Jitendra bhatta

BESE 8th (161747)

SPM(Assignment chapter 2)

Q1) Define software process life cycle phase.

Software Process Life Cycle Phases:

- **Software life cycle** models describe **phases** of the **software cycle** and the order in which those **phases** are executed.
- Each **phase** produces deliverables required by the next **phase** in the **life cycle**.
- Requirements are translated into design.
- Code is produced according to the design which is called **development phase**

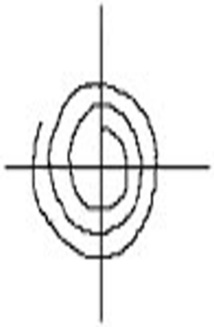
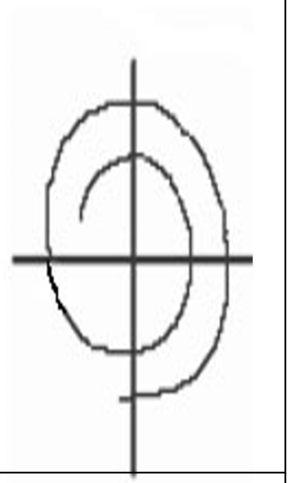
Life-Cycle Phases:

- ➢ Engineering and Production Stages
- ➢ Inception Phase
- ➢ Elaboration Phase
- ➢ Construction Phase
- ➢ Transition Phase

Life-Cycle Phases [Engineering and Production Phase/Stages]

- ❑ **Two stages of the life-cycle :**
    - ❑ ***The engineering stage*** – driven by smaller teams doing design and synthesis ***.*** ***The production stage*** – driven by larger teams doing construction, test, and deployment activities
- The most conventional life cycles, the phases are named after the primary activity within each phase :
    - o Requirement analysis, design, coding, unit test, integrationtest, and system test
- So,conventional software development efforts sequential process, in which one activity was required to be complete before the next was begun.
- Attributing only two stages to a life cycle is too coarse

The phases of the Life Cycle-Process:

Engineering Stage		Production Stage	
Inception	Elaboration	Construction	Transition
			
Idea	Architecture	Beta Releases	Products

Inception Phase:

- Overriding goal – to achieve concurrence among stakeholders on the life-cycle objectives
- Essential activities :
  - Formulating the scope of the project (capturing the requirements and operational concept in an information repository)
  - Synthesizing the architecture (design trade-offs, problem space ambiguities, and available solution-space assets are evaluated)
  - Planning and preparing a business case (alternatives for risk management, iteration planes, and cost/schedule/profitability trade-offs are evaluated)

Elaboration Phase:

- During the elaboration phase, an executable architecture prototype is built
- Essential activities :
  - *Elaborating the vision* (establishing a high-fidelity understanding of the critical use cases that drive architectural or planning decisions)
  - *Elaborating the process and infrastructure* (establishing the construction process, the tools and process automation support)

  - ➢ *Elaborating the architecture and selecting components* (lessons learned from these activities may result in redesign of the architecture)

Construction Phase:

- □ During the construction phase :

All remaining components and application features

are integrated into the application

All features are thoroughly tested

- □ Essential activities :
  - ➢ Resource management, control, and process optimization
  - ➢ Complete component development and testing against evaluation criteria
  - ➢ Assessment of the product releases against acceptance criteria of the vision

Transition Phase:

- □ The transition phase is entered when baseline is mature enough to be deployed in the end-user domain
- □ This phase could include beta testing, conversion of operational databases, and training of users and maintainers
- □ Essential activities :
  - ➢ Synchronization and integration of concurrent construction into consistent deployment baselines
  - ➢ Deployment-specific engineering (commercial packaging and production, field personnel training)

Q2) Explain about the various phases of software Development life cycle.

- SDLC is a process which defines the various stages involved in the development of software for delivering a high-quality product.

Given below are the various phases of SDLC:

- Requirement gathering
- Analysis
- Design
- Coding/Implementation
- Testing
- Final Implementation

- Testing/Deployment
- Maintenance

#1) Requirement Gathering and Analysis

- During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.
- Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.
- **For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.
- Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.
- Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

#2) Analysis:

In the analysis phase, end user business requirements are analyzed and project goals converted into the defined system functions that the organization intends to develop. The three primary activities involved in the analysis phase are as follows:

1. Gathering business requirement
2. Creating process diagrams
3. Performing a detailed analysis

Business requirement gathering is the most crucial part at this level of SDLC. Business requirements are a brief set of business functionalities that the system needs to meet in order to be successful. Technical details such as the types of technology used in the implementation of the system need not be defined in this phase. A sample business requirement might look like “The system must track all the employees by their respective department, region, and the designation”. This requirement is showing no such detail as to how the system is going to implement this requirement, but rather what the system must do with respect to the business.

#3) Design

- In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

#4) Implementation or Coding

- Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

#5) Testing

- Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.
- Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

#6) Deployment

- Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.
- In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

#7) Maintenance

- After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

Q3) Explain about the software Architecture on the basis of management perspective and technical perspective.

Software Architectures: A Management Perspective

❑ From a management perspective, there are three different aspects of an architecture :

➔ An *architecture* (the intangible design concept) is the design of software system, as different or similar to design of a component.

➔ An *architecture baseline* (the tangible artifacts) is a slice (portion) of information across the engineering artifact sets sufficient to satisfy all stakeholders that the vision can be achieved within the parameters of the business case (cost, profit, time, people).

➔ An ***architecture description*** (a human-readable representation of an architecture) is an organizes subsets of information extracted from the design set model.

The importance of software architecture can be summarized as follows:

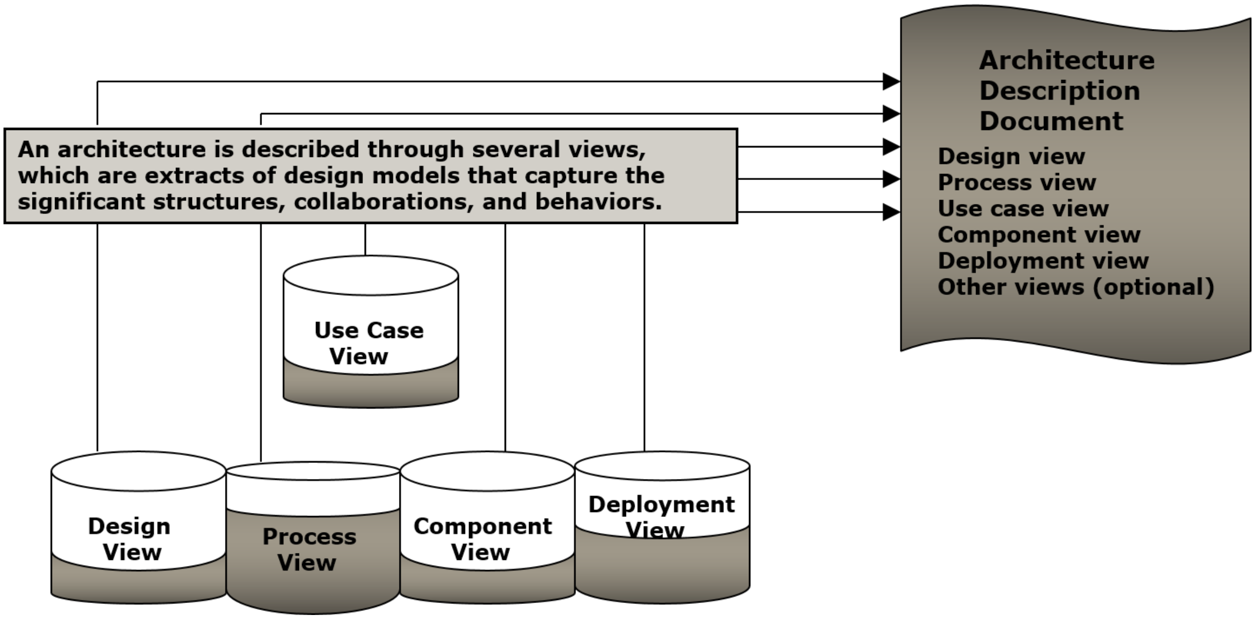
➢ Architecture representations provide a basis for balancing the trade-offs between the problem space and the solution space.

➢ Poor architectures and immature processes are often given as reasons for project failures.

- A mature process, an understanding of the primary requirements, and a demonstrable architecture are important prerequisites for predictable planning.
- Architecture development and process definition are the intellectual steps that map the problem to a solution without violating the constraints.

Software Architectures: A Technical Perspective:

The model which draws on the foundation of architecture developed at *Rational Software Corporation* and particularly on Philippe Kruchten's concepts of software architecture :


An architecture is described through several views, which are extracts of design models that capture the significant structures, collaborations, and behaviors.
Architecture Description Document
Design view
Process view
Use case view
Component view
Deployment view
Other views (optional)
Use Case View
Design View
Process View
Component View
Deployment View

- The *use case view* describes how the system's critical use cases are realized by elements of the design model. It is modeled statically using case diagrams, and dynamically using any of the UML behavioral diagrams.
- The *design view* addresses the basic structure and the functionality of the solution.
- The *process view* addresses the run-time collaboration issues involved in executing the architecture on a distributed deployment model, including the logical software network topology, interprocess communication and state management.
- The ***component view*** describes the architecturally significant elements of the implementation set and addresses the software source code realization of the system from perspective of the project's integrators and developers.
- The ***deployment view*** addresses the executable realization of the system, including the allocation of logical processes in the distribution view to physical resources of the deployment network.

Q4) Define software process Workflows.

Software Process Workflows:

❑ The term *workflow* is used to mean a thread of cohesive and most sequential activities.

There are seven top-level workflows:

1. Management workflow: controlling the process and ensuring win conditions for all stakeholders

2. Environment workflow: automating the process and evolving the maintenance environment

3. Requirements workflow: analyzing the problem space and evolving the requirements artifacts

4. **Design** workflow: modeling the solution and evolving the architecture and design artifacts

5. **Implementation** workflow: programming the components and evolving the implementation and deployment artifacts

6. **Assessment** workflow: assessing the trends in process and product quality

7. **Deployment** workflow: transitioning the end products to the user

Q5) what are the four basic key principles of software process workflows?

Four basic key principles:

1. Architecture-first approach:

Implementing and testing the architecture must precede full-scale development and testing and must precede the downstream focus on completeness and quality of the product features.

2. Iterative life-cycle process:

The activities and artifacts of any given workflow may require more than one pass to achieve adequate results.

3. Roundtrip engineering:

Raising the environment activities to a first-class workflow is critical; the environment is the tangible embodiment of the project's process and notations for producing the artifacts.

4. Demonstration-based approach:

Implementation and assessment activities are initiated nearly in the life-cycle, reflecting the emphasis on constructing executable subsets of the involving architecture.

Q6) Discuss about the software process checkpoints and milestone.

– Software Process Checkpoints and Milestone

Checkpoints of the Process:

❑ It is important to have visible milestones in the life cycle , where various stakeholders meet to discuss progress and planes.

The purpose of this events is to:

- Synchronize stakeholder expectations and achieve concurrence on the requirements, the design, and the plan.
- Synchronize related artifacts into a consistent and balanced state
- Identify the important risks, issues, and out-of-rolerance conditions
- Perform a global assessment for the whole life-cycle.

Three types of joint management reviews are conducted throughout the process:

1.Major milestones –provide visibility to system-wide issues, synchronize the management and engineering perspectives and verify that the aims of the phase have been achieved.

2. Minor milestones – iteration-focused events, conducted to review the content of an iteration in detail and to authorize continued work.
3. Status assessments – periodic events provide management with frequent and regular insight into the progress being made.