AUM

American University Of The Middle East

MIS360

Foundations for Systems Development

Chapter 2: The Sources of Software

Learning Objectives

- ✓ Understand 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.

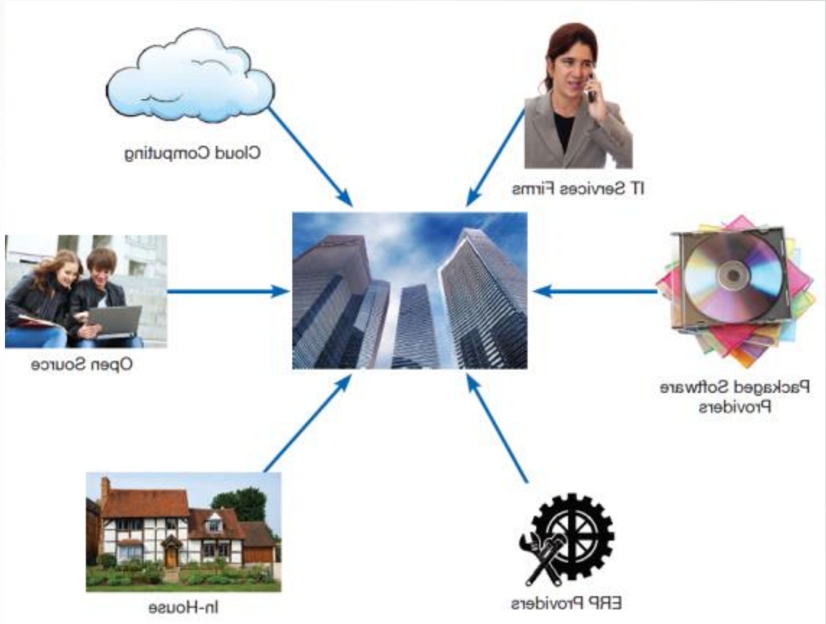
Outsourcing:

- Practice of turning over some or all responsibility for information systems applications and operations to another firm.
- Examples:
  - Outsourcing firm develops and runs applications on their computers (Payroll application)
  - Outsourcing firm runs applications at your site on your computers.
- Outsourcing should be considered during systems analysis

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)

Sources of Software:



- Six sources of software:
  1. Information technology service firms
  2. Packaged software providers
  3. Vendors of enterprise-wide solution software
     - ❑ Enterprise Resource Planning (ERP)
  4. Cloud computing
  5. Open-source software
  6. In-house development

Sources of Software (Cont.)

1. **Information Technology (IT) Service Firms**
   - Utilized when
     - Organization lacks resources to develop in-house system
     - Suitable off-the-shelf solution is not available
   - Help companies develop custom information systems for internal use
   - Develop, host and run applications for customers
   - Provide other services (management, accounting, auditing, financial)
   - IT Service firms employ consultants with domain-specific experience

Sources of Software (Cont.)

2. **<u>Packaged Software Producers</u>**
   - Serve many market segments
   - Produce pre-packed or off the shelf systems
   - Products range from *broad-based packages* (i.e. general ledger) to *industry specific packages* (i.e. day care management)
   - Two types of software
     - Turnkey – cannot be modified to meet specific users needs
     - Non-turnkey – can be modified
   - Off-the-shelf software can meet up to 70% of an organization’s needs.

Sources of Software (Cont.)

3. <u>Enterprise Solutions Software</u>

- Also called Enterprise Resource Planning (ERP) Systems as it integrate individual traditional business functions into modules enabling a single seamless transaction to cut across functional boundaries.

- Consist of a series of integrated modules
  - Each module supports individual traditional business function
    - Accounting
    - Distribution
    - Manufacturing
    - Human Resources
  - Integrated to focus on business processes rather than business functional areas

- Enables an organization to integrate all parts of a business process in a unified information system
  - All aspects of a single transaction occur seamlessly in single information system

Sources of Software: Enterprise Solutions Software(Cont.)

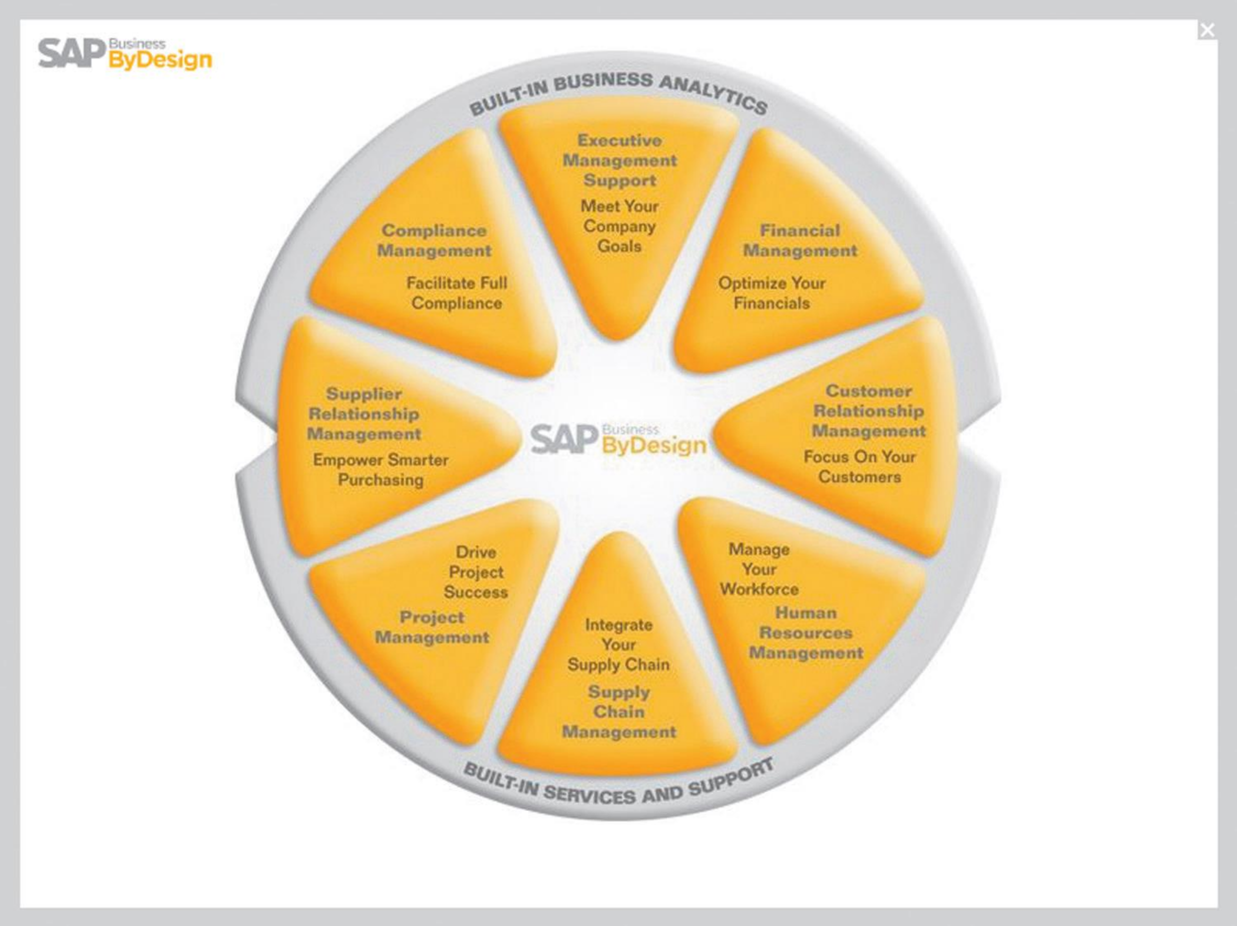
- **Benefits**
  1. Single repository for all aspects of a business process
     - Ensures more consistent and accurate data
     - Less maintenance
  2. Flexibility of modules
     - Additional modules can be added as needed
     - Additional modules are immediately integrated into existing system

Sources of Software: ERP Systems(Cont.)

- **<u>Disadvantages</u>**
  1. Complexity of implementation
     - Extended implementation time
  2. Reliance upon consultants
  3. Often, organizations must change the way that they do business in order to use systems

- ERP leading vendors
  - SAP
  - Oracle

Enterprise Solutions Software (Cont.)


SAP Business ByDesign
BUILT-IN BUSINESS ANALYTICS
Executive Management Support
Meet Your Company Goals
Financial Management
Optimize Your Financials
Customer Relationship Management
Focus On Your Customers
Manage Your Workforce
Human Resources Management
Integrate Your Supply Chain
Supply Chain Management
Drive Project Success
Project Management
Supplier Relationship Management
Empower Smarter Purchasing
Compliance Management
Facilitate Full Compliance
SAP Business ByDesign
BUILT-IN SERVICES AND SUPPORT

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.)

Sources of Software (Cont.)

4. **<u>Cloud Computing</u>**

- Software is rented or licensed from third party providers
- Applications are accessed through the Internet.
- Applications bought, installed and maintained by service provider
- Users pay per-use or month-to-month license
- Includes software as a service as well as hardware as a service
- customers do not have to invest in the computing infrastructure needed to run and maintain the resources

Sources of Software: 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)

Sources of Software: Cloud Computing (Cont.)

- Heavy growth predicted: by 2015 will create 14 million new jobs.
  - Total market of $1.1 trillion.

- **Benefits:**
  1. Frees company of internal IT staff requirements
  2. Faster access to application than via internal development
  3. Lower cost than internal development
- **Concerns**
  1. Security
  2. Reliability
  3. Compliance with government regulations

Sources of Software (Cont.)

5. **Open-Source Software**

- — Freely available
- — Includes source-code
- — Developed and Maintained by a group of interested individuals, instead of employees of a particular company.
- — Performs the same functions as commercial software, such as OS, email, database systems, and web browsers.
- — Some examples
  - Linux
  - Firefox
  - mySQL

Sources of Software (Cont.)

6. <u>In-House Development</u>

› **Complete system**

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 systems**

Hybrid solutions involving some purchased and some in-house components are common.

Sources of Software (Cont.)

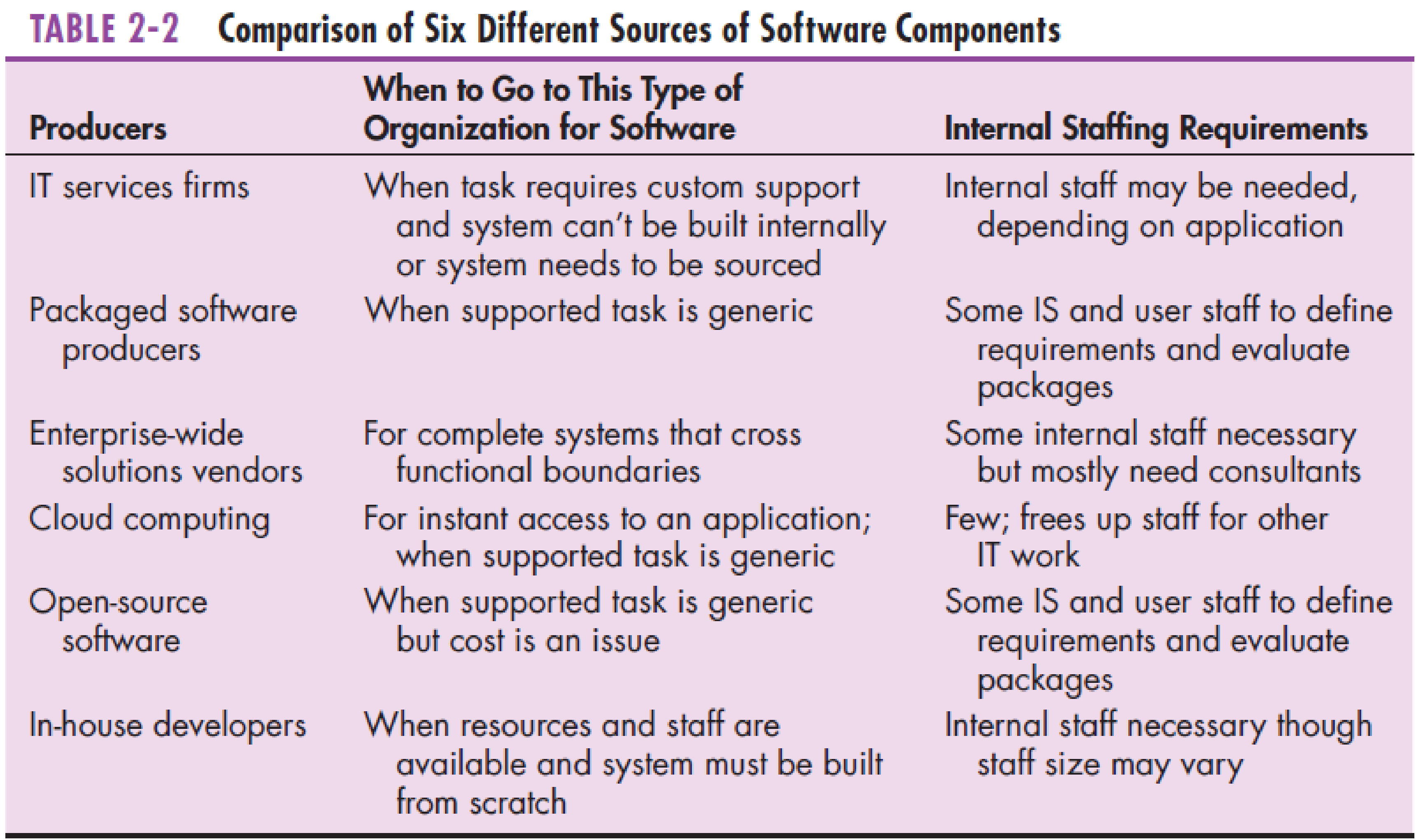


TABLE 2-2 Comparison of Six Different Sources of Software Components

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

Sources of Software (Cont.)

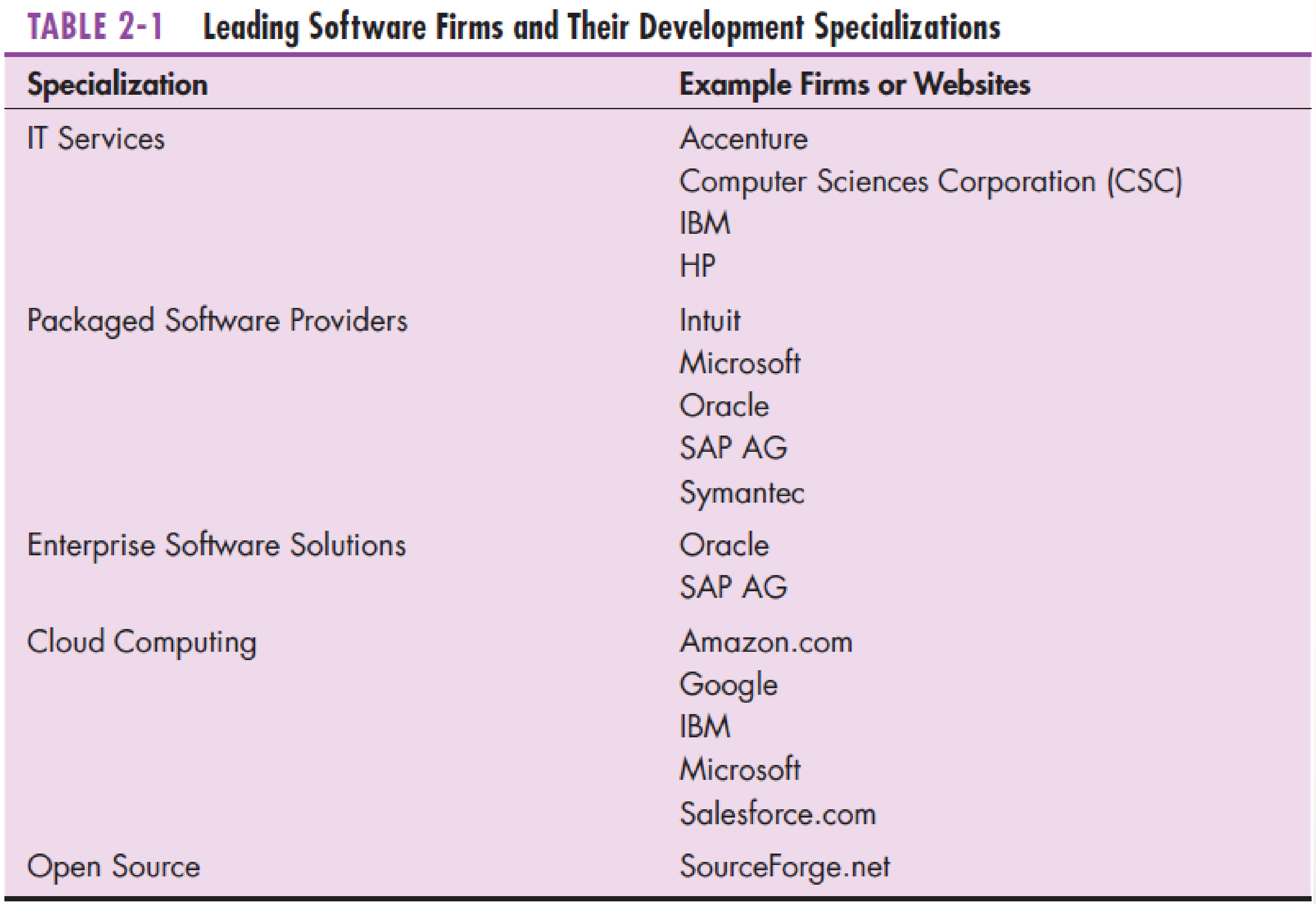


TABLE 2-1 Leading Software Firms and Their Development Specializations

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

Choosing Off-the-Shelf Software

- **Criteria:**
  - **Cost**- In-house versus purchase: 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
  - **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

Choosing Off-the-Shelf Software (Cont.)

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

- Collect information from vendor
  - Software Documentation
  - Technical Marketing literature
  - Questionnaire
- 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
- Software Test drive
- Feedback from other users
- Independent software testing services

Validating Purchased Software Information (Cont.)

- Analyst selects best candidates based on:
  - vendor bids
  - a variety of information sources
    - 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

REUSE

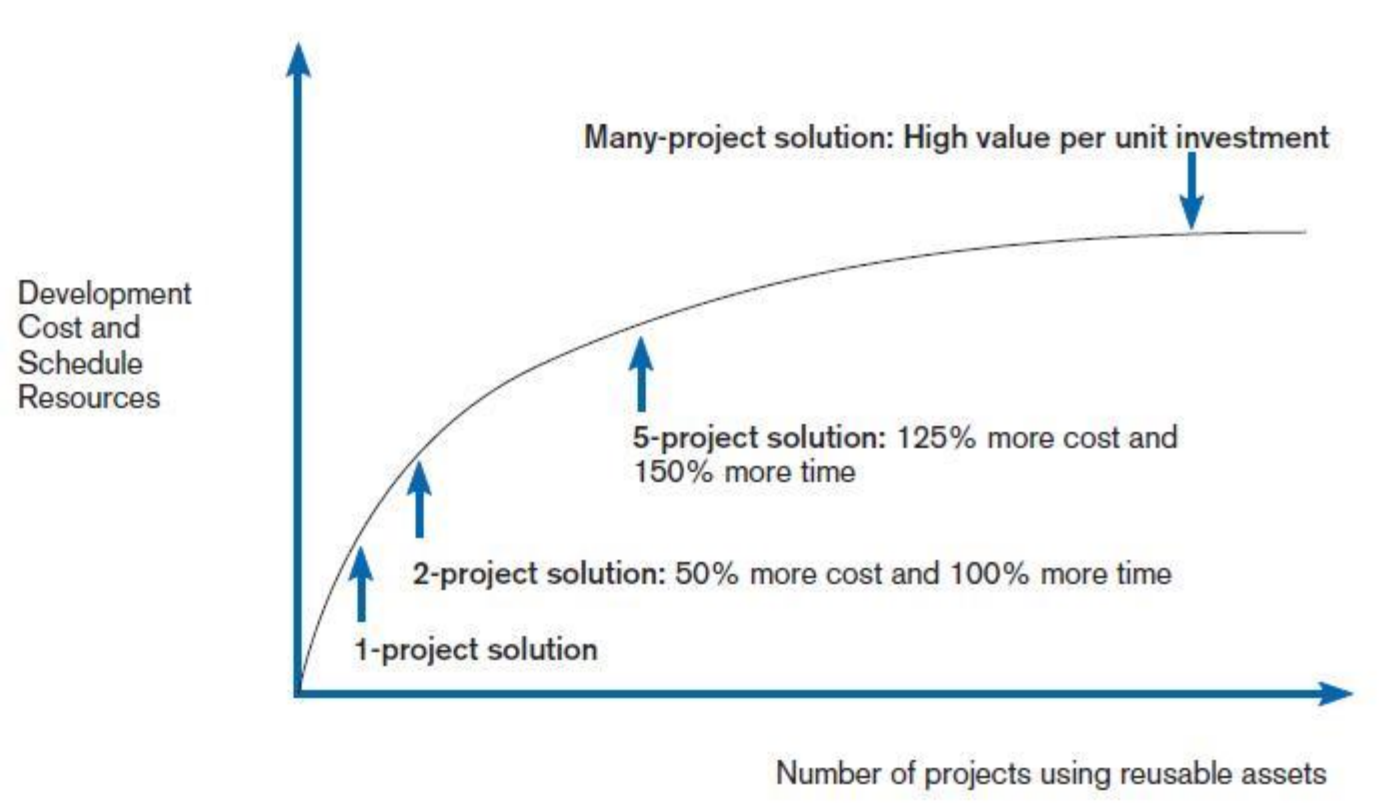
- ⦿ Use of previously written software resources in new applications
- ⦿ Most often applied to *object-oriented* and *component-based* development technologies.

- **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 by
  - Increases productivity
  - Reduces defect density
  - Reduces rework
- Reuse plan must be matched with companies strategic business goals
- 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



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.)

Three 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

- Four Approaches

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

(Griss 2003)

Approaches to Reuse (Cont.)

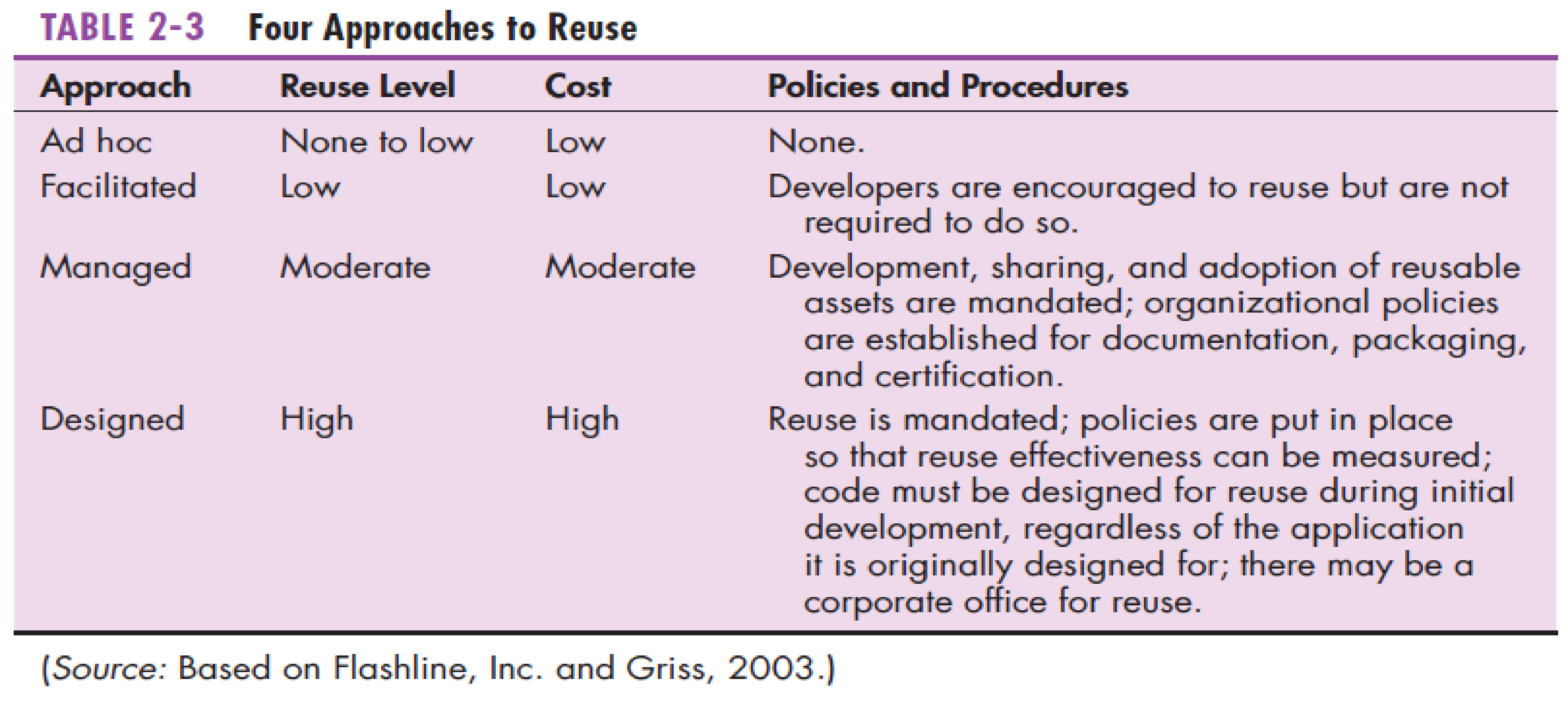


TABLE 2-3 Four Approaches to Reuse

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.