UNIT-IV
PART - C

Checkpoints of the Process: Major Milestones, Minor Milestones, Periodic Status Assessments.

Checkpoints of the process

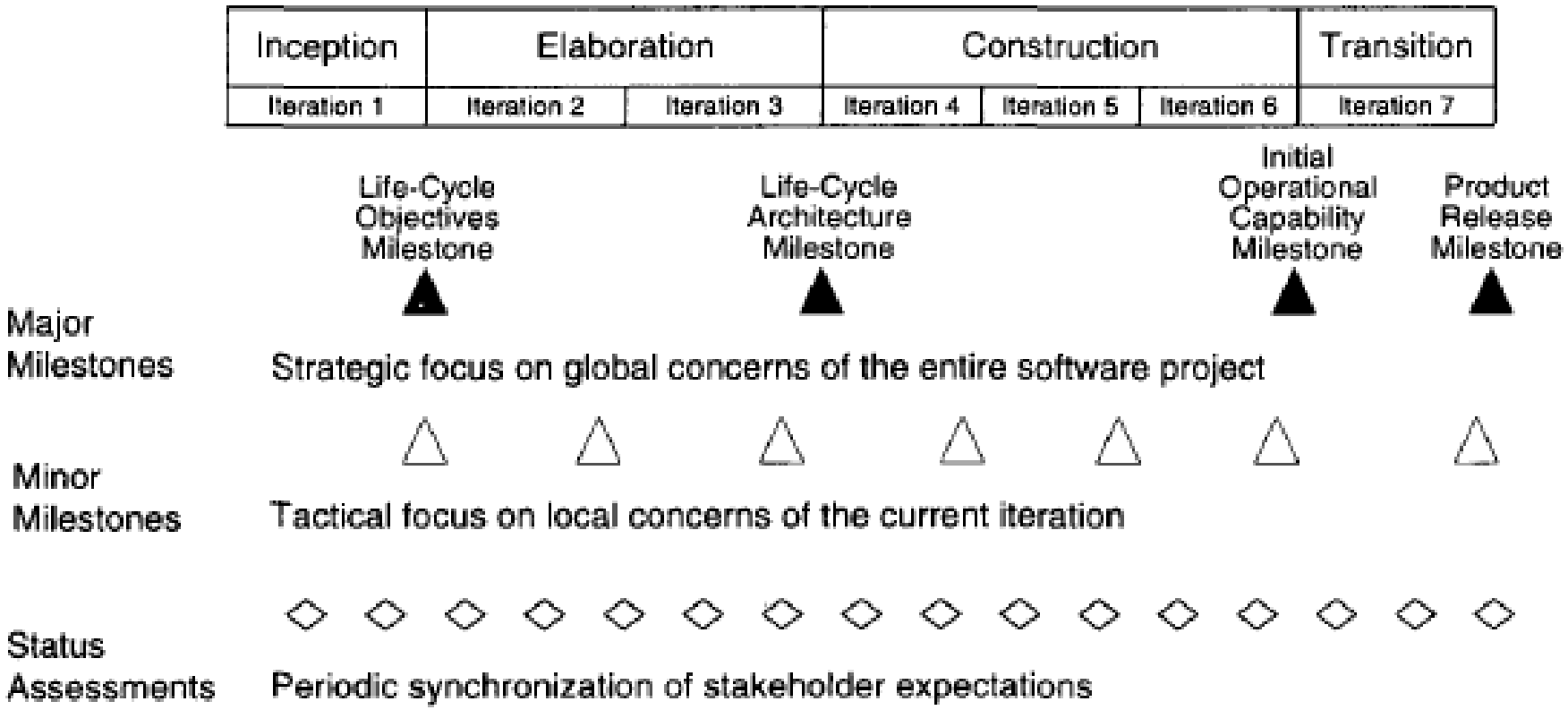
- Three types of joint management reviews are conducted throughout the process:

1. **Major milestones** - These system wide events are held at the end of each development phase. They 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** - These iteration-focused events are conducted to review the content of an iteration in detail and to authorize continued work.
3. **Status assessments** - These periodic events provide management with frequent and regular insight into the progress being made.

Checkpoints of the process

- Each of the four phases-inception, elaboration, construction, and transition consists of one or more iterations and concludes with a major milestone when a planned technical capability is produced in demonstrable form.
- An iteration represents a cycle of activities for which there is a well-defined intermediate result-a minor milestone-captured with two artifacts: a release specification (the evaluation criteria and plan) and a release description (the results).
- Major milestones at the end of each phase use formal, stakeholder-approved evaluation criteria and release descriptions; minor milestones use informal, development-team-controlled versions of these artifacts.

Checkpoints of the process



Major Milestones

- The four major milestones occur at the transition points between life-cycle phases.
- They can be used in many different process models, including the conventional waterfall model.
- In an iterative model, the major milestones are used to achieve concurrence among all stakeholders on the current state of the project.
- Different stakeholders have very different concerns:

Major Milestones

- **Customers:** schedule and budget estimates, feasibility, risk assessment, requirements understanding, progress, product line compatibility
- **Users:** consistency with requirements and usage scenarios, potential for accommodating growth, quality attributes
- **Architects and systems engineers:** product line compatibility, requirements changes, trade-off analyses, completeness and consistency, balance among risk, quality, and usability

Major Milestones

- **Developers:** sufficiency of requirements detail and usage scenario descriptions, . frameworks for component selection or development, resolution of development risk, product line compatibility, sufficiency of the development environment
- **Maintainers:** sufficiency of product and documentation artifacts, understandability, interoperability with existing systems, sufficiency of maintenance environment
- **Others:** possibly many other perspectives by stakeholders such as regulatory agencies, independent verification and validation contractors, venture capital investors, subcontractors, associate contractors, and sales and marketing teams

Major Milestones

- Below table summarizes the balance of information across the major milestones.

TABLE 9-1. *The general status of plans, requirements, and products across the major milestones*

MILESTONES	PLANS	UNDERSTANDING OF PROBLEM SPACE (REQUIREMENTS)	SOLUTION SPACE PROGRESS (SOFTWARE PRODUCT)
Life-cycle objectives milestone	Definition of stakeholder responsibilities Low-fidelity life-cycle plan High-fidelity elaboration phase plan	Baseline vision, including growth vectors, quality attributes, and priorities Use case model	Demonstration of at least one feasible architecture Make/buy/reuse trade-offs Initial design model
Life-cycle architecture milestone	High-fidelity construction phase plan (bill of materials, labor allocation) Low-fidelity transition phase plan	Stable vision and use case model Evaluation criteria for construction releases, initial operational capability Draft user manual	Stable design set Make/buy/reuse decisions Critical component prototypes
Initial operational capability milestone	High-fidelity transition phase plan	Acceptance criteria for product release Releasable user manual	Stable implementation set Critical features and core capabilities Objective insight into product qualities
Product release milestone	Next-generation product plan	Final user manual	Stable deployment set Full features Compliant quality

1. Life-Cycle Objectives Milestone

- The life-cycle objectives milestone occurs at the end of the inception phase.
- The goal is to present to all stakeholders a recommendation on how to proceed with development, including a plan, estimated cost and schedule, and expected benefits and cost savings.
- A successfully completed life-cycle objectives milestone will result in authorization from all stakeholders to proceed with the elaboration phase.

2. Life-Cycle Architecture Milestone

- The life-cycle architecture milestone occurs at the end of the elaboration phase.
- The primary goal is to demonstrate an executable architecture to all stakeholders.
- A more detailed plan for the construction phase is presented for approval.
- Critical issues relative to requirements and the operational concept are addressed.

2. Life-Cycle Architecture Milestone

- The baseline architecture consists of both a human-readable representation (the architecture document) and a configuration-controlled set of software components captured in the engineering artifacts.
- A successfully completed life-cycle architecture milestone will result in authorization from the stakeholders to proceed with the construction phase.

2. Life-Cycle Architecture Milestone

- The content of this milestone will vary across project domains. It should include at least the following:
  - A presentation and overview of the current project state
  - A configuration-controlled set of engineering information, available electronically or in hard copy
  - An executable demonstration of capability
- The technical data listed in figure (a) should have been reviewed by the time of the life-cycle architecture milestone. Figure (b) provides default agendas for this milestone.

Engineering artifacts available at the life-cycle architecture milestone

I. **Requirements**
   A. Use case model
   B. Vision document (text, use cases)
   C. Evaluation criteria for elaboration (text, scenarios)

II. **Architecture**
   A. Design view (object models)
   B. Process view (if necessary, run-time layout, executable code structure)
   C. Component view (subsystem layout, make/buy/reuse component identification)
   D. Deployment view (target run-time layout, target executable code structure)
   E. Use case view (test case structure, test result expectation)
      1. Draft user manual

III. **Source and executable libraries**
   A. Product components
   B. Test components
   C. Environment and tool components

Default agendas for the life-cycle architecture milestone

Presentation Agenda

I. **Scope and objectives**
   A. Demonstration overview
II. **Requirements assessment**
   A. Project vision and use cases
   B. Primary scenarios and evaluation criteria
III. **Architecture assessment**
   A. Progress
      1. Baseline architecture metrics (progress to date and baseline for measuring future architectural stability, scrap, and rework)
      2. Development metrics baseline estimate (for assessing future progress)
      3. Test metrics baseline estimate (for assessing future progress of the test team)
   B. Quality
      1. Architectural features (demonstration capability summary vs. evaluation criteria)
      2. Performance (demonstration capability summary vs. evaluation criteria)
      3. *Exposed architectural risks and resolution plans*
      4. Affordability and make/buy/reuse trade-offs
IV. **Construction phase plan assessment**
   A. Iteration content and use case allocation
   B. Next iteration(s) detailed plan and evaluation criteria
   C. Elaboration phase cost/schedule performance
   D. Construction phase resource plan and basis of estimate
   E. Risk assessment

Demonstration Agenda

I. **Evaluation criteria**
II. **Architecture subset summary**
III. **Demonstration environment summary**
IV. **Scripted demonstration scenarios**
V. **Evaluation criteria results and follow-up items**

3. Initial Operational Capability Milestone

- The initial operational capability milestone occurs late in the construction phase.
- The goals are to assess the readiness of the software to begin the transition into customer/user sites and to authorize the start of acceptance testing.
- Acceptance testing can be done incrementally across multiple iterations or can be completed entirely during the transition phase.
- The initiation of the transition phase is not necessarily the completion of the construction phase.
- These phases typically overlap until an initial product is delivered to the user for self-sufficient operation.

4. Product Release Milestone

- The product release milestone occurs at the end of the transition phase.
- The goal is to assess the completion of the software and its transition to the support organization, if any.
- The results of acceptance testing are reviewed, and all open issues are addressed.
- Software quality metrics are reviewed to determine whether quality is sufficient for transition to the support organization.

Minor Milestones

- The number of iteration-specific, informal milestones needed depends on the content and length of the iteration.
- For most iterations, which have a one-month to six-month duration, only two minor milestones are needed: the iteration readiness review and the iteration assessment review.
- For longer iterations, more intermediate review points may be necessary.

Minor Milestones

- The milestones of an iteration and its associated evaluation criteria need to focus the engineering activities on the project priorities as defined in the overall software development plan, business case, and vision.

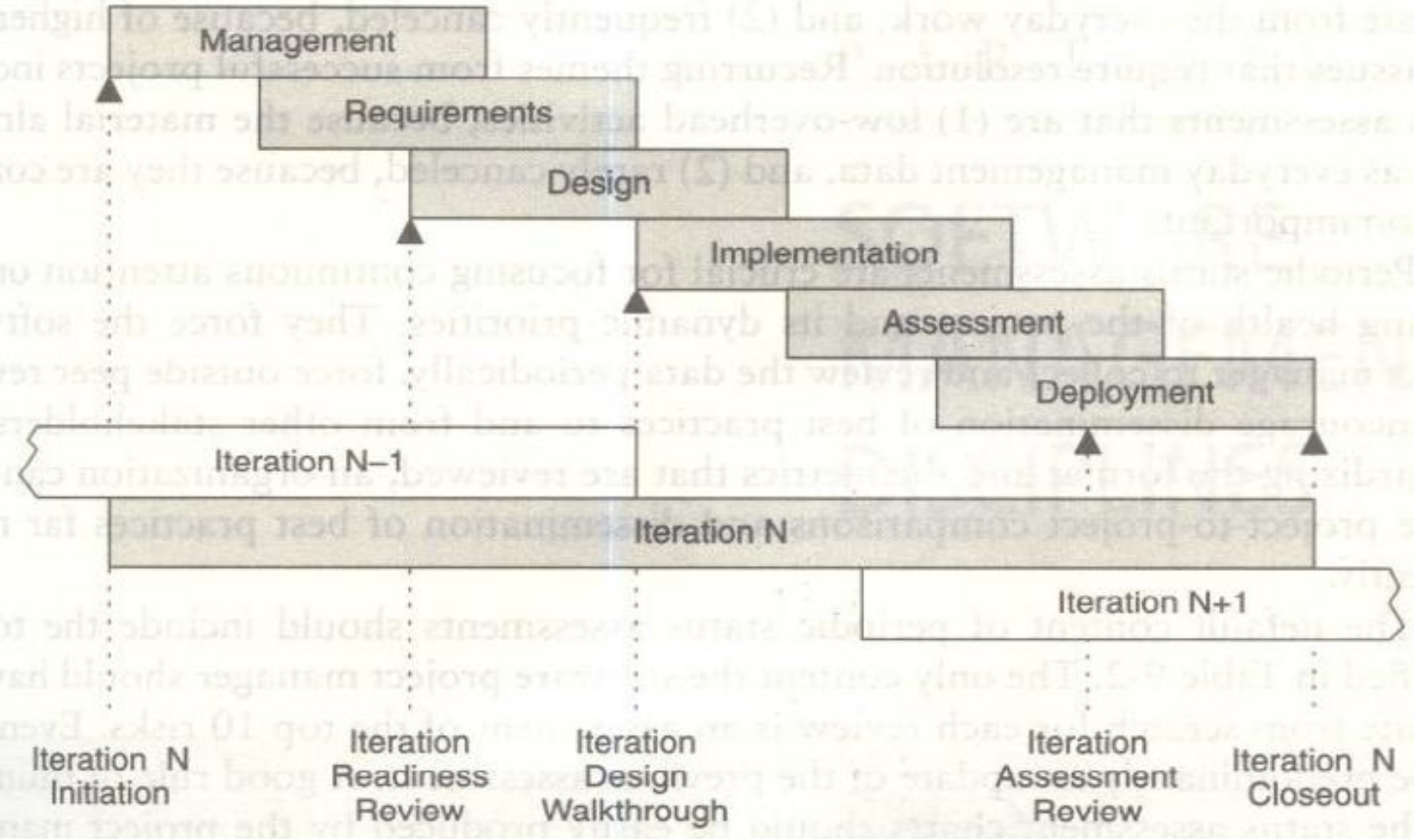
Minor Milestones

- **Iteration Readiness Review** - This informal milestone is conducted at the start of each iteration to review the detailed iteration plan and the evaluation criteria that have been allocated to this iteration.
- **Iteration Assessment Review** - This informal milestone is conducted at the end of each iteration to assess the degree to which the iteration achieved its objectives and satisfied its evaluation criteria, to review iteration results, to review qualification test results (if part of the iteration), to determine the amount of rework to be done, and to review the impact of the iteration results on the plan for subsequent iterations.

Minor Milestones

- The format and content of these minor milestones tend to be highly dependent on the project and the organizational culture.
- Below figure identifies the various minor milestones to be considered when a project is being planned.

Minor Milestones



Periodic Status Assessments

- Periodic status assessments are management reviews conducted at regular intervals (monthly, quarterly) to address progress and quality indicators, ensure continuous attention to project dynamics, and maintain open communications among all stakeholders.

Periodic Status Assessments

- Periodic status assessments serve as project snapshots.
- While the period may vary, the recurring event forces the project history to be captured and documented.
- Status assessments provide the following:
  - A mechanism for openly addressing, communicating, and resolving management issues, technical issues, and project risks
  - Objective data derived directly from on-going activities and evolving product configurations
  - A mechanism for disseminating process, progress, quality trends, practices, and experience information to and from all stakeholders in an open forum

Periodic Status Assessments

- Periodic status assessments are crucial for focusing continuous attention on the evolving health of the project and its dynamic priorities.
- They force the software project manager to collect and review the data periodically, force outside peer review, and encourage dissemination of best practices to and from other stakeholders.
- The default content of periodic status assessments should include the topics identified in below table.

Default content of status assessment reviews

TOPIC	CONTENT
Personnel	Staffing plan vs. actuals
	Attritions, additions
Financial trends	Expenditure plan vs. actuals for the previous, current, and next major milestones
	Revenue forecasts
Top 10 risks	Issues and criticality resolution plans
	Quantification (cost, time, quality) of exposure
Technical progress	Configuration baseline schedules for major milestones
	Software management metrics and indicators
	Current change trends
	Test and quality assessments
Major milestone plans and results	Plan, schedule, and risks for the next major milestone
	Pass/fail results for all acceptance criteria
Total product scope	Total size, growth, and acceptance criteria perturbations