What are the benefits and importance of following a staged life cycle?

- Simple and easy to understand and follow
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time. Phases do not overlap.
b) Integration Testing  
c) System Testing  
d) Acceptance Testing  

Pros and Cons of the V-Model

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a highly disciplined model and Phases are completed one at a time.</td>
<td>High risk and uncertainty.</td>
</tr>
<tr>
<td>Simple and easy to understand and use.</td>
<td>Not a good model for complex and object-oriented projects.</td>
</tr>
<tr>
<td>Easy to manage due to the rigidity of the model</td>
<td>Poor model for long and ongoing projects.</td>
</tr>
<tr>
<td>each phase has specific deliverables and a review process.</td>
<td>Not suitable for the projects where requirements are at a moderate to high risk of changing.</td>
</tr>
<tr>
<td>Works well for smaller projects where requirements are very well understood.</td>
<td>No working software is produced until late during the life cycle.</td>
</tr>
</tbody>
</table>
### Pros
- Increased user involvement in the product even before implementation.
- Since a working model of the system is displayed, the users get a better understanding of the system being developed.
- Reduces time and cost as the defects can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily.

### Cons
- Risk of insufficient requirement analysis owing to too much dependency on prototype.
- Users may get confused in the prototypes and actual systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Developers may try to reuse the existing prototypes to build the actual system, even when it’s not technically feasible.
- The effort invested in building prototypes may be too much if not monitored properly.

---

5) **Spiral Model**
FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements

User

- View doctors
- View hospital details
- View services

Member

- Login
- Search availability doctors/dates
- Make payment
- View Doctors
- View Hospitals

Operator

- Login
- Add doctor details
- Add appointment details
- Add hospital details
- Edit Doctor details
- Edit Hospital details
- View reports

Hospital Operator

- Make appointment
- Make payments

Administrator
Task 6

1) CONTEXT DIAGRAM
2) LEVEL 0 DFD
3) LEVEL 1 DFD
4) USE CASE DIAGRAM
5) CLASS DIAGRAM
Following are the functional user requirements.

User

- View doctors
- View hospital details
- View services

Member

- Login
- Search availability doctors/dates
- Make payment
- View Doctors
- View Hospitals

Operator

- Login
- Add doctor details
- Add appointment details
- Add hospital details
- Edit Doctor details
- Edit Hospital details
- View reports

Hospital Operator

- Make appointment
- Make payments

Administrator

- Create users
- Edit user information
REFERENCE


