សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

System Acquisition and Development

MIS

Chea Daly

Buy versus Build

- Organizations continue to spend considerable time and resources developing and acquiring software, including BI and analytics; e-commerce, enterprise level functions, and mobile apps.
- Organizations can obtain software using one of two basic approaches: **buy** or **build**.

Buy versus Build

- Buying off-the-shelf software:
  - Less risky and leads to quicker deployment;
  - however, maintenance and support costs may become expensive with this approach, and the software may not be an exact match to the needs and work processes of the organization.

Buy versus Build

- Building custom software:
  - Can provide a better match to the current work processes and provide a potential competitive advantage;
  - However, software development can be extremely costly, and it can take months or even years to develop custom software.

System Investigation

- **System investigation** is the initial phase in the development of a new or modified business information system whose purpose is to gain a clear understanding of the specifics of the problem to solve or the opportunity to address.

System Investigation

- What is the scope of the problem?
- Who is affected and how?
- Is the problem worth addressing? Given limited resources—people and money.
- What are the potential costs, both the one-time initial costs and recurring costs?
- What risks are associated with the project?
- If successful, what benefits will the system provide?

Feasibility Analysis

- Assesses:
  - Technical feasibility
  - Economic feasibility
  - Legal feasibility
  - Operational feasibility
  - Schedule feasibility

Participants in Systems Development

- **Development team**:
  - Determines objectives of the information system.
  - Delivers system that meets objectives.
- **Project manager**:
  - Responsible for coordinating all people and resources needed to complete a project on time.

Participants in Systems Development

- **Stakeholders**:
  - People who ultimately benefit from project.
- **Users**:
  - People who will interact with the system regularly.
- **Programmers**:
  - Responsible for modifying or developing programs to satisfy user requirements.

Participants in Systems Development

- **Systems analysts**:
  - Professional who specializes in analyzing and designing systems.
  - An important role in development team and is often the only person who see the system in its totally.

Systems Development Life Cycles (SDLC)

- Common systems development life cycles:
  - Traditional
  - Water Fall
  - Prototyping
  - Incremental Development
  - …

Stages of The Traditional SDLC

- Investigation
- Analysis
- Design
- Implementation
- Maintenance and review

Traditional SDLC Stages

- **Systems investigation**
  - Identifies problems and opportunities.
- **Systems analysis**
  - Studies existing systems and work processes to identify strengths, weaknesses, and opportunities for improvement.
- **Systems design**
  - Design the architecture, components, and interfaces for a system so that it meets the end-user requirements.

Traditional SDLC Stages

- **Systems implementation**
  - Define how the information system should be built.
- **Systems maintenance and review**:
  - Modify and enhance the system so that it continues to meet changing business needs.
  - Analyze systems to make sure that they are operating as intended.

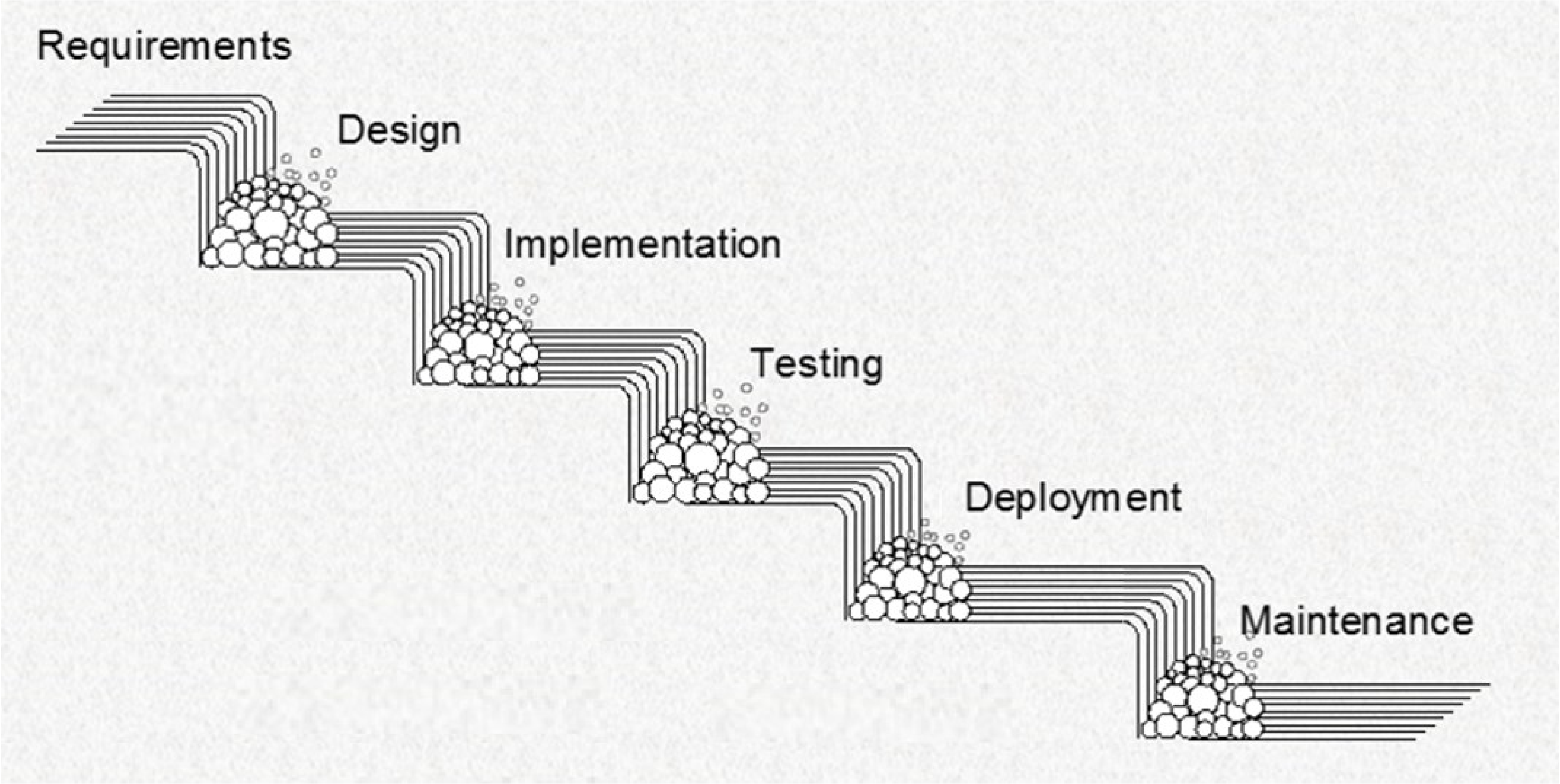
Systems Maintenance

- Systems maintenance:
  - Checking, changing, and enhancing the system to make it more useful in achieving user and organizational goals
- Maintenance team:
  - Responsible for modifying, fixing, and updating existing software.

Reasons for Maintenance

- Some reasons for program maintenance:
  - Changes in business processes
  - New requests from stakeholders, users, and managers
  - Bugs or errors in program
  - Technical and hardware problems
  - Corporate mergers and acquisitions
  - Government regulations
  - Changes in the operating system or hardware on which the application runs

Waterfall Model



Waterfall Model

- The waterfall system development process is a sequential, multistage system development process in which work on the next stage cannot begin until the results of the current stage are reviewed and approved or modified as necessary.

Waterfall Model Stages

- **Requirement Analysis**: breaking a complex topic into smaller parts.
- **Design**: a process of problem-solving and planning for a software solution.
- **Implementation**: The process of writing the source code.
- **Testing**: is conducted to get information about the quality of the software under test.

Waterfall Model Stages

- **Deployment**: When the system meets approval and is ready to “go live”, it is time to get it deployed.
- **Maintenance**: Systems evolve after deployment, and they will be continuously improved and changed as needed.

Waterfall Model Advantages

- Simple and easy to understand
- Easy to explain to the user
- Phases are processed and completed one at a time. Phases do not overlap. Which allows maximum management control.

Waterfall Model Disadvantages

- In practice, not all the requirements can be received at once to properly design the system.
- High amounts of risk and uncertainty.

Relationship between Timing of Errors and Costs

- The later in the SDLC an error is detected, the more expensive it is to correct
  - Reason for mounting costs:
    - Previous phases must be reworked
    - More people are affected

Relationship between Timing of Errors and Costs

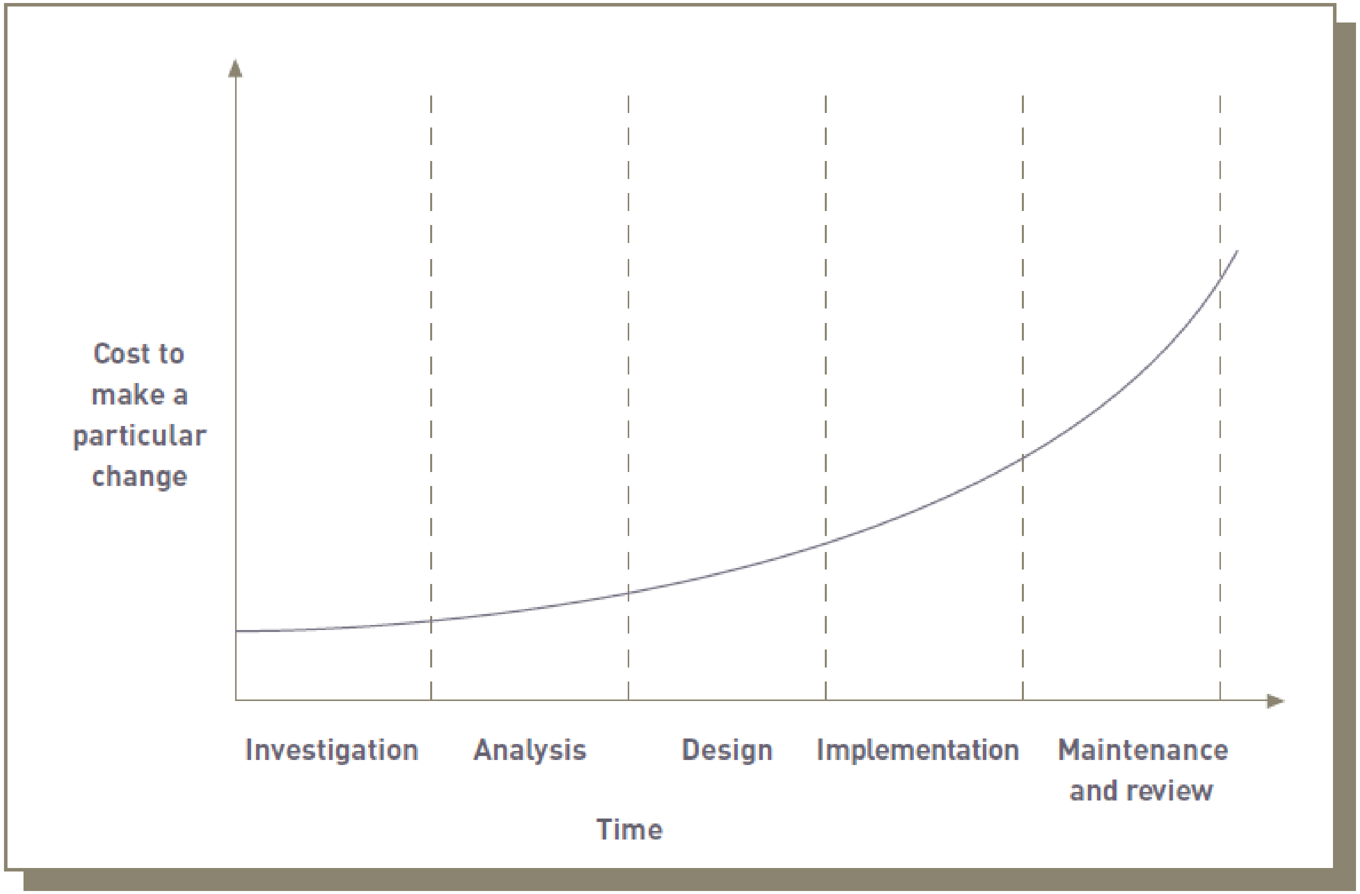


Figure 12.5

Relationship between Timing of Errors and Costs

The later that system changes are made in the SDLC, the more expensive these changes become.

Waterfall Model Overall

- The waterfall approach to system development allows for a high degree of management control.
- It is for this reason that this 50-year-old approach is frequently followed when an organization contracts with another to build its information system, even though the other models can be faster and can lead to higher quality results.

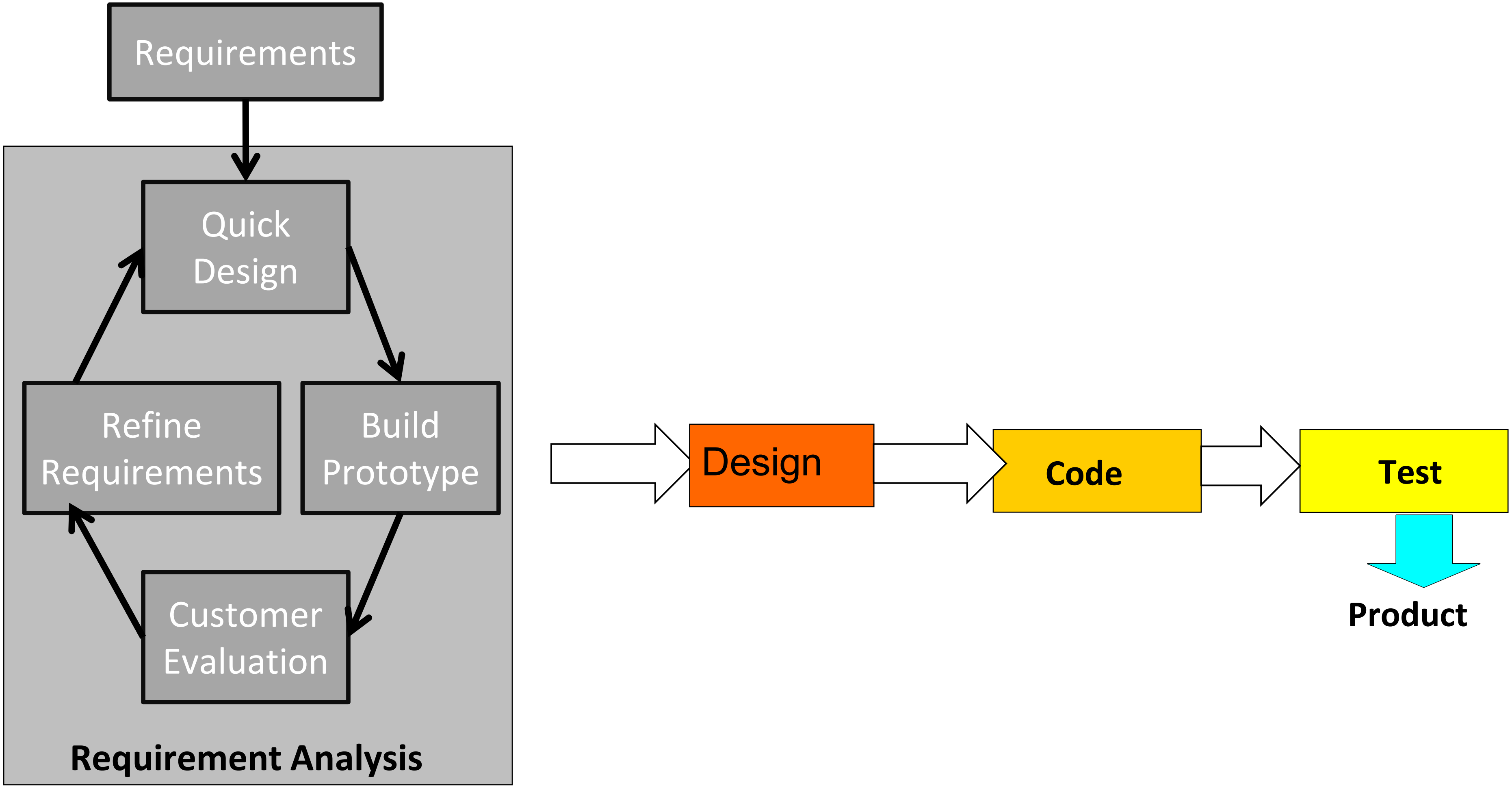
Waterfall Model Overall

- The major problem with this approach is that users do not interact with the solution until the testing phase when the system is nearly complete.
- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the Requirement stage.

Waterfall Model Overall

- Waterfall Model works well for smaller projects where requirements are very well understood.
- Poor model for long and ongoing projects.
- Not a right process model in situations where requirements change quickly or at a high risk of changing.

Prototyping Model



Prototyping Model Overall

- Prototyping Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved.
- It also creates base to produce the final system or software.
- It works best in scenarios where the project’s requirements are not known in detail.

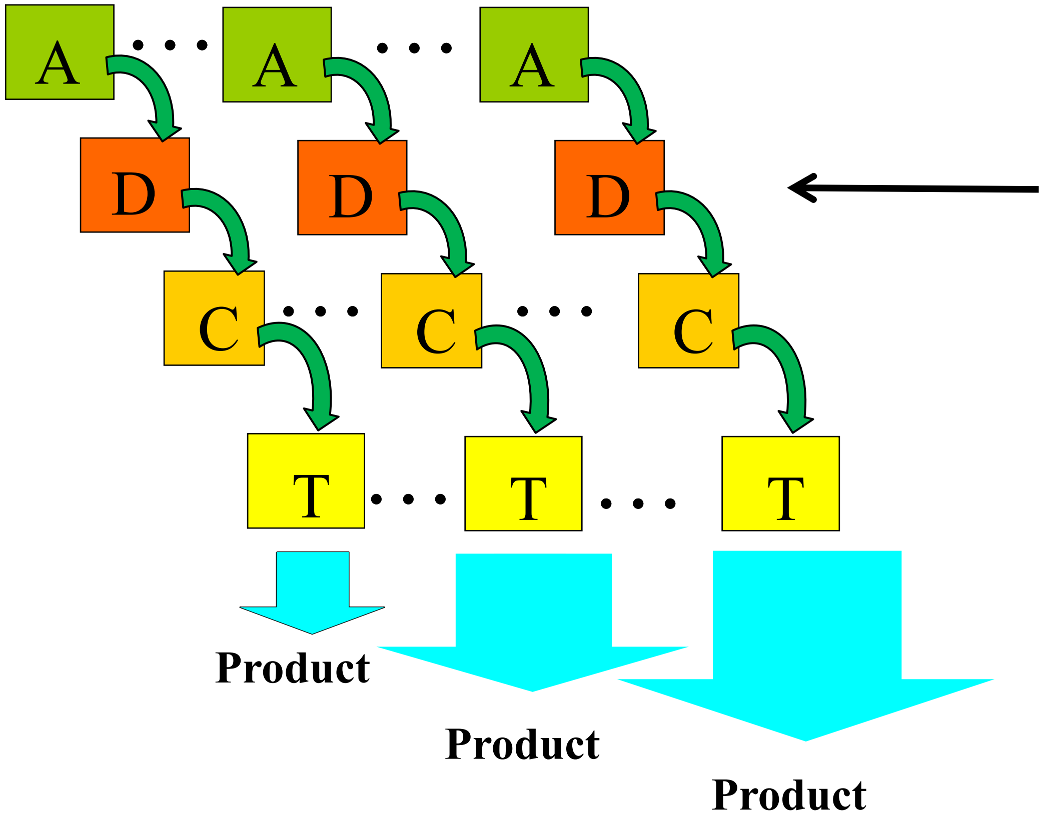
Prototyping Model Advantages

- Quicker user feedback is available leading to better solutions
- Errors can be detected much earlier.
- Missing functionality can be identified easily
- New requirements can be easily added.

Prototyping Model Disadvantages

- Users may not use the prototype in the same way as they use the final system.
- Excessive development time of the prototype.

Incremental Development Model



Feedback from iteration N leads to refinement and adaptation of the requirements and design in iteration N+1

The system grows incrementally

4 weeks (for example)

Incremental Development Model Overall

- Incremental development is based on the idea of developing an initial implementation, getting feedback from users and others, and evolving the software through several versions until the required system has been developed.
- Generally, the early increments of the system include the most important or most urgently required functionalities.

Incremental Development Model Overall

- In an incremental delivery process, customers define which of the services are most important and which are least important to them.
- This model can be used when major requirements of the complete system are clearly defined and understood; some details can evolve with time.

Incremental Development Model Overall

- Once the system increments have been identified, the requirements for the services to be delivered in the first increment are defined in detail and that increment is developed.

Incremental Development Model Overall

- During development, further requirements analysis for later increments can take place, but requirements changes for the current increment are not accepted.

Incremental Development Model Advantages

- It is easier to get customer feedback on the development work that has been done. Customers can use the early increments as prototypes and gain experience that informs their requirements for later system increments. Customers find it difficult to judge progress from software design documents or prototypes.

Incremental Development Model Advantages

- Customers do not have to wait until the entire system is delivered before they can gain value from it. The first increment satisfies their most critical requirements, so they can use the software immediately.
- It is easier to test and debug during a smaller iteration.
- New functionality can be defined for later increments.

Incremental Development Model Disadvantages

- In the incremental approach, there is no complete system specification until the final increment is specified. This requires a new form of contract, which large customers such as government agencies may find difficult to accommodate.

Incremental Development Model Disadvantages

- System structure tends to degrade as new increments are added. Regular change leads to messy code as new functionality is added in whatever way is possible.
- It becomes increasingly difficult and costly to add new features to a system.
- Total cost is higher than waterfall.

Practice

Which SDLC do you think is most suitable for each scenario described below?

- □ **Scenario 1**: A software engineer from a company asks you to develop a software for him.
- □ **Scenario 2**: You want to develop a mobile application that helps grade one students to learn English alphabets.

Practice

- **Scenario 3**: You are asked to develop a software for the study office to generate teaching schedules for all departments.

References

- **Reynolds, George Walter, Stair, Ralph M.** “Principles of information systems”, 13e – 2018