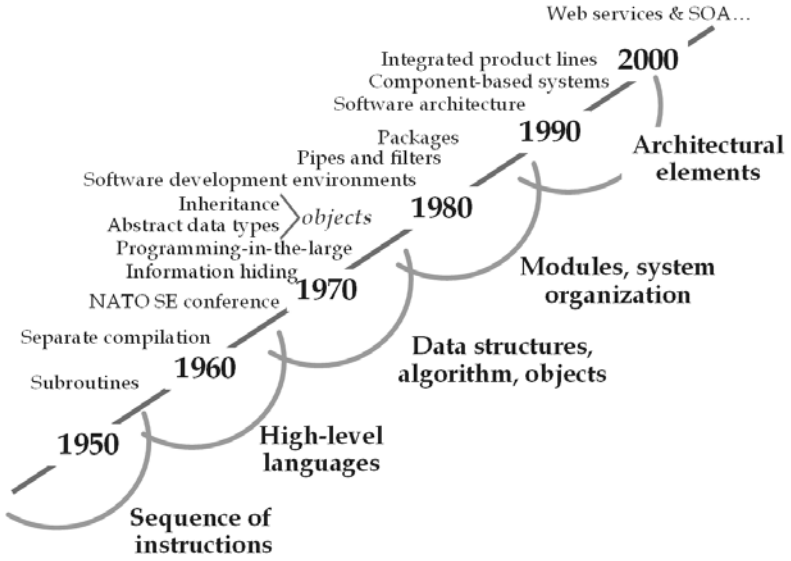
Technologies for Developing Systems

Chapter 9

Outline

- Introduction
- Foundations of Systems Development
  - Structured Development
  - Fourth-Generation Languages
  - Software Prototyping
  - Computer-Aided Software Engineering (CASE)
  - Object-Oriented Development
  - Client-Server Computing
- System Integration
  - ERP Systems
  - Middleware
- Inter-organizational System Development
- Internet-Based Systems
  - Application Servers
  - Java
  - Web Services

The Evolution of System Development



History of System Development

1970s	■Structured system development ■System development life cycle
1980s	■4GL (4th Generation Language) ■Automation of parts of development such as code generation
1990s	■Reliance increased on packages ◆Increased developer productivity ◆Decreased maintenance cost ■Business process reengineering led to growth on integrated enterprise systems ◆ERP (Enterprise Resource Planning) System
Late 1990s	■Sudden emergence of e-business and Internet based systems
2000s	■Faster systems development cycles ■Integrated enterprise systems

Business Applications Today

- Business process reengineering movement growth on integrated enterprise systems and adoption of enterprise resource planning systems (ERP)
- Virtually every application is a network application
  - The network is becoming the system
- Web-based applications were the first generation of Internet-centric computing
  - Web Services is touted as the second
- In addition, the increasing interconnectedness of supply chains is leading companies to build inter-organizational systems
  - Far more complex undertaking than any single-company systems

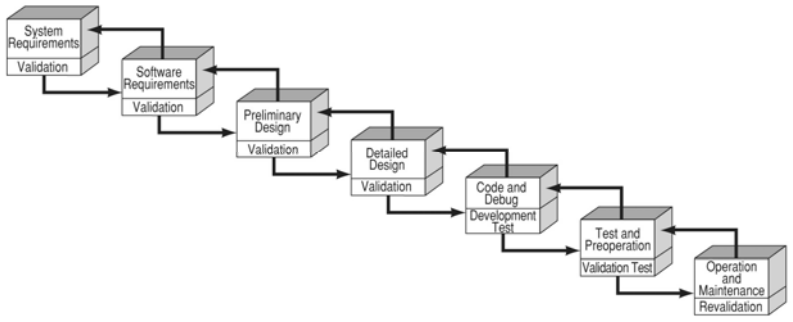
Characteristics of Structured System Development Approach

- Hand coding in third generation language, e.g. COBOL
- A Structured Programming development methodology
- Automated Project management system
- A database management system
- A mix of online and batch applications in the same system
- Development of mostly mainframe applications
- Programming by professional programmers only
- Various automated, but not well integrated s/w tools
- A well-defined sign-off process for system delivery
- User participation mainly in require definition and installation phases

The Waterfall Development Life Cycle

- Waterfall Approach is a way to view the system development process as a series of steps that, when diagrammed, appear as a waterfall.

FIGURE 9-1 The "Waterfall" Development Life Cycle



Source: Barry Boehm, *Software Engineering Economics* (Upper Saddle River, NJ: Prentice Hall, 1981).

The Spiral Model

- Spiral Diagram is a way of viewing the application development process as a spiral, as opposed to a waterfall

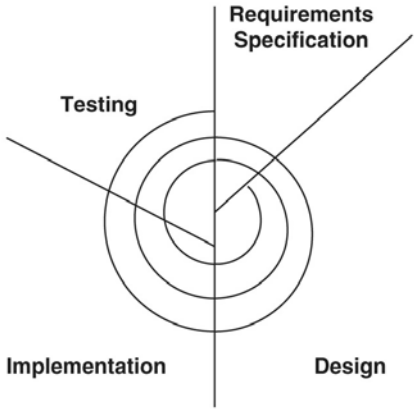


FIGURE 9-2 The Spiral Model of Systems Development

Source: Barry Boehm, "A Spiral Model of Software Development and Enhancement," *IEEE Computer*, Vol. 21, No. 5, May 1988, pp. 61–72.

Structured Development

- Structured development methodologies accompanied this system development life cycle in an attempt to manage the complexities of systems design and development
  - **More Discipline:** Established standards for process and documentation to increase productivity and developers' ability to deal with complexity
  - **Higher Reliability and Fewer Errors:** To catch errors as early as possible through inspection
  - **More Efficient Use of Resources:** Thorough project management approaches resulted in:
    - Cost savings
    - Increased productivity
    - Better allocation of human resources
    - Reduced the tendency for system development project overruns

Fourth-Generation Languages

- Fourth Generation Language (4GL) is a computer programming language used by end users, as opposed to COBOL (a third generation language), or Assembler (a second generation language), or programming via plug boards (first generation programming).
- 4GL specifies the purpose without details on procedures
  - E.g. SQL (SELECT NAME FROM STUDENT)

Features and Functions of 4GL

FIGURE 9-3 Features and Functions of Fourth-Generation Languages

- Database management system (DBMS)
- Data dictionary
- Nonprocedural language
- Interactive query facilities
- Report generator
- Selection and sorting
- Screen formatter
- Word processor and text editor
- Graphics
- Data analysis and modeling tools
- Library of macros
- Programming interface
- Reusable code
- Software development library
- Backup and recovery
- Security and privacy safeguards
- Links to other DBMSs

Software Prototyping

- A quickly created system aiming to test out assumptions
  - User requirements
  - Application design
  - Program logic
- Prototyping corresponds to the iterative process of application development
  - Starting with a simple prototype
  - New requirements added in and new refinements incorporated
  - Each version sees some incremental additions

Computer-Aided Software Engineering (CASE)

- The advent of Computer Aided Software Engineering (CASE) occurred to automate structured techniques and reduce tediousness of the 1970s structured programming and analysis techniques.
- A late ’80s use of CASE is Timeboxing
  - Timeboxing is a technique that uses CASE to guarantee delivery of a system within 120 days
- Today, IS departments that aim for speed over complexity are turning to a development technique like:
  - Rapid Application Development (RAD)

Read Case Example P.376-378

Components of CASE

- **Information Repository:** stores and organizes all information needed to create, modify, and develop software system
- **Front-end Tools:** used in all phases that lead up to coding
- **Back-end Tools:** used to automatically generate source code
- **Development Workstation:** the more powerful the better

Object-Oriented Development

- Object-oriented (OO) Development was a revolutionary change in the late 1980s – develop objects that can be reused
- It allowed point-and-click programming of graphical user interfaces
- It is not so much a coding technique as a code-packaging technique. The objects are:
  - Receives request (message)
  - Chooses and executes operation, then
  - Returns the results to the requester
- It is very modular, so a change in one part of a system need not affect the other parts

Client-Server Computing and Web Based Development

- Two major developments in 1990s
  - Client-server systems
    - Work split between a client and a server
    - Far more flexibility than mainframe-based systems
    - Desktop: graphics, animation, video
    - Servers: production updating
  - Web-based or network centric development
- Underlying these two (continuing) trends is the increasing use of packages and system integration
  - As much as possible, companies prefer to buy a package rather than build an application in-house
- To develop large applications, they integrate various hardware and software components

Read Case Example P.379-380

Systems Integration

- System Integration is the current process of building systems by piecing together hardware components, software packages, database engines, and network products from numerous vendors into a single, cohesive system.
- Integration is by far the biggest software problems CIO face
  - Integrating legacy systems from various eras and technologies
  - Integration was also a major problem for IT vendors
    - E.g. IBM with its various product lines

Three Traditional Integration Approaches

- Database Management system (DBMS)
  - A data-centered approach, allowing applications to share data stored in a single or distributed database
- Enterprise Resource Planning (ERP)
  - An application-centered approach, all applications come from a single vendor and are specifically designed to communicate with each other
- Middleware
  - A third-party approach, applications communicate with each other through a third-party translation software

Enterprise Resource Planning Systems (ERP)

- ERP system aims to integrate corporate systems by providing a single set of applications from a single vendor operating with a single database
  - The goal is providing the means to integrate business departments and functions across an organization
- History of ERP contains both successes and failures, many of which have been especially notable:
  - Average cost overrun – 179%
  - Average schedule overrun 230%
  - Functionally 59% below expectations
  - 10% projects completed on time & in budget
  - 35% projects cancelled
- Common to hold systems large size and complexity responsible as well as:
  - Too much attention to technical aspects
  - Not enough to organizational aspects

Read Case Example P.383-384

Middleware

- Most organizations have a wide range of applications
  - New and old
  - From a variety of vendors
  - Running on numerous platforms
- Employ a class of products known as middleware
  - Software that works between and connects applications allowing them to share data
  - Needed as wide range of applications used and run on numerous platforms
  - Simplifies development by acting as the glue that binds the components together

Type of Middleware Used in Client-Server Applications

FIGURE 9-4 Types of Middleware Used in Client-Server Applications

INTERAPPLICATION COMMUNICATIONS FACILITIES: LINK COMPONENTS

- Application programming interfaces (APIs): provide a standard way of interfacing
- Remote procedure call (RPCs): enable a dialogue between two geographically dispersed applications
- Object request brokers (ORBs): allow applications or utilities to interwork in standard ways
- Message-oriented middleware (MOM): uses asynchronous message passing for interapplication communications

TRANSACTION MANAGERS: HANDLE TRANSACTIONS ACROSS MULTIPLE PLATFORMS

- Standard query languages (SQLs): standardize the way in which databases are accessed
- TP monitors (CICS, for example): monitor online transaction processing with a database
- Two-phase commit: a protective mechanism for transactions that fail to complete successfully

UTILITIES: PROVIDE GENERAL SERVICES

- Directory services: resource allocation
- Time services: timing
- Security services: encryption, and so on
- Software distribution: including configuration control

Source: Roger Woolfe, "Managing the Move to Client-Server," Wentworth Research Program (now part of Gartner EXP, Stamford, CT), 1995.

Enterprise Application Integration (EAI)

- One type of middleware that has gained popularity is Enterprise Application Integration (EAI):
  - Typically use a message broker to transfer data between applications
  - Add a new level of functionality that distinguishes them
  - Allow users to define business processes and make data integration subject to rules that govern those processes
    - E.g. a rule might state that data can only move from purchasing to accounts receivable when 'X' has signed off on the purchase

Read Case Example P.385-386

Interorganizational System Development

- **Interorganizational Systems (IOS)** is a systems that require at least two parties with different objectives to collaborate on the development and operation of a joint computer-based system.
- Business ecosystems became one of the main business trends
  - SCM integrates supply chains
  - Development of such inter-organizational systems requires teams from the different organizations to work together
- Another type of inter-organizational system is a platform, which provides the infrastructure for the operation of a business ecosystem, a region, or an industry

Read Case Example P.388-391

Internet-based Systems

- Internet users have become so sophisticated that Internet-based systems must be:
  - Scalable
  - Reliable
  - Integrated both internally and externally with systems of customers or business partners
- To do this companies recognize they must negotiate language differences
  - E.g. a system may have to port old COBOL applications to Java, reconcile interface discrepancies and interface with back-end legacy applications, often without documentation or past experience with those systems
    - Tools are available to help
    - Open systems etc

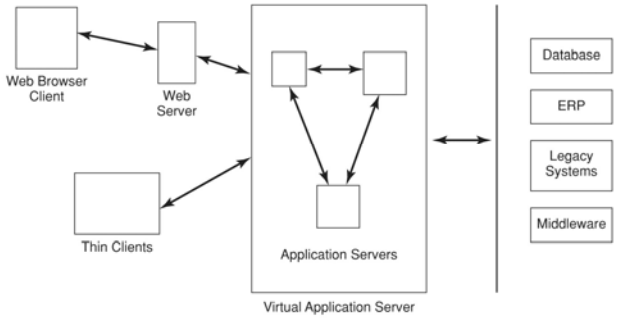
Application Servers

- Originally conceived as a piece of middleware to link a Web server to applications on other company systems
  - The application server has grown into a framework for developing Internet-based applications
- The goal of the application server:
  - automate
  - manage technical tasks in the development and running of Internet based applications
- The result:
  - Developers can focus

Basic Application Server Architecture

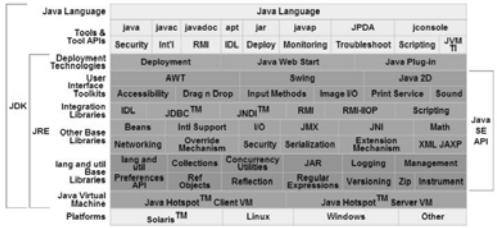
- The virtual server takes requests from clients and Web servers (on the left), runs the necessary business logic & provides connectivity to the entire range of back-end systems (on the right)

FIGURE 9-6 An Application Server Architecture



Java Development Platform

- Java has been in many cases the starting point for the development of Internet-based systems with an open system architecture.
  - Originally developed to provide applets that run on Web clients
    - Evolved into a full programming language
    - The goal is become the platform for independent language that could run on any system



Java Development Platform

- Major components in Java server-side platform
  - Enterprise Java Beans (EJB)
    - Preconfigured pieces of code that IS staff no longer have to build from scratch
  - Java 2 Enterprise Edition (J2EE)
    - Defines a standard for developing Internet-based enterprise applications
  - Provide an alternative to building online business systems from scratch or buying packaged online business systems because of their multi-vendor platform capability and pre-built, reusable components
- Microsoft competes with its own versions:
  - Component Object Model (COM)
  - Distributed Component Object Model (DCOM)

Web Services

- The vision of Web Services is that modules of code can be assembled into services, which, in turn, can be linked to create a business process at the moment it is needed and run across enterprises, computing platforms, and data models
- There are two development modes:
  - One is to wrap an XML wrapper around an existing piece of code that performs a specific function
    - Exposes it
    - Then give it a Internet address and let others use it for a fee
  - Second way to build a Web Service use one someone else has already exposed

Read Case Example P.394-398