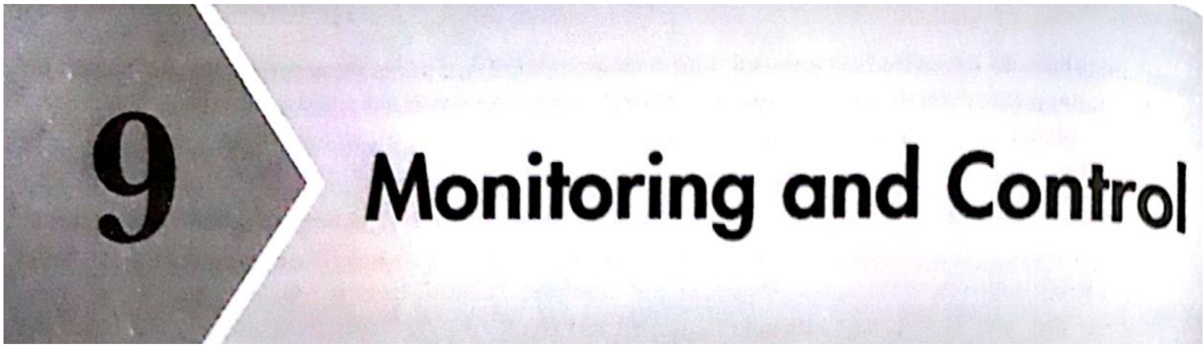
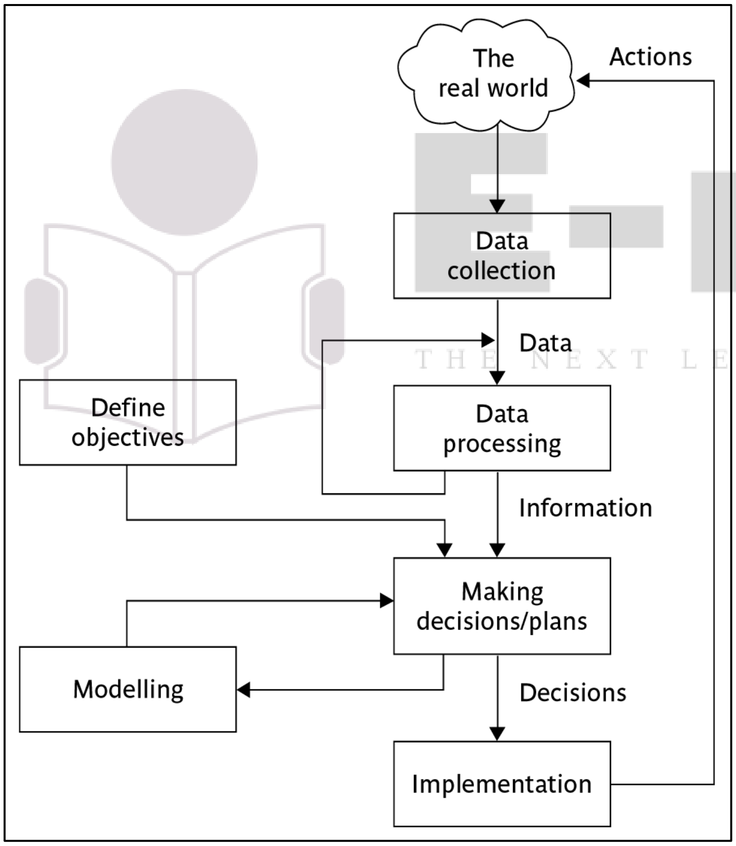
Unit IV



9 Monitoring and Control

The control cycle



Define objectives – at the beginning of the project we decide on what we want to achieve

Making decisions/plans – we decide how we are going to achieve the objectives i.e. we create a plan

Modelling – as part of the process of creating a plan we will consider different approaches and attempt to assess the consequences of each of these approaches in terms of how much it will cost and how long it will take, and so on.

Implementation – the plan is now carried out

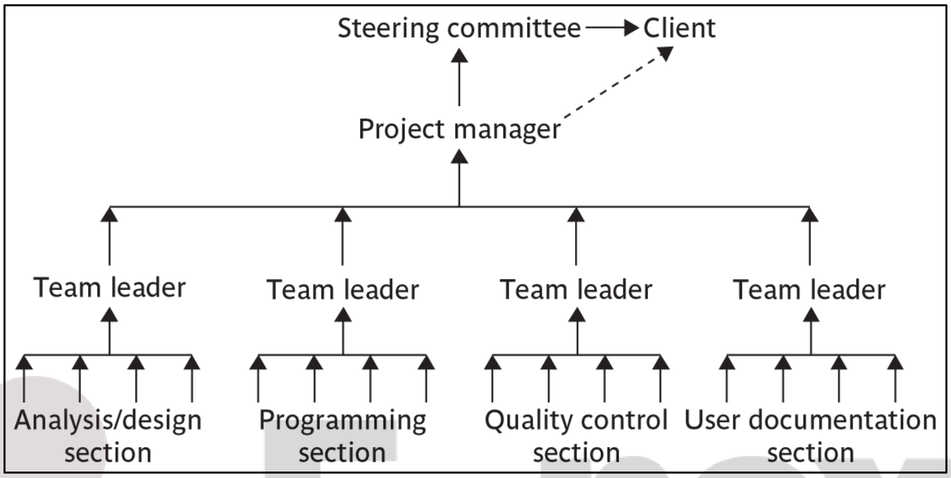
Data collection – we gather information at regular intervals about how the project is progressing. These raw details could be quite numerous and complex on a large project

Data processing – we process the progress data and convert it into 'information' which makes it easier for the project managers and others to understand the overall condition of the project

Making decisions/plans – in the light of the comparison of actual project progress with that planned, the plans are modified. This may require the modelling of the outcomes of different possible courses of action

…and so the cycle goes on.

Responsibilities



The concept of a reporting hierarchy was introduced in Chapter 1.

The main lesson here is that the details relating to project progress have to originate with the people actually doing the work and have then to be fed up through the management structure. At each management level there is going to be some summarising and commentary before information is passed up to the next level. This means that there is always a danger of 'information overload' as information passes from the many to the few.

Assessing progress

Checkpoints – predetermined times when progress is checked

- Event driven: check takes place when a particular event has been achieved
- Time driven: date of the check is pre-determined

Frequency of reporting

The higher the management level then generally the longer the gaps between checkpoints

Collecting progress details

Need to collect data about:

- Achievements
- Costs

A big problem: how to deal with *partial completions*

99% completion syndrome

Possible solutions:

- Control of products, not activities
- Subdivide into lots of sub-activities

- Projects have to be delivered on time and within budget, hence the concern with monitoring achievements and costs.
- Partial completion is where, for example, data is being collected at the end of Week 2 of an activity that should take four weeks. We want to know if it is about 50% completed.
- An example of the '99% completion syndrome' would be in the above case if the developer reported at the end of weeks 1,2 and 3 that the task was respectively 25%, 50% and 75% complete. However at the end of week 4 it is reported that the task is 99% complete. The same thing is reported at the end of week 5 and so on until the task is actually completed.
- Control on products implies that actual examination of intermediate allows us to verify independently and objectively that sub-tasks have been completed.

Red/Amber/Green reporting

- Identify key tasks
  - Break down into sub-tasks
- Assess subtasks as:

Green – 'on target'

Amber – 'not on target but recoverable'

Red – 'not on target and recoverable only with difficulty'

- Status of 'critical' tasks is particularly important

- RAG reporting highlights those activities which need particular attention. The status of a troubled activity might typically move from green to amber; if corrective action is possible it might go back to green, otherwise it could switch to red. If there are lots of instances where activities switch directly from green to red, this could indicate more management control.
- 'Critical tasks' would be those on the critical path and/or reliant on critical resources.

Review

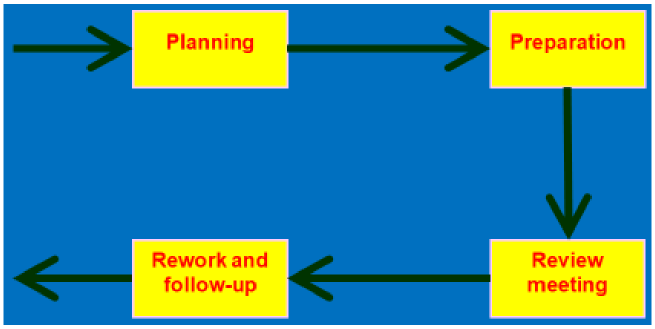
- Review of work products is an important mechanism for monitoring the progress of a project and ensuring the quality of the work products.
- Testing is an effective defect removal mechanism.
    - However, testing is applicable to only executable code.
    - Review is applicable to all work products.

Utility of Review

- A cost-effective defect removal mechanism.
- Review usually helps to identify any deviation from standards.
- Reviewers suggest ways to improve the work product
- a review meeting often provides learning opportunities to not only the author of a work product, but also the other participants of the review meeting.
- The review participants gain a good understanding of the work product under review, making it easier for them to interface or use the work product in their work.

Review Roles

- Moderator:
    - Schedules and convenes meetings, distributes review materials, leads and moderates review sessions.
- Recorder:
    - Records the defects found and the time and effort data.
- Reviewers.



Review Process

Project Termination Review

- Project termination reviews provide important opportunities to learn from past mistakes as well as successes.
- Project termination need not necessarily mean project failure or premature abandonment.
    - A project may be terminated on successful completion

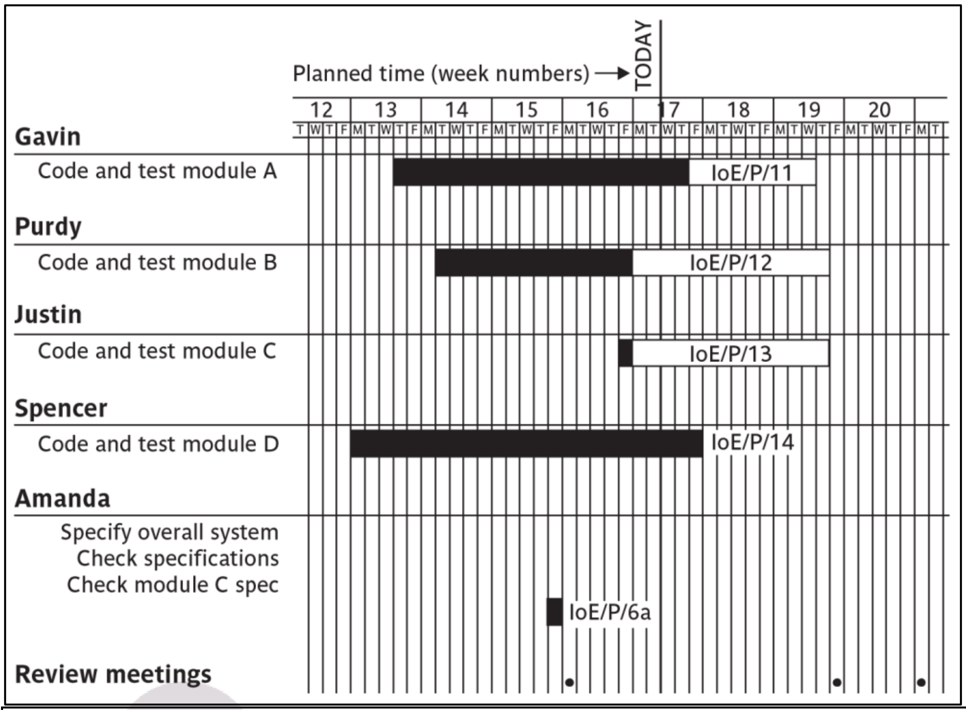
Reasons for Project Termination

- Project is completed successfully handed over to the customer.
- Incomplete requirements
- Lack of resources
- Some key technologies used in the project have become obsolete during project execution
- Economics of the project has changed, for example because many competing product may have become available in the market.

Project Termination Process

- Project survey
- Collection of objective information
- Debriefing meeting
- Final project review
- Result publication

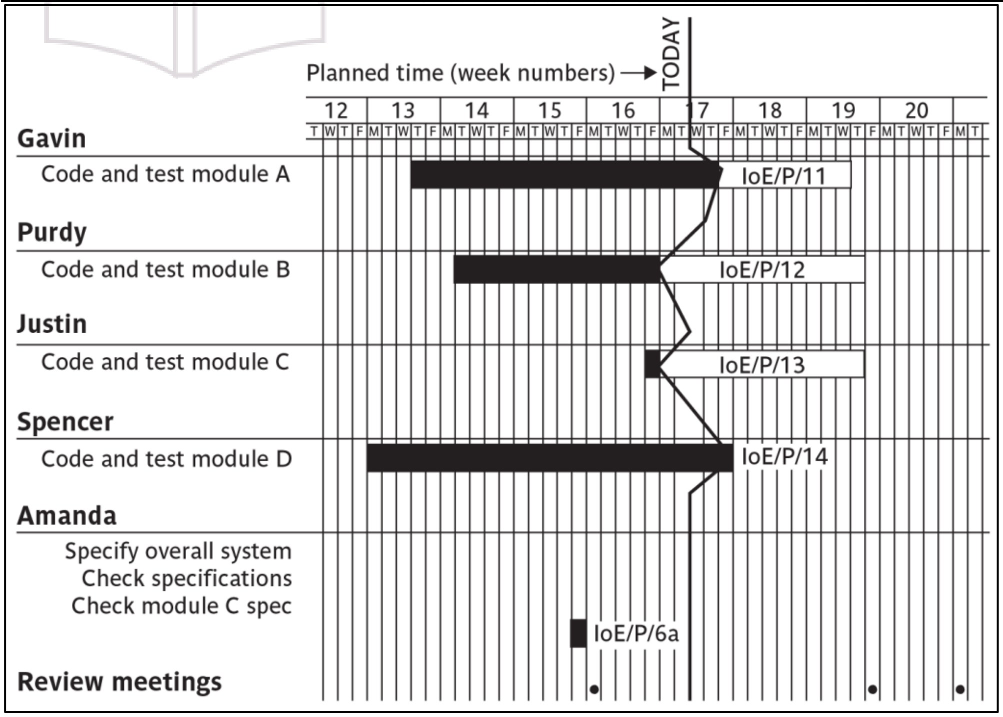
Gantt charts



Note that the Gantt chart is named after Henry Gantt (1861-1919) and so should not be written in capitals!

The format of the Gantt chart here differs from the format used in Microsoft project as the activities for each team member are grouped together. You could input the details so that they came out in this format, but it would not occur automatically.

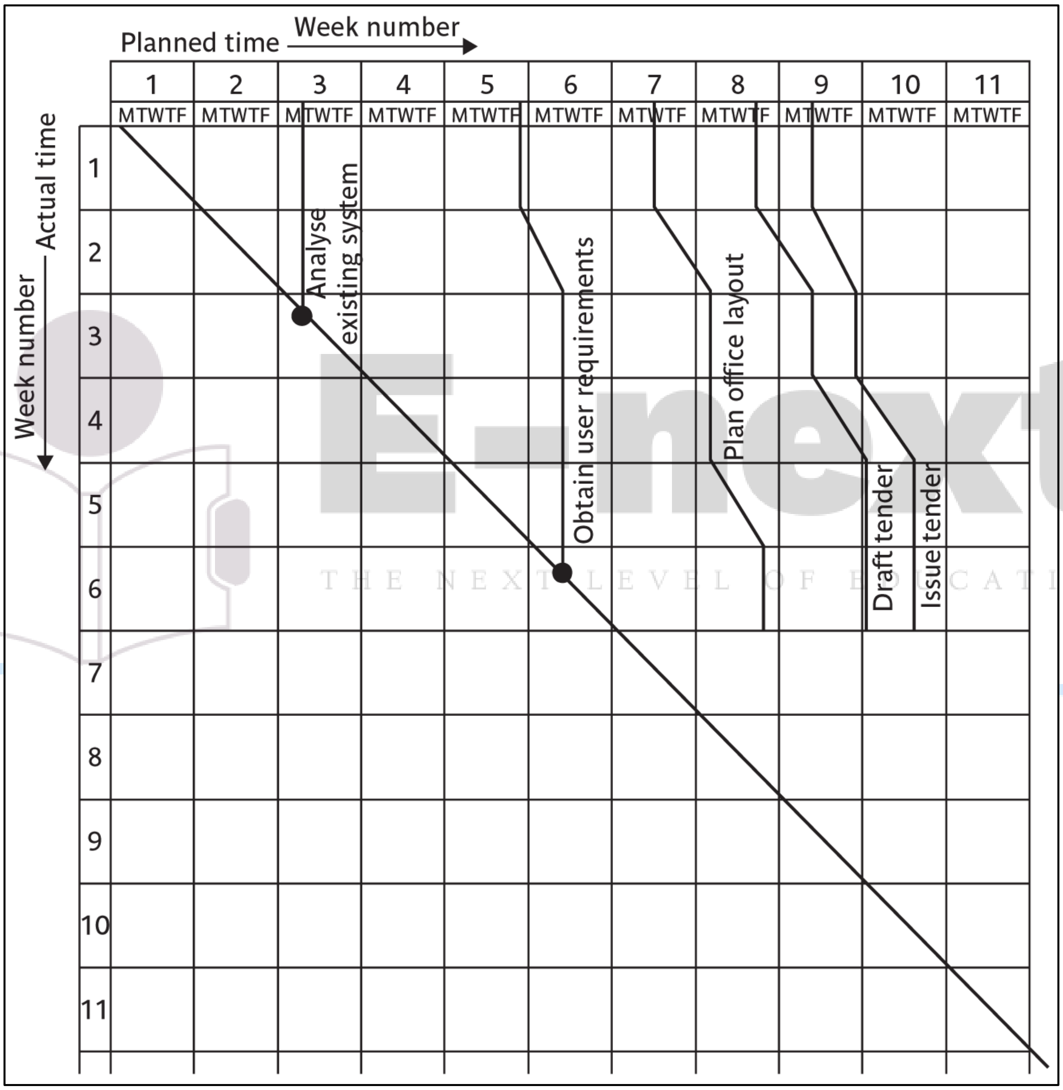
Slip charts



A slip chart is a version of the Gantt chart where a line is drawn from top to bottom. To the left of the line are all the completed activities and to the right those activities (or parts of activities) that have not been completed.

The more jagged the line, the more it means that that there are some activities that are lagging to various degrees and some that are ahead of themselves. A very jagged line means that there is scope for re-planning to move resources from those activities that are ahead to those that are behind.

The timeline



This records the way that targets have changed throughout the project.

Planned time is plotted on the horizontal axis, and actual time on the vertical axis. The bendy lines going from top to bottom represent the scheduled completion date for each activity e.g.

‘analyse existing system’ – at start this was due finish on the Monday of week 3 and it did finish then

‘obtain user requirements’ was originally planned to finish on the Thursday of week 5, but at the end of the first week it was rescheduled to finish on the Tuesday of week 6.

Cost monitoring

- A project could be late because the staff originally committed have not been deployed
- In this case the project will be *behind time* but *under budget*
- A project could be on time but only because additional resources have been added and so be *over budget*
- Need to monitor both achievements and costs

Earned value analysis

- *Planned value (PV)* or *Budgeted cost of work scheduled (BCWS)* – original estimate of the effort/cost to complete a task (compare with idea of a ‘price’)
- *Earned value (EV)* or *Budgeted cost of work performed* (BCWP) – total of PVs for the work completed at this time

Accounting conventions

- Work completed allocated on the basis
  - *50/50* half allocated at start, the other half on completion. These proportions can vary e.g. 0/100, 75/25 etc
  - *Milestone* current value depends on the milestones achieved
  - *Units processed*
- Can use money values, or staff effort as a surrogate

Earned value – an example

- Tasks

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

- At the beginning of day 20, PV = 19 days
- If everything but testing completed EV = 13 days
- Schedule variance = EV-PV i.e. 13-19 = -6
- Schedule performance indicator (SPI) = 13/19 = 0.68
- SV negative or SPI <1.00, project behind schedule

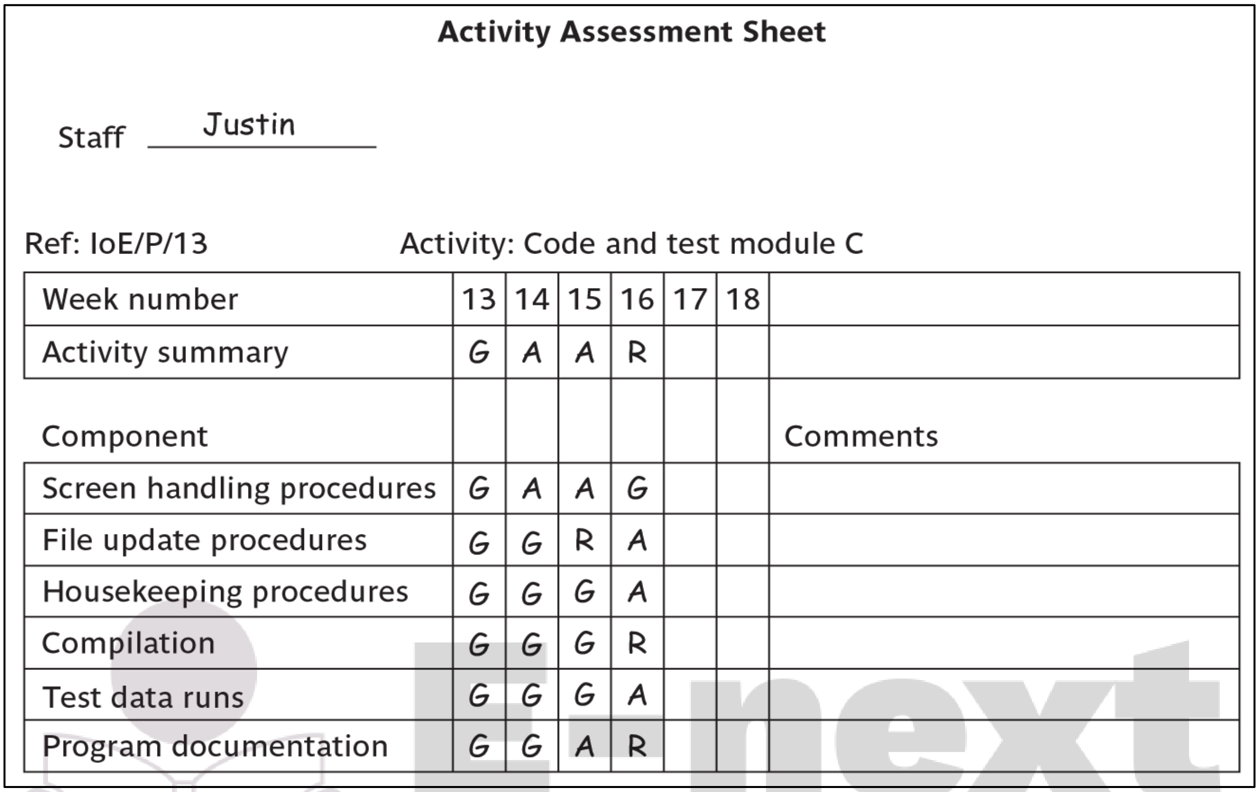
Earned value analysis – actual cost

- Actual cost (AC) is also known as Actual cost of work performed (ACWP)
- In previous example, if
  - ‘Specify module’ actually took 3 days
  - ‘Code module’ actually took 4 days
- Actual cost = 7 days
- Cost variance (CV) = EV-AC i.e. 13-7 = 6 days
- Cost performance indicator = 13/7 = 1.86
- Positive CV or CPI > 1.00 means project within budget
- CPI can be used to produce new cost estimate
- Budget at completion (BAC) – current budget allocated to total costs of project
- Estimate at completion (EAC) – updated estimate = BAC/CPI
  - e.g. say budget at completion is £19,000 and CPI is 1.86
  - EAC = BAC/CPI = £10,215 (projected costs reduced because work being completed in less time)

Time variance

- Time variance (TV) – difference between time when specified EV should have been reached and time it actually did reach.
- For example say an EV of £19000 was supposed to have been reached on 1st April and it was actually reached on 1st July then TV = - 3 months

Earned value chart with revised forecasts



Activity Assessment Sheet

Staff Justin

Ref: IoE/P/13 Activity: Code and test module C

Week number	13	14	15	16	17	18	
Activity summary	G	A	A	R			
Component							Comments
Screen handling procedures	G	A	A	G			
File update procedures	G	G	R	A			
Housekeeping procedures	G	G	G	A			
Compilation	G	G	G	R			
Test data runs	G	G	G	A			
Program documentation	G	G	A	R			

This shows how the planned value (PV), earned value (EV) and actual cost (AC) can be tracked over the lifetime of a project.

It also shows how the graph can be used to show adjustments to the final estimated cost and duration. A revised assessment of the budget at completion (EAC estimate at completion) can be produced by dividing the original estimated budget at completion (BAC) by the current CPI.

Similarly a forecast of the actual duration of the project can be derived by dividing the original estimated duration by the SPI.

Prioritizing monitoring

We might focus more on monitoring certain types of activity e.g.

- Critical path activities
- Activities with no free float – if delayed later dependent activities are delayed
- Activities with less than a specified float
- High risk activities

- Activities using critical resources

- **Critical path activities** – by definition if these are late then the project as a whole will be delayed
- **Activities with no free float** – free float was defined in Chapter 6. A project with no free float will delay following dependent activities, although the project end date may not be directly threatened.
- **Activities with less than a specified float** – projects when being executed can be **very dynamic**: some activities will take longer than estimated others less; this could lead to the critical shifting. 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** – recall the calculation of activity standard deviations in Chapter 7. If the standard deviation for an activity is large, this indicates that there is a lot of uncertainty about how long it will actually take.

  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 back on track: options

- Renegotiate the deadline – if not possible then
- Try to shorten critical path e.g.
  - Work overtime
  - Re-allocate staff from less pressing work
  - Buy in more staff
- Reconsider activity dependencies
  - Over-lap the activities so that the start of one activity does not have to wait for completion of another
  - Split activities

- **Renegotiating the deadline** – one way of doing this is to divide the deliverables into ‘tranches’ (see Chapter 3), delivering the ones most valuable to the client on or before the deadline, but delaying less valuable ones.
- **Shortening the critical path** – the idea is to try to get things done more quickly by adding more staff. Some activities lend themselves to this more readily than others – it is often quite difficult to do this with software development. It also increases costs
- **Reconsidering activity dependencies** – allowing activities to overlap often increases the risk of quality shortfalls

Exception planning

- Some changes could affect
    - Users
    - The business case (e.g. costs increase reducing the potential profits of delivered software product)
- These changes could be to
    - Delivery date
    - Scope
    - Cost
- In these cases an **exception report** is needed
- First stage
    - Write an **exception report** for sponsors (perhaps through project board)
        - Explaining problems
        - Setting out options for resolution
- Second stage
    - Sponsor selects an option (or identifies another option)
    - Project manager produces an **exception plan** implementing selected option
    - Exception plan is reviewed and accepted/rejected by sponsors/Project Board

Change control

The role of configuration librarian:

- Identifying items that need to be subject to change control
- Management of a central repository of the master copies of software and documentation
- Administering change procedures
- Maintenance of access records

- *Identifying items that need to be subject to change control* – it is unlikely, for example, that a feasibility report would be subject to change control once agreement has been obtained to start the project
- *Management of a central repository of the master copies of software and documentation*
- *Administering change procedures* It is important that someone ensures that there is adherence to change control procedures.
- *Maintenance of access records*. A situation to be avoided is where two different developers are making changes to the same software component.

Typical 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 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; user management decide whether to go ahead
5. One or more developers are authorized to make copies of components to be modified
6. Copies modified. After initial testing, a test version might be released to users for acceptance testing
7. When users are satisfied then operational release authorized – master configuration items updated

- 1 and 2. The user community itself must come to a consensus about whether a proposal for a change should go forward. A change deemed desirable by one part of the user community could cause opposition with other users.
- 2 and 3. This part of the process often involves a multipart form, initially raised by a user representative and then completed with a response by the developers.
- 4. There could be a change control board with user and developer representatives that oversees this decision-making process
- 5. The configuration librarian would control this release
- 6. Note that it is a copy that is modified; the original would still exist as the current operational version
- 7. The previous version of the configuration items would be archived but preserved. If there are unforeseen problems with the new version when it is made operational then a fall-back to the previous version could be considered

Software Configuration Management (SCM)

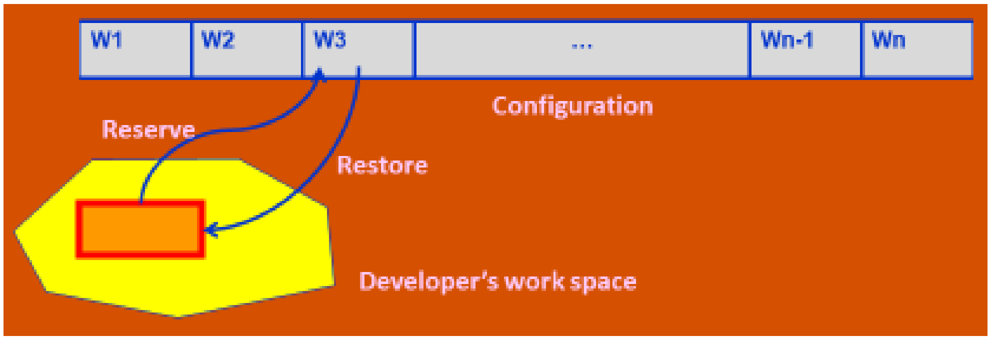
- SCM is concerned with tracking and controlling changes to a software.
- Development and maintenance environment:
  - Various work products associated with the software continually change.
  - Unless a proper configuration management system is deployed, several problems can appear.

Why Use SCM?

- Problems associated with concurrent access
- Undoing Changes
- System accounting
- Handling variants
- Accurate determination project status
- Preventing unauthorized access to the work products

Configuration Control

- Two main operations:
  - Reserve
  - Restore



10 Managing Contracts

Acquiring software from external supplier

This could be:

- a *bespoke system* - created specially for the customer
- *off-the-shelf* - bought 'as is'
- *customised off-the-shelf* (COTS) - a core system is customised to meet needs of a particular customer

Section 10.2

Types of contract

- fixed price contracts
- time and materials contracts
- fixed price per delivered unit

Note difference between goods and services

Often licence to use software is bought rather than the software itself

Section 10.4 of the textbook provides more detail about the types of contract.

Fixed price contracts

Advantages to customer

- known expenditure
- supplier motivated to be cost-effective

Disadvantages

- supplier will increase price to meet contingencies
- difficult to modify requirements
- cost of changes likely to be higher
- threat to system quality

Even though the supplier will have to add a margin to the price to deal with contingencies, the cost could still be less than doing the work in-house as the supplier may be able to exploit economies of scale and the expertise that the have from having done similar projects in the past.

When competing for work, there will be pressure on the suppliers to reduce prices. Once a contract has been won and signed, the contractor is in a stronger negotiating position when it come to negotiating the price of additional work as the customer is now locked in.

Time and materials

Advantages to customer

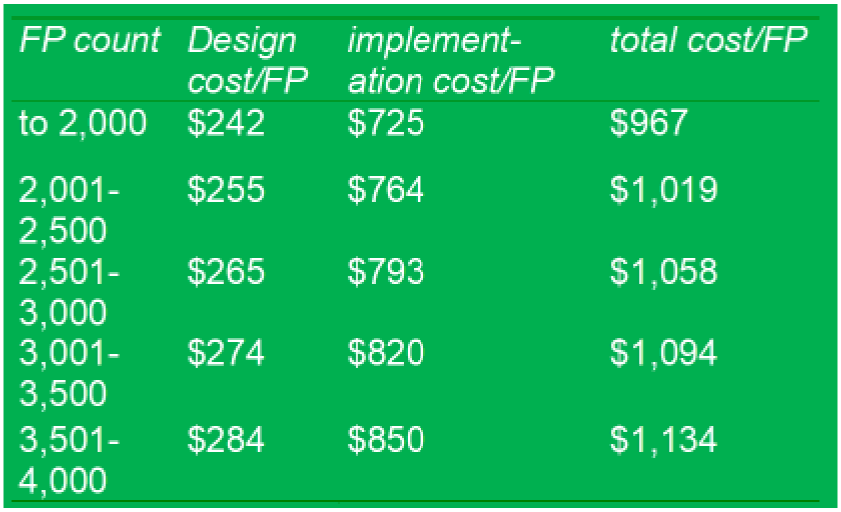
- easy to change requirements
- lack of price pressure can assist product quality

Disadvantages

- Customer liability - the customer absorbs all the risk associated with poorly defined or changing requirements
- Lack of incentive for supplier to be cost-effective

Because suppliers appear to be given a blank cheque, this approach does not normally find favour with customers. However, the employment of contract developers may involve this type of contract.

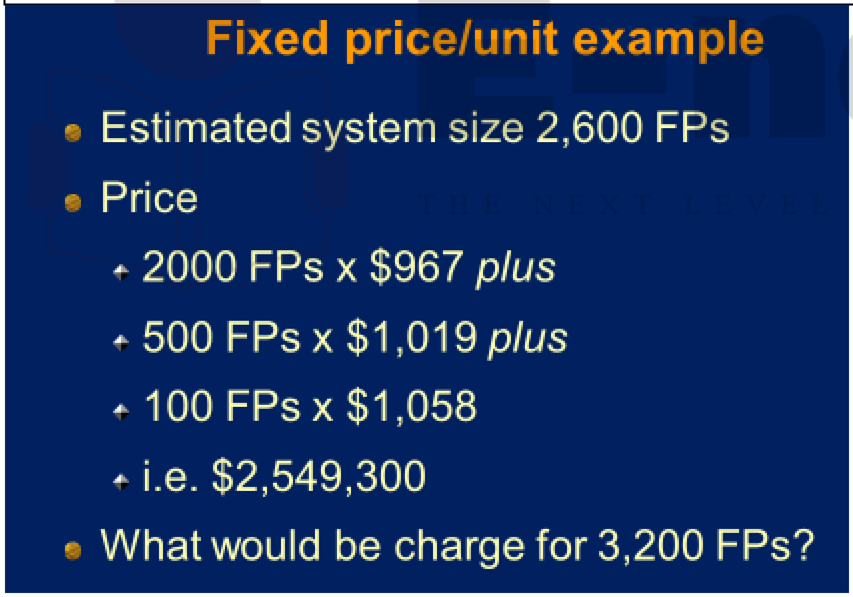
Fixed price per unit delivered



FP count	Design cost/FP	implement-ation cost/FP	total cost/FP
to 2,000	$242	$725	$967
2,001-2,500	$255	$764	$1,019
2,501-3,000	$265	$793	$1,058
3,001-3,500	$274	$820	$1,094
3,501-4,000	$284	$850	$1,134

This is Table 10.1

These figures do come from a real source (RDI Technologies in the USA). These are now several year old. The bigger the project, the higher the cost per function point. Recall that function points were covered in Chapter 5 on software effort estimation.



Fixed price/unit example

- Estimated system size 2,600 FPs
- Price
  - 2000 FPs x $967 *plus*
  - 500 FPs x $1,019 *plus*
  - 100 FPs x $1,058
  - i.e. $2,549,300
- What would be charge for 3,200 FPs?

2000 FPs at	$967 =	$1,934,000
500 FPs at	$1019 =	$509,500
500 FPs at	$1058 =	$529,000
200 FPs at	$1094	$218,800
total		**$3,191,300**

Advantages for customer

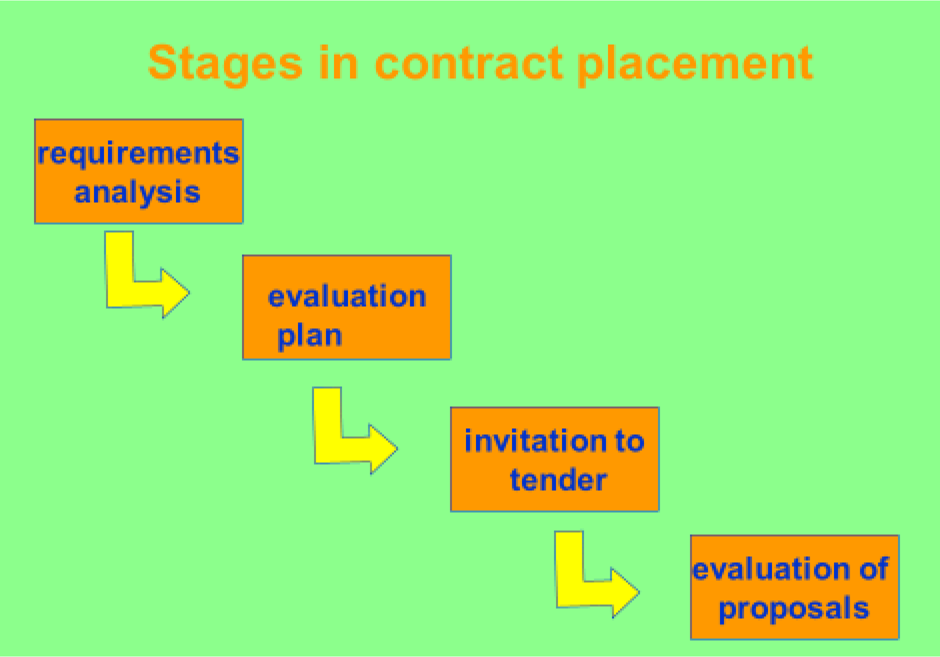
- customer understanding of how price is calculated
- comparability between different pricing schedules
- emerging functionality can be accounted for
- supplier incentive to be cost-effective

Disadvantages

- difficulties with software size measurement - may need independent FP counter
- changing (as opposed to new) requirements: how do you charge?

The tendering process

- **Open tendering**
  - any supplier can bid in response to the *invitation to tender*
  - all tenders must be evaluated in the same way
  - government bodies may have to do this by local/international law (including EU and WTO, World Trade Organization, requirements
- **Restricted tendering process**
  - bids only from those specifically invited
  - can reduce suppliers being considered at any stage
- **Negotiated procedure**
  - negotiate with one supplier e.g. for extensions to software already supplied



Stages in contract placement

requirements analysis

evaluation plan

invitation to tender

evaluation of proposals

Requirements document: sections

- introduction
- description of existing system and current environment
- future strategy or plans
- system requirements -
  - mandatory/desirable features
- deadlines
- additional information required from bidders

- The requirements document is sometimes referred to as the operational requirement or OR.
- If a mandatory requirement cannot be met the proposed application would have to be rejected regardless of how good it might be in other ways.
- A shortfall in one desirable requirement might be compensated for by other qualities or features.

Requirements

- These should include
  - functions in software, with necessary inputs and outputs

  - standards to be adhered to
  - other applications with which software is to be compatible
  - quality requirements e.g. response times

Evaluation plan

- How are proposals to be evaluated?
- Methods could include:
  - reading proposals
  - interviews
  - demonstrations
  - site visits
  - practical tests

Off the shelf software clearly has an advantage here as there is actually product that can be evaluated in existence.

- Need to assess value for money (VFM) for each desirable feature
- VFM approach an improvement on previous emphasis on accepting lowest bid
- Example:
  - feeder file saves data input
  - 4 hours work a month saved at £20 an hour
  - system to be used for 4 years
  - if cost of feature £1000, would it be worth it?

£(4 x 10 x 12 x 4) would be saved i.e. £3,840. The payback period would be just over a year and so this feature would be worth the additional cost.

Invitation to tender (ITT)

- Note that bidder is making an *offer* in response to ITT
- *acceptance* of offer creates a *contract*
- Customer may need further information
- Problem of different technical solutions to the same problem

ISO 12207 refers to an ITT as a Request for Proposal or RFP.

Memoranda of agreement (MoA)

- Customer asks for technical proposals
- Technical proposals are examined and discussed
- Agreed technical solution in MoA
- Tenders are then requested from suppliers based in MoA
- Tenders judged on price
- Fee could be paid for technical proposals by customer

Contracts

- A project manager cannot be expected to be a legal expert – needs advice
- BUT must ensure contract reflect true requirements and expectations of supplier and client

Contract checklist

- Definitions – what words mean precisely e.g. ‘supplier’, ‘user’, ‘application’
- Form of agreement. For example, is this a contract for a sale or a lease, or a license to use a software application? Can the license be transferred?
- Goods and services to be supplied – this could include lengthy specifications
- Timetable of activities
- Payment arrangements – payments may be tied to completion of specific tasks
- **Ownership of software**

  - Can client sell software to others?
  - Can supplier sell software to others? Could specify that customer has ‘exclusive use’
  - Does supplier retain the copyright?
  - Where supplier retains source code, may be a problem if supplier goes out of business; to circumvent a copy of code could be deposited with an **escrow** service
- Environment – for example, where equipment is to be installed, who is responsible for various aspects of site preparation e.g. electricity supply?
- Customer commitments – for example providing access, supplying information
- Standards to be met

Contract management

Some terms of contract will relate to management of contract, for example,

- Progress reporting
- Decision points – could be linked to release of payments to the contractor
- Variations to the contract, i.e. how are changes to requirements dealt with?
- Acceptance criteria

How would you evaluate the following?

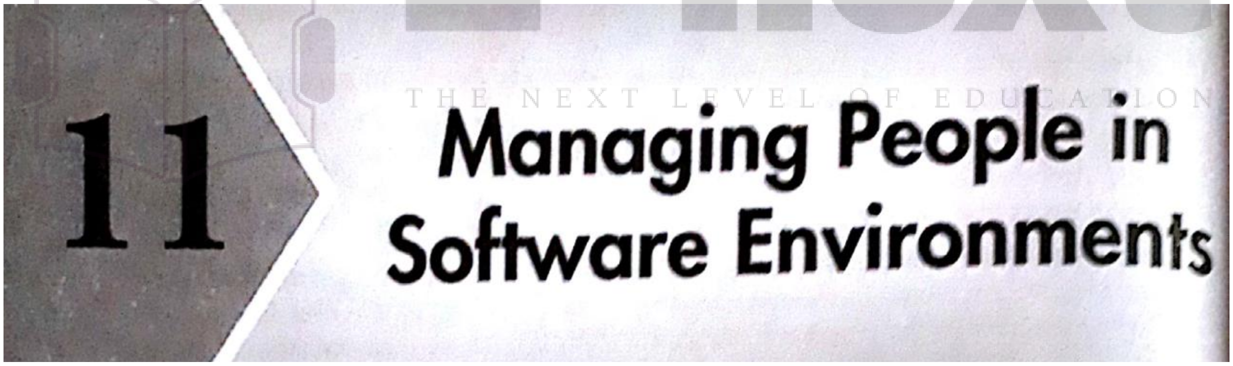
- usability of an existing package
- usability of an application yet to be built
- maintenance costs of hardware
- time taken to respond to requests for software support
- training

- Usability of existing package – you could try out a demo or ask existing users
- Usability of application to be built – here you would have to make stipulation about the process e.g. on the development of interface prototypes; you could also specify performance requirements

- Maintenance costs of hardware – this could be incorporated in a maintenance agreement and you could compare the terms offered by different potential suppliers; another approach is ask current users of the hardware about their experience of it
- Time taken to respond to support requests – this could once again be made a contractual matter and the terms offered by different suppliers could be compared; suppliers could be asked to supply evidence of their past performance (but they might refuse, or not tell the truth); you could ask for references from current customers of the supplier;
- Training – once again references could be taken up; you could ask for the CV of the trainer; you could even get them to give a short presentation

Contract management

- Contracts should include agreement about how customer/supplier relationship is to be managed e.g.
    - *decision points* - could be linked to payment
    - *quality reviews*
    - *changes to requirements*



11 Managing People in Software Environments

Main topics

- What is organizational behaviour?
- Staff selection and induction
- Models of motivation – focus on the individual
- The dark side of motivation - stress
- The broader issues of health and safety
- Some ethical and professional concerns

Before organizational behaviour

- Frederick Taylor (1856-1915) 'the father of scientific management'
- Focus:
  - To select the best people for the job;
  - To instruct them in the best methods;
  - To give financial incentives in the form of piece work
- One problem: 'group norms'

- Much of the work of Taylor was in factories and mines, working with manual workers. The 'instruction in best methods' involved breaking down a manual task into its component activities, identifying the best way of carrying out those activities and then teaching the workers to copy the approved method. This can be seen as treating the workers as little better than automatons – but it is also the way the sporting coaches often work!
- The individual workers were encouraged to maximize output by paying them piece-rates e.g. by the units processed.
- One difficulty with this is that workers learn that increasing output can in fact lead to the piece-rate being adjusted in a downward direction. Maximizing output can also be physically and mentally exhausting. Groups of workers therefore tend to converge on an agreed output rate which does not require a constant 100% effort.
- See Sections 11.2 and 11.3.

Hawthorne effect

- 1920's – series of experiments at the Hawthorne Plant of Western Electric, Chicago
- Found that simply showing an interest in a group increased productivity
- Theory X: there is a need for coercion, direction, and control of people at work
- Theory Y: work is as natural as rest or play

- The Hawthorne experiments investigated the effect of various factors such as improved lightning on productivity. It was found that the productivity of the control group (whose working conditions such as lighting were not changed) increased – the fact that someone singled them out for observation improved their motivation.
- Donald McGregor Theory X and Theory Y management approaches.
- See Section 11.3

Selecting the best people

- Belbin distinguishes between **eligible** (having the right qualifications) and **suitable** candidates (can do the job).
- The danger is employ someone who is eligible but not suitable
- The best situation is to employ someone who is suitable but not eligible! For example, these are likely to be cheaper and to stay in the job.

Do good software developers have innate characteristics?

- 1968 study – difference of 1:25 in time taken by different programmers to code program
- Other research found experience better than maths skills as a guide to software skills
- Some research suggested software developers less sociable than other workers
- Later surveys have found no significant social differences between IT workers and others – this could be result of broader role of IT in organizations

- There is some evidence that there is a very wide variation in software development skills – going back many years.
- Some research found that computer people had fewer social needs than other professionals. Later research has not found any significant difference – this may be because the 'ICT profession' has become broader in scope.

A selection process

1. **Create a job specification**.

Content includes types of task to be carried out.

2. **Create a job holder profile**

Describes the characteristics of the person who could do the job

3. **Obtain applicants**

Identify the media that potential job holders are likely to consult. Elicit CVs

4. **Select potential candidates from CVs.**

Do not waste everybody's time interviewing people whose CV clearly indicates are unsuitable.

5. **Further selection, including interview**

Selection processes could include aptitude tests, examination of work portfolios. Make sure selection processes map to the job holder profile

6. **Other procedures**.

e.g. taking up references, medicals etc

Instruction in the best methods

- The induction of new staff should be carefully planned – worst case where new recruit is simply ignored and not given any tasks
- Good induction leads to new recruit becoming productive more quickly
- Need to review staff progress frequently and provide feedback
- Need to identify training that could enhance staff effectiveness.

Section 11.5

Motivation

- Motivation and application can often make up for shortfalls in innate skills
- Taylor's approach - financial incentives
- Abraham Maslow (1908-1970)
  - motivations vary from individual to individual
  - hierarchy of needs – as lower ones fulfilled, higher ones emerge
  - Lowest level – food, shelter
  - Highest level – self-actualization

- Maslow's model implies that people will be motivated by different things at different times. Also that people always feel dissatisfied, but the focus of the dissatisfaction changes over time.
- This and other aspects of motivation are discussed in Section 11.6 of the text
- Exercises 11.2 and 11.3 are designed to get students to think a little about the implications of the Taylor and Maslow models of motivation.

Maslow’s Hierarchy of Needs

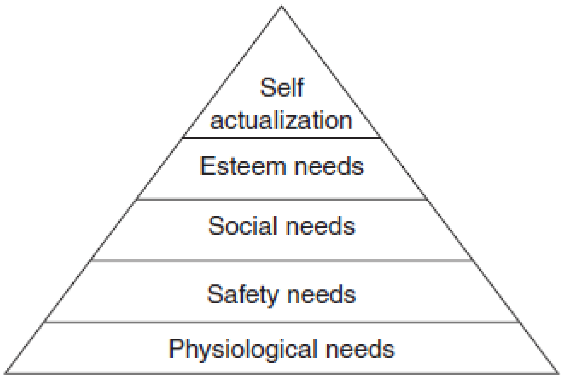


FIGURE 11.2 Maslow’s hierarchy of needs.

As a lower level of needs is satisfied then gradually a higher level of need emerges.

Herzberg

Herzberg suggested two sets of factors affected job satisfaction

1. *Hygiene* or *maintenance factors* – make you dissatisfied if they are not right e.g. pay, working conditions

2. *Motivators* – make you feel the job is worthwhile e.g. a sense of achievement

Exercise 11.4 illustrates the approach that Herzberg and associates used to gather data about motivational factors.

Vroom

Vroom and colleagues identified three influences on motivation

1. *Expectancy* – the belief that working harder leads to better performance

2. *Instrumentality* – the belief that better performance will be rewarded

3. *Perceived value* of the reward

- Note: if any of the factors has a zero value, then motivation will be zero.
- Example from the text book: expectancy – trying to use a compiler to compile software code; the code has a bug which causes a compilation error regardless of what you do. In this case motivation will collapse.
- Instrumentality – you are working on removing a fault from a software tool used by a client; you find that the client has given up using the tool and has acquired a different one to do the job. Low perceived value of reward: a reward that everyone gets is less highly regarded than one which only outstanding people get. Getting a first is more valuable if only 5% of students get a first compared to where 90% get a first!

Oldham-Hackman job characteristics

Identified the following characteristics of a job which make it more 'meaningful'

- Skill variety
- Task identity
- Task significance

Two other factors contributed to satisfaction:

- Autonomy
- Feedback

- Skill variety – number of different skills the job holder has the opportunity to exercise
- Task identity – the degree to which your work and its results are associated with you
- Task significance – the degree to which your job has an influence on others
- Two other factors contributed to satisfaction:
- Autonomy – the freedom that you have about the way that you do the job;
- Feedback – the information you get back about the results of your work.
- Software developers will tend to be associated with their code – task identity; analyst programmers will have a to use a wider range of skills than lower level programmers – more skill variety. If you have direct contact with the end-users of your software you are likely to be more aware of the results of your work – task significance, and more likely to get feedback on it.

Methods to improve job satisfaction

- Set specific goals
- Provide feedback on the progress towards meeting those goals
- Consider job redesign
  - Job enlargement
  - Job enrichment

- Job enlargement – widening the range of tasks carried out by a worker
- Job enrichment – delegating some management roles to the worker e.g for re-ordering raw materials.

Stress

- Edward Yourdon quotes a project manager: ‘*Once a project gets rolling, you should be expecting members to be putting in at least 60 hours a week….The project manager must expect to put in as many hours as possible.’*
- 1960 study in US: people under 45 who worked more than 48 hours a week twice the risk of death from coronary heart disease.
- XP practice – maximum 40 hour working week

Stress can be reduced by good project management

Good project management should lead to:

- Reasonable estimates of effort
- Good project control leading fewer unexpected crises
- Making clear what is expected of each team member – reduces **role ambiguity**
- Reduced **role conflict** where a person is torn between conflicting responsibilities

Bullying tactics are a symptom of incompetent project management.

Stress Management

- Imagery, relaxation, and meditation
  - An example of a simple relaxation technique can be rolling the head from side to side
- Cognitive behavioral approaches
  - Include self-monitoring of stress intensity, thought record-keeping and rewriting, time management, assertiveness training and increased social interactions.
- Systemic approach
  - Altering the factors which contribute to stress

Health and safety

- Apart from stress, health and safety less likely to be an issue compared to other engineering projects.

- …but sometimes IT infrastructure may be set up as other building work is going on
- UK law lays down that organizations employing over 5 staff should have a **written safety policy**
- Management of safety should be embedded in project management.
- Top management must be committed to health and safety (H&S) policy
- Delegation of responsibilities relating to H&S should be clear
- Job descriptions should include H&S related responsibilities
- Need to ensure those given H&S responsibilities should understand and accept them
- There should be a designated safety officer
- Staff, particularly knowledgeable technical specialists, should consulted about safety
- There should be an adequate H&S budget

Ethical and professional concerns

Ethics relates to the moral obligation to respect the rights and interests of others – goes beyond strictly legal responsibilities

Three groups of responsibilities:

- Responsibilities that everyone has
- Responsibilities that people in organizations have
- Responsibilities relating to your profession or calling

Organizational ethics

There are some who argue that ethical organizational ethics are limited:

Stockholder theory (e.g. Milton Friedman). An employee's duty is to the owners of the business (which often means the stakeholders) above all others – although legal requirements must be met.

Competitive relationships between businesses. Competition may cause you to do things that could have a negative impact on the owners or employees of competitive businesses

Exercise

Identify some of the possible objections and criticisms that can be made of the stockholder business ethics model described above.

Giving this exercise to students can be an interesting (and sometimes scary) experience. The argument against stockholder theory is that the work of an organization is not purely that of money-making. Banks, for example, have a social and economic role facilitating day to day commerce and business. If the owners of the banks decided to opt out of this role the community at large (through the government) would have to step in to ensure that that role is still carried out. Freidman's arguments unwittingly support a greater role for state ownership of important organizations.

Professional ethics

- Professionals have knowledge about the technical domain that the general public does not
- Ethical duty of the expert to warn lay people of the risks involved in a particular course of action
- Many professions, or would be professions, have codes of conduct for their members e.g.

<http://www.bcs.org/upload/pdf/cop.pdf>

<http://www.ieee.org/web/aboutus/ethics>

<http://www.apm.org/about/se_code>