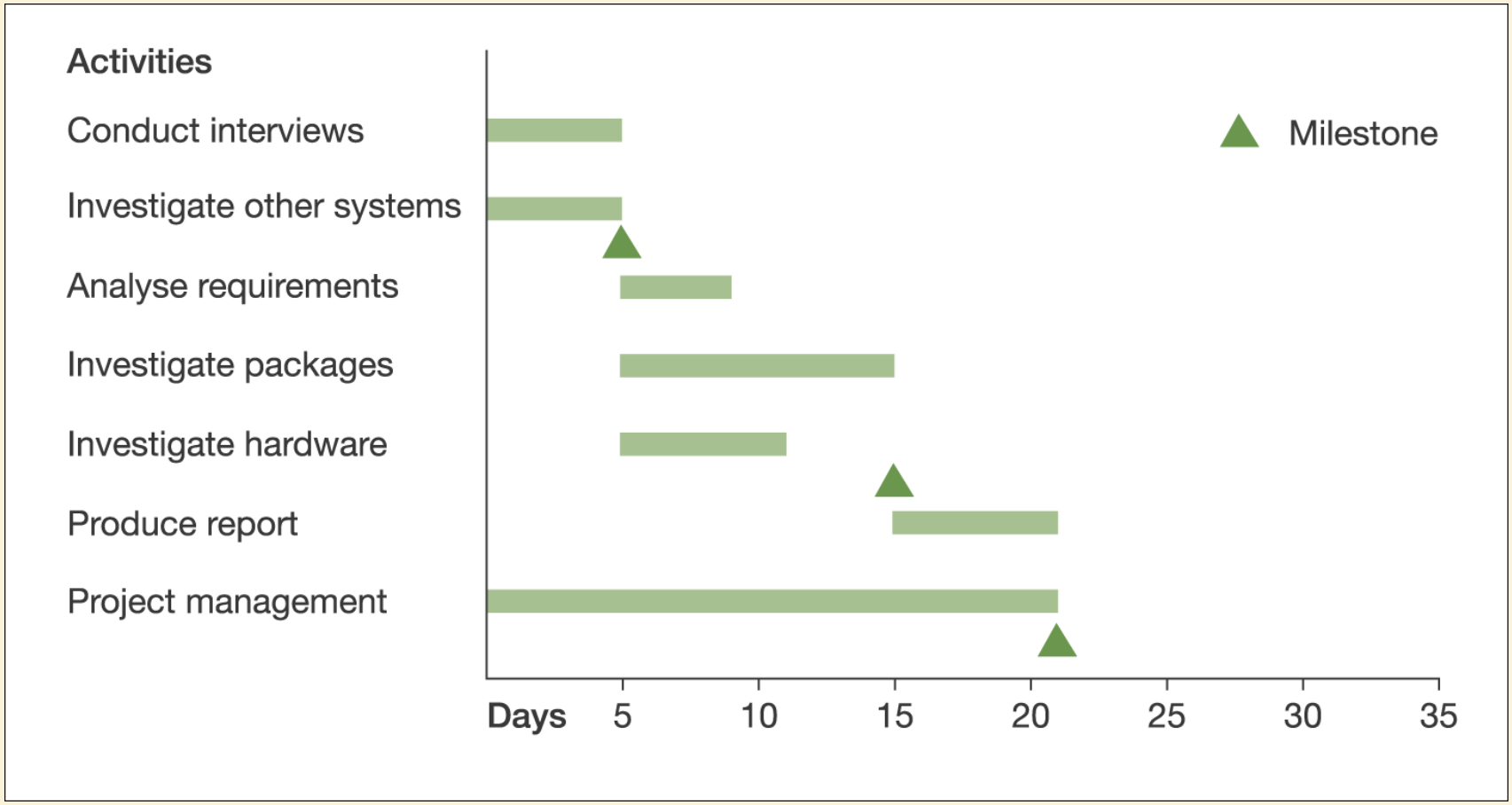


Figure 10.7 Bar chart showing project management as continuous activity over project

Bar chart and resource histogram

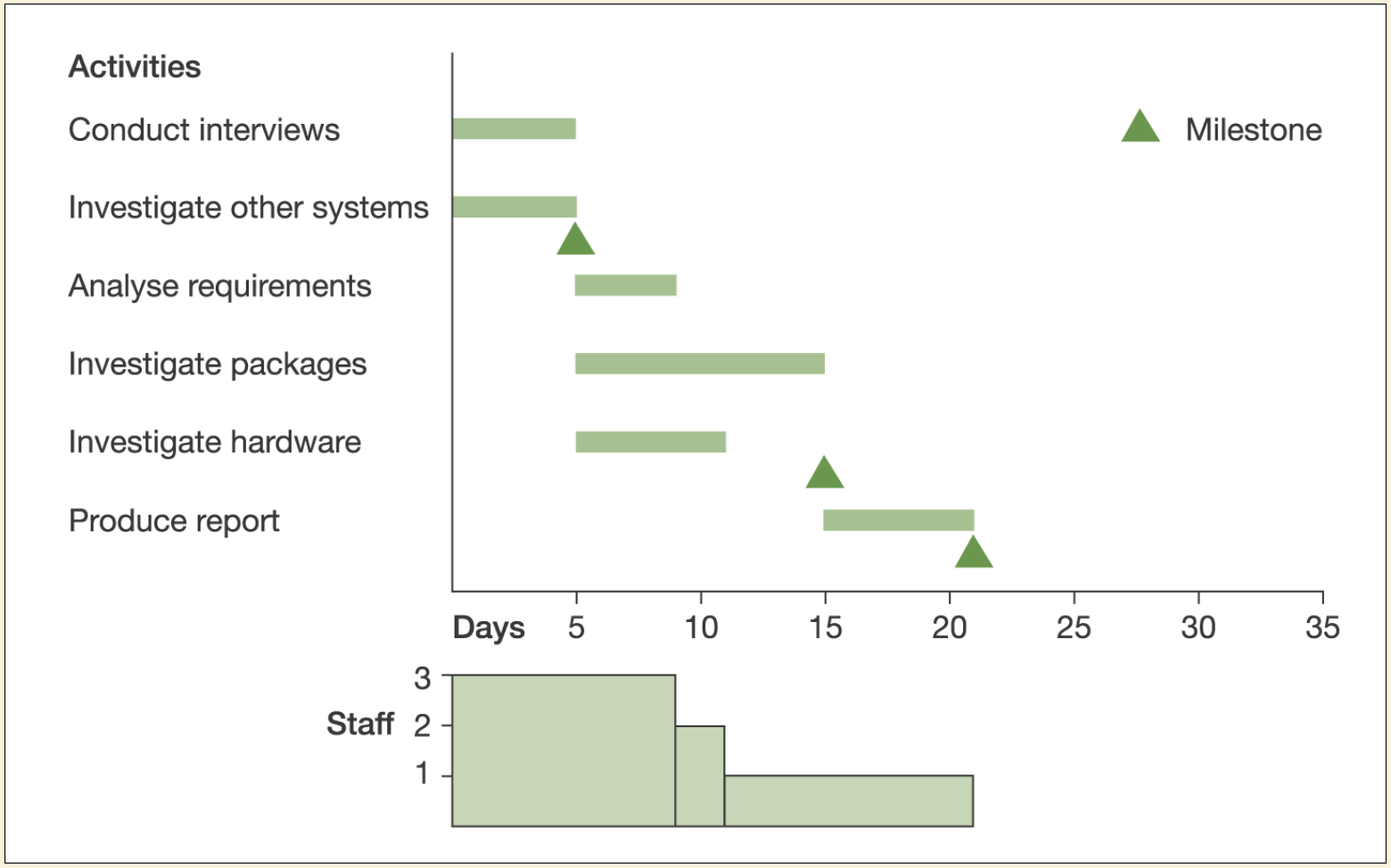
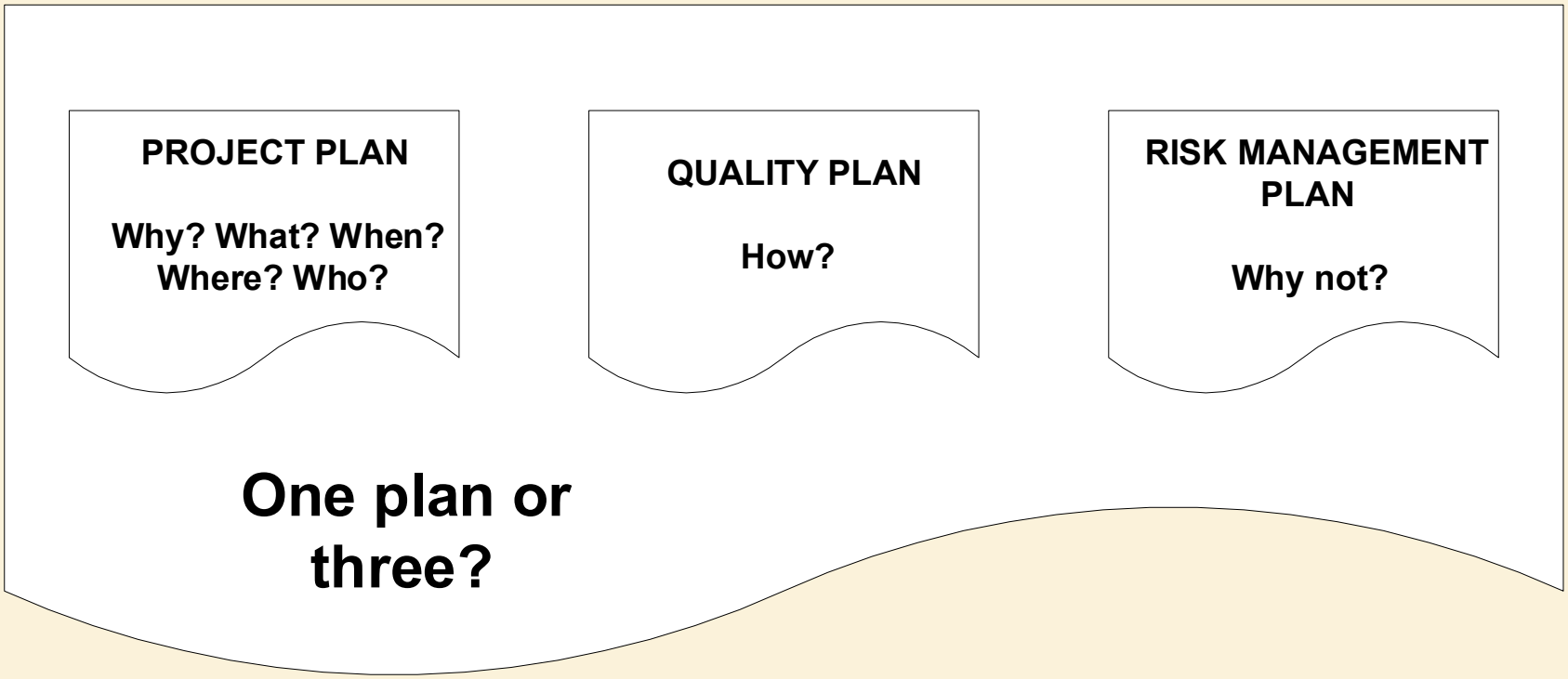


Figure 10.8 Bar chart with resource histogram

The project and other plans



PRINCE2® plans

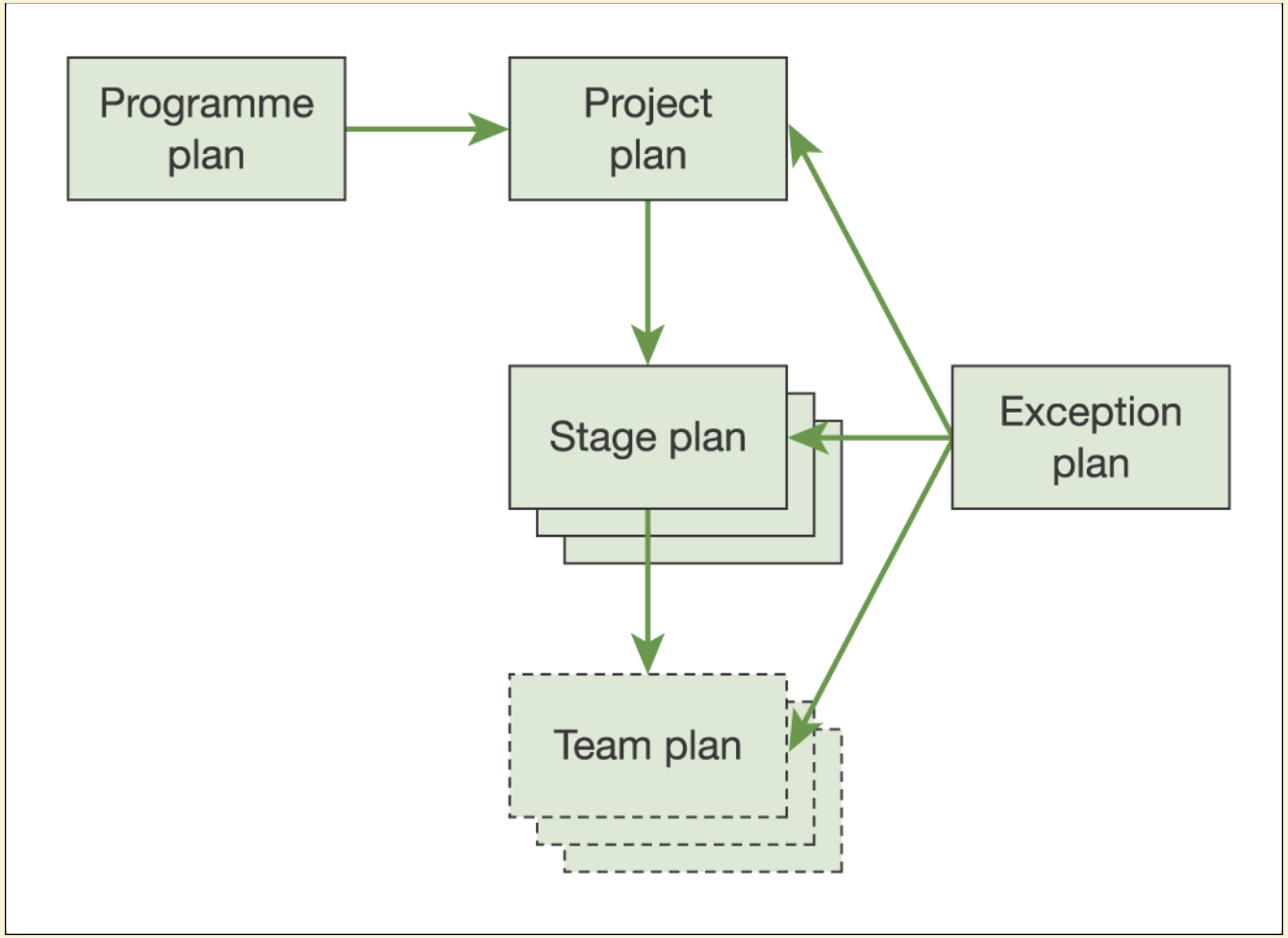


Figure 10.10 PRINCE2® plans

Contents of PRINCE2® project/stage plan

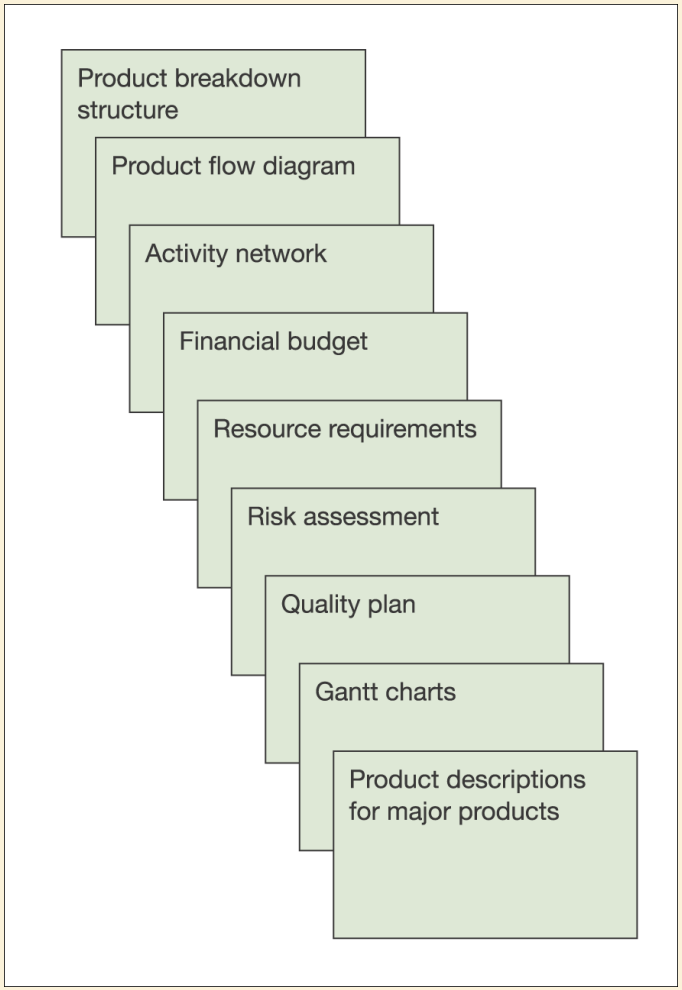


Figure 10.11 Contents of PRINCE2® project and stage plans

Project budget

BUDGET FOR: NEW CUSTOMER CONTACT SYSTEM									
Expenditure code and heading		Monthly figures							Totals
		Mar	Apr	May	Jun	Jul	Aug	Sep	
A	Direct labour	50	50	70	90	120	70	30	480
B	Subcontract work		30	30	60	60	30		210
C	Hardware	100				200			300
D	Software	30				60			90
E	Telecommunications	10				60			70
F	Travel	3	3	1	1	3	2	1	14
G	Accommodation and subsistence	2	2	1	1	2	2	1	11
H	Project-specific training	10							10
I	Support services					2	6	5	13
J	Consultancy support	2	2	2	2	6	2	1	17
Contingency (10%) – items B–J only		*16*	*4*	*3*	*6*	*39*	*4*	*1*	*74*
Monthly totals:		207	87	104	154	513	112	38	1289

Figure 10.13 Example budget for an IT project