1. **What is Software Cost Estimation?**

**-** Software cost estimation means calculating the money, time, and effort needed to build a software project.It's based on:

* Past experience
* Available documents
* Assumptions
* Identified risks

1. **What are the components of software cost?**

**-**

* Hardware/software cost (example: tools, devices)
* Training/travel cost (example: sending team to workshop)
* Effort cost (example: developer salaries,building cost,network & communication costs,social & insurance cost)

Here, Effort cost is dominant cost because of developer salaries,office space,electricity etc.

1. **What are the main factors that affect software pricing?**

**-**

| **Factor** | **Description** |
| --- | --- |
| **Market Opportunity** | Price may be lower to grab a market chance or new customer. |
| **Cost Estimate Uncertainty** | If cost is unsure, extra money is added as a safety. |
| **Contractual Terms** | Fixed deadlines or conditions can raise the price. |
| **Requirements Volatility** | Changing needs increase time, effort, and cost. |
| **Financial Health of Customer** | Weak budget may lead to offering lower price. |

1. **What is algorithmic cost modeling?**

**-** Algorithm cost modelling is a mathematical method to estimate project cost.

* Uses mathematical equations to estimate software cost.
* Depends on software size and other cost factors.
* Cost increases faster than size because of team coordination, complexity, etc.
* Common in models like **COCOMO**.
* An algorithmic cost estimate for software cost can be expressed as:

**Effort = A ∗Size^B ∗ M**

**Effort -** Total amount of work needed to develop the software . Measured in Person-Months (PM)

* Example: 100 PM = 10 people working for 10 months.

**A -** A fixed number based on project type (like organic, semi-detached, or embedded) **.** It sets the base level of effort

* Think of it like a “starting point” for how hard the project will be.

**Size -** The size of the software, usually measured in KLOC

* Bigger size = more work = more cost

**B -** This shows how effort increases as the size increases .

* It makes the formula non-linear

**M -** Represents other factors that affect productivity like:

* Team experience
* Tool support
* Reliability requirement
* Development environment
* These are called cost drivers. They adjust the effort based on real project conditions.

1. **What are the problems with algorithmic models?**

**-**

* Size is hard to estimate early in the project.
* Function-point or object-point estimates are better but still not exact.
* Inputs like B and M (in formulas) are subjective—they vary by person.
* Estimation depends on experience and guesswork, so results differ.

1. **Types of COCOMO models?**

**-**

**Organic mode**: Small teams, Simple project, Experienced People.  
Example: A group of experienced people decorating a small birthday party. They know what to do, so it goes smoothly.

**Semi-detached mode**: Mix of skilled and new team members.  
Example: Organizing a school function where some teachers have done it before, but some students are new to managing events.

**Embedded mode**: Complex hardware + software; strict rules  
Example: Setting up a hospital emergency system where machines, power supply, and staff must all work together under strict rules.

1. **Why COCOMO 2 sub-model introduced?**

**-**

**Application Composition Model** – Based on reusable components.  
>> Used when building software by combining ready-made parts like templates or modules.

**Early Design Model** – Before design starts.  
>> Used at the beginning stage when only requirements are known, and full design hasn’t started yet.

**Reuse Model** – For effort needed to reuse software.  
>> Helps to estimate time and cost when using already-built software in a new project.

**Post-architecture Model** – After design is available.  
>> Used when the full system design is ready, and detailed planning and development can begin.

1. **What is Software Quality Assurance (SQA)?**

-

* **Makes Sure Software is Good:**

SQA is a process that checks if the software meets quality rules and works properly.

* **Focuses on the Right Way of Working:**

It ensures the team follows correct steps and methods while building the software.

* **Stops Problems Early:**

SQA helps find and fix mistakes early, so they don’t cause bigger issues later.

* **Uses Checks and Reviews:**

It includes doing code reviews, testing, and audits to keep everything on track.

* **Works in All Project Steps:**

SQA is used from the beginning of the project to the end, to keep quality high at every stage.

1. **What is Six Sigma in QA?**

**-**

* **Define customer needs:**

Understand what the customer wants from the software.

* **Measure current performance:**

Check how the software or process is working right now.

* **Analyze defects:**

Find out what problems are happening and why they are happening.

* **Improve by removing root causes:**

Fix the main reasons behind the problems to make the software better.

* **Control to avoid future issues:**

Keep checking the process to make sure the same problems don’t happen again.

1. **Core steps of improve or develop the Six Sigma**

-

**Design defect-free process**  
Create a process in a way that problems don’t happen from the beginning.

**Verify process meets needs**  
Check if the process really works well and satisfies customer requirements.

1. **What is ISO 9000?**

**-** A set of international quality management standards to ensure consistent product quality.

Not limited to software – can be used in any industry.

1. **Principles of ISO 9000.**

**-**

**Customer focus** – Make customers happy by giving them what they need.

**Leadership** – Good leaders guide the team with clear goals.

**Engagement of people** – Everyone should be involved and do their part well.

**Process approach** – Do work step by step in an organized way.

**Improvement** – Always try to make things better.

**Evidence-based decision making** – Take decisions based on facts, not guesses.

**Relationship management** – Keep good connections with others like suppliers or partners.

1. **What is SEI CMM model?Explain the maturity level of CMM model.**

**- Definition:** A method to measure the maturity of a software company’s processes (1 to 5 scale).

**Level 1 – Initial** – Work is performed informally  
 Example: A small team builds software without any planning, just based on what they feel like doing.

**Level 2 – Repeatable** – Work is planned & tracked  
Example: A company uses past project plans to manage similar new projects, like reusing a checklist.

**Level 3 – Defined** – Work is well-defined  
 Example: The team follows a written process for every project step like coding, testing, and delivery.

**Level 4 – Managed** – Work is quantitatively controlled  
 Example: The company uses tools to measure bugs, time, and productivity with charts and reports.

**Level 5 – Optimizing** – Work is based upon continuous improvement  
Example: The company regularly reviews feedback and updates its process to work faster and better.

1. **Difference between ISO 9000 & CMM**

**-**

| **ISO 9000** | **CMM** |
| --- | --- |
| Industry-independent | Software industry focused |
| Can be used in any type of company | Mainly used in software companies |
| Focuses on quality system | Focuses on process maturity |
| Makes sure a company follows quality rules | Helps a company grow step by step in process |
| External certification (given by outside agency) | Internal improvement (used for self-growth) |