Republic of the Philippines
Department of Education
REGION III
SCHOOLS DIVISION OFFICE OF NUEVA ECIJA

LEARNING ACTIVITY SHEET
SPECIAL PROGRAM IN ICT 10
INFORMATION SYSTEM AND RESEARCH 10
Third Quarter, Week 4

Name of Learner: _____ Date:_____

Grade Level /Section: _____

TRADITIONAL VS OBJECT-ORIENTED APPROACH

BACKGROUND INFORMATION FOR LEARNERS

All software, especially large pieces of software produced by many people, should be produced using some kind of methodology. Even small pieces of software developed by one person can be improved by keeping a methodology in mind. A methodology is a systematic way of doing things. It is a repeatable process that we can follow from the earliest stages of software development through to the maintenance of an installed system. As well as the process, a methodology should specify what were expected to produce as we follow the process. A methodology will also include recommendation or techniques for resource management, planning, scheduling and other management tasks. Good, widely available methodologies are essential for a mature software industry.

In the effort to improve the systems analysis and design processes, different approaches have been developed: the traditional approach (structured) and object oriented. Information engineering includes the traditional system approach, which is also called the structured analysis and design technique. These approaches all have different advantages and disadvantages in a way that they could be used to fit and optimize different kinds of projects.

TRADITIONAL APPROACH

The traditional method uses a linear approach, where the stages of the software development process must be completed in a sequential order. This means that a stage must be completed before the next one begins. At the completion of each activity or phase, a milestone has been reached and a document is produced to be approved by the stakeholders before moving to the next activity or phase; painstaking amounts of documentation and signoffs through each part of the development cycle is required.

The list below shows how traditional approach is being used in systems development:

A. Requirements
Requirements capture is about discovering what is going to achieve with new piece of software and has two aspects. Business modeling involves understanding the context in which software will operate. A system requirement modeling (or functional specification) means deciding what capabilities the new software will have and writing down those capabilities

B.Analysis
Before designing a solution, it needs to be clear about the relevant entities, their properties and their inter-

relationships. Also needs to be able to verify understanding. This can involve customers and end users, since they're likely to be subject matter experts

C.Design

The design phase works on how to solve the problem. In other words, make decisions based on experience, estimation and intuition, about what software which will write and how will deploy it. System design breaks the system down into logical subsystems (processes) and physical subsystems (computers and networks), decides how machines will communicate, and chooses the right technologies for the job.

D.Specification

Specification is an often-ignored, or at least oftenneglected, phase. The term specification is used in different ways by different developers. For example, the output of the requirements phase is a specification of what the system must be able to do; the output of analysis is a specification of what are dealing with.

E.Implementation

In this phase is writing pieces of code that work together to form subsystems, which in turn collaborate to form the whole system. The sort of the task which is carried out during the implementation phase is "Write the method bodies for the Inventory class, in such a way that they conform to their specification" .

F.Testing

When the software is complete, it must be tested against the system requirements to see if it fits the original goals. It is a good idea for programmers to perform small tests as they go along, to improve the quality of the code that they deliver.

G.Deployment

The deployment phaseare concerned with getting the hardware and software to the end users, along with manuals and training materials. This may be a complex process, involving a gradual, planned transition from the old way of working to the new one.

H.Maintenance

When the system is deployed, it has only just been born. A long life stretches before it, during which it has to stand up to everyday use – this is where the real testing happens. Some of the problems usually discovered during the maintenance phase are: 1.when the log-on window opens and 2. it still contains the last password entered. Software developers are normally interested in maintenance because of the faults (bugs) that are found in the software in which these faults must be removed as quickly as possible, rolling out fixed versions of the software to keep the end users happy. As well as faults, users may discover deficiencies (things that the system should do but doesn"t) and extra requirements (things that would improve the system).

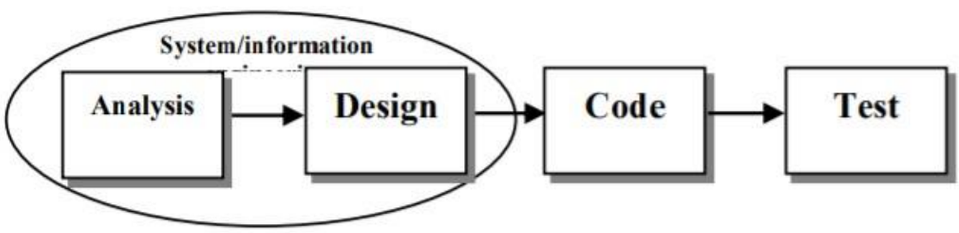

Figure I: The Linear Sequential Model

OBJECT-ORIENTED APPROACH

Object-oriented (O-O) analysis and design is an approach that is intended to facilitate the development of systems that must change rapidly in response to dynamic business environments. Object-oriented techniques are thought to work well in situations in which complicated information systems are undergoing continuous maintenance, adaptation, and redesign. Object-oriented approaches use the industry standard for modeling object-oriented systems, called the unified modeling language (UML), to break down a system into a use case model.

Object-oriented programming differs from traditional procedural programming by examining objects that are part of a system. Each object is a computer representation of some actual thing or event. Objects may be customers, items, orders, and so on. Objects are represented by and grouped into classes that are optimal for reuse and maintainability. A class defines the set of shared attributes and behaviors found in each object in the class.

The phases in UML are similar to those in the SDLC. Since those two methods share rigid and exactmodeling, they happen in a slower, more deliberate pace than the phases of agile modeling. The analyst goes through problem and identification phases, an analysis phase, and a design phase. the following steps give a brief description of the UML process.

1.Define the use case model.

In this phase the analyst identifies the actors and the major events initiated by the actors. Often the analyst will start by drawing a diagram with stick figures representing the actors and arrows showing how the actors relate. This is called a use case diagram and it represents the standard flow of events in the system. Then an analyst typically writes up a use case scenario, which describes in words the steps that are normallyperformed.

2. During the systems analysis phase, begin drawing UML diagrams.

In the second phase, the analyst will draw Activity Diagrams, which illustrate all the major activities in the use case. In addition, the analyst will create one or more sequence diagrams for each use case, which show the sequence of activities and their timing. This is an opportunity to go back and review the use cases, rethink them, and modify them if necessary.

3. Continuing in the analysis phase, develop class diagrams.

The nouns in the use cases are objects that can potentially be grouped into classes. For example, every automobile is an object that shares characteristics with other automobiles. Together they make up a class.

4.Begin systems design by modifying the UML diagrams. Then complete the specifications. Systems design means modifying the existing system and that implies modifying the diagrams drawn in the previous phase. These diagrams can be used to derive classes, their attributes, and methods (methods are simply operations). The analyst will need to write class specifications for each class including the attributes, methods, and their descriptions. They will also develop methods specifications that detail the input and output requirements for the method, along with a detailed description of the internal processing of the method.

5.Develop and document the system.

UML is, of course, a modeling language. An analyst may create wonderful models, but if the system isn't developed there is not much point in building models. Documentation is critical. The more complete the information you provide through documentation and UML diagrams, the faster the development and the more solid the final production system.

Object-oriented methodologies often focus on small, quick iterations of development, sometimes called the spiral model. Analysis is performed on a small part of the system, usually starting with a high-priority item or perhaps one that has the greatest risk. This is followed by design and implementation. The cycle is repeated with analysis of the next part, design, and some implementation, and it is repeated until the project is completed. Reworking diagrams and the components themselves is normal. UML is a powerful modeling tool that can greatly improve the quality of your systems analysis and design and the final product.

LEARNING COMPETENCY

Compare traditional approach from object-oriented approach

ACTIVITIES

ACTIVITY 1:
Direction:Write True if the statement is correct. If not, underline the word or statement that makes the sentence incorrect and write the correct answer on the space provided.
_____1. The design phase works on how to solve the problem.
_____2. The traditional method uses a linear approach, where the stages of the software development process must be completed in a randomized order.
_____3. The use case diagram represents the standard flow of events in the system.

_____4. UML is a modeling language.

_____5. The activity diagram illustrates the minor activities in the use case.

ACTIVITY 2:

Compare and contrast traditional approach from object-oriented approach.

_____

_____

_____

_____

REFLECTION:

Cite a specific example on how you can apply traditional approach in your daily living.

_____

_____

_____

REFERENCES

traditional approach in system development - Google Search
Various approaches of systems analysis and design (umsl.edu)
Traditional vs. Object-Oriented Approaches: Object-Oriented Approach | Saylor Academy
Object Oriented Approach - Tutorialspoint
Object-Oriented Systems Analysis and Design (w3computing.com)

Prepared by:

HAYDEE C. BADUA
Name of Writer

Noted by:

LABERNE A. LADIGNON, JR
Division ICT Coordinator/ OIC EPS