

Software Project Management

Course Contents:

Unit 02: Software Process Primitives and Process Management Framework

- 2.1 Software Process Life Cycle Phases**
- 2.2 Various Elements of the Software Process (Management, Engineering and Pragmatic)**
- 2.3 Technical and Management Perspective of Software Architecture**
- 2.4 Software Process Workflow and Iteration Workflow**
- 2.5 Status Monitoring – Software Process Checkpoints and Milestone**

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]

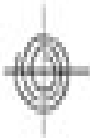
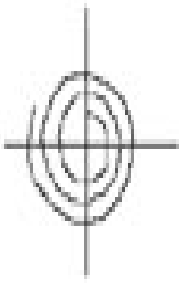
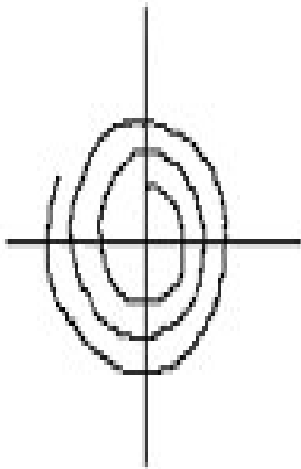
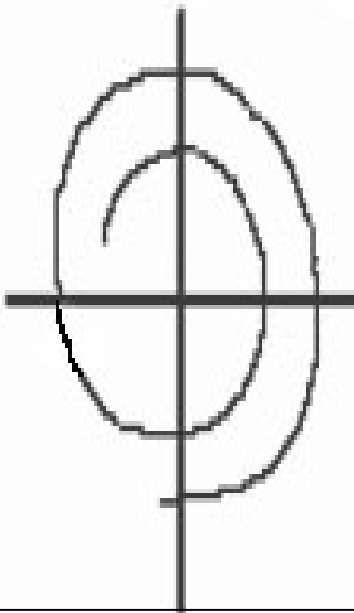
❑ Two stages of the life-cycle :

- 1. The engineering stage*** – driven by smaller teams doing design and synthesis activities
- 2. The production stage*** – driven by larger teams doing construction, test, and deployment activities

LIFE-CYCLE ASPECT	ENGINEERING STAGE EMPHASIS	PRODUCTION STAGE EMPHASIS
Risk reduction	Schedule, technical feasibility	Cost
Products	Architecture baseline	Product release baselines
Activities	Analysis, design, planning	Implementation, testing
Assessment	Demonstration, inspection, analysis	Testing
Economics	Resolving diseconomies of scale	Exploiting economics of scale
Management	Planning	Operations

- The most conventional life cycles, the phases are named after the primary activity within each phase :
 - 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)

❑ **Evaluation Criteria :**

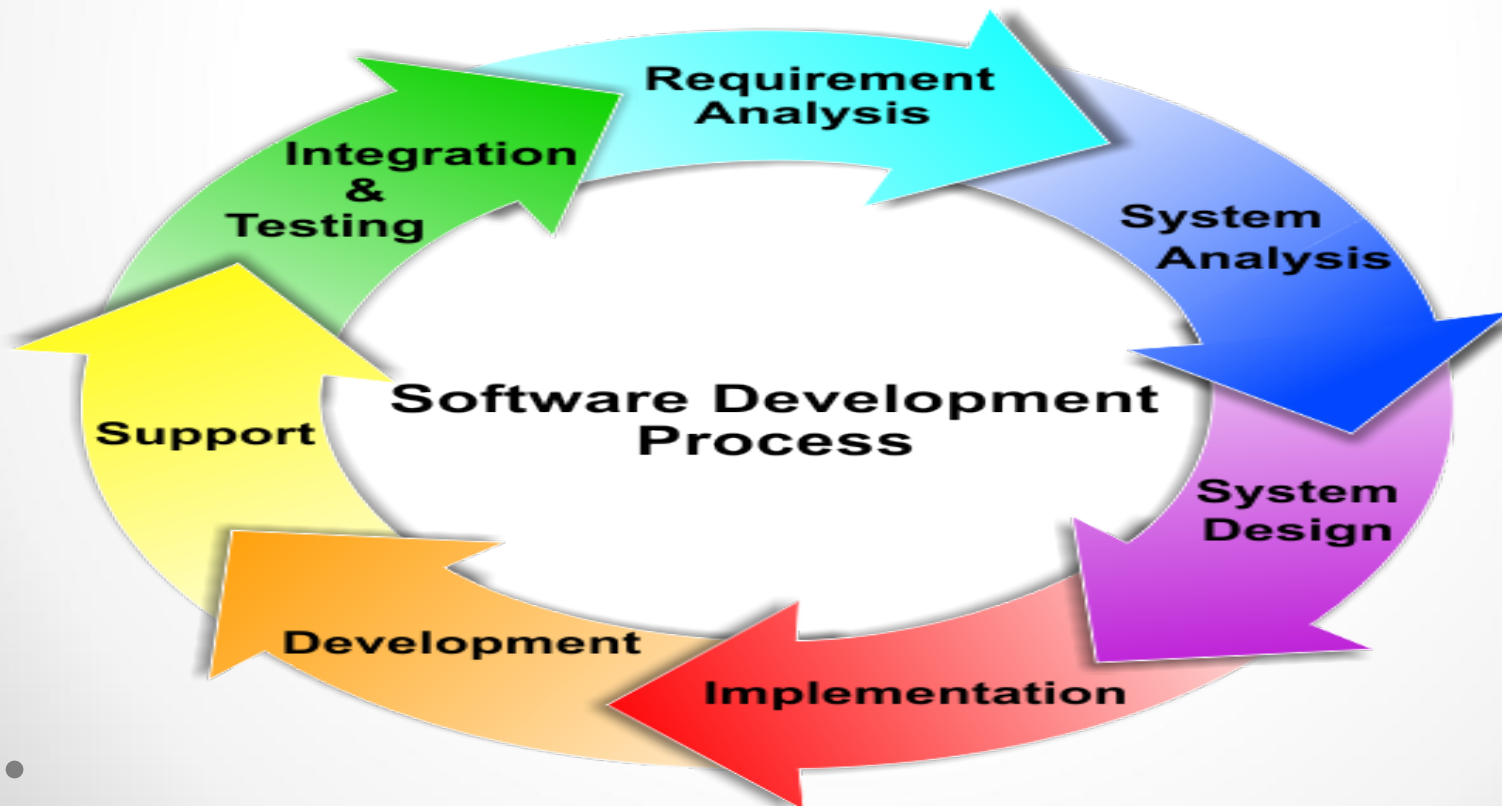
- Is the user satisfied?

- Are actual resource expenditures versus planned expenditures acceptable?

- ❑ Each of the four phases consists of one or more iterations in which some technical capability is produced in demonstrable form and assessed against a set of the criteria.

- ❑ The transition from one phase to the next maps more to a significant business decision than to the completion of specific software activity.

- So, in software engineering, a software development process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a 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.
- SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.
- Adhering to the SDLC process leads to the development of the software in a systematic and disciplined manner.

Purpose of SDLC:

- Purpose of SDLC is to deliver a high-quality product which is as per the customer's requirement.
- SDLC has defined its phases as, Requirement gathering, Designing, Coding, Testing, and Maintenance.
- It is important to adhere (to stick an object) to the phases to provide the Product in a systematic manner.

For Example:

- A software has to be developed and a team is divided to work on a feature of the product and is allowed to work as they want.
- One of the developers decides to design first whereas the other decides to code first and the other on the documentation part.
- This will lead to project failure because of which it is necessary to have a good knowledge and understanding among the team members to deliver an expected product.

Given below are the various phases of SDLC:

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

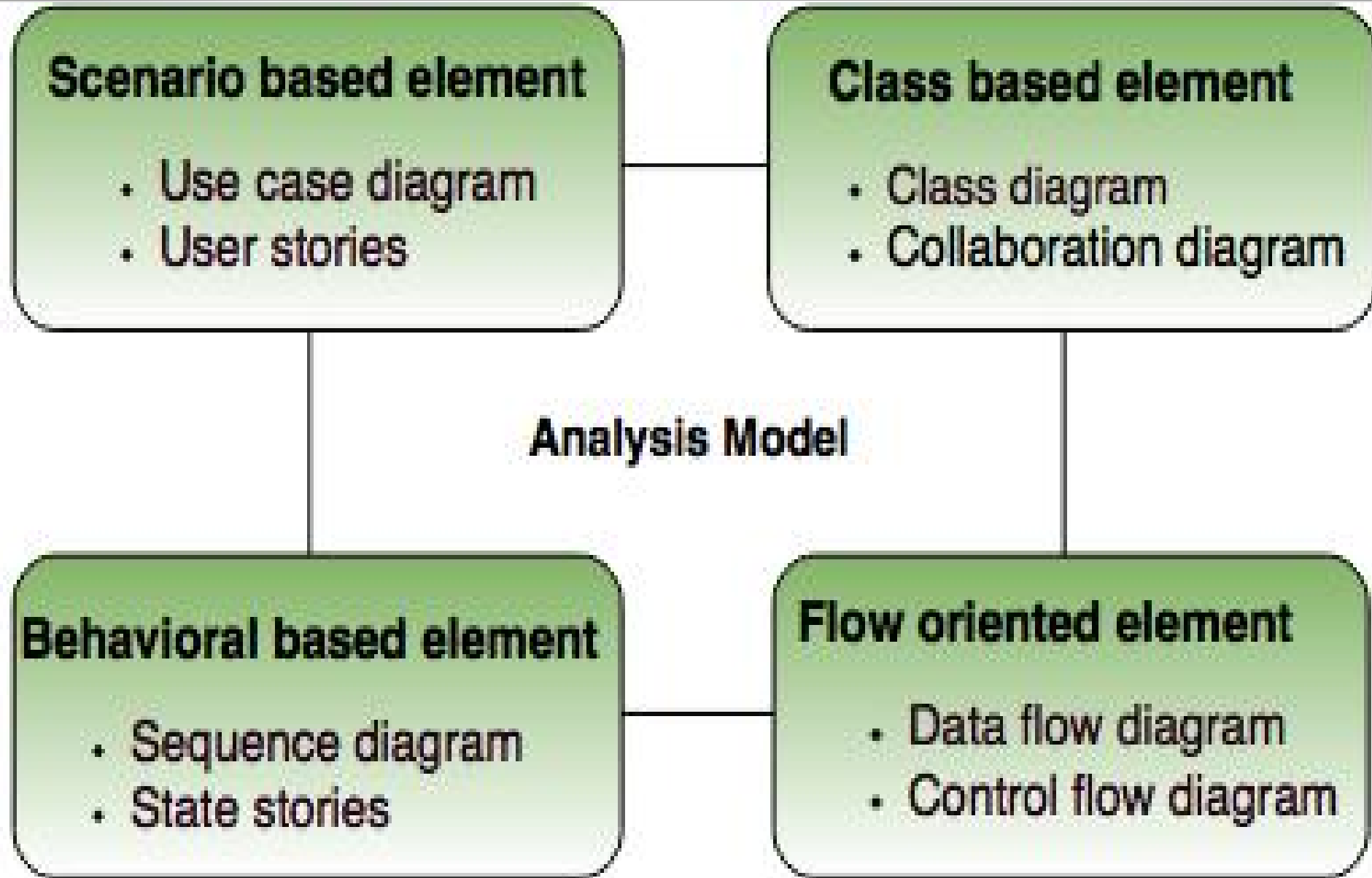


Fig. - Elements of analysis model

Elements of the Software Process

- Management elements
- Engineering elements
- Pragmatic elements

Management Elements

□ The management set includes several elements :

➤ **Work Breakdown Structure** – budgeting and collecting costs.

- The software project manager must have insight into project costs and how they are expended.
- If the WBS is structured improperly, it can drive the evolving design in the wrong direction.



Business Case – provides all the

information necessary to determine

whether the project is worth investing in.

- It details the expected revenue, expected cost, technical and management plans.

Release Specifications

Typical release specification outlines are:

I. Iteration (step-by-step process) content

II. Measurable objectives

A. Evaluation criteria

B. Follow-through approach

III. Demonstration plan

A. Schedule of activities

B. Team responsibilities

IV. Operational scenarios (use cases demonstrated)

A. Demonstration procedures

➤ **Software Development Plan** – the defining document for the project's process.

- It must comply with the contract, comply with the organization standards, evolve along with the design and requirements.

➤ **Deployment** – depending on the project, it could include several document subsets for transitioning the product into operational status.

- It could also include computer system operations manuals, software installation manuals, plans and procedures for cutover etc.

➤ **Environment** – A robust development environment must support automation of the development process.

- It should include : requirements management visual modeling document automation automated regression testing

Engineering Element of Software Process

- ❑ **Software process**
- ❑ **Reuse:** Not just of software, but also sets of requirements, parts of designs, or group of test scripts.
- ❑ **Measurement:** Trying to quantify project goals to evaluate progress (for example – number of bugs per 100 lines of code)
- ❑ **Tools and Integrated Environment:** For example, CASE (Computer-aided software engineering) tools.

➤ **Architecture Description** – it is extracted from the design model and includes views of the design, implementation, and deployment sets sufficient to understand how the operational concept of the requirements set will be achieved.

➤ **Software User Manual** – it should include installation procedures, usage procedures and guidance, operational constraints, and a user interface description. (written by test team)

Pragmatic Elements

- ❑ Over the past 30 years, the quality of documents become more important than the quality of the engineering information they represented.
- The reviewer must be knowledgeable in the engineering notation.
- Human-readable engineering artifacts should use rigorous notations that are complete, consistent, and used in a self-documenting manner.
- Paper is tangible, electronic artifacts are too easy to change.
- Short documents are more useful than long ones.

Artifacts of the Process

Artifacts of the Process

Requirements Set 1.Vision document 2.Requirements model(s)	Design Set 1.Design model(s) 2.Test model 3.Software architecture description	Implementation Set 1.Source code baselines 2.Associated compile-time files 3.Component executables	Deployment Set 1.Integrated product executable baselines 2.Associated run-time files 3.User manual
Management Set <div>Planning Artifacts 1.Work breakdown structure 2.Bussines case 3.Release specifications 4.Software development plan</div> <div>Operational Artifacts 5.Release descriptions 6.Status assessment 7.Software change order database 8.Deployment documents 9.Envioirement</div>			

Technical and Management Perspective of Software Architecture

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 organized 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.

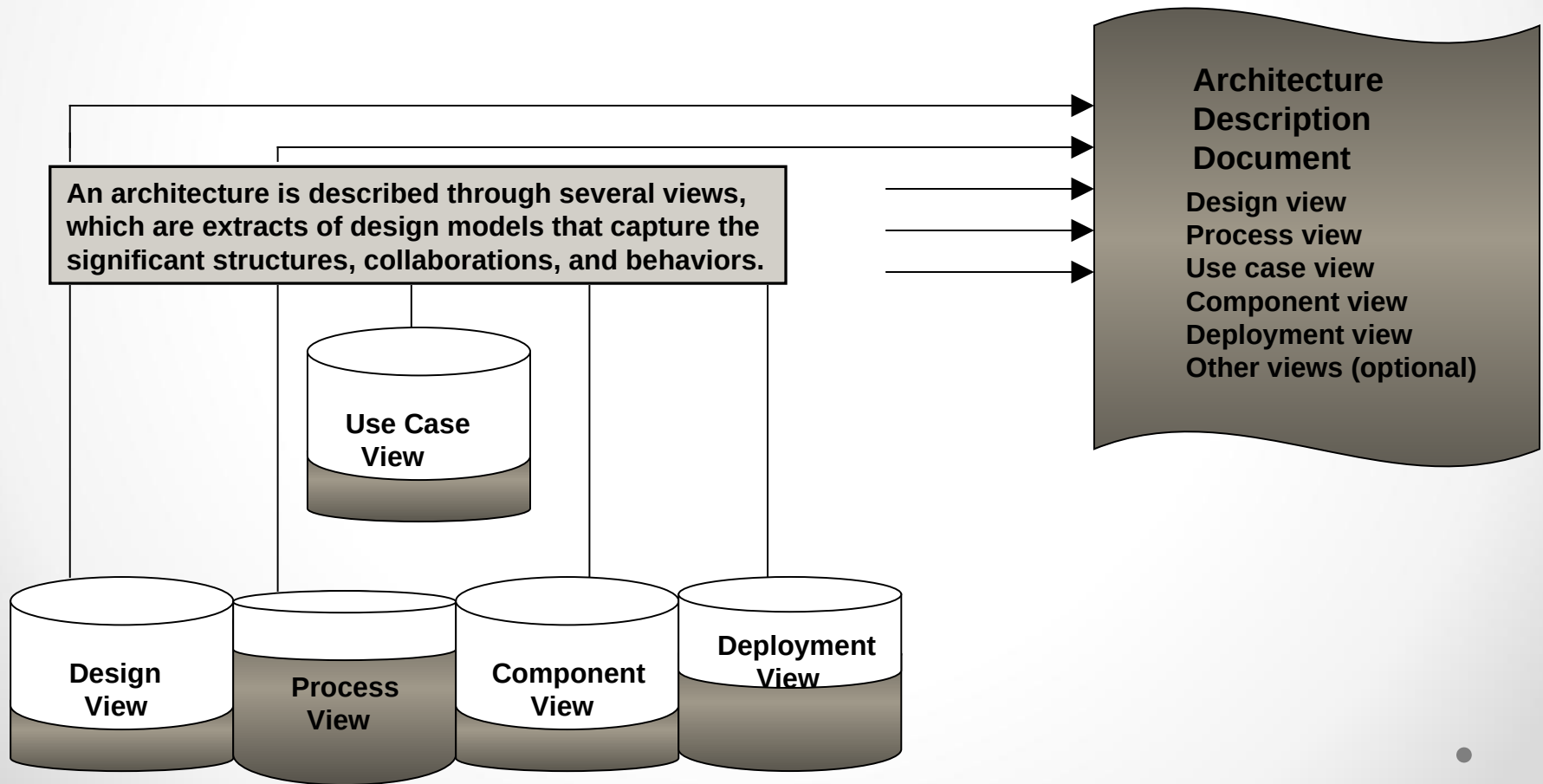
➤ Architecture development and process definition are the intellectual steps that map the problem to a solution without violating the constraints.

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 :

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 :



- 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.

Software Process Workflow and Iteration Workflow

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



Four basic key principles:

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:

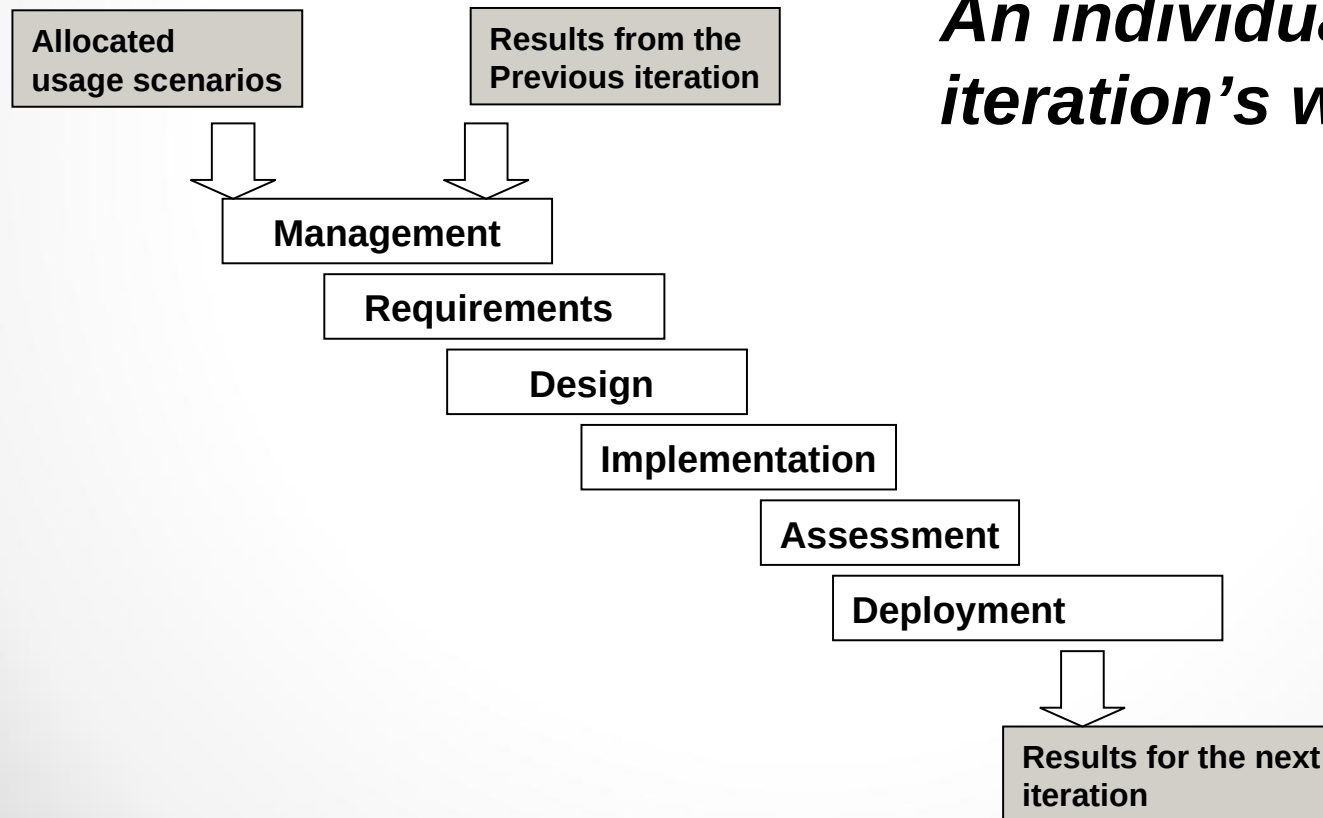
Raising the environment activities to a first-class workflow is critical; the environment is the tangible embodiment of the project's process and notations for producing the artifacts.

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.

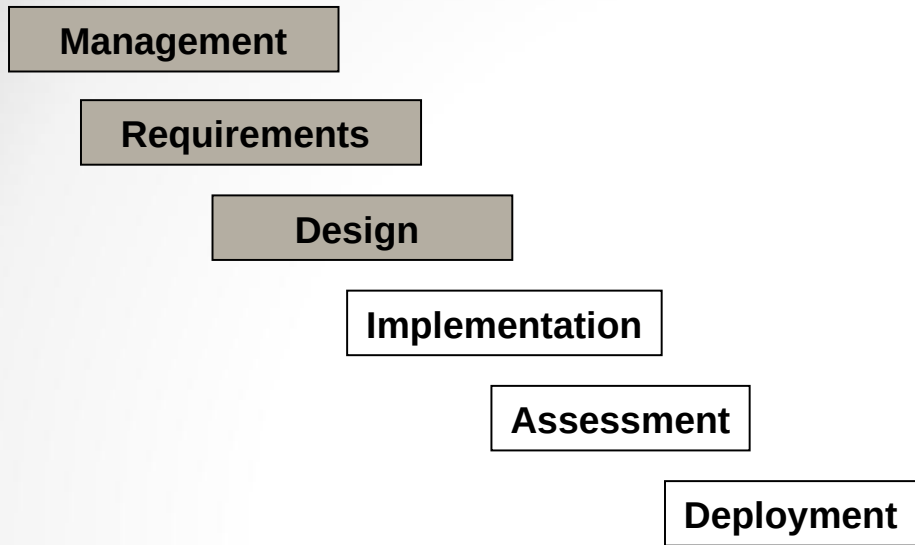
Iteration Workflow

□ An iteration consist of sequential set of activities in various proportions, depending on where the iteration is located in the development cycle.

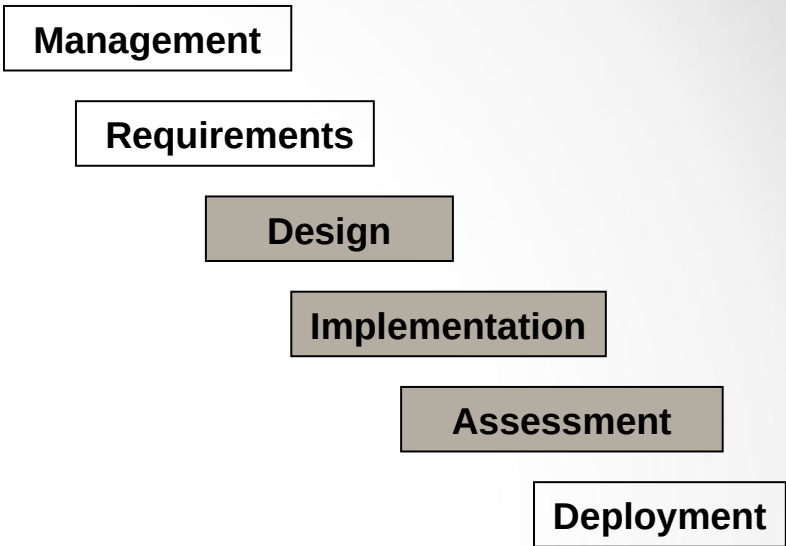


An individual iteration's workflow:

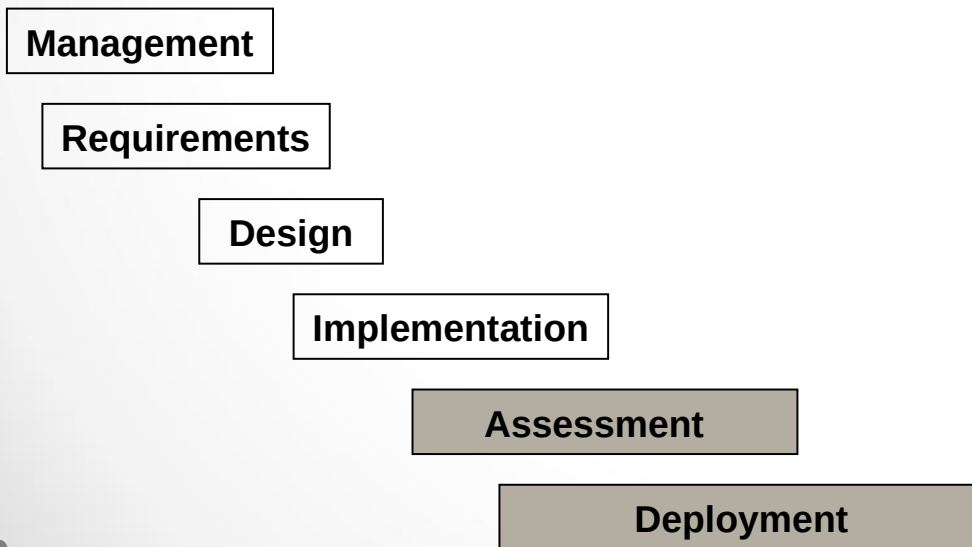
Inception and Elaboration Phases



Construction Phase



Transition Phase



Status Monitoring – 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:**

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.

Assignment Questions of Unit 02

Attempt all question and submit this assignment within Dasain.

1. Define Software Process Life Cycle Phase.
2. Explain about the various phases of Software Development Life Cycle.
3. Explain about the Software Architecture on the basis of Management Perspective and Technical Perspective .
4. Define Software Process Workflows.
5. What are the four basic key principles of software process workflows.
6. Discuss about the Software Process Checkpoints and milestone.

Thank you