Turner, J. Rodney
The Handbook of Project-Based Management
Improving the Process for Achieving Strategic Objectives
McGraw-Hill Companies, London, 1999:445-481
ISBN 0-07-709161-2

18 Applications of project-based management

18.1 Introduction

In this last part of the book, I deal with applications of project-based management, to illustrate the use of project management in different circumstances. I said in Section 14.4, that the approach to project management needed to be adapted to the type of project. In Section 1.6, I introduced the goals and methods matrix, and showed how projects need a different emphasis in their management depending on how well defined are the goals and methods of delivering the goals. In Section 14.4, I primarily concentrated on the size of the project and the type of resource working on the project, although the latter was related back to the goals and methods matrix. I also introduced degrees of distance as a way of classifying projects. In the first edition of this book, I classified projects according to the stage of the product life cycle at which they occur, and according to the industry or sector in which they arose, and devoted a chapter to each. The latter, however, is not a strong determinant of project type, with the resource type, linked to the goals and methods matrix, being more significant.

Hence, what I do here is discuss the product life cycle, and the different types of project that arise at different stages. I then discuss several different types of project, to illustrate different needs and approaches. I consider product development projects, research projects, concurrent engineering, information systems projects and business process re-engineering projects. In Chapter 19, I discuss international projects, and managing projects from different cultures.

18.2 Managing the product life cycle

Several versions of the project life cycle set the project into a wider view of the life cycle of the product the project produces. Wearne[1] proposed a model (Figure 18.1), which is essentially a life cycle of the facility, and is reminiscent of the problem-solving cycle (Figure 1.6). It starts with a survey of demand for the product produced by the facility. That part of the cycle on or within the circumference of the circle describes the life of the facility built by the project. The six steps from study to commissioning relate to the four steps of the life cycle used in Part Three. The next three steps extend the life beyond the project to use of the facility, its maintenance, and monitoring of its performance.

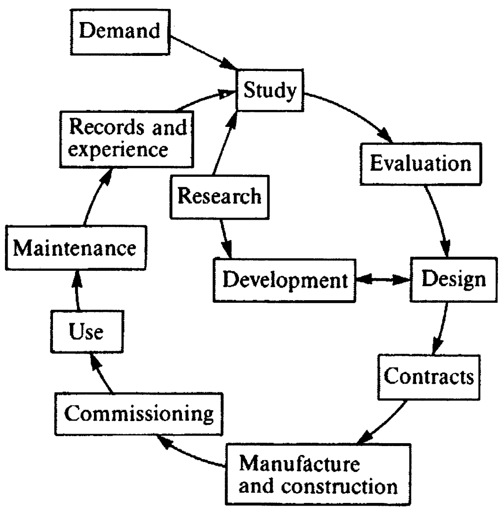

Figure 18.1 Wearne's life cycle for industrial projects

Kerzner[2] proposed a model addressing the life cycle of the product produced by the facility. It is the classic marketing view[3] (Figure 18.2). This is the view of projects filling the planning gap (Figure 2.5 (a)), and draws very little distinction between the project and the product. Some people differentiate between the project and the product life cycles, saying the project is the period up to and including commercialization in Figure 18.2, and the product life is the period from introduction of the product until its decline.[4] The project period can last anything from three months in the electronics industry, to ten years in the pharmaceutical industry, to 200 years for the Channel Tunnel.

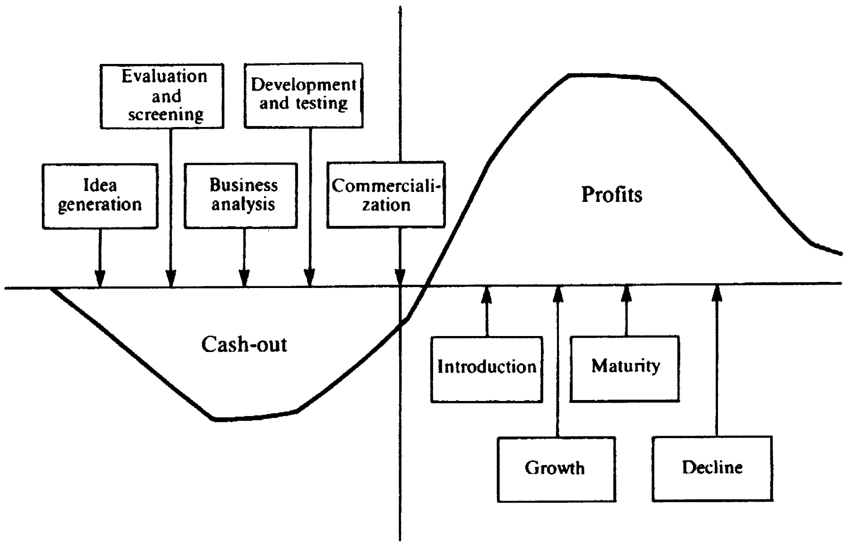

Figure 18.2 Classic marketing view of the life cycle of a product

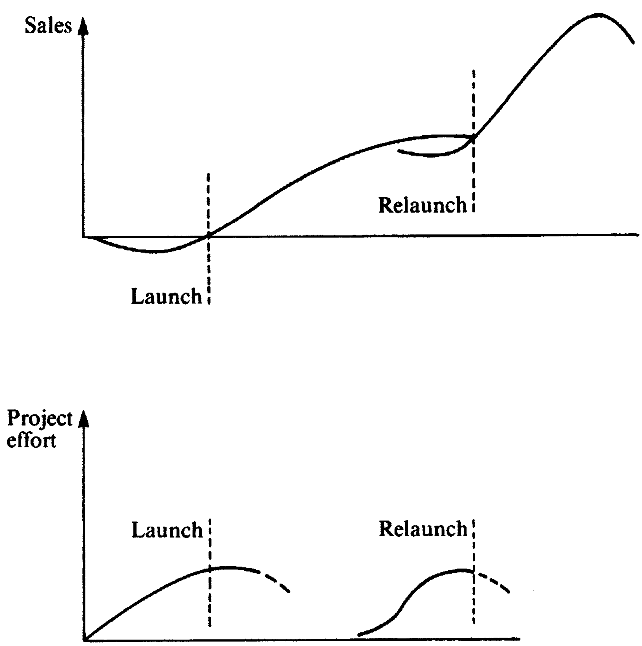

Figure 18.3 Product relaunch

Implied in Figures 18.1 to 18.3 is an assumption that the facility is an engineering plant that will make a product (or perhaps provide a service through its operation, as with the Channel Tunnel). In this book, I have taken a wider view of projects and the facilities they deliver. As well as an engineering plant, the facility may be a computer system, a design, trained managers, a set of procedures. With this view, projects can occur at any step in the product life cycle. There are projects to conduct a marketing survey, research and development projects, and maintenance projects. From the marketing model, there are projects to launch the new product at market introduction, and to relaunch the product at deterioration (Figure 18.3). Projects therefore occur throughout the product life cycle, or at any stage in the strategic development of organizations (see Section 2.2).

In this chapter, I shall examine several types of project arising directly from the product life cycle, which are common for many industries.

18.3 New product development[a]

Let us start with new product development, which in turn can lead to many types of project, including:

- research and development
- product design
- concurrent engineering
- facility design and construction
- product launch.

In this section, we shall describe how organizations create a culture for innovation, the types of organization they adopt for new product development, and how they plan and control the process.

Organization, people and culture

New product development has a key role to play in organizational competitiveness, yet it is one of the most difficult aspects of the company to manage. Organizations which choose in-house development must create a climate which favours innovation. Top management have a key role in this process, to encourage the establishment of a creative environment, which has three key components:

CLIMATE FOR INNOVATION

The innovative climate of an organization and its development policies are inseparable. Product development demands a flexible structure which encourages creativity, entrepreneurship and provides necessary conditions which favour development. However, there can be many pressures within an organization which act to hinder enterprise, and encourage bureaucratic policies and procedures which constrain change. Many of these were identified in Chapter 3.

INNOVATIVE ORGANIZATION

In order to harness innovation, organizations must be versatile and adaptable in their approach to their circumstances.[5] In essence product development is at its best in organizations which encourage imagination and are organic in nature, rather than those with bureaucratic structures based on routine management processes.

INDIVIDUAL INNOVATION

Whether bureaucratic or organic, organizations consist of people, whose personalities and performance directly affect the success of projects and overall business performance. Thus, organizations need to adopt structures which harness individual innovation. This will be reflected in recruitment and selection procedures, opportunities for development, removal of bureaucratic restraint, and rewards to innovators. It is not possible to prescribe the definitive organizational structure to achieve this; much depends on the company's response to its environment.

Product development organization

In a climate which welcomes creativity, the marketing function has two distinct roles:

1. *Routine, operational marketing tasks*: demanding a structure based on routine activities, planning, coordination of the marketing mix for products which form part of the existing product line.
2. *Novel projects*: requiring less defined structure. New product development projects operate in uncertain conditions, and though planned require freedom from routine organization.

In order to implement in-house product development, the first problem is to find the right organizational format. By nature, innovation is individualistic, requiring each company to develop their own working arrangements. There are several ways in which a business can organize itself for product development

NEW PRODUCT COMMITTEES

These are senior committees meeting on a continual or ad-hoc basis, responsible for coordinating product development. Members are senior functional managers and executives from research, marketing, finance, production, engineering, etc. The principal responsibilities include reviewing and screening proposals, determining policy, planning and coordination. Often the committee is considered to be the coordinating function which ensures the product maintains its momentum and controls the activities of the multi-functional team developing the product.

PRODUCT MANAGERS

Product managers may be given the responsibility for developing new products alongside their routine duties of managing existing product lines. There are several economies associated with this. In addition to monetary benefits, product managers may assume this responsibility as they are sympathetic to the customer requirements and considered to be in the best position to ensure synergy with the existing product portfolio. The disadvantages are that additional management time required may not be forthcoming, nor can the product managers give this unique activity the specialized attention, resources and expertise required while maintaining responsibility for routine activities.

NEW PRODUCT MANAGERS

They are given overall responsibility for product development from planning to implementation. Often the new product manager works alongside existing product managers, but without their operational responsibility, and can thus turn their attention to the creative role and generate practical new product ideas. Although the establishment of a new product manager formalizes the product development role, there are strong links with existing product lines, leading to minor changes, rather than independent, novel or radical innovations.

NEW PRODUCT DEPARTMENTS

These are common in large organizations, take a high profile, working alongside new product managers in generating ideas, and evaluating their feasibility. In contrast to other methods, new product departments place the responsibility with a senior manager. The department provides the umbrella for coordination of various functions for continuous project management. It does not have responsibility for operational duties so may dedicate its efforts to producing quality new products. Sometimes a new product department may be situated within a larger department, such as planning, marketing, research and development, projects or engineering.

VENTURE TEAMS

These are composed of functional specialists working to a closely defined brief, and generally recruited on an ad-hoc basis for a short time. While located in the team, the individuals are removed from day-to-day activities. The team ideally would report to a non-operating executive.

TASK FORCES

These groups are organized on an ad-hoc basis. Members are seconded from operational duties for the duration of a project, or divide their time between routine activities and project work. The aim of task force management is to ensure continued support from the functions throughout a project. As a project reaches the latter stages, task forces may recruit more members with specialist skills.

PROJECT-BASED PRODUCT DEVELOPMENT

Product development involves individuals with specialist skills, from various functions and from managerial levels. The formation of project teams can be effective in solving problems and creating benefits which cannot be achieved in routine ways.[5] However, one particular structure may not be appropriate at all stages of the project. Just as the activity needs to be fluid and flexible, the perfect organization must also adapt to accommodate the different expertise needed throughout the project.

New product planning

The project teams have primary responsibility for the new product planning aspects of the market strategy. The nature of product development creates several planning problems, as projects range from modest expenditure to major investments, combined with indeterminate time constraints incompatible with routine reporting cycles. The diverse activities involved in new product programmes should move through a logical sequence of events. Though considered contrary to flexibility and creativity, development plans are necessary as they help determine critical components of the project. The sequence (project life cycle) suggested by Kotler[3] is often used to illustrate the new product planning process (Figure 18.4).

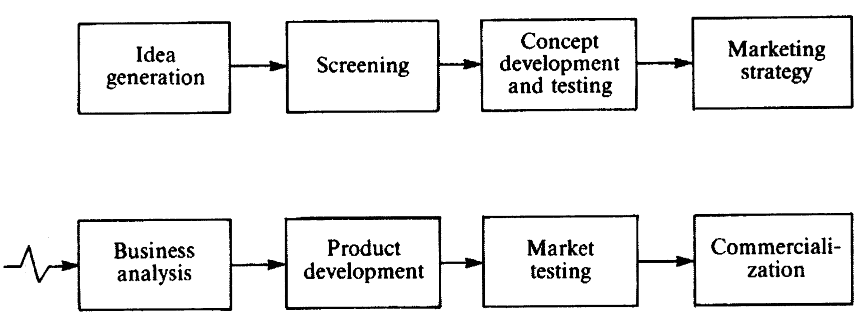

Figure 18.4 New product planning process (project management life cycle)

Plans should be used to enhance rather than hinder the development process. Management should not be limited by this logical progression. The sequence outlined is a guideline to help development, not constrain it. Idea generation, for example, does not always automatically occur as part of the formal planning sequence. Ideas may be initiated by users or employees during normal work (Figure 18.5). Similarly, product development does not always require radical change. Projects may be initiated to modify existing product lines (Figure 18.6). The project may also have several stages running simultaneously (Figure 18.7).

The planning process so far has not established links with business purpose or corporate strategy. Although not part of the routine of the company, project plans should be fully integrated into the strategic plans (Section 2.2). Product development should be complementary to existing products and meet the needs of the product portfolio against market

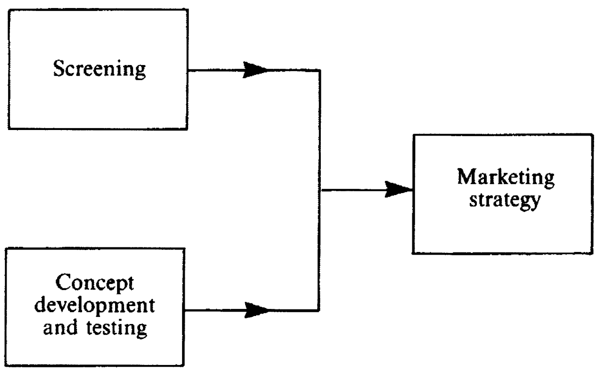

Figure 18.5 Revised sequence for ideas generated by users and employers

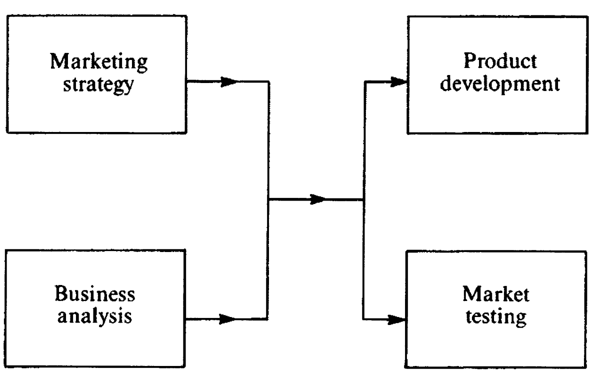

Figure 18.6 Revised sequence for products not requiring radical change

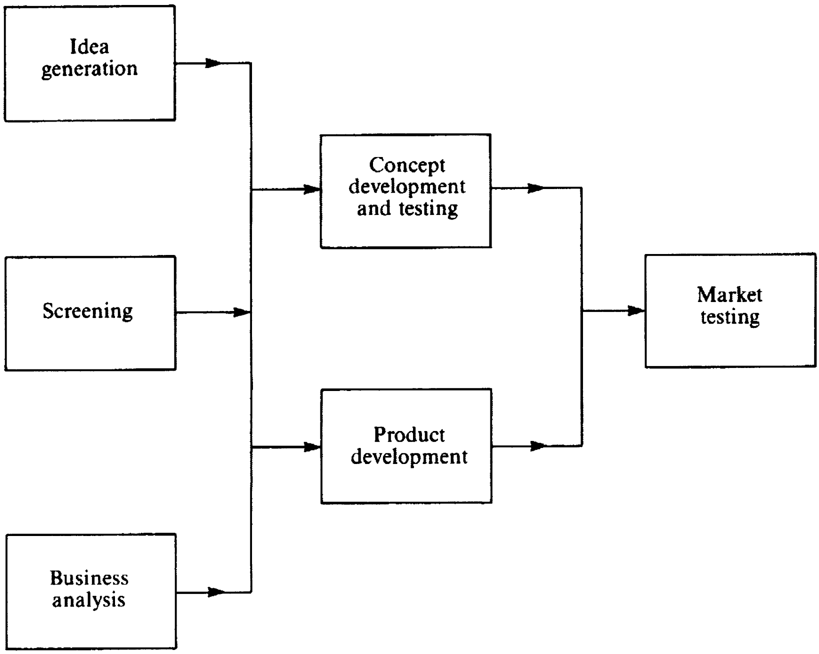

Figure 18.7 Sequence with several simultaneous stages

demands. New products provide an important strategic capability for achieving corporate and business objectives.[6] Strategic issues should direct and influence the new product project in three ways: strategic focus; technical criteria; and market acceptance.[7] Handscome[4] proposes a revised product development planning process which combines strategic focus with the need to combine phases of new product development (Figure 18.8).

Controlling new product development

The control function is an important aspect of product development. The application of marketing control systems to a new product development process reduces the risk. Control processes should therefore be integrated into all aspects of the plan and linked to critical components mentioned earlier. A continuous monitoring programme provides project teams with valuable information which may determine the successful outcome of projects. The key of any system is the extent to which it allows the manager to influence the success or otherwise of the outcome of the venture.[8] Several planning and control techniques may be used to monitor new product projects. Handscome[4] illustrates how these methods may be combined to monitor progress based on project objectives (Figure 18.9).

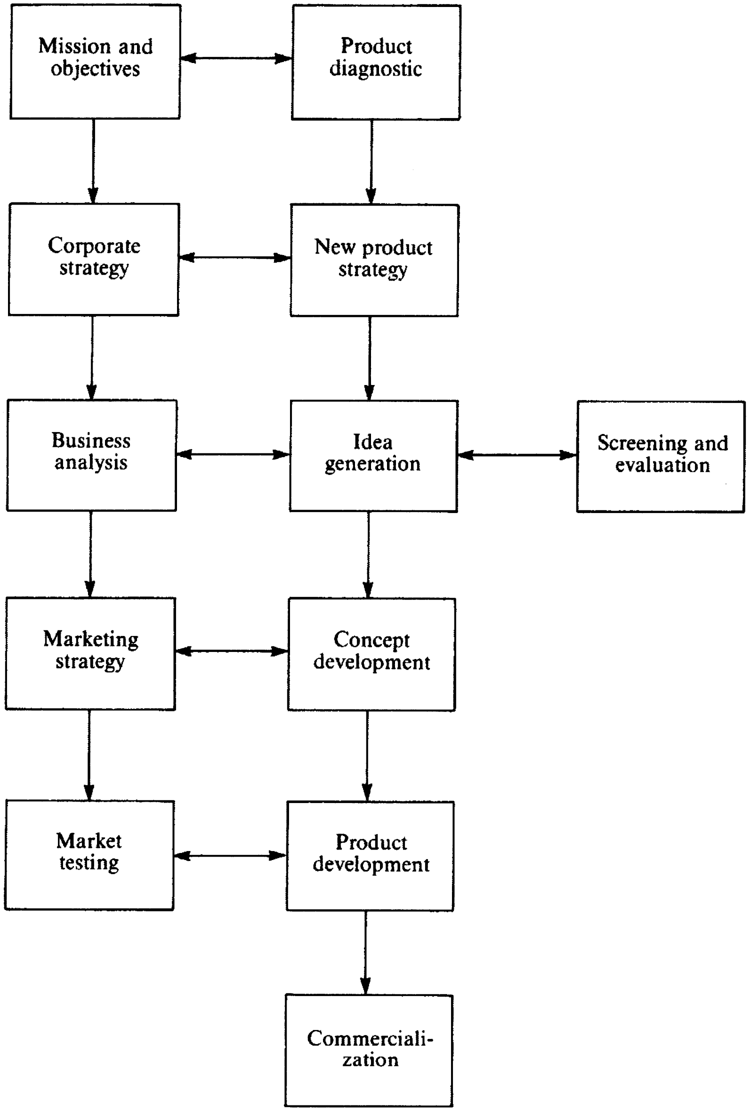

Figure 18.8 Revised planning sequence incorporating strategic focus

18.4 Technological projects[b]

Let us now turn to the management of projects in industrial and technological research and development (R&D). The discussion is relevant to process development as well as to product development. In this section, I focus on those aspects of project management within the R&D context

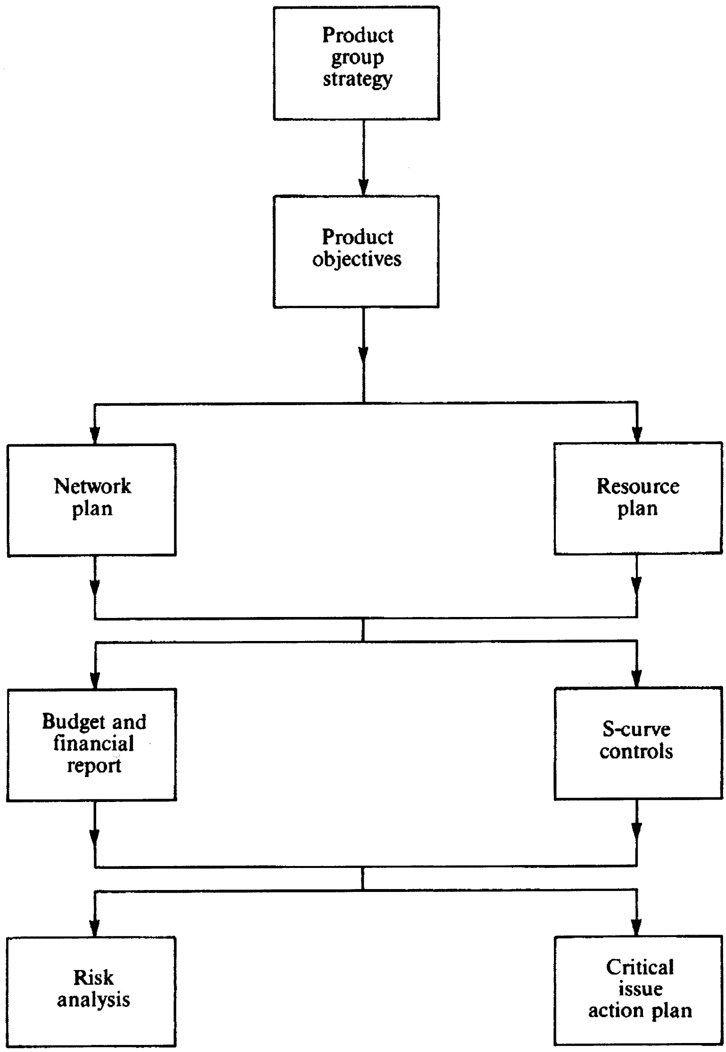

Figure 18.9 Schema for a hierarchy of plans for new product development

which poses specific problems, emphasizing those features, while accepting that most of the principles of project management can be applied in the R&D environment. Well-recognized difficulties of building teams of collaborative scientists, of managing such specialists and of communicating with other functions in the corporation are discussed, together with more positive notes for management. I shall describe what project management means in practice for industrial R&D, so that the non-R&D project manager can understand the requirements when liaising with R&D projects, and R&D managers can put project management principles into the context of their work environment.

The difference between research and development should also be recognized (respectively type 4 projects and type 2 projects in Figure 1.15). Different skills are required for exploratory research than for focused development. The definition of rigid time scale and cost constraints has less relevance in research. However, in the corporate world, research and development are often managed together as one unit, and so to outsiders appear to have similar management issues. In this section, I focus primarily on development, but indicate where generalizations can be made.

Project evaluation and selection

Project selection for research or development can be formal or informal, qualitative or quantitative. Rarely is it clear that a project must be undertaken, and even more rarely clear why that project should be done. There are several choices, and a lack of data allowing rational selection between them. This is one feature of R&D. Even with strong and clear business objectives, the direction of R&D effort often carries a great deal of discretion. There is a lot of guesswork involved in evaluating options, and this may tempt decision makers to rely on gut feel, or do something because it seems interesting or exciting. Hence, there is always some uncertainty in R&D; both in selecting the 'right' projects and in evaluating the chances of success.

One criterion which is often handled badly for the business is time scales. Often the development programme is slowed down, apparently to contain costs. This may have a disastrous effect on the ability of the company to compete in a rapidly changing market, where it is important to be first with a new product, or a swift second with a product modification. Speedy analysis of the risk issues at an early stage can allow more valid selection of major projects for fast tracking.

Important questions for the effectiveness of R&D are: 'Where have the proposals come from?' 'What is the push or pull?' Ideally, communications between marketing, sales, operations and development are close and open enough for productive cross-fertilization to occur, so that market pull and technology push are well balanced. Such open stimulation across functions must be a goal of the technology-based company. Ideas should also come from all levels within the organization, and this only happens if it is encouraged actively and openly, and if mental risk taking is rewarded. There should be some formal or informal flexibility for preliminary evaluation of ideas from outside the formal project control system.

Project evaluation needs to be based on a strong awareness of business objectives and strategy (Section 2.2). A research or development team can only be effective in adding value if it knows where the business is going. This sounds obvious. Unfortunately in practice it is too common for the scientist or engineer to be blissfully unaware of whether or how their activity is contributing to the business. In addition, the industrial R&D organization should have a technology strategy. This should recognize the nature of enabling technologies which are currently the foundation of the business, and emerging technologies which will produce the opportunities or threats in the future. It should provide a consistent statement of how the organization will access the required skills; and how it will use these skills to achieve business results. Surprisingly few organizations have technology strategies which are expressed in these terms to guide project selection. The organization cannot afford to have a prescriptive strategy or approach which imposes blinkers on visionary or radical thinking. Flexibility is an important key, but the strategy should provide the framework on which all decision processes are initially considered. Any decision to break with established guidelines can then be seen for what it is and may indeed prompt reconsideration of the guidelines.

It is clear that some kind of formal evaluation procedure is desirable. This may use an equation in which key parameters are estimated. It should also allow a statement of coherence with, or divergence from, accepted objectives. Even when the decision to proceed with a project can be taken internally in R&D, it is wise to seek support from the operations and marketing functions wherever possible, even for a research project which is a long way from the market place. Research managers are often accused of being insular, and of marching to their own tune, not bothering to communicate with other functions. This works both ways: research managers can be exacerbated by the different languages spoken by functional managers. However, it is a powerful aid to the research manager to have support and understanding from those who will make or sell the product, once successfully developed.

Selection of the project manager

The project manager may or may not be a line manager in the organization. Many R&D organizations operate matrix management systems (see Sections 3.4 and 6.2). Multiple skills are often required for a technology development, and these skill requirements may cross disciplinary boundaries. It is common for physicists, electronics and systems engineers, and bio-technologists to work together on a project. The managers of these teams must be able to coordinate resources whose skills they appreciate but may not understand. They need to be as technically aware as possible, to have a strong sense of the true project objectives, and to have strong people management skills in order to be able to manage diverse individual objectives. In a radical matrix structure, the project manager may not be a line manager, and may have line managers as part of the team. This works well where it is an accepted part of the culture and where the project management line is key to operational success. It is difficult, though, to operate in a more formal, hierarchical organization.

An important feature of the leader in a research project, and to a lesser extent in a development project, is the ability to contribute creatively. Adventurous thinking is required from the team, and this may be reinforced or discouraged by the manager. The manager must be able to use networks of contacts, both internally in the company and externally, to access information and resources, and should have a strong ability to communicate at all levels within the company, and across functional boundaries. The promotion of the excellent scientist to a management position may be disastrous. Creative bench scientists may be lousy managers, so the organization sometimes loses brilliant doers, putting them in positions where they actually do harm if they do not know how to motivate their team. The need to separate rank and role, allowing experienced scientists to continue to contribute to technical development while developing their status and career, is a problem in many organizations.

Project planning

R&D projects are difficult, if not impossible, to plan in detail. The very process of the project is to determine how to achieve the objectives. Therefore you can only plan at the integrative and strategic levels. Indeed, you tend to plan the project to a series of gateways, through which the project must pass by a certain time or fail. Some people say you cannot hurry creativity. That may be the case. But if creativity does not occur by a certain time, to allow the project to pass through the gateway, then it need not occur at all. The plan will use as its starting point schemes which have been produced during proposal and initiation. Ideally the project manager will have been involved in that process and will be familiar with the objectives and rationale. A milestone plan should be developed to indicate intermediate control points against intermediate deliverables, and also the gateways. The planning process may involve some definition of time scales and costs to achieve the milestones. However, these may not be developed from any estimating process, but may just be constraints imposed by market conditions. If milestones are not achieved by a certain date and for a certain cost, the product will not be worth while. Some kind of critical path analysis may be desirable for large projects, which may also require use of project planning software. Project managers learn which packages work best for them in which situations, and selections by different individuals for the same project might vary from a computer-based package, to the simple paper and pencil bar chart.

Whatever planning tool is used, and they all have strengths and weaknesses, an issue for effective management is how strictly deadlines and critical paths are treated. The wise handling of plans is perhaps the most demanding skill required of project managers. This may require them to:

- optimize motivation and productivity
- maintain momentum
- control waste
- prevent divergence

but also may require them to allow divergence down a more promising path; or accept delay a promising route suffers an unexpected set-back.

You may say that this is true of any project, but it is most critically needed in R&D, where components of the critical paths may simply not be physically possible to achieve, and other approaches may lead to new unpredictable possibilities. By its very nature, the uncertainty of R&D requires *more* of this skill than most other project types.

Access to information may be an issue for planning and administration. Information systems are required to analyse previous work on similar systems, both inside and outside the company, and to assess the relevance of new ideas. Ideally project managers should have sophisticated data sources at their disposal and should access their networks of informal contacts. From the internal administrative viewpoint, they will need some kind of management information system. This may use sophisticated software or manual paperwork or word of mouth. Working against this is an attitude which pervades the R&D environment, that all paperwork and bureaucracy is a dreadful constraint on creativity.

The planning of the scope time and cost is, of course, only part of the story. Selecting the project team members is another. In many R&D organizations this will be quite simple in that there may only be one person with each of the required skills. This assumes that an analysis of skills required has been done. Often, little consideration is given to this, which means that only the obvious sources of skills are considered. Project launch workshops (discussed in Chapter 11) overcome this, especially if people with a wide range of skills are invited, thereby introducing new views. For a large organization some kind of skills database may be needed. These are rarely well-constructed, and rarely used by project managers, who are more likely to ask for resources from people they have worked with before than from people they do not know. In a large organization a good relational database of skills can be invaluable for strategic skills analysis and rational skills sourcing. These should be well thought out, with key word connections underlying free text skill description, building on technical classifications relevant to the company's market focus.

The team should always include people who will take the product through to the next stages of development (production or marketing), even if they cannot contribute directly to the current technical issues. However, it is sometimes difficult to obtain their commitment. The manager must achieve that by involving them in the planning process.

Research and development scientists often work on several projects at a time. There is some experimental evidence,[9] which suggests that most scientists and engineers are more productive when working on several projects, with peak effectiveness at about three projects. The rationale is that multiple contacts are more stimulating, and that there is less risk of becoming stale or blinkered if tackling several issues. There are, of course, exceptions, and one meets individuals who give their best only in total dedication to one project at a time. Whether it is more appropriate to have a large, self-sufficient project team, or a core team who can call on additional resources as necessary depends on the organization structure and culture.

Building the project team and getting results

A technological R&D team includes people with diverse skills, working across disciplines and several projects. Industrial scientists are usually specialists, often with an academic research career behind them which they still use as the basis for their approach. They are often uncommunicative, preferring to explore ideas alone. This may be for fear of failure, a wish not to share success, or because the culture encourages this, or perhaps because of the kind of people they are. For whatever reasons, many project managers in R&D have trouble building coherent teams, rather than a set of individuals working separately on the same problem. Lack of management skills in the leader exacerbate this. The extent of team working depends very much on the culture of the organization, departments and groups, and on the personalities concerned. Standard management and people development skills are, of course, valid, although they may need targeted interpretation in this culture, as they would in any other (see Chapters 3 and 17). Administration and bureaucracy are almost universally seen as drains and dampeners of creativity. Imposition of unnecessary paperwork of any sort by the project manager may be the worst thing they could do in the eyes of the team!

Communication needs

Communication is a high priority to ensure that new ideas are shared as early as possible. There is also a strong need for communication outside the team. Again, this is true of all project types, but in R&D it is often ignored.

There is a danger that the technical team may not feel that it is necessary to communicate with others until a breakthrough is achieved, or the project is successfully completed. Communication is all important for continuity, validity and congruent relationships, and may be:

- upwards, with senior management
- outwards, with functions who will take the successful development through production and sales
- sideways, with other R&D managers who may not be involved at all.

R&D managers should see information sharing as an important part of their job. This is rarely taken seriously and is the cause of many unhappy endings to otherwise successful projects. The challenge of communicating across language and vocabulary barriers is often disabling, and the onus is really on project managers to put their ideas and results across in English, in a way which is understood by others outside their speciality.

Project review and follow-up

The project must be reviewed regularly, and progress judged against the plan and other initiatives. It follows from the comments made in the last subsection that representatives of other functions should participate, as should senior managers or decision makers, and ideally also one or more scientist with different perspectives or background. One important requirement is that such review processes should not become unwieldy. Although it may be more efficient to hold a series of reviews of all projects in one day, better results are achieved if each is considered separately by fresh minds, unjaded by a series of apparently similar discussions.

When the project has been completed, and the product is moving from research into development and then to manufacture, it is important to have continuity. There should be representatives of the follow-up functions on the project team, and ideally the project manager will continue to be part of the team (if not managing it) in the next stage. They are then the product champion, and so commit themselves more deeply to the programme.

Effectiveness vs efficiency

A theme running through this section is that there is often more concern about efficiency than effectiveness in the management of R&D. Effectiveness is doing the right things, well; efficiency is doing what you have decided to do, as quickly or for as little cost as possible. In order for industrial R&D to add maximum value to the business, it must be effective as well as efficient. Results and business success are not correlated with R&D spend in technology-based companies, nor are they correlated with productivity (efficiency), unless the objectives are wise. They are correlated with effectiveness, in the sense of flexible and top-quality thinking and appropriate actions. Since projects are the unit of operation for the R&D department, effective project management is the key to ensuring that R&D adds. You should demand flexibility and quality in approach, thinking, communication and practical experimentation. This means that the organization must have coherent, clear objectives which are shared with everyone (a clear mission). It must have a high skill intensity in its resources, and it must invest in their development through wise recruitment and training. The structure and systems of the organization must support innovation, and there should be formal and informal systems to encourage it. Lastly, very nebulous but very important, the culture must be congruent with objectives, standards and expectations, with the working environment supporting these. An attitude which says that we, the team, can win, and leadership which reinforces this expectation, can ensure success against all odds.

18.5 Concurrent engineering[c]

Figures 18.1, 18.2 and 18.4 show the product development process taking place sequentially. Traditionally, product development took place as a relay race: research, followed by development, followed by product engineering and prototyping, followed by production process engineering. Product development processes became artificially extended, not only as it was insisted that one step was finished before the next started, but inevitably there was a delay between one step and the next. Concurrent engineering attempts to overcome the built in delays by running the product development process as a rugby front line, with the steps in parallel. The concept was first adopted in the development of fast-moving consumer goods as early as the late 1970s, but became widely adopted across a range of industries in the 1980s and 1990s. Concurrent engineering is a systematic approach to the integrated concurrent design of products and their related processes, including manufacture and logistics support. This approach is now used in areas other than manufacturing, including construction and business process re-engineering projects. The objectives of concurrent engineering are to achieve:

- decreased product development times and hence earlier time to market
- improved profitability and competitiveness
- greater control of design and development
- reduction in product costs
- improved product quality.

Requirements of concurrent engineering

Several changes are required within the organization and in its approach to projects, in order to allow this to happen:

A CHANGE IN THE ORGANIZATIONAL CULTURE

A shift is needed to the flatter, more flexible approaches of project-based management. There needs to be decentralization of authority, with managers empowered to take decisions without referring them up the line, which builds in delay. However, this environment creates an almost greater requirement for senior management support, to show their faith and support for the product development process.

CROSS-FUNCTIONAL TEAM WORKING

The use of cross-functional teams is inherent to concurrent engineering. It is the only way to achieve the necessary parallel development. This requires people to communicate freely across the functional hierarchy, demonstrating the need for the empowerment and support of managers mentioned above. It also requires teams to work closely with suppliers since their development must take place in parallel. This may require partnering or integrated supply chain management. It also requires advanced contracting methods to allow contracts to be signed with suppliers long before the closure of design.

USE OF TECHNOLOGY

Concurrent engineering only really becomes possible with the use of modern information systems. This includes the use of computer-aided design, engineering and manufacture systems (CAD, CAE and CAM), to aid design integration through shared product and process models and databases. It also requires the use of project management information systems to coordinate the work of the people involved, and to manage the data involved. Configuration management becomes a significant element of concurrent engineering and so the PMIS must be able to perform the status accounting involved.

TECHNIQUES

Concurrent engineering requires the extensive use of iterative working techniques to develop all the aspects of the product, process and logistics design simultaneously.

Risks and pitfalls of concurrent engineering

There are some risks and pitfalls in the concurrent engineering approach:

ATTITUDES OF MIDDLE MANAGEMENT

There may be resistance from middle management. Not only does the cross-functional working threaten their influence within the organization, the process initially increases their costs. Although the increased costs will be repaid through the earlier completion times, managers can see an early fall off in the profitability of their departments, and hence a reduction in their bonus in the early years. When told that the higher costs associated with faster development times will be repaid through increased sales over a three-year period, managers may say that they are only in post for two years, and hence do not get the enhanced payback within their period of tenure of office.[10]

AUTHORIZATION OF THE CONCURRENT ENGINEERING PROJECT

The first problems to be overcome are those of obtaining sanction for the project and for executing it on a concurrent engineering basis. If this has been done before it should not be a problem, otherwise it may be a long, hard battle with all of the organization and departments involved.

The project definition stage is of paramount importance in this respect as it has to satisfy the following major criteria prior to project authorization and major commitment:

1. The product must be within the organization's aims and objectives.
2. The market need for the product must be established beyond doubt.
3. The supply of raw materials must be shown to be secure.
4. The design of the product must be carried out to a sufficient level to establish its feasibility (if necessary including models and/or prototypes).
5. The design of the product manufacturing system and its associated support systems must be evaluated to an acceptable level. They must be within the organization's intended capability.
6. Economic and financial evaluations must show the product to be viable bearing in mind the predicted life cycle, development and production costs.

It is usual during project definition to survey the industry, benchmarking to obtain typical implementation costs for similar products. It may also be possible to use the organization's standard investment appraisal techniques, but these do not always take account of combined development and implementation stages (which are conducted in parallel with concurrent engineering).

The risk associated with authorization of a concurrent engineering project is significant as this is an all-or-nothing approach that commits the organization to prosecute the project to completion. (The only factors to prevent its continuation after authorization would be due to external items such as a dramatic market shift.) In authorizing a concurrent engineering project, senior management must be seen to give both the project and the approach their full support. They are not only authorizing a technical development, they are authorizing radical change and way of life in their organization.

ORGANIZATIONAL AND CULTURAL CHANGE

The adoption of a concurrent engineering policy will inevitably involve significant changes to an organization and its culture. Some of the more important of these may be:

- departmental organization shift to project orientation
- conventional project-oriented organization shift to concurrent project organization
- move out of deep hierarchical structure into shallow, multidiscipline teams
- new organizational reporting structures
- change to business practices and procedures
- establishing long-term customer/supplier relationships and elimination of counter-productive competitive tendering policies
- reorientation of accounting policies away from departments and towards projects.

The importance of these changes will be determined by many factors including:

- the degree of support from senior management
- the existing organization's size and culture
- resistance from established functions
- degree of product novelty and complexity
- difficulties in implementation.

There may be conflict between those who are charged with implementing the concurrent engineering policy and others in the organization. There will almost inevitably be culture clashes between those involved with the concurrent engineering project and those in the rest of the organization (similar to those experienced when projects are set up within functional/departmental-oriented organizations).

MANAGING INTERFACES

There will be many interfaces to be managed including those between:

- management and design
- commercial considerations and design

- suppliers and design
- the various design functions within the concurrent engineering design team
- new product production and support facilities and those of existing products.

Careful selection of the concurrent engineering team, its working procedures and the control facilities employed ensure that these are managed effectively.

TECHNICAL MANAGEMENT

The most important function to control is that of design as this largely determines how a product is to be made or implemented and its associated costs. The designer is often only limited by a relatively few critical constraints, but his work may have great impact on the work of others and on downstream costs. The following aspects of the project are identified at an early date and monitored closely:

- *differentiation*: where there are linkages between highly differentiated departments
- *cross-functional requirements*: where there is a need to take account of the requirements of the other function, particularly those downstream in the development process
- *uncertainty*: where there is a high level of uncertainty in the use, interpretation or content of data
- *intensity and frequency of two-way flow*: where there are major feedback requirements between departments or functions
- *complexity*: where there is a need to liaise between groups because of the complexity of the product or task.

Standardization policies help to ensure conformity, and the extensive use of common electronic data management tools helps to keep all parties working to the same model and standards.

The early development of prototypes and prototype testing is a powerful tool used extensively in concurrent engineering. Its major value is that of identifying and forcing problems out into the open at an early stage. These can then be solved before they become too serious.

COST CONTROL AND RELEASE OF FINANCE

The implementation of concurrent engineering requires a significant departure from conventional financial release and cost control methods as it involves:

- initial release of greater funds on more preliminary information in the early stages

– negotiation of less well-defined contracts with suppliers or contractors who are to assist in the design process
– commitment of greater funding for production facilities at an early stage when parameters are not well defined.

The chief departure from conventional methods is the acceptance of significant financial risk at an early stage and the greater requirement for an effective and continuous cost review procedure that gives early warning of possible cost risk areas. (Bear in mind that the rolling-wave approach to both planning and technical design will have a major impact on the increasing confidence in cost estimates.)

RISK CONTROLS

More stringent risk management procedures are required with concurrent engineering than with conventional developments. In particular, they have to operate across the whole range of project activities, including sales, marketing, personnel, production and support. They are required to impose a consistent risk approach on a continuous basis, covering all items that would otherwise be analysed at major stage reviews. As with conventional developments, they fall into the usual categories of:

– technical risk
– commercial risk
– financial risk
– time risk.

Most of the concurrent engineering techniques and procedures operate with the reduction of risk as a prime motive.

18.6 Information systems projects[d]

As software becomes more complex managers have a greater need to understand its production. Several models of the software life cycle have been developed to aid this process (the need for which is explained in Section 1.5). Many of the models are applicable to other areas of technology, and to R&D projects. The function of a life cycle model is to determine the order in which software development should be undertaken, and to establish transition criteria to progress from one stage to the next (see Section 15.3). Transition criteria include completion criteria for the current stage, and entry criteria for the next. More sophisticated models of software development life cycles have evolved because traditional models discouraged effective approaches to software development such as prototyping and software reuse. This section traces the evolution of the different models, and explains their strengths and weaknesses.

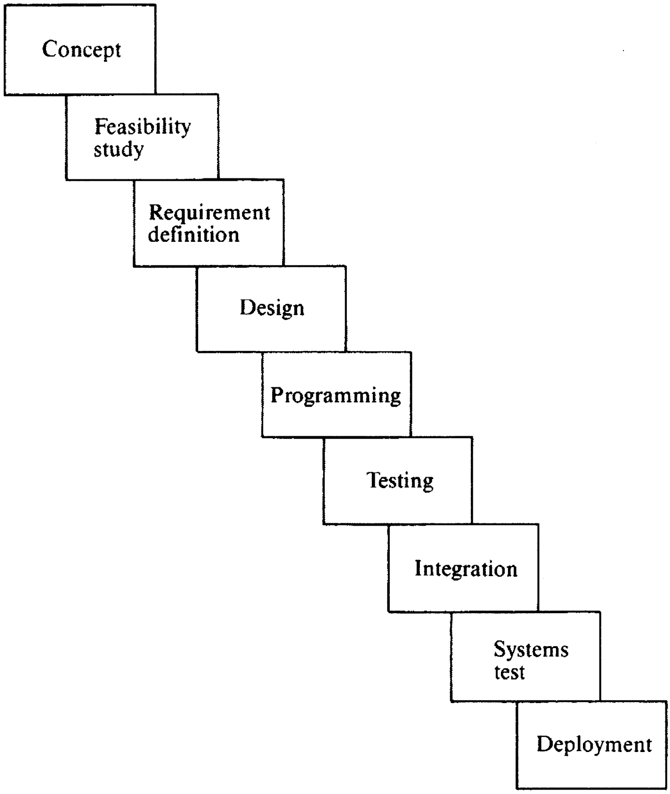

Figure 18.10 The stage-wise model

The code-and-fix model

The earliest model for software development had two, simple stages:

Stage 1: write some code.
Stage 2: fix the problems in the code.

Code was written before requirements were fully defined, design done and test and maintenance procedures described. The strength of this approach was its simplicity, but that is also the source of its weaknesses. There are three main difficulties:

1. *Maintainability*: after a number of fixes the code becomes so poorly structured that subsequent fixes are very expensive. This reinforces the need for design prior to coding.
2. *User requirements*: often the software is a poor match to users needs so it is either rejected or requires extensive redevelopment.

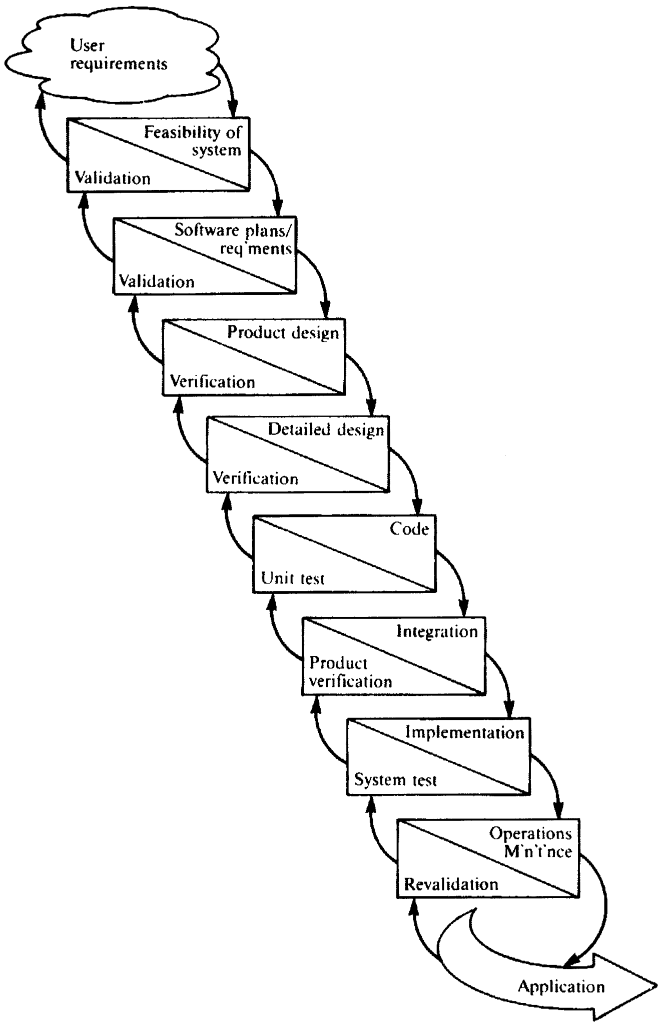

Figure 18.11 The waterfall model (1)

3. *Cost*: code is expensive to fix because of poor preparation for testing and modification. This highlights the need for these stages, as well as planning and preparation for them in early stages.

The stage-wise and waterfall models

Experience on large software systems as early as the mid-1950s led to the recognition of these problems, which resulted in the development of a stage-wise model. This stipulates that software should be developed in successive stages (Figure 18.10). The waterfall model (Figures 18.11 and 18.12), is a refinement of the stage-wise model from the late 1960s. The major enhancement was that it recognized feedback loops between stages, but with a requirement to confine loops back to the previous stage only, to minimize the expensive rework resulting from feedback over several stages. We shall use the second waterfall model (Figure 18.12) to illustrate principles common to many of the life cycles, as it can be easily related to other models, although the specific stages and names vary between models.

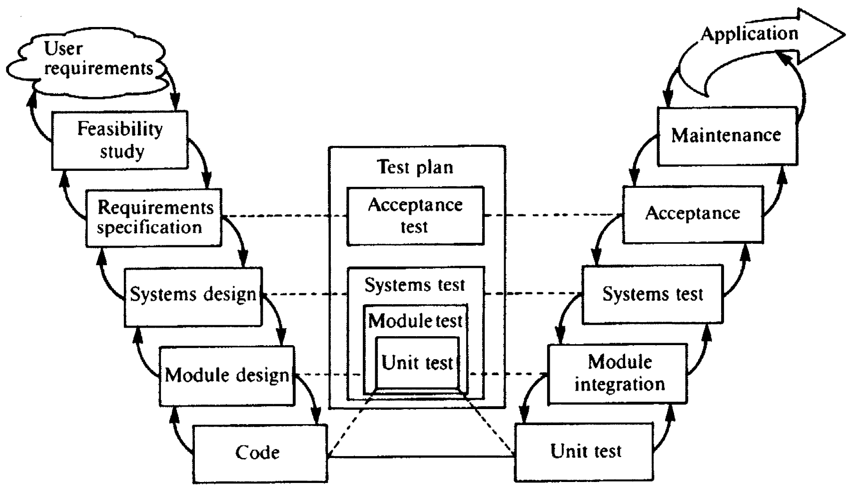

Figure 18.12 The waterfall model (2)

The second waterfall model is characterized by its V shape. Down the left-hand side are stages which derive elements of the system, while up the right-hand side is the delivery of the elements to form the system (Table 18.1). Each stage is defined by its outputs, the deliverable, rather than its constituent activities. A tangible output is the only criterion of progress, the only thing which people can assess objectively. Only in this way can the 95 per cent complete syndrome be avoided. The products of each stage represent points along the development path where there is a clear change of emphasis, where one viewpoint of the design or emerging system is established, and is used as the basis for the next. As such, these intermediate products are natural milestones of the development progression and offer objective visibility of that progression.

Table 18.1 Stages of the software development life cycle

Stage	*Description*
Feasibility study	Production of verified/validated system architecture based on a design study, including allocation of tasks to staff and machines, milestone plan, responsibility chart, schedules of major activities, and outline quality plan
Requirements specification	Production of complete/validated specification of requirements (functional/non-functional) the system must satisfy. Produced in close liaison with the end user. Means of system acceptance also agreed with end user
Systems design	Production of complete/verified specification of overall architecture, control structure and data structure for the system. Production of draft user manuals, and training and test plans for integration
Module design	Production of detailed designs for each module, together with module test plans. This may actually consist of more than one level of design
Code	Module designs are converted into code units in the target language (such as C, Pascal and FORTRAN)
Unit test	Code units are tested by the programmer. Errors are corrected immediately by the programmer. Once complete, code units are frozen and pass to integration
Module integration (structural testing)	Component units of a module are integrated together, and tested as specified in module test plan. Errors detected are formally documented, and the affected area returns to a stage where the error was introduced
Systems test (functional testing)	Modules are integrated together to form the system, and tested against the system test plan. Errors detected are handled as for module testing
Acceptance test	Client formally witness the exercising of the system against agreed criteria for acceptance
Maintenance	Service life is often grossly underestimated. Software written in the 1960s is still being used. The cost of development can be small compared to maintenance, but the latter is given little consideration.

To provide management control, the concepts of *baseline* and *configuration management* are introduced. The completion of a stage is determined by the satisfactory assessment of the quality of the intermediate products (or deliverables) of that stage. These deliverables then form the baseline for the work in the next stage. Thus the deliverables of the next stage can be verified against the previous baseline as part of configuration management and the quality assessment, before they become the new baseline. Each baseline is documented, and the quality assessment includes reviews of the intermediate products by development personnel, other project and company experts, and usually customer and user personnel. However, there the focus was primarily on baselining time and cost, whereas here it is on quality and scope. For these, you cannot baseline the whole project, only one stage at a time, as the definition of scope and quality evolve throughout the project. This evolution is controlled through configuration management (Section 7.4). It is the documentation and reviews which provide the tangible and objective milestones throughout the entire development process. The waterfall model shows how confidence in the project's progress is built on the successive baselines.

This simplistic description of the life cycle could imply that control of software development can only be achieved by rigorous control of the staging, so that no stage is considered complete until all prescribed documents have been completed to specified standards, and no stage can be started until all its input documents are complete (giving non-overlapping stages). Although the intended rigour of such an approach is commendable, it is unrealistic on a large development project. It is not intended that the life cycle should be interpreted in such a simplistic way.

The strengths of the waterfall model are that it overcomes the problems in the code-and-fix model. However, its great weakness is its emphasis on fully elaborated documentation as completion criteria for early stages. This is effective only for some specialist classes of software, such as compilers and operating systems. It does not work well for the majority of software, for example user applications and especially those involving interactive interfaces. Document-driven standards have pushed many projects to write elaborate specifications of poorly understood user interfaces and decision support functions, which have resulted in the design and development of large amounts of unusable code.

The spiral model

The spiral model[11] (Figure 18.13), which is still evolving, can accommodate all the previous models as special cases. The radial dimension represents the cumulative cost of undertaking the work to date. The angular dimension represents the progress of each cycle of the spiral. The model reflects the concept that each cycle involves a progression through a repeated sequence of steps for each portion of the product, and for each elaboration from overall concept document to coding of each individual program. Each loop of the spiral passes through four quadrants:

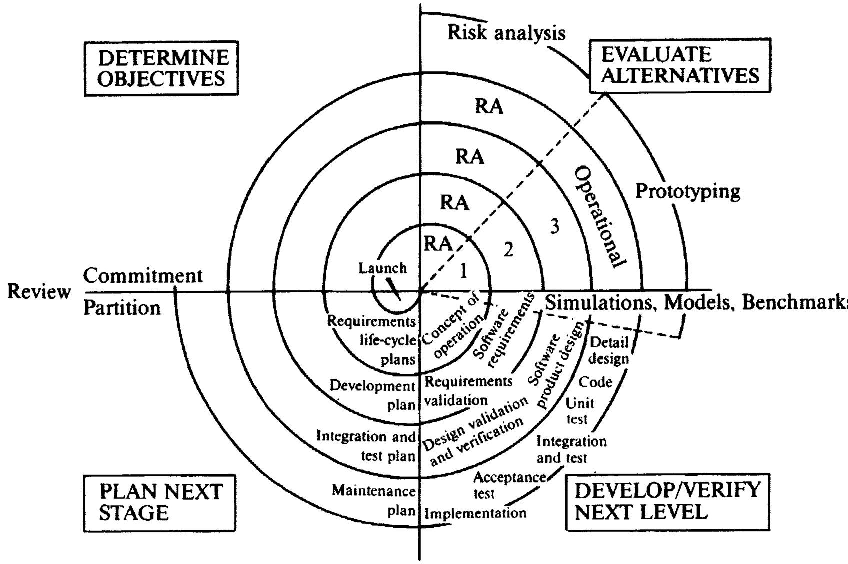

Figure 18.13 The spiral model

- determine objectives, alternatives and constraints
- evaluate alternatives, identify and resolve risks
- develop and verify the next level of product
- plan the next stage.

DETERMINE OBJECTIVES, ALTERNATIVES AND CONSTRAINTS

After planning and launching, each cycle begins with identification of:

- objectives of this portion of the product being set, including performance, functionality, ability to accommodate change, etc.
- alternative means of delivering this portion of the product, including alternative designs, reuse, or buying in
- the constraints imposed on final deliverable by the various alternatives, including cost, schedule, interfaces, etc.

EVALUATE, IDENTIFY AND RESOLVE RISKS

The next step is to evaluate the alternatives against the objectives and constraints. Frequently this process identifies areas of uncertainty which are significant sources of risk. If so, this stage should involve the formulation of a cost effective strategy for resolving the sources of risk.

DEVELOP AND VERIFY THE NEXT LEVEL OF PRODUCT

Once the risks are evaluated, the next stage is determined by the relative importance of remaining risks. This risk-driven basis of the spiral model, allows the model to accommodate any appropriate mixture of different approaches to software development including specification-oriented, prototype-oriented, simulation-oriented, transformation-oriented, etc. The appropriate mixed strategy is chosen by considering the relative magnitude of the program risks, and the relative effectiveness of the various approaches to resolving risk.

PLAN THE NEXT STAGE

This completes the cycle. An important feature of the spiral model, as with others, is that each cycle is completed by a review involving the primary parties concerned with the product.

Management and the spiral model

There are four key points:

INITIATING AND TERMINATING THE SPIRAL

The spiral is initiated by the hypothesis that a particular operational objective can be improved by a software solution. The spiral evolves as a series of tests of this hypothesis. If at any time the hypothesis fails, the spiral is terminated. Otherwise it terminates with the installation of new or modified software.

FEATURES OF THE SPIRAL MODEL

The model has three essential features:

1. It fosters the development of specifications which need not be uniform, exhaustive or formal. They defer detailed elaboration of low-risk software elements, and avoid unnecessary breakages in their design until the high-risk elements of the design are stabilized. (This is rolling-wave design, and developing work breakdown to a lower level in areas of high risk at an earlier stage.)
2. It incorporates prototyping as a risk reduction option at any stage of development. Prototyping and the reuse of risk analysis were previously used in going from detailed design to code.
3. It accommodates reworking or a return to earlier stages as more attractive alternatives are identified, or as new risk issues need resolution.

EVALUATION

The main advantage of the spiral model is that its range of options accommodates the good features of existing software, while its risk-driven approach avoids many difficulties. Other advantages include:

– it focuses early attention on options reusing existing software
– it accommodates evolution, growth and changes of the product
– it provides a mechanism for incorporating software quality objectives into product development
– it eliminates errors and unattractive alternatives early
– it identifies the required amount of each resource
– it uses the same approach for software development, software enhancement, or maintenance
– it provides a viable framework for integrated hardware and software system development.

However, the model is still evolving, and there are three areas which must be addressed before it can be called a mature, universal model:

1. *Matching to contract software*: the model works well on internal development projects, but needs further work for contact software. Its adaptability makes it inappropriate for fixed-price contracts.
2. *Relying on risk assessment expertise*: the spiral model places a great deal of reliance on the ability of software developers to identify and manage sources of project risk.
3. *Need for the further elaboration of the stages of the model*: the steps of the model need further elaboration to ensure all software developers are operating in a consistent manner. This includes detailed specification of deliverables and procedures, guidelines and checklists to identify the most likely sources of project risk, and techniques for the most effective resolution of risk.

RISK MANAGEMENT

Efforts to apply and refine the model have focused on creating a discipline of software risk management (Chapter 10). A top-ten list of software risk items[12] (Table 18.2), is one result of this activity. Another is the risk management approach (Section 10.6).

The problems of real life

Unfortunately, software development is not quite as simplistic as these models might imply:

– exploratory work on subsequent stages, including costing, can be required before the current stage is complete – for example design investigation is almost invariably required before it can be stated that the user requirement can be achieved within a realistic budget
– problems encountered in later stages may require reworking of earlier stages – failure to recognize this leads to earlier documentation becoming inaccurate and misleading

– the users' requirements may not remain stable throughout a protracted development process – it is then necessary to consider changed requirements and consequential changes in later stages.

Table 18.2 A prioritized top-ten list of software risk items

Risk item	*Risk management technique*
Personnel shortfalls	Staff with top talent: team/morale building: cross training prescheduling key people
Unrealistic schedules and budgets	Detailed/checked cost and duration estimates; design to cost; incremental development; reuse of software; requirements scrubbing
Developing wrong functionality	Organization/mission analysis; ops concept formulation; user surveys; prototyping. early user manuals
Developing wrong user interface	Task analysis; prototyping; scenarios; user profiles (functionality, style, workload)
Gold plating	Requirements scrubbing; prototyping; cost-benefit analysis; design to cost; value engineering
Continuing changes to requirements	High change threshold; information hiding; incremental development (defer changes to later increment)
Shortfalls in procured components	Benchmarking; inspection; expediting; reference checking; quality auditing; compatibility analysis
Shortfalls in subcontracted tasks	Reference checking; preaward audits; fixed price contracts; competitive design/prototyping; team building
Shortfalls in real-time performance	Simulation; benchmarking; modelling; prototyping; instrumentation; tuning
Straining the capabilities of computer science	Technical analysis; cost-benefit analysis; prototyping; reference checking

It is therefore important that the life cycle is not rigidly imposed. In reality, there are no clearly defined breakpoints between the stages. Equally, all the stages are composed of several substages or packages of work. However, once this is recognized, it leads not to the conclusion that the life-cycle model must be discarded, but it represents a valuable model of what is involved in the technical work of software development. The biggest single problem in the software cycle is the communication across the boundary from one stage to the next. At each stage there can be a degradation of the definition of the users' requirements. Quality assurance (Section 7.3), and configuration management (Section 7.4), play a crucial role in managing this flow of information.

Resourcing the life cycle

Many of the names of stages in the life cycle are similar to resource types working in software development. This results in each type becoming primarily associated with a stage: systems analysts with design, programmers with coding. You will hear IT people referring to the work of each resource types as an 'activity', and then they confuse the 'activity' of the resource with the work of the stage. Then resource types are not assigned to the project until the work of the stage with which they are associated is about to begin. The result is that items with long lead times are ignored by earlier resource types, resources have no time to prepare before starting, and resources cannot complete their input within the time allotted. In reality, most resource types should work throughout the project. Where the work of one resource type overlaps with a stage, that defines a work package. Early work packages are in support of the design process, and in preparation for the stage in which the resource is primarily involved. Later work packages are in support of implementation.

18.7 Business process re-engineering projects

Business process re-engineering or organizational change projects carry the added complexity that people's working lives are being radically changed, and that can lead to resistance. Key lessons arise from the study of organizational change projects:[12]

1. Define a clear and explicit strategy for the change, to which the improvement projects can easily be linked. This strategy needs to spell out what the organization is moving from and where it is moving to. It also needs to define the main thrusts of the change (quality, simplification, employee involvement, new technology, etc.), which will provide the key elements of the change.
2. Define strategic objectives for each change project, and show how these interrelate. Manage the interdependencies involved.
3. Define the key issues involved:
   - at the strategic level
   - for each of the main thrusts of change
   - project-by-project.
4. Use workshops to share the outputs from change projects, and to test options and plans.
5. Use cross-functional teams to collect and analyse data, and to generate and test options.
6. Build ownership for the change projects by soliciting input from a variety of sources.

7. Manage stakeholders explicitly at each stage of the improvement project.
8. Ensure all outputs from projects are defined, and that these mesh with assumed inputs to other improvement projects.
9. Allow for the possibility of emergent projects crystallizing (like the structure project), rather than resisting change in project definition at all costs.
10. Analyse and evaluate the difficulty of change projects in terms of:
    - scope and complexity
    - duration
    - 'iceberg issues' (especially behavioural resistance)
    - fluid outcomes.

It is also recognized that people undergo a cycle of emotional response to the change process. Table 18.3 shows the cycle and the attitudes encountered. There is no point trying to ignore this cycle. It will occur, and you will have a more successful outcome if you manage the cycle than if you try to pretend it will not occur.

Table 18.3 Emotional response to extreme change

Stage	*Response*
Stability	Management communicates their vision, the need for change and the consequences
Immobilization denial	People are taken by surprise. Their reaction is anxiety and confusion. People defend themselves against what they see as a threat to their life or livelihood: – 'They can't mean me!' – 'Is that what we get for years of loyalty!' – 'Management is overreacting; it can't be that bad!'
Anger	Openly displayed anger towards management emerges. People try to take control, through their power base in the organization, through trade unions, etc. Alliances are formed; efforts to divide management are made; all means to reverse the situation. Management must persistently argue the case, and not indulge in personal warfare.
Bargaining	People begin to aim for a modified solution. All kinds of remedies will be proposed in order to try to reduce the impact of the change: – 'If we take a cut in salary?' – 'If we increase our productivity?' Management must be steadfast and stick to reality
Depression	Frustration and a feeling of having lost spreads. People find it difficult to work and organizational paralysis sets in.

Table 18.3 *(continued)*

Stage	*Response*
Testing	The individual and the organization start working with alternative exit strategies to try to facilitate the individual's transition: – 'Did you say I could have six months' pay while looking for a new job?' – 'Being paid through a year's MBA programme would help the transition.' Management helps to find realistic alternatives
Acceptance	Individuals and the organization deal realistically with the situation. They may not like it but they accept it. Management gives recognition and support towards future plans New stability is achieved.

18.8 Summary

1. Projects can be classified in many ways. Different types of projects require a different approach or emphasis to their management.
2. In addition to traditional projects to deliver and commission a facility, projects can conduct:
   - marketing surveys and product development
   - research and development
   - maintenance and decommissioning.
3. New product development can lead to many projects:
   - research and development
   - product design and prototyping
   - facility design and delivery
   - product launch.
4. New product development can be managed through:
   - new product committees
   - product managers
   - new product managers
   - new product departments
   - venture teams
   - task forces.
5. The stages of the product development life cycle include:
   - idea generation and screening
   - concept development and testing
   - marketing strategy
   - business analysis

- product development
- market testing
- commercialization.

6. Selection of R&D projects should be against the corporate strategy, in particular a technology strategy which will recognize the enabling technologies of the future and how the company will exploit them.
7. The project manager should be chosen for their ability as a leader and communicator, and not for their technical expertise. Communication must be upwards, outwards and sideways.
8. R&D projects cannot be planned in detail, so must be planned against key milestones (important intermediate deliverables), with time and cost constraints set by the market requirements. Skills types must be chosen systematically and creatively.
9. An organization must be effective in achieving its R&D objectives. It is no good to achieve nothing efficiently.
10. Concurrent engineering is used to overlap stages in the product development cycle, to speed up the delivery of new products.
11. Concurrent engineering requires the adoption of new project management practices, including:
    - a change in organizational culture
    - cross-functional team working
    - use of new technology, information systems and other new techniques.
12. There are risks and pitfalls associated with concurrent engineering, including:
    - attitudes of middle management
    - authorization of the project
    - organizational and cultural change
    - managing interfaces
    - technical management
    - cost and risk controls and release of finance.
13. There are four types of model of the life cycle for software development projects:
    - code-and-fix models
    - stage-wise models
    - waterfall models
    - spiral models.
14. Stages in all the models are identified not by the work done, but by the deliverables, or intermediate products, in which they result. The control process focuses on the quality of these deliverables.
15. Spiral models, which can incorporate the others as special cases, view the project as moving repeatedly through four quadrants:

- plan the forthcoming stage
- determine objectives, alternatives, constraints
- evaluate alternatives, and identify and resolve risks
- develop and verify the next level product
- plan the forthcoming stage.

16. The management of business process re-engineering projects requires the careful management of the response of the people affected.

References

1. Wearne, S.H., *Principles of Engineering Organisation*, Edward Arnold, 1973.
2. Kerzner, H., *Project Management: A systems approach to planning, scheduling and controlling*, Van Nostrand Reinhold, 1984.
3. Kotler, P., *Marketing Management: analysis, planning and control*, Prentice-Hall, 1988.
4. Handscome, R., *The Product Management Handbook*, McGraw-Hill, 1989.
5. Foxall, G., *Corporate Innovation, Marketing and Strategy*, Croom Helm, 1984.
6. Craven, D.W., *Strategic Marketing*, Irwin, 1985.
7. Buell, V.P., *Marketing Management: A strategic planning approach*, McGraw-Hill, 1968.
8. McDonald, M.H.B., *Marketing Plans: How to prepare them, how to use them*, Heinemann, 1984.
9. Pelz, D.C. and Andrews, F.M., *Scientists in Organizations: Productive climates for research and development*, Institute for Social Research, 1976.
10. Toney, F., 'Good results yield . . . resistance: Proving that project management pays is met with denial in one Fortune 500 company', *Project Management Network*, October 1996.
11. Boehm, B.W., 'A spiral model of software development and enhancement', *Computer*, May, 61–72, The Institute of Electrical and Electronic Engineers, 1988.
12. Turner, J.R., Grude, K.V. and Thurloway, L. (eds), *The Project Manager as Change Agent*, McGraw-Hill, 1996.

Notes

a. Section 18.3 incorporates material from the first edition based on a contribution originally made by Dr Susan Foreman of Henley Management College.
b. Section 18.4 incorporates material from the first edition based on a contribution originally made by Dr Janice Light and Professor Gordon Edge.
c. Section 18.5 incorporates material based on research done by Michael Hougham. In using his research, I have sometimes incorporated his text.
d. Section 18.6 incorporates material from the first edition based on a contribution originally made by Anne French.