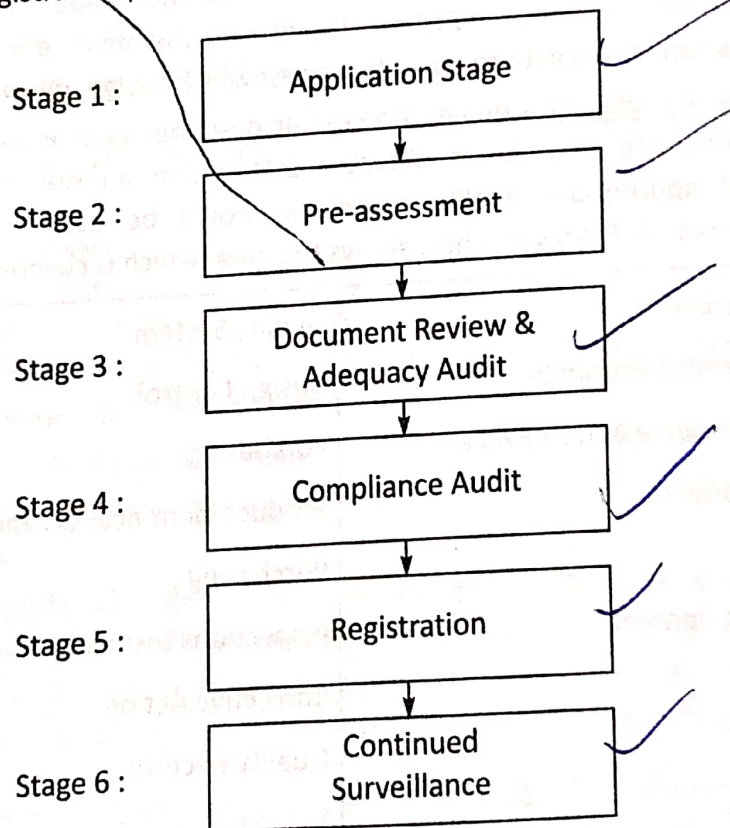


### ✓ ISO 9000 for Software Industry/Process of getting ISO Certification

An organization interested to obtain the ISO 9000 certification applies to as ISO registrar for registration. The ISO 9000 registration process consists of following stages.



#### Stage 1 : Application Stage

(When an organization decides to go for the ISO 9000 certification, it applies to a registrar for registration.)

#### Stage 2 : Pre-assessment

(In this stage, the registrar makes a rough assessment of the organization.)

**Stage 3 : Document Review and Adequacy Audit**

(At this stage, the registrar reviews the process, related documents of the organization and makes suggestions for possible improvements.)

**Stage 4 : Compliance Audit**

(At this stage, the registrar checks whether the suggestions made during review have been complied with by the organization or not.)

**Stage 5 : Registration**

(The registration awards the ISO 9000 certificate after successful completion of all the previous phases by the organization.)

**Stage 6 : Continued Surveillance**

(The registrar continues to monitor the organization periodically.)

**✓ Requirements of ISO 9001**

The main requirements of ISO 9001 as related to software development are as follows :

**1. Management Responsibility**

- Management must have an effective quality policy.
- The responsibility and authority of all those whose work affect quality must be defined and documented.
- A management representative, who is not part of the development process must be responsible for quality system. This type of requirement enables the person responsible for the quality system to work in an unbiased manner.
- The effectiveness of the quality system must be reviewed periodically by audit.

**2. Quality System**

A quality control system must be maintained and documented.

**3. Contract Reviews**

Contract must be ensured that the concerned organization has the necessary capability to carry out its contractual obligations.

**4. Design Control**

Design process must be properly controlled, including the control of the coding process. The requirement means that a good configuration control system must be in place.

- Design inputs must be verified and confirmed adequately.
- Design must be verified.
- Design output must be of required quality.
- Design changes must be controlled.



### 5. Document Control

- There must be proper procedure for document approval, issue and removal.
- Document changes must be controlled.

### 6. Purchasing

Purchased materials, including bought in software must be checked for conformity to requirements.

### 7. Product Identification

The product must be identifiable at all stages of the process.

### 8. Process Control

- The development must be properly managed.
- Quality requirements must be identified in a quality plan.

### 9. Inspection and Testing

This requires effective testing which means unit testing, integration testing and system testing. Test records must be documented.

### 10. Inspection, Measuring and Test Equipment

If inspection, measuring and test equipment and used, they must be properly maintained and elaborated.

### 11. Inspection and Test Status

The status of an item must be identified.

### 12. Control of Non-Conforming Product

It means keeping untested or faulty software out of the released product.

### 13. Corrective Action

This is both about correcting errors when occur, reasons for their occurrence is investigating and improving the process to prevent occurrence.

### 14. Handling

Handling deals with the storage, packing and delivery of the software product.

---

## ✓ Capability Maturity Model

---

Capability maturity model is the outcome of decades of research and study of successful and unsuccessful projects. The major philosophy of CMM is very similar to life itself. When a child is born it is at a very 'initial' level of maturity. The child grows up, learns and attains a higher level of maturity. This keeps on going until he/she becomes a fully mature adult, and even after that the learning goes on. According to CMM, a software company also goes through similar maturity evolutions.

The Software Engineering Institute (SEI) capability maturity model was proposed by the software engineering institute of Carnegie Mellon University, USA. The SEI CMM model was originally developed to