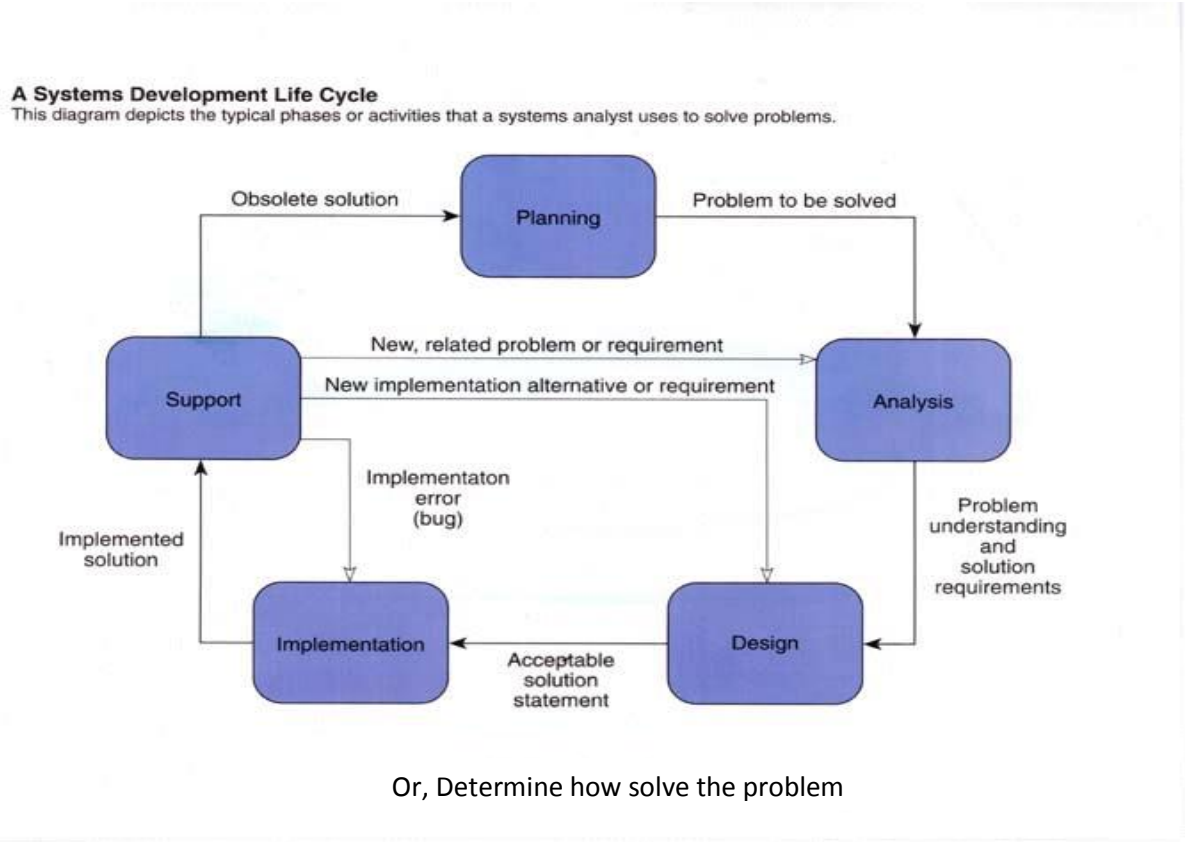
Systems Development Life Cycles:

- The **systems development life cycle** (SDLC) is a general term used to describe the method and process of developing a new information system
- Without the structure and organization provided by SDLC approach projects are at **risk** for missed deadline, low quality etc
- SDLC provides

⇨ SDLC provides

- Structure
- Methods
- Controls
- Checklist



Or, Determine how solve the problem

Systems Investigation and Planning

- Problems and opportunities are identified
- The project planning phase includes **five activities**:

⇨ Define the problem.

⇨ Confirm project feasibility.

⇨ Produce the project schedule.

⇨ Staff the project.

⇨ Launch the project.

Systems Analysis

- Existing systems and work processes are studied
- The analysis phase includes **six activities**:
    - ⇨ Gather information (e.g. interview, read, observe etc.)
    - ⇨ Define system requirements (reports, diagrams etc.)
    - ⇨ Build prototypes for discovery of requirements
    - ⇨ Prioritize requirements
    - ⇒ Generate and evaluate alternative solutions
    - ⇨ Review recommendations with management

Systems Design

- Defines how the information system will do what it must do to solve the problem.
- The design phase includes **seven activities**:
    - ⇨ Design and integrate the network
    - ⇨ Design the application network
    - ⇨ Design the user interfaces
    - ⇨ Design the system interfaces
    - ⇨ Design and integrate the database
    - ⇨ Prototype for design details
    - ⇨ Design and integrate the system controls

Systems Implementation

- System components are assembled and the new or modified system is placed into operation.
- The implementation phase includes **six activities**:
    - ⇨ Construct software components
    - ⇨ Verify and test
    - ⇨ Develop prototypes for tuning
    - ⇨ Convert data
    - ⇨ Train and document

Systems Maintenance and Review

- Ensures the system operates and is modified to keep up with business changes.
- The support phase includes **two**

activities: Provide support to **end users**

- ⇨ Help desks
- ⇨ Training

programs Maintain and enhance

the computer system

⇨ Simple program error correction

⇨ Make sure that the system operates as expected

⇨ Modify functionalities that are not working properly

In "classical" life cycle these phases are sequential, but there are variations

When to Use SDLC

® Requirements are stable and not changed frequently.
® An application is small.
® There is no requirement which is not understood or not very clear.
® The environment is stable.
® The tools and techniques used is stable and is not dynamic.
® Resources are well trained and are available.

Advantages of using the Waterfall model are as follows:

- Simple and easy to understand and use.
- For smaller projects, the waterfall model works well and yield the appropriate results.
- Since the phases are rigid and precise, one phase is done one at a time, it is easy to maintain.
- The entry and exit criteria are well defined, so it easy and systematic to proceed with quality.
- Results are well documented.

Disadvantages of using Waterfall model:

- Cannot adopt the changes in requirements
- It becomes very difficult to move back to the phase. For example, if the application has now moved to the testing stage and there is a change in requirement, It becomes difficult to go back and change it.
- Delivery of the final product is late as there is no prototype which is demonstrated intermediately.
- For bigger and complex projects, this model is not good as a risk factor is higher.
- Not suitable for the projects where requirements are changed frequently.
- Does not work for long and ongoing projects.
- Since the testing is done at a later stage, it does not allow identifying the challenges and risks in the earlier phase so the risk mitigation strategy is difficult to prepare.

When to Use Agile

® There is a project champion of agile methods in the organization

® Applications need to be developed quickly in response to a dynamic environment

® A rescue takes place (the system failed and there is no time to figure out what went wrong)

® The customer is satisfied with incremental improvements

® Executives and analysts agree with the principles of agile methodologies

- ® When new changes need to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- ® To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- ® Unlike the waterfall model in agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- ® Both system developers and stakeholders alike find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

Advantages of Agile model:

1. Customer satisfaction by rapid, continuous delivery of useful software.
2. People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
3. Working software is delivered frequently (weeks rather than months).
4. Face-to-face conversation is the best form of communication.
5. Close, daily cooperation between business people and developers.
6. Continuous attention to technical excellence and good design.
7. Regular adaptation to changing circumstances.
8. Even late changes in requirements are welcomed.

Listed below are a few disadvantages of Agile:

1. In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
2. There is lack of emphasis on necessary designing and documentation.
3. The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
4. Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to Use Object-Oriented

- ® The problems modeled lend themselves to classes
- ® An organization supports the UML learning
- ® Systems can be added gradually, one subsystem at a time
- ® Reuse of previously written software is a possibility
- ® It is acceptable to tackle the difficult problems first

Why object-oriented model is required?

- What are the advantages of object oriented analysis and design over procedure oriented analysis and design?
- How do you do object oriented analysis and design

What are the features of object oriented analysis?

Sl No	Job title	Responsibilities
	Database administrator	Hdfrhjdfgj Hdfjhdftji xdfgjkjhkl

Database administrator

Responsibilities