

UNIT - 4

7-Risk Management

Risk as ‘an uncertain event or condition that, if it occurs, has a positive or negative effect on a project’s objectives’

Risk is ‘the chance of exposure to the adverse consequences of future events’.

The key elements of a risk follow.

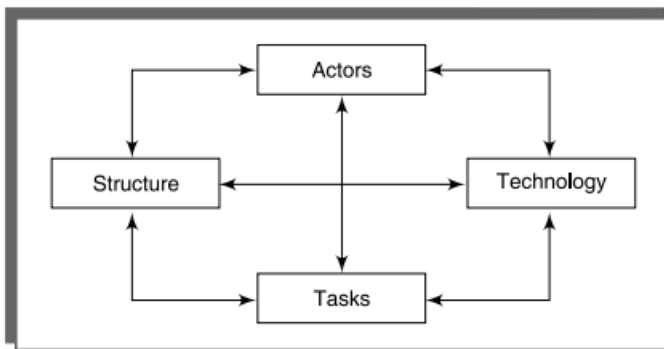
It relates to the future The future is inherently uncertain. Some things which seem obvious when a project is over, for example that the costs were under estimated or that a new technology was overly difficult to use, might not have been so obvious during planning.

It involves cause and effect For example, a ‘cost over-run’ might be identified as a risk, but ‘cost over-run’ describes some damage, but does not say what causes it.

Categories of Risk

An ICT project manager is normally given the objective of installing the required application by a specified deadline and within an agreed budget. Other objectives might be set, especially with regard to quality requirements. Project risks are those that could prevent the achievement of the objectives given to the project manager and the project team

Kalle Lyytinen and his colleagues, for instance, have proposed a sociotechnical model of risk, a diagrammatic representation of which appears in Figure 7.2



Lyytinen-Mathiassen-Ropponen risk framework

Actors - refers to all the people involved in the project including both developers, users and managers e.g. a risk could be that high staff turnover leads to expertise of value to the project being lost.

Technology – includes both the technology used to implement the project and that embedded in the project deliverables – risk could be that the technologies selected are not in fact appropriate.

Structure – this includes management structures. For example the implementation might need user participation in some tasks, but the responsibility for managing users is not clearly allocated.

Tasks – the work to be carried out. A typical risk is that the amount of effort needed to carry out the task is underestimated.

Boxes are interlinked. Risks often arise from the relationships between factors – For example, estimates being wrong could be influenced by problems with actors due to lack of experience with a technical domain) or the structure (over optimism of managers keen to win work).

A Framework for Dealing with Risk

Planning for risk includes these steps:

- i. risk identification;
- ii. risk analysis and prioritization;
- iii. risk planning;

iv. risk monitoring

Steps (i) to (iii) above will probably be repeated. When risks that could prevent a project success are identified, plans can be made to reduce or remove their threat. The plans are then reassessed to ensure that the original risks are reduced sufficiently and no new risks inadvertently introduced.

Risk Identification

The two main approaches to the identification of risks are the use of checklists and brainstorming.

- **Checklists** are simply lists of the risks that have been found to occur regularly in software development projects. A specialized list of software development risks by Barry Boehm appears in Table 7.1 in a modified version. Ideally a group of representative project stakeholders examines a checklist identifying risks applicable to their project. Often the checklist suggests potential countermeasures for each risk.

Risk	Risk reduction techniques
Personnel shortfalls	Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel
Unrealistic time and cost estimates	Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods
Developing the wrong software functions	Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals
Developing the wrong user interface	Prototyping; task analysis; user involvement
Gold plating	Requirements scrubbing; prototyping; cost-benefit analysis; design to cost

(Contd)

(Contd)

Late changes to requirements	Stringent change control procedures; high change threshold; incremental development (deferring changes)
Shortfalls in externally supplied components	Benchmarking; inspections; formal specifications; contractual agreements; quality assurance procedures and certification
Shortfalls in externally performed tasks	Quality assurance procedures; competitive design or prototyping; contract incentives
Real-time performance shortfalls	Simulation; benchmarking; prototyping; tuning; technical analysis
Development technically too difficult	Technical analysis; cost-benefit analysis; prototyping; staff training and development

- **Brainstorming** Ideally, representatives of the main stakeholders should be brought together once some kind of preliminary plan has been drafted. They then identify, using their individual knowledge of different parts of the project, the problems that might occur. This collaborative approach may generate a sense of ownership in the project.

Risk Assessment

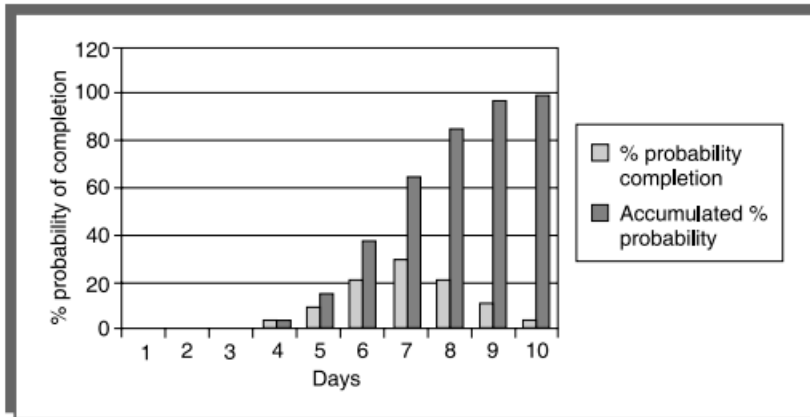
A common problem with risk identification is that a list of risks is potentially endless. A way is needed of distinguishing the damaging and likely risks. This can be done by estimating the risk exposure for each risk using the formula: **risk exposure = (potential damage) * (probability of occurrence)**

Using the most rigorous – but not necessarily the most practical – approach, the potential damage would be assessed as a money value. Say a project depended on a data centre vulnerable to fire. It might be estimated that if a fire occurred a new computer configuration could be established for £500,000. It might also be estimated that where the computer is located there is a 1 in 1000 chance of a fire actually happening, that is a probability of 0.001.

The risk exposure in this case would be: $£500,000 * 0.001 = £500$

A crude way of understanding this value is as the minimum sum an insurance company would require as a premium. If 1000 companies, all in the same position, each contributed £500 to a fund then, when the 1 in 1000 chance of the fire actually occurred, there would be enough money to cover the cost of recovery

With some risks, there could be not only damage but also gains. The testing of a software component is scheduled to take six days, but is actually done in three days. A team leader might therefore feel justified in producing a probability chart like the one in Figure 7.3. This shows the probability of a task being completed in four days (5%), then five days (10%), and so on. The accumulated probability for the seventh day (65%) means that there is a 65% chance that the task will be finished on or before the seventh day.



7.3 Probability chart

Risk Planning

Having identified the major risks and allocated priorities, the task is to decide how to deal with them. The choices discussed will be:

- **Risk acceptance** This is the do-nothing option. We will already, in the risk prioritization process, have decided to ignore some risks in order to concentrate on the more likely or damaging. We could decide that the damage inflicted by some risks would be less than the costs of action that might reduce the probability of a risk happening.
- **Risk avoidance** Some activities may be so prone to accident that it is best to avoid them altogether. If you are worried about sharks then don't go into the water. For example, given all the problems with developing software solutions from scratch, managers might decide to retain existing clerical methods, or to buy an off-the-shelf solution.
- **Risk reduction** Risk reduction –actions are taken to reduce its likelihood e.g. prototypes ought to reduce the risk of incorrect requirements or enough incentives are given to development team to avoid departure of staff.
- **Risk transfer** In this case the risk is transferred to another person or organization. With software projects, an example of this would be where a software development task is outsourced to an outside agency for a fixed fee. You might expect the supplier to quote a higher figure to cover the risk that the project takes longer than the 'average' expected time. On the other hand, a well-established external organization might have productivity advantages as its developers are experienced in the type of development to be carried out. The need to compete with other software development specialists would also tend to drive prices down.

Risk Management

- **Contingency** Risk reduction activities would appear to have only a small impact on reducing the probability of some risks, for example staff absence through illness. While some employers encourage their employees to adopt a healthy lifestyle, it remains likely that some project team members will at some point be brought down by minor illnesses such as flu. These kinds of risk need a contingency plan. This is a planned action to be carried out if the particular risk materializes. If a team member involved in urgent work were ill then the project manager might draft in another member of staff to cover that work

- Deciding on the risk actions** Whatever the countermeasures that are considered, they must be cost-effective. On those occasions where a risk exposure value can be calculated as a financial value using the (value of damage) ³ (probability of occurrence) formula
 - the cost-effectiveness of a risk reduction action can be assessed by calculating the risk reduction leverage (RRL). **risk reduction leverage = (REbefore – REafter)/(cost of risk reduction)**
 REbefore is the risk exposure, before risk reduction actions have been taken. REafter is the risk exposure after taking the risk reduction action.
 e.g.
 1% chance of a fire causing £200k damage
 REafter is risk exposure after risk reduction e.g. fire alarm costing £500 reduces probability of fire damage to 0.5%

$$RRL = (1\% \text{ of } £200k) - (0.5\% \text{ of } £200k) / £500 = 2$$

$$RRL > 1.00 \text{ therefore worth doing}$$
- Creating and maintaining the risk register** When the project planners have picked out and examined what appear to be the most threatening risks to the project, they need to record their findings in a risk register. The typical content of such a register is shown in Figure 7.5. After work starts on the project more risks will emerge and be added to the register

TABLE 7.5 Fairley's four commercial off-the-shelf (COTS) software acquisition risks

Integration	Difficulties in integrating the data formats and communication protocols of different applications.
Upgrading	When the supplier upgrades the package, the package might no longer meet the users' precise requirements. Sticking with the old version could mean losing the supplier's support for the package.
No source code	If you want to enhance the system, you might not be able to do so as you do not have access to the source code.
Supplier failures or buyouts	The supplier of the application might go out of business or be bought out by a rival supplier.

Applying the PERT Technique

PERT((Program Evaluation Review Technique) was developed to take account of uncertainty surrounding estimates of task durations. The method is very similar to CPM technique but instead of using a single estimate for the duration of each task , three estimates are required for each activity

- Most likely time: the time we would expect the task to take under normal circumstances. We shall identify this by the letter m.
- Optimistic time: the shortest time in which we could expect to complete the activity, barring outright miracles. We shall use the letter a for this.
- Pessimistic time: the worst possible time, allowing for all reasonable eventualities but excluding 'acts of God and warfare' (as they say in most insurance exclusion clauses). We shall call this b.

PERT then combines these three estimates to form a single expected duration, te, using the formula

$$te = (a+4m+b)/6$$

Table 7.6 provides additional activity duration estimates for the network shown in Figure 6.29. There are new estimates for a and b and the original activity duration estimates have been used as the most likely times, m . Calculate the expected duration, t_e , for each activity.

TABLE 7.6 PERT activity time estimates

Activity	Optimistic (a)	Activity durations (weeks). Most likely (m)	Pessimistic (b)
A	5	6	8
B	3	4	5
C	2	3	3
D	3.5	4	5
E	1	3	4
F	8	10	15
G	2	3	4
H	2	2	2.5

Expected times and Standard Deviation

Activity	Optimistic (a)	Activity Duration(m)	Pessimistic (b)	Expected (t_e)	SD (s)
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

Using expected durations

The expected durations are used to carry out a forward pass through a network, using the same method as the CPM technique.

Unlike the CPM approach, the PERT method does not indicate the earliest date by which we could complete the project but the expected (or most likely) date. An advantage of this approach is that it places an emphasis on the uncertainty of the real world. Rather than being tempted to say 'the completion date for the project is. . .' we are led to say 'we expect to complete the project by. . .'.

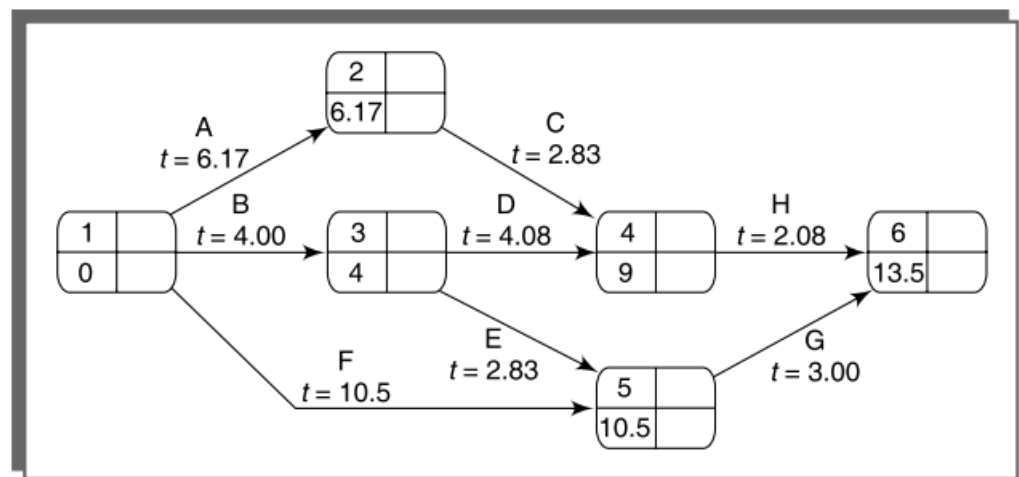


FIGURE 7.6 The PERT network after the forward pass

Activity standard deviations A quantitative measure of the degree of uncertainty of an activity duration estimate may be obtained by calculating the standard deviation s of an activity time, using the formula $s = (b-a)/6$

The activity standard deviation is proportional to the difference between the optimistic and pessimistic estimates, and can be used as a ranking measure of the degree of uncertainty or risk for each activity.

Calculating the z values The z value is calculated for each node that has a target date. It is equivalent to the number of standard deviations between the node's expected and target dates. It is calculated using the formula

$Z = (T - t_e) / s$ where t_e is the expected date and T the target date.

9. Monitoring and Control

Once the project is started attention must be focused on progress. This requires

Monitoring: Concerns the activity of collecting information(data) related to what is happening.

Control: Concerns the interpretation of the information(data) and the comparison with the schedule to understand if the project is proceeding as planned or a revision is needed.

Creating the Framework Exercising control over a project and ensuring that targets are met is a matter of regular monitoring – finding out what is happening and comparing it with targets

If there is mismatch between the planned outcomes and the actual ones then either replanning is needed to bring the project back on target or the target will have to be revised. It also illustrates the important steps that must be taken after completion of the project so that experience gained from one project can be fed into planning stages of future projects, thus allowing us to learn from previous mistakes.

- **Responsibility** The overall responsibility for ensuring satisfactory progress on a project is often the role of the project steering committee, project management board or, in PRINCE2, Project Board. Day-to-day responsibility will rest with the project manager and, in all but the smallest of projects, aspects of this can be delegated to team leaders.
- **Assessing progress** Some information used to assess project progress will be collected routinely, while other information will be triggered by specific events. Wherever possible, this information should be objective and tangible – whether or not a particular report has been delivered, for example. Sometimes, however, assessment will have to depend on estimates of the proportion of the current activity that has been completed.
- **Setting checkpoints** It is essential to set a series of checkpoints in the initial activity plan. Checkpoints may be:
 - regular (monthly, for example);
 - tied to specific events such as the production of a report or other deliverable
- **Taking Snapshots:** The frequency of reporting depends on the size and risk of the project. In general the higher the level, the less frequent and less detailed reporting is needed. Team leaders may want to assess progress daily while project managers may prefer weekly or monthly reporting

Collecting the Data

As a rule, managers will try to break down long activities into more controllable tasks of one or two weeks' duration. However, it will still be necessary to gather information about partially completed activities and, in particular, forecasts of how much work is left to be completed. It can be difficult to make such forecasts accurately.

- **Partial completion reporting** Many organizations use standard accounting systems with weekly timesheets to charge staff time to individual jobs. The staff time booked to a project indicates the work carried out and the charges to the project. It does not, however, tell the project manager what has been produced or whether tasks are on schedule. It is therefore common to adapt or enhance existing accounting data collection systems to meet the needs of project control. Weekly timesheets, for example, are frequently adapted by breaking jobs down to activity level and requiring information about work done in addition to time spent.
- **Red/amber/green (RAG) reporting** One popular way of overcoming the objections to partial completion reporting is to avoid asking for estimated completion dates, but to ask instead for the team members' estimates of the likelihood of meeting the planned target date. One way of doing this is the traffic-light method. This consists of the following steps:
 1. identify the key (first level) elements for assessment in a piece of work;
 2. break these key elements into constituent elements (second level);
 3. assess each of the second-level elements on the scale green for 'on target', amber for 'not on target but recoverable', and red for 'not on target and recoverable only with difficulty';
 4. review all the second-level assessments to arrive at first-level assessments;
 5. review first- and second-level assessments to produce an overall assessment.

Visualizing Progress

Having collected data about project progress, a manager needs some way of presenting that data to greatest effect.

- **The Gantt chart** One of the simplest and oldest techniques for tracking project progress is the Gantt chart. This is essentially an activity bar chart indicating scheduled activity dates and durations, frequently augmented with activity floats. Reported progress is recorded on the chart (normally by shading activity bars) and a 'today cursor' provides an immediate visual indication of which activities are ahead or behind schedule
- **The slip chart** A slip chart is a very similar alternative favoured by some project managers who believe it provides a more striking visual indication of those activities that are not progressing to schedule – the more the slip line bends, the greater the variation from the plan. Additional slip lines are added at intervals and, as they build up, the project manager will gain an idea as to whether the project is improving (subsequent slip lines bend less) or not. A very jagged slip line indicates a need for rescheduling.
- **The timeline:** is a method of recording and displaying the way in which targets have changed throughout the duration of the project. It indicates the slippage of the project completion date.

Cost Monitoring

Expenditure monitoring is an important component of project control, not only in itself, but also because it provides an indication of the effort that has gone into (or at least been charged to) a project. A project might be on time but only because more money has been spent on activities than originally budgeted. A cumulative expenditure chart such as that shown in Figure 9.9 provides a simple method of comparing actual and planned expenditure.

Cost charts become much more useful if we add projected future costs calculated by adding the estimated costs of uncompleted work to the costs already incurred.

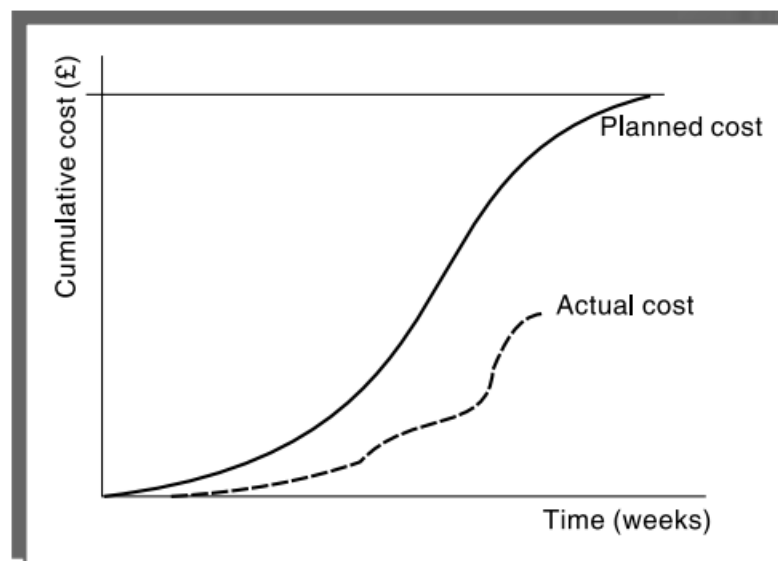


FIGURE 9.9 Tracking cumulative expenditure

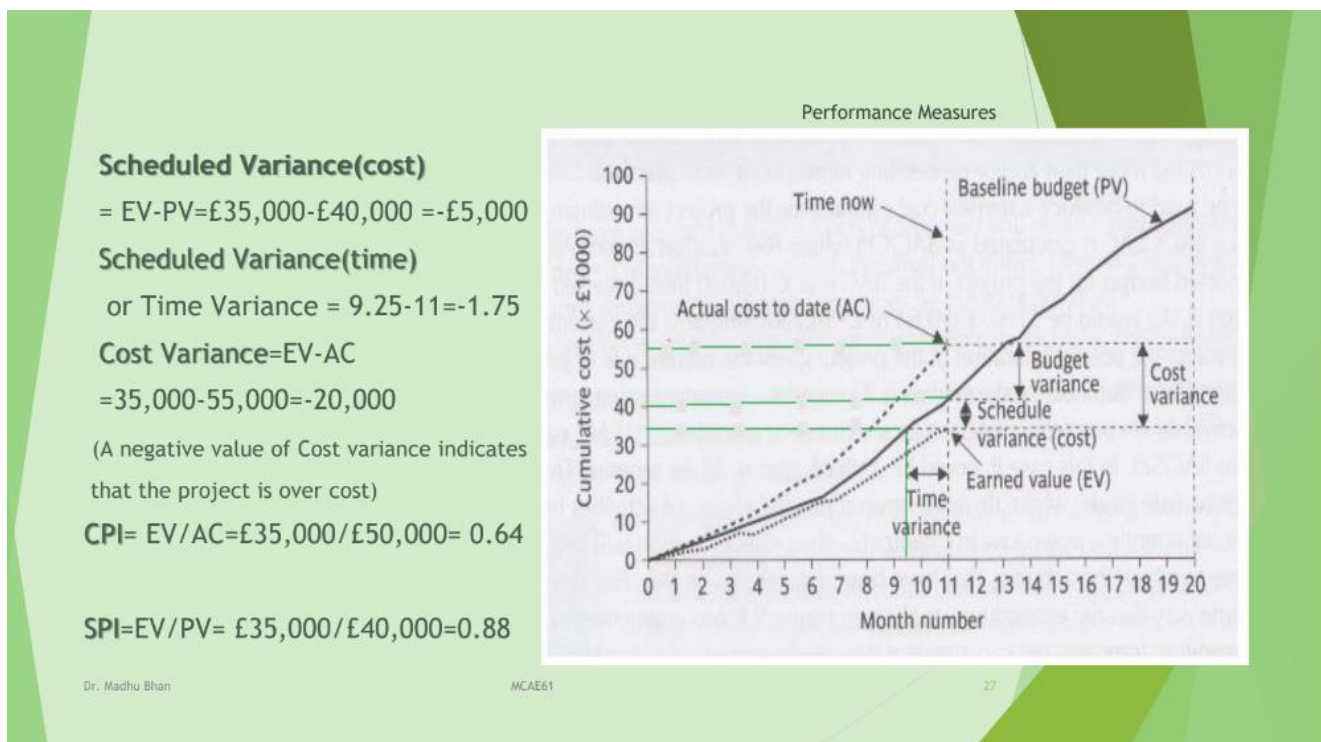
Earned Value Analysis (EVA)

The value assigned in the plan is considered as “earned” when the corresponding activity or work package is performed/terminated. This is also called as budgeted cost of work performed (BCWP) (The original budgeted cost for the task is the planned value – PV - or budget cost of work scheduled - BCWS).

A task that has not been started is assigned an earned value of zero and when it is completed it is credited with the original planned value of the task. Earned value can be represented as a money value, an amount of staff-time or as a percentage of PV

Where tasks have been started but are not yet complete, some consistent method of assigning an earned value must be applied. Common methods in software projects are:

- **the 0/100 technique:** where a task is assigned a value of zero until such time that it is completed when it is given a value of 100% of the budgeted value;
- **the 50/50 technique:** where a task is assigned a value of 50% of its value as soon as it is started and then given a value of 100% once it is complete – this matches some contractual arrangements where a contractor is given half the agreed price when starting the work, perhaps to help pay for raw materials, and the remainder on successful completion;
- **the 75/25 technique:** where the task is assigned 75% on starting and 25% on completion – this is often used when a large item of equipment is being bought: 75% is paid when the equipment is actually delivered and the remainder when installation and testing has been satisfactorily completed;
- **the milestone technique:** where a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan;
- **percentage complete:** in some cases there may be a way of objectively measuring the amount of work completed – for example, as part of the implementation of an information system, a number of data records have to be manually typed into a database and the actual number so far completed can be objectively counted.
- **The baseline budget** The first stage in setting up an earned value analysis is to create the baseline budget. The baseline budget is based on the project plan and shows the forecast growth in earned value through time. Earned value may be measured in monetary values but, in the case of staff-intensive projects such as software development, it is common to measure earned value in person-hours or workdays
- **Monitoring earned value** Having created the baseline budget, the next task is to monitor earned value as the project progresses. This is done by monitoring the completion of tasks (or activity starts and milestone achievements in the case of the other crediting techniques). As well as recording EV, the actual cost of each task can be collected as actual cost (AC). This is also known as the actual cost of work performed (ACWP).
- **Schedule variance (SV)** The schedule variance is measured in cost terms as $EV - PV$ and indicates the degree to which the value of completed work differs from that planned
- **Time variance (TV)** is the difference between the time when the achievement of the current earned value was planned to occur and the time now.
- **Cost variance (CV)** This is calculated as $EV - AC$ and indicates the difference between the earned value or budgeted cost and the actual cost of completed work
- **Performance ratios** Two ratios are commonly tracked: the cost performance index ($CPI = EV/AC$) and the schedule performance index ($SPI = EV/PV$).



Performance Ratios

► Tasks

- Specify module 5 days
- Code module 8 days
- Test module 6 days



PV = 19 days

- If everything is completed except testing then EV = 13 days

- If 'Specify module' actually took 3 days and 'Code module' actually took 4 days then Actual cost = 7 days

Madhu Bhan

Schedule variance = EV-PV(BCWP-BCWS) i.e. 13-19 = -3

Schedule performance indicator (SPI) = EV/PV = 13/19 = 0.68

Cost variance (CV) = EV-AC i.e. 13-7 = 6 days

Cost performance indicator(CPI) = EV/AC=13/7 = 1.86

- A negative schedule variance (SV) means that the project is behind schedule as does a SPI that is less than 1.0

- For SPI and CPI, a value greater than one indicates that work is being completed better than planned whereas a value of less than one means that work is costing more than and/or proceeding more slowly than planned.

Dr. Madhu Bhan

MCAE61

28

Prioritizing Monitoring

- **Critical path activities** : A delay in the activity on critical path(float=0) will cause a delay in completion date for the project. Critical path activities are therefore likely to have a very high priority for close monitoring
- **Activities with no free float** : Free float is the amount of time an activity may be delayed without affecting any subsequent activity. A delay in activity with no free float will delay subsequent activities, although the project end date may not be directly threatened. Sometimes the delay of subsequent activities could mean the resources for that activity will be unavailable because they are scheduled somewhere else. Hence close monitoring is required for activities with no free float.
- **Activities with little float** : Activities with small floats are the most likely to find themselves turned into activities on the critical path if their floats get eroded.
- **High risk activities**: Identify the high risk activities. For example using the PERT three estimate approach , activities with high standard deviation are considered as high risk activities. Close attention needs to be given to such activities for avoiding overrun or overspend.
- **Activities using critical resources**: some resources may only be available for a limited period and if the activities that need the resource are delayed the resource could become unavailable.

Getting the Project Back to Target

Almost any project will, at one time or another, be subject to delays and unexpected events. One of the tasks of the project manager is to recognize when this is happening (or, if possible, about to happen) and, with the minimum delay and disruption to the project team, attempt to mitigate the effects of the problem. In most cases, the project manager, at least initially, tries to ensure that the scheduled project end date remains unaffected. This can be done by shortening remaining activity durations or shortening the overall duration of the remaining project in the ways described in the next section.

There are two main strategies to consider when drawing up plans to bring a project back on target – shortening the critical path or altering the activity precedence requirements.

1. **Try to shorten critical path**: The overall duration of the project is determined by the critical path, so speeding up of critical path activities can shorten the completion time of project.
 - a. **Adding Resources**: Encourage the staff to Work harder, Buy in more staff
 - b. **Increase the use of current resources**: Staff might be asked to work overtime, computing resources to be made accessible

- c. **Reallocate staff to critical activities:** Allocating more efficient staff to critical activities, swapping resources from non-critical activities to critical activities
 - d. **Reduce Scope:** The amount of work to be done could be reduced. The client may prefer to have subset of promised features-especially if they are the most useful ones-rather than have to deliver of the whole application delayed
 - e. **Reduce Quality:** Some quality related activities such as system testing could be curtailed. Can be done later once the system is implemented.
2. **Reconsider activity dependencies:** Consider activities which have to be deferred because of pending completion of others.
- a. **Over-lap the activities** so that the start of one activity does not have to wait for completion of another: For example in some organization it is a normal routine to complete testing and then start User Training. In order to avoid late completion of the project it might however be considered acceptable to alter normal practice and start training earlier.
 - b. **Split activities:** To overcome precedence constraints, subdivide the activity into components that can be started immediately. For example a draft user handbook can be drawn from the system specifications and then can be revised later to take account of subsequent changes.

Change Control

Change Control is a formal process used to ensure that changes to a product or system are introduced in a controlled and coordinated manner. It reduces the possibility that unnecessary changes will be introduced to a system.

Inconsistencies in program specification become apparent only when the program is coded. This would result in amendments in specification document. Similarly the requirement document may also undergo frequent changes. At some point a final version is created. This is baselined, effectively frozen. Baselined products are the foundation for the development of further products. Interface design documents may be developed from baselined user requirements. Any change to the baselined products could have a knock-on effect on other parts of the project.

Steps in Change Control Process

1. One or more users might perceive the need for a change.
2. User management decide that the change is valid and worthwhile and pass it to development management.
3. A developer would receive the Request For Change(RFC) and is assigned to assess the practicality and cost of making the change.
4. Development management report back to user management on the cost of the change;
5. The project board/Change control board considers and approves the changes up to certain level of expenditure. The larger the amendment the higher in the control hierarchy it would have to be reported.
6. One or more developers are authorized to make copies of components that is to be modified.
7. The copies are modified. After initial testing, a test version might be released to users for acceptance testing
8. When users are satisfied then operational release is authorized – master copies of components are updated

Configuration librarian's role

Control of changes and documentation ought to be the responsibility of someone who may variously be named the configuration librarian, the configuration manager or the project librarian. Among this person's duties would be:

- the identification of all items that are subject to change control;
- the establishment and maintenance of a central repository of the master copies of all project documentation and software products;
- the setting up and running of a formal set of procedures to deal with changes;
- the maintenance of records of who has access to which library items and the status of each library item (e.g. whether under development, under test or released)

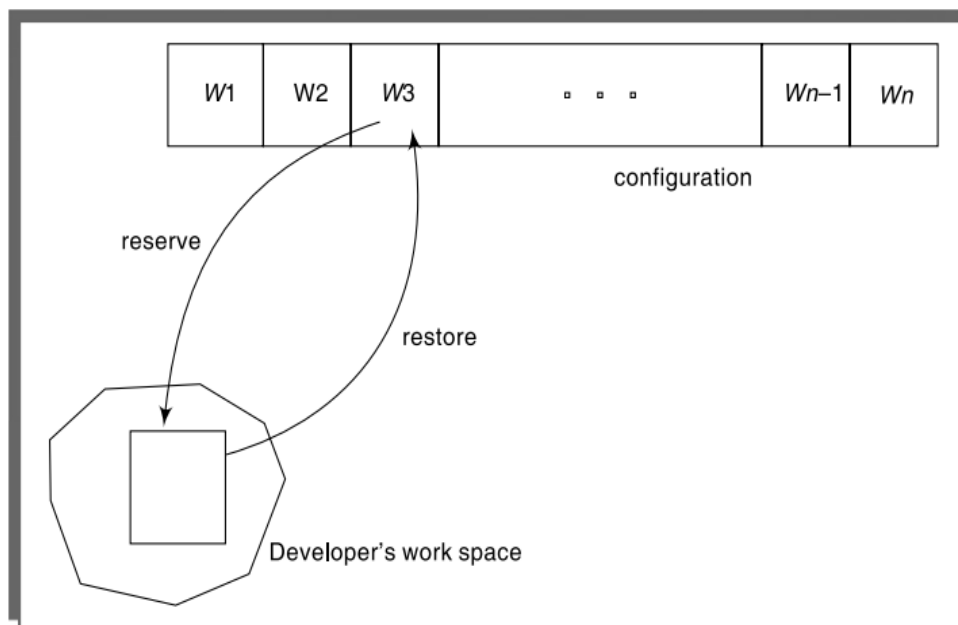
Software Configuration Management (SCM)

There are many available tools, for effective configuration management. The basic approach is to create a software repository, usually on a network drive or online, and then to keep a working copy of the software on your local hard drive. Software changes are made in the working copy and then checked in to the repository.

For example, if your latest change introduced an error, it is easy to revert back to the previous version or to compare the current version of the software to the previously working version. Generally, it is also possible to label, or tag, a particular version of the software, for example, to indicate a particular release.

Configuration management process Configuration management is carried out through the following two principal activities:

- **Configuration Identification:** This activity involves deciding which parts of the system should be kept under configuration management.
Project managers normally classify the work products associated with a software development process into three main categories, viz., controlled, pre-controlled, and uncontrolled.
Controlled work products are those that are put under configuration control. The team members must follow some formal procedures to change these.
Pre-controlled work products are not yet under configuration control, but will eventually be under configuration control.
Uncontrolled work products will not be subject to configuration control. Controllable work products include both controlled and pre-controlled work products.
Typical controllable work products include the following:
 - Requirements specification document
 - Design documents
 - Tools used to build the system such as compilers, linkers, lexical analysers, parsers, etc.
 - Source code for each module
 - Test cases
 - Problem report
- **Configuration Control:** This activity is used to ensure that changes to a system occur smoothly.
Configuration control is part of a configuration management system that most directly affects the day-to-day operations of developers. Configuration control allows only authorized changes to the controlled objects and prevents unauthorized changes. The project manager can give permission to some members to be able to change or access specific work products.
In order to change a controlled work product such as a code module, a developer can get a private copy of the module through a reserve operation (see Fig. 9.15). Configuration management tools allow only one team member to reserve a module at any time. Once a work product is reserved, it does not allow anyone else to reserve this module until the reserved module is restored. Thus, by preventing more than one developer to simultaneously reserve a module, the problems associated with concurrent access are taken care of



9.15 Work product modifications under configuration management

Modifications to a work product under configuration control When developers need to change a work product they first make a reserve request. A reserve request by a team member is honoured only if appropriate authorization has been given by the project manager to that member for the specific work product. After the reserve command successfully executes, a private copy of the work product is created in their local directory. Then, they can carry out all necessary changes to the work product on their private copy. Once they have satisfactorily completed all necessary changes, the changes need to be restored in configuration management repository.

The CCB is usually constituted from among the development team members. For every change that needs to be carried out, the CCB reviews the changes made to the controlled work product and certifies certain aspects about the change such as

- Change is well-motivated
- Developer has considered and documented the effects of the change
- Changes interact well with the changes made by other developers
- Appropriate people (CCB) have validated the change, e.g., someone has tested the changed code, and has verified that the change is consistent with the need