Chapter 1
The Systems Development Environment

Course Content
- Fundamental of System Development
- System Development Life Cycle
  - System Planning
  - System Analysis
  - System Design
  - System Implementation and Maintenance
- Standard
- Sustainable System Development

IS: Enterprise Applications

Kaizen (for SME)
++ Overall
+ Quality Control
1. Introduction

Information Systems Analysis and Design
- Complex organizational process
- Used to develop and maintain computer-based information systems
- Used by a team of business and systems professionals
Modern Approach to SA&D

- 1950s: efficient automation of existing processes
- 1960s: 3GL, faster and reliable computers
- 1970s: system development, using engineering discipline
- 1980s: 4GL, CASE tools, object oriented methods
- 1990s: system integration, GUI applications, client/server platforms, Internet
- The new century: Web application development, wireless PDAs, component-based applications, ubiquitous technology

Modern Approach to SA&D (Cont.)

- Application Software
  - Computer software designed to support organizational functions or processes
- Systems Analyst
  - Organizational role most responsible for analysis and design of information systems

Systems Analyst as a Facilitator

Knowledge and Skills of Systems Analyst

- Knowledge
  - Information technology
  - Computer programming experience and expertise
  - General business knowledge
- Skills
  - Problem-solving
  - Interpersonal communication
  - Interpersonal relations
  - Flexibility and adaptability
  - Character and ethics
  - Systems analysis and design
2. Types of Information Systems and Systems Development

- **Transaction Processing Systems (TPS)**
  - Automate handling of data about business activities (transactions)
  - Process orientation

- **Management Information Systems (MIS)**
  - Converts raw data from transaction processing system into meaningful form
  - Data orientation

---

Types of Information Systems

- **Decision Support Systems (DSS)**
  - Designed to help decision makers
  - Provides interactive environment for decision making
  - Involves data warehouses, executive information systems (EIS)
  - Database, model base, user dialogue

- **Expert Systems**

---

Summary of Information Systems Types

<table>
<thead>
<tr>
<th>IS Type</th>
<th>IS Characteristics</th>
<th>Systems Development Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction processing system</td>
<td>High-volume, data capture focus; goal is efficiency of data movement and processing and interfacing different TPSs</td>
<td>Process orientation; concern with capturing, validating, and storing data and with moving data between each required step</td>
</tr>
<tr>
<td>Management information system</td>
<td>Draws on diverse yet predictable data resources to aggregate and summarize data; may involve forecasting future data from historical trends and business knowledge</td>
<td>Data orientation; concern with understanding relationships among data so data can be accessed and summarized in a variety of ways; builds a model of data that supports a variety of uses</td>
</tr>
<tr>
<td>Decision support system</td>
<td>Provides guidance in identifying problems, finding and evaluating alternative solutions, and selecting or comparing alternatives; potentially involves groups of decision makers; often involves semi-structured problems and the need to access data at different levels of detail</td>
<td>Data and decision logic orientations; design of user dialogue; group communication may also be key; and access to unpredictable data may be necessary; nature of systems requires iterative development and almost constant updating</td>
</tr>
</tbody>
</table>
3. Developing Information Systems

- **System Development Methodology** is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.

**Systems Development Life Cycle (SDLC)**

- Traditional methodology used to develop, maintain, and replace information systems.
- Phases in SDLC:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Maintenance

**Standard and Evolutionary Views of SDLC**

- **Planning** – an organization’s total information system needs are identified, analyzed, prioritized, and arranged.
- **Analysis** – system requirements are studied and structured → **logical system specification** (model).
- **Design** – a description of the recommended solution is converted into **physical system specifications**.
Systems Development Life Cycle (Cont.)

- **Logical design** – all functional features of the system chosen for development in analysis are described independently of any computer platform.
- **Physical design** – the logical specifications of the system from logical design are transformed into the technology-specific details from which all programming and system construction can be accomplished.

Systems Development Life Cycle (Cont.)

- **Implementation** – the information system is coded, tested, installed and supported in the organization.
- **Maintenance** – an information system is systematically repaired and improved.

### Table 1-1 Products of SDLC Phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Products, Outputs, or Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems Detailed steps, or work plan, for project Specification of system scope and planning and high-level system requirements or features Assignment of team members and other resources System justification or business case</td>
</tr>
<tr>
<td>Analysis</td>
<td>Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system Explanation of alternative systems and justification for chosen alternative</td>
</tr>
<tr>
<td>Design</td>
<td>Functional, detailed specifications of all system elements (data, processes, inputs, and outputs) Technical, detailed specifications of all system elements (programs, files, network, system software, etc.) Acquisition plan for new technology</td>
</tr>
<tr>
<td>Implementation</td>
<td>Code, documentation, training procedures, and support capabilities</td>
</tr>
<tr>
<td>Maintenance</td>
<td>New versions or releases of software with associated updates to documentation, training, and support</td>
</tr>
</tbody>
</table>
4. Heart of the Systems Development Process

Current practice combines analysis, design, and implementation into a single iterative and parallel process of activities.

Problems with Waterfall Approach *

- System requirements “locked in” after being determined (can’t change)
- Limited user involvement (only in requirements phase)
- Too much focus on milestone deadlines of SDLC phases to the detriment of sound development practices
4.1 Different Approaches to Improving Development

- Prototyping
- Computer-Aided Software Engineering (CASE) Tools
- Joint Application Design (JAD)
- Rapid Application Development (RAD)
- Agile Methodologies
- eXtreme Programming

a) Prototyping *

- Iterative development process
- Requirements quickly converted to a working system
- System is continually revised
- Close collaboration between users and analysts

Prototyping (Cont.)

**Figure 1-11** The prototyping methodology

1. Identify Problem
2. Develop Prototype
3. Implement and Use Prototype
4. Revise and Enhance Prototype
5. Convert to Operational System

b) Computer-Aided Software Engineering (CASE) Tools

- **Diagramming tools** enable graphical representation
- Computer displays and report generators help prototype how systems “look and feel”
- **Analysis tools** automatically check for consistency in diagrams, forms, and reports

CASE Tools (Cont.)

- **Central repository** for integrated storage of diagrams, reports, and project management specifications
- **Documentation generators** standardize technical and user documentation
- **Code generators** enable automatic generation of programs and database code directly from design documents, diagrams, forms, and reports

---

c) Joint Application Design (JAD)

- Structured process involving users, analysts, and managers
- Several-day intensive workgroup sessions
- **Purpose** to specify or review system requirements
d) Rapid Application Development (RAD)
- Methodology to radically decrease design and implementation time
- Involves: extensive user involvement, prototyping, JAD sessions, integrated CASE tools, and code generators

Service-Oriented Architecture (SOA)
- An approach to systems development based on building complete systems through assembling software components, each of which model generic business functions

Service-Oriented Architecture (SOA) (Cont.)
- Motivated by recognition of software development as fluid, unpredictable, and dynamic
- Three key principles
  - Adaptive rather than predictive
  - Emphasize people rather than roles
  - Self-adaptive processes

e) Agile Methodology
- Involves: extensive user involvement, prototyping, JAD sessions, integrated CASE tools, and code generators

FIGURE 1-11
RAD life cycle

FIGURE 1-12
Illustration of a service, a credit check, used by applications and other services
The Agile Methodologies group argues that software development methodologies adapted from engineering generally do not fit with real-world software development.

When to use Agile Methodologies

- If your project involves:
  - Unpredictable or dynamic requirements
  - Responsible and motivated developers
  - Customers who understand the process and will get involved

---

TABLE 1-4 Five Critical Factors That Distinguish Agile and Traditional Approaches to Systems Development

<table>
<thead>
<tr>
<th>Factor</th>
<th>Agile Methods</th>
<th>Traditional Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Well matched to small products and teams. Reliance on tacit knowledge limits scalability.</td>
<td>Methods evolved to handle large products and teams. Hard to tailor down to small projects.</td>
</tr>
<tr>
<td>Criticality</td>
<td>Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.</td>
<td>Methods evolved to handle highly critical products. Hard to tailor down to products that are not critical.</td>
</tr>
<tr>
<td>Dynamism</td>
<td>Simple design and continuous refactoring are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments.</td>
<td>Detailed plans and Big Design Up Front excellent for highly stable environments but a source of expensive rework for highly dynamic environments.</td>
</tr>
<tr>
<td>Personnel</td>
<td>Requires continuous presence of a critical mass of scarce experts. Risky to use no-agile people.</td>
<td>Needs a critical mass of scarce experts during project definition but can work with fewer later in the project, unless the environment is highly dynamic.</td>
</tr>
<tr>
<td>Culture</td>
<td>Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom (thriving on chaos).</td>
<td>Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear practices and procedures (thriving on order).</td>
</tr>
</tbody>
</table>

(Source: Boehm and Turner, 2004. Used by permission.)

---

f) eXtreme Programming

- Short, incremental development cycles
- Automated tests
- Two-person programming teams
- Coding and testing operate together
- Advantages:
  - Communication between developers
  - High level of productivity
  - High-quality code
5. Object-Oriented Analysis and Design (OOAD)

- Based on objects rather than data or processes
- Object: a structure encapsulating attributes and behaviors of a real-world entity
- Object class: a logical grouping of objects sharing the same attributes and behaviors
- Inheritance: hierarchical arrangement of classes enable subclasses to inherit properties of superclass

Rational Unified Process (RUP)

- An object-oriented systems development methodology
- RUP establishes 4 phases of development: inception, elaboration, construction, and transition
- Each phase is organized into a number of separate iterations

Summary

- Information systems analysis and design (SAD)
- Different types of information systems
- Information Systems Development Life Cycle (SDLC)
- Rapid Application Development (RAD), Prototyping, Joint Application Development (JAD), and Computer Aided Software Engineering (CASE)
- Agile methodologies and eXtreme programming
- Object Oriented Analysis and Design and the Rational Unified Process (RUP)

Questions and Answers

Further Read


Teamwork Assignment

- Topic
- Form a team of 10 students:
  - No, ID, name, email addresses (sort by ID)
  - Mark '*' for a leader & '***' for an auditor after ID
- Email to: wichian@sit.kmutt.ac.th
  cc: all team members (to build a mailing list)
- Subject: SA-N# (xxx) or SA-W# (xxx) where # is a team no. & ‘xxx’ is week-No(1..15)
- Submit
  - Date: (next week) one day before a class / Time: 1.00 p.m.

Quiz 1

1. The ________________ is the traditional methodology used to develop, maintain, and replace information systems.
2. The first phase in the systems development life cycle (SDLC) is ________.
3. ____________________________ is a structured process in which users, managers, and analysts work together for several days in a series of intensive meetings to specify or review system requirements.
4. The output from the analysis phase is the ________________________.
Exercise

1 List & explain some of the problems with the traditional waterfall SDLC.
2 Explain how prototype might be used as part of SDLC.

Enemy

- Inadequate & unstable requirements
- Inadequate customer communication
- Poor team communication
- Unnecessary complexity
- Ineffective team behavior

(0O project management using UML)