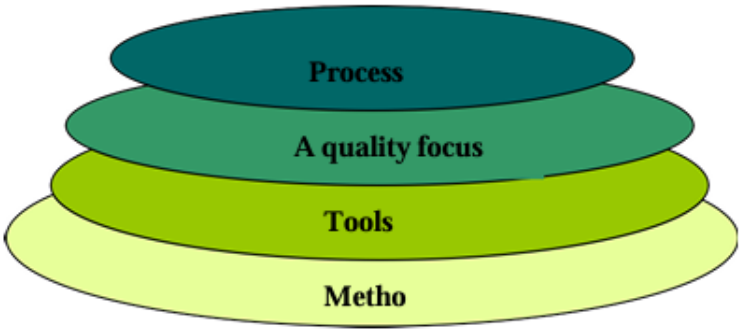
By Maha Rana

CS615 Finals Subjective Solved

1. Software Engineering Layers (Long Question)

Software engineering is a **layered technology**, composed of the following layers:

Figure 2: Software engineering layers



2. WBS: Work, Benefits, and Related Concepts

Benefits of WBS:

- Defines 100% of the work per scope.
- Helps manage cost, schedule, and responsibilities.
- Tracks progress and aids in communication with stakeholders.
- Serves as a **budgeting and reporting tool**.
- Allows for **task delegation and control** over time

By Maha Rana

Using a WBS provides a number of benefits to the management and to the development teams.

First, it gives the management an idea about the size and complexity of the project.

Second, it helps in planning, scheduling, and monitoring a project realistically. This is possible because all the tasks in the project can be preformed measurable targets for each task.

To aid planning, scheduling, and monitoring a project, you can use tools such as:

- Program Evaluation and Review Techniques (PERT)
- Critical Path Method (CPM)
- Timeline charts
- Gantt charts

These tools use WBS as the fundamental basis for assessing resources to tasks, computing the number of days needed, and the cost required to complete the tasks.

Characteristics of High-Quality WBS:

- Hierarchical and logical structure.
- Must receive team buy-in.
- Deliverable-oriented and clear.
- Should be in sync with scope and project plan.
- Updated regularly to reflect current status.

WBS Conceptual:

- A **functional decomposition** of the entire project.
- Bridges between project **scope and schedule**.
- Includes **all deliverables, internal/external/interim**.
- Produced **after the scope statement** but before detailed scheduling.

□ 3. Four Cornerstones of Management

1. **Delegation**
2. **Authority**
3. **Responsibility**
4. **Supervision**
   These ensure a structured and accountable project organization.

□ 4. Review Techniques

By Maha Rana

1. **Inspections**
   - Formal, structured reviews.
   - Roles: moderator, author, reader, recorder, inspector.
2. **Walkthroughs**
   - Informal, developer-led reviews.
   - Less rigorous but encourage discussion.
3. **Formal Technical Reviews (FTR)**
   - Reviews logic, consistency, standards adherence.
   - End with summary reports and defect lists.

□ 5. COCOMO Model: Intermediate Form

Intermediate COCOMO uses:

Effort = a_1 × (KDSI)^a_2 × EAF

Where:

- a_1 and a_2 are model constants based on project type.
- KDSI = Thousands of Delivered Source Instructions.
- EAF = Effort Adjustment Factor (based on cost drivers).

Make sure you're given values for a_1, a_2, and SLOC (convert to KDSI) in the question to compute this

□ 6. Initial Process: Inputs, Outputs, Tasks

Initiation Phase Includes:

- **Input:** Business case, project charter.
- **Output:** Project approval, stakeholder identification.
- **Tasks:** Define goals, select team, assess feasibility, identify risks.

□ 7. Activities of SQA for Quality Control

- **Software Reviews:** Identify defects early (inspection, walkthrough, FTR).
- **Quality Metrics:** Estimate cost, monitor productivity.
- **SQA Audits:** Ensure adherence to process and documentation.
- **Configuration Management (SCM):** Track changes, maintain integrity.
- **Reporting & Records:** Maintain logs for future audits and improvements.

✅1. Initial Estimates – Formula + Example

Formula for Initial Estimation (Effort):

Effort = a_1 × (SLOC/1000)^a_2 × EAF

By Maha Rana

Where:

- a_1 and a_2 are **project-type constants** (organic, semi-detached, embedded)
- SLOC = Source Lines of Code
- EAF = Effort Adjustment Factor (from cost drivers)
  This formula is from **Intermediate COCOMO Model**.

Example (Assume values from the question):

- SLOC = 32,000 → KDSI = 32
- $a_1 = 3.0$
- $a_2 = 1.12$
- EAF = 1.15

Effort = 3.0 × (32)^1.12 × 1.15 = ~147 person-months

 Reference: Page with COCOMO Intermediate Formula explanation

✓2. SDP (Software Development Plan) – Any 3 Activities

According to the handout, SDP includes:

1. **Project Planning** – Define scope, resources, schedules, and risk assessment.
2. **Software Requirements** – Clarify and document what the software must do.
3. **Software Design and Implementation** – Convert requirements into system/software architecture and code.

 Reference: Activities under SDP

✓3. QA (Quality Assurance) – Any 3 Activities

1. **Formal Technical Reviews (FTR):**
   - Detect and remove defects early.
2. **Software Quality Metrics:**
   - Analyze productivity and defect trends.
3. **Software Configuration Management (SCM):**
   - Track versions, manage changes.

 Reference: QA Activities

✓4. Maintenance Activities (Long Question)

Maintenance includes:

1. **Corrective Maintenance** – Fixing bugs post-deployment.

2. **Adaptive Maintenance** – Updating software for changes in the environment.
3. **Perfective Maintenance** – Improving performance or maintainability.
4. **Preventive Maintenance** – Future-proofing or cleaning up legacy issues.

 Reference: Maintenance types mentioned under software lifecycle

✅5. Primary Planning Tasks

Planning tasks in software project management include:

1. **Scope Definition** – What the project must achieve.
2. **Task Scheduling** – Breaking down deliverables into WBS.
3. **Effort and Cost Estimation** – Using models like COCOMO.
4. **Resource Allocation** – Assigning tools, people, budget.
5. **Risk Management** – Identify and plan for potential failures.

 Reference: Planning-related explanations

6. Categories in Typical WBS

Typical WBS (Work Breakdown Structure) includes these **categories**:

1. **Project Management** – Coordination, reporting, meetings.
2. **Product Requirements and Design** – Analysis and architecture.
3. **Implementation** – Coding, unit testing.
4. **Integration and Testing** – System testing, bug fixing.
5. **Deployment and Maintenance** – Final delivery, support.
6. **Documentation and Training** – Manuals, user training.

 Reference: WBS categories and examples

✅3 Marks Questions

1. Three Activities of Software Quality Assurance (SQA)

1. **Formal Technical Reviews (FTR):**
   Structured examination of work products to find and fix defects.
2. **Software Configuration Management (SCM):**
   Version control and change management.
3. **Software Quality Metrics:**
   Track performance and defect rates using measurable indicators.

 Reference: [10:4†CS615 highlighted.pdf]

2. Three Risk Projection Activities

By Maha Rana

1. **Estimate the Probability of Risk Occurrence**
2. **Estimate the Risk Impact** – Cost, time, resources affected
3. **Assess Risk Exposure** – Based on probability × impact

 Reference: Risk Management & Projection section

3. Two Activities in Software Project Management

1. **Project Planning:**
   Defining scope, effort, cost, and scheduling.
2. **Monitoring & Controlling:**
   Tracking progress, making adjustments as needed.

 Reference: Project phases and management activities

4. Three Techniques for Accurate Estimation

1. **Expert Judgment**
2. **Delphi Technique** (multiple experts anonymously)
3. **COCOMO Model** (constructive cost model)

 Reference: Estimation techniques section

4. Name of Three Types of Prototype Methods

There are different types of prototyping methods that an organization can implement:

- Rapid prototyping
- Reusable prototyping
- Modular prototyping

 Reference: Rapid Prototyping explained

✓5 Marks Questions

6. Five Types of WBS

1. **Phase-Oriented WBS** – Based on SDLC phases (e.g., design, coding).
2. **Deliverable-Oriented WBS** – Based on project outputs.
3. **Functional WBS** – Grouped by departments/functions.
4. **Geographical WBS** – Based on location (e.g., international rollouts).
5. **Time-Based WBS** – Milestone or duration-based.

 Reference: WBS structure and types

6. Five Schedule-Related Constraints in Microsoft Project

In Microsoft Project, you can specify constraints as limitations that are enforced on a task. For example, you can specify that a task must start on a particular date or not finish later than a particular date. Therefore, schedule-related constraints are available in Microsoft Project.

- As soon as possible (ASAP)
- As late as possible (ALAP)
- Start no later than (SNLT)
- Finish on later than (FNLT)
- Start no earlier than (SNET)
- Finish no earlier than (FNET)
- Must start on (MSO)
- Must finish on (MFO)

 Reference: Microsoft Project constraints section

7. Name All Four Tasks of Project Initiation Process

The tasks performed for project initiation are mentioned below:

Requirement gathering: The first task is to gather the customer requirements. Customer requirements may be spoken or unspoken. Therefore, the challenge for the project manager is to elicit the requirements in such a way that both the spoken and unspoken customer needs and wants are gathered. After collecting the required information, you need to translate the customer requirements into technical specifications for the software project.

Scope determination: The scope of a software project can be defined as the combination of the software product arid services to be delivered to the customer. You carry out the scope determination exercise to define the scope of the software project. The scope determination exercise enables you to refine and understand the customer requirements. You can refine the scope definition further by breaking down each deliverable into smaller and more manageable activities. The scope determination exercise also helps you identify the technology for creating the software product.

Resource allocation: During project initiation, you identify the resources required and allocate them to the software project. The resources identified may be people, reusable software components, and hardware or software tools. You allocate the resource to the software project on the basis of the activities defined in the scope determination exercise. While allocating appropriate resources for a software project, you also need to calculate the cost of each resource. The cost of a resource is calculated according to the duration of the resource in the software project. Estimating the cost of resources also helps you prepare a budget for the software project.

By Maha Rana

Initial project plan: Another exercise that you carry out during project initiation is the creation of a rough project plan. This plan is a draft version and carries only the primitive project plan features. This project plan carries the initial risk analysis of the software project, the initial start and end dates, the duration of the activities in the project, and the sequencing of these activities.

Planning process task

⇒ **Planning Process Tasks**

1. **Scope Planning**
2. **Scope Definition**
3. **Activity Definition**
4. **Activity Sequencing**
5. **Activity Duration Estimating**
6. **Resource Planning**
7. **Cost Estimating**
8. **Cost Budgeting Risk Planning**
9. **Schedule Development**
10. **Quality Planning**
11. **Communications Planning**
12. **Organization Planning**
13. **Staff Acquisition**
14. **Procurement Planning**
15. **Project Plan Development**

Project Planning objective:

4.1 Project Planning Objectives

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of:

- **Resources**
- **Cost, and**
- **Schedule**

Project planning key task:

- Set goal and scope
- Select lifecycle
- Set org teams
- Start team selection
- Determine risk
- Create wbs
- Identify task
- Estimate size
- Estimate effort

Primary Planning steps in planning phase

- Project Plan
- Resource plan
- Financial plans
- Quality Plan
- Risk Plan
- Acceptance plan
- Communication plan

Input to SDP (Software development plans)

1. Other Planning outputs
2. Historical info
3. Organization polices
4. constraint
5. assumption

Tool and techniques of SDP?

1. Project planning methodology
2. Stakeholder skills knowledge and experience
3. Project management info sys (PMIS)
4. Earned Value Management

Output from SDP

1.Project plan

1. Input to SDP execution
2. Project plan
3. Support details
4. Organizational policies
5. Prevention action
6. Corrective action

Output from SDP execution

- Work results
- Change request

Element of software project plan

1. Scope planning
2. Objective/business requirement
3. Technical Approach
4. Contractual aspect

Waterfall model & prototyping model:

1. **The *Waterfall* model**: This is the traditional life cycle model. It assumes that all phases in a software project are carried out sequentially and that each phase is completed before the next is taken up.
2. **The *Prototyping* Model**: A model that works on an iterative cycle of gathering customer requirements, producing a prototype based on the requirement specifications, and getting the prototype validated by the

customer. Each iteration of the life cycle builds on the prototype produced in the previous iteration.

3. **The *Incremental* Model**: The Incremental model is an example of an evolutionary life cycle model. It combines the linear nature of the Waterfall model and the iterative nature of the Prototyping model. The Incremental model divided the development life cycle into multiple linear sequences, each of which produces an increment of the final software product. In this model, the software product is developed in builds. A build is defined as a self-contained unit of the development activity. The entire development cycle is planned for a specific number of logical builds, each having a specific set of features.
4. **The *Spiral* model**: Another evolutionary life cycle model that combines the linear nature of the Waterfall model and the iterative nature of the Prototyping model. The project life cycle is divided into phases, and each phase is executed in all of the iteration of the Spiral Model.
5. **The RAD Model**:

A Process defines the overall processes that an organization needs to follow to manage a project efficiently. It defines a structured approach to sequence the phases and identify the requirements of each phase in the SDLC. The definition of the phases in a process model helps organize, monitor, and execute a project efficiently.

Drawback of waterfall model

- There cannot be a sudden crossover from one phase to next
- Another reason for the unpopularity of the waterfall model is that it requires too much effort for stringent documentation in every phase.
- No working model is developed until the end. Client feedback is only considered after the project is complete
- Finally, the waterfall model causes as a “blocking state’’

Phases of waterfall model

- Analysis
- Design
- Coding
- Testing
- Implementing & Design

- You should use prototyping model when:
- The client is not clear about the requirement of the proposed system or
- When there is a conflict in client requirement

Types of organizational system:

1. Project hosted org
2. Non project based org

Organizational cultural and style

- Shared values
- Norms
- Beliefs
- Expectation
- Policies and
- Procedures

Inputs to Organizational Planning

i. **Project interfaces.** Project interfaces generally fall into one of three categories:

- ***Organizational interfaces***—formal and informal reporting relationships among different organizational units. Organizational interfaces may be highly complex or very simple. For example, developing a complex telecommunications system may require coordinating numerous subcontractors over several years, while fixing a programming error in a system installed at a single site may require little more than notifying the user and the operations staff upon completion.

- ***Technical interfaces***—formal and informal reporting relationships among different technical disciplines. Technical interfaces occur both within project phases (e.g., the site design developed by the civil engineers must be compatible with the superstructure developed by the structural

engineers) and between project phases (e.g., when an automotive design team passes the results of its work along to the retooling team that must create the manufacturing capability for the vehicle).

- ***Interpersonal interfaces***—formal and informal reporting relationships among different individuals working on the project. These interfaces often occur simultaneously, as when an architect employed by a design firm explains key design considerations to an unrelated construction contractor's project management team.

Tool and technique for organizational planning

- **Template**
- **Human resources practices**
- **Organizational theory**
- **Stockholder analysis**

Output of organizational planning

- Roles and responsibility assignment
- Staffing management plan
- Org charts

Microsoft Proj type risk

- General
- Milestone
- Recuring

Software Project type risk

1. Development process risk
2. Product related risk

Pert and cpm common things

1. Activities
2. Nodes
3. Critical path
4. Networks

By Maha Rana

Networking scheduling technique

1. Pert
2. cpm

Guidelines for creating software project schedule

1. Classification
2. Interdependence
3. Time and effort allocation
4. Validation
5. Defined responsibilities and output

Create schedule with these proj management tools

- Gantt chart
- Network scheduling

Output form staff acquisition

Project staff assigned
Project team directory

SOW Template:

Scope of work: Describes the work to be done in detail.
Location of work: Describe where the work must be performed.
Period of performance: specify when the work is expected to start and end.

By Maha Rana

May Allah grant you success, ease your efforts, and bless you with wisdom and perseverance. Keep your faith strong, for with prayer, every challenge becomes an opportunity. And please remember me in your prayers.

I hope this file helps you a lot; I have tried my best to find the most relevant and authenticated answers for the questions. However, in some cases, the questions might slightly differ from the ones in your slides. I recommend that every student kindly verify the answers on their own before using them. I am not responsible for any discrepancies. Best wishes for Finals!