




Modern Systems Analysis and Design

Eighth Edition, Global Edition

**Joseph S. Valacich
Joey F. George**

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

- Historically, software development for a corporate information systems department was done primarily in-house.
- Now it involves use of components from external sources.
- Much in-house application coding involves making the components work together.



Introduction (cont.)

- Six sources of software:
 - Information technology service firms
 - Packaged software providers
 - Vendors of enterprise-wide solution software
 - Cloud computing
 - Open-source software
 - In-house development
- There are ways to evaluate software from sources

Sources of Software

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

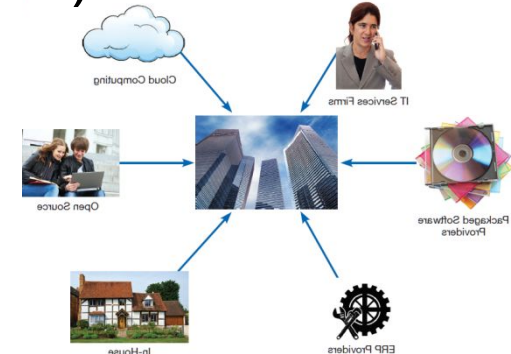


FIGURE 2-1
Sources of Application Software

- There are ways to evaluate software from sources



Systems Acquisition: Outsourcing

- **Outsourcing:** The practice of 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
- Political reasons (e.g. labor disputes)



Global Outsourcing

- Top outsourcing countries: India, China, Malaysia (A.T. Kearny report 2014)
- Top 10 are in Asia, Latin America, Europe, and Africa
- Some U.S. firms are switching to nearshoring (same time zone, low labor costs)

Sources of Software (Cont.)

TABLE 2-1 Leading Software Firms and Their Development Specializations

Specialization	Example Firms or Websites
IT Services	Accenture Computer Sciences Corporation (CSC) IBM HP
Packaged Software Providers	Intuit Microsoft Oracle SAP AG Symantec
Enterprise Software Solutions	Oracle SAP AG
Cloud Computing	Amazon.com Google IBM Microsoft 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 (management, accounting, auditing, financial)



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

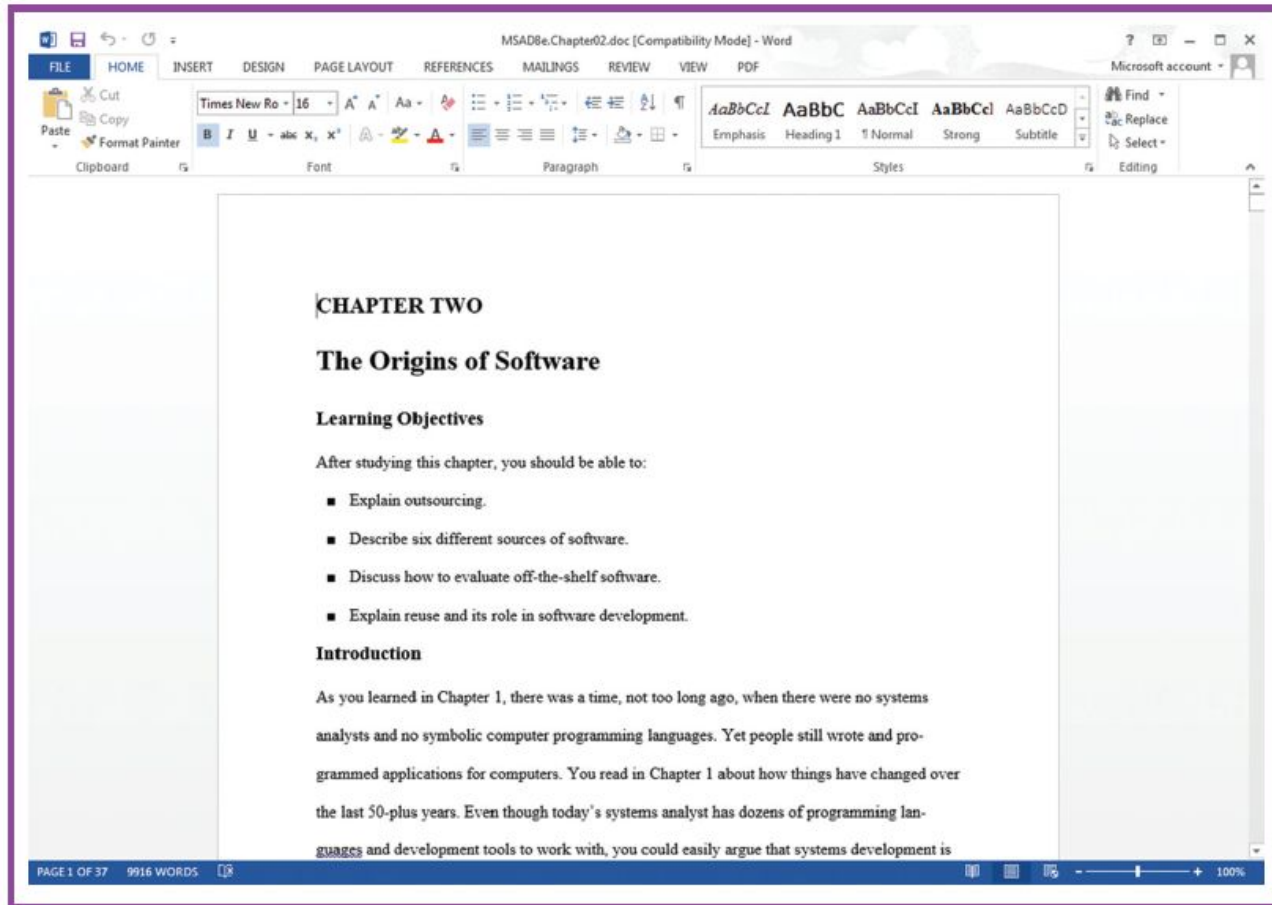


Figure 2-2 A document created in Microsoft's Word
(Source: Microsoft Corporation.)



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.)

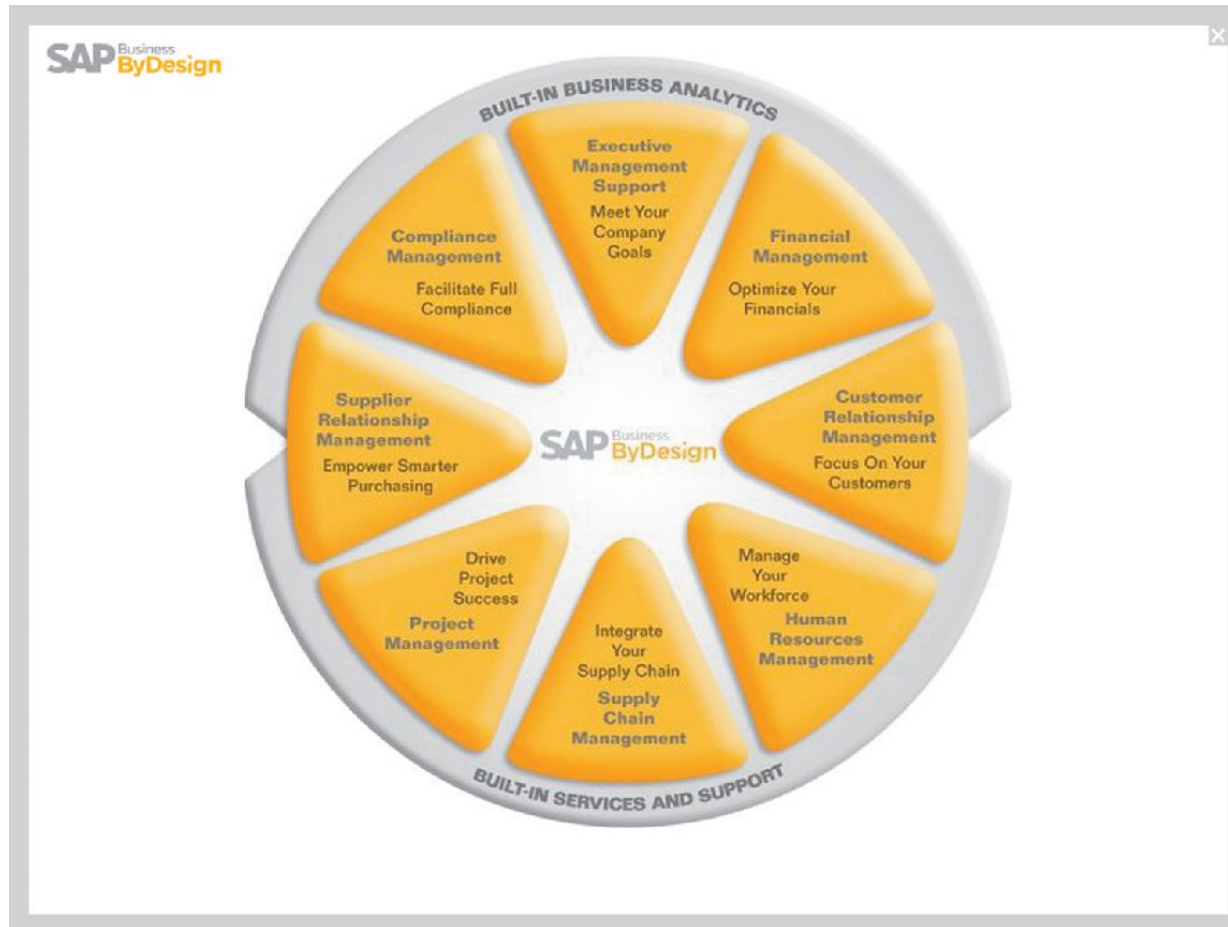


Figure 2-3 SAP's Business ByDesign, a product designed for medium sized companies.
(Source: www.sap.com/usa/solutions/Sme/Businessbydesign/Flash/bsm/A1S.html. © Copyright SAP AG. All rights reserved.)



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
- Pay-per-use or monthly/yearly licenses



Cloud Computing (Cont.)

■ Examples:

- Google Apps— for sharing documents, spreadsheets, and presentations
- Salesforce.com – online customer relationship management (CRM) software
 - An example of software as a service (SaaS)
- Microsoft Azure platform
- Amazon.com cloud infrastructure and services
 - An example of hardware as a service (HaaS)



Cloud Computing (Cont.)

- Heavy growth predicted
- Benefits:
 - Frees company of internal IT staff requirements
 - 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.
 - In-house development usually leads to more maintenance burden than other approaches
- 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
- **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
- **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

- Send a **request for proposal (RFP)** to vendors.
 - RFP – a document provided to vendors to ask them to propose hardware and system software that will meet the requirements of a new system
- Use a variety of information sources:
 - Collect information from vendor
 - Software documentation
 - Technical marketing literature



Request For Proposal (RFP)

- 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
- Customer surveys
- Articles in trade publications are sometimes biased (seeded by manufacturer)



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.)

- Can be effective (increased productivity, less defects, reduced rework)
- Technical issues – lack of methodology for component library (creating and labeling reusable components)
- Organizational issues – lack of commitment, training, and organizational support; hard to measure economic benefits; legal and contractual issues

Costs and Benefits of Reuse

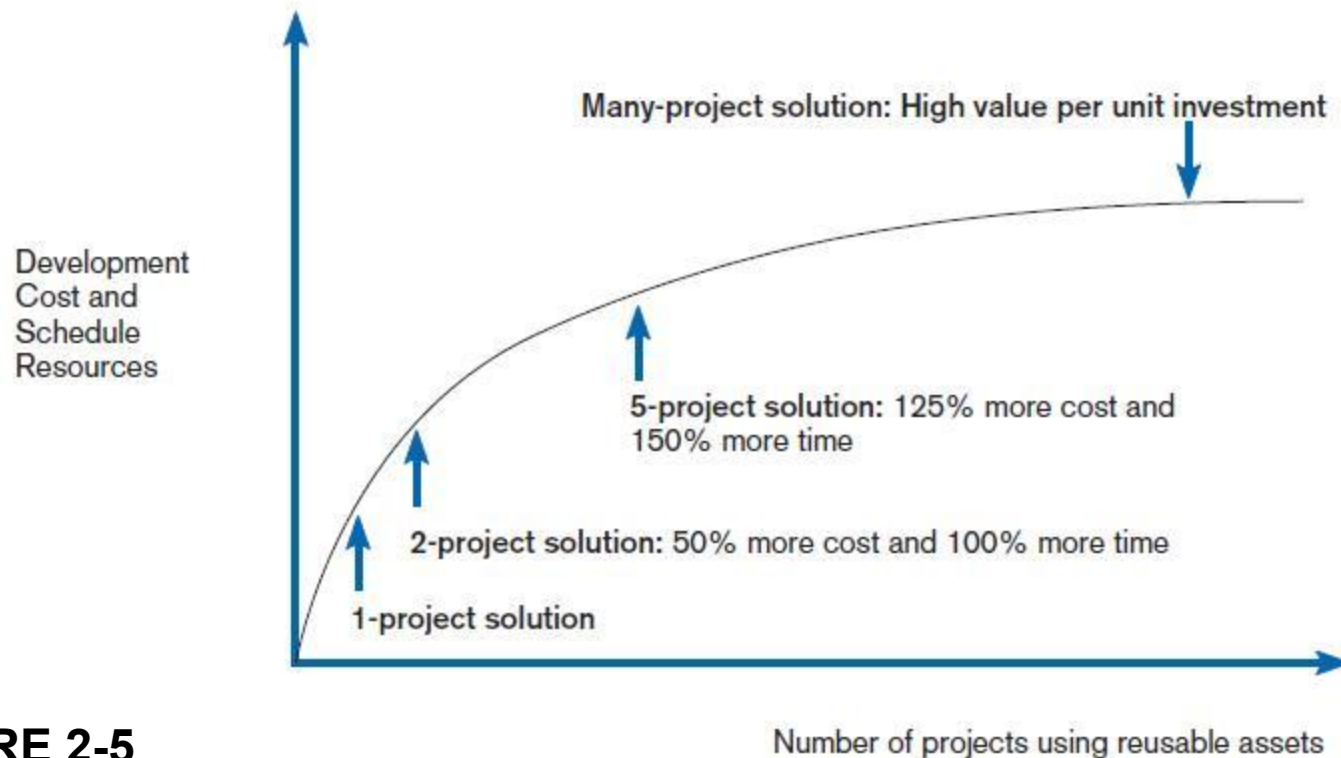


FIGURE 2-5

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.)



3 Steps of Software Reuse

- Abstraction – design of reusable piece of software
- Storage – making software assets available for others
- Recontextualization – making the software understandable to developers

(Grinter, 2001)



Approaches to Reuse

- **Ad-hoc:** individuals are free to find or develop reusable assets on their own
- **Facilitated:** developers are encouraged to practice reuse
- **Managed:** the development, sharing, and adoption of reusable assets is mandated
- **Designed:** assets mandated for reuse as they are being designed for specific applications

(Griss 2003)

Approaches to Reuse (Cont.)

TABLE 2-3 Four Approaches to Reuse

Approach	Reuse Level	Cost	Policies and 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.