

System Analysis and Design (SAD)

Dhiraj Bashyal

Computer – Aided Systems Engineering (CASE) (Introduction)

- Computer-aided software engineering (CASE) is the application of computer-assisted tools and methods in software development to ensure a high-quality and defect-free software.
- CASE ensures a check-pointed and disciplined approach and helps designers, developers, testers, managers and others to see the project milestones during development.
- CASE tools are programs (software) that automate or support one or more phases of a systems development life cycle

CASE Classification

1. CASE tools are classified according to which phases of the life cycle they support.
 - ✓ **Upper-CASE:** The term **upper-case** describes tools that automate or support the 'upper' phases of systems development – the survey, study, definition, and design phases.
 - ✓ **Lower-CASE:** The term **lower-CASE** describes tools that automate or support the 'lower' phases of systems development – detailed design, construction, and implementation (and also support).

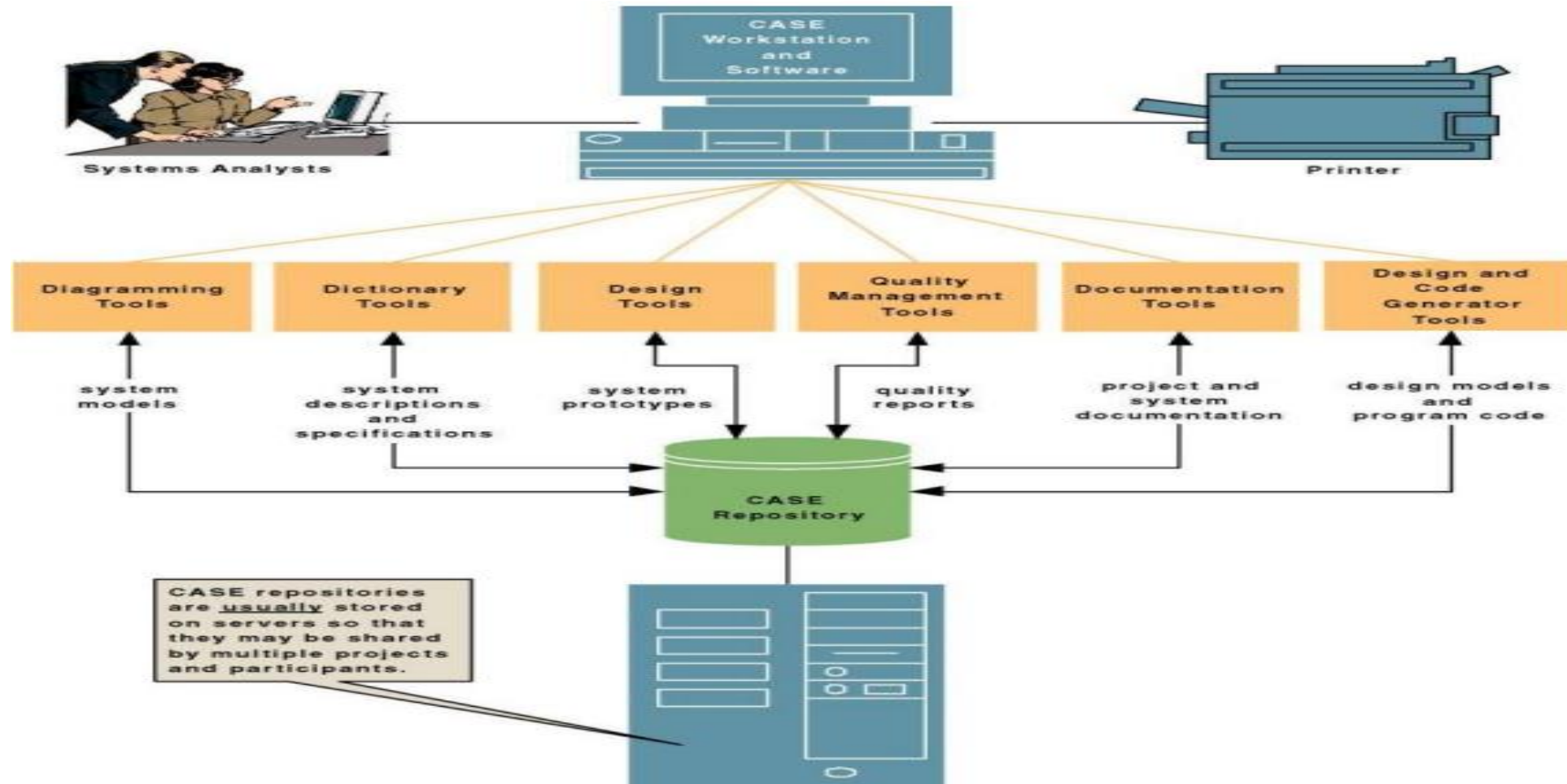
CASE Repository

- At the center of any CASE tool's architecture is a developer's database called a CASE repository.
- CASE repository is a system developer's database where developers can store system models, detailed description and specification, and other products of system development.
- It is also called dictionary or encyclopedia.
- Around the CASE repository is a collection of tools or facilities for creating system models and documentation.

CASE Repository Facilities

- **Diagramming tools** – These tools are used to draw system models.
- **Dictionary tools** – These tools are used to record, delete, edit, and output detailed documentation and specification.
- **Design tools** – These tools are used to construct system components including system inputs and outputs. These are also called prototyping tools.
- **Documentation tools** – These tools are used to assemble, organize, and report on system models, descriptions and specifications, and prototypes.
- **Quality management tools** – These tools are used to analyze system models, descriptions and specifications, and prototypes for completeness, consistency, and conformance to accepted rules of methodologies.
- **Design and code generator tools** – These tools automatically generate database designs and application programs or significant portions of those programs.

CASE TOOL



Object Oriented Analysis and Design[1]

- Object-oriented analysis and design (OOAD) is a Systems development methodologies and techniques based on objects rather than data or processes.
- The object-oriented approach combines data and processes (called methods) into single entities called objects.
- Objects usually correspond to the real things an information system deals with, such as customers, suppliers, contracts, and rental agreements.
- The goal of OOAD is to make systems elements more reusable, thus improving system quality and the productivity of systems analysis and design.

Object Oriented Analysis and Design[2]

- Object-oriented techniques are thought to work well in situations in which complicated information systems are undergoing continuous maintenance, adaptation, and redesign.
- Object-oriented approaches use the industry standard for modeling object-oriented systems, called the unified modeling language (UML), to break down a system into a use case model.

Object[1]

- Object is an abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interact with it, or both.
- Objects are entities in a software system which represents instances of real-world and system entities.

Object[2]

Object	Identity	Behaviors	State
An Employee	"Mr. John"	Join(), Retire()	Joined. Retired
A Book	"Book with title object oriented analysis design"	Add Exemplar,	Rent, Available, Reserved
A Sale	"Sale no 9491, 06/06/2021"	SendInvoiced(), Cancel()/	Invoiced, Cancelled

Class[1]

- A class represents a collection of objects having same characteristic properties that exhibit common behavior.
- It gives the blueprint or description of the objects that can be created from it. Creation of an object as a member of a class is called instantiation. Thus, object is an instance of a class.

Difference between class and object

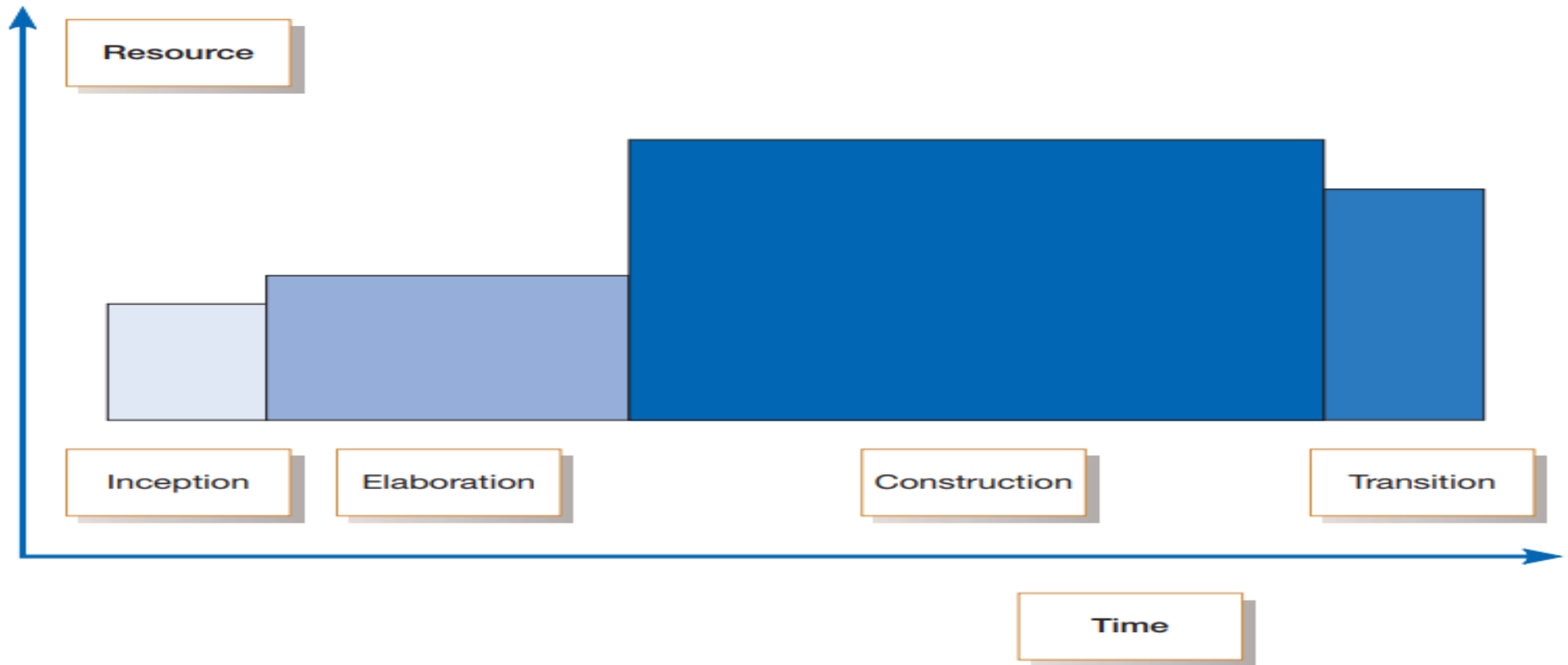
Employee
Name: string Address: string dateOfBirth: Date Employee No: integer Department: Dept Salary: integer
join() leave() retire() changedetails()

Employee 16
Name: Ram Thapa Address: Kathmandu dateOfBirth: 10/08/2005 Employee No: 98 Department: IT Salary: 50000
Employee16.join(03/05/2020) Employee16.retire(02/08/2021)

OO Analysis

- Examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain. In other words, the world (of the system) is modelled in terms of objects and classes
 - Elicit requirements: Define what does the software need to do, and what's the problem the software trying to solve.
 - Specific Requirements: Describe the requirements, usually, using use cases(and scenarios) or user stories
 - Conceptual Model: Identify the important objects, refine them, and define their relationships and behavior and draw them in a simple diagram.

Object Oriented Analysis and Design



Advantages of OO Analysis and Design

- Focuses on data rather than the procedures as in structured analysis
- The principles of encapsulation and data hiding help the developer to develop systems that cannot be tampered by other parts of the system.
- It allows effective management of software complexity by the virtue of modularity.
- It can be upgraded from small to large systems at a greater ease than in systems following structured analysis.

Disadvantages of OO Analysis and Design

- Functionality is restricted within objects. This may pose a problem for systems which are intrinsically procedural or computational in nature.
- It cannot identify which objects would generate an optimal system design.
- The object-oriented models do not easily show the communications between the objects in the system.
- All the interfaces between the objects cannot be represented in a single diagram.
- Lack of manpower who are efficient in the OO analysis and design.

Assignment 2

- Explain outsourcing
- Describe six different sources of software
- Explain reuse and its role in software development

System Acquisition

Although there will always be some debate about when and where the first administrative information system was developed, there is general agreement that the first such system in the united kingdom was developed at J.Lyons & Sons.

In the United States, the first administrative information system was General Electric's (GE) Payroll system, which was developed in 1954 (Computer History museum, 2003)

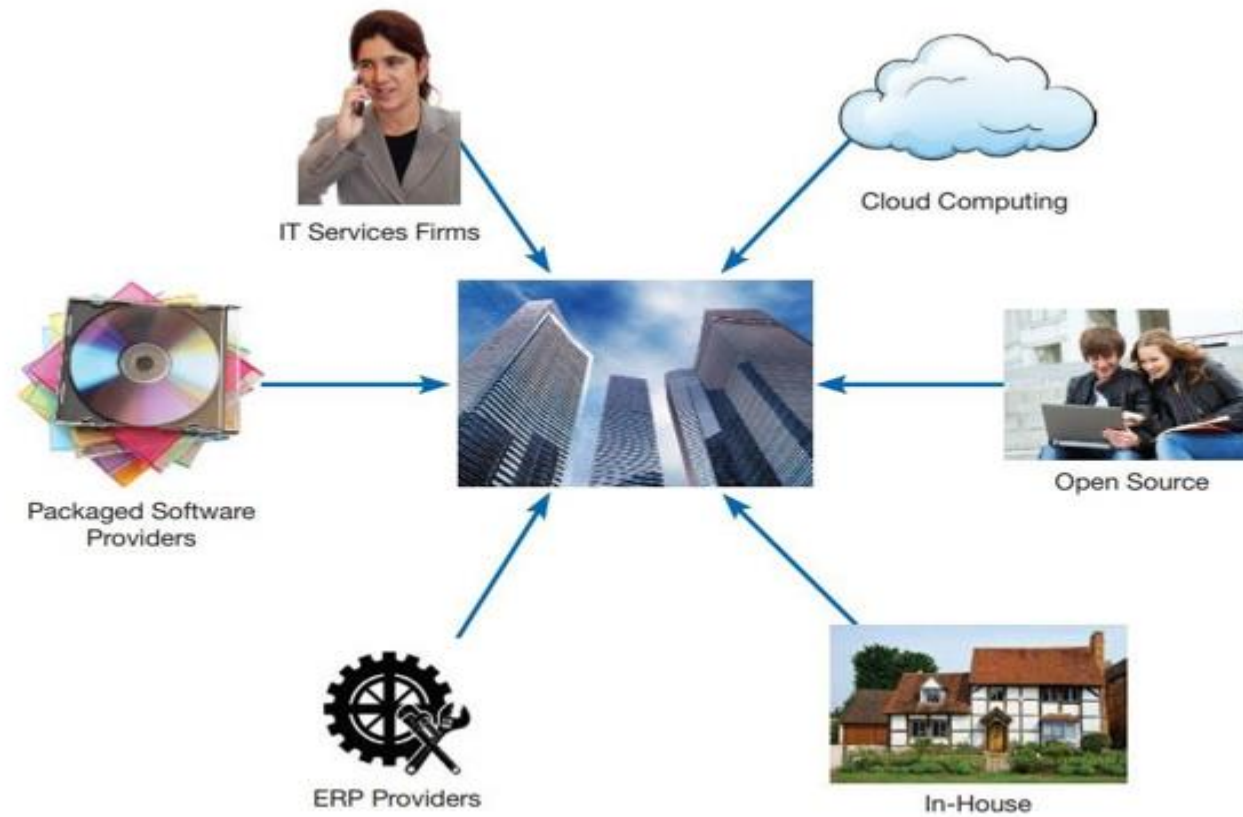
Outsourcing

- If one organization develops or runs a computer application for another organization, that practice is called outsourcing.
- Outsourcing includes a spectrum of working arrangements.
- At one extreme is having a firm develop and run your application on its computers – all you do is simply input and take output.
- A common example of such an arrangement is a company that runs payroll applications for clients so that clients do not have to develop an independent in-house payroll system. Instead, they simply provide employee payroll information to the company, and, for a fee, the company returns completed paychecks, payroll accounting reports, and tax and other statements for employees.

Reasons behind outsourcing are

- Freeing up internal resources
- Increasing the revenue potential of the organization
- Reducing time to market
- Increasing process efficiencies
- Outsourcing non core activities

Sources of Software



Information Technology Services Firms

- If a company needs an information system but does not have the expertise or the personnel to develop the system in-house, and a suitable **off-the-shelf system** is not available, the company will likely consult an information technology services firm.
- IT services firms help companies develop custom information systems for internal use, or they develop, host, and run applications for customers, or they provide other services.
- These firms employ people with expertise in the development of information systems. Their consultants may also have expertise in a given business area.

Packaged Software Producers

- The growth of the software industry has been phenomenal since its beginnings in the mid-1960s.
- Some of the largest computer companies in the world are companies that produce software exclusively.
- A good example is Microsoft, probably the best-known software company in the world. Almost 87 percent of Microsoft's revenue comes from its software sales, mostly for its Windows operating systems and its personal productivity software, the Microsoft Office Suite.

Enterprise Software Solutions

- Many firms have chosen complete software solutions, called enterprise solutions or enterprise resource planning (ERP) systems, to support their operations and business processes.
- These ERP software solutions consist of a series of integrated modules.
- Each module supports an individual, traditional business function, such as accounting, distribution, manufacturing, or human resources.
- The traditional approach would use different systems in different functional areas of the business, such as a billing system in accounting and an inventory system in the warehouse.

Enterprise Software Solutions

- Using enterprise software solutions, a firm can integrate all parts of a business process in a unified information system.
- All aspects of a single transaction occur seamlessly within a single information system, rather than as a series of disjointed, separate systems focused on business functional areas.
- The benefits of the enterprise solutions approach include a single repository of data for all aspects of a business process and the flexibility of the modules.

Cloud Computing Vendors

- Cloud computing is the provision of computing resources, including applications, over the Internet, so customers do not have to invest in the computing infrastructure needed to run and maintain the resources.
- Another method for organizations to obtain applications is to rent them or license them from third-party providers who run the applications at remote sites.
- Users have access to the applications through the Internet or through virtual private networks.
- The application provider buys, installs, maintains, and upgrades the applications. Users pay on a per-use basis or they license the software, typically month to month.

Cloud Computing Vendors

- A well-known example of cloud computing is Google Apps, where users can share and create documents, spreadsheets, and presentations

Open Source Software

- Open-source software is different because it is freely available, not just the final product but the source code itself.
- It is also different because it is developed by a community of interested people instead of by employees of a particular company.
- Open-source software performs the same functions as commercial software, such as operating systems, e-mail, database systems, web browsers, and so on.
- Some of the most well-known and popular open-source software names are Linux, an operating system; MySQL, a database system; and Firefox, a web browser.

Open Source Software

- Open source is developed and maintained by communities of people, and sometimes these communities can be very large.
- If the software is free, you might wonder how anybody makes any money by developing open-source software.
- Companies and individuals can make money with open source in two primary ways:
 - by providing maintenance and other services or
 - by providing one version of the software free and selling a more fully featured version.

In House Developers

- In-house development has become a progressively smaller piece of all systems development work that takes place in and for organizations.
- In-house development can lead to a larger maintenance burden than other development methods, such as packaged applications.
- Of course, in-house development need not entail development of all of the software that will constitute the total system.
- Hybrid solutions involving some purchased and some in-house software components are common.
- If you choose to acquire software from outside sources, this choice is made at the end of the analysis phase. The choice between a package and an external supplier will be determined by your needs, not by what the supplier has to sell.

In House Developers

TABLE 2-2 Comparison of Six Different Sources of Software Components

Producers	When to Go to This Type of Organization for Software	Internal Staffing Requirements
IT services firms	When task requires custom support and system can't be built internally or system needs to be sourced	Internal staff may be needed, depending on application
Packaged software producers	When supported task is generic	Some IS and user staff to define requirements and evaluate packages
Enterprise-wide solutions vendors	For complete systems that cross functional boundaries	Some internal staff necessary but mostly need consultants
Cloud computing	For instant access to an application; when supported task is generic	Few; frees up staff for other IT work
Open-source software	When supported task is generic but cost is an issue	Some IS and user staff to define requirements and evaluate packages
In-house developers	When resources and staff are available and system must be built from scratch	Internal staff necessary though staff size may vary

Reuse

- Reuse is the use of previously written software resources, especially objects and components, in new applications.
- Because so many bits and pieces of applications are relatively generic across applications, it seems intuitive that great savings can be achieved in many areas if those generic bits and pieces do not have to be written a new each time they are needed.
- Reuse should increase programmer productivity because being able to use existing software for some functions means they can perform more work in the same amount of time.
- Reuse should also decrease development time, minimizing schedule overruns. Because existing pieces of software have already been tested, reusing them should also result in higher-quality software with lower defect rates, decreasing maintenance costs.

Advantages of Reuse

- Increase software productivity
- Shorten software development time
- Improve software system interoperability
- Develop software with fewer people
- Move personnel more easily from project to project
- Reduce software development and maintenance costs
- Produce more standardized software
- Produce better quality software and provide a powerful competitive advantage