UNIT-IV
PART - B

Workflows of the Process : Software Process Workflows, Iteration Workflows.

Software Process Workflows

- The term WORKFLOWS is used to mean a thread of cohesive and mostly sequential activities.
- Workflows are mapped to product artifacts. 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

- Below figure illustrates the relative levels of effort expected across the phases in each of the top-level workflows.

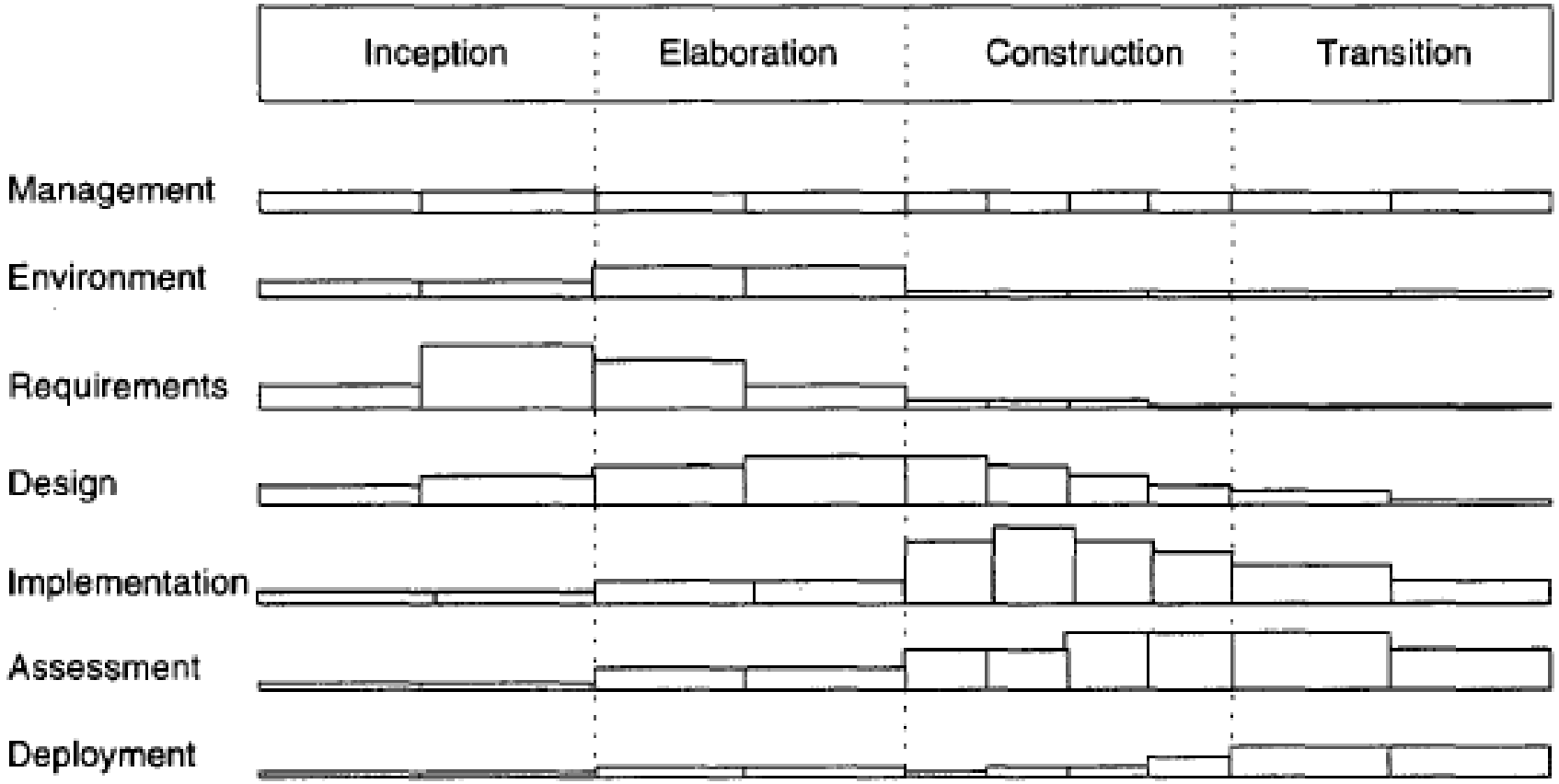


Fig: Activity levels across the life-cycle phases

Software Process Workflows

1. **Architecture-first approach** - Extensive requirements analysis, design, implementation, and assessment activities are performed before the construction phase, when full-scale Implementation is the focus. This early life-cycle focus on implementing and testing the architecture must precede full-scale development and testing of all the components and must precede the downstream focus on completeness and quality of the entire breadth of the product features.

Software Process Workflows

2. **Iterative life-cycle process** - In above figure, each phase portrays at least two iterations of each workflow. This default is intended to be descriptive, not prescriptive. Some projects may require only one iteration in a phase; others may require several iterations. The point is that the activities and artifacts of any given workflow may require more than one pass to achieve adequate results.

Software Process Workflows

3. **Round-trip engineering -** Raising the environment activities to a first-class workflow is critical. The environment is the tangible embodiment of the project's process, methods, and notations for producing the artifacts.
4. **Demonstration-based approach -** Implementation and assessment activities are initiated early in the life cycle, reflecting the emphasis on constructing executable subsets of the evolving architecture.

Software Process Workflows

- Below table shows the allocation of artifacts and the emphasis of each workflow in each of the life-cycle phases of inception, elaboration, construction, and transition.

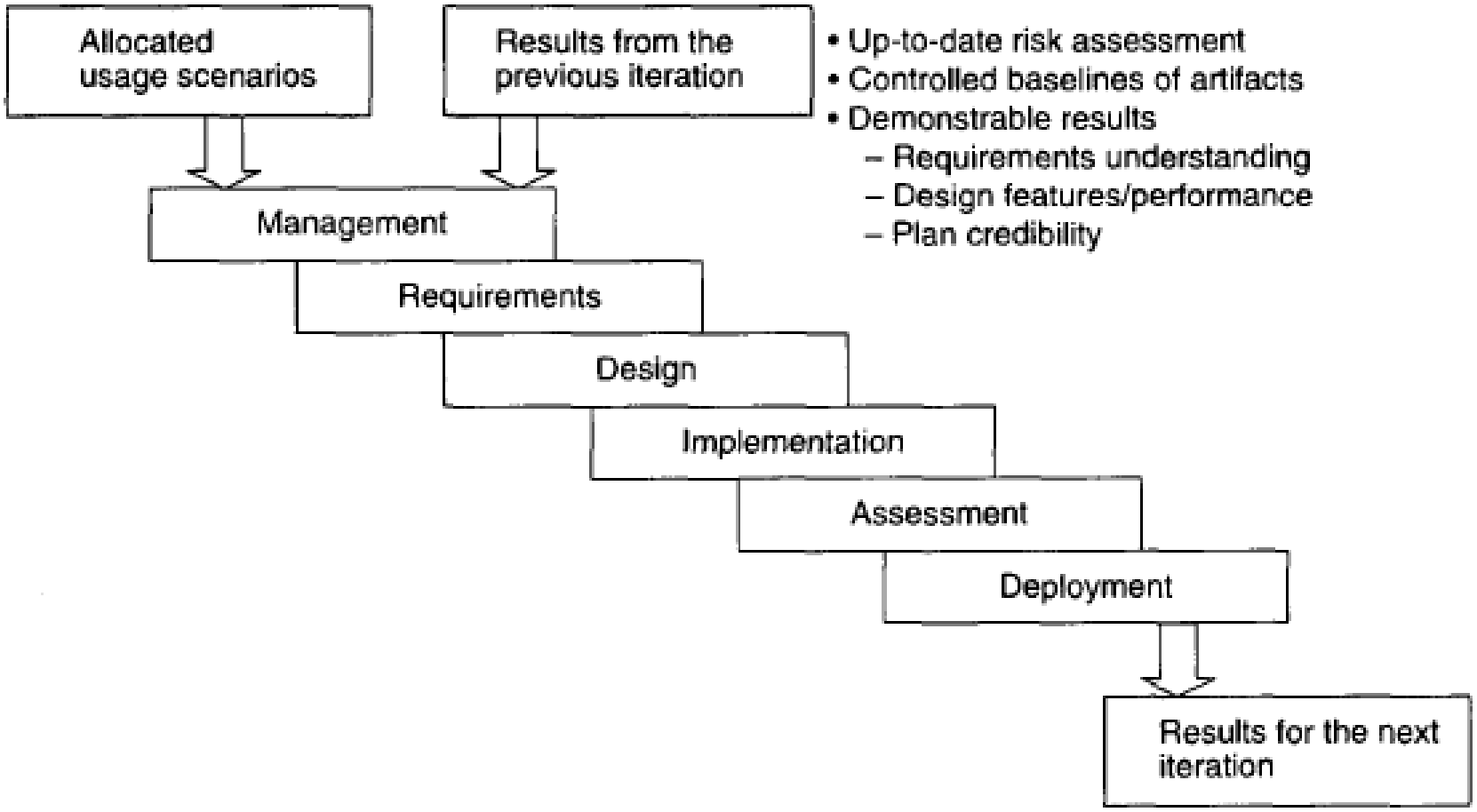
TABLE 8-1. *The artifacts and life-cycle emphases associated with each workflow*

WORKFLOW	ARTIFACTS	LIFE-CYCLE PHASE EMPHASIS
Management	Business case Software development plan Status assessments Vision Work breakdown structure	Inception: Prepare business case and vision Elaboration: Plan development Construction: Monitor and control development Transition: Monitor and control deployment
Environment	Environment Software change order database	Inception: Define development environment and change management infrastructure Elaboration: Install development environment and establish change management database Construction: Maintain development environment and software change order database Transition: Transition maintenance environment and software change order database
Requirements	Requirements set Release specifications Vision	Inception: Define operational concept Elaboration: Define architecture objectives Construction: Define iteration objectives Transition: Refine release objectives
Design	Design set Architecture description	Inception: Formulate architecture concept Elaboration: Achieve architecture baseline Construction: Design components Transition: Refine architecture and components
Implementation	Implementation set Deployment set	Inception: Support architecture prototypes Elaboration: Produce architecture baseline Construction: Produce complete componentry Transition: Maintain components
Assessment	Release specifications Release descriptions User manual Deployment set	Inception: Assess plans, vision, prototypes Elaboration: Assess architecture Construction: Assess interim releases Transition: Assess product releases
Deployment	Deployment set	Inception: Analyze user community Elaboration: Define user manual Construction: Prepare transition materials Transition: Transition product to user

Iteration Workflows

- An iteration consists of a loosely sequential set of activities in various proportions, depending on where the iteration is located in the development cycle.
- Each iteration is defined in terms of a set of allocated usage scenarios.
- The components needed to implement all selected scenarios are developed and integrated with the results of previous iterations.
- An individual iteration's workflow, illustrated in below figure, generally includes the following sequence:

Iteration Workflows



Iteration Workflows

- **Management:** iteration planning to determine the content of the release and develop the detailed plan for the iteration; assignment of work packages, or tasks, to the development team
- **Environment:** evolving the software change order database to reflect all new baselines and changes to existing baselines for all product, test, and environment components

Iteration Workflows

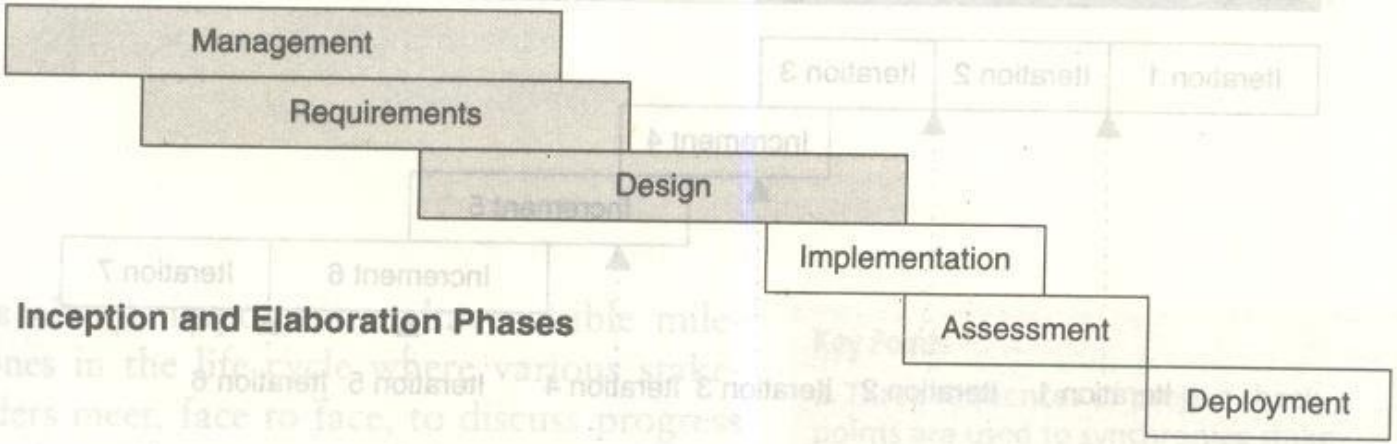
- **Requirements:** analyzing the baseline plan, the baseline architecture, and the baseline requirements set artifacts to fully elaborate the use cases to be demonstrated at the end of this iteration and their evaluation criteria; updating any requirements set artifacts to reflect changes necessitated by results of this iteration's engineering activities
- **Design:** evolving the baseline architecture and the baseline design set artifacts to elaborate fully the design model and test model components necessary to demonstrate against the evaluation criteria allocated to this iteration; updating design set artifacts to reflect changes necessitated by the results of this iteration's engineering activities

Iteration Workflows

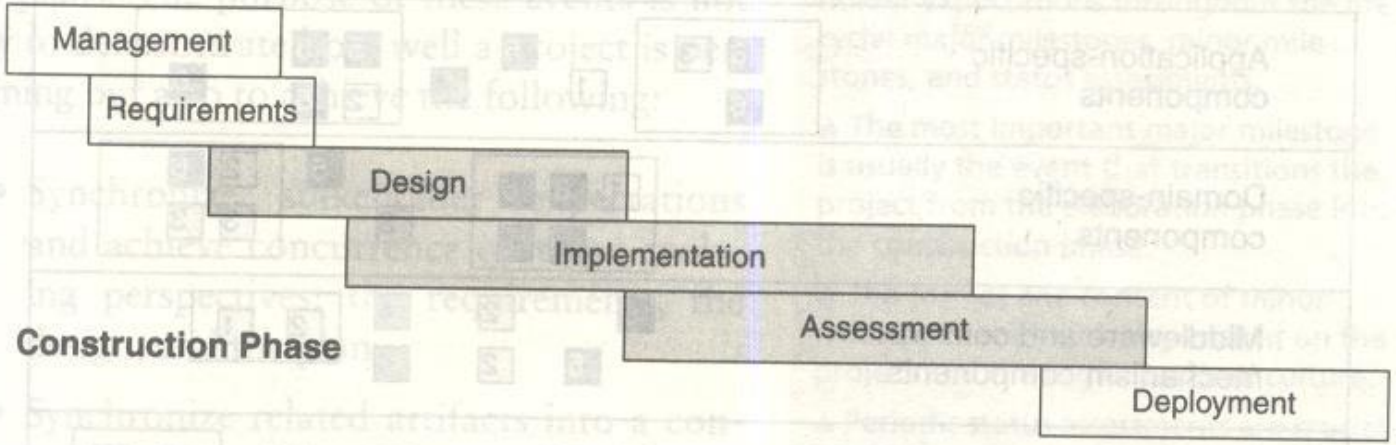
- **Implementation:** developing or acquiring any new components, and enhancing or modifying any existing components, to demonstrate the evaluation criteria allocated to this iteration; integrating and testing all new and modified components with existing baselines
- **Assessment:** evaluating the results of the iteration, including compliance with the allocated evaluation criteria and the quality of the current baselines; identifying any rework required and determining whether it should be performed before deployment of this release or allocated to the next release; assessing results to improve the basis of the subsequent iteration's plan

Iteration Workflows

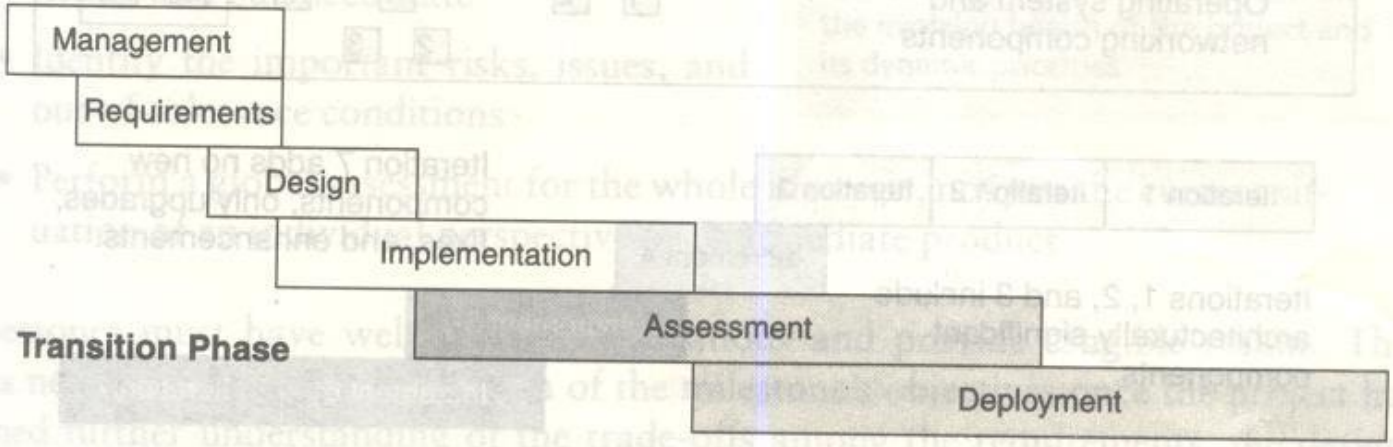
- **Deployment:** transitioning the release either to an external organization (such as a user, independent verification and validation contractor, or regulatory agency) or to internal closure by conducting a post-mortem so that lessons learned can be captured and reflected in the next iteration
- Iterations in the inception and elaboration phases focus on management, requirements, and design activities. Iterations in the construction phase focus on design, implementation, and assessment. Iterations in the transition phase focus on assessment and deployment.
- Below figure shows the emphasis on different activities across the life cycle.



Inception and Elaboration Phases



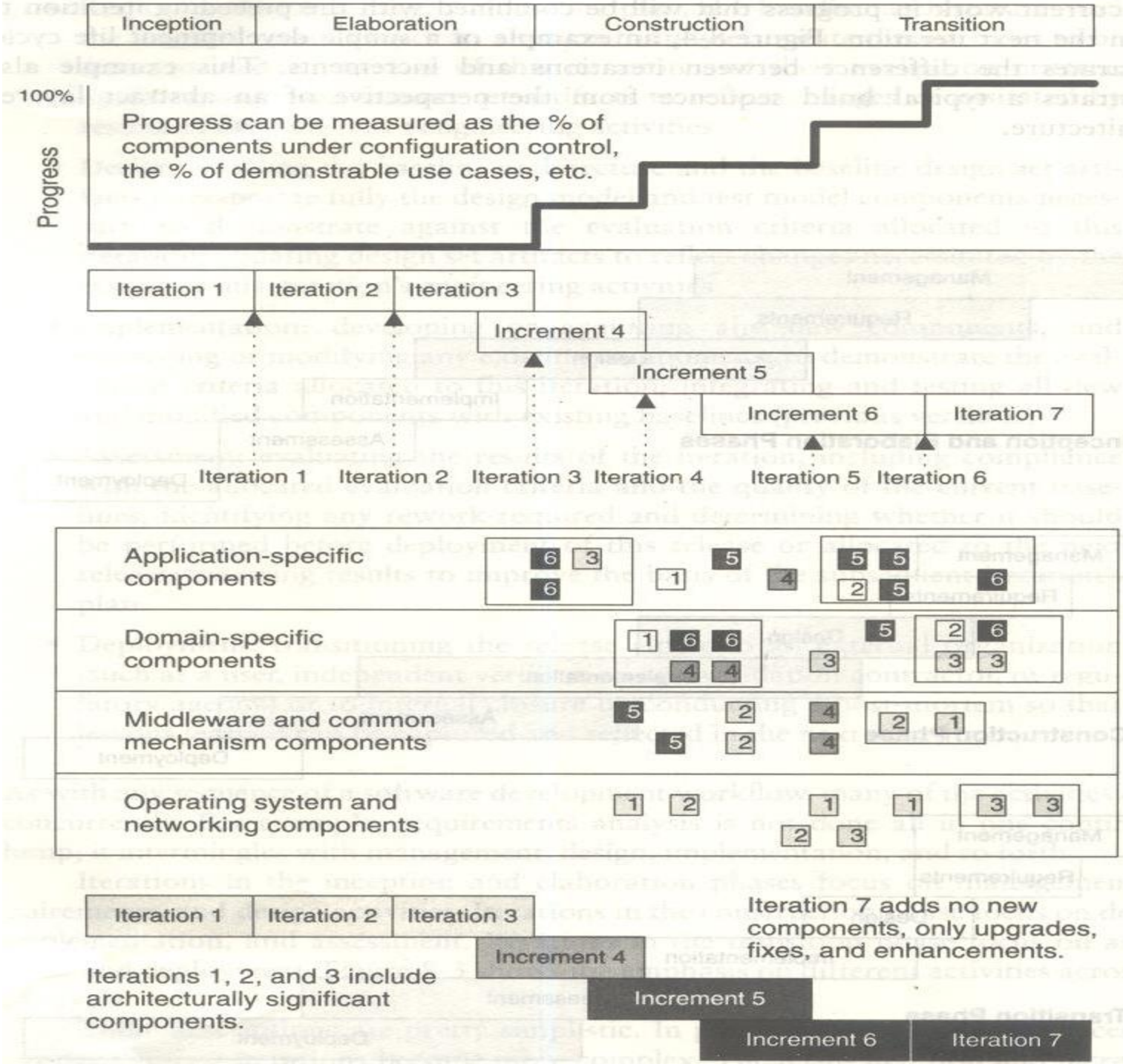
Construction Phase



Transition Phase

Iteration Workflows

- These descriptions are pretty simplistic. In practice, the various sequences and overlaps among iterations become more complex.
- The terms iteration and increment deal with some of the pragmatic considerations.
- An iteration represents the state of the overall architecture and the complete deliverable system.
- An increment represents the current work in progress that will be combined with the preceding iteration to form the next iteration.
- Below figure, an example of a simple development life cycle, illustrates the differences between iterations and increments. This example also illustrates a typical build sequence from the perspective of an abstract layered architecture.


Inception
Elaboration
Construction
Transition
100%
Progress
Progress can be measured as the % of components under configuration control, the % of demonstrable use cases, etc.
Iteration 1
Iteration 2
Iteration 3
Increment 4
Increment 5
Increment 6
Iteration 7
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Application-specific components
Domain-specific components
Middleware and common mechanism components
Operating system and networking components
Iteration 1
Iteration 2
Iteration 3
Increment 4
Increment 5
Increment 6
Iteration 7
Iterations 1, 2, and 3 include architecturally significant components.
Iteration 7 adds no new components, only upgrades, fixes, and enhancements.