The process of prototyping involves the following steps :

1. **Identify basic requirements :** Determine basic requirements including the input and output information desired. Details such as security, can typically be ignored.
2. **Develop Initial Prototype :** The initial prototype is developed that includes only user interfaces.
3. **Review :** The customers, including end-users, examine the prototype and provide feedback on additions or changes.
4. **Revise and Enhancing the Prototype :** Using the feedback both the specifications and the prototype can be improved. Negotiation about what is within the scope of the contract/product may be necessary. If changes are introduced then a repeat of steps 3 and step 4 may be needed.

Types of prototyping

Software prototyping has many variants. However, all the methods are in some way based on two major types of prototyping :

1) *Throwaway Prototyping* (small part developed)
2) *Evolutionary Prototypi* (overall Project)

Third Generation (3GL)

Definition - What does *Third Generation (Programming) Language (3GL)* mean?

A third generation (programming) language (3GL) is a grouping of programming languages that introduced significant enhancements to second generation languages, primarily intended to make the programming language more programmer-friendly.

English words are used to denote variables, programming structures and commands, and Structured Programming is supported by most 3GLs. Commonly known 3GLs are FORTRAN, BASIC, Pascal and the C-family (C, C+, C++, C#, Objective-C) of languages.

Also known as a 3rd generation language, or a high-level programming language.

Fourth generation (4 GLs) :)A fourth generation (programming) language (4GL) is a grouping of programming languages that attempt to get closer than 3GLs to human language, form of thinking and conceptualization.

4GLs are designed to reduce the overall time, effort and cost of software development. The main domains and families of 4GLs are: database queries, report generators, data manipulation, analysis and reporting, screen painters and generators, GUI creators, mathematical optimization, web development and general purpose languages.

Also known as a 4th generation language, a domain specific language, or a high productivity language.

Some examples of **4GL** are: **database query language** e.g.**SQL**; Focus, S, **IDL-PV, WAVE, Gauss, Mathematic**, and data-stream languages such as **AVS**, APE, **Iris Explorer.**

Object-oriented analysis:

The use of modelling to define and analyze the requirements necessary for success of a system. Object-oriented analysis is a process that groups items that interact with one another, typically by class, data or behaviour, to create a model that accurately represents the intended purpose of the system as a whole. Object-oriented analysis does not factor implementation limitations into the model.

What is Object-Oriented Analysis?

To define object-oriented analysis we must first define what we mean by an object. The definition of an object, according to most dictionaries, is "a tangible, material thing." Drilling down a bit more to the realm of computer science, an object can be most anything in a programmatic sense, from a variable or data model to a function, class, or method. Moving even deeper into the realm of object-oriented programming, an object is an instance of a thing that typically represents a real world object and has all the same types of characteristics (properties), behaviors (methods), and states (data). When discussing OOAD concepts, an object most closely resembles the object-oriented programming version of an object, in that it is a representation of a real world object with behaviors, characteristics, and states.

With that out of the way, we can define object-oriented analysis (OOA). In short, OOA is an iterative stage of analysis, which takes place during the software development life

cycle, that aims to model the **functional** requirements of the software while remaining completely independent of any potential **implementation** requirements. To accomplish this task via OOAD practices, an object-oriented analysis will focus everything through the lens of objects. This forces OOA to combine *all* behaviors, characteristics, and states together into one analysis process, rather than splitting them up into separate stages, as many other methodologies would do.

To accomplish this goal, a typical OOA phase consists of five stages:

- 1) Find and define the objects.
- 3) Organize the objects.
- 2) Describe how the objects interact with one another.
- 4) Define the external behavior of the objects.
- 5) Define the internal behavior of the objects.

For example, a typical implementation of OOA is to create an object model for an application. The object model might describe the names, relationships, behaviors, and characteristics of each object in the system. With this information established for each object, the design process that follows is much simpler.

Advantages of Object-Oriented Analysis and Design

collection/combine

- **Encourages Encapsulation**: Since everything within OOAD revolves around the concept of objects (specifically, the object-oriented variety), one of the biggest advantages of OOAD is that it encourages planning and development of systems that are truly independent of one another. Just like a class written using object-oriented techniques, all the systems and objects produced during an OOAD development life cycle can be mixed and matched as necessary, since they will ideally be built as completely self-contained entities.
- **Easy to Understand**: Since OOAD principles are fundamentally based on real world objects, it's quite easy for everyone on the team to quickly understand what an object name means or how a particular behavior, well, behaves. This makes the overall development life cycle a much smoother process, particularly if your team needs to frequently interact with customers or other non-technical users about the objects and components in the system. In such cases, most people still understand how system components and modelled objects work when they're based on real world objects and ideas.

Disadvantages of Object-Oriented Analysis and Design

- **Ill-Suited to Procedural Applications**: Given the object-oriented nature of OOAD, it is quite difficult (although not impossible) to practice OOAD techniques within a procedural programming language, or often to apply the techniques to non-object business logic. Whereas procedural applications are often logically bound by concepts of scope and modularity, object-oriented applications, of course, emphasize *objects* that simulate the real world, making OOAD methods ill-suited for procedural languages and applications.
- **Too Complex for Simple Applications**: While arguably not a disadvantage that is applicable to all projects, it's certainly the case that OOAD practices are generally not ideal for simpler projects. Many developers have their own personal hard and fast rules to help when deciding whether a project should be procedural or object-oriented, but in most cases, the more basic the needs of the application, the more likely a less-structured, procedural approach is the best fit. As always, we must always use our own best judgment.

*Systems design :

Systems design :is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

System Design

1. System Model:

A logical or mathematical representation of a system is known as system model.

Model can be built for the existing system to better understand the proposed system.

2. Process Modeling:

Process modeling is technique which involves graphical representation of functions or processes that capture , manipulates, stores or distribute data between a system & its environment or among components within a system. DFD(Data Flow Diagram) is one of the common form of process model.

3. Logical & Physical design:

System design involves two categories-

a. Logical design:

It is concerned with the specification of major features of the system that would meet the objectives. The delivered product of logical design may be called as a blueprint of a new system.

Logical design of a system includes content requirements and some of following component

i. Output(Reports & Displays)

ii. Input forms

iii Procedures(Structured of procedures to collect, transforms & output data)

iv. Storage(Requirements for data to be stored in the database)

v. Control (Requirements for data integrity, security & procedure for recovery)

b. Physical design:

It requires logical design or blueprint and produces the program specification, physical files or database definitions. It also includes user interface design and selects hardware & software packages.

4. Input Output Design:

a. Input Design:

Part of the system which deals with the design of the interface through which user communicates with the system and feeds the input data to the system.

b. Output Design:

The part of the system which deals with determining how the output is to be presented, in what format or shape is known as output design of the system. Computer produces displays and print reports that are to be read and used by users therefore the output must be clear and easy to understand.

What is I/O Forms? Types

c. I/O Forms: It carry the inputed data form use

It is primary carrier for data or information. They are the user requests for some action. A properly designed form is necessary for efficient functioning of a system.

Form is used to –

i. To obtain information efficiently

ii. Distributes information easily

iii. Store information effectively and economically

Types of Forms:

1. Action Form:

This type of form requests the user to perform certain actions and they are generally moves from one person to organization or another person.

2. Memory Form:

tube history, c

It record historical data such as stock ledger, purchase record, bond form.

3. Report Form:

summery of reports /

It provides summarized information. These are generally used by managers and such people who are required to make decision making. They guide supervisors and administrators for decision making.

Form designing:

Steps for form designing-

i) Define the purpose of form

ii) Specify its data contents

What is I/O Forms? Types

c. I/O Forms: It carry the inputed data from user

It is primary carrier for data or information. They are the user requests for some action. A properly designed form is necessary for efficient functioning of a system.

Form is used to –

i. To obtain information efficiently

ii. Distributes information easily

iii. Store information effectively and economically

Types of Forms:

1. Action Form:

This type of form requests the user to perform certain actions and they are generally moves from one person to organization or another person.

2. Memory Form:

It record historical data such as stock ledger, purchase record, bond form. tube history, go

3. Report Form:

It provides summarized information. These are generally used by managers and such people who are required to make decision making. They guide supervisors and administrators for decision making. summery of reports /

Form designing:

Steps for form designing-

i) Define the purpose of form

ii) Specify its data contents

iii) Using a guide sheet enter title, form number and indicates position of any specially located mate ex. notes

iv) Review with the user & revised if necessary

v) Create few test forms.

Output Design

1. Designing layout:

The arrangement of information on a monitor screen or printed on a paper is termed as layout. The output design is specified on layout forms, sheets that describe the location such as length and type of text, format of column headings, pagination, etc. Plot layout / Page layout house building planning

2. Category of output:

i. External

ii. Internal

3. Report generation:

Printed reports are formal documents that provide users with the information so that users can perform their jobs effectively. It must be clear and easy to understand. We have to consider some features while designing a report such as : titles, subtitles, date, page no and summary.

4. Design of screens:

The reports which are to be shown on a computer screen must be given a clear instruction how a user proceed to retrieve the information. It also include some facility such as user friendly message or pop ups for helping the users.

5. User interface design: saved to gallary

userfriendly and system friendly

A User Interface is a combination of menus, screen design, keyboard commands and language which together create the way a user interacts with the system. It determines how user interacts with system. The hardware part of user interface consists of monitor, keyboard and mouse. The software part of interface

determines what things look like on the screen and how user gives commands to get work done.

Features of good user interface:

i. Know the prospective users of the system and their requirements.

ii. The interface should be robust i.e. the interface should not be fail because of some erroneous actions taken by the user. It requires sufficient checks to prevent the users from taking some false actions.

iii If feasible, graphical icons must be used so that key-strokes can be minimized.

iv. Test the user interface on actual use

v. Practice iterative (repeated) design.

Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Output Design Guidelines:

There are many issues that apply to output design. The following general principles are important for output design:

1,. Computer outputs should be simple to read and interpret :

- Every output should be dated and time-stamped. This helps the reader appreciate the currency of information.
- Reports and screens should include sections and headings to segment
- In form-based outputs, all fields should be clearly labeled.
- In tabular-based outputs, columns should be clearly labeled.
- Information should never have to be manually edited to become usable.
- Users must be able to easily find the output, move forward and backward, and exit the report.
- Computer jargon and error messages should be omitted from all outputs.

2. **The timing of computer outputs is important :** Output information must reach recipients while the information is pertinent to, transactions or decisions. This can affect how the output is designed and implemented.
3. **The distribution of computer outputs must be sufficient to assist all relevant system users :** The choice of implementation method affects distribution.
4. **The computer output must be acceptable to the system users who will receive them :** An output design may contain the required information and still not be acceptable to the system user.
5. **Quality :** This relates to the contents, appearance and accuracy of the output. Outputs generated for external users should be given special attention in respect of its getup, quality of paper etc.

Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Welldesigned input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding –
  - What are the inputs needed for the system?
  - How end users respond to different elements of forms and screens.

Objectives for Input Design

The objectives of input design are –

- To design data entry and input procedures
- To reduce input volume specific & specific data only
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Data collection

Data collection is the systematic approach to gathering and measuring information from a variety of sources to get a complete and accurate picture of an area of interest. Data collection enables a person or organization to answer relevant questions, evaluate outcomes and make predictions about future probabilities and trends.

Data collection Techniques

To study any system the analyst needs to do collect facts and all relevant information. the facts when expressed in quantitative form are termed as data. The success of any project is depended upon the accuracy of available data. Accurate information can be collected with help of certain methods/ techniques. These specific methods for finding information of the system are termed as data collection techniques.

1) Interview
2) Questionnaire
3) Record View and
4) Observations

Are the different **Data collection** techniques used by the analyst. The analyst may use more than one technique for investigation.

1)Interview

This method is used to collect the information from groups or individuals. Analyst selects the people who are related with the system for the interview. In this method the analyst sits face to face with the people and records their responses. The interviewer must plan in advance the type of questions he/ she is going to ask and should be ready to answer any type of question. He should also choose a suitable place and time which will be comfortable for the respondent.

The information collected is quite accurate and reliable as the interviewer can clear and cross check the doubts there itself. This method also helps gap the areas of misunderstandings and help to discuss about the future problems. Structured and unstructured are the two sub categories of Interview. Structured interview is more formal interview where fixed questions are asked and specific information is collected whereas unstructured interview is more or less like a casual conversation where in-depth areas topics are covered and other information apart from the topic may also be obtained.

2)Questionnaire

It is the technique used to extract information from number of people. This method can be adopted and used only by an skillful analyst. The Questionnaire consists of series of questions framed together in logical manner. The questions are simple, clear and to the point. This method is very useful for attaining information from people who are concerned with the usage of the system and who are living in different

countries. The questionnaire can be mailed or send to people by post. This is the cheapest source of fac finding.

3) Record View

The information related to the system is published in the sources like newspapers, magazines, journals, documents etc. This record review helps the analyst to get valuable information about the system and the organization.

4) Observation

Unlike the other fact finding techniques, in this method the analyst himself visits the organization and observes and understand the flow of documents, working of the existing system, the users of the system etc. For this method to be adopted it takes an analyst to perform this job as he knows which points should be noticed and highlighted. In analyst may observe the unwanted things as well and simply cause delay in the development of the new system.

Data Entry process /Data Input Methods:

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by –

- Clear form design by leaving enough space for writing legibly.
- Clear instructions to fill form.
- Clear form design.
- Reducing key strokes. जास्त Type करायची गरज नसली पाहिजे
- Immediate error feedback.

Some of the popular data input methods are –

- Batch input method (Offline data input method)
- Online data input method
- Computer readable forms
- Interactive data input

Database Design

Database design is the process of producing a detailed data model of a **database**. This data model contains all the needed logical and physical **design** choices and physical storage parameters needed to generate a **design** in a data definition language, which can then be used to create a **database**.

The database design phase is divided into three steps:

1) conceptual database design

2) logical database design

3) physical database design

1) conceptual database design

In the conceptual database design phase, the model of the data to be used independent of all physical considerations is to be constructed. The model is based on the requirements specification of the system.

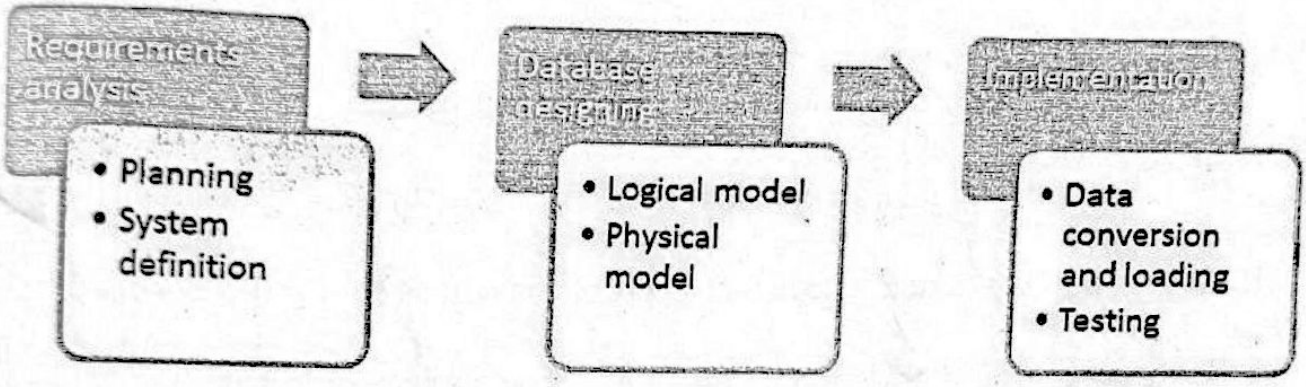
2)logical database design

In the logical database design phase, the model of the data to be used is based on a specific data

model, but independent of a particular database management system is constructed. This is based on the target data model for the database e.g. relational data model.

3)physical database design

In the physical database design phase, the description of the implementation of the database on secondary storage is created. The base relations, indexes, integrity constraints, security, etc. are defined using the SQL language.



Qu 3)

Software design : Software design is the process by which an agent creates a specification of **software** artifact, intended to accomplish goals, using a set of primitive components and subject to constraints.

Module design :

Module is the way to improve the structure design by break down the problem for solving it into independent task.

Advantages of Module –

I. It breakdown the problem into independent modules so the complexity of the problem can be minimized.

II. Each independent module can be easily assigned to the various members of the development team.

III. Module can be easily run and tested independently from another.

Top-Down design approach:-

It is a technique of breakdown a problem into major tasks to be performed. Each task is then further broken down into separate sub task and so on until each sub task is sufficiently simple to be written as a self contained module.

In Top-Down design we initially describes the problem at the highest level that descript what must be done and It does not show how it must be done. Top-Down methods are used throughout the system analysis and design process. The value of using top-down approach, starting at general level and to understand and gain the system and moving down to the levels of greater details.

Advantages of Top-Down Approach –

I. By dividing of the problem into number of sub problems we have made it easier to share problem development.

II. It is easy to debug a large program as a number of smaller units rather than one big problem.

III. It is good way to delay decision on problems whose solution is not readily prepared.

IV. It allows a programmer to remain on top of a problem and view the developing solutions. The solution always proceeds from the highest level to the lowest level.

V. It becomes an ideal structure for managing the implementation of a computer program using team of programmers.

Bottom-Up design approach:-

When we face a large and complex problem , it is difficult to see how the whole thing can be done so it may easier to solve the part of the problem individual, taking the common and easy aspects first and then more difficult task and finally gather them all together to form complete solution, this is called bottom-up approach.

The bottom-up approach suffers from disadvantage that the part of the program may not fit together very easily and there may be a lack of consistency between modules and reprogramming have to be done.

Module Cohesion & Coupling:-

i.) Modules: