# Data Science Joint Education Program

# INFO 200
# Information Systems Analysis

## Chapter #3

# Project Management (1)

# Keywords and Phrases

| | |
|---|---|
| actual cost (AC) | ecommerce project management |
| benchmarking | economic feasibility |
| break-even analysis | estimate at completion (EAC) |
| bring your own device (BYOD) | estimate to complete (ETC) |
| bring your own technology (BYOT) | expediting |
| budget | forecasting |
| budget at completion (BAC) | function point analysis |
| cloud computing | Gantt chart |
| cost performance index (CPI) | Infrastructure as a Service (IaaS) |
| cost variance (CV) | Hardware as a Service (HaaS) |
| critical path | intangible benefits |
| earned value (EV) | intangible costs |
| earned value management (EVM) | moving average |

# Keywords and Phrases

| | |
|---|---|
| operational feasibility  payback | tangible costs |
| planned value (PV) | task leader |
| Program Evaluation and Review Technique (PERT) | team motivation |
| problem definition | team norms |
| productivity goals | team process |
| project charter | technical feasibility |
| Platform as a Service (PaaS) | vendor support |
| schedule performance index (SPI) | work breakdown structure (WBS) |
| schedule variance (SV) | |
| socioemotional leader | |
| Software as a Service (SaaS) | |
| systems proposal | |
| tangible benefits | |

# Project Management Fundamentals

- Project initiation
- Determining project feasibility
- Activity planning and control
- Project scheduling
- Managing systems analysis team members

# Overview

- Chapter #3 provides an understanding of how:
  - Projects are initiated and selected
  - Define a business problem
  - Determine the feasibility of a proposed project
  - To evaluate hardware and software alternatives by addressing the trade-offs
  - To forecast and analyse tangible and intangible costs and benefits
  - To manage a project by preparing a budget, creating a work breakdown structure, scheduling  activities, and controlling the schedule and costs
  - To build and manage a project team
  - To write an effective systems proposal, concentrating on both content and design

# Major Topics Introduced

- Initiating projects, determining project feasibility, scheduling projects, estimating costs,  budgeting, and planning and then managing activities and team members for productivity (the project management fundamentals
- A systems project begins with problems or with opportunities for improvement in a  business that come up as the organization adapts to change.
- The increasing popularity of  ecommerce means that some fundamental changes are occurring as businesses either originate their enterprises on or move their internal operations and external relationships to the Internet
- Changes that require a systems solution occur in the legal environment  as well as in the industry's environment.
- A systems analyst works with users to create a  problem definition that reflects current business systems and concerns.
- Once a project is suggested, the systems analyst works quickly with decision makers  to determine whether the project is feasible

# Major Topics Introduced

- When a project is approved for a full systems study:
  - The project activities are scheduled through the use of tools such as:
    - Gantt charts
    - Program Evaluation and Review Technique (PERT) diagrams
  - The goal is to implement managed planning of the project from a time and cost perspective
- Part of ensuring the productivity of systems analysis team members is effectively managing their scheduled activities
  - This chapter is sets out a discussion of project management fundamentals

# Project Initiation

# Project Initiation

- Systems projects are initiated by many different sources for a variety of reasons
- Some of the  projects suggested:
  - Will survive various stages of evaluation to be worked on by you (or you and  your team)
  - Other proposed projects will not reach the evaluation stage (and should not)
- Systems projects are suggested by people in organizations for two broad reasons:
  1. Because they experience problems that lend themselves to systems  solutions
  2. Because they recognize opportunities for improvement through upgrading,  altering, or installing new systems
- Both situations can arise as an organization adapts to and  copes with natural, evolutionary change

# Problems in an Organization

# Project Initiation

- Managers do not like to:
    - Conceive of their organization as having problems
    - Talk  about those problems
    - Share them with someone from outside
- However - good managers:
    - Realize that recognizing symptoms of problems
    - Or (at a later stage) diagnosing the problems
    - Then confronting the problems
    - Is imperative if the business is to keep functioning at  its full potential

# Project Initiation

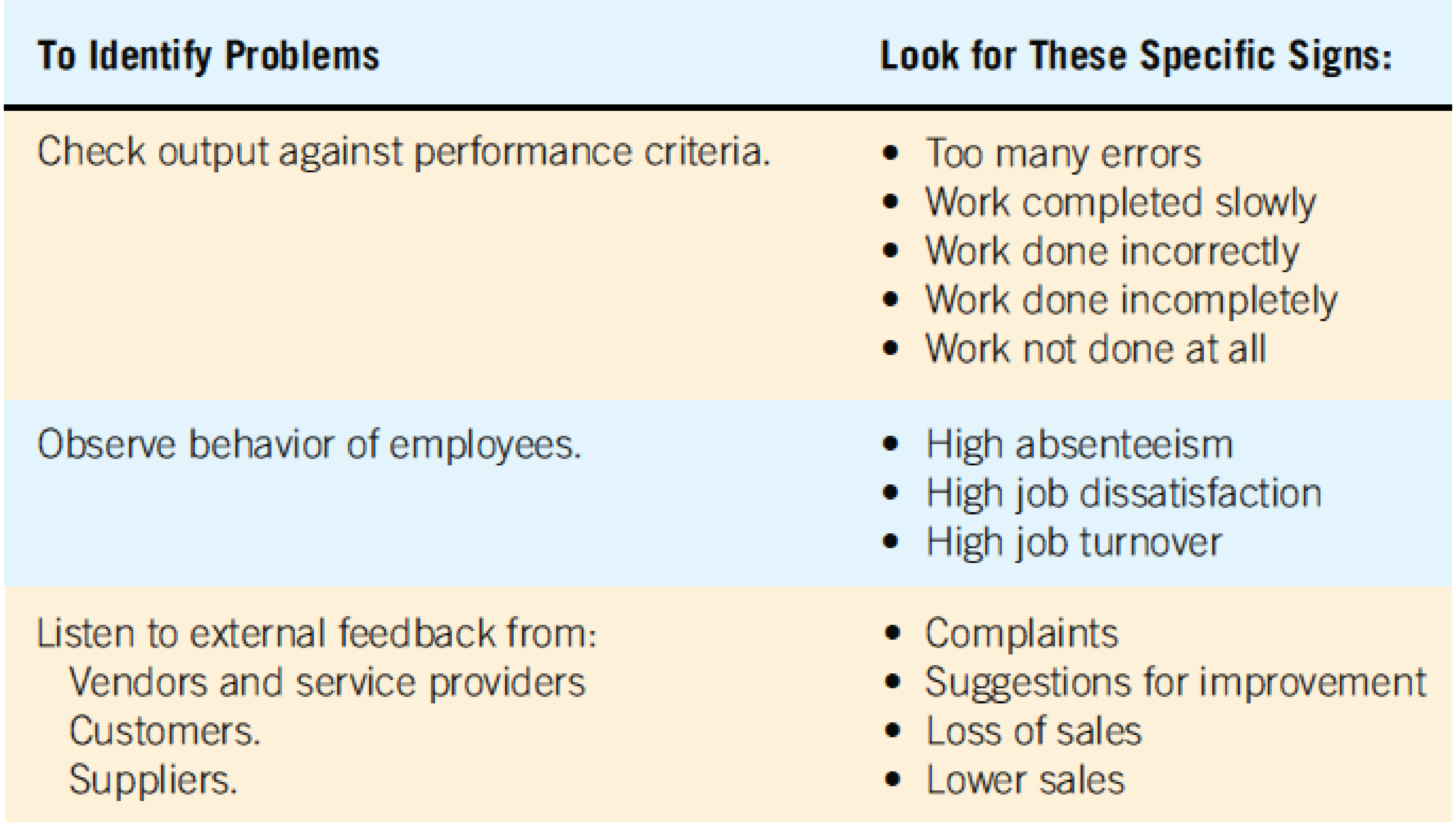
- Problems arise in different ways including:
  - Problems are uncovered because performance measures are not being met
  - Problems (or symptoms of problems) with processes relate to output issues
    - Such problems may require a systems analysis to evaluate excessive errors and work performed too slowly, incompletely, incorrectly, or not at all
  - Other  symptoms of problems become evident when people do not meet baseline performance goals
- Changes in:
  - Employee behaviour (such as unusually high absenteeism)
  - High job dissatisfaction
  - High staff turnover
  - Should alert managers to potential problems

# Project Initiation

- Any of these changes (alone or in combination):
  - May be a sufficient reason to request the help of a systems analyst
- While problems:
  - Such as those just described occur in an organization
  - Feedback on  how well the organization is meeting intended goals may come from outside in the form of:
    - Complaints or suggestions from customers, vendors, or suppliers, as well as lost or unexpectedly low sales.
  - This feedback from the external environment is extremely important and should not be ignored
- A summary of symptoms of problems and approaches useful in problem detection is  provided in Figure 3.1.

# Problem Identification (Figure 3.1)

- Checking Output, Observing Employee Behavior, and Listening to Feedback Are All Ways to Help the Analyst Pinpoint Systems Problems and Opportunities (Figure 3.1)



| To Identify Problems | Look for These Specific Signs: |
|---|---|
| Check output against performance criteria. | • Too many errors<br>• Work completed slowly<br>• Work done incorrectly<br>• Work done incompletely<br>• Work not done at all |
| Observe behavior of employees. | • High absenteeism<br>• High job dissatisfaction<br>• High job turnover |
| Listen to external feedback from:<br>Vendors and service providers<br>Customers.<br>Suppliers. | • Complaints<br>• Suggestions for improvement<br>• Loss of sales<br>• Lower sales |

# Defining the Problem

# Problems in an Organisation

- Whether using the classic systems development life cycle (SDLC) approach or an object-oriented approach:
  - Analysts first define the problem(s) and objectives of the system
  - These set the basis for determining what needs to be accomplished by the system
- A problem definition usually contains:
  - A problem statement (summarized in a paragraph or two)
  - This is followed by a series of issues or major independent pieces of the problem
  - The issues are followed by a series of objectives or goals that match the issues point by point
  - Issues are:
    - The current situation (the objectives are the desired situation)
    - The objectives may be very specific or worded as a general statement
  - Use the problem definition to create a preliminary test plan

# Business Objectives

- Some examples of business questions relating to business objectives are:
    - What are the purposes of the organization?
    - Is the organization for-profit or non-profit?
    - Does the company plan to grow or expand?
    - What is the organization’s attitude (organisational culture) about technology?
    - What is the organization’s budget for IT?
    - Does the organization’s staff have the required knowledge and expertise?
- A systems analyst needs to understand how a business works.

# Problem Definition

- Finally: the problem definition contains requirements
  - Things that must be  accomplished (along with the possible solutions and the constraints that limit the development  of the system
  - The requirements section may include:
    - Security / usability / government requirements / etc
  - Constraints often include the word NOT (indicating a limitation, and may contain budget restrictions or time limitations)
- The problem definition is produced following:
  - Completing interviews
  - Observations
  - Analysis documentation with users
- The result of information gathering is:
  - A wealth of facts and  important opinions in need of summary
- The first step in producing the problem definition is  to find a number of points that may be included in one issue

# Interviews

- There are essentially three types of interview:
  - Structured interviews
  - Semi-structured interviews
  - Unstructured interviews
- Major points can be identified in an interview in a number of ways:
  1. Users may identify an issue, a topic, or a theme that is repeated several times, sometimes by different people in several interviews
  2. Users may communicate using the same metaphors, such as saying the business is a journey,  a war, a game, an organism, a machine, a family, and so on
  3. Users may tell a story to illustrate a problem that includes a beginning, a middle, and an ending; a hero; obstacles to overcome; and a successful (or hoped for) resolution
  4. Users may speak at length on a topic
  5. Users may tell you outright, “This is a major problem”
  6. Users may communicate importance using body language or may speak emphatically on  an issue
  7. The problem may be the first thing the user mentions

# Problems in an Organisation

- Once the issues have been identified:
  - The objectives must be stated
  - An analyst may conduct follow-up interviews to obtain more precise information about the objectives
- Once the objectives are identified and stated:
  - The relative importance of the issues or objectives must  be determined
  - If there are insufficient funds to develop the complete system – the critical  objectives must be completed first
- Users are the best people to identify critical objectives  (with the support of analysts) because users are domain experts in their business area and  know how they work best with technologies in the organization
- Additionally:
  - In addition to looking  through data and interviewing people
  - A systems analyst should try to witness the problem ‘first-hand’

# Catherine’s Catering

# A Problem Definition Example

- Catherine’s Catering (pp. 89-91)
- Catherine’s Catering is a small business that:
  - Provides (caters) meals
  - Organizes receptions
  - Organizes banquets
- The client base includes:
  - Organizations (business)
  - Private clients
- The range of functions address:
  - Social occasions (such as luncheon)
  - Meetings
  - Weddings
- Refer to the Catherine’s Catering example and problem definition in the course textbook

# Selection of Projects

# Problem Proposals

- Projects are drawn from many different sources and for many reasons
- Following a preliminary analysis:
  - Not all should be selected for further analysis and development:
- There must be a compelling and clear reason(s) for recommending a systems study on a project that seems to address a problem or could bring about some clearly identified improvement
- Consider  the motivation that prompts a proposal on the project:
  - You need to be sure that the project under consideration is not being proposed simply to enhance some reputation or power-base of the proposing person or group
  - Such ill-conceived proposals will have a high probability of failure
  - Such projects will may be poorly accepted by users

# Problems from a Systems Perspective

- As outlined in Chapter #2:
  - Prospective  projects need to be examined from a systems perspective
  - The impact on the whole organization must be considered
- Recall that the various sub-systems of an organization are interrelated and interdependent:
  - A change to one subsystem may affect all the others
- While decision makers directly involved ultimately set the boundaries for a systems project
  - A systems project cannot be contemplated or selected in isolation  from the rest of the organization

# Problems in an Organisation

- Beyond these general considerations are five specific questions that need to be asked regarding  project selection:
  1. Does it have backing from management?
  2. Is the timing of the project commitment appropriate?
  3. Is it possible to improve attainment of strategic organization goals?
  4. Is it practical in terms of resources for the systems analyst and organization?
  5. Is it a worthwhile project compared with other ways the organization could invest resources?

# Problems in an Organisation

- First and foremost is backing from management
- Nothing will be accomplished without the endorsement of the people who fund the project
  - This statement is not meant to imply that you lack influence in directing the project or that people other than management can’t  be included
  - However: management backing is essential
- Another important criterion for project selection is timing for you and the organization
  - Ask  yourself and the others who are involved whether the business is presently capable of making the required commitment (time and resources) for the design, development, and installation of new systems or improvement to existing ones
  - You must also  be able to commit all or a portion of your time for the duration.

# Problems in an Organisation

- A third criterion is the possibility of improving attainment of strategic organization goals such as:
    1. Improving corporate profits
    2. Supporting the competitive strategy of the  organization
    3. Improving cooperation with vendors and partners
    4. Improving internal operations support so that goods and services are produced efficiently and effectively
    5. Improving internal decision support so that decisions are more effective
    6. Improving customer service
    7. Improving employee morale. The project should put the organization  on target, not deter it from its ultimate goals
- A fourth criterion is selecting a project that is practicable in terms of your resources and  capabilities as well as those of the business
    - Some projects will not fall within your realm of  expertise, and you must be able to recognize them.

# Project Motivation

- Finally there is a requirement for a basic agreement with the organization about the value of a systems project relative to any other project being considered
- There are  many possibilities for improvements including:
  1. Speeding up a process
  2. Streamlining a  process through the elimination of unnecessary or duplicated steps
  3. Combining processes
  4. Reducing errors in input through changes of forms and display screens
  5. Reducing  redundant storage
  6. Reducing redundant output
  7. Improving integration of systems  and subsystems
- Remember that when a business commits to one project, it is committing  resources that thereby become unavailable for other projects
  - It is useful to view all possible  projects as competing for the business resources of time, money, and pe

# Determine Feasibility

# Project Feasability

- Once the number of projects has been narrowed according to the criteria discussed:
    - It is still necessary to determine whether the selected projects are feasible
    - Our definition of feasibility goes much deeper than the common usage of the term
    - Systems project feasibility is assessed in three principal ways:
        - Operationally
        - Technically
        - Economically
- A feasibility study is not a comprehensive systems analysis
    - A feasibility study is used to gather broad data for the members of management that enables them to make a decision about whether to proceed with a systems study
- While it is time consuming and potentially painstaking:
    - Studying feasibility is worthwhile because it saves businesses and systems analysts time and money
- For an analyst to recommend further development:
    - A project must show that it is feasible in all three principal ways (see Figure 3.3).

# Determining Whether It Is Possible

# Technical Feasibility

- An analyst must determine:
  - If it is possible to develop a new system  given the current technical resources
  - If not, can the system be upgraded or added to in a  manner that fulfils the request under consideration?
  - If existing systems cannot be added to  or upgraded is the required technology in existence that meets the specifications
- An analyst must also determine:
  - If the organization has staff who are technically proficient to accomplish the objectives – if the answer is NO:
    - Can the required additional experienced staff (programmers / testers / and systems experts) be recruited?
    - Perhaps outsourcing the project completely is an option?
- Additionally:
  - Are the required software packages available to realise the objectives?
  - To what degree will the software need customization for the organization?

# Economic Feasibility

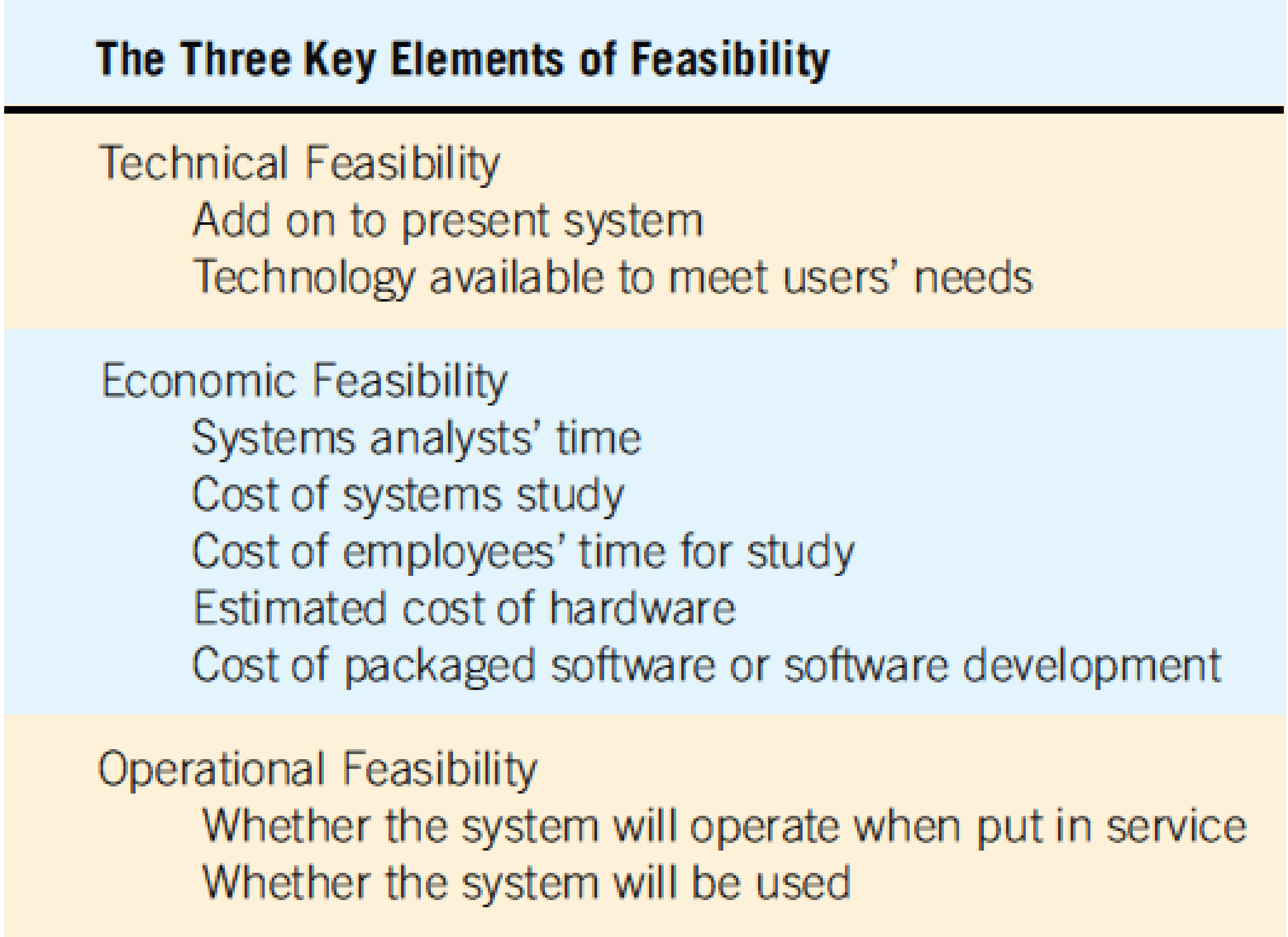
- This is the second part of resource determination:
  - The  basic resources to consider are:
    - Your time and that of the systems analysis team
    - The cost of  doing a full systems study (including the time of employees you will be working with)
    - The  cost of the business employee time
    - The estimated cost of hardware
    - The estimated cost of  software, software development, or software customization
- The organization must be able to see the value of the investment before committing to an entire systems study:
  - If short-term costs are not overshadowed by long-term gains (or) if the project produces no immediate reduction in operating costs, the system  is not economically feasible and should not proceed further

# Operational Feasibility

- If the technical and economic resources are both judged adequate
  - The systems analyst must now consider the operational feasibility of the requested project
  - Operational feasibility is dependent on the human resources available for the project and involves projecting whether the system will operate and be used once it is put into service
- If users are:
  - Happy with the current system(s)
  - Have little or no problems with it
  - Are not involved in requesting a new system
  - Resistance to implementing the new system will be strong and the new system has a low chance of becoming operational.
- Alternatively:
  - If users have expressed a need for a system that is:
    - Operational for more of the time
    - Is more efficient and more accessible
  - There is a high potential that the new (requested) system will be accepted and used , chances are better that the requested system will eventually be used.
- Much of the art of determining operational feasibility rests with the user interfaces that are chosen (see in Chapter 14)

# Feasibility Considerations (Figure 3.3)

- The Three Key Elements of Feasibility Include:
  - Technical feasibility
  - Economic feasibility
  - Operational Feasibility



**The Three Key Elements of Feasibility**

Technical Feasibility

- Add on to present system
- Technology available to meet users' needs

Economic Feasibility

- Systems analysts' time
- Cost of systems study
- Cost of employees' time for study
- Estimated cost of hardware
- Cost of packaged software or software development

Operational Feasibility

- Whether the system will operate when put in service
- Whether the system will be used

# Estimating Workloads

# Estimating Workloads

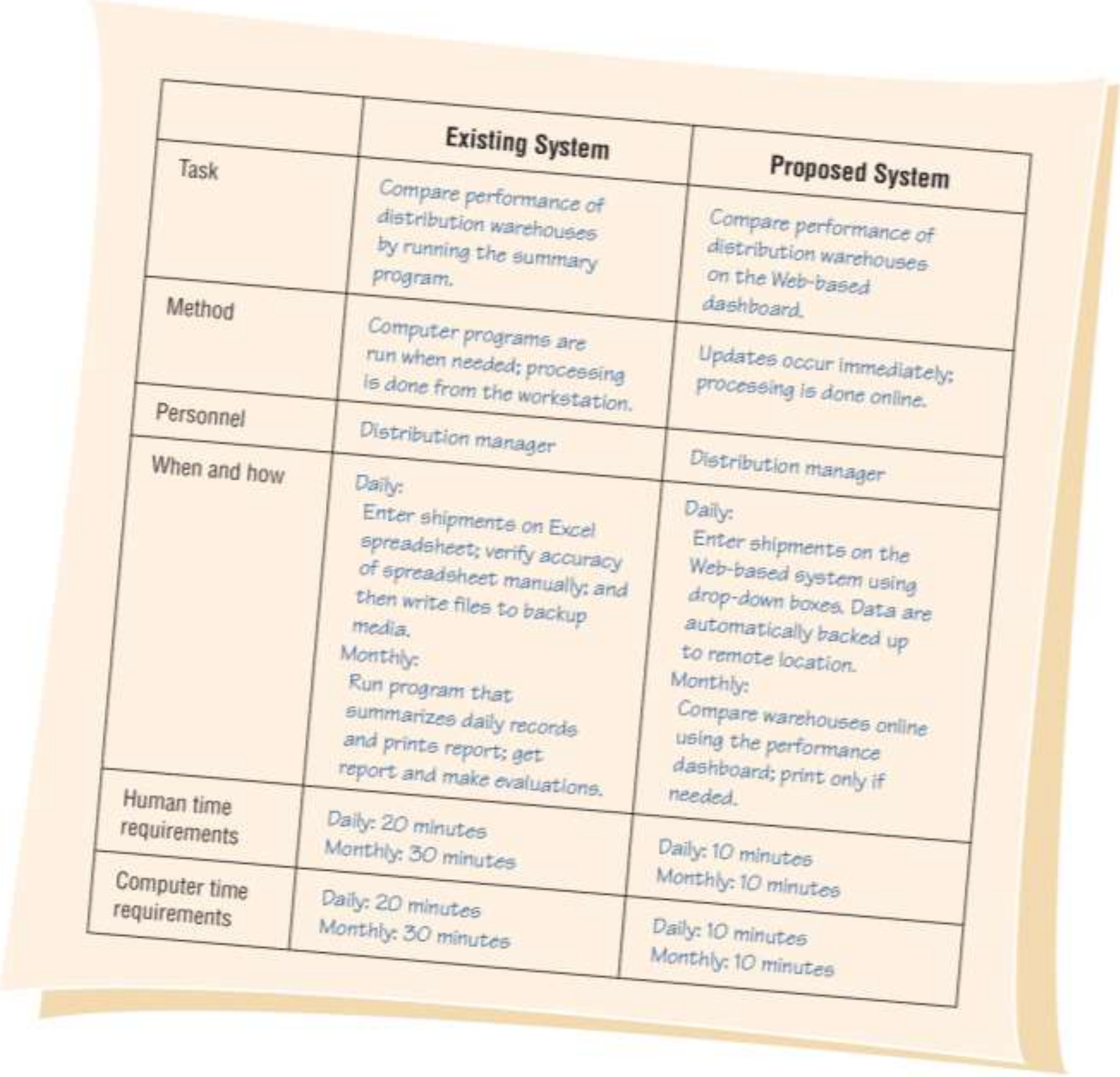
- The next step in ascertaining hardware needs is to estimate workloads where:
  - Systems analysts formulate numbers that represent both current and projected workloads for the system so that any  hardware obtained will be capable of handling current and future workloads
  - If estimates are accurate:
    - The business should not have to replace hardware solely due to unforeseen growth in system use
    - Other events (such as superior technological innovations) may dictate hardware replacement if the business wants to maintain its competitive edge
- Out of necessity:
  - Workloads are sampled rather than actually run through several computer systems
  - The guidelines (see Chapter 5) can be of use here because in workload sampling the  systems analyst is taking a sample of necessary tasks and the computer resources required to complete them

# Estimating Workloads

- Figure 3.4 compares:
  - The time required by existing and proposed information systems to handle a given workload
- Where a company:
  - Is currently using a legacy computer system to prepare a summary of shipments to its distribution warehouses (and)
  - A new Web-based dashboard is being suggested
  - The workload comparison looks at:
    - When and how each process is carried out?
    - How much human time is required?
    - How much computer time  is needed?

# Estimating Workloads

- Comparisons of Workloads between Existing and Proposed Systems (Figure 3.4 )
- Note:
  - The new proposed system should reduce the required human and  computer time significantly



| | Existing System | Proposed System |
|---|---|---|
| Task | Compare performance of distribution warehouses by running the summary program. | Compare performance of distribution warehouses on the Web-based dashboard. |
| Method | Computer programs are run when needed; processing is done from the workstation. | Updates occur immediately; processing is done online. |
| Personnel | Distribution manager | Distribution manager |
| When and how | Daily:<br>Enter shipments on Excel spreadsheet; verify accuracy of spreadsheet manually; and then write files to backup media.<br>Monthly:<br>Run program that summarizes daily records and prints report; get report and make evaluations. | Daily:<br>Enter shipments on the Web-based system using drop-down boxes. Data are automatically backed up to remote location.<br>Monthly:<br>Compare warehouses online using the performance dashboard; print only if needed. |
| Human time requirements | Daily: 20 minutes<br>Monthly: 30 minutes | Daily: 10 minutes<br>Monthly: 10 minutes |
| Computer time requirements | Daily: 20 minutes<br>Monthly: 30 minutes | Daily: 10 minutes<br>Monthly: 10 minutes |

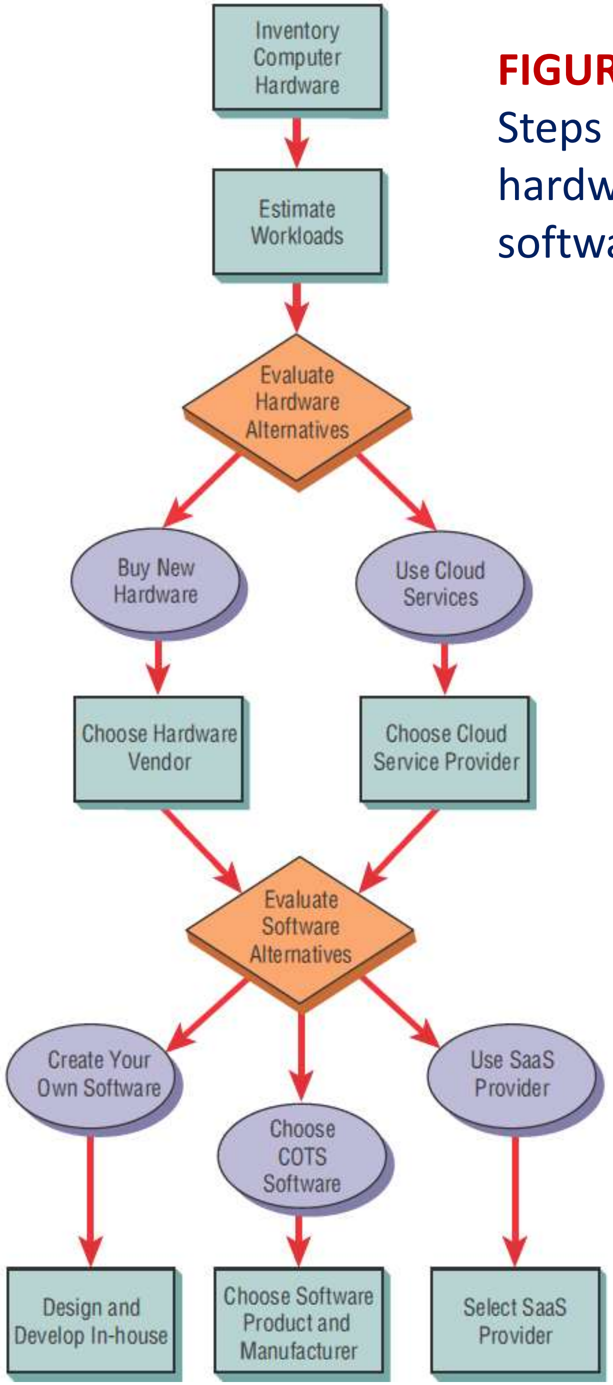
# Ascertaining Hardware and Software Needs

# Identifying Hardware and Software Needs

- Assessing technical feasibility includes evaluating the ability of computer hardware and software to handle workloads adequately
- First:
  - All current computer hardware the organization owns must be inventoried to discover what is on hand and what is usable
  - A systems analyst needs to work with users to determine what hardware will be needed
  - Hardware determinations can be made only in conjunction with determining human information requirements
- Knowledge of the organization structure and how users interact with  technologies in an organization setting also can be helpful in hardware decisions
  - Only  when systems analysts, users, and management have a good grasp of what kinds of tasks  must be accomplished can hardware options be considered

# Identifying Hardware and Software Needs

- Figure 3.5 shows
  - The steps a systems analyst takes in identifying hardware and software needs



Inventory Computer Hardware
Estimate Workloads
Evaluate Hardware Alternatives
Buy New Hardware
Use Cloud Services
Choose Hardware Vendor
Choose Cloud Service Provider
Evaluate Software Alternatives
Create Your Own Software
Choose COTS Software
Use SaaS Provider
Design and Develop In-house
Choose Software Product and Manufacturer
Select SaaS Provider

**FIGURE 3.5**
Steps in choosing hardware and software

# Prepare an Inventory for the Computer Hardware

# Prepare an Inventory

- A systems analyst needs to inventory what computer hardware is already available in the organization
- Hardware options involve expanding or recycling  current hardware
- It is important to know what is available
- If an updated computer hardware inventory is unavailable:
  - The systems analyst needs to set  one up quickly and carry through on it. It is important to know the following:
    - The type of equipment (including model number(s) and manufacturer(s)
    - The operational status of the equipment including:
      - Equipment on order / operating / in storage / in need  of repair

# Prepare an Inventory

  - The estimated age and operational status of the equipment
  - The projected life of the equipment
  - The physical location of the equipment
  - The department or person considered responsible for the equipment
  - The financial arrangement for the equipment including:
    - Owned / leased / rented
- Ascertaining the current hardware available will result in a sounder decision-making process when hardware decisions are finally made as much of the guesswork about what  exists will have been eliminated
- Through your earlier interviews with users, questionnaires, surveys, and research of archival data:
  - It will already be known how many of people available for data processing as well as their skills and capabilities
  - Use this information to project how well the staffing needs for new hardware can be met

# Acquisition of Computer Equipment

- When organisations consider the provision of computerised services they have a number of available options:
  - Owning (purchasing or leasing) and operating computer hardware and software in (for example) a datacentre
  - Owning (purchasing or leasing) limited computing facilities with computing services provided using cloud-based systems
- In the following slides we consider these options
- We must remember with cloud-based services:
  - While there may be significant cost savings (and)
  - There are significant scalability benefits
  - There are also significant legal and contractual implications

# Evaluating Computer Hardware for Purchase

# Hardware Evaluation

- Evaluating new computer hardware is the shared responsibility of:
  - Management
  - Users
  - Systems analysts
- While equipment vendors will be supplying equipment details:
  - Analysts need to oversee the evaluation process as they work in the interest of the organisation
  - Moreover:
    - Systems analysts may have to educate users and management about the general advantages and disadvantages of hardware before they can capably evaluate it

# Identify Hardware and Software Needs

- Estimating hardware and software requirements requires a reasonable forecasts of the potential workload over the systems life cycle
- Based on the current inventory of computer equipment (and adequate estimates of current and forecast workloads:
    - The next step in the process is to consider the kinds of equipment available that appear to meet projected needs
    - Information from vendors on possible systems and system configurations becomes more important at this stage and should be reviewed with management and users
- Additionally:
    - Workloads can be simulated and run on different systems
    - Including systems  already in use by the organization
    - This process is referred to as *benchmarking*

# Apply Criteria

- Criteria that the systems analysts and users
  - Should use to evaluate performance of different systems hardware include the following:

1. The time required for average transactions
   1. Including how long it takes to input data and how long it takes to receive output
2. The total volume capacity of the system (how much can be processed at the same time  before a problem arises)
3. The idle time of the CPU or network
4. The size of the memory provided

# Apply Criteria

- Some criteria will be shown in formal demonstrations
- Some criteria cannot be simulated and must  obtained from manufacturers' or service providers' specifications
- It is important to be clear about the required and desired functions before inviting supplier demonstrations
- Once functional requirements are known and the products currently available are:
    - Identified and the capabilities and limitations known
    - Supplier equipment compared (including comparisons with current equipment used in the organization)
    - The systems analysts (in conjunction with users and management) can make decisions on the need for new  hardware

# The Disadvantages of Buying Computer Hardware

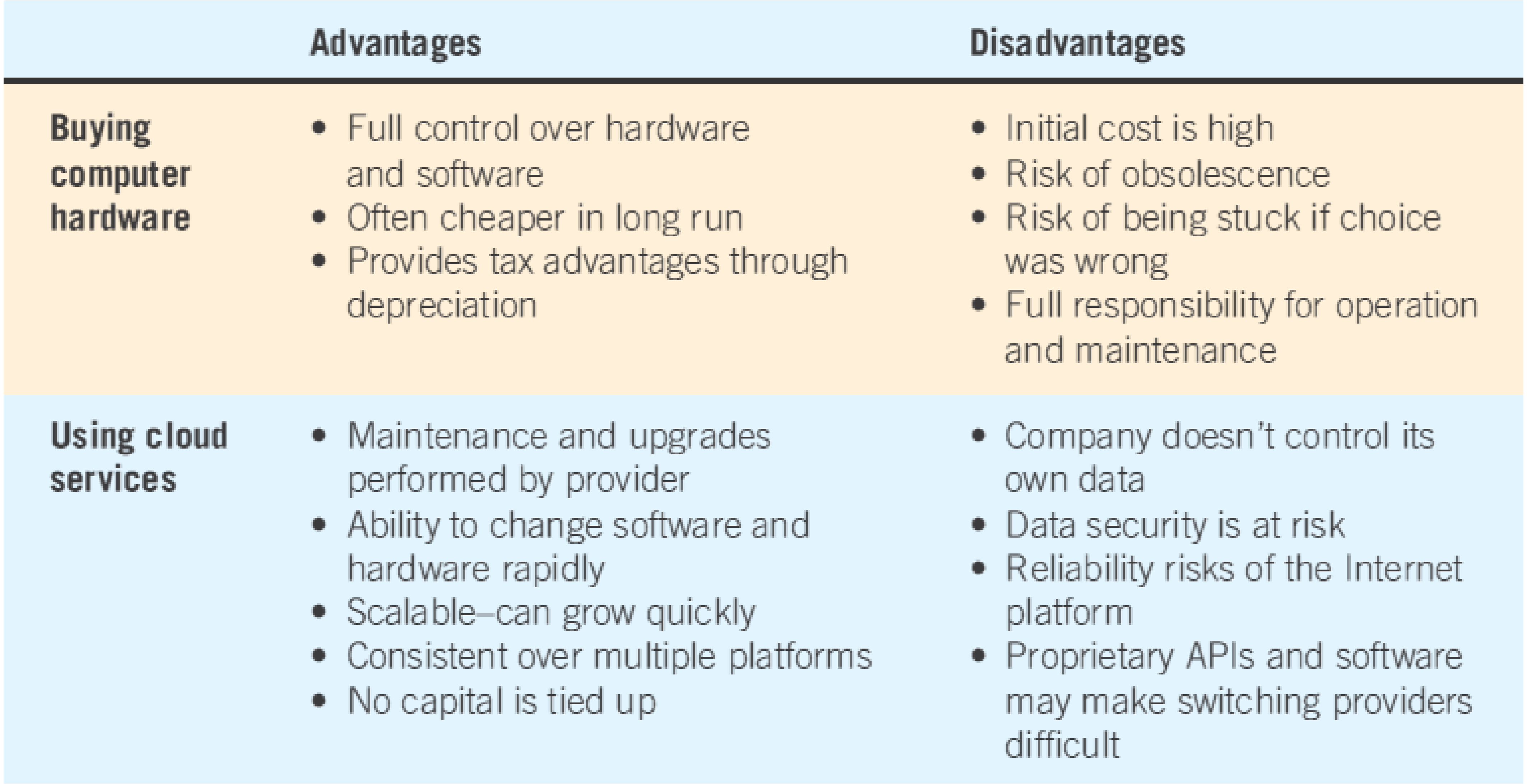
- Purchasing computers may present a problem because:
  - The initial cost of buying the equipment is often very high
  - Phasing in computer equipment  over time doesn't always work due to compatibility problems that exist with different models and legacy devices
  - Buying everything at once means tying up capital or borrowing money
  - The risk of equipment obsolescence is a serious concern
  - Obtaining equipment  that is not useful because someone made the wrong decision is a very large risk
- Finally, the  organization needs to remember that:
  - The full responsibility for the operation and maintenance stays with the company that buys the equipment
  - This can be a big disadvantage because it ties up people and money

# Renting Time and Space in the Cloud

# Renting Time and Space in the Cloud

- As an alternative to purchasing their own equipment companies can use cloud services which include:
    - Web hosting
    - E-mail hosting
    - Application hosting
    - Backup, archiving, and disaster recovery services
    - Data storage
    - Data processing
    - The provision of databases
    - E-commerce support
- The advantages and disadvantages to buying hardware versus using cloud services are shown in Figure 3.6

# Cloud Services (Figure 3.6)



| | Advantages | Disadvantages |
|---|---|---|
| **Buying computer hardware** | • Full control over hardware and software<br>• Often cheaper in long run<br>• Provides tax advantages through depreciation | • Initial cost is high<br>• Risk of obsolescence<br>• Risk of being stuck if choice was wrong<br>• Full responsibility for operation and maintenance |
| **Using cloud services** | • Maintenance and upgrades performed by provider<br>• Ability to change software and hardware rapidly<br>• Scalable–can grow quickly<br>• Consistent over multiple platforms<br>• No capital is tied up | • Company doesn't control its own data<br>• Data security is at risk<br>• Reliability risks of the Internet platform<br>• Proprietary APIs and software may make switching providers difficult |

# Cloud-Based Services

- Cloud services are scalable which means if the business grows the organization can add computing power without procuring new hardware
- Although few organizations save money by purchasing cloud services they recognize the agility afforded by being able to add or stop cloud services as needed
- Three main categories of cloud computing are
  - Software as a Service (SaaS),
  - Infrastructure as a Service (IaaS), and
  - Platform as a Service (PaaS).

# Cloud-Based Services

- With SaaS:
  - The cloud vendor sells access to applications (discussed later in this chapter when we discuss software)
- With IaaS (also called Hardware as a Service (HaaS)):
  - An organization outsources its hardware operations needs to the service provider
- Using the PaaS approach:
  - Companies use the Internet to rent a range or:
    - Hardware / operating systems / storage / and even network capacity
- Another option is to use a hybrid approach where:
  - Some of the hardware is maintained in the local organization
  - Some of the software is cloud-based
  - This option includes public and private cloud services
- Decisions on cloud computing can first be addressed on a strategic level

# Decisions for Cloud-Computing Systems

- Organisations should:
  1. Define a high-level business case that focuses on the high-level benefits of cloud computing for the organization
  2. Define the legal and contractual requirement of the organisation
  3. Define core requirements (perhaps initially making a quick list) to include:
     1. System performance
     2. Security desired in cloud relationships
     3. IT governance
     4. Predicted growth of the enterprise
  4. Define core technologies for the enterprise
     1. Starting with a list of cloud computing technologies that are useful.

As an analyst:

You may be asked to facilitate any or all of these steps

Or your help might be requested at some point in the process of moving computing services to the cloud

# The Benefits of Cloud Computing

- Benefits of cloud computing include
  - Reduced (or no) time spent maintaining legacy systems or even performing routine tasks such as maintenance or upgrading of present systems
  - Simpler acquisition of IT services
    - It can also be easier and quicker to separate from or discontinue services that are no longer necessary
  - Ensuring applications are scalable
    - Which means you can grow them easily  by adding more cloud resources
  - Cloud computing also has the potential to offer consistency across multiple platforms that were previously disjointed or difficult to integrate
  - Finally: no  capital is tied up and no financing is required

# The Drawbacks of Cloud Computing

- As with any other innovation with cloud computing there are disadvantages including:
  - Reduced (or complete loss) of control of data stored in the cloud (an important concern)
  - If the cloud services provider ceased to exist it may be unclear what would happen to the organization’s data (and in what format would the data be returned)
  - There are potential security threats to data that is not stored on premises or even on the organization’s own computers
  - Reliability of the Internet as a platform forms the third main concern for organizations
- To offset these potential benefits organisations must:
  - Have bespoke legal contracts drawn up between organisations and cloud service providers

# Potential Cloud Services problems

- Further potential issues are:
  - An organisation may have it’s manoeuvrability restricted
  - The cloud provider may restrict the available application programming interfaces (APIs) to proprietary APIs
    - This can make switching among cloud computing providers more difficult
  - Other developments in cloud computing of which analysts should be aware are:
    - The data protection legislation (e.g., the GDPR in Eurpop) which restrict the locations where data can be legally processed and stired
    - Some countries are now mandating that any company that does business there must use cloud service providers based in that country
    - For example: Apple is now obeying legislation like this in China

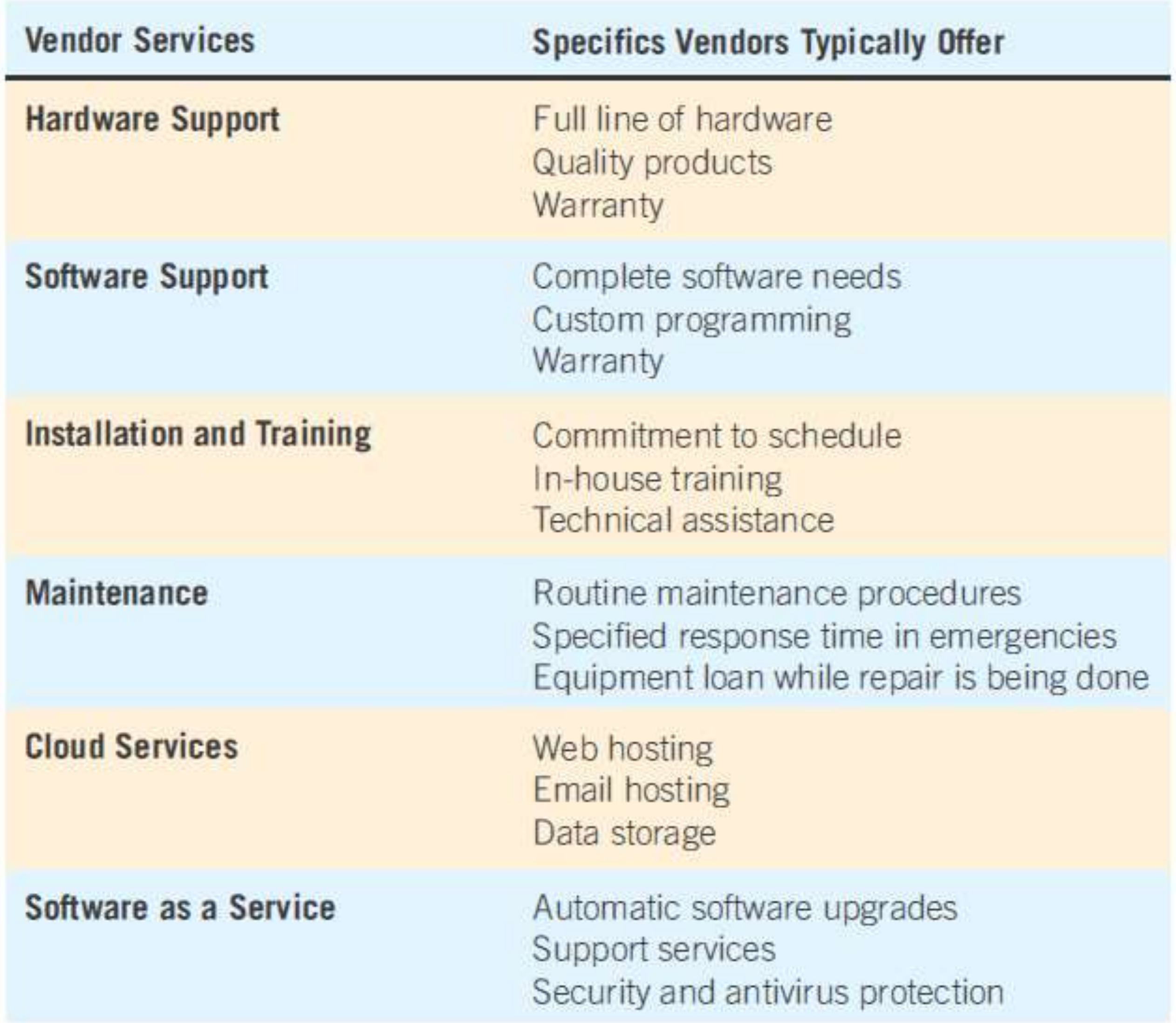
# Evaluation of Vendor Support for Computer Hardware

# Vendor Evaluation

- Several key areas need to be evaluated when considering the support services vendors offer to businesses and organisations
- Most hardware vendors offer testing of hardware on delivery and a 90-day warranty covering any factory defects
  - However: you must ascertain what else the vendor has to offer
- Cloud service providers offer 30- or 60-day money-back guarantees.
- Vendors of comparable quality frequently distinguish themselves from others in the range of support services they offer
- A list of key criteria that ought to be checked when evaluating vendor support is provided in Figure 3.7.

# Cloud Computing Vendor Key Criteria (Figure 3.7)

- Support services include:
  - Routine and preventive maintenance of hardware
  - Specified response time (within six hours, next working day, and so on) in case of emergency equipment breakdowns
  - Loan of equipment in the event that hardware must be permanently replaced or off-site repair is required
  - In-house training or off-site group seminars for users



| Vendor Services | Specifics Vendors Typically Offer |
|---|---|
| **Hardware Support** | Full line of hardware<br>Quality products<br>Warranty |
| **Software Support** | Complete software needs<br>Custom programming<br>Warranty |
| **Installation and Training** | Commitment to schedule<br>In-house training<br>Technical assistance |
| **Maintenance** | Routine maintenance procedures<br>Specified response time in emergencies<br>Equipment loan while repair is being done |
| **Cloud Services** | Web hosting<br>Email hosting<br>Data storage |
| **Software as a Service** | Automatic software upgrades<br>Support services<br>Security and antivirus protection |

# Identify Hardware and Software Needs

- Cloud providers make claims for uptime and promises to get back to you on a question within a certain amount of time (e.g., 24 hours)
- Carefully read the support services documents accompanying the purchase of equipment and remember to involve appropriate legal staff before signing contracts for equipment or services
  - ***Organisations will generally conclude a bespoke cloud services contract agreement***
- Unfortunately, the process of evaluating computer hardware and cloud providers is not as straightforward as simply comparing costs and choosing the least expensive option
- Before becoming convinced that buying cheaper compatibles is the way to endow your system with add-on capability
  - You need to research the service provision to feel confident that a hardware vendor or cloud provider is a stable corporate entity

# Understanding the Bring Your Own Device (BYOD) Option

# Bring your own device (BYOD) or Technology (BYOT)

- Bring your own device (BYOD) or bring your own technology (BYOT) is a trend appearing in a  variety of organizations in different industries
- These organizations range in size and have employees of differing skills (albeit with a minimum level of computer literacy)
- In general BYOD or BYOT means that rather than using a smartphone or tablet issued by the organization
  - The employee uses his or her own device to access corporate networks,  data, and services remotely when working outside the office.
- The BYOD approach is often presented as a way to:
  - Minimise the organization’s hardware and infrastructure costs down
  - Improve morale among employees who (generally) bring one or more mobile devices to work for personal tasks

# Bring your own device (BYOD) or Technology (BYOT)

- As a systems analyst:
  - You will work to discover (through a combination of interviews, surveys, and observation) what personal technology is being used by the majority of the group
- For  example:
  - If you observe that all the executives in the decision-making group already bring in  iPads every day
  - You might conclude that designing dashboards for the iPad has some merit
  - They are devices you want to include in your original design rather than as an afterthought
  - If  your observations show that most executives prefer face-to-face meetings
  - However: they use their  handheld devices (such as mobile phones) to check other corporate data and emails during  meetings
  - Then you might want to consider supporting corporate email on their personal phones

# Benefits and Drawbacks of BYOD

- The ***benefits*** of BYOD computing include:
  - Potentially building employee morale
  - Potential for lowering the initial cost of organization IT hardware purchase
  - Facilitating remote *anytime and anywhere* 24/7/365 access to corporate computer networks on a familiar user interface to access:
    - Corporate computing services
    - Applications
    - Databases and reports
    - Data storage
- Security risks posed by untrained users are probably the biggest ***drawback*** include:
  - Identified threats are loss of the device itself
  - Theft of the device and its data
  - Unauthorized access to corporate networks using personal mobile devices
  - Other threats are posed by commonly occurring behaviour with mobile devices (that may be fine on the private side but not on the corporate side) include:
    - Using free insecure Wi-Fi hotspots
    - Using apps such as Dropbox

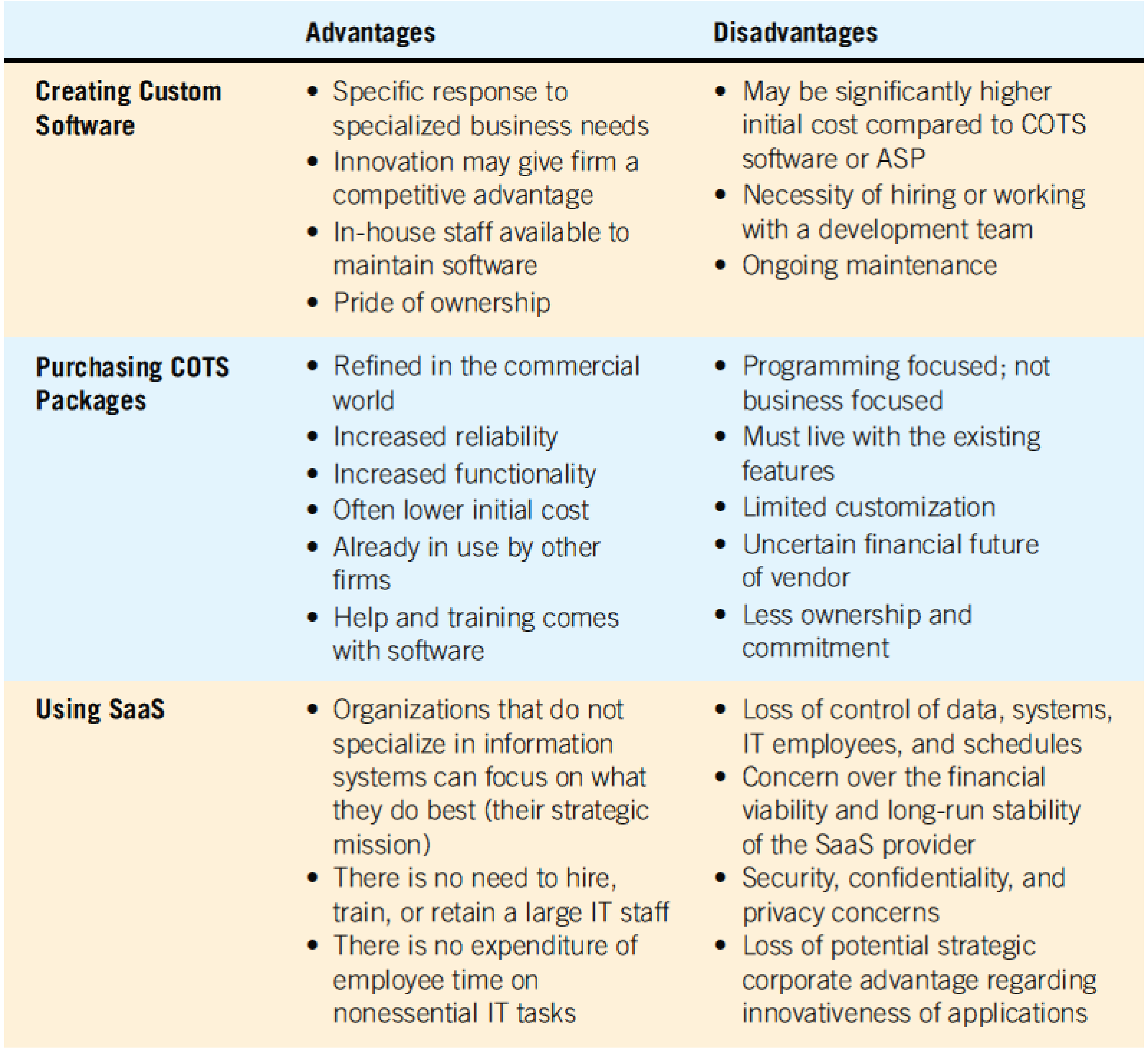
# Creating Custom Software

# Custom Software

- When considering the provision of information systems projects software:
  - Analysts and organizations are increasingly faced with a decision to:
    - Created custom (bespoke) software
    - Purchased as COTS (commercial-of-the-Shelf) software
    - Outsource software (Provided as a software as a service (SaaS))
  - This decision is particularly important when contemplating upgrades to existing or legacy systems
- Analysts have three choices:
  - Creating their own software
  - Purchasing commercial off-the-shelf (COTS) software
  - Using software from a SaaS provider
- Figure 3.8 summarizes the advantages and disadvantages of each of these options

# Custom Software Alternatives (Figure 3.8)

- FIGURE 3.8
- Comparing the advantages and disadvantages of:
  - Creating custom software
  - Purchasing COTS packages
  - Outsourcing to a SaaS provider



| | Advantages | Disadvantages |
|---|---|---|
| **Creating Custom Software** | • Specific response to specialized business needs<br>• Innovation may give firm a competitive advantage<br>• In-house staff available to maintain software<br>• Pride of ownership | • May be significantly higher initial cost compared to COTS software or ASP<br>• Necessity of hiring or working with a development team<br>• Ongoing maintenance |
| **Purchasing COTS Packages** | • Refined in the commercial world<br>• Increased reliability<br>• Increased functionality<br>• Often lower initial cost<br>• Already in use by other firms<br>• Help and training comes with software | • Programming focused; not business focused<br>• Must live with the existing features<br>• Limited customization<br>• Uncertain financial future of vendor<br>• Less ownership and commitment |
| **Using SaaS** | • Organizations that do not specialize in information systems can focus on what they do best (their strategic mission)<br>• There is no need to hire, train, or retain a large IT staff<br>• There is no expenditure of employee time on nonessential IT tasks | • Loss of control of data, systems, IT employees, and schedules<br>• Concern over the financial viability and long-run stability of the SaaS provider<br>• Security, confidentiality, and privacy concerns<br>• Loss of potential strategic corporate advantage regarding innovativeness of applications |

# Custom and COTS Software

- Several situations call for the creation of original bespoke custom software or software components:
  - The most likely cause is when COTS software does not exist or cannot be identified for the desired application
  - Alternatively: the software may exist but be unaffordable or not easily purchased or licensed
- Original software should be created when:
  - An organization is attempting to gain a competitive advantage through the leveraged use of information systems
  - This is often the case when an organization is creating e-commerce or other innovative novel applications where none have existed

# Custom and COTS Software

- An organization also may be a “first mover” in the use of a particular technology or in its particular industry
- Organizations that have highly specialized requirements or exist in niche industries also may benefit from creating original software
- The advantages of creating custom software include:
  - Being able to respond to specialized user and business needs
  - Gaining a competitive advantage by creating innovative software
  - Having inhouse staff available to maintain the software
  - The corporate pride in developing and owning something bespoke created by the organization

# Custom and COTS Software

- There are also drawbacks of developing custom software which include:
  - The potential for a significantly  higher initial cost compared to purchasing COTS software or contracting with a SaaS provider
  - The necessity of hiring or working with a development team
  - The responsibility for the ongoing maintenance because you created the software
  - The need for in-house user training and support services (which would be the responsibility of the outsourced software provider
- We may also note that there may be a hybrid system using a combination of COTS and bespoke software

# Purchasing COTS Software

# COTS Software

- COTS software includes such products as
  - The Microsoft Office suite
  - Word processing
  - Image processing
  - AutoCAD software
- Other types of COTS software are:
  - For organization-level systems rather than office or personal use
  - A website such as softwareadvice.com lists popular (but costly) enterprise resource planning (ERP) packages such as Oracle and SAP in their reviews of COTS software
    - These packages differ radically in the amount of customization, support, and maintenance required compared to Microsoft Office
  - COTS software also can refer to software components or objects (also called building blocks) that can be purchased to provide a particular needed functionality in a system

# COTS Integration

- Consider using COTS software when you can:
  - Easily integrate the applications or  packages into existing or planned systems
  - When you have identified no need to immediately (or continuously) change or customize systems for users
- Forecasts should demonstrate that:
  - The organization you are designing the system for is unlikely to undergo major changes after the proposed purchase of COTS software such as:
    - A dramatic increase in customers
    - A large physical expansions of the organisation

# Advantages in purchasing COTS

- There are advantages in purchasing COTS software to be considered as alternatives are evaluated:
  - One advantage is that these products have been refined through the process of commercial use and distribution with additional functionalities available
  - Another advantage is that packaged software is typically extensively tested and thus can be reliable
    - It may also be very well-known (as are any bugs)
    - However: the support should quickly patch any known bugs
  - Increased functionality is often offered with COTS software because a commercial product is likely to have companion products, add-on features, and upgrades that enhance its attractiveness
  - Also analysts often find that the initial cost of COTS software is lower than the cost for either in-house software development or the use of a SaaS provider
  - Another advantage of purchasing COTS packages is that many other companies use them:
    - Thus: analysts are not experimenting on their clients with one-of-a-kind software applications
  - Finally: COTS software boasts an advantage in the help and training that accompanies the purchase of the packaged software

# Downside in the use of COTS

- There is a downside to the use of COTS software (See the theatre company example (pp. 101)):
- Because COTS it is not meant to be fully customizable:
  - The theatre company lost its ability to change the software to include key features in its donor database on which users relied
- COTS may also include errors that could expose an organization to liability issues
- There are other disadvantages to consider with the purchase of COTS including:
  - Packages are programmed and generic rather than being focused on human users working in a business
  - Users must live with whatever features exist in the software (appropriate or not)
  - There is (by design) very limited customizability
- Other disadvantages to purchasing COTS software include:
  - The need to investigating the financial stability of the software vendor
  - The diminished sense of ownership and commitment that is inevitable when the software is considered a product rather than a process

# Systems Perspective

- To get perspective on systems being developed:
  - Over 50% of all projects are built from the ground up:
    - Two-thirds use traditional methods like SDLC
    - One-third using agile or object-oriented technologies
  - Most systems are  developed using an internal systems analysis team
    - Computer programmers may be in-house or outsourced
  - Fewer than 50% of all projects are developed from existing applications or components
    - The  great majority are modified (some extensively)
    - Less than 5% of software is off-the-shelf software that requires no modifications at all

# Using the Services of a SaaS Provider

# Software-as-a-Service

- Organizations may realize some benefits from taking an entirely different approach to procuring software:
    - By outsourcing some of the organization’s software needs to a SaaS provider that specializes in IT applications.
- There are specific benefits to outsourcing applications to a SaaS provider – for example:
    - Organizations that want to retain their strategic focus (and do what they’re best at) may want to outsource the production of information systems applications
    - Additionally: outsourcing software needs means that the organization doing the outsourcing may be able to sidestep the need to hire / train / and retain a large IT staff
        - This can result in significant savings.
    - When an organization uses SaaS:
        - There is little or no expenditure of valuable employee time on nonessential IT tasks (these are handled professionally by the SaaS provider)

# Drawbacks in Using SaaS

- Hiring a SaaS provider should not be considered a magic formula for addressing software requirements
- Drawbacks to the use of SaaS must be seriously considered including:
  - A general loss of control over corporate data / information systems / IT  employees / and even processing and project schedules
  - The  heart of their business is their information and the loss of control may be an issue for an organisation
  - The financial viability of a SaaS provider can be a potential issue
  - There might also be concerns about the security of the organization's data  and records, along with concern about confidentiality of data and client privacy
  - Finally: when choosing a SaaS provider:
    - There is a potential loss of strategic corporate advantage that  might have been gained through the company's deployment of its own innovative applications created by its employees

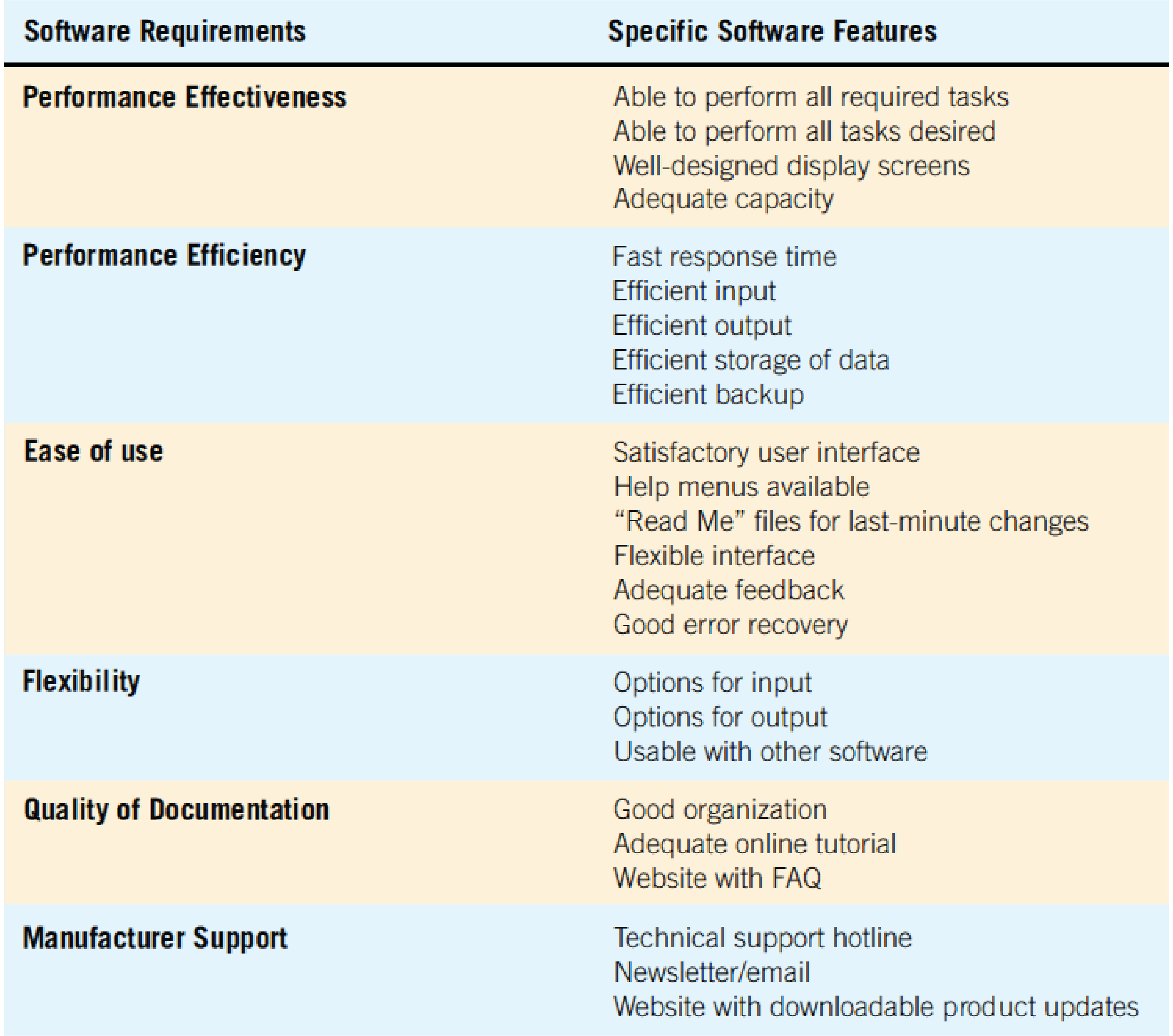
# Evaluation of Vendor Support for Software and SaaS

# Evaluating COTS Vendors

- Whether you purchase a COTS package or contract for SaaS from a provider:
  - You will be  working with vendors who may have their own best interests at heart.
  - You must be willing  to evaluate software with users and not be influenced by a vendors’ sales pitch
- Specifically, there are six main categories on which to grade software (see Figure 3.9)
- Evaluation considerations for packaged software:
  - Base the performance evaluation on test data from your client’s business
  - Examine accompanying documentation (Vendors’ sales descriptions alone are not adequate)
- Vendors typically certify that software is working:
  - However: they do not guarantee that it  will be error-free in every instance (or)
  - That it will not crash when users take incorrect actions (or)
  - That it will be compatible with all other software the organization is currently running
  - Obviously: they will not guarantee their packaged software if it is used in conjunction with  faulty or incompatible hardware

# Evaluating Vendors

- There are six main categories on which to grade software (as shown in Figure 3.9):
  - Performance effectiveness
  - Performance efficiency
  - Ease of use
  - Flexibility
  - Quality of documentation
  - Manufacturer support



| Software Requirements | Specific Software Features |
|---|---|
| **Performance Effectiveness** | Able to perform all required tasks<br>Able to perform all tasks desired<br>Well-designed display screens<br>Adequate capacity |
| **Performance Efficiency** | Fast response time<br>Efficient input<br>Efficient output<br>Efficient storage of data<br>Efficient backup |
| **Ease of use** | Satisfactory user interface<br>Help menus available<br>“Read Me” files for last-minute changes<br>Flexible interface<br>Adequate feedback<br>Good error recovery |
| **Flexibility** | Options for input<br>Options for output<br>Usable with other software |
| **Quality of Documentation** | Good organization<br>Adequate online tutorial<br>Website with FAQ |
| **Manufacturer Support** | Technical support hotline<br>Newsletter/email<br>Website with downloadable product updates |

# Project Management (2)

# Identifying, Forecasting, and Comparing Costs  and Benefits

# Costs and Benefits

- Costs and benefits of a proposed computer system must always be considered together because:
  - They are interrelated and often interdependent
  - While a systems analyst will try to propose a  system that fulfils various information requirements:
    - Decisions to continue with the proposed system will be based on a cost-benefit analysis and not on information requirements
    - In many ways: benefits are measured by costs

# Forecasting

# Key Variables

- Systems analysts are required to predict certain key variables before submitting a proposal to the client
- To some degree a systems analyst will rely on a what-if analysis such as:
  - "What if labour costs rise only 5 percent per year for the next three years, rather than 10 percent?"
- The systems analyst should realize that he or she cannot rely only on what-if analysis for everything if the proposal is to be credible, meaningful, and valuable

# Forecasting Models

- A systems analyst has many forecasting models available
- The main reason for choosing a model is the availability of historical data
- If historical data are unavailable:
  - An analyst must turn to one of the judgment methods:
    - Estimates from the sales force
    - Surveys to estimate customer demand
    - Delphi studies (a consensus forecast developed independently by a group of experts through a series of iterations)
    - Creation of scenarios
    - Historical analogies.

# Conditional or Unconditional Forecasts

- If historical data are available:
  - The next differentiation between classes of techniques involves whether the forecast is *conditional* or *unconditional*
- *Conditional* forecasting implies that there is an association among variables in the model or that such a causal relationship exists
  - Common methods in this group include correlation / regression /leading indicators / econometrics / and input/output models
- *Unconditional* forecasting means the analyst isn’t required to find or identify any causal relationships
  - Consequently: systems analysts find that these methods are low-cost and easy to implement alternatives
  - Included in this group are: graphical judgment / moving averages/ and analysis of time-series data
  - Because these methods are simple, reliable, and cost-effective we will focus on them

# Estimation of Trends

- Trends can be estimated using a number of different research methods (RM):
- One RM is a moving average:
  - This method is useful because some seasonal,  cyclical, or random patterns may be smoothed, leaving the trend pattern
- The principle behind  moving averages is to calculate the arithmetic mean of data from a fixed number of periods
- A  three-month moving average is simply the average of the past three months – for example:
  - The  average sales for January, February, and March are used to predict the sales for April
  - Then the  average sales for February, March, and April are used to predict the sales for May (and so on)
  - When the results are shown in a graphical format (it will be clearly seen that widely fluctuating data are smoothed)

# Estimation of Trends

- The moving average method is useful for its smoothing ability, but it also has disadvantages:
- Moving averages are more strongly affected by extreme values than by using graphical judgment or estimating by using other methods such as least squares
- An analyst should learn forecasting as it often provides information that is valuable in justifying an entire project.
- There are many RM available which should be considered and used where appropriate in a systems analysis

# Identifying Benefits and Costs

# Tangible and Intangible Benefits and Costs

- Benefits and costs can be either:
  - Tangible (or)
  - Intangible
- In systems analysis and design:
  - Both tangible and intangible benefits and costs must considered
  - The result of a systems analysis may include:
    - A *cost-benefit* analysis
    - The *Return on Investment* (RoI)
  - As we have seen:
    - For projects to be implemented there must provide a realistic RoI
    - RoI can be in financial terms (or) human benefits  s

# Tangible Benefits

- Tangible benefits are advantages that are measurable in positive of negative financial terms to the organization using information systems
- Examples of tangible benefits are:
  - An increase in the speed of processing
  - Access to otherwise inaccessible information
  - Improved access to information
  - Leveraging computing calculating power
  - Reductions in the amount of employee time needed  to complete specific tasks
- Although measurement is not always easy:
  - Tangible benefits can be measured in terms of money / resources / time saved / human factors

# Intangible Benefits(1)

- Some benefits derived from the use of an information system are difficult to measure (but are important)
  - They are known as *intangible* benefits
- Intangible benefits include:
  - Improving the decision-making process
  - Improving accuracy
  - Improving competitiveness in customer service
  - Maintaining a good business image
  - Increasing job satisfaction for employees by eliminating tedious tasks

# Intangible Benefits (2)

- As you can see from this list:
  - Intangible benefits are extremely important and can have far-reaching implications for a business as it relates to people both outside and within the organization
- Although intangible benefits of an information system are important factors that must be considered when deciding whether to proceed with a system:
  - A system built solely for its intangible benefits will not be successful.
- You must discuss both tangible and intangible benefits in your proposal because presenting both will enable decision makers in the business to make a well-informed decision about the proposed system

# Tangible Costs

- The concepts of tangible and intangible costs present a conceptual parallel to tangible and intangible benefits
- Tangible costs are costs that a systems analyst and the business's accounting personnel can accurately project including the cost of:
  - Equipment (such as computers and printers)
  - Cost  of resources
  - Cost of systems analysts' time
  - Cost of computer programmers' time
  - Other employees' salaries
- These costs are usually well established or can be discovered quite  easily:
  - They are costs that will require the business to make a cash outlay

# Intangible Costs

- Intangible costs are difficult to estimate and may not be known and include losing a:
    - Competitive edge
    - A reputation for being first with an innovation or the leader in a field
    - Declining company image due to increased customer dissatisfaction
    - Ineffective decision making due to untimely or inaccessible information
- As you can be seen:
    - It is difficult to accurately project financial amount for intangible costs
    - To consider a proposed system and all its implications:
        - Intangible costs that are not quantifiable must be included in the analysis

# Comparing Costs and Benefits

# Comparing Costs and Benefits

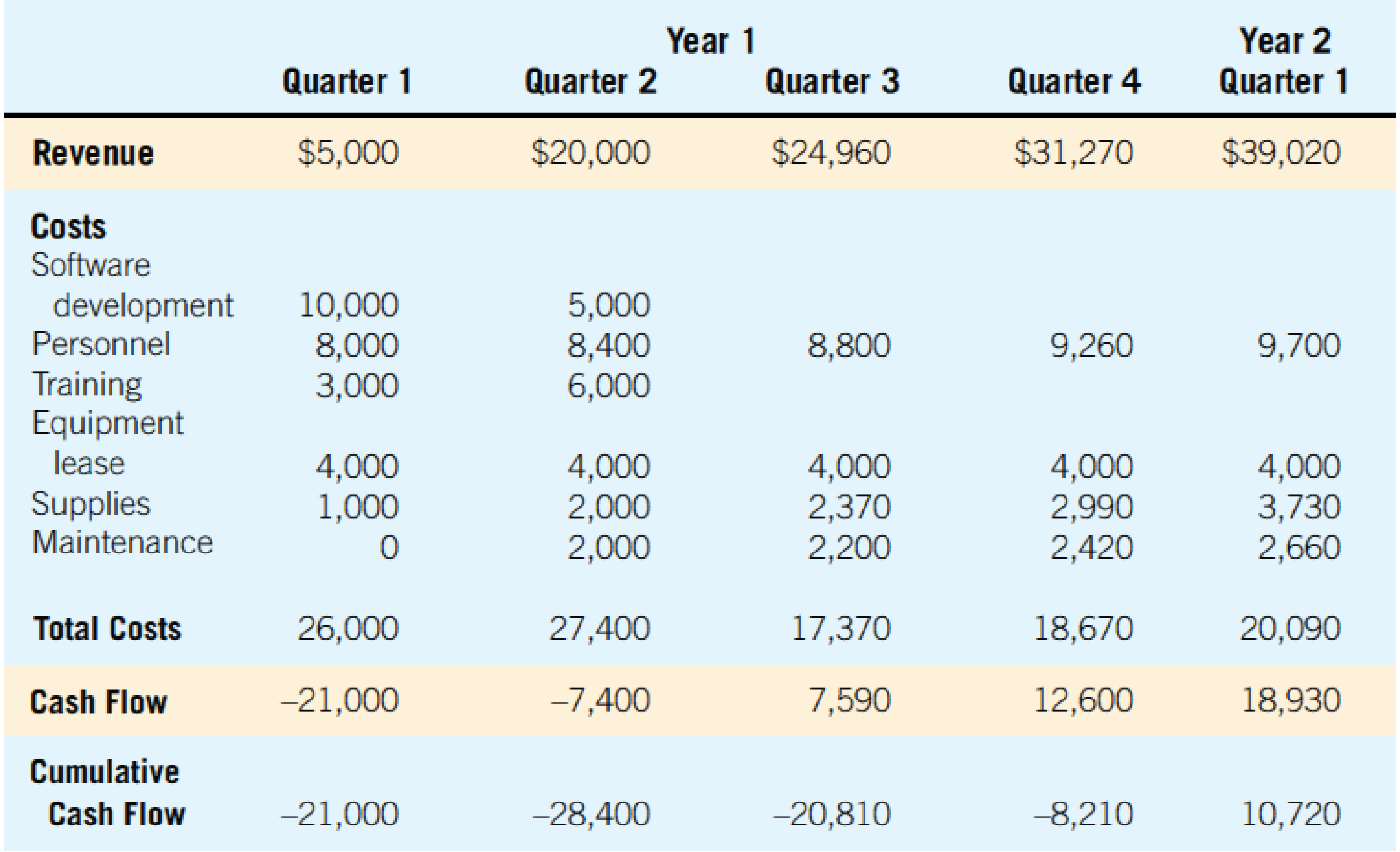
- There are many known techniques for comparing the costs and benefits of a proposed system
- The methods are:
  - Break-even analysis
    - Use break-even analysis if the project needs to be justified in terms of cost, not benefits
  - Payback
    - Use payback when the improved tangible benefits form a convincing argument for the proposed system
- The methods employ:
  - Cash-flow analysis
    - Use cash-flow analysis when the project is expensive, relative to the size of the company
  - Present value analysis
    - Use present value when the payback period is long or when the cost of borrowing money is high

# Comparing Costs and Benefits

- Cash-flow analysis:
  - Examines the direction, size, and pattern of cash flow that is associated with the proposed information system
  - Determines when cash outlays and revenues will occur for both; not only for the initial purchase, but over the life of the information system
- Present value analysis:
  - Assess all the economic outlays and revenues of the information system over its economic life
  - To compare costs today with future costs and today's benefits with future benefits
  - Presents the time value of the investment in the information system as well as the cash flow

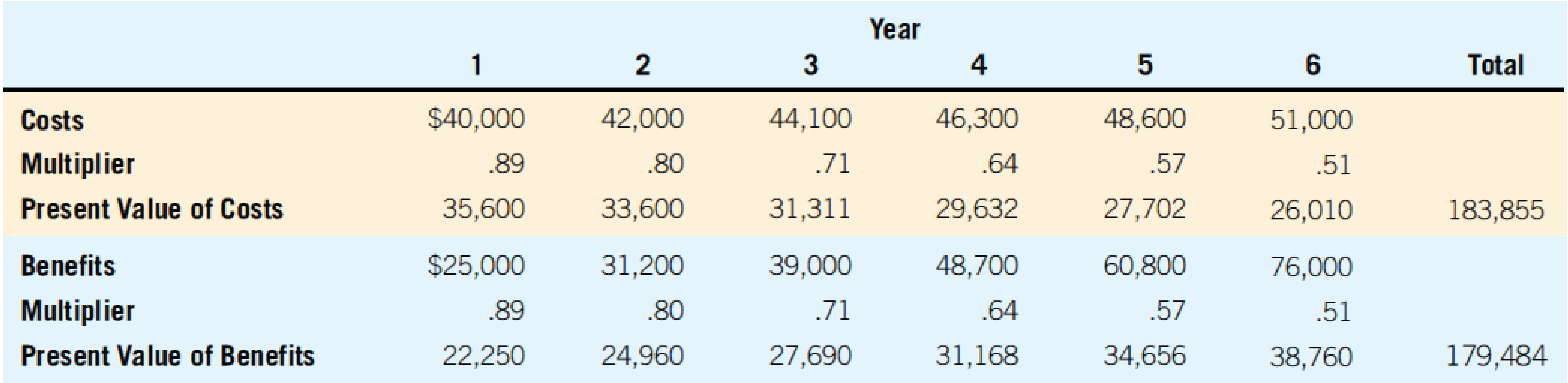
# Comparing Costs and Benefits

- Cash-Flow Analysis for the Computerized Mail-Addressing System
- Figure 3.12



| | | Year 1 | | | Year 2 |
|---|---|---|---|---|---|
| | Quarter 1 | Quarter 2 | Quarter 3 | Quarter 4 | Quarter 1 |
| **Revenue** | $5,000 | $20,000 | $24,960 | $31,270 | $39,020 |
| **Costs** | | | | | |
| Software development | 10,000 | 5,000 | | | |
| Personnel | 8,000 | 8,400 | 8,800 | 9,260 | 9,700 |
| Training | 3,000 | 6,000 | | | |
| Equipment lease | 4,000 | 4,000 | 4,000 | 4,000 | 4,000 |
| Supplies | 1,000 | 2,000 | 2,370 | 2,990 | 3,730 |
| Maintenance | 0 | 2,000 | 2,200 | 2,420 | 2,660 |
| **Total Costs** | 26,000 | 27,400 | 17,370 | 18,670 | 20,090 |
| **Cash Flow** | –21,000 | –7,400 | 7,590 | 12,600 | 18,930 |
| **Cumulative Cash Flow** | –21,000 | –28,400 | –20,810 | –8,210 | 10,720 |

# Present Value Analysis



| | Year | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | Total |
| **Costs** | $40,000 | 42,000 | 44,100 | 46,300 | 48,600 | 51,000 | |
| **Multiplier** | .89 | .80 | .71 | .64 | .57 | .51 | |
| **Present Value of Costs** | 35,600 | 33,600 | 31,311 | 29,632 | 27,702 | 26,010 | 183,855 |
| **Benefits** | $25,000 | 31,200 | 39,000 | 48,700 | 60,800 | 76,000 | |
| **Multiplier** | .89 | .80 | .71 | .64 | .57 | .51 | |
| **Present Value of Benefits** | 22,250 | 24,960 | 27,690 | 31,168 | 34,656 | 38,760 | 179,484 |

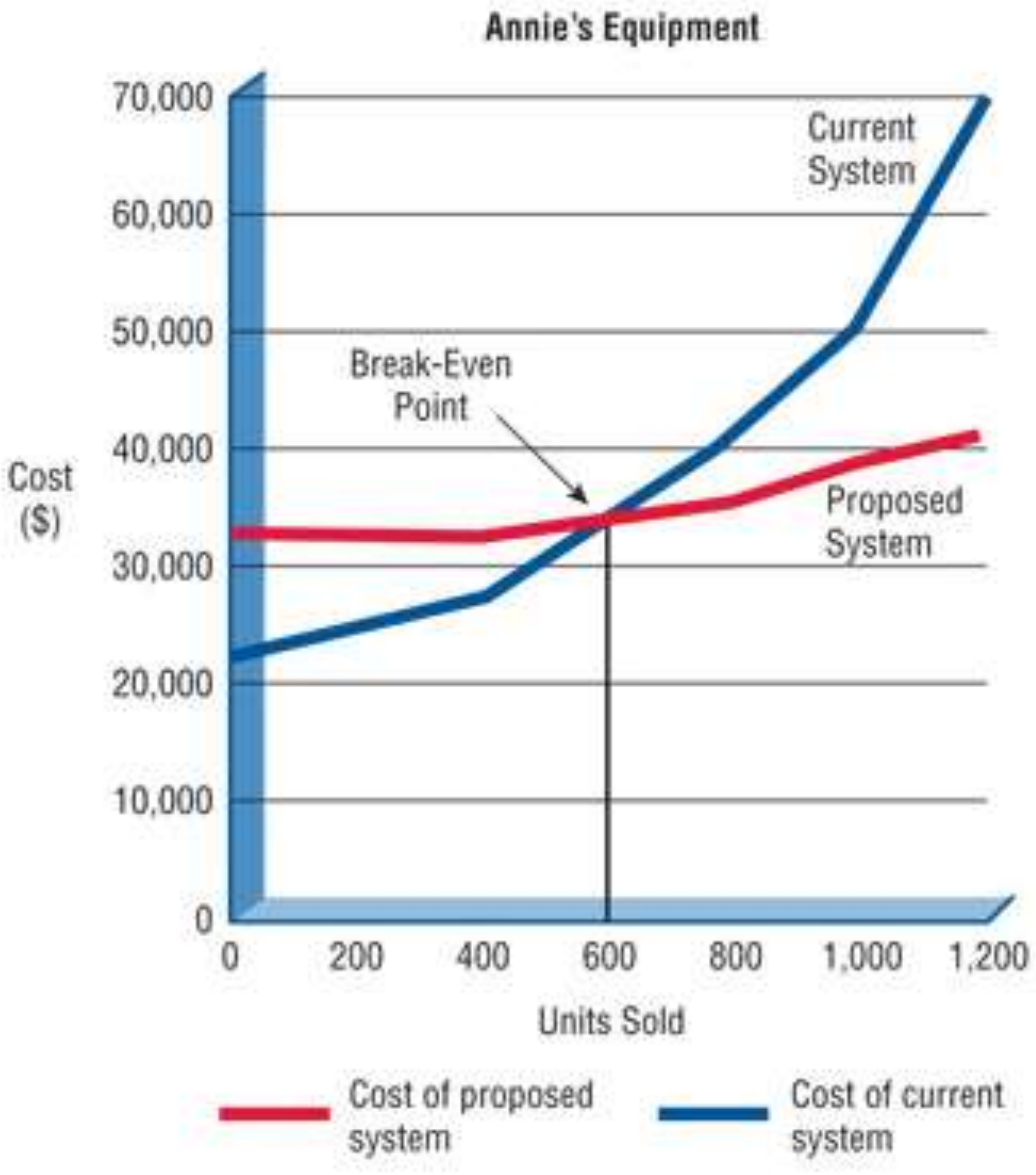
- Taking into account present value – the conclusion is that the costs are greater than the benefits
- The discount rate (*i*), is assumed to be (.12) in calculating the multipliers in this table

# Break-Even Analysis

- By comparing costs alone a systems analyst can use break-even analysis to
  - Determine the break-even capacity of a proposed information system
  - The point at which the total costs of the current system and the proposed system intersect represents  the break-even point
    - The point where it becomes profitable for the business to get the new information system
    - Useful when a business is growing and volume is a key variable in costs
  - Total costs include:
    - Costs that recur during operation of a system plus the developmental costs that occur only once (one-time costs of installing a new system)

# Break-Even Analysis

- Figure 3.10 illustrates a break-even analysis on a small store that maintains inventory using a manual system:
  - As volume rises the costs of the manual system rise at an increasing rate
  - A new computer system would cost a substantial sum up front, but the incremental costs for higher volume would be rather small
  - The graph shows that the computer system would be cost-effective if the business sold around 600 units per week



# Payback

- Payback is useful when a business is growing and volume is a key variable in costs
- A disadvantage of break-even analysis is that benefits are assumed to remain the same regardless  of which system is in place.
- From our study of tangible and intangible benefits:
  - We can see that this  is clearly not the case
  - Payback analysis can determine how long it will take for the benefits of the system to pay back the costs of developing it
- Figure 3.11 illustrates a system with a payback period of three  and a half years

# Break-Even Analysis Showing a Payback Period

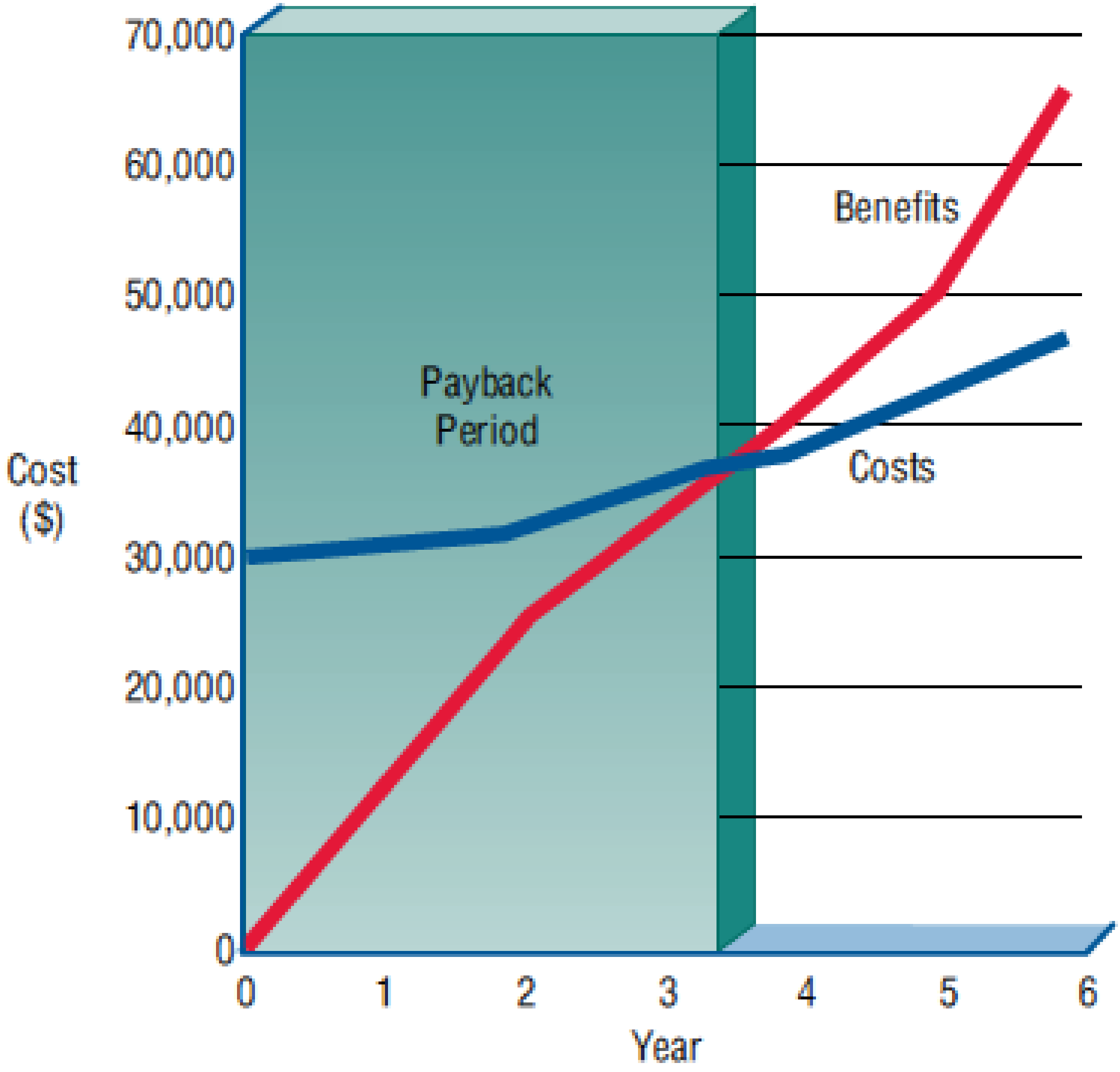


**FIGURE 3.11**
Payback analysis showing a payback period of three and a half years.

| Year | Cost ($) | Cumulative Costs ($) | Benefits ($) | Cumulative Benefits ($) |
|---|---|---|---|---|
| 0 | 30,000 | 30,000 | 0 | 0 |
| 1 | 1,000 | 31,000 | 12,000 | 12,000 |
| 2 | 2,000 | 33,000 | 12,000 | 24,000 |
| 3 | 2,000 | 35,000 | 8,000 | 32,000 |
| 4 | 3,000 | 38,000 | 8,000 | 40,000 |
| 5 | 4,000 | 42,000 | 10,000 | 50,000 |
| 6 | 4,000 | 46,000 | 15,000 | 65,000 |

# Managing Time and Activities

# Managing Time and Activities

- The process of analysis and design:
  - Can become unwieldy
  - Especially when the system being  developed is large
- To keep the development activities as manageable as possible
  - A systems analyst can utilise the techniques of project management to achieve an organized and productive approach

# The Work Breakdown Structure

# Developing a WBS

- The WBS approach can use:
  - Decomposition:
    - Starting with large ideas
    - Then breaking them down into manageable activities
  - Product oriented:
    - Building a website can be broken down into many parts
  - Process-oriented:
    - Emphasizes the importance of each phase

# Work Breakdown Structures

- Systems analysts are responsible for completing projects on time and within budget and for including the features in the proposal:
- To accomplish all three of these goals:
  - A project needs to be broken down  into smaller tasks or activities which taken together make up a work breakdown structure (WBS)
- When defined properly, the tasks that comprise a WBS have special properties:
  1. Each task or activity contains one deliverable, or tangible outcome, from the activity
  2. Each task can be assigned to a single individual or a single group
  3. Each task has a responsible person monitoring and controlling performance
  4. Activities in a work breakdown structure do not need to take the same amount of time or involve the same number of team members

# Work Breakdown Structures

- The activities defined must add up to 100 percent of the work (however):
  - The main method for developing a WBS is decomposition,(or)
  - Starting with large ideas and then breaking them down into manageable activities.
  - This subdivision of ideas into smaller ideas and eventually into tasks stops when each task has only one deliverable.
  - There are different types of work breakdown structures
- A WBS can be product oriented:
  - Building a website can be broken down into many parts (with each set of pages having a specific purpose)
  - A website can be divided into its home page / product description pages / FAQ page / a contact page / ecommerce page
  - Each of these pages could contain activities that you could use in your work breakdown structure.

# Process-Oriented Approach to WBS

- An alternative approach is the creation of a process-oriented work breakdown structure
- An example of  this is shown in Figure 3.12
  - This type of WBS is typical in systems analysis and design
  - In this example:
    - We show the development of a website
    - But rather than show the development of each page
    - This example emphasizes the importance of each phase in the systems development life cycle

# Time Estimation Techniques

# Time Estimation

- Time estimation techniques:
  - Relying on experience
  - Using analogies
  - Using three-point estimation
  - Identifying function points
  - Using time estimation software
- For details of the time estimation techniques:
  - See page 108
  - Where the calculations are detailed with descriptive text

# Project Scheduling

# IMPORTANT

- In this part of the course we introduce project management and scheduling techniques
- It is important to note that:
  - In the ‘real-world’ project programmes (schedules) **NEVER** work as planned
  - The project schedule is a means by which the actual progress can be measured and the current state of a project (*ahead* (or) *behind* the planned progress) identified
  - Projects will experience delays due to:
    - Unforeseen events
    - Additional work instructed by the client
- The principal purpose of a project plan is to manage the project to:
  - Understand how additional work (there is always variations) can be planned and implemented
  - Achieve the planned completion date (or the earliest completion date)
  - Take the required action (practical and financial) to manage the project

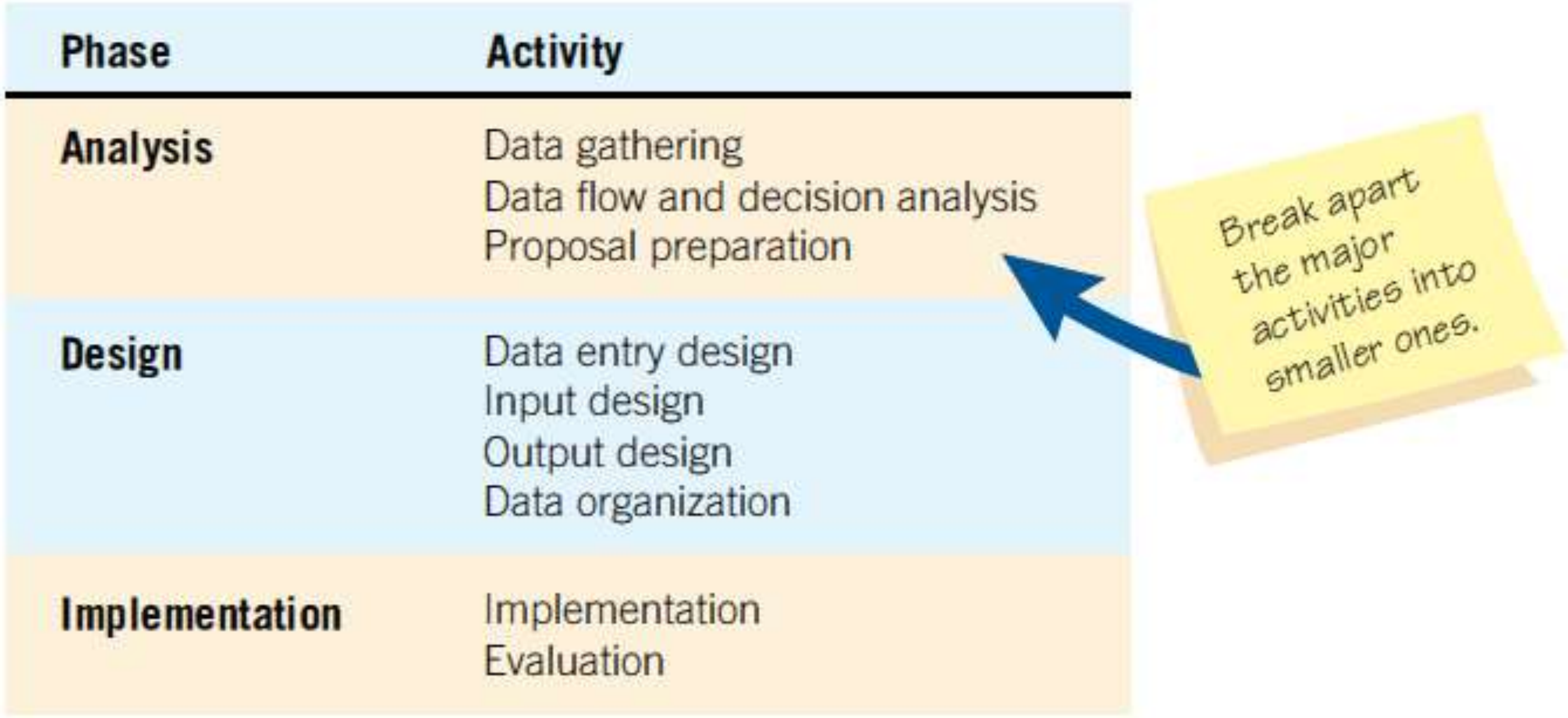
# Project Planning

- Planning includes all the activities required to
  - Select a systems analysis team
  - Assign members of  the team to appropriate projects
  - Estimate the time required to complete each task
  - Prepare a schedule for the project (the tasks with dates and duration)
- Control means using feedback to:
  - Monitor the project
  - Comparing the plan for the project with its actual evolution
  - Taking appropriate action to expedite or reschedule activities to meet project deadlines
  - Motivating team members

# Project Planning

- This section contains an example where a systems analyst:
    - Act as a project manager
    - Begins with the basic activities of analysis, design, and implementation
- The analyst then implements the following steps:
    - The main activities are decomposed into smaller subtasks (as shown in Figure 3.13)
    - The analysis phase is further broken down into:
        - Data gathering / data flow and decision analysis / proposal preparation
    - Design is broken down into:
        - Data entry design / input and output design / data organization
    - The implementation phase is divided into implementation and evaluation

# Project Planning (Figure 3.13)



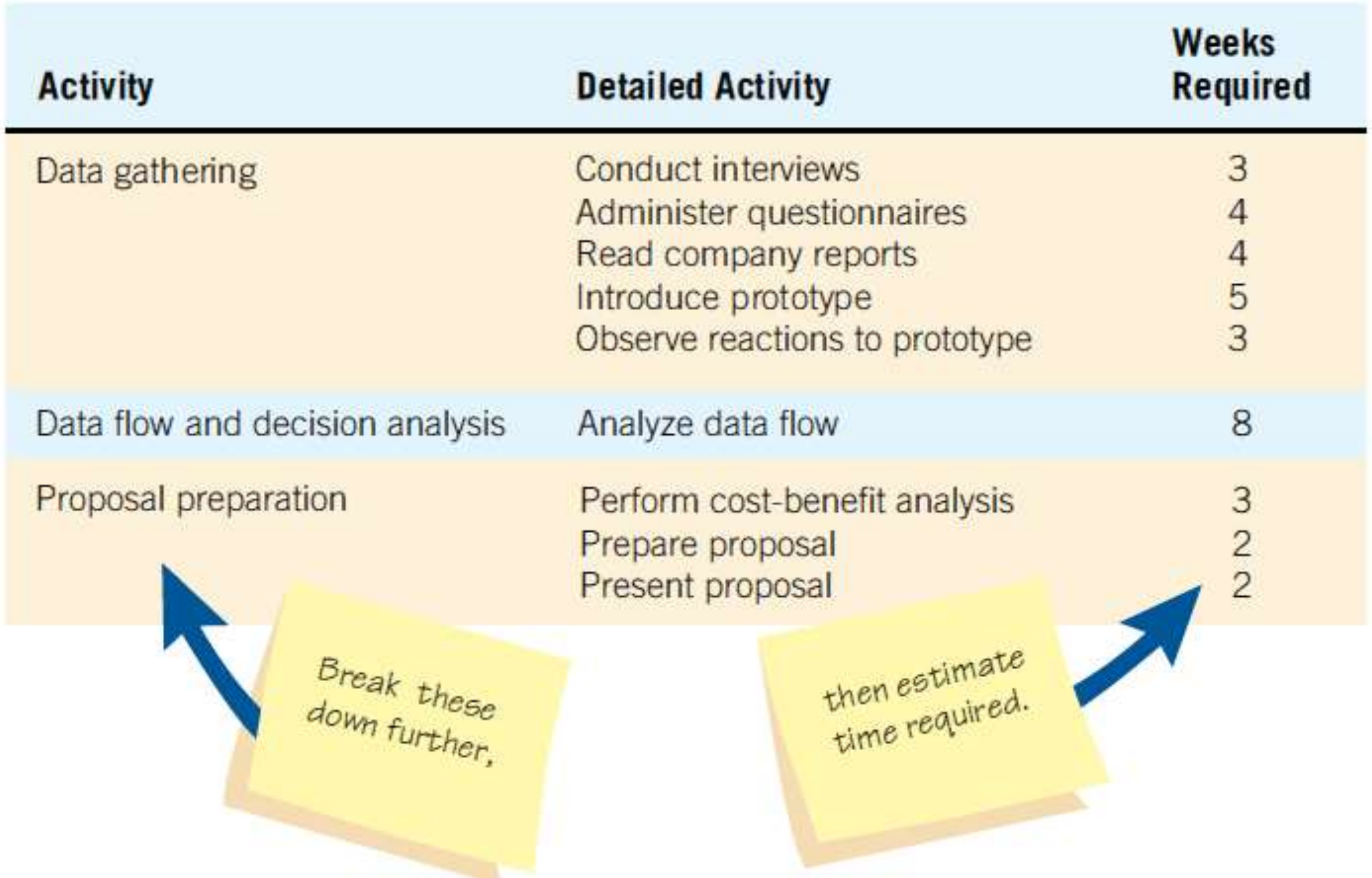
| Phase | Activity |
|---|---|
| Analysis | Data gathering<br>Data flow and decision analysis<br>Proposal preparation |
| Design | Data entry design<br>Input design<br>Output design<br>Data organization |
| Implementation | Implementation<br>Evaluation |

Break apart the major activities into smaller ones.

- Beginning to Plan a Project by Breaking it into Three Major Activities:
  - Analysis
  - Design
  - Implementation

# Project Planning (Figure 3.14)

- FIGURE 3.14 shows:
- Refining the planning
- Scheduling of analysis activities
- Adding detailed tasks
- Establishing the time required to complete each task(s)



| Activity | Detailed Activity | Weeks Required |
|---|---|---|
| Data gathering | Conduct interviews | 3 |
| | Administer questionnaires | 4 |
| | Read company reports | 4 |
| | Introduce prototype | 5 |
| | Observe reactions to prototype | 3 |
| Data flow and decision analysis | Analyze data flow | 8 |
| Proposal preparation | Perform cost-benefit analysis | 3 |
| | Prepare proposal | 2 |
| | Present proposal | 2 |

Break these down further,

then estimate time required.

# Project Planning

- In subsequent steps:
  - The analysis must to consider each of these tasks and break them down further so that planning and scheduling can take place
  - Figure 3.14 shows how  the analysis phase is described in more detail – for example:
    - Data gathering is broken down into five activities from conducting interviews to observing reactions to the prototype system
    - This  particular project requires a data flow analysis but not a decision analysis
      - So the systems analyst  has written in “analyse data flow” as the single step in the middle phase
    - Finally: proposal  preparation is broken down into three steps:
      - Perform cost-benefit analysis / prepare proposal /present proposal

# Project Planning

- The plan can be broken down further – for example:
    - The analyst could specify each of the people to be interviewed
    - The amount of detail necessary depends on the project
        - But all critical steps need to appear in the project plan(s)
    - Potentially the most difficult part of project planning is estimating the  time it takes to complete each task or activity:
        - Post project reports have cited the following reasons for lateness on a particular project: “Poor scheduling estimates”
- There is no substitute for experience in estimating time requirements:
    - Systems analysts who have had the opportunity of an apprenticeship or training in project management are fortunate

# Project Planning

- Project management has tried to reduce the inherent uncertainty in determining time estimates by:
  - Projecting most likely pessimistic and optimistic estimates (and)
  - Use a weighted average formula to determine the expected time an activity will take
    - This approach offers little more in  the way of confidence
  - The best strategy is to adhere to a structured approach in identifying activities and describing these activities in sufficient detail
  - In  this manner the systems analyst will at least be able to limit unpleasant surprises (mitigate the impact)
  - It is usual for any project plan to include some time for “contingencies”
    - All projects will experience unforeseen problems

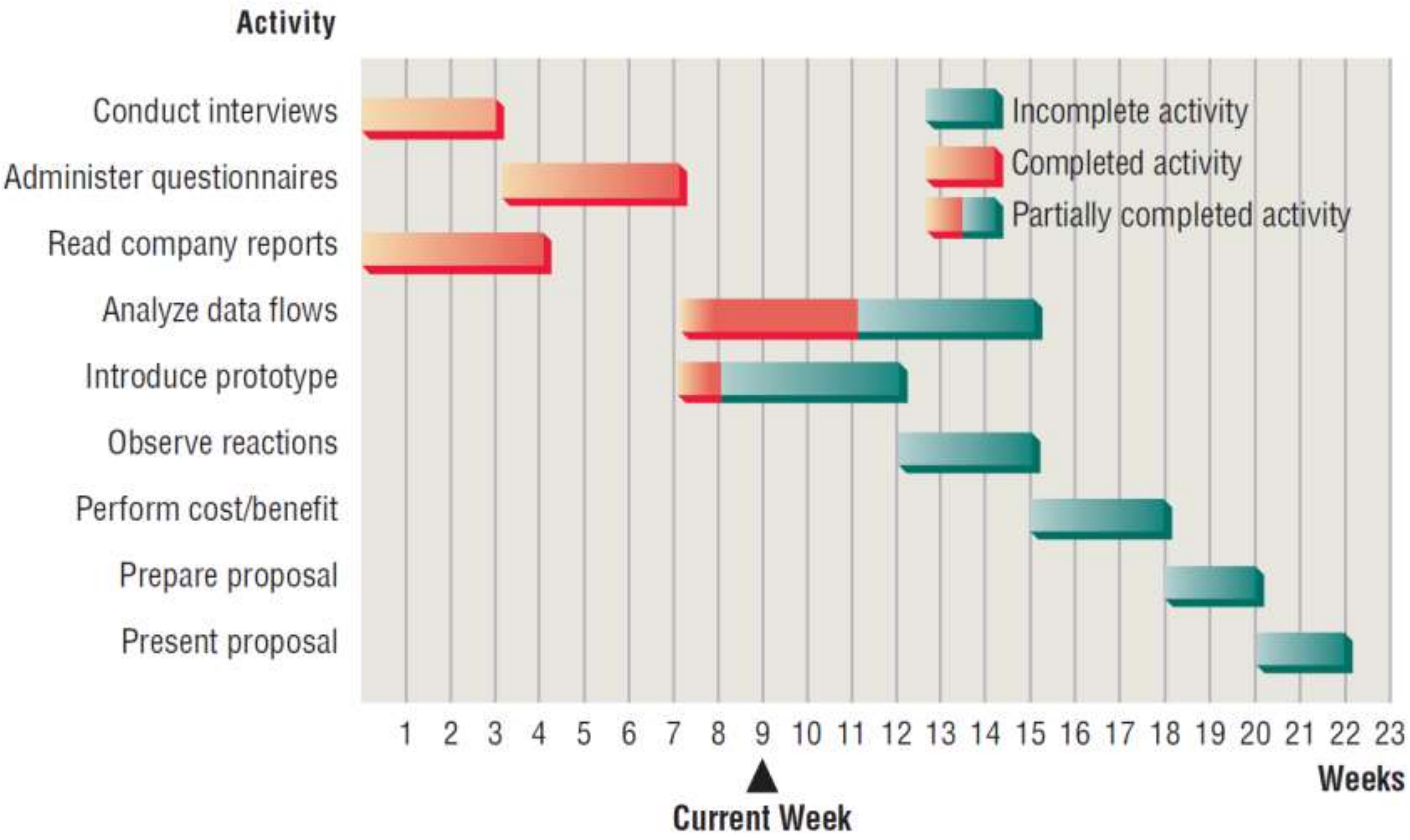
# Using Gantt Charts for Project Scheduling

# Gantt Charts

- A Gantt chart is a planning tool that enables you to easily plan and schedule tasks (time and other resources)
  - It may be created using planning software such as Microsoft Office (Project Manager: see the following screen shots)
- In a Gantt chart tasks are represented using “bars” where:
  - The length of each bar represents the relative length of the task
  - Figure 3.15 is an example of a two-dimensional Gantt chart in which:
    - Time is indicated on  the horizontal dimension
    - The description of activities is shown in the vertical dimension.

# A Gantt Chart Example (Figure 3.15)

- Using a Two-Dimensional Gantt Chart for Planning Activities that Can Be Accomplished in Parallel
- Note:
  - The time allocated for tasks
  - Tasks overlap



# Gantt Charts

- In the example (see Figure 3.15) the chart shows the analysis for the information-gathering phase of the project
- Shown are the tasks with the planned completion dates and the current progress:
  - The time allocated for tasks is shown (conducting interviews: three weeks / administering the questionnaire (four weeks) / etc
  - These activities overlap
  - In the chart the *special symbol* ▲ signifies the current date (it is week 9)
  - The bars with *colour shading* represent projects or parts of projects that have been completed
  - The chart shows that the project is in delay (*late*) in introducing prototypes but *ahead* in analysing data flows

# Gantt Charts

- In the example (see Figure 3.15):
  - Action must be taken on  introducing prototypes soon so that other activities or even the project itself will not be delayed  as a result.
- The main advantage of a Gantt chart is its simplicity
  - This technique easy to use
  - It also lends itself to worthwhile communication with:
    - All stakeholders in the project (including end users)
  - An advantage of using  a Gantt chart is that the bars representing activities or tasks are drawn to scale
    - The size  of the bars indicates the relative length of time it will take to complete each task
  - You can use  Microsoft Office Project Manager to develop a Gantt chart

PROJECTMANAGER

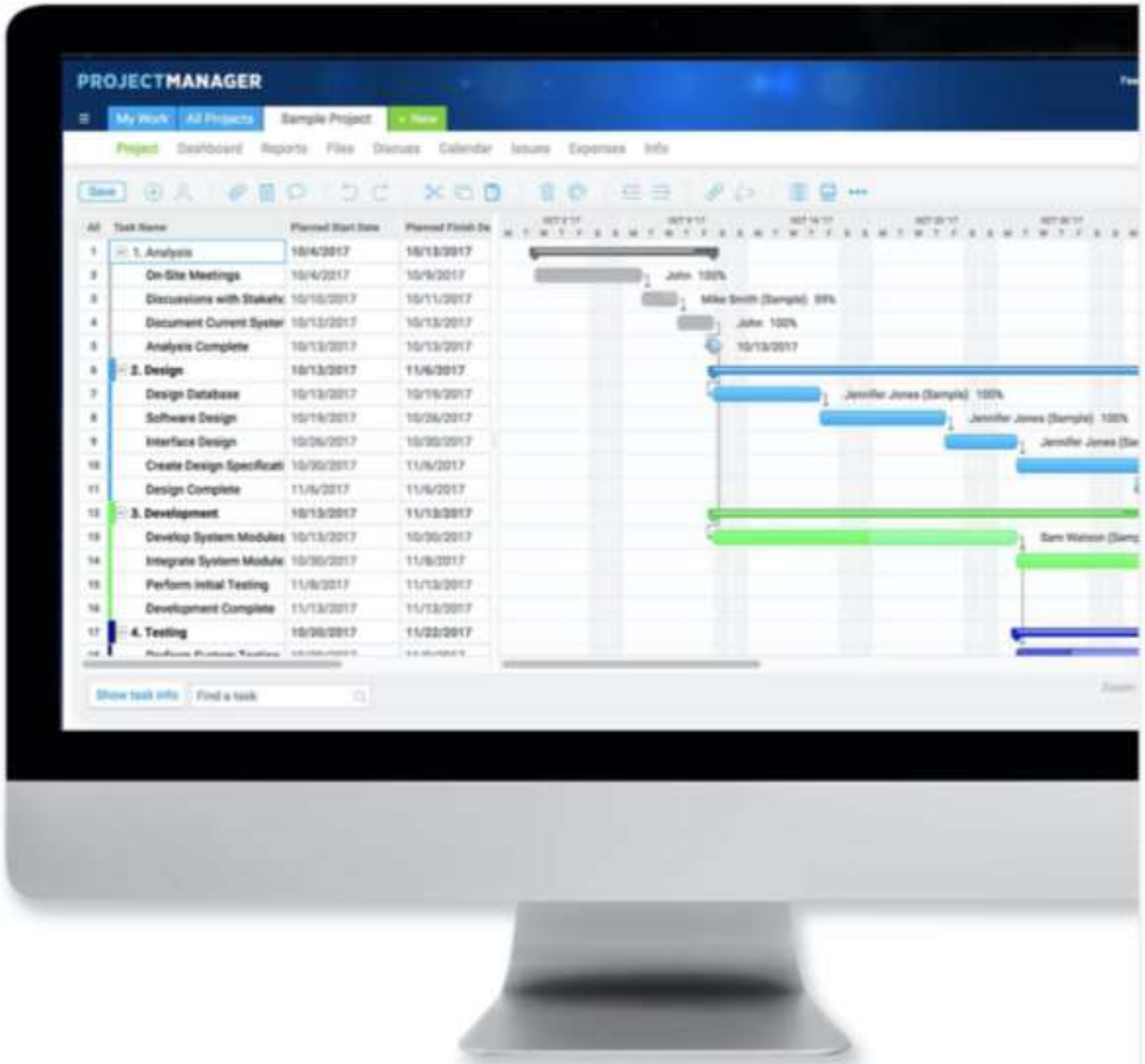
START FREE TRIAL

# Award Winning Microsoft Project Alternative

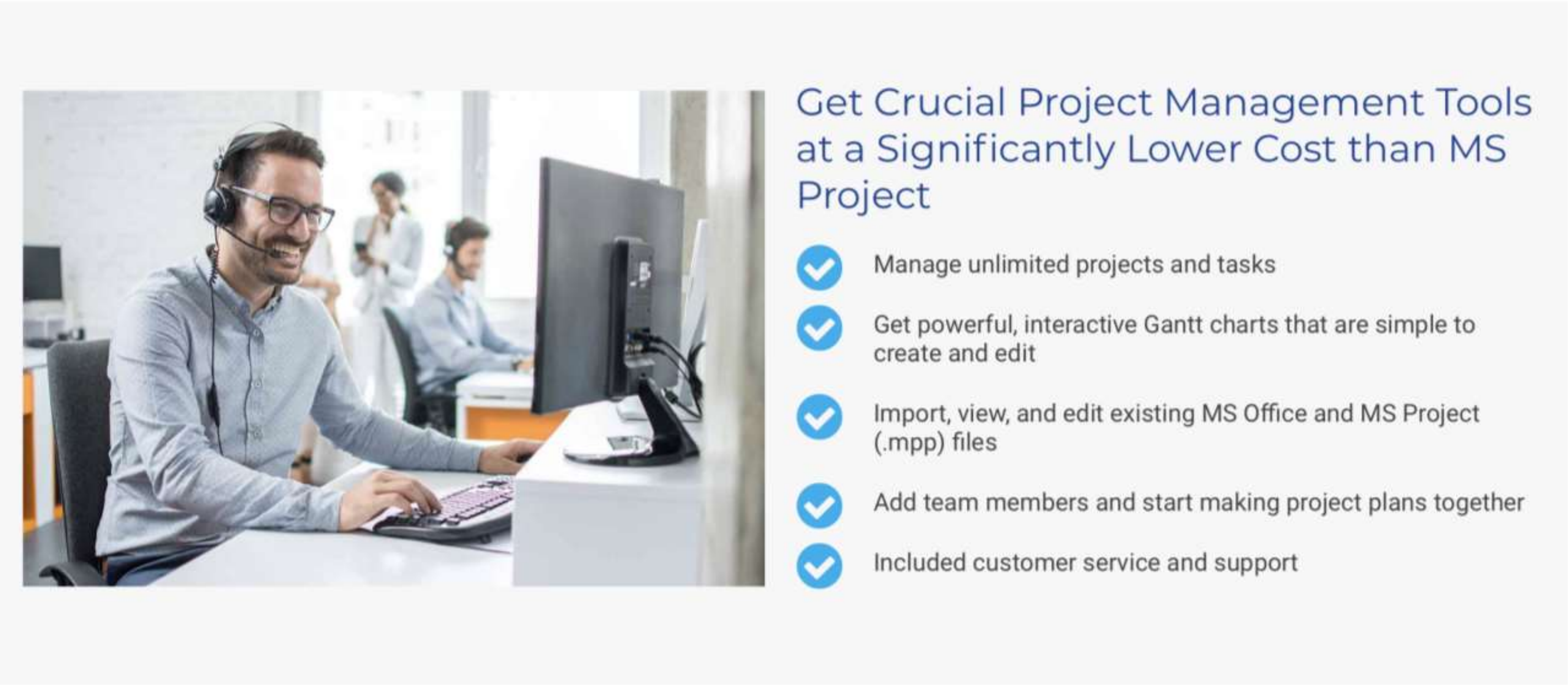
Try our user-friendly software built for project managers with large teams, lots of deadlines, and complicated projects.

START YOUR 30-DAY FREE TRIAL

Web based, works with all modern browsers and computers



| All | Task Name | Planned Start Date | Planned Finish Da |
|---|---|---|---|
| 1 | 1. Analysis | 10/4/2017 | 10/13/2017 |
| 2 | On-Site Meetings | 10/4/2017 | 10/9/2017 |
| 3 | Discussions with Stakeh | 10/10/2017 | 10/11/2017 |
| 4 | Document Current Syster | 10/12/2017 | 10/13/2017 |
| 5 | Analysis Complete | 10/13/2017 | 10/13/2017 |
| 6 | 2. Design | 10/13/2017 | 11/6/2017 |
| 7 | Design Database | 10/13/2017 | 10/19/2017 |
| 8 | Software Design | 10/19/2017 | 10/26/2017 |
| 9 | Interface Design | 10/26/2017 | 10/30/2017 |
| 10 | Create Design Specificati | 10/30/2017 | 11/6/2017 |
| 11 | Design Complete | 11/6/2017 | 11/6/2017 |
| 12 | 3. Development | 10/13/2017 | 11/13/2017 |
| 13 | Develop System Modules | 10/13/2017 | 10/30/2017 |
| 14 | Integrate System Module | 10/30/2017 | 11/8/2017 |
| 15 | Perform Initial Testing | 11/8/2017 | 11/13/2017 |
| 16 | Development Complete | 11/13/2017 | 11/13/2017 |
| 17 | 4. Testing | 10/30/2017 | 11/22/2017 |



## Get Crucial Project Management Tools at a Significantly Lower Cost than MS Project

- ✓ Manage unlimited projects and tasks
- ✓ Get powerful, interactive Gantt charts that are simple to create and edit
- ✓ Import, view, and edit existing MS Office and MS Project (.mpp) files
- ✓ Add team members and start making project plans together
- ✓ Included customer service and support

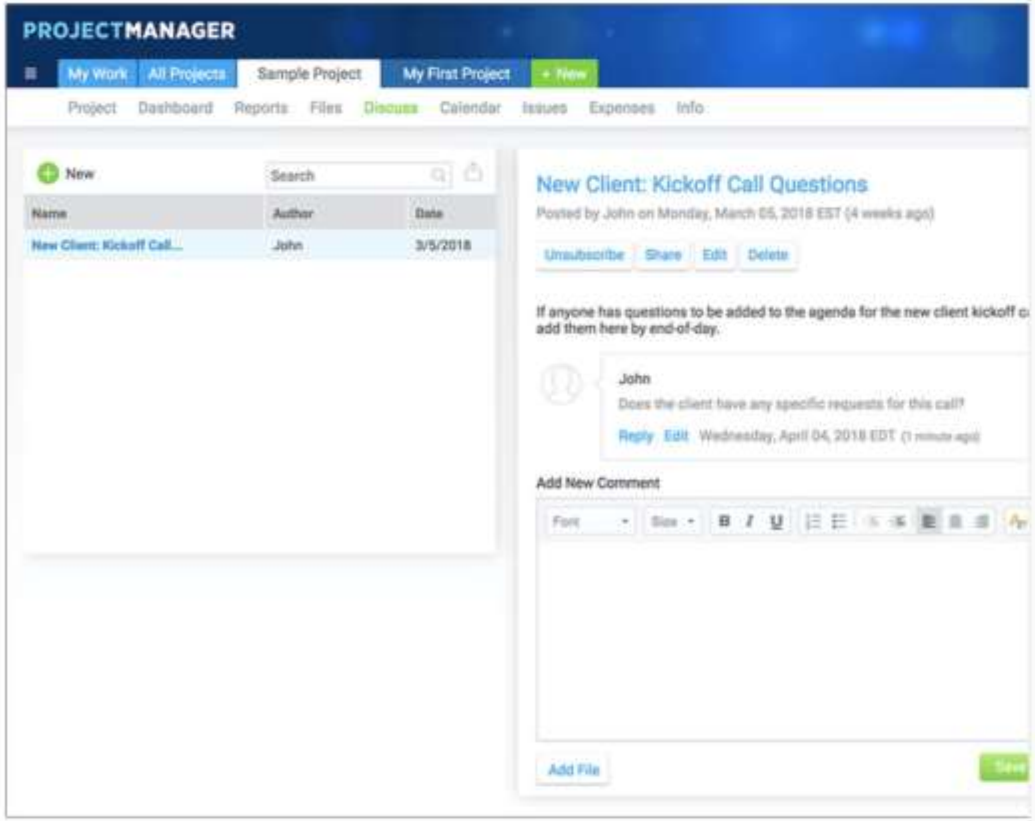
**PROJECTMANAGER**

START FREE TRIAL

Included customer service and support

## Collaborate with coworkers in real time

- ✓ Add team members to projects and create their tasks in seconds
- ✓ Use on any computer, including on Mac
- ✓ Collaborate on projects in real time
- ✓ Create Gantt charts in seconds
- ✓ Add comments, files and documents directly to tasks
- ✓ Automatic email alerts are sent to update the right people at the right time



| Name | Author | Date |
|---|---|---|
| New Client: Kickoff Call... | John | 3/5/2018 |

New Client: Kickoff Call Questions

Posted by John on Monday, March 05, 2018 EST (4 weeks ago)

If anyone has questions to be added to the agenda for the new client kickoff c add them here by end-of-day.

John

Does the client have any specific requests for this call?

Reply Edit Wednesday, April 04, 2018 EDT (1 minute ago)

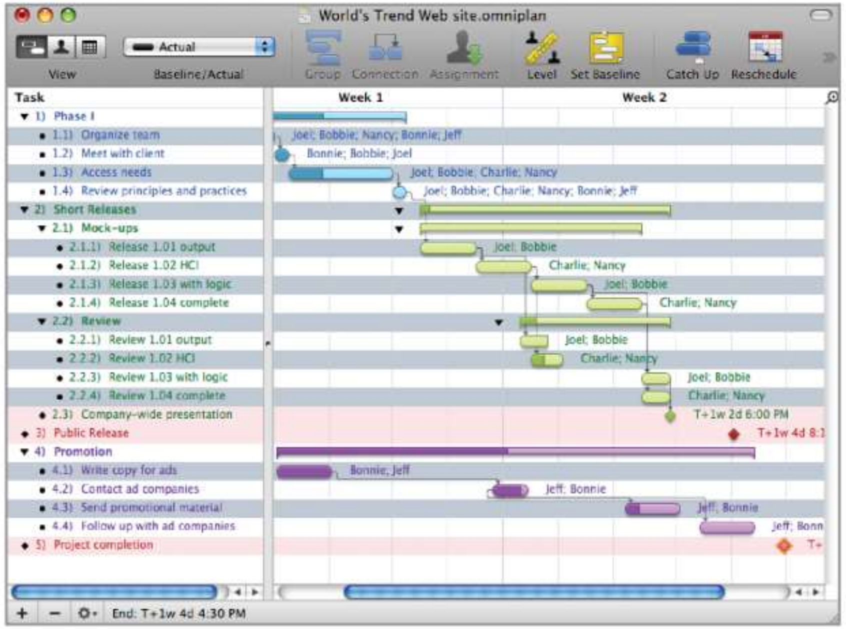
Add New Comment

# Mac Appeal Example

- Color-coding helps a project manager sort out similar phases, tasks, and resources.
    - *OmniPlan* (available for Macs) takes advantage of color-coding to set up a project, identify tasks, identify the critical path, and flag impossible situations.
    - OmniPlan project management software from The Omni Group. (Used by permission of the Omni Group)

## MAC APPEAL

Color-coding helps a project manager sort out similar phases, tasks, and resources. OmniPlan, available for Macs, takes advantage of color-coding to set up a project, identify tasks, identify the critical path, and flag impossible situations.



| Task | |
|---|---|
| 1) Phase I | |
| 1.1) Organize team | Joel; Bobbie; Nancy; Bonnie; Jeff |
| 1.2) Meet with client | Bonnie; Bobbie; Joel |
| 1.3) Access needs | Joel; Bobbie; Charlie; Nancy |
| 1.4) Review principles and practices | Joel; Bobbie; Charlie; Nancy; Bonnie; Jeff |
| 2) Short Releases | |
| 2.1) Mock-ups | |
| 2.1.1) Release 1.01 output | Joel; Bobbie |
| 2.1.2) Release 1.02 HCI | Charlie; Nancy |
| 2.1.3) Release 1.03 with logic | Joel; Bobbie |
| 2.1.4) Release 1.04 complete | Charlie; Nancy |
| 2.2) Review | |
| 2.2.1) Review 1.01 output | Joel; Bobbie |
| 2.2.2) Review 1.02 HCI | Charlie; Nancy |
| 2.2.3) Review 1.03 with logic | Joel; Bobbie |
| 2.2.4) Review 1.04 complete | Charlie; Nancy |
| 2.3) Company-wide presentation | T+1w 2d 6:00 PM |
| 3) Public Release | T+1w 4d 8:1 |
| 4) Promotion | |
| 4.1) Write copy for ads | Bonnie; Jeff |
| 4.2) Contact ad companies | Jeff; Bonnie |
| 4.3) Send promotional material | Jeff; Bonnie |
| 4.4) Follow up with ad companies | Jeff; Bonn |
| 5) Project completion | T+ |

End: T+1w 4d 4:30 PM

**FIGURE 3.MAC**

OmniPlan project management software from The Omni Group. (Used by permission of the Omni Group.)

# Using PERT Diagrams

# PERT Diagrams Overview

- PERT diagrams are an alternative approach to project planning and scheduling
  - PERT charts are called network diagrams in Microsoft Project
- The critical path method (CPM) (or critical path analysis (CPA)) is an algorithm for scheduling a set of project activities
  - It is commonly used in conjunction with the *Program Evaluation and Review Technique* (PERT)
  - A critical path is determined by identifying the longest stretch of dependent activities and measuring the time required to complete them from start to finish
- PERT diagrams have advantages:
  - Clear identification of the order the tasks must be executed
  - Clear identification of the critical path (and thus critical activities)
  - Clear determination of “float” in a project program

# PERT Diagrams

- The PERT method:
  - Is represented by a network of nodes and arrows that are evaluated to:
    - Determine project flow and critical activities
    - Improve the schedule if necessary
    - Review progress once the project is undertaken
- PERT was developed in the late 1950s for use in the U.S. Navy's Polaris nuclear submarine project
  - It reportedly saved the U.S. Navy two years of development time.
- PERT is useful when activities are executed concurrently and in parallel (rather than in sequence)
- A systems analyst can benefit from using PERT by applying it to systems projects on a smaller scale
  - Especially when some team members can be working on certain activities at the same time that  other team members are working on other tasks

# PERT Diagrams

- Figure 3.16 compares a simple Gantt chart with a PERT diagram
- Arrows in the PERT diagram represent the activities expressed as bars in the Gantt chart
- The length of the arrows have no direct relationship with the activity durations
- In the PERT example shown:
  - Activity C may not be started until activity A is completed
  - Precedence is not indicated at all in the Gantt chart:
    - Therefore: it is not possible to tell whether activity C is scheduled to start on day 4 by design (or) by coincidence

# PERT Diagram Example

- Circles on the PERT diagram are called events and can be identified by:
  - Numbers / letters / or any other arbitrary form of designation
  - The circular nodes are present to:
    1. Recognize that an activity is completed
    2. Indicate which activities need to be completed before new activities may be undertaken (which is called precedence)

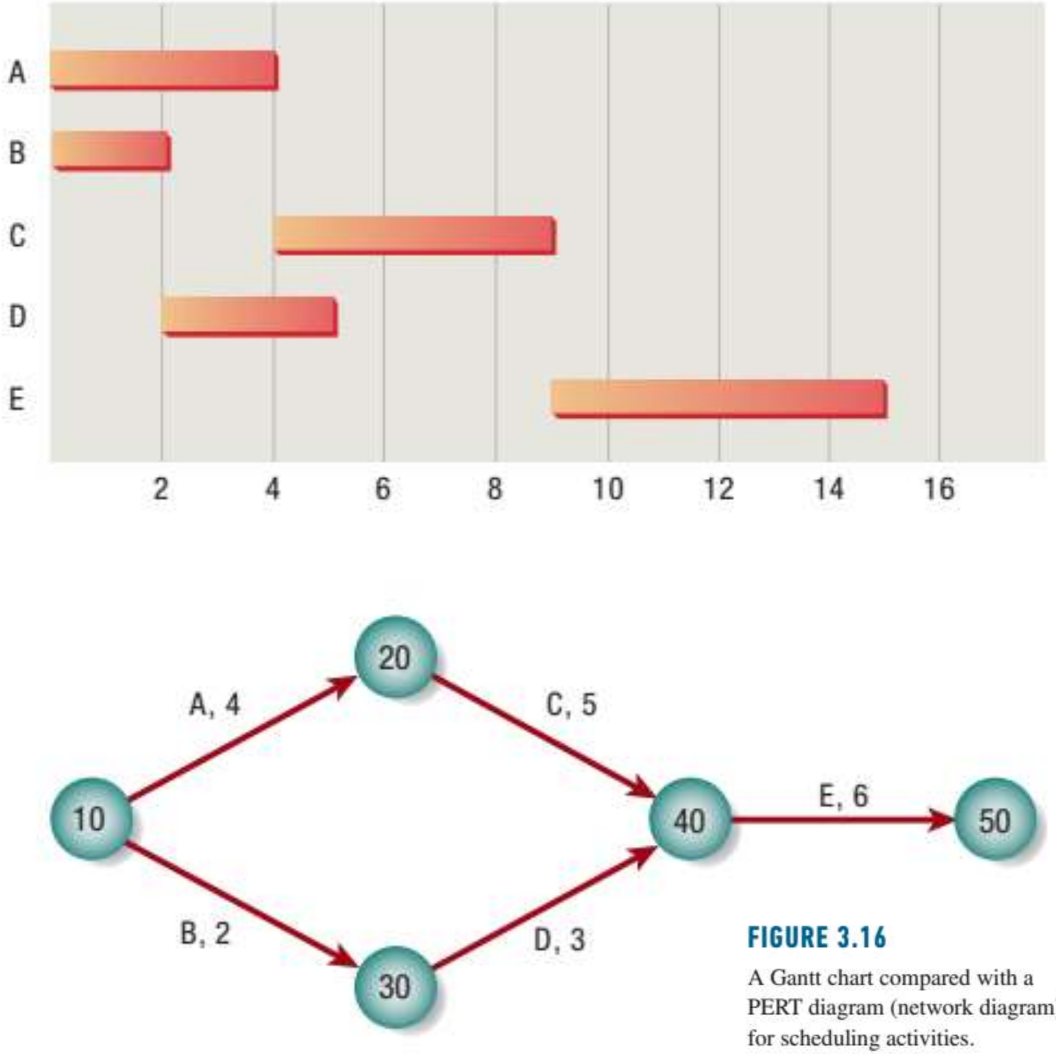


**FIGURE 3.16**

A Gantt chart compared with a PERT diagram (network diagram) for scheduling activities.

# PERT Diagrams

- A project has a:
  - Beginning
  - Middle
  - End
- In the example:
  - , The beginning is event 10 and the end is event 50
- To find the length of the project:
  - Each path from beginning to end is identified
  - The length of each path is calculated.
- In this example:
  - Path 10–20–40–50 has a length  of 15 days (the critical path)
  - Path 10–30–40–50 has a length of 11 days

# PERT Diagrams

- The longest path is referred to as the critical path
- The critical path is determined  by calculating the longest path:
  - It is defined as the path that will cause the project to fall behind if delays (caused by individual tasks overrunning) are encountered on that path
- Note:
  - That if you are delayed (by a single day) on path 10–20–40–50
    - The project will take longer
  - If you are delayed on path 10–30–40–50
    - The project completion will not suffer (the leeway to fall behind on noncritical paths is called “float”

# PERT Diagram Example (Figure 3.17)

- Occasionally PERT diagrams need pseudo-activities
  - Referred to as dummy activities
  - To  preserve the logic of or to clarify the diagram
- Figure 3.17 shows two PERT diagrams with  dummy activities
  - Project 1 and Project 2 are quite different (The way the dummy activities are drawn makes clarifies the difference – see the following slide)
  - Project 1 takes 14 days to complete whereas Project 2 takes only 9 days
  - The dummy in  Project 1 is necessary because it indicates a crucial precedence relationship
  - The  dummy in Project 2 is not required and activity A could have been drawn from 10 to 40 eliminating event 20 completely

# PERT Diagram Dummy Activity Example (Figure 3.17)

- Project 1 and Project 2 are quite different
- The way the dummy activities are drawn makes  clarifies the difference
- In Project 1:
  - Activity C can be started only if both A and B are finished because all arrows coming into a node must be completed before leaving the node
- In Project 2:
  - Activity C requires only activity B's completion and can therefore be under way while  activity A is still being executed

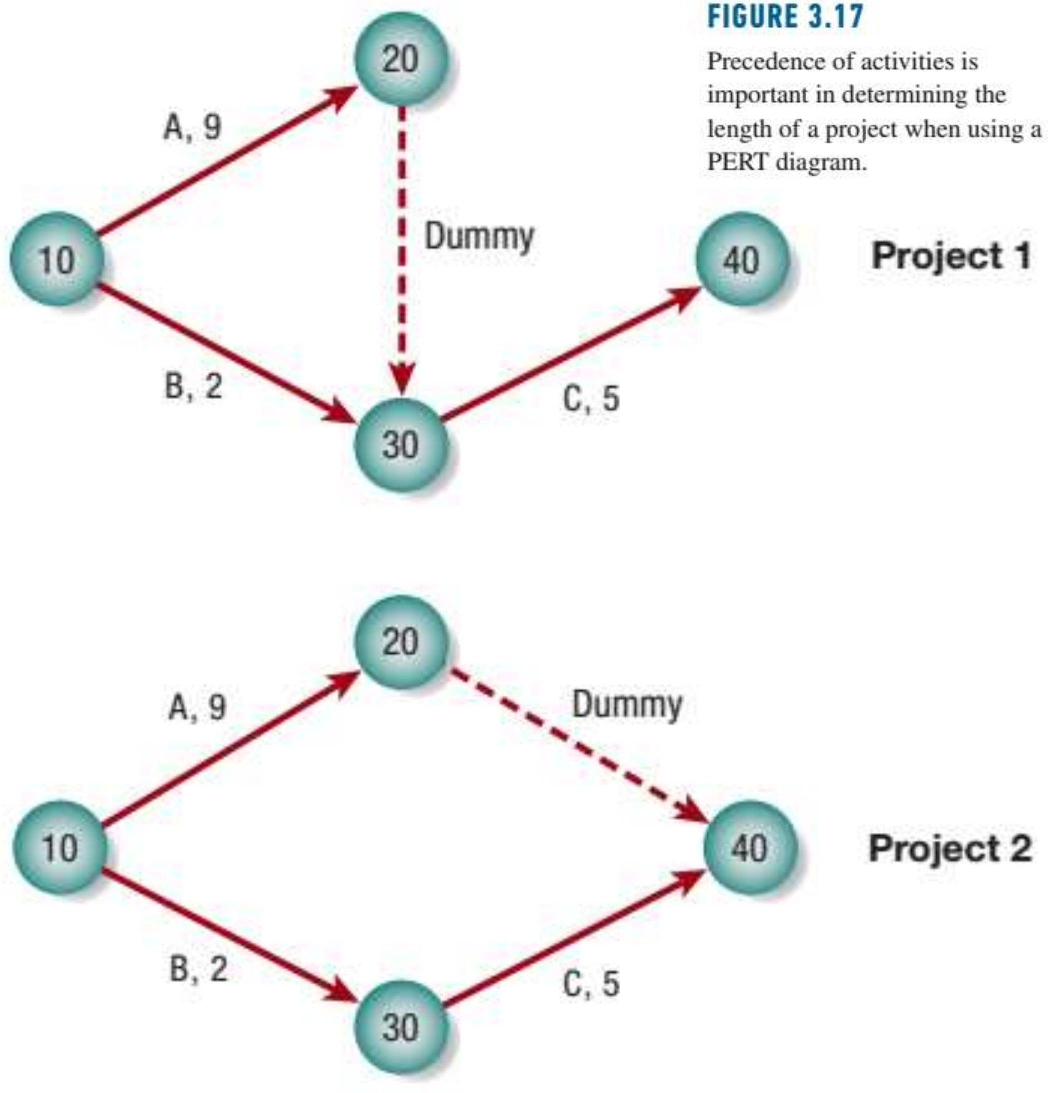


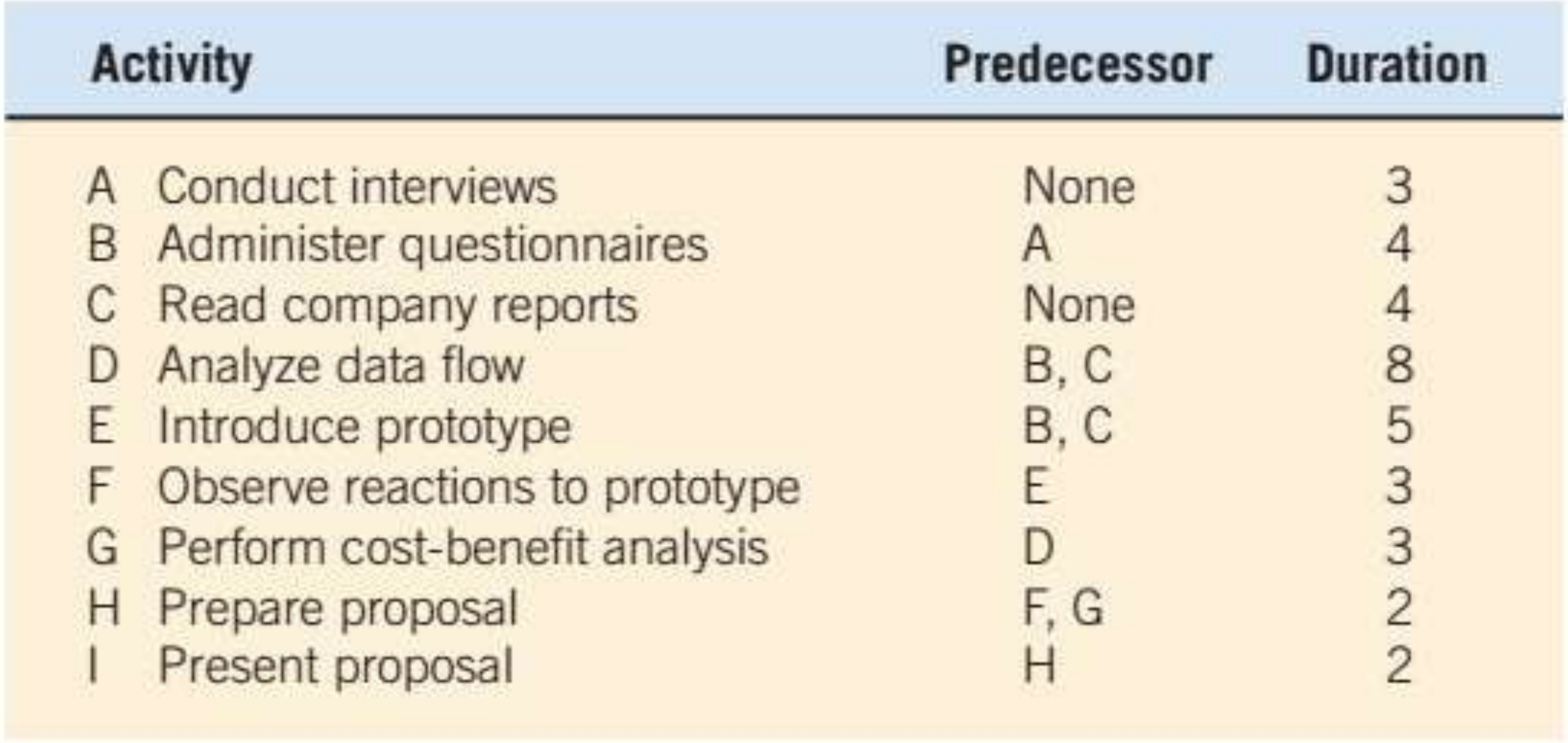
**FIGURE 3.17**

Precedence of activities is important in determining the length of a project when using a PERT diagram.

# PERT Diagram Example

- From the previous examples:
  - There are many reasons for using a PERT diagram over a Gantt chart
- Consider a scenario where a systems analyst is trying to set up a realistic schedule for the data gathering and proposal phases of the systems analysis and design life cycle
  - The systems considers the situation and lists activities that need to be accomplished
  - Such a list appears in Figure 3.18
    - Shown is the order of activities and the precedence
    - The time estimates were determined as discussed in an earlier section of this chapter.

# PERT Diagram Example



| Activity | | Predecessor | Duration |
|---|---|---|---|
| A | Conduct interviews | None | 3 |
| B | Administer questionnaires | A | 4 |
| C | Read company reports | None | 4 |
| D | Analyze data flow | B, C | 8 |
| E | Introduce prototype | B, C | 5 |
| F | Observe reactions to prototype | E | 3 |
| G | Perform cost-benefit analysis | D | 3 |
| H | Prepare proposal | F, G | 2 |
| I | Present proposal | H | 2 |

**FIGURE 3.18**

Listing activities for use in drawing a PERT diagram.

# Drawing the PERT Diagrams (1)

- In constructing the PERT diagram for this example the analyst initially considers:
  - Activities that require no predecessor activities
    - Case A (conduct interviews) and case C (read company reports)
- In the example in Figure 3.19:
  - The analyst chose to number the  nodes 10, 20, 30 (etc) and drew two arrows out of the beginning node 10
  - These arrows represent activities A and C and are labelled as such
  - Nodes numbered 20 and 30 are drawn at  the end of these respective arrows

# Drawing the PERT Diagrams (2)

- The next step is to look for any activity that requires only A as a predecessor
  - Task B (administer questionnaires) is the only one so it can be represented  by an arrow drawn from node 20 to node 30
  - Because activities D (analyse data flow) and E (introduce prototype) require both activities B and C to be completed before they are started
    - Arrows labelled D and E are drawn from node 30
      - The event that recognizes the completion of both B and C
- This process is continued until the entire PERT diagram is completed
  - Notice that the entire project ends at an event  called node 80

# Drawing a PERT Diagram (Figure 3.19)

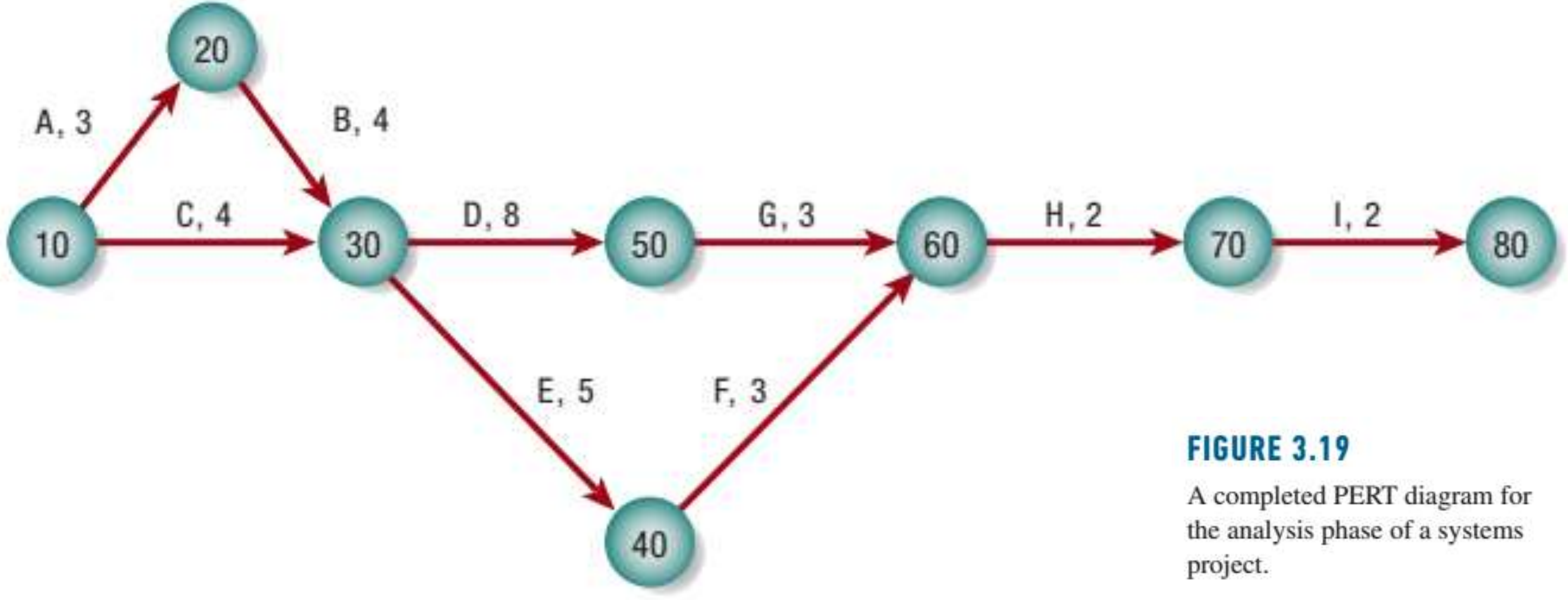


**FIGURE 3.19**

A completed PERT diagram for the analysis phase of a systems project.

# Identifying the Critical Path

- Once the PERT diagram is drawn:
  - It is possible to identify the  critical path by calculating the sum of the activity times on each path and choosing the  longest path
  - In this example, there are four paths:
    - 10–20–30–50–60–70–80
    - 10–20–30–40–  60–70–80
    - 10–30–50–60–70–80
    - 10–30–40–60–70–80
  - Clearly the longest path is 10–20–30– 50–60–70–80
    - The period is 22 days
- It is essential that the systems analyst:
  - Monitor the activities on the critical path
  - To keep the entire project on time (or possible aim for an earlier completion date)

# Controlling Projects

# Project Management

- Regardless of how well a project is planned and scheduled
    - A project is a dynamic process where there will inevitably be unforeseen events and problems
- In this section we introduce:
    - How to estimate cost
    - How to prepare a budget
    - How to carry out a risk analysis and try predict and prepare for risk
    - How to address delays (make up time) using expediting
    - How to control costs when:
        - The budget needs revising
        - The  team falls behind or experiences cost overruns.

# Estimating Costs and Preparing the Budget

# Estimating Costs and Preparing the Budget

- It is also essential that the cost of a project be managed
- Once a work breakdown structure is created and a schedule is planned the analyst needs to:
    1. Estimate costs for each activity in the work breakdown structure
    2. Prepare a budget for the project and have it approved by the organization or client (who is funding the project)
    3. Manage and control the costs throughout the project
- We have discussed cost estimates for equipment and off-the-shelf software and resources we need to complete the project
    - We are nor concerned with other types of resources needed to complete each task in the work breakdown structure
- The main resources for this part of the project are the time of the team members and the type of special equipment and tools needed to finish each of the activities

# Estimating Costs and Preparing the Budget

- There are two types of cost estimation:
  - An *estimate*
  - A *quotation* (or tender for a project)
- In this section we are considering estimates
- Many approaches to cost estimation are available to a systems analyst
- The approaches are similar to  the estimates used for time estimates and include the following:
  1. Basing estimates on similar projects
     1. The top-down approach
     2. Building bottom-up estimates
     3. Using parametric modelling

# Top-Down Approach

- Previous experience with estimating costs is very important in a project manager (or systems analyst)
  - Experience is particularly important if the project you are estimating is similar to previous project(s)
- For example:
  - If you have been responsible for the development of a website with similar features (to the current project)
  - It is possible to estimate costs with a high degree of reliability
  - Customization is possible:
    - The new website may feature a different number of products for sale but it is possible to adjust the costs accordingly

# Bottom-Up-Approach

- Often an analyst is faced with a unique project:
  - For example the project may require a different programming language
- In such a use-case:
  - A bottom-up approach may be recommended
- A systems analyst can take the work breakdown structure and ask each responsible project team member to estimate the cost involved for completing the activity that is his or her responsibility
  - This method results in estimates that may be *good* or *bad* depending on the abilities of each team member (the analyst will ultimately be responsible!)
  - The analyst, acting as a project manager:
    - Must review each of these estimates and arrive at a cost estimation that satisfies the team and the client
- An issue with the bottom-up method is the time it takes to make each estimate

# Parametric Modelling

- This method involves making estimates for each of the many factors (or  parameters) that combine to create a project
- For example:
  - It can be estimated that it will cost $75 per line of  code and $80 per hour for the computer programmers needed
  - An estimate of the number of lines of program code can be made
  - These parameters will enable an overall estimate (to complete the project) to be arrived at
- Special parametric modeling software (such as  COCOMO II, introduced previously) is often useful when modelling the project

# Why Cost Estimates Fail (2)

- In practice most analyses use a combination of all three of the methods
- There are two main reasons cost estimates fail:
  1. An analyst may be overly optimistic
     - He/she believes in his or her team members
     - The estimate will generally show the team completing the project quickly and without errors
     - An analyst is may underestimate the lines of code and effort in general
  2. An analyst may want to move past the cost estimation process and get on with preparing and presenting the budget and beginning the actual work on the project:
     1. In such a case there is a temptation to “cut corners”, “make optimistic and unfounded assumptions”, and spend less time than is needed to properly prepare the estimates
     2. To avoid this issue: the  analyst should be as accurate as possible even though the estimates will be revised as the project continues (which is generally the case with all projects).

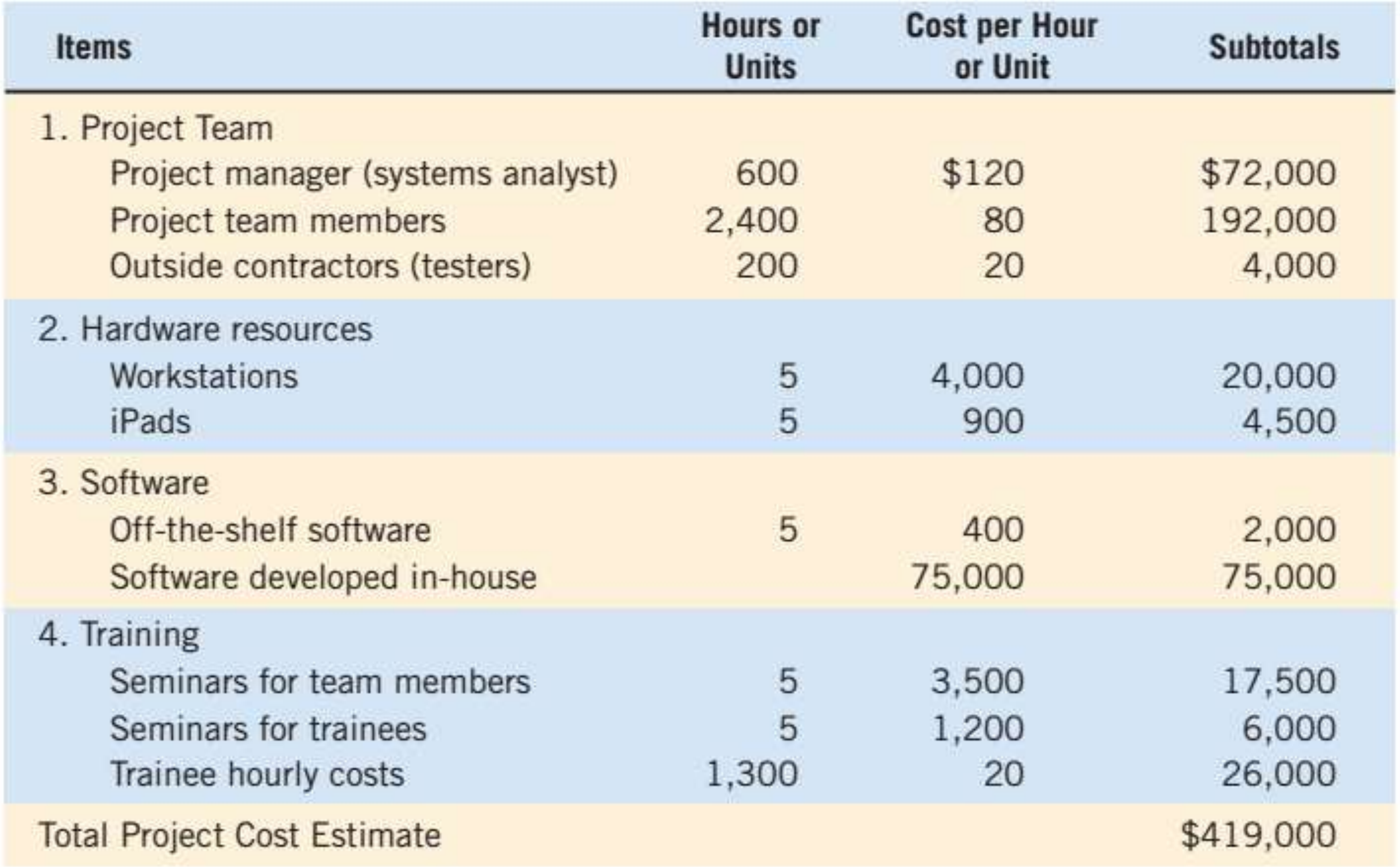
# Why Cost Estimates Fail (2)

- In practice most analyses use a combination of all three of the methods
- There are two main reasons cost estimates fail:
  1. An analyst may be overly optimistic
     - He/she believes in his or her team members
     - The estimate will generally show the team completing the project quickly and without errors
     - An analyst is may underestimate the lines of code and effort in general
  2. An analyst may want to move past the cost estimation process and get on with preparing and presenting the budget and beginning the actual work on the project:
     1. In such a case there is a temptation to “cut corners”, “make optimistic and unfounded assumptions”, and spend less time than is needed to properly prepare the estimates
     2. To avoid this issue: the  analyst should be as accurate as possible even though the estimates will be revised as the project continues (which is generally the case with all projects).

# Preparing a Budget

- All project require a budget (which is an estimation of the potential cost of completing a project) which will certainly change if the project proceeds
- A budget is a critical deliverable:
    - Every client wants to see a detailed budget very early in the process
    - A systems analyst needs to conform to the standard process the client uses to develop a budget
    - Most of the time, the client will have standard forms it uses for this purpose
    - A sample budget is shown in Figure 3.20 (see the next slide)

# ?



| Items | Hours or Units | Cost per Hour or Unit | Subtotals |
|---|---|---|---|
| 1. Project Team | | | |
| Project manager (systems analyst) | 600 | $120 | $72,000 |
| Project team members | 2,400 | 80 | 192,000 |
| Outside contractors (testers) | 200 | 20 | 4,000 |
| 2. Hardware resources | | | |
| Workstations | 5 | 4,000 | 20,000 |
| iPads | 5 | 900 | 4,500 |
| 3. Software | | | |
| Off-the-shelf software | 5 | 400 | 2,000 |
| Software developed in-house | | 75,000 | 75,000 |
| 4. Training | | | |
| Seminars for team members | 5 | 3,500 | 17,500 |
| Seminars for trainees | 5 | 1,200 | 6,000 |
| Trainee hourly costs | 1,300 | 20 | 26,000 |
| Total Project Cost Estimate | | | $419,000 |

**FIGURE 3.20**

Part of a sample budget for a software development project, showing hardware, software, and personnel costs.

# Preparing a Budget

- The budget contains estimates for:
  - The hours worked
  - The rates of pay for each of the internal or outsourced workers
  - Hardware costs
  - Software costs
  - The quantities for each type of equipment
  - Training and support costs
  - Events (seminars) to implement training
    - Note the different levels of training for different roles
- Remember: the budget is:
  - Only an initial estimation of the final cost
  - Based on assumptions which must be set out in a project proposal

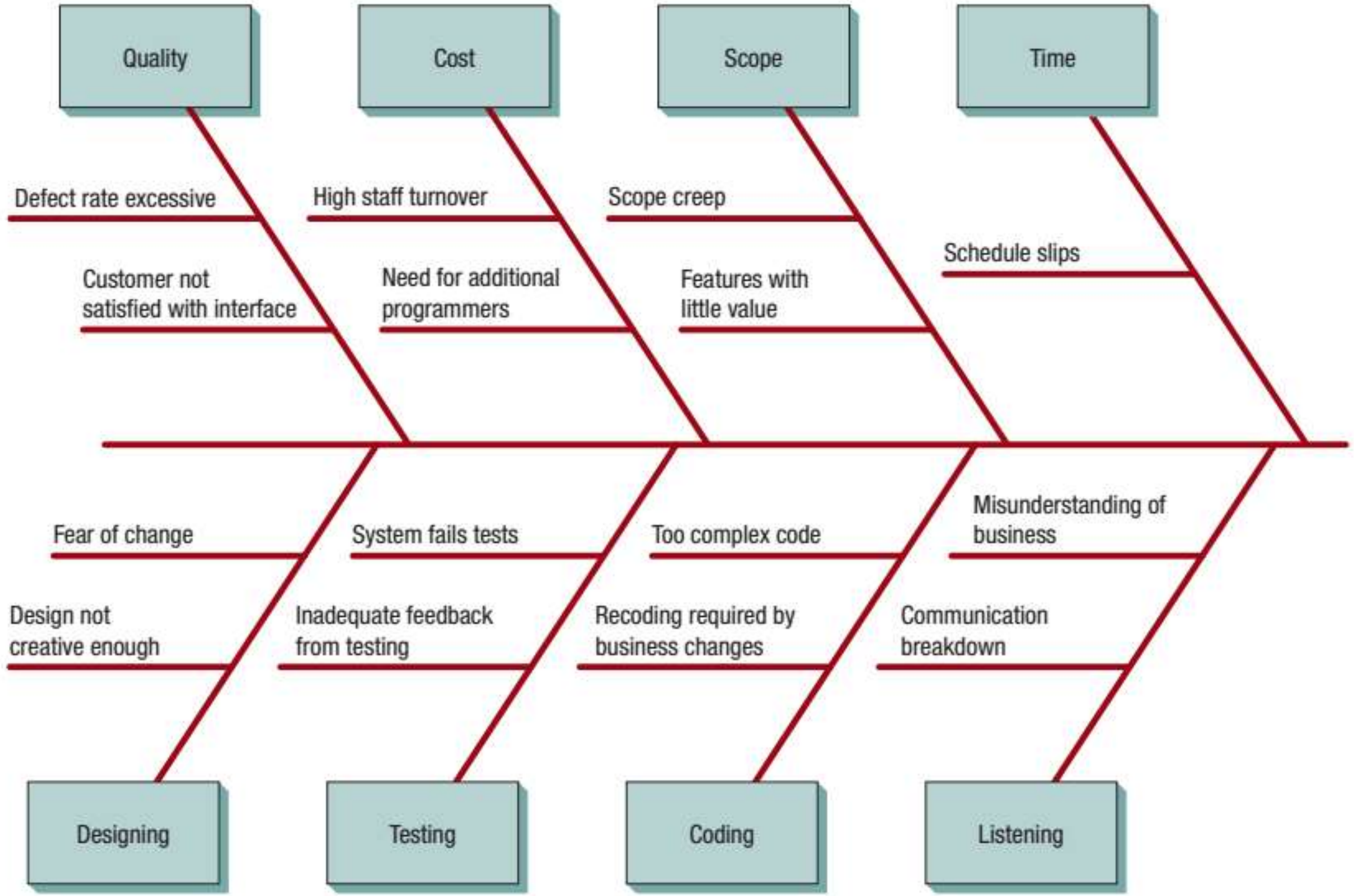
# Managing Risk

# Managing Risk

- Managing risk begins at the pre-project (early) stage where initial discussions are undertaken to evaluate the project feasibility
- An analyst will use a number of techniques to reduce the potential risk of project failure including:
    - Feasibility studies to identify projects that have a high probability of failure
    - Identify clearly the metrics which will define success (or) failure
    - Training and experience will improve the ability to judge the worthiness of projects and the motivations that prompt others to request projects
- In-house systems analysis teams must be aware of:
    - The organisational culture and politics
    - The financial and business competitive situation(s)
- It is important to note that systems projects can (and do) have serious problems

# Agile Methods

- Projects developed using agile methods are not immune to risk and the related potential problems
- To illustrate what  can go wrong in a project:
  - An analysis may include a fishbone diagram (also called  a cause-and-effect diagram, or an Ishikawa diagram)
  - From Figure 3.21: it can be seen why it is called a fishbone diagram
    - It resembles the skeleton of a fish
  - When using a fishbone diagram:
    - All the foreseen problems that can occur are systematically listed
    - Note: there will be unforeseen and unexpected events and problems that cannot be anticipated – this is the reason for contingencies (time and cost) is a project plan



Quality

Defect rate excessive

Customer not satisfied with interface

Cost

High staff turnover

Need for additional programmers

Scope

Scope creep

Features with little value

Time

Schedule slips

Fear of change

Design not creative enough

Designing

System fails tests

Inadequate feedback from testing

Testing

Too complex code

Recoding required by business changes

Coding

Misunderstanding of business

Communication breakdown

Listening

**FIGURE 3.21**

A fishbone diagram may be used to identify all the things that can go wrong in developing a system.

# ?

- In the case of the agile approach, it is useful to organize a fishbone diagram by listing all  the resource control variables on the top and all the activities on the bottom. Some problems,  such as schedule slips, might be obvious, but others, such as scope creep (the desire to add  features after the analyst hears new stories) or developing features with little value, are not as  obvious.  You can learn from the wisdom gained by people involved in earlier project failures.

# ?

- When asked to reflect on why projects had failed, professional programmers cited management setting impossible or unrealistic dates for completion, belief in the myth that simply adding more people to a project would expedite it (even though the original target date on the project was unrealistic), and management behaving unreasonably by forbidding the team to seek professional expertise from outside the group to help solve specific problems.

# ?

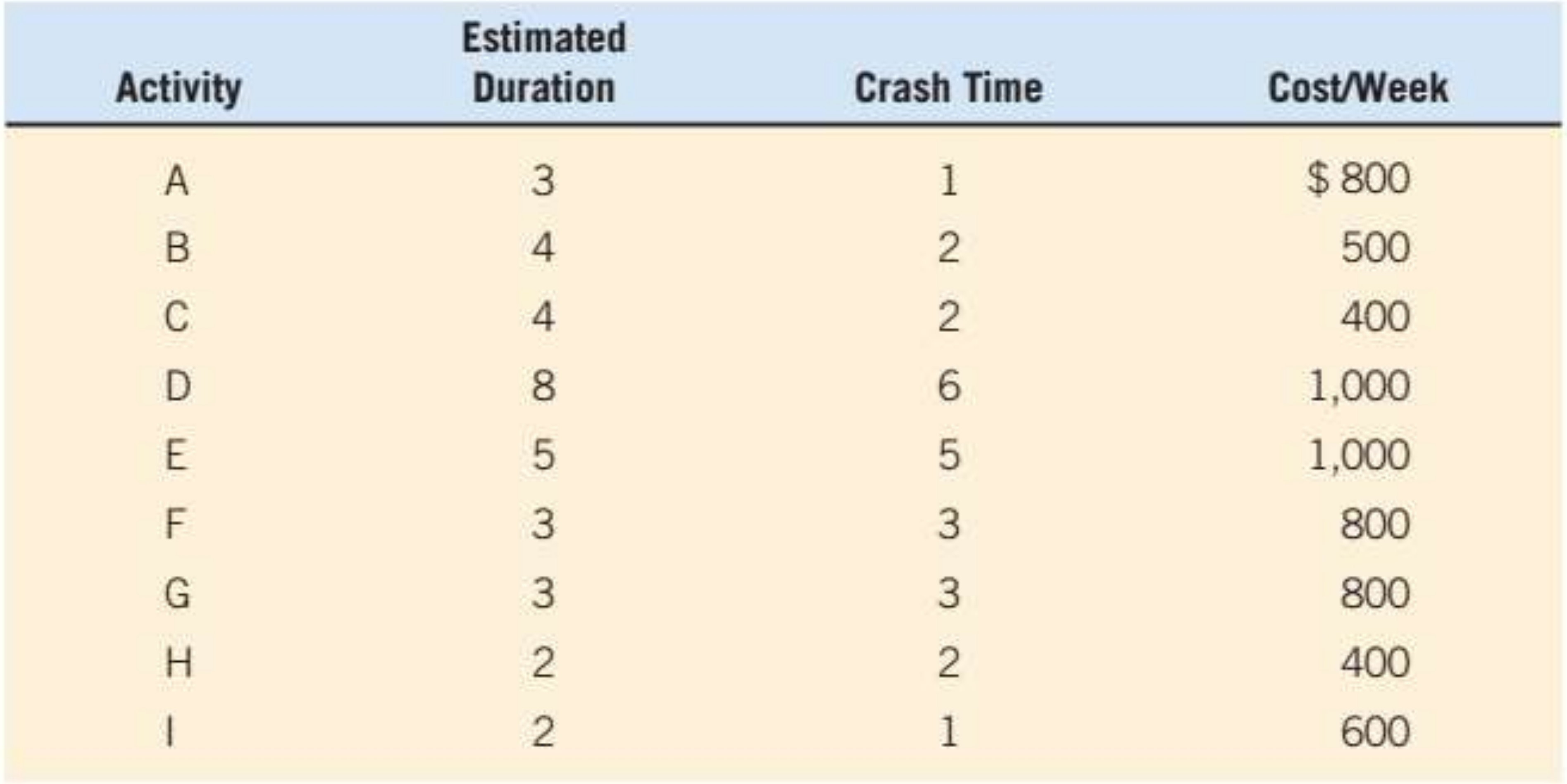
- Remember that you are not alone in the decision to begin a project. Although apprised of your team’s recommendations, management will have the final say about whether a proposed project is worthy of further study (that is, further investment of resources). The decision process of your team must be open and stand up to scrutiny from those outside it. The team members should consider that their reputation and standing in the organization are inseparable from the projects they accept.

# Managing Time Using Expediting

# Expediting

- Speeding up a process is called expediting
  - Completing a project early may result in a bonus for a contractor
    - Late completion may result in a penalty (liquidated damages)
    - Project team resources and team members can be used for another project if you finish the current project before the due date
- Figure 3.22:
  - Lists each activity in the project with the original estimated time required to complete each of these activities
  - The column, labelled “crash time” refers to the absolute minimum time in which an activity can be completed if additional money is allocated to that activity
  - The final column contains the cost of reducing the activity duration by one week.

# Expediting Example (Figure 3.22)



| Activity | Estimated Duration | Crash Time | Cost/Week |
|---|---|---|---|
| A | 3 | 1 | $ 800 |
| B | 4 | 2 | 500 |
| C | 4 | 2 | 400 |
| D | 8 | 6 | 1,000 |
| E | 5 | 5 | 1,000 |
| F | 3 | 3 | 800 |
| G | 3 | 3 | 800 |
| H | 2 | 2 | 400 |
| I | 2 | 1 | 600 |

**FIGURE 3.22**

Estimated duration, crash time, and the cost of expediting project activities.

# Expediting Example

- Expediting can help reduce the time it takes to complete an entire project:
  - However: the expedited activities have to be on the critical path for expediting to make a time difference
  - Which activity is expedited on the critical path
    - Depends on the cost-benefit analysis
    - Assuming the activity is not already at its “crash time”
- The maximum number of weeks each activity can be reduced is the difference between the expected time and its crash time
- For example:
  - Activity B (administering questionnaires) could be reduced from four weeks to two weeks at a cost of $500 per week
    - But it cannot be reduced to less than two weeks
  - Activity H (preparing the proposal) cannot be reduced because it is already at its “crash time”

# Expediting Analysis

- The expediting analysis for this example is provided in Figure 3.23
- The expediting process  takes place one step at a time until it is impossible to expedite any further
- The columns in the table include:
  - Eligible activities (tasks that are on the critical path and can be reduced by expediting)
  - The  activity chosen (because it is the cheapest alternative)
  - The time it currently takes to complete each of the paths
  - The cost of expediting the chosen activity, and finally the cumulative cost

# Expediting Analysis

- In the first step:
  - The critical path is 10–20–30–50–60–70–80
    - So the eligible activities are  A, B, D, and I
  - Activities G and H are also on the critical path
    - But they are already at their  crash times and are consequently ineligible for expediting
  - The cheapest alternative is to expedite activity B by one day
    - Reducing the first path from 22 to 21 weeks
    - Reducing the second path from  19 to 18 weeks
    - The third and fourth paths are not affected by the reduction because activity B is not on either of those paths

# The Critical Path

- The critical path (and therefore the entire project):
  - Is reduced from 22 to 21 weeks (circled on  the table)
- We can repeat this reduction and reduce the project time by another week:
  - When activity B reaches its crash time another activity must be chosen
  - Row 3 in the table shows that activities A, D, and I are eligible, and activity I is the cheapest alternative
  - Reducing  activity I reduces not only the critical path but all paths because it is common to all of them
- In the fourth step:
  - Activity A is chosen reducing paths 1 and 2
  - The result is now two critical paths
    - This implies that any reduction of the project time will take place only if both of the critical paths are reduced at the same time

# ?

- We can shorten both paths in the fifth and sixth steps by selecting:
  - A combination of activities A and C (one activity from each of the critical paths) (or)
  - Activity D (an activity common to both critical paths)
- Reducing activity D by 2 days:
  - shortens the paths to 16, 16, 16, and  15 days respectively
  - There are three critical paths
- Finally:
  - When activity D reaches its crash time
  - The only available choice is a combination of activities A and C

# Expediting Example

- The minimum project time is therefore 15 weeks:
- The minimum time is obtained by reducing:
    - Activity A by 2 days
    - Activity B by 2 days
    - Activity C by 1 day
    - Activity D by 2 days
    - Activity  I by one day
- The total cost is $5,600
- This example describes all-out expediting to obtain the minimum project time at any cost

# Expediting Constraints

- However: a systems analyst may be faced with a budget
  - In our example a budget of $4,000 would result in expediting up to and including step 5
  - The project would be shortened from 22 to 17 weeks, at a cost of $3,400
- Another possible criterion would be the net amount that could be saved if the project were shortened:
  - Suppose that in the above example the analyst would save $750 per week (mostly consisting of opportunities available for the project team to begin new projects sooner)
  - In this case: expediting would not take place until Step 3 because the incremental cost of Step 4 ($800 for expending activity A) would exceed the $750 saved

# Expediting

- Expediting can make or break a successful project
- The analyst (project manager) must ensure that project costs are constantly monitored and managed effectively
- The systems analyst will ideally remain “on top” of the process by managing the project throughout the entire development process

# Controlling Costs Using Earned Value Management

# Earned Value Management (EVM)

- It is an essential task for a project manager (and the team) to monitor a project to identify:
  - The progress of the project and delays (measured in terms of time)
  - The costs of the project (measured against the budgets)
  - The scope and changes in the project requirements
- Earned Value Management (EVM) is a technique used to help determine progress (or setbacks) on a project which addresses:
  - Project cost
  - Project schedule
  - Performance of the project team
- The following slides introduce the principal components of EVM
  - For full details including the calculations see Chapter #3 (pp. 119-121)

# Earned Value Management

- Once a budget has been approved budget:
  - Working within the budget is essential to manage costs
  - This requires continuous monitoring of costs and updating the budgets
  - All the stakeholders need to be informed of the developing costs and budgets
- Changes will always pe a part of a project:
  - The client may request new features
  - New technologies may be introduced that will change the way the system is developed
  - Such changes must be reflected in revised and updated budgets
- One tool available to systems analysts is earned value management (EVM)
  - It is a technique used to help determine progress (or setbacks) on a project and icludes:
    - Cost / schedule / performance (of the project team)

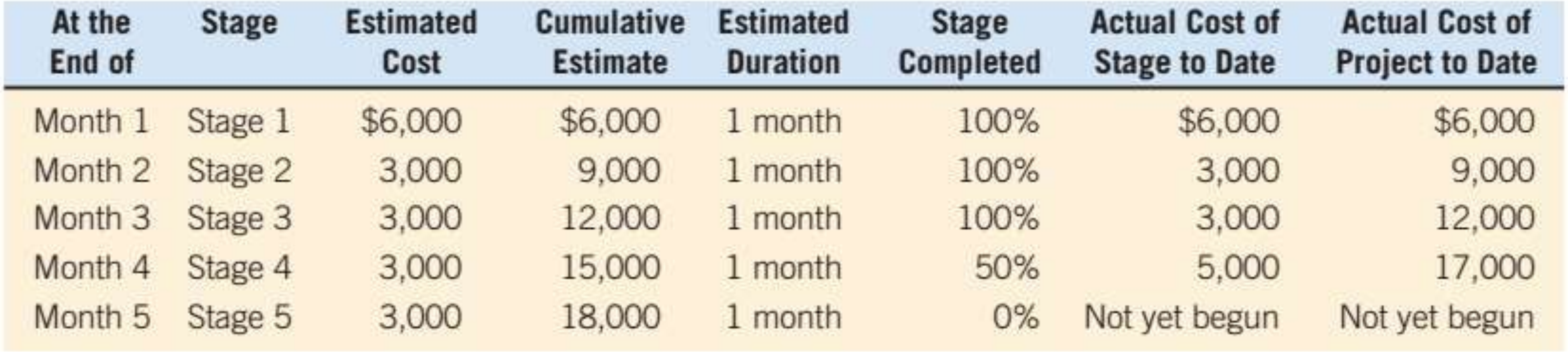
# ?

- There are the four key measures in EVM:
  - Budget at completion (BAC):
    - The total budget for the project
    - In calculating the performance measures for a task it is the total budget for that task
  - Planned value (PV):
    - Is the value of the work that is to be completed on the project (or, alternatively, the work completed on any task)
    - The value of the work completed reflects the effort and finance (PV is the budgeted cost of work scheduled)
  - Actual Cost (AC):
    - Is the total cost (direct and indirect) incurred in completing the work on the project (or, alternatively, a task) up to that particular point in time
    - Alternatively: AC refers to the actual cost of work performed to date
  - Earned value (EV):
    - Is an estimate of the work performed thus far
    - EV refers only to the work completed to date

# ?

- There are the four key measures in EVM:
  - Budget at completion (BAC):
    - The total budget for the project
    - In calculating the performance measures for a task it is the total budget for that task
  - Planned value (PV):
    - Is the value of the work that is to be completed on the project (or, alternatively, the work completed on any task)
    - The value of the work completed reflects the effort and finance (PV is the budgeted cost of work scheduled)
  - Actual Cost (AC):
    - Is the total cost (direct and indirect) incurred in completing the work on the project (or, alternatively, a task) up to that particular point in time
    - Alternatively: AC refers to the actual cost of work performed to date
  - Earned value (EV):
    - Is an estimate of the work performed thus far
    - EV refers only to the work completed to date

# Expected Costs



| At the End of | Stage | Estimated Cost | Cumulative Estimate | Estimated Duration | Stage Completed | Actual Cost of Stage to Date | Actual Cost of Project to Date |
|---|---|---|---|---|---|---|---|
| Month 1 | Stage 1 | $6,000 | $6,000 | 1 month | 100% | $6,000 | $6,000 |
| Month 2 | Stage 2 | 3,000 | 9,000 | 1 month | 100% | 3,000 | 9,000 |
| Month 3 | Stage 3 | 3,000 | 12,000 | 1 month | 100% | 3,000 | 12,000 |
| Month 4 | Stage 4 | 3,000 | 15,000 | 1 month | 50% | 5,000 | 17,000 |
| Month 5 | Stage 5 | 3,000 | 18,000 | 1 month | 0% | Not yet begun | Not yet begun |

**FIGURE 3.24**

The anticipated cost of a website development project over a five-month period.

# EV Values and the Actual Development

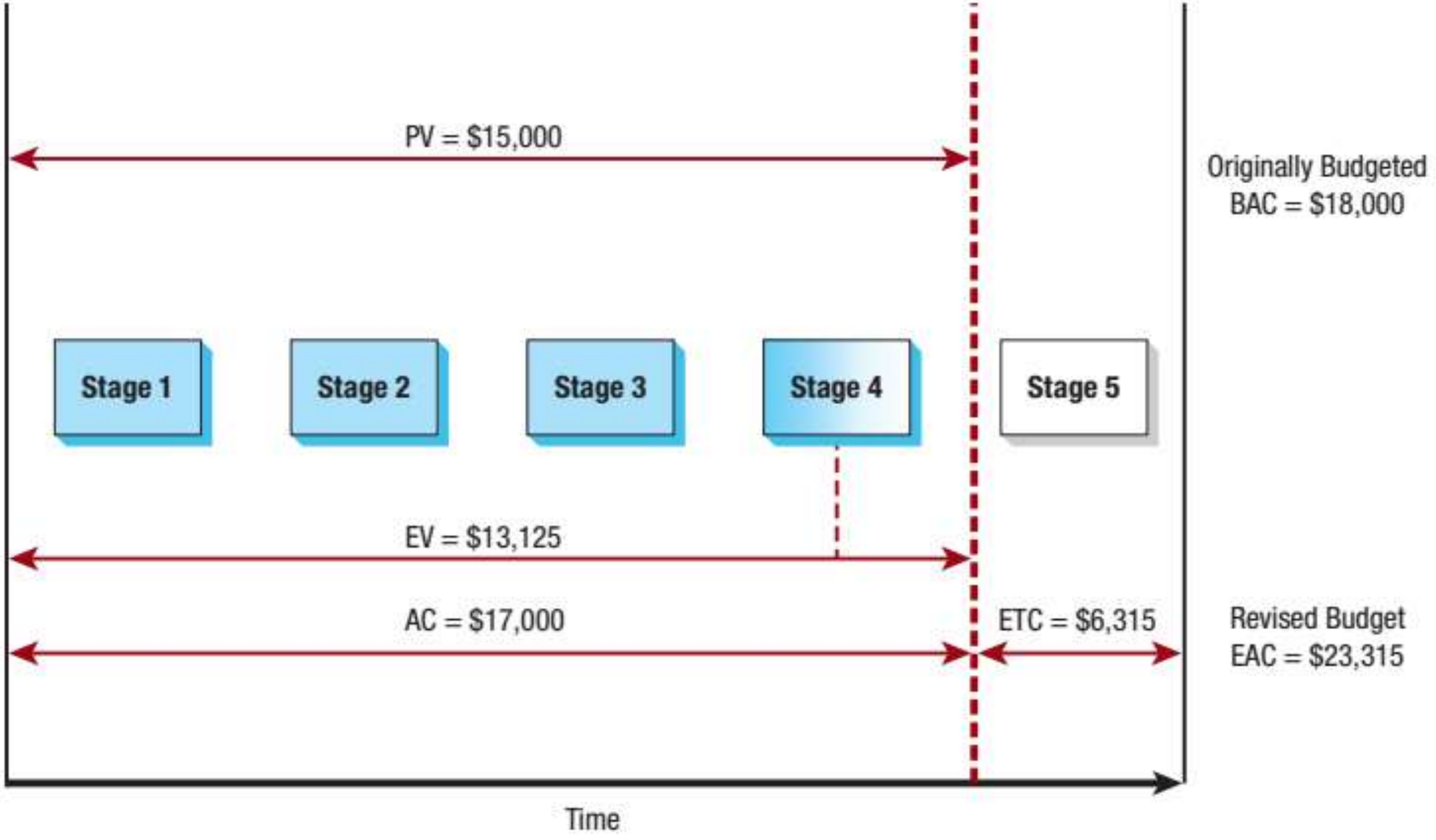


**FIGURE 3.25**

How the earned value management values relate to the planned and actual development of the website.

# Project Management (3)

# Managing a Project Team

# Managing a Project Team

- Along with managing time and resources systems analysts must also:
  - Manage people
  - Communicate with team members (selected for their competence and compatibility)
  - Set project goals and motivate teams to achieve the goals
- The stages in team management are:
  - Assemble a team
  - Set team communication strategies
  - Set project productivity goals
  - Set a policy for team member motivation

# Assembling a Team

# Assemble a Team

- Project management is a collaborative team activity
  - Teams must be built with members who have the skills required to complete the project
  - A project manager has an opportunity to create a dream team of skilled and experienced people
  - Project managers need to look for team members who share their values of teamwork and are guided by the desire to deliver a high-quality system on time and on budget.
  - Other desirable team member characteristics include:
    - A good work ethic / honesty / competence / readiness to  take on leadership based on expertise / motivation / enthusiasm for the project / trust of teammates

# Assemble a Team

- Project manage need to know about business principles:
  - It is useful to have  at least one other person on the team who understands how a business operates (Such a person should be a specialist in the area of the system being developed)
  - For example:
    - When developing an ecommerce site
    - A team can enlist the help of someone in marketing
    - Those developing an inventory system can ask a person versed in production and operations  to provide expertise
  - A team ideally should include at least two systems analysts:
    - They can work together
    - Check each other's work
    - Share the workloads

# Assemble a Team

- Software projects need to have people with programming skills
  - While programming is important knowing how to conduct walkthroughs / reviews / testing, and documenting systems is important
- Some people are good at seeing the big picture while others perform well when tasks are broken down into smaller ones for them
  - Teams need both types of individuals.
- Beyond the basics just described:
  - A project manager should look for people with both experience and enthusiasm
  - Experience is especially important when trying to estimate the time required for completing a project
- Experience in computer programming can mean code is developed five times faster than if it is developed by an inexperienced team

# Assemble a Team

- A usability expert is also a useful addition to the team
- The team must be motivated
  - One way to keep a team motivated throughout the  project is to select good people at the outset
  - Look for enthusiasm, imagination, and an ability to communicate with different kinds of people.
- These basic attributes hold the potential for success
  - Additionally: presentation of proposals needs good writers and articulate speakers who can work directly with customers
- Trust is an important part of a team:
  - All members of the project team need to act responsibly and agree to complete their part of the project
  - People may have different work  styles, but they all need to agree to work together toward a common goal

# Communication Strategies for Managing Teams

# Teams and Leadership

- Teams have both *internal* (staff)and *external* (specialists and consultants)
  - Teams have their own personalities as a result of combining team to create new network of interactions
- A way to organize thinking about teams is to visualize them as always seeking a balance between accomplishing the work at hand and maintaining relationships among team members
- In fact, teams often have two leaders
  - Usually one person (ideally the project manager) will emerge who leads members to accomplish tasks
  - Another person will emerge who is concerned with the social relationships among group members
  - Both are necessary for the team
  - These individuals have been labelled by other researchers as task leader and socio-emotional leader respectively
  - Every team is subject to tensions that are an outgrowth of seeking a balance between accomplishing tasks and maintaining relationships among team members.

# Team Effectiveness

- For a team to continue its effectiveness:
  - Tensions must be continually resolved
  - Minimizing or ignoring tensions will lead to ineffectiveness and eventual disintegration of the team
  - Much of the tension release necessary can be gained through skillful use of feedback by all team members
  - All members, however, need to agree that the way they interact (that is, the process) is important enough to merit some time
  - Productivity goals for processes are discussed in a later section.
- Securing agreement on appropriate member interaction involves creating explicit and implicit team norms (collective expectations, values, and ways of behaving) that guide members in their relationships
- A team's norms belong to it and will not necessarily transfer from one team to another
  - These norms change over time and are better thought of as a team process of interaction than a product.

# Behavioral Norms

- Norms can be:
  - Functional (or)
  - Dysfunctional
- Because a particular behaviour is a norm for a team does not mean it is helping the team to achieve its goals
  - For example:
    - An expectation that junior team members should do all project scheduling may be a team norm
    - By adhering to this norm – the team puts pressure on new members and may fail to leverage the experience in the team
    - It is a norm that can waste team resources.
- Team members must:
  - Make norms explicit
  - Periodically assess whether norms are functional or dysfunctional (to helping achieve goals) The overriding expectation for your team must be that change is the norm
  - Ask whether team norms are helping or hindering the team's progress

# Setting Project Productivity Goals

# Progress Management

- As team leader:
  - You will get experience in projecting what the team can achieve in a specific amount of time
  - Using the hints on methods for estimating time required and coupling them with experience will enable the team to set worthwhile productivity  goals
- Systems analysts are accustomed to thinking about productivity goals for employees with tangible outputs such as:
  - The number of pairs of blue jeans sewn per hour
  - The  number of entries keyed in per minute
  - The number of items scanned per second

# Progress Management

- As  manufacturing productivity rises it clear that managerial productivity must keep pace
  - It is with this aim in mind that productivity goals for the systems analysis team are set
  - A team needs to formulate and agree to goals
  - These goals should be based on:
    - Team  members' expertise and former performance
    - The nature of the specific project
- Goals will vary for each project undertaken because:
  - Sometimes a project will involve an entire (new) system
  - Other projects might involve limited modifications to a portion of an  existing system

# Motivating Project Team Members

# Motivation and Cognitive Processes

- While motivation is an extremely complex topic a brief overview is appropriate at this point
- People join organizations to provide for some of their  basic needs such as food / clothing / shelter
- All humans also have higher-level (cognitive) needs including affiliation / control / independence / creativity
- People are motivated to fulfil unmet needs on several levels
  - Team members can be motivated (at least partially) through participation in goal setting (it can create ownership of a goal)
  - Setting a challenging (but achievable) goal and then periodically measuring performance against the goal seems to work in motivating people

# Motivation and Cognitive Processes

- Goals can promote achievement
  - Part of the reason goal setting motivates people is that team members know prior to  any performance review exactly what is expected of them
  - The success of goal setting (in motivation) can also be ascribed to affording each team member some autonomy in achieving the goals
- Although a goal is predetermined:
  - The means to achieve it may not be
  - Team members are free to use their own expertise and experience to meet their goals

# Setting Goals

- Setting goals can motivate team members by clarifying for them and others what must be done to get results
  - Goals define the level of achievement that is expected of team members
- This use of goals:
  - Simplifies the working atmosphere
  - Provides encouragement with the possibility that what is expected can indeed be achieved

# Managing Ecommerce Projects

# E-Commerce

- Many of the approaches and techniques discussed so far are transferable to e-commerce project management
  - While there are many similarities
  - There are also many differences
  - One difference is that the data used by ecommerce systems are widely dispersed and scattered over an organization.
  - Therefore, you are not just managing data in a self-contained department or even in one solitary unit.
- Organization culture (politics) may be an issue:
  - Departments often feel protective of the data they generate and do not understand the need to share them across the organization
  - You are not just managing data in a self-contained department or even in one solitary department

# E-Commerce

- E-commerce project teams typically need more staff:
  - Drawn from across the organization
  - With a variety of skills including developers /. Consultants / database experts / system integrators.
- Stable project groups that exist within a cohesive information systems (IS) group or systems development team are the exception rather than the rule
- Because so much help may be required initially:
  - E-commerce project managers need to build partnerships externally and internally well ahead of the implementation
  - Sharing talent across projects to defray costs of ecommerce implementations
  - Recruit the required number of people with the necessary expertise
- The potential for organization culture (politics) to drive be disruptive (to the team members) is very real

# Cultural Issues

- To prevent politics from sabotaging a project an e-commerce project manager must:
  - Be aware of the integration of the e-commerce system with the organization's internal systems
  - Emphasize the organizational aspect embedded in the ecommerce project
- As one ecommerce project manager told us:
  - "Designing the front end [what  the consumer sees] is the easy part of all this. The real challenge comes from integrating  ecommerce strategically into all of the organization's systems."
- A difference between traditional project management and ecommerce project management is security:
  - The e-commerce system will be linking with the outside world via the Internet
  - Security is of the utmost importance
  - Developing and implementing a security plan is a project in itself

# Creating a Project Charter

# The Planning Process

- Part of the planning process is to agree on what will be done and at what time
- Analysts who are external consultants (as well as those who are organization members) need to specify what they will eventually deliver and when they will deliver it
- The entire team as well as the client need to be on board
- We have introduced techniques to:
  a) Estimate the delivery date for the completed system
  b) How to identify organization goals
  c) Assess the feasibility of the proposed system

# The Project Charter (1)

- The project charter is a written narrative that clarifies the following questions:
  1. What does the user expect of the project (that is, what are the objectives)?
  2. What will the  system do to meet the needs (achieve the objectives)?
  3. What is the scope (or what are the boundaries) of the project? (What does the user consider  to be beyond the project’s reach?)
  4. What analysis methods will the analyst use to interact with users in gathering data and  developing and testing the system?
  5. Who are the key participants? How much time are users willing and able to commit to  participating?

# The Project Charter (2)

- The project charter is a written narrative that clarifies the following questions:
  1. What are the project deliverables? (What new or updated software, hardware, procedures,  and documentation do the users expect to have available for interaction when the project  is done?)
  2. Who will evaluate the system, and how will they evaluate it? What are the steps in the  assessment process? How will the results be communicated, and to whom?
  3. What is the estimated project timeline? How often will analysts report project  milestones?
  4. Who will train the users?
  5. Who will maintain the system?

# The Project Charter (3)

- The project charter describes in a written document:
  - The expected results of the systems project (deliverables)
  - The projected project time frame for delivery
- The project charter is:
  - A specification for the team to refer to in the event of any queries or disputes relating to the project
  - A contract between the  chief analyst (or project manager) and the analysis team
  - The basis for the contract with the commissioning organization (the users requesting the new system)

# The System Proposal

# ?

- §The project charter serves the purpose of identifying objectives, determining scope, and assigning responsibilities, but an analyst still needs to prepare a systems proposal that includes much  of the detail about system needs, options, and recommendations.

# What to Include in a Systems Proposal

# The Project proposal

- A written systems proposal has 10 main sections
  - Each section has a particular function and eventually the proposal should be arranged in the following order:
    1. Cover letter
    2. Title page of project
    3. Table of contents
    4. Executive summary (including recommendations)
    5. Outline of systems study with appropriate documentation
    6. Detailed results of the systems study
    7. Systems alternatives (three or four possible solutions)
    8. Systems analysts' recommendations
    9. Conclusions and proposal summary
    10. Appendices (assorted documentation, summary of phases, correspondence, and so on)

# The Project Proposal

- A cover letter to managers and the IT task force should accompany the systems proposal
    - It should list the study team and summarize the principal objectives of the study
    - Keep the  cover letter concise and friendly
- Include on the title page the name of the project, the names of the systems analysis team  members, and the date the proposal is submitted
- The proposal title must accurately express the content of the proposal, but it can also exhibit some imagination
- The table of contents can  be useful to readers of long proposals
- The executive summary (generally 1 page) provides an overview of the proposal with the recommendations and actions (it is written on completion of the main report)
- The outline of the systems study provides:
    - Information about all the methods used in  the study and who or what was studied
    - Any questionnaires, interviews, sampling of archival data, observation, or prototyping used in the systems study should be discussed in this  section

# The Results

- The results must set out clearly:
  - What the analysis has discovered regarding the human and systems needs through all the methods described in the proposal
  - Conclusions about problems workers experience when interacting with technologies and systems that have been identified
  - List the problems or suggest opportunities identified
  - Alternative solutions that directly address the problems identified in the analysis
  - The alternatives you present  should include:
    - One that recommends keeping the system the same
    - Each alternative should  be explored separately
    - Describe the costs and benefits of each situation

# Trade-Offs

- Because there are usually trade-offs involved in any solution:
    - Include the advantages and disadvantages of each
    - Each alternative must clearly indicate what users and managers must do to implement it
    - The wording should be as clear as possible such as:
        - “Buy notebook computers for all middle managers”
        - “Purchase packaged software to support users in managing inventory”
        - “Modify the existing system through funding in-house programming efforts”
- After the systems analysis team has considered the alternative solutions:
    - It will have a definite professional opinion about which solution is most workable
    - The systems analysts’ recommendations section expresses the recommended solution
    - Include the reasons supporting the team’s recommendation so that it is easy to understand why it is being made

# The Recommendation(s)

- The  recommendation(s):
  - Should flow logically from the analysis of alternative solutions
  - Should clearly relate the human–computer interaction findings for the choice offered
  - The proposal conclusions and summary is a brief statement that mirrors the content of the executive summary
    - It gives the objectives of the study and the recommended solution
    - Stress the project’s importance and feasibility with the value of the recommendations for reaching the users’ goals and improving the business
  - Conclude the proposal on a positive note
  - The appendices are the last part of the systems proposal and can include:
    - Any  information that you feel may be of interest to specific individuals but that is not essential for understanding the systems study and what is being proposed

# The Presentation

- Once the systems proposal is written and edited:
  - Select who should receive the report
  - Personally hand the report to the people you have selected
  - Visibility is important for the acceptance and eventual success of the system
  - A special meeting (of key decision makers) may be proposed for the purpose of delivering the results of the systems proposal
  - Develop a separate oral presentation (illustrated with PowerPoint slides) that features highlights of your written report
  - Keep your presentation brief (30  to 40 minutes maximum)
  - Devote the majority of the allotted time to questions
- Never read your  report aloud:
  - It is better to prepare and deliver a dynamic slide presentation that takes advantage of the fact that you can interact with key decision makers face to face

# Using Figures for Effective Communication.

# Figures and Tables

- In this section we have considered the audience for a systems proposal
- Tables, graphs, and written text (words) are important in capturing and communicating the basics of the proposed system
- Never underestimate good design
  - Using figures in a proposal helps demonstrate that you are responsive to the different ways people absorb information
  - Figures in the report supplement written information  and must always be interpreted in words (they should never be used alone.

# Effective use of Tables

- Tables are (technically) not visual aids:
  - However: they provide a different way of grouping and presenting analysed data to proposal  readers
- Tables use labelled columns and rows:
  - To present statistical or alphabetical data in an  organized way
  - Each table must be numbered according to the order in which it appears in the proposal and should be meaningfully titled.
- Figure 3.26 shows the appropriate layout and  labelling for a table

# Effective use of Tables

- Figure 3.26 shows some guidelines for effective tables:
    1. Integrate tables into the body of the proposal (don’t relegate them to the appendices)
    2. Try to fit an entire table (if possible) vertically on a single page
    3. Number and title a table at the top of the page and make the title descriptive and meaningful
    4. Label each row and column (use more than one line for a title if necessary)
    5. Use a boxed table if room permits (vertically ruled columns will enhance readability)
    6. Use footnotes (if required) to explain detailed information contained in the table

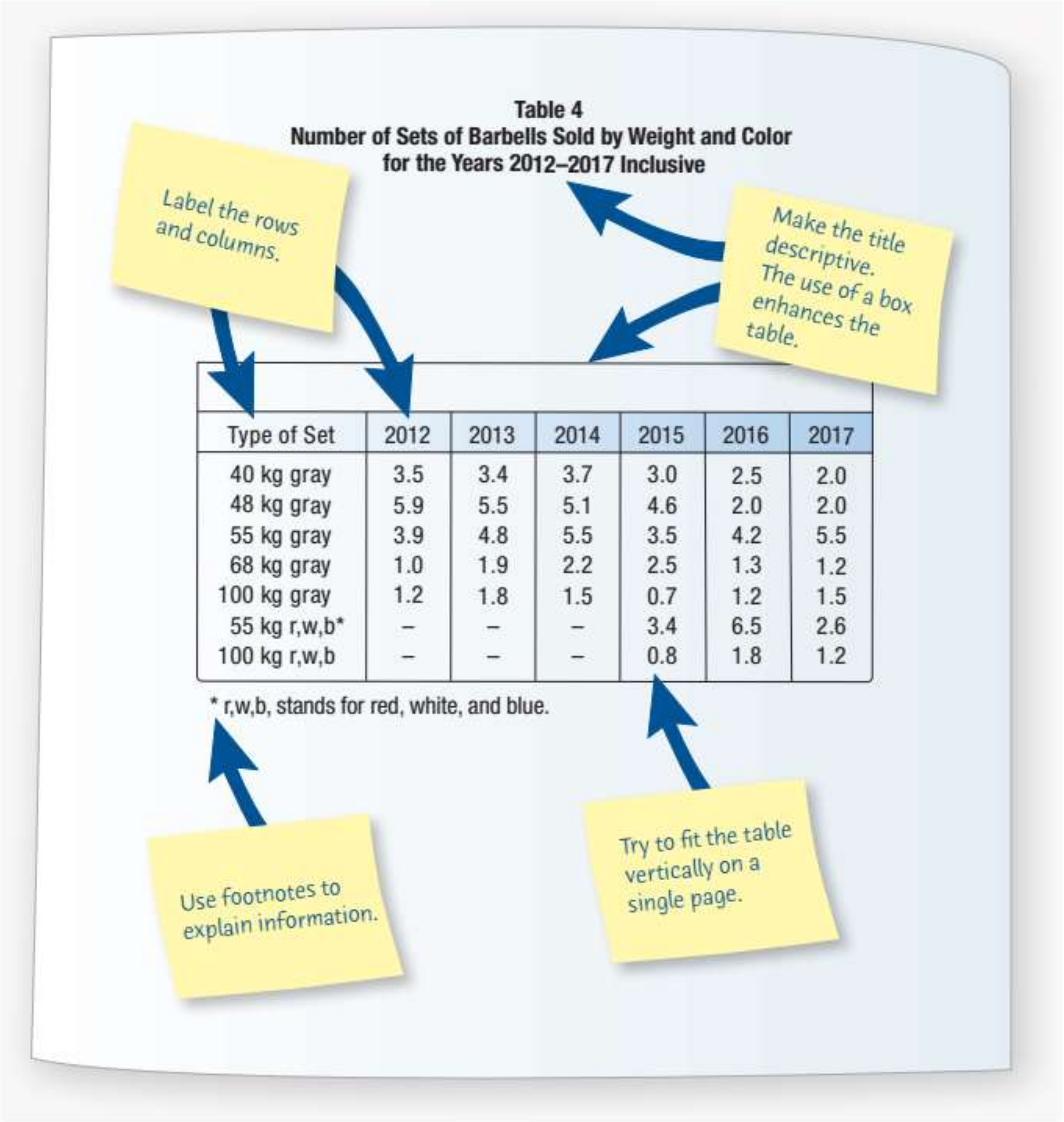


Table 4
Number of Sets of Barbells Sold by Weight and Color
for the Years 2012–2017 Inclusive

| Type of Set | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
|---|---|---|---|---|---|---|
| 40 kg gray | 3.5 | 3.4 | 3.7 | 3.0 | 2.5 | 2.0 |
| 48 kg gray | 5.9 | 5.5 | 5.1 | 4.6 | 2.0 | 2.0 |
| 55 kg gray | 3.9 | 4.8 | 5.5 | 3.5 | 4.2 | 5.5 |
| 68 kg gray | 1.0 | 1.9 | 2.2 | 2.5 | 1.3 | 1.2 |
| 100 kg gray | 1.2 | 1.8 | 1.5 | 0.7 | 1.2 | 1.5 |
| 55 kg r,w,b* | – | – | – | 3.4 | 6.5 | 2.6 |
| 100 kg r,w,b | – | – | – | 0.8 | 1.8 | 1.2 |

* r,w,b, stands for red, white, and blue.

Label the rows and columns.

Make the title descriptive. The use of a box enhances the table.

Use footnotes to explain information.

Try to fit the table vertically on a single page.

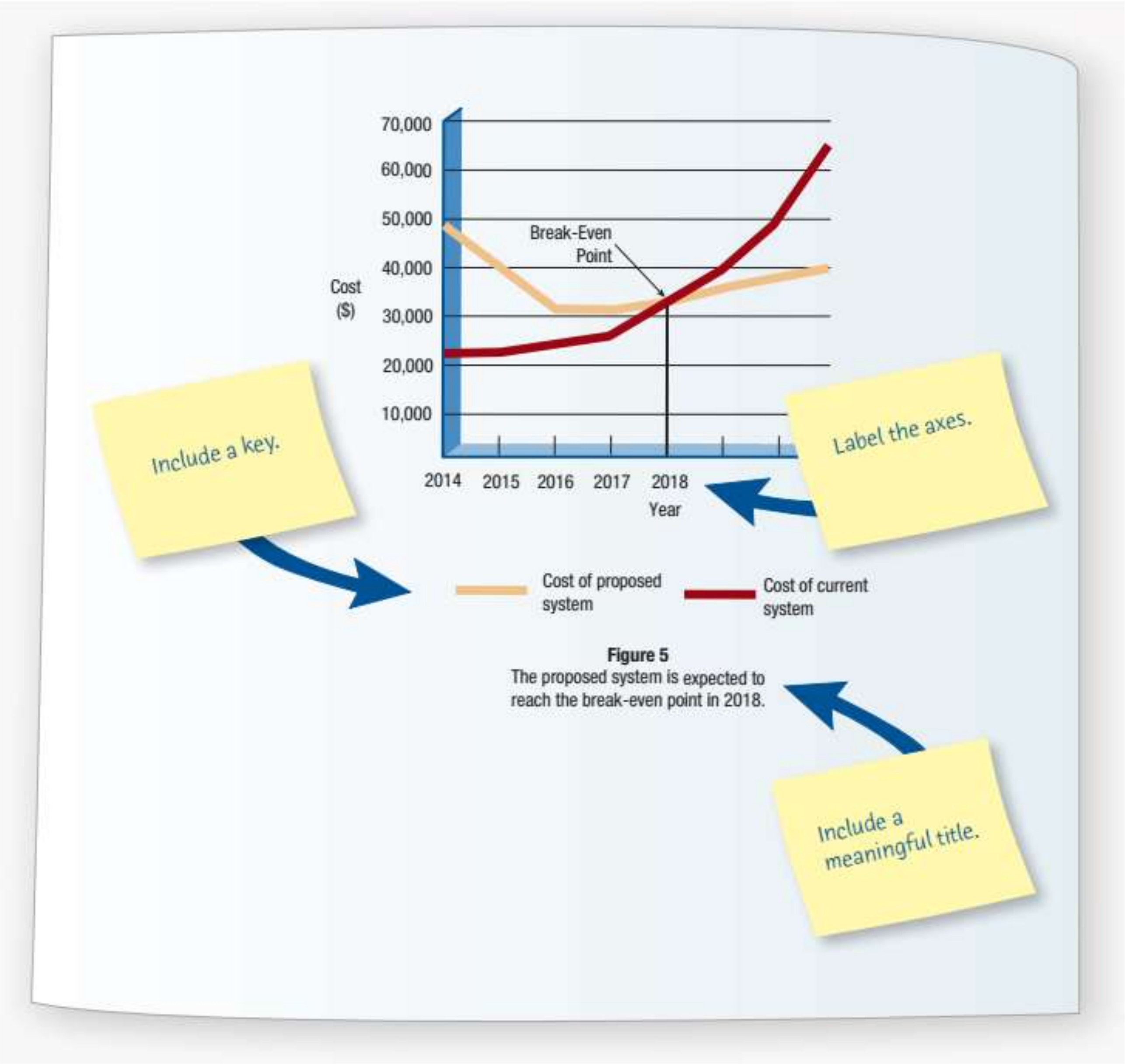
# Effective use of Graphs

- There are different types of graph including:
  - Line graphs / column graphs / bar charts / pie charts / area charts /scatter graphs / statistical graphs
  - Line graphs, column graphs, and bar charts compare variables
  - Pie charts and area charts illustrate the percentage composition of an entity
  - Scatter graphs can show data distribution
  - Statistical graphs can show the relationship in the data or trends in the data
- The following guidelines address the inclusion of effective graphs in a proposal (see Figure 3.27):
  1. Choose a style of graph that communicates your intended meaning clearly
  2. Integrate the graph into the body of the proposal
  3. Give the graph a sequential figure number and a meaningful title
  4. Label each axis (generally the X and Y but may include Z) and any lines, columns, bars, or pieces of the pie on the graph
  5. Include a key to indicate differently coloured lines / shaded bars / or crosshatched areas
- Much of the detail that goes into a systems proposal is obtained through interviews, questionnaires, sampling, discovery of other hard data, and observation
  - These topics are discussed in Chapters #4 and #5

# Effective use of Graphs

Figure 3.27 shows some guidelines for effective graphs:

1. Choose a style of graph that communicates your intended meaning clearly
2. Integrate the graph into the body of the proposal
3. Give the graph a sequential figure number and a meaningful title
4. Label each axis (generally the X and Y but may include Z) and any lines, columns, bars, or pieces of the pie on the graph
5. Include a key to indicate differently coloured lines / shaded bars / or crosshatched areas



# Summary

# Summary

- The five major project management fundamentals a systems analyst must handle are:
  1. Project initiation (defining a problem(s)
  2. Determining project feasibility
  3. Activity planning and control
  4. Project scheduling
  5. Managing systems analysis team members
- To meet the set goals and solve systems problems an analyst creates a problem definition which is a formal statement of the problem including:
  1. The issues of the present situation
  2. The objectives for each issue
  3. The requirements that must be included in all proposed systems
  4. The constraints that  limit system development

# Summary

- Selecting a project can be a difficult process which is driven by 5 criteria:
  - A project must be supported by all stakeholders including management
  - There must be a commitment of resources
  - The identification (and attainment) of pre-set goals
  - The project must be a practical proposition
  - The project must be important to the commissioning organization (and users)
- Projects must be feasible on:
  - An operational level
  - A technical level
  - An economic level

# Summary

- Additional factors to consider are:
  - Acquiring hardware and software
    - Including *in-house* and *external* supply (including e.g., cloud-based services)
  - Work breakdown structure
  - Project planning
    - Gantt charts
    - PERT charts
    - Management of costs
  - Team management
    - Including control and motivation
  - Ecommerce projects
  - Preparing a system proposal
    - The written document
    - The oral presentation

# Summary

- In Chapter #3 we have introduced the steps and processes which are required to effectively initiate and manage a project
- While all the steps and processes are central to effective project management:
  - Effective project management is both a *skill* and an *art*
  - Good systems analysts and project managers will have acquired *experience* over time working on a range of projects
  - The experience gained will be on a technical, planning, and social level
  - In managing projects both explicit and tacit knowledge will be leveraged to addres (and anticipate) problems

# Use Cases