




# **Modern Systems Analysis and Design**

**Seventh Edition**

**Jeffrey A. Hoffer  
Joey F. George  
Joseph S. Valacich**

## **Chapter 2**

### **The Origins of Software**



# Learning Objectives

- ✓ Explain outsourcing.
- ✓ Describe six different sources of software.
- ✓ Discuss how to evaluate off-the-shelf software.
- ✓ Explain reuse and its role in software development.



# Introduction

- There are various sources of software for organizations.
- Most of a corporation's application software is created by external sources.
- Much in-house coding involves making components work together.
- There are criteria to evaluate software from different sources.



# Systems Acquisition: Outsourcing

- **Outsourcing:** Turning over responsibility of some or all of an organization's information systems applications and operations to an outside firm



# Systems Acquisition: Outsourcing (Cont.)

## ■ Outsourcing Example

- Shell Oil outsource spending: \$3.2 billion (2008)
- Shell's outsourcing vendors (2008-2011): EDS, T-Systems, AT&T, IBM, Logica, Wipro, Accenture



# Outsourcing (Cont.)

## ■ Reasons to outsource

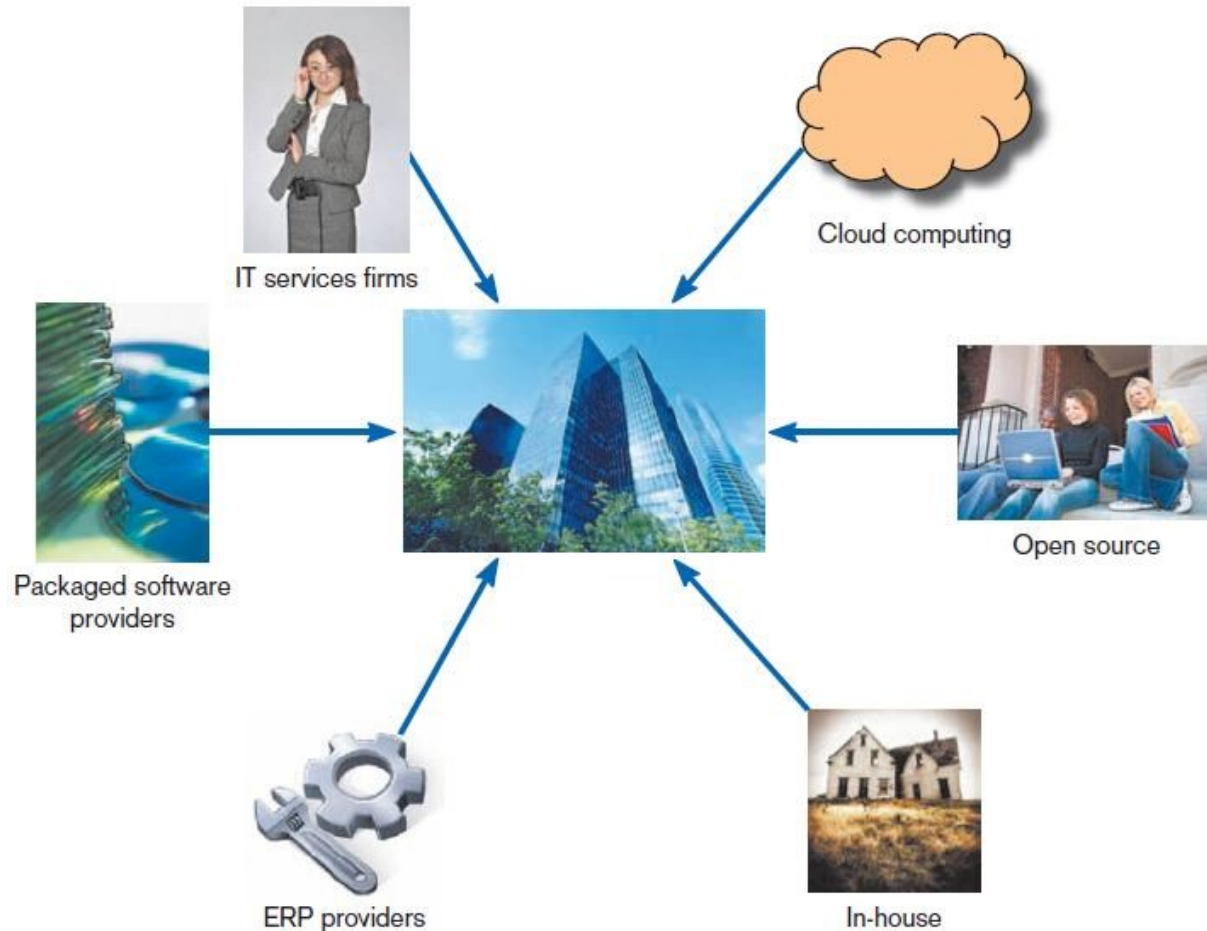
- Cost-effectiveness
- Take advantage of economies of scale
- Make up for lack of in-house knowledge
- Free up internal resources
- Reduce time to market
- Increase process efficiencies
- System development is a non-core activity for the organization



# Sources of Software

- Information technology services firm
- Packaged software producers
- Enterprise-wide solutions
  - Enterprise Resource Planning (ERP)
- Cloud computing
- Open source software
- In-house developers

# Sources of Software (Cont.)



**FIGURE 2-1**  
Sources of Application Software



# Sources of Software (Cont.)

**TABLE 2-1** Leading Software Firms and Their Development Specializations

| Specialization                | Example Firms or Websites                                     |
|-------------------------------|---|
| IT Services                   | Accenture<br>Computer Sciences Corporation (CSC)<br>IBM<br>HP |
| Packaged Software Providers.  | Intuit<br>Microsoft<br>Oracle<br>SAP AG<br>Symantec           |
| Enterprise Software Solutions | Oracle<br>SAP AG  |
| Cloud Computing               | Amazon.com<br>Google<br>IBM<br>Microsoft<br>Salesforce.com    |
| Open Source                   | SourceForge.net   |



# Information Technology (IT) Services Firms

- Help companies develop custom information systems for internal use
- Develop, host, and run applications for customers
- Provide other services



# Packaged Software Producers

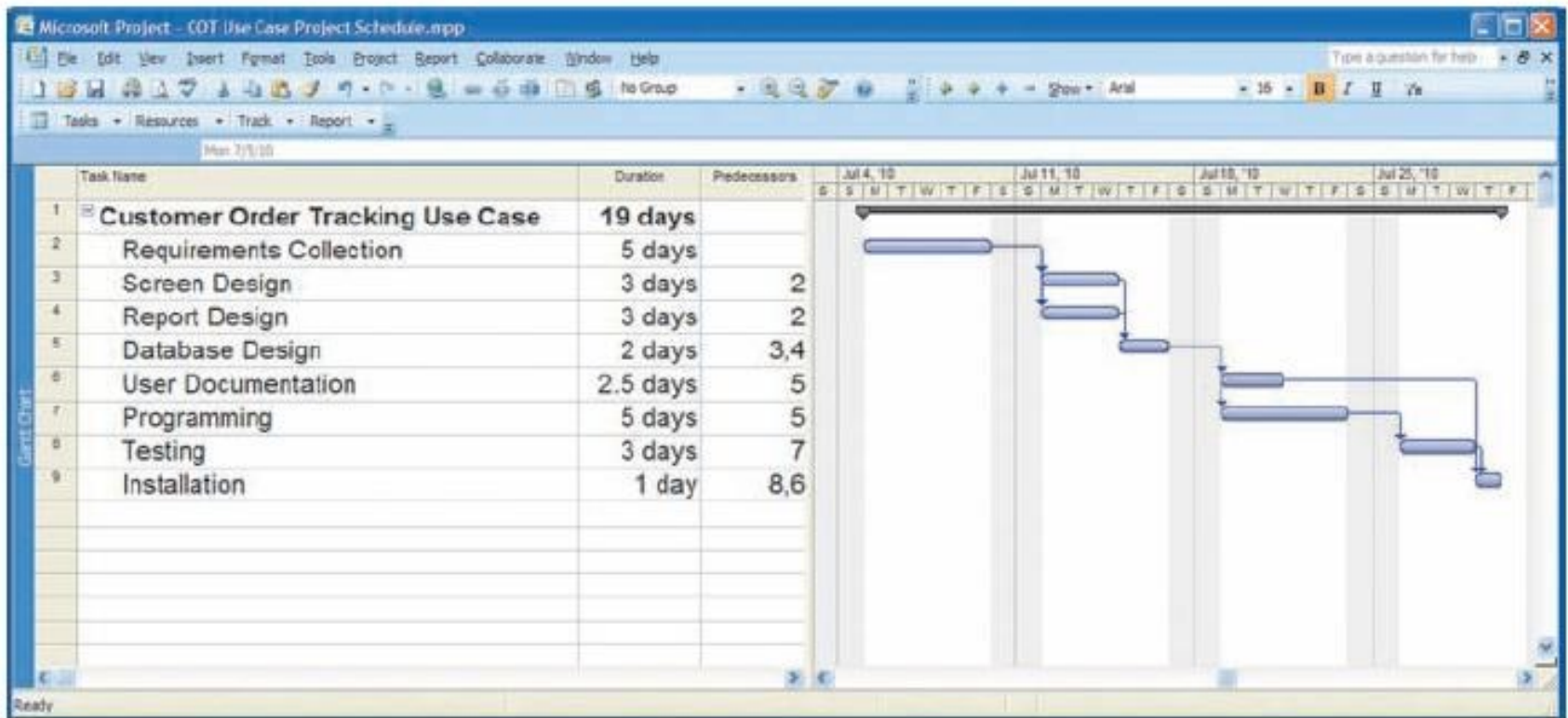
- Serve many market segments
- Provide software ranging from broad-based packages (i.e. general ledger) to niche packages (i.e. day care management)
- Pre-packaged, off-the-shelf software



# Packaged Software Producers (Cont.)

- Software runs on all size computers, from microcomputers to large mainframes.
- Prepackaged software is off-the-shelf, turnkey software (i.e. not customizable).
- Off-the-shelf software, at best, meets 70% of organizations' needs.

# Prepackaged Software



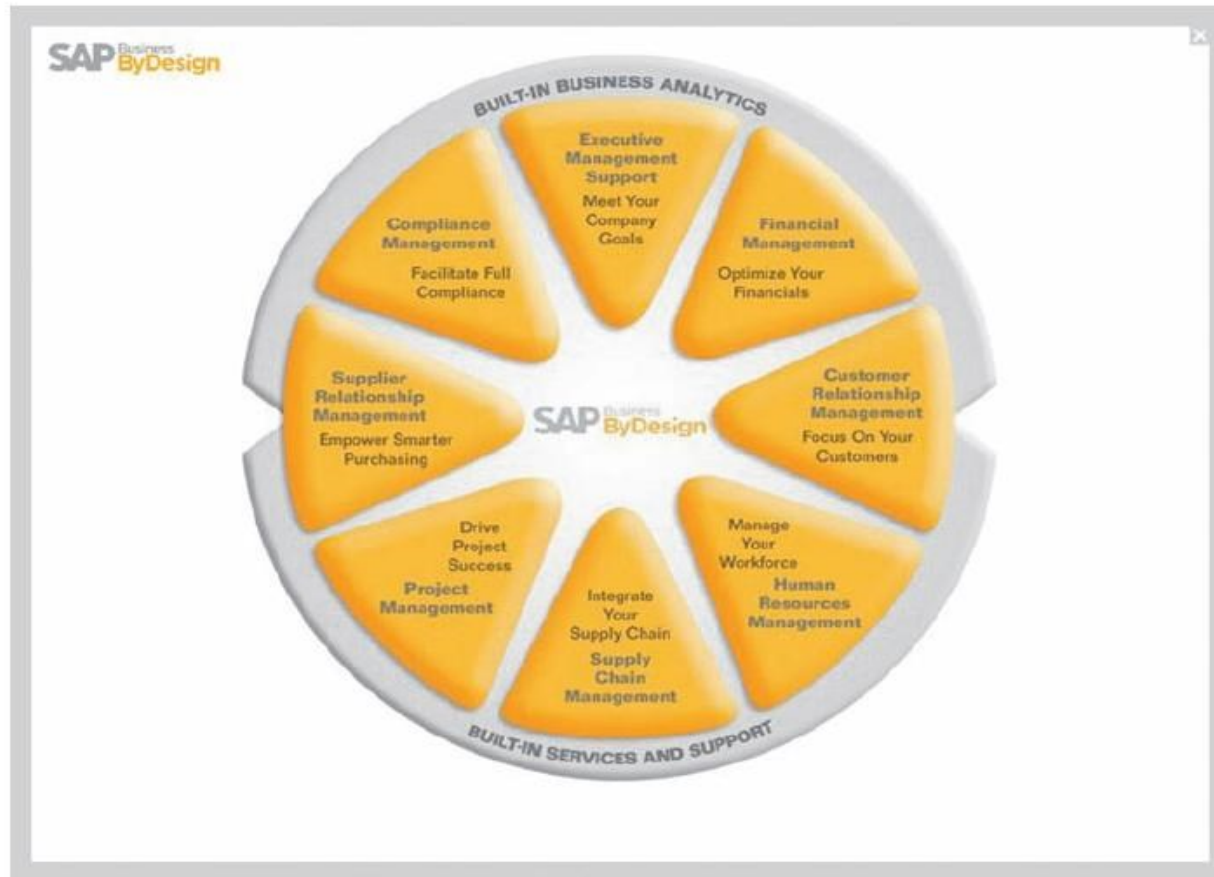
**Figure 2-2** Microsoft Project (*Source:* Microsoft Corporation.)



# Enterprise Solutions Software

- ***Enterprise Resource Planning (ERP)*** systems integrate individual traditional business functions into modules enabling a single seamless transaction to cut across functional boundaries.
- SAP AG is the leading vendor of ERP systems.

# Enterprise Solutions Software (Cont.)



**Figure 2-3** SAP's Business ByDesign, a product designed for medium sized companies.

(Source: [www.sap.com/usa/solutions/Sme/Businessbydesign/Flash/bsm/A1S.html](http://www.sap.com/usa/solutions/Sme/Businessbydesign/Flash/bsm/A1S.html).)

© Copyright SAP AG. All rights reserved.)



# Cloud Computing

- 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





# Cloud Computing (Cont.)

## ■ Examples:

- Google Drive – users store what they want on servers
- Salesforce.com – online customer relationship management (CRM) software
- Microsoft Azure platform
- Amazon.com cloud infrastructure and services



# Cloud Computing (Cont.)

- Heavy growth predicted
- Benefits:
  - Free internal IT staff
  - Faster access to application than via internal development
  - Lower cost than internal development
- Concerns
  - Security
  - Reliability
  - Regulation compliance



# Open Source Software

- Freely available including source code
- Developed by a community of interested people
- Performs the same functions as commercial software
- Examples: Linux, mySQL, Firefox
- How to make money?
  - Provide maintenance/services
  - Sell a more featured version of the free software




# In-House Development

- If sufficient system development expertise with the chosen platform exists in-house, then some or all of the system can be developed by the organization's own staff.
- Hybrid solutions involving some purchased and some in-house components are common.

# Sources of Software Components

**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                 |



# Selecting Off-the-Shelf Software

- **Cost:** comparing the cost of developing the same system in-house with the cost of purchasing or licensing the software package



# Selecting Off-the-Shelf Software (Cont.)

- **Functionality:** the tasks that the software can perform and the mandatory, essential, and desired system features



# Selecting Off-the-Shelf Software (Cont.)

- **Vendor support:** whether and how much support the vendor can provide and at what cost





# Selecting Off-the-Shelf Software (Cont.)

- **Viability of vendor:** can vendor continue to adapt/update software to changes in systems software and hardware



# Selecting Off-the-Shelf Software (Cont.)

- **Flexibility:** the ease with which software is customized
- **Documentation:** understandable and up-to-date user's manual and technical documentation



# Selecting Off-the-Shelf Software (Cont.)

- **Response time:** how long it takes the software package to respond to the user's requests in an interactive session
- **Ease of installation:** a measure of the difficulty of loading the software and making it operational



# Validating Purchased Software Information

- Use a variety of information sources:
  - Collect information from vendor
  - Software documentation
  - Technical marketing literature



# Request For Proposal (RFP)

- **A request for proposal (RFP)** is a document provided to vendors to ask them to propose hardware and system software that will meet the requirements of a new system.



# Request For Proposal (RFP) (Cont.)

- Sometimes called a **Request For Quote (RFQ)**
- Analyst selects best candidates based on:
  - vendor bids
  - a variety of information sources



# Information Sources For RFP


- Vendor's proposal
- Running software through a series of tests
- Feedback from other users of the vendor's product
- Independent software testing services
- Articles in trade publications



# Reuse

- The use of previously written software resources, especially objects and components, in new applications
- Commonly applied to two different development technologies:
  - Object-oriented development
  - Component-based development






# Reuse (Cont.)

- **Object-oriented development**

- Object class encapsulates data and behavior of common organizational entities (e.g. employees)


- **Component-based development**

- Components can be as small as objects or as large as pieces of software that handle single business functions



# Reuse (Cont.)

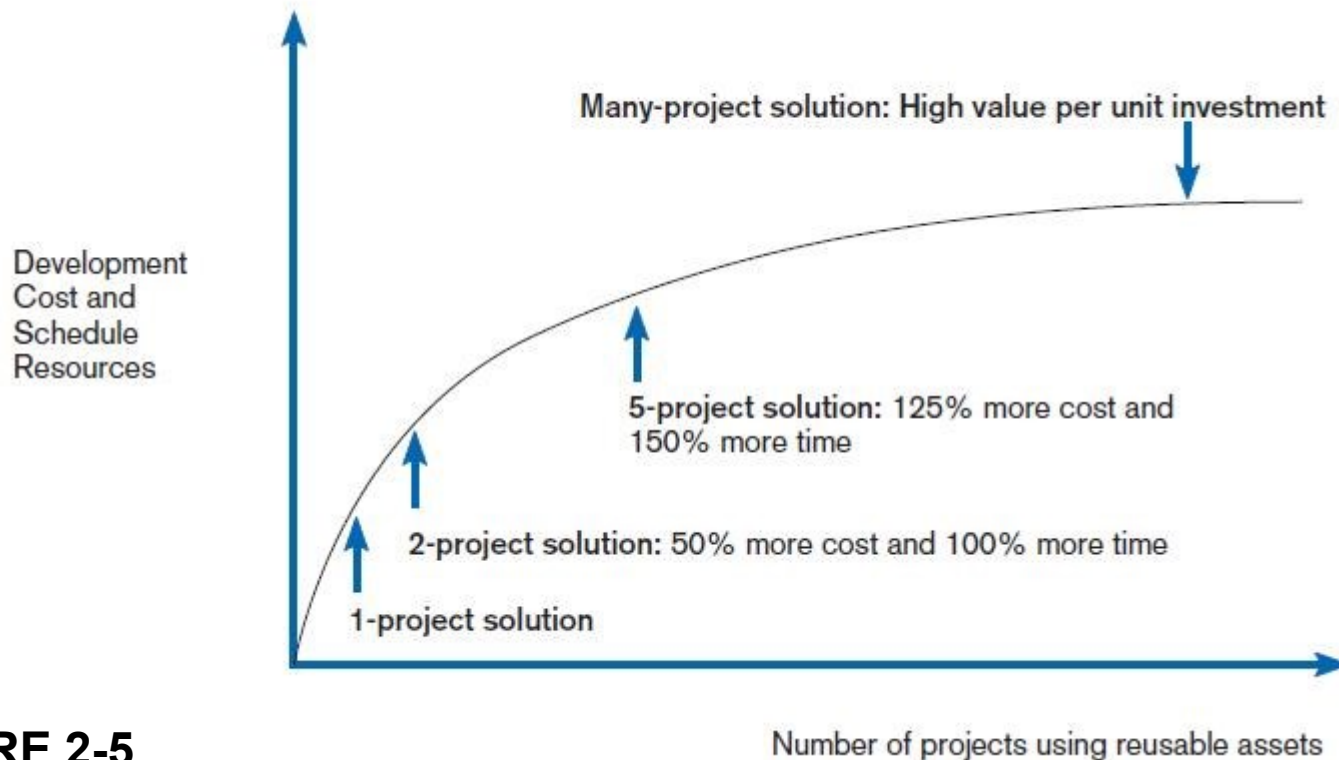
- Object-oriented development reuse is the use of object classes in more than one application (e.g. Employee).



# Reuse (Cont.)

- Component-based development reuse is the assembly of an application from many different components at many different levels of complexity and size (e.g. Currency conversion).

# Costs and Benefits of Reuse



**FIGURE 2-5**

Investments necessary to achieve reusable components

(Source: Royce, Walker, *Software Project Management: A Unified Framework*, 1st ed., ©1998. Reprinted and Electronically reproduced by permission of Pearson Education, Inc. Upper Saddle River, New Jersey.)



# Approaches to Reuse

- **Ad-hoc:** individuals are free to find or develop reusable assets on their own
- **Facilitated:** developers are encouraged to practice reuse



# Approaches to Reuse (Cont.)

- **Managed:** the development, sharing, and adoption of reusable assets is mandated
- **Designed:** assets mandated for reuse as they are being designed for specific applications

# Approaches to Reuse (Cont.)

**TABLE 2-3 Four Approaches to Reuse**

| Approach    | Reuse Level | Cost     | Policies & Procedures   |
|-------------|-------------|----------|---|
| Ad hoc      | None to low | Low      | None  |
| Facilitated | Low         | Low      | Developers are encouraged to reuse but are not required to do so.   |
| Managed     | Moderate    | Moderate | Development, sharing, and adoption of reusable assets are mandated; organizational policies are established for documentation, packaging, and certification.  |
| Designed    | High        | High     | Reuse is mandated; policies are put in place so that reuse effectiveness can be measured; code must be designed for reuse during initial development, regardless of the application it is originally designed for; there may be a corporate office for reuse. |

(Source: Based on Flashline, Inc. and Griss, 2003.)



# Summary

- In this chapter you learned how to:
  - ✓ Explain outsourcing.
  - ✓ Describe six different sources of software.
  - ✓ Discuss how to evaluate off-the-shelf software.
  - ✓ Explain reuse and its role in software development.