

SYSTEM DEVELOPMENT LIFE CYCLE

- ix. A process is said to be in ready state if it is waiting for some external event which has not yet been occurred.
- x. Batch processing systems does not support interactivity.

Q.4 What is an Operating system? Explain different types of operating systems.

Q.5 Differentiate between the following:

- Single-user and Multi-users operating systems
- Thread and Process
- Multiprogramming and Multiprocessing

Q.6 Define distributed operating system and describe its advantages and disadvantages.

Q.7 Write short notes on the following.

- Different states of a process
- DOS
- Real time processing systems
- Embedded operating systems

Q.8 Describe functions of an operating system.

Q.9 Compare DOS, Windows and UNIX operating systems.

After the completion of Unit-2, Students will be able to:

- Define a system.
- Explain System Development Life Cycle and its importance.
- Describe objectives of SDLC.
- Describe stakeholder of SDLC and their roles.
- Explain the following phases of SDLC:
 - Planning
 - Feasibility
 - Analysis
 - Requirement engineering
 - ❖ Requirement gathering
 - Functional requirements
 - Non-functional requirements
 - ❖ Requirement validation
 - ❖ Requirement management
 - Design (algorithm, flowchart and pseudo code)
 - Coding
 - Testing/verification
 - Deployment/implementation
 - Maintenance/support
- Explain the role of the following in :
 - Management
 - Project manager
 - System analyst
 - Programmer
 - Software tester
 - Customer

2.1 INTRODUCTION

Systems are created to solve problems. Nowadays, systems are so big and complex that teams of architects, analysts, programmers, testers and users must work together to create the millions of lines of custom-written code that drive our enterprises. To manage this, a number of system development life cycle (SDLC) models have been created. This Unit is focused on SDLC, its importance, objectives, stakeholders and logical phases that are followed for the development of a software product.

2.1.1 A System

The term system is originated from the Greek term systēma, which means to "place together." It can be defined as a set of interrelated components having a clearly defined boundary that work together to achieve a common set of objectives.

A system can be developed by applying a set of methods, procedures and routines in a proper sequence to carry out some specific task. When all these functions are applied to build software then the system will be called as a software system.

System Development Life Cycle (SDLC)

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application.

2.1.2 System development life cycle and its importance

System Development Life Cycle (SDLC) is a step wise process of creating computer systems. It is also known as information system development or application development. It is a conceptual model which represents the necessary steps used for the development process of a software system.

The SDLC is a problem-solving process through which a series of steps helps to produce a new computer information system. The entire steps conducted in a sequence should provide the answers to a problem or opportunity. This step-wise procedure to build a system has a lot of importance.

Importance of (SDLC)

The following points summarize the importance of the use of SDLC.

- SDLC is important because it breaks down the entire life cycle of software development into phases thus making it easier for the development team members to easily evaluate each part of software development.
- SDLC makes it easier for programmers to work concurrently on each phase.
- It provides a rough time estimate that when the software will be available for use.
- It delivers quality software which meet or exceed customer expectations.
- It provides the basic framework for the developing of quality software.
- SDLC helps the project managers to establish a project management structure to be followed strictly during the system development.
- SDLC clearly defines and assigns the roles and responsibilities of all the involved parties.
- It ensures that the requirements for the development of the software system are well defined and subsequently satisfied.

2.1.3 Objectives of SDLC

The objectives of the System Development Life Cycle (SDLC) are as follows:

- Delivery of quality software that meet the customer expectations.
- Delivery of inexpensive and cost-effective software which are easily maintainable.
- Maximize productivity in terms of the software systems delivered.

- One of the major objectives of SDLC is to establish an appropriate level of management authority to direct, coordinate, control, review, and approve the software development project.
- SDLC should ensure the project management accountability.
- Proper documentation of all the requirements needed for the development of the new software system.
- Ensuring that projects are developed within the current and planned information technology infrastructure.
- SDLC should identify the potential project risks in advance so that proper planning should be done.

2.1.4 Stakeholders of SDLC

Those entities which are either within the organization or outside of the organization that sponsor a project, or have an interest or have the intention to get it after its successful completion, or may have a positive or negative influence in the project completion are called **stakeholders**. Project stakeholders include the customers, the user group, the project manager, the development team and the testers. All those who have some interest in the project can be considered as stakeholders of that project. The individuals as well as the organizations that are actively involved in the project, or whose interests may be affected as a result of project execution or project completion are the part of stakeholders. It is the duty of the project management team to identify the stakeholders, determine their requirements, expectations and manage their influence in relation to the requirements to ensure a successful project.

Role of stakeholders

The basic roles of the stakeholders are:

- For the development of a software system, resources such as time, money, equipment etc. are needed which should be provided to the project team by the stakeholders.

- Stakeholders educate the developers about their business.
- They spend more time to provide information and clarify requirements to the analysts and developers.
- The stakeholders should be specific and precise about the requirements.
- Make timely decisions.
- Respect a developer's assessment of cost and feasibility.
- Set requirement priorities.
- Review and provide timely feedback.
- Promptly communicate changes to requirements.

2.1.5 System development life cycle phases

The most commonly used phases of SDLC are given below.

- Planning
- Feasibility Study
- Analysis
- Requirement Engineering
- Design
- Coding
- Testing / Verification
- Deployment / Implementation
- Maintenance / Support

These phases are diagrammatically shown in Figure 2.1.

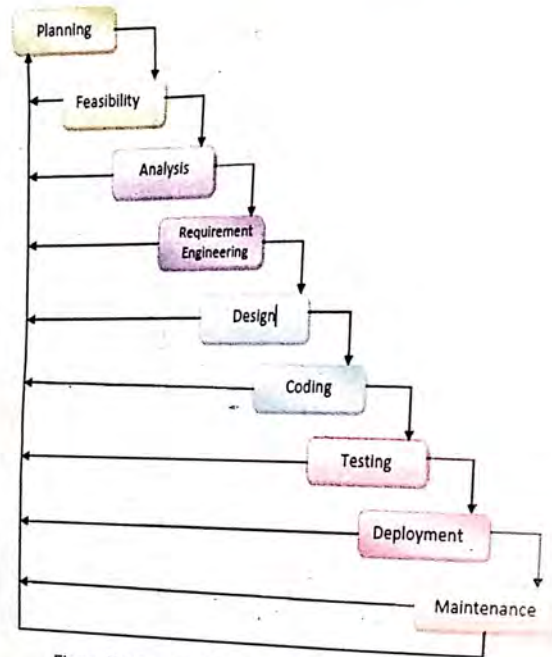


Figure 2.1: Phases of System Development Life Cycle

a. Planning phase

Planning phase is the first phase of System Development Life Cycle. During this phase, the group that is responsible for creating the software system must first determine what the system needs to do for the organization. Often, the planning phase tries to find answers for the questions such as:

- What do we need this system for?
- What the new software system will do for the organization?
- How this new software system will be developed?

In this initial phase of SDLC, all the resources, both human and technology resources are put together and a project plan is devised by the project manager.

Activities of the planning phase

- Define business problem or opportunity
- Define Project Scope and Constraints
- Produce detailed project schedule (time) and cost
- Define project benefits
- Write a report to management

b. Feasibility

Analysis and evaluation of a proposed project or system, to determine, whether it is technically, economically and operationally feasible within the estimated cost and time is called feasibility study. Feasibility study is one of the important steps in SDLC. It is divided into:

i. Technical feasibility

It refers to the technical resources needed for the development of the proposed software system.

ii. Economic feasibility

This refers to the cost of the project. Here, it tries to determine; total cost of ownership (TCO), tangible benefits and intangible benefits.

iii. Operational feasibility

In this type of feasibility it is determined that whether the proposed system will be used effectively after it has been developed. It means whether the organization has much experienced personals to operate the system or not?

iv. Schedule feasibility

It means that a project can be implemented in an acceptable time frame.

c. Analysis phase

In this phase, the incharge of the project team must decide if the project should go ahead with the available resources or not. Analysis is also looking at the existing system to see what and how it is doing its job. The project team asks the following questions during the analysis.

- Can the proposed software system be developed with the available resources and budget?
- Will this system significantly improve the organization?
- Does the existing system even need to be replaced?

Activities of the analysis phase

- Gather information to learn problem domain
- Define system requirements.
- Prioritize requirements.
- Build prototypes for feasibility and discovery of requirements.
- Generate and evaluate alternative solutions for the problem in hand.
- Review recommendations with top level management to decide about the project.

d. Requirement engineering

Requirements Engineering (RE) is a set of activities used to identify and communicate the purpose of a software system, and the contexts in which it will be used. Requirement engineering consists of the following steps.

- Requirement gathering
- Requirement validation
- Requirements management

Requirement gathering

Requirement gathering is usually the first part of any software product. In this phase, meetings with the customers or prospective customers are arranged; the market requirements and features that are in demand are analyzed. It is also to find out if there is a real need in the market for the software product that is proposed to be developed.

These requirements are of two types:

- Functional requirements
- Non-Functional Requirements

Functional requirements: Functional requirement are those requirements of a software system which describe a function of a software system or its component. It includes calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Examples of functional requirements are:

a) Interface requirements

- Field accepts numeric data entry.
- Field only accepts dates before the current date.
- Screen can print on-screen data to the printer.

b) Business requirements

- Data must be entered before a request can be approved

c) Regulatory/Compliance requirements

- The database will have a functional audit trail.
- The system will limit access to authorized users.
- The spreadsheet can secure data with electronic signatures.

d) Security requirements

- Member of the Data Entry group can enter requests but not approve or delete requests.

Members of the Managers group can enter or approve a request, but not delete requests.

Members of the Administrators group cannot enter or approve requests, but can delete requests. ⑧

⑧ **Non-functional requirements:** Non-functional requirements are those requirements which specify criteria for the judgment of the operations of a system. It describes that how well the system performs its duties. Nonfunctional requirements are often called qualities of a system. These requirements depend upon the nature of the software.

Different types of non-functional requirement are:

Accessibility Requirements
Accuracy Requirements
Backup and Recovery Requirements
Memory Capacity Requirements
Compatibility Requirements
Error-Handling Requirements
Maintainability Requirements
Performance Requirements
Security Requirements ⑧

Requirements validation

Requirement validation is concerned with examining the requirements to certify that they meet the intentions of the stakeholders, and to ensure that they define the right system, the essence of the agreement and understandings between developer and acquirer about what to build, in a manner that ensures a common understanding across the project team

and among the stakeholders. The validation differs from verification in the sense that verification occurs after requirements have been accepted.

In requirements validation, the requirements elicited are reviewed to check that requirements are complete, correct, unambiguous, consistent, prioritized, modifiable, and traceable.

Requirements management

Requirements management is performed to ensure that the software continues to meet the expectations of the acquirer and users. The requirements document should always accurately reflect those expectations, as changes occur, in order to communicate to the development team.

Requirements management, therefore, needs to gather new requirements that arise from changing expectations, new regulations, or other sources of change. It needs to analyze the impact of those changes on the project. The conflicts that arise among the acquirer and the user needs to be addressed and changes should be negotiated and validated before such requirements are added to the baseline.

e. Design phase

During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase can be very expensive to solve in later stages of the software development, therefore, a variety of elements are considered in the design to minimize the risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.

- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.

The design phase normally consists of three different architectures. These are:

- Algorithms
- Flow chart
- Pseudo code

i. Algorithms

An algorithm is a specific set of instructions for carrying out a procedure or solving a problem.

For Example the following algorithm will find average of the 5 numbers.

```

Sum = 0, Average = 0
Read a, b, c, d, e
Sum = a+b+c+d+e
Average = Sum/5
Print Average
  
```

ii. Flowcharts

A flowchart is a type of diagram that represents an algorithm or a process. It shows the steps of the algorithms with the help of boxes and their order by arrows connecting the boxes. This diagrammatic representation of the algorithm gives a step-by-step solution to a given problem. The operations are represented in these boxes, and the flow of control on the arrows. These flowcharts are used for analyzing, designing, documenting or managing a process or program in various fields.

Some most commonly used flowchart symbols are shown in Figure 2.3.

Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g., PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IFTHEN/ELSE).

Figure 2.2 Flowchart Symbols

Example: Flowchart for the above algorithm will be as follows.

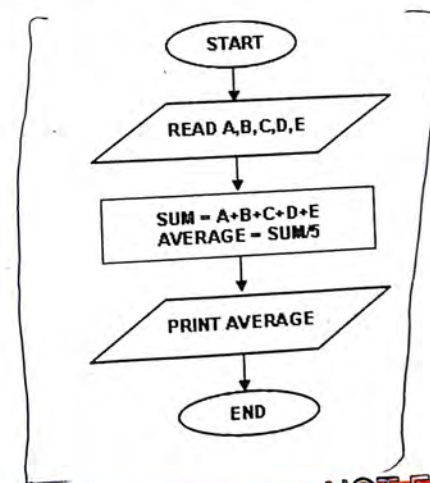


Figure 2.3 Flowchart

iii. Pseudo code

Pseudo code is an outline of a program, written in a form that can easily be converted into real programming statements. Pseudo code neither is compiled nor executed, and there is no specific formatting and syntax rules for it.

Pseudo code is beneficial in the sense that it enables the programmer to focus on the algorithms without worrying about all the syntactic details of a particular programming language. In fact, one can write pseudo code without even knowing what programming language will be used for the final implementation.

The following example shows a pseudo code to display the result of a student as "Pass" if the percentage is greater than or equal to 60% and "Fail" otherwise.

IF Percentage $\geq$ 60%

Print "Pass"

Else

Print "Fail"

f. Coding

Coding is the process of designing, writing, testing, debugging, and maintaining the source code of computer programs. This source code is written in programming languages. The purpose of coding is to create a program that shows a certain desired behavior. For coding some other specialties are also needed. These include knowledge of the application domain, specialized algorithms and formal logic. Coding is also called computer programming.

For coding computer programming languages are used. There are different types of programming languages such as procedural languages and object oriented programming (OOP) languages. Nowadays, object oriented programming languages are the most

commonly used languages. C++, Java, C#.Net, and VB.Net etc. are the examples of this group of languages.

The coding style of the C++ programming language is shown as follows.

Example 1

```
/* program to add N numbers*/
#include <iostream.h>
#include<conio.h>
int main ()
{
    int N, sum=0, y ;
    cout<< "Enter value for N"<<endl;
    cin>>N ;
    for (int x=1 ; x<=N ; ++x)
    {
        cin>>y ;
        sum= sum+y ;
    }
    cout<< "The sum of N values is: "<<sum;
    getch();
    return 0;
}
```

Example 2

```
/* program to display result s of the students*/
#include <iostream.h>
#include<conio.h>
int main ()
```



```

{
float Percentage;
cout<< "Enter Percentage of the student"<<endl;
cin>> Percentage;
if (Percentage >=60.0)
cout<< "Passed";
else
cout<< "Failed";
getch();
return 0;
}

```

chart
chart in

g. Testing/Verification

The execution of a program to find its errors is called testing. Here, the bugs are identified in the programmed modules. The purpose of testing is to evaluate an attribute or capability of a program or system and determine that whether it meets its required results. Testing/verification the software is actually operating the software under controlled conditions. It is the process of checking the items for consistency by evaluating the results against pre-specified requirements.

h. Deployment / Implementation

Software deployment is a set of activities that are used to make the software system available for use. The deployment is also called implementation. Its activities occur at the producer site or at the consumer site or both. Because every software system is unique, the precise processes or procedures within each activity are different from others and thus cannot be clearly defined.

The main activities that are involved during deployment are:

- Installation and activation of the equipments and software.

- In some cases the users and the computer operation personals are trained on the developed software system.
- Conversion: The process of changing from the old system to the new one is called conversion.

i. Maintenance / Support

Keeping a system in its proper working condition is called maintenance. In other words, it can also be defined as the modification of a software product after delivery to correct faults, to improve performance. Maintenance is an important activity that takes place throughout the life of software in both the computer equipments and computer software. Programmers spend more time in maintaining programs than writing them.

In SDLC, the system maintenance is an ongoing process. The system is monitored continually for performance in accordance with user requirements and needed system modifications are incorporated. When modifications are identified, the system may reenter the planning phase. This process continues until a complete solution is provided to the customer. Maintenance can be either be repairing or modification or some enhancement in the existing system.

2.1.6 Role of the Personnel involved in SDLC

The activities of all the phases of software development life cycle are being performed by different groups of people and individual personnel. These personnel are professionals in performing their particular jobs and include:

- Management
- Project Manager
- System Analyst
- Programmer
- Software Tester
- Customer

a. Management

Organizing, coordinating and controlling the activities of a software development by the managers and executives in accordance with certain standard procedures is called management. A strong management has the ability to satisfy the customers and acquirers of the software system. Also, a proposed project or a product will only meet its objectives if managed properly, otherwise, will result in failure.

The role of good Management is to:

- Provide consistency of success of the software with regard to Time, Cost, and Quality objectives.
- Ensure that customer expectations are met.
- Collect historical information and data for future use.
- Provide a method of thought for ensuring all requirements are addressed through a comprehensive work definition process.
- Reduce risks associated with the project.
- Minimize scope creep by providing a process for managing changes.

b. Project Manager

A **project manager** is a professional in the field of project management responsible for planning, execution, and closing of any project. Apart from management skills, a software project manager will typically have an extensive background in software development. He is also expected to be familiar with the whole software development life cycle process. The specific responsibilities of the project manager vary depending on the company size, the company maturity, and the company culture.

The role of project manager is to manage:

- Project plan

- Project stakeholders
- Project team
- Project risk
- Project schedule
- Project budget
- Project conflicts

c. System Analyst

A systems analyst is a professional in the field of software development that studies the problems, plans solutions for them, recommends software systems, and coordinates development to meet business or other requirements. System analyst has expertise in a variety of programming languages, operating systems, and computer hardware platforms. Because they often write user requests into technical specifications, the systems analyst is the contact person between vendor and information technology professionals. Analyst may be responsible for developing cost analysis, design considerations, and implementation time.

The role of system analyst is to:

- Plan a system flow from the ground up.
- Interact with customers to learn and document requirements that are then used to produce business requirements documents.
- Write technical requirements from a critical phase.
- Interact with designers to understand software limitations.
- Help programmers during system development, e.g. provide use cases, flowcharts or even database design.
- Manage system testing.
- Document requirements and contribute to user manuals.

- Whenever a development process is conducted, the system analyst is responsible for designing components and providing that information to the developer.

d. Programmer

A programmer is a technical person that writes computer programs in computer programming languages to develop software. A programmer writes, tests, debugs, and maintains the detailed instructions that are executed by the computer to perform their functions. Other names for programmer are coder and developer.

The roles of a programmer are:

- Writing, testing, and maintaining the instructions of computer programs.
- Updating, repairing, modifying and expanding existing programs
- Testing the code by running to ensure its correctness
- Developing new methods and approaches to computer programming
- Consultancy with outside parties in relation to the construction of computer programming methods and the programs themselves.
- Supervision of the information systems (in some cases).
- Maintenance of computer databases is another type of specific duty which a computer programmer may find himself/herself responsible.
- Preparing graphs, tables and analytical data displays which show the progress of a computer program.

e. Software Tester

A software tester is a computer programmer having specialty in testing the computer programs using different testing techniques. Software tester is responsible for understanding requirements, creating test scenarios, test scripts, preparing test data, executing test scripts and reporting defects and reporting results to test lead.

The roles of a software tester are:

- Create test scenarios, test conditions and expected results and test cases.
- Run and maintain automated test scripts.
- Create required test data and maintain common test data sheet of the project.
- Create test deliverables as per testing standards followed by the company or project.
- Execute test scripts and document test results.
- Inform testing team leader on any issues that could potentially impact quality, schedule or budget of the project.

f. Customer

A Customer is an individual or an organization that is a current or potential buyer or user of the software product. Customers usually purchase software from software manufacturer companies (software houses), users groups and individuals. Customers are also called clients but the only difference between the two is that the customers purchase the software products and the clients purchase services. Customers are the real evaluators of a software product by using it and identifying its merits and demerits.

Summary

- System can be defined as a set of interrelated components having clearly defined boundary that work together to achieve a common set of objectives.
- The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance.
- The advantage of System Development Life Cycle (SDLC) is that it provides a step wise process to the development of computer Software.
- The main objective of SDLC is to obtain quality software that cost low price and satisfy the customers.
- Stakeholders are those entities which are either within the organization or outside of the organization that sponsor a project, or have an interest or have the intention to get it after its successful completion, or may have a positive or negative influence in the project completion.
- Planning phase is the first phase of Software Development Life Cycle in which it is determined what the system needs to do for the organization.
- Analysis and evaluation of a proposed project or system is to determine, whether it is technically, economically and operationally feasible within the estimated cost and time is called feasibility study.
- Requirements Engineering (RE) is a set of activities used to identify and communicate the purpose of a software system, and the contexts in which it will be used.
- Analysis is the process of looking at the existing system to determine what and how it is doing its job.
- Design phase is that phase of SDLC in which the system is designed to satisfy the functional requirements identified in the requirement engineering phase.
- Coding is the process of designing, writing, testing, debugging, and maintaining the source code or computer programs.

- Testing is the process of executing users programs to find its errors. In this process, bugs are identified in the programmed modules.
- Software deployment is a set of activities that are used to make the software system available to the users for use. The deployment is also called implementation.
- Software maintenance is the process of keeping a system in its proper working condition.

Activities

For the following activities the instructor should divide the students into groups. Each group should be assigned one of the projects discussed in the activities. The groups are to prepare a description of what needs to be carried out for their projects for each phase of the software development life cycle (SDLC).

ACTIVITY 1

An automotive manufacturing company contacts your IT Company to develop a new system for stock control of automotive parts. The current system is manual.

ACTIVITY 2

A government department contacts your IT Company to develop a new system for maintaining patients' appointments at the hospital. The current system is manual.

ACTIVITY 3

A college contacts your IT department to develop a new system for lecturers to enter student's grades and attendance. The current system is manual. List the personnel involved in this project.

ACTIVITY 4

A cinema manager wants to upgrade his current system of booking seats so that people will also be able to book through the internet. He has contacted your company for the job.

ACTIVITY 5

A large travel agency has contacted your IT department within the travel agency to upgrade the current outdated computerized system which the company is using to book seats.

ACTIVITY 6

Teacher should help the students in drawing flowcharts for the above five (5) projects mentioned in activities 1, 2, 3, 4, and 5, and assign some new problems as well as assignments to draw flowcharts for them.

Exercise**Q.1 Fill in the Blanks.**

- i. The requirements which define the function of a software system are called _____ requirements.
- ii. All the activities that make a software system available for use is known as software _____.
- iii. The person who is familiar with a variety of programming languages, operating systems and computer hardware is known as _____.
- iv. A software tester is basically a _____ having a specialty in testing the computer programs.
- v. A project manager is a professional in the field of _____.

Q.2 Select the correct choice for the following Multiple Choice Questions.

- i. The first step in the system development life cycle is:
 - a) Analysis
 - b) Design
 - c) Problem Identification
 - d) Development and Documentation

- ii. The organized process or set of steps that needs to be followed to develop an information system is known as:
 - a) Analytical cycle
 - b) Design cycle
 - c) Program specifications
 - d) System development life cycle
- iii. Enhancements, upgradation and bugs fixation are done during the _____ step in the SDLC.
 - a) Maintenance and Evaluation
 - b) Problem Identification
 - c) Design
 - d) Development and Documentation
- iv. The _____ determines whether the project should go forward.
 - a) Feasibility
 - b) Problem identification
 - c) System evaluation
 - d) Program specification
- v. _____ spend most of their time in the beginning stages of the SDLC, talking with end-users, gathering requirements, documenting systems and proposing solutions.
 - a) Project managers
 - b) System analysts
 - c) Network engineers
 - d) Database administrators
- vi. The entities having a positive or negative influence in the project completion are known as _____.
 - a) Stakeholders
 - b) Stake supervisors
 - c) Stake owners
 - d) None of the above