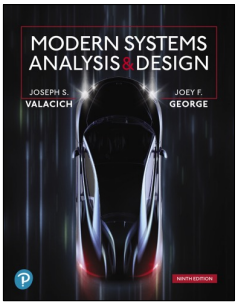
Modern Systems Analysis and Design

Ninth Edition



Chapter 2

The Origins of Software

Pearson

1

Learning Objectives

2.1 Explain outsourcing

2.2 Describe six different sources of software

2.3 Discuss how to evaluate off-the-shelf software

2.4 Explain reuse and its role in software development

Pearson

2

Introduction

- In the past software development for organizations was done mostly in-house and from scratch
- Today, there are many different sources of software
- Now focus is on where you can obtain the many pieces and components that you will combine into an application that you have been asked to create

Pearson

3

Outsourcing

2.1 Explain outsourcing

- **Outsourcing** – turning over responsibilities for some or all of an organization's information systems applications and operations to an outside firm
- Reasons for outsourcing:
  - Freeing up internal resources
  - Increasing the revenue potential of the organization
  - Reducing time to markets
  - Increasing process efficiencies
  - Outsourcing noncore activities

Pearson

4

Sources of Software

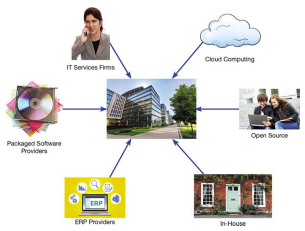
2.2 Describe six different sources of software

- Sources of software can be categorized into six major groups:
  - Information technology services firms
  - Packaged software producers
  - Enterprise solutions software
  - Cloud computing
  - Open-source software
  - In-house development

Pearson

5

Figure 2-1: Sources of Application Software



(**Sources:** Middle: Pixachi/Shutterstock, Clockwise starting with upper left:Kamira/Shutterstock; Amit John / Pearson India Education Services Pvt. Ltd; 1000 Words/Shutterstock; Aa Amie/Shutterstock; grgroup/Fotolia; Le Do / Shutterstock)

Pearson

6

Information Technology Service Firms

2.2 Describe six different sources of software

- IT service firms help companies develop customer information systems for internal use
- These firms employ skilled IT people
- Some examples of these companies are seen in the table on the next slide

Pearson

7

Table 2-1: Leading Software Firms and Their Development Specializations

Specialization	Example Firms or Websites
IT Services	Accenture Computer Sciences Corporation (CSC) IBM HPE
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 Solution	SourceForge.net

Pearson

8

Package Software Producers

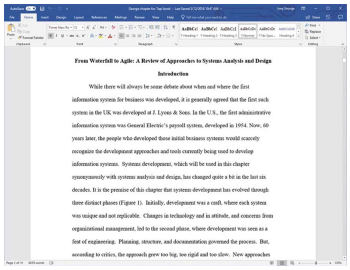
2.2 Describe six different sources of software

- Serve many market segments
- Prepackaged software is off-the-shelf and is not customizable
  - Examples include Quicken, QuickBooks, Microsoft Word, TurboTax
  - Meets 70% of an organization's needs
- An example of Microsoft Word is on the next slide

Pearson

9

Figure 2-2: Document Created in Microsoft Word



From Waterfall to Agile: A Review of Approaches to Systems Analysis and Design

Introduction

While there will always be some debate about when and where the first information system for business was developed, it is generally agreed that the first such system in the UK was developed at J. Lyons & Sons. In the U.S., the first administrative information system was General Electric's payroll system, developed in 1954. Now, 60 years later, the people who developed those initial business systems would scarcely recognize the development approaches and tools currently being used to develop information systems. Systems development, which will be used in this chapter synonymously with systems analysis and design, has changed quite a bit in the last six decades. It is the premise of this chapter that systems development has evolved through three distinct phases (Figure 1). Initially, development was a craft, where each system was unique and not replicable. Changes in technology and in attitude, and concerns from organizational management, led to the second phase, where development was seen as a feat of engineering. Planning, structure, and documentation governed the process. But, according to critics, the approach grew too big, too rigid and too slow. New approaches

(**Source:** Microsoft Corporation)

Pearson

10

Enterprise Solutions Software

2.2 Describe six different sources of software

- **Enterprise resource planning (ERP) systems** integrate individual traditional business functions into a series of modules so that a single transaction occurs seamlessly within a single information system rather than several separate systems
- Benefits include:
  - A single repository resulting in more consistent and accurate data
  - Less maintenance
- SAP AG is the leading vendor of ERP systems

Pearson

11

Cloud Computing (1 of 2)

2.2 Describe six different sources of software

- **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
- Users pay on a per-use system or license the software
- Global market, based on cloud computing, is estimated at $209.2 billion and estimated to grow to $383.3 billion by 2020

Pearson

12

Cloud Computing (2 of 2)

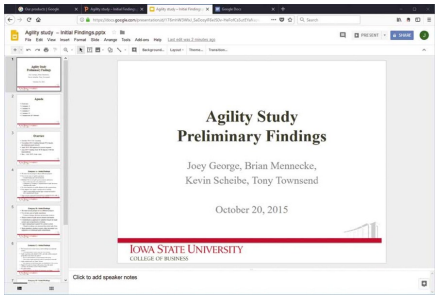
2.2 Describe six different sources of software

- Cloud computing examples include:
  - Google docs, Sheets, and Slides
    - Share, create docs, spreadsheets, presentations
  - SalesForce.com (Customer relationship management)
  - Allows software as a service (SaaS)
- Benefits include:
  - Freeing internal staff
  - Faster access to applications
  - Lower-cost to corporate-quality applications
- Concerns include security and reliability

Pearson

13

Figure 2-3: Presentation Edited in Google Slides



(**Source:** Reprinted by permission from Joey F. George)

Pearson

14

Open-Source Software

2.2 Describe six different sources of software

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

Pearson

15

In-House Development

2.2 Describe six different sources of software

- Involves using an organization's staff to create systems
- Can lead to more maintenance needs
- Not unusual to incorporate a hybrid of in-house and purchased components as seen in the table on the next slide

Pearson

16

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 customer support and system can't be build 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

Pearson

17

Choosing Off-the-Shelf Software (1 of 2)

2.3 Discuss how to evaluate off-the-shelf software

- Criteria to consider when choosing off-the shelf software:
  - **Cost** (compare developing in-house vs. purchasing or licensing the software package)
  - **Functionality** (the tasks that the software can perform and the mandatory, essential, and desired system features)
  - **Vendor Support** (can vendor provide support and how much it can provide)

Pearson

18

Choosing Off-the-Shelf Software (2 of 2)

2.3 Discuss how to evaluate off-the-shelf software

- Criteria to consider when choosing off-the shelf software:
  - **Flexibility** (the ease with which software is customized)
  - **Documentation** (understandable and up-to-date user's manual and technical documentation)

Pearson

19

Validating Purchased Software Information

2.3 Discuss how to evaluate off-the-shelf software

- **Request for Proposal (RFP) –** document provided to vendors asking them to propose hardware and system software that will meet the requirements of a new system
- If soliciting RFPs from more than one vendor be sure to:
  - Create a scoring value for each item requested
  - Calculate a total score for each vendor after RFPs are received
  - Select a vendor with a high score

Pearson

20

Reuse (1 of 2)

2.4 Explain reuse and its role in software development

- **Reuse** – the use of previously written software resources, especially objects and components, in new applications
- Reuse is commonly applied to two different 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

Pearson

21

Reuse (2 of 2)

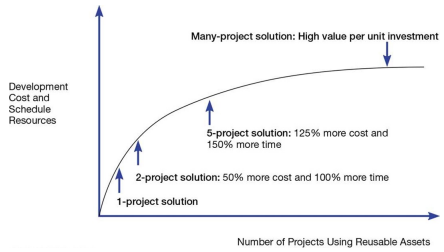
2.4 Explain reuse and its role in software development

- Reuse has become a standard part of corporate development
- Factors relating to successful reuse include:
  - Knowledge of the domain in which reuse is to occur
  - Customer support
  - Commitment and understanding of senior management
  - Sound organizational processes
  - Skilled and experienced developers

Pearson

22

Figure 2-4: 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)

Pearson

23

Three Basic Steps of Software Reuse

2.4 Explain reuse and its role in software development

1. Abstraction
   - Involves design of reusable software
2. Storage
   - Making software assets available for others to use
3. Recontextualization
   - Making the software understandable to developers

Grinter, 2001

Pearson

24

Approaches to Reuse

2.4 Explain reuse and its role in software development

- **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

Pearson

25

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	Developers, 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 into 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 Griss, 2003.)

Pearson

26

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

Pearson

27

Copyright

28