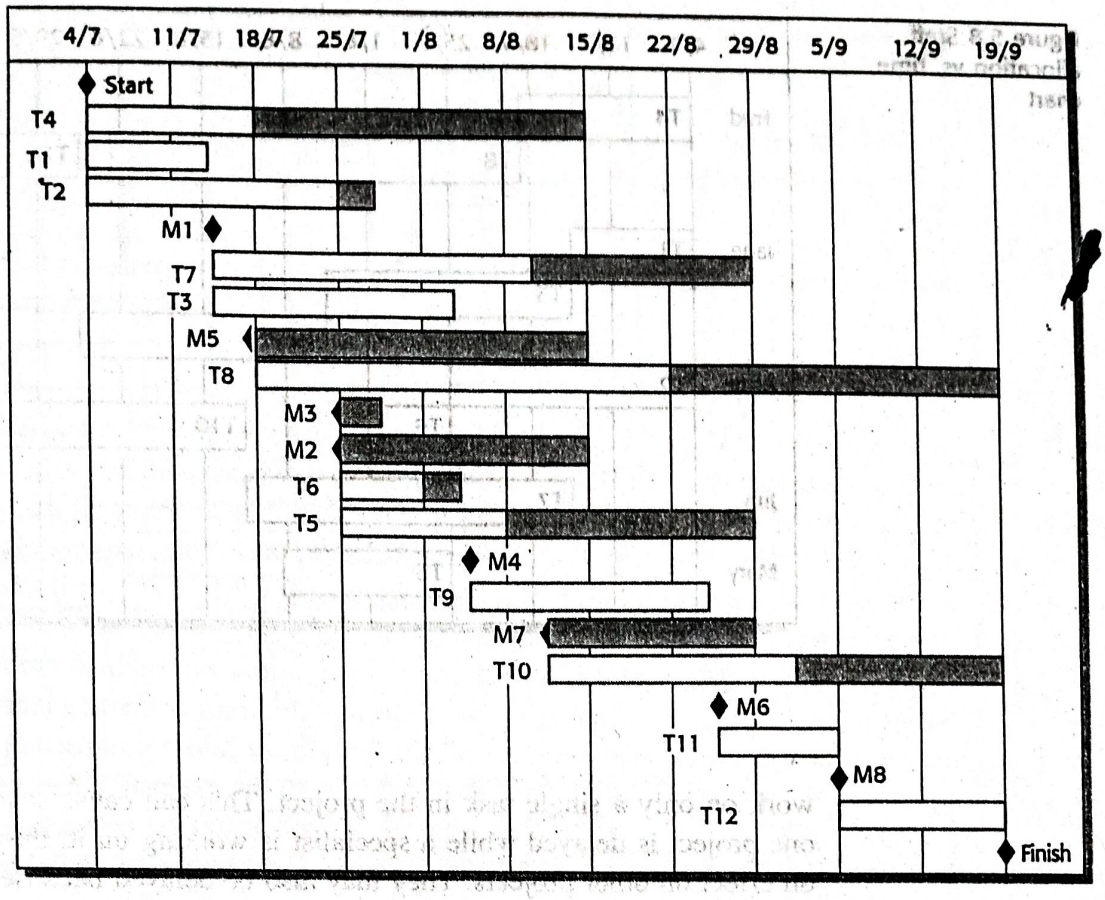
Figure 5.7 Activity bar chart



are known, the activity chart should be reviewed. Later project activities may then be reorganised to reduce the length of the critical path.

Figure 5.7 is a complementary way of representing project schedule information. It is a bar chart showing a project calendar and the start and finish dates of activities. Sometimes these are called *Gantt charts*, after their inventor. Reading from left to right, the bar chart clearly shows when activities start and end.

Some of the activities shown in the bar chart in Figure 5.7 are followed by a shaded bar whose length is computed by the scheduling tool. This highlights the flexibility in the completion date of these activities. If an activity does not complete on time, the critical path will not be affected until the end of the period marked by the shaded bar. Activities that lie on the critical path have no margin of error and can be identified because they have no associated shaded bar.

In addition to considering schedules, as a project manager you must also consider resource allocation and, in particular, the allocation of staff to project activities. This allocation can also be input to project management tools and a bar chart generated that shows when staff are employed on the project (Figure 5.8). People don't have to be assigned to a project at all times. During intervening periods they may be on holiday, working on other projects, attending training courses or engaging in some other activity.

Large organisations usually employ a number of specialists who work on a project when needed. In Figure 5.8, you can see that Mary and Jim are specialists who

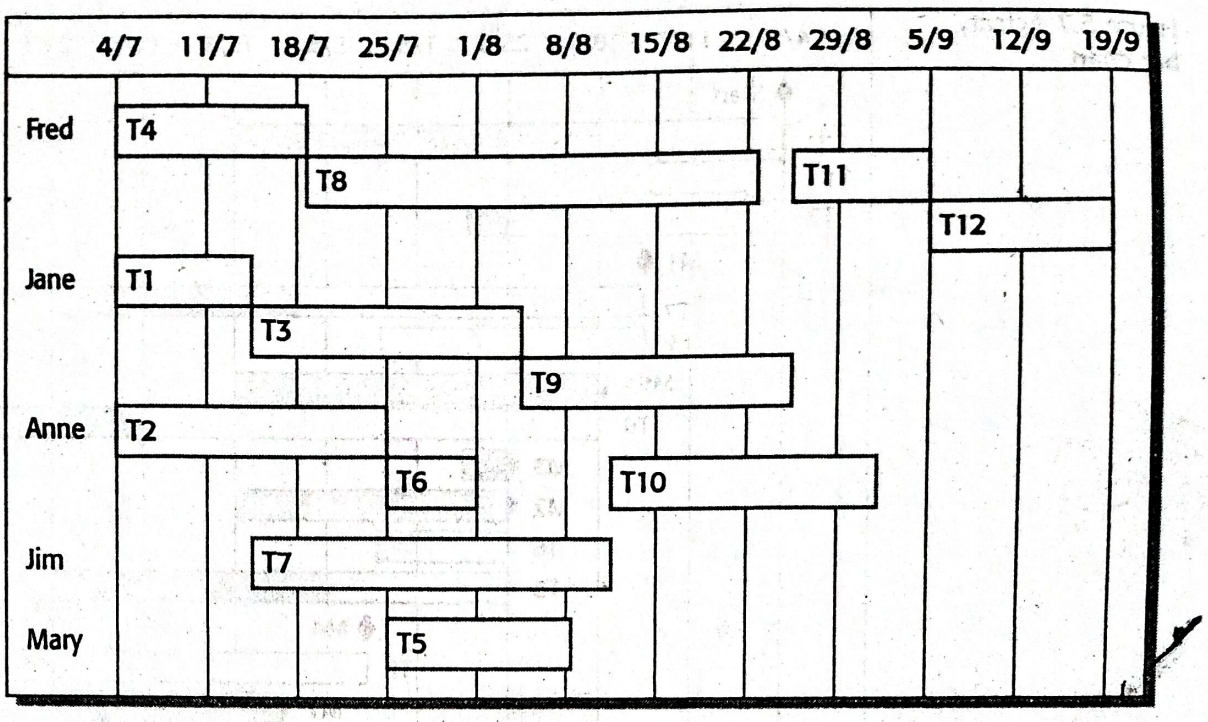


Figure 5.8 Staff allocation vs. time chart

work on only a single task in the project. This can cause scheduling problems. If one project is delayed while a specialist is working on it, this may have a knock-on effect on other projects. They may also be delayed because the specialist is not available.

5.4 Risk management

Risk management is increasingly seen as one of the main jobs of project managers. It involves anticipating risks that might affect the project schedule or the quality of the software being developed and taking action to avoid these risks (Hall, 1998) (Ould, 1999). The results of the risk analysis should be documented in the project plan along with an analysis of the consequences of a risk occurring. Effective risk management makes it easier to cope with problems and to ensure that these do not lead to unacceptable budget or schedule slippage.

Simplistically, you can think of a risk as something that you'd prefer not to have happen. Risks may threaten the project, the software that is being developed or the organisation. There are, therefore, three related categories of risk:

1. *Project risks* are risks that affect the project schedule or resources. An example might be the loss of an experienced designer.

2. *Product risks* are risks that affect the quality or performance of the software being developed. An example might be the failure of a purchased component to perform as expected.

3. *Business risks* are risks that affect the organisation developing or procuring the software. For example, a competitor introducing a new product is a business risk.

Of course, these risk types overlap. If an experienced programmer leaves a project, this can be a project risk because the delivery of the system may be delayed. It can also be a product risk because a replacement may not be as experienced and so may make programming errors. Finally, it can be a business risk because the programmer's experience is not available for bidding for future business.

The risks that may affect a project depend on the project and the organisational environment where the software is being developed. However, many risks are universal—some of the most common risks are shown in Figure 5.9.

Risk management is particularly important for software projects because of the inherent uncertainties that most projects face. These stem from loosely defined requirements, difficulties in estimating the time and resources required for

igure 5.9 Possible oftware risks

Risk	Risk type	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

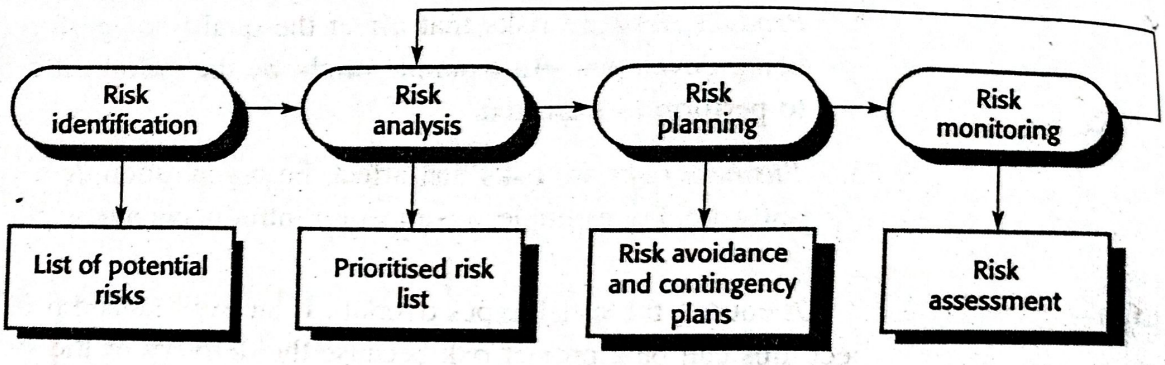


Figure 5.10 The risk management process

software development, dependence on individual skills and requirements changes due to changes in customer needs. You have to anticipate risks, understand the impact of these risks on the project, the product and the business, and take steps to avoid these risks. You may need to draw up contingency plans so that, if the risks do occur, you can take immediate recovery action.

The process of risk management is illustrated in Figure 5.10. It involves several stages:

1. *Risk identification* Possible project, product and business risks are identified.
2. *Risk analysis* The likelihood and consequences of these risks are assessed.
3. *Risk planning* Plans to address the risk either by avoiding it or minimising its effects on the project are drawn up.
4. *Risk monitoring* The risk is constantly assessed and plans for risk mitigation are revised as more information about the risk becomes available.

The risk management process, like all other project planning, is an iterative process which continues throughout the project. Once an initial set of plans are drawn up, the situation is monitored. As more information about the risks becomes available, the risks have to be reanalysed and new priorities established. The risk avoidance and contingency plans may be modified as new risk information emerges.

You should document the outcomes of the risk management process in a risk management plan. This should include a discussion of the risks faced by the project, an analysis of these risks and the plans that are required to manage these risks. Where appropriate, you should also include in the plan results of the risk management process such as specific contingency plans to be activated if the risk occurs.

5.4.1 Risk identification

Risk identification is the first stage of risk management. It is concerned with discovering possible risks to the project. In principle, these should not be assessed or prioritised at this stage, although, in practice, risks with very minor consequences or very low probability risks are not usually considered.

Risk identification may be carried out as a team process using a brainstorming approach or may simply be based on experience. To help the process, a checklist of different types of risk may be used. There are at least six types of risk that can arise:

1. *Technology risks* Risks that derive from the software or hardware technologies that are used to develop the system.
2. *People risks* Risks that are associated with the people in the development team.
3. *Organisational risks* Risks that derive from the organisational environment where the software is being developed.
4. *Tools risks* Risks that derive from the CASE tools and other support software used to develop the system.
5. *Requirements risks* Risks that derive from changes to the customer requirements and the process of managing the requirements change.
6. *Estimation risks* Risks that derive from the management estimates of the system characteristics and the resources required to build the system.

Figure 5.11 gives some examples of possible risks in each of these categories. When you have finished the risk identification process, you should have a long list of risks that could occur and which could affect the product, the process and the business.

5.4.2 Risk analysis

During the risk analysis process, you have to consider each identified risk and make a judgement about the probability and the seriousness of it. There is no easy way to do this—you must rely on your own judgement and experience, which is why experienced project managers are generally the best people to help with risk management. These risk estimates should not generally be precise numeric assessments but should be based around a number of bands:

- The probability of the risk might be assessed as very low (<10%), low (10–25%), moderate (25-50%), high (50–75%) or very high (>75%).
- The effects of the risk might be assessed as catastrophic, serious, tolerable or insignificant.

You should then tabulate the results of this analysis process using a table ordered according to the seriousness of the risk. Figure 5.12 illustrates this for the risks identified in Figure 5.11. Obviously, the assessment of probability and seriousness is arbitrary here. In practice, to make this assessment you need detailed information about the project, the process, the development team and the organisation.

Figure 5.11 Risks and risk types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components which should be reused contain defects which limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organisational	The organisation is restructured so that different management are responsible for the project. Organisational financial problems force reductions in the project budget.
Tools	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements which require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

Of course, both the probability and the assessment of the effects of a risk may change as more information about the risk becomes available and as risk management plans are implemented. Therefore, you should update this table during each iteration of the risk process.

Once the risks have been analysed and ranked, you should assess which are most significant. Your judgement must depend on a combination of the probability of the risk arising and the effects of that risk. In general, catastrophic risks should always be considered, as should all serious risks that have more than a moderate probability of occurrence.

Boehm (Boehm, 1988) recommends identify and monitoring the 'top 10' risks, but I think that this figure is rather arbitrary. The right number of risks to monitor must depend on the project. It might be 5 or it might be 15. However, the number of risks chosen for monitoring should be manageable. A very large number of risks would simply require too much information to be collected. From the risks identified in Figure 5.12, it is appropriate to consider all 8 risks that have catastrophic or serious consequences.

5.4.3 Risk planning

The risk planning process considers each of the key risks that have been identified and identifies strategies to manage the risk. Again, there is no simple process that

Figure 5.12 Risk analysis

Risk	Probability	Effects
Organisational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Software components which should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements which require major design rework are proposed.	Moderate	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious
The database used in the system cannot process as many transactions per second as expected.	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
CASE tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The code generated by CASE tools is inefficient.	Moderate	Insignificant

can be followed to establish risk management plans. It relies on the judgement and experience of the project manager. Figure 5.13 shows possible strategies that have been identified for the key risks from Figure 5.12.

These strategies fall into three categories:

1. *Avoidance strategies* Following these strategies means that the probability that the risk will arise will be reduced. An example of a risk avoidance strategy is the strategy for dealing with defective components shown in Figure 5.13.

2. *Minimisation strategies* Following these strategies means that the impact of the risk will be reduced. An example of a risk minimisation strategy is that for staff illness shown in Figure 5.13

Figure 5.13 Risk management strategies

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying-in components, investigate the use of a program generator.

3. *Contingency plans* Following these strategies means that you are prepared for the worst and have a strategy in place to deal with it. An example of a contingency strategy is the strategy for organisational financial problems in Figure 5.13.

You can see here the analogy with the strategies used in critical systems to ensure reliability, security and safety. Essentially, it is best to use a strategy that avoids the risk. If this is not possible, use one that reduces the chances that the risk will have serious effects. Finally, have strategies in place that reduce the overall impact of a risk on the project or product.

5.4.4 Risk monitoring

Risk monitoring involves regularly assessing each of the identified risks to decide whether or not that risk is becoming more or less probable and whether the effects of the risk have changed. Of course, this cannot usually be observed directly, so you have to look at other factors that give you clues about the risk probability and its effects. These factors are obviously dependent on the types of risk. Figure 5.14 gives some examples of factors that may be helpful in assessing these risk types.

Risk monitoring should be a continuous process, and, at every management progress review, you should consider and discuss each of the key risks separately.

Figure 5.14 Risk factors

Risk type	Potential indicators
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team members, job availability
Organisational	Organisational gossip, lack of action by senior management
Tools	Reluctance by team members to use tools, complaints about CASE tools, demands for higher-powered workstations
Requirements	Many requirements change requests, customer complaints
Estimation	Failure to meet agreed schedule, failure to clear reported defects

KEY POINTS

- Good software project management is essential if software engineering projects are to be developed on schedule and within budget.
- Software management is distinct from other engineering management. Software is intangible. Projects may be novel or innovative so there is no body of experience to guide their management. Software processes are not well understood.
- Software managers have diverse roles. Their most significant activities are project planning, estimating and scheduling. Planning and estimating are iterative processes. They continue throughout a project. As more information becomes available, plans and schedules must be revised.
- A project milestone is a predictable outcome of an activity where some formal report of progress should be presented to management. Milestones should occur regularly throughout a software project. A deliverable is a milestone that is delivered to the project customer.
- Project scheduling involves the creation of various graphical plan representations of part of the project plan. These include activity charts showing the interrelationships of project activities and bar charts showing activity durations.
- Major project risks should be identified and assessed to establish their probability and the consequences for the project. You should make plans to avoid, manage or deal with likely risks if or when they arise. Risks should be explicitly discussed at each project progress meeting.

FURTHER READING

Waltzing with Bears: Managing Risk on Software Projects. A very practical and easy-to-read introduction to risks and risk management. (T. DeMarco and T. Lister, 2003, Dorset House.)

Managing Software Quality and Business Risk. Chapter 3 of this book is simply the best discussion of risk that I have seen anywhere. The book is oriented around risk and I think it is probably the best book on this topic currently available. (M. Ould, 1999, John Wiley & Sons.)

The Mythical Man Month (Anniversary Edition). The problems of software management have been unchanged since the 1960s and this is one of the best books on the topic. An interesting and readable account of the management of one of the first very large software projects, the IBM OS/360 operating system. The anniversary edition (published 20 years after the original edition in 1975) includes other classic papers by Brooks. (F. P. Brooks, 1995, Addison-Wesley.)

Software Project Survival Guide. This is a very pragmatic account of software management, but it contains good practical advice. It is easy to read and understand. (S. McConnell, 1998, Microsoft Press.)

See Part 6 for other readings on management.

EXERCISES

5.1 Explain why the intangibility of software systems poses special problems for software project management.

5.2 Explain why the best programmers do not always make the best software managers. You may find it helpful to base your answer on the list of management activities in Section 5.1.

5.3 Explain why the process of project planning is iterative and why a plan must be continually reviewed during a software project.

5.4 Briefly explain the purpose of each of the sections in a software project plan.

5.5 What is the critical distinction between a milestone and a deliverable?

5.6 Figure 5.15 sets out a number of activities, durations and dependencies. Draw an activity chart and a bar chart showing the project schedule.

5.7 Figure 5.5 gives task durations for software project activities. Assume that a serious, unanticipated setback occurs and instead of taking 10 days, task T5 takes 40 days. Revise the activity chart accordingly, highlighting the new critical path. Draw up new bar charts showing how the project might be reorganised.

5.8 Using reported instances of project problems in the literature, list management difficulties that occurred in these failed programming projects. (I suggest that you start with Brooks's book, as suggested in Further Reading.)