

Alternative Systems-Building Methods

1. Traditional systems lifecycle
2. Prototyping
3. End-user development
4. Application software packages
5. Outsourcing

Traditional systems lifecycle

Oldest method for building information systems

Traditional systems lifecycle Follows “waterfall” approach [T]

Prototyping

Building experimental system rapidly and inexpensively for end users to evaluate

Steps in prototyping

1. Identify user requirements
2. Develop initial prototype
3. Use prototype
4. Revise and enhance prototype

Advantage

Useful if some uncertainty in requirements

Often used for end-user interface design

More likely to fulfill end-user requirements

Disadvantage

May gloss over essential steps

May not accommodate large quantities of data or large number of users

End-user development

Uses fourth-generation languages to allow end-users to develop systems with little or no help from technical specialists

Less procedural than conventional programming languages [T]

Advantage

More rapid completion of project

High-level of user satisfaction

Disadvantages

Not designed for processing-intensive applications

Loss of control over data

Application software packages Save time and money[T]

Outsourcing

TYPES:

Cloud and SaaS providers

- Subscribing companies use software and computer hardware provided by vendors.

External vendors

- Hired to design, create software.
- Domestic outsourcing
 - a. Driven by firms need for additional skills, resources, assets.
- Offshore outsourcing
 - a. Driven by cost-savings.

Advantage

Allows organization flexibility in IT needs.

Disadvantages

Hidden costs

Rapid application development (RAD)

Process of creating workable systems in a very short period of time

RAD Utilizes techniques such as:

1. Visual programming
2. Iterative prototyping
3. Automation of program code generation
4. Close teamwork

Joint application design (JAD)

Used to accelerate generation of information requirements and to develop initial systems design.

Agile development

Focuses on rapid delivery of working software by breaking large project into several small sub-projects.

Component-based development

Groups of objects that provide software for common functions.