CS342 Software Engineering

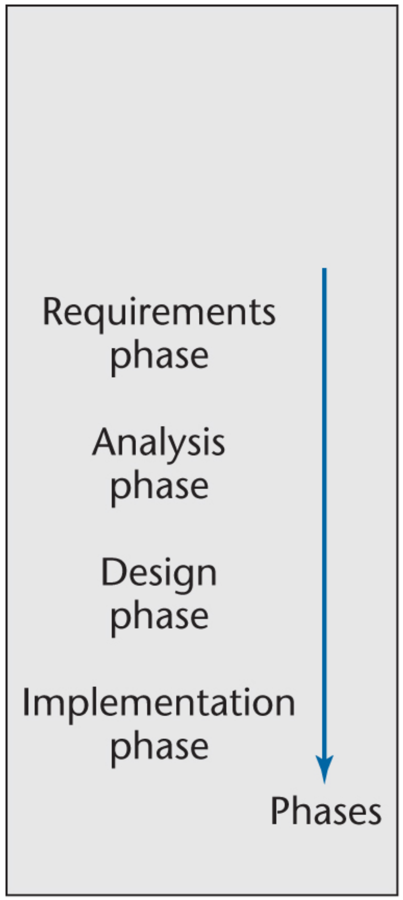
Dr. Ismail Hababeh
German Jordanian University

Lecture 5

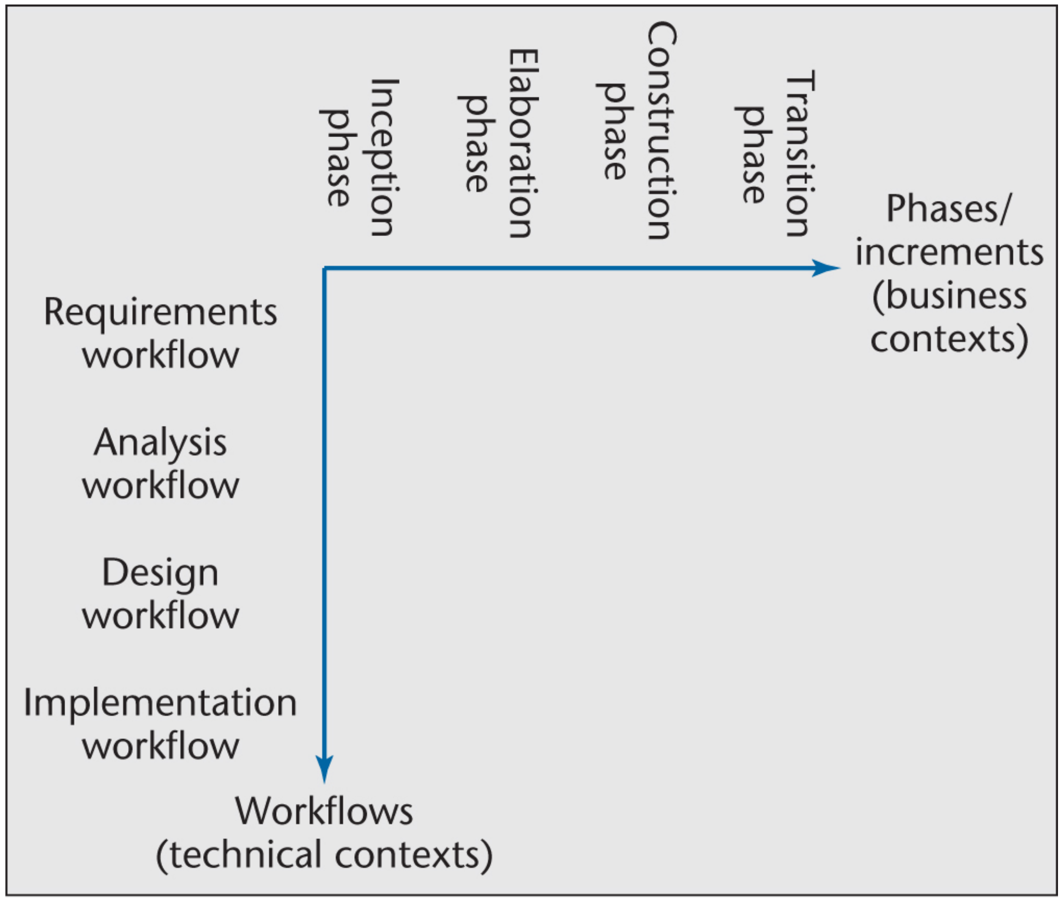
SOFTWARE LIFE-CYCLE
MODELS

Adapted from Software Engineering, by Dr. Paul E. Young
& slides by Dr. Mohammad Daoud

One- and Two-Dimensional Life-Cycle Models



(a)

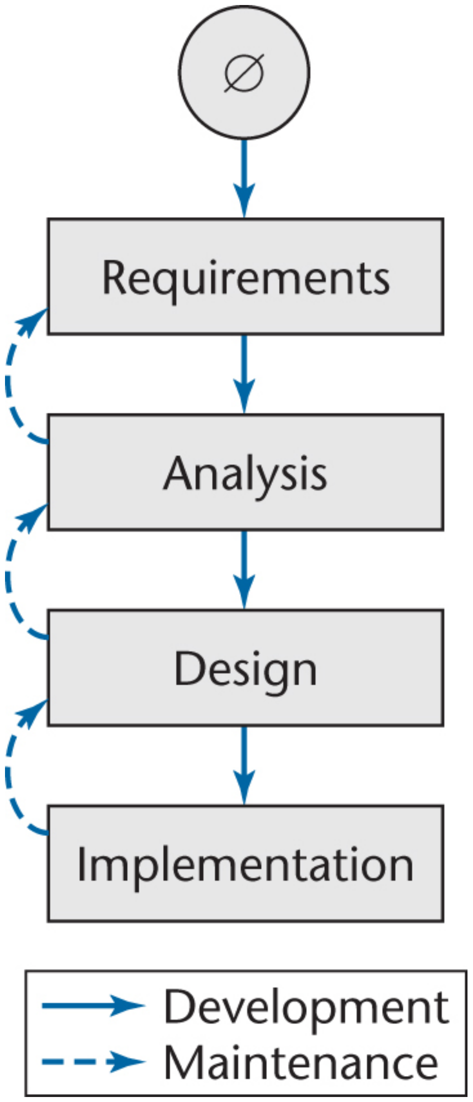


(b)

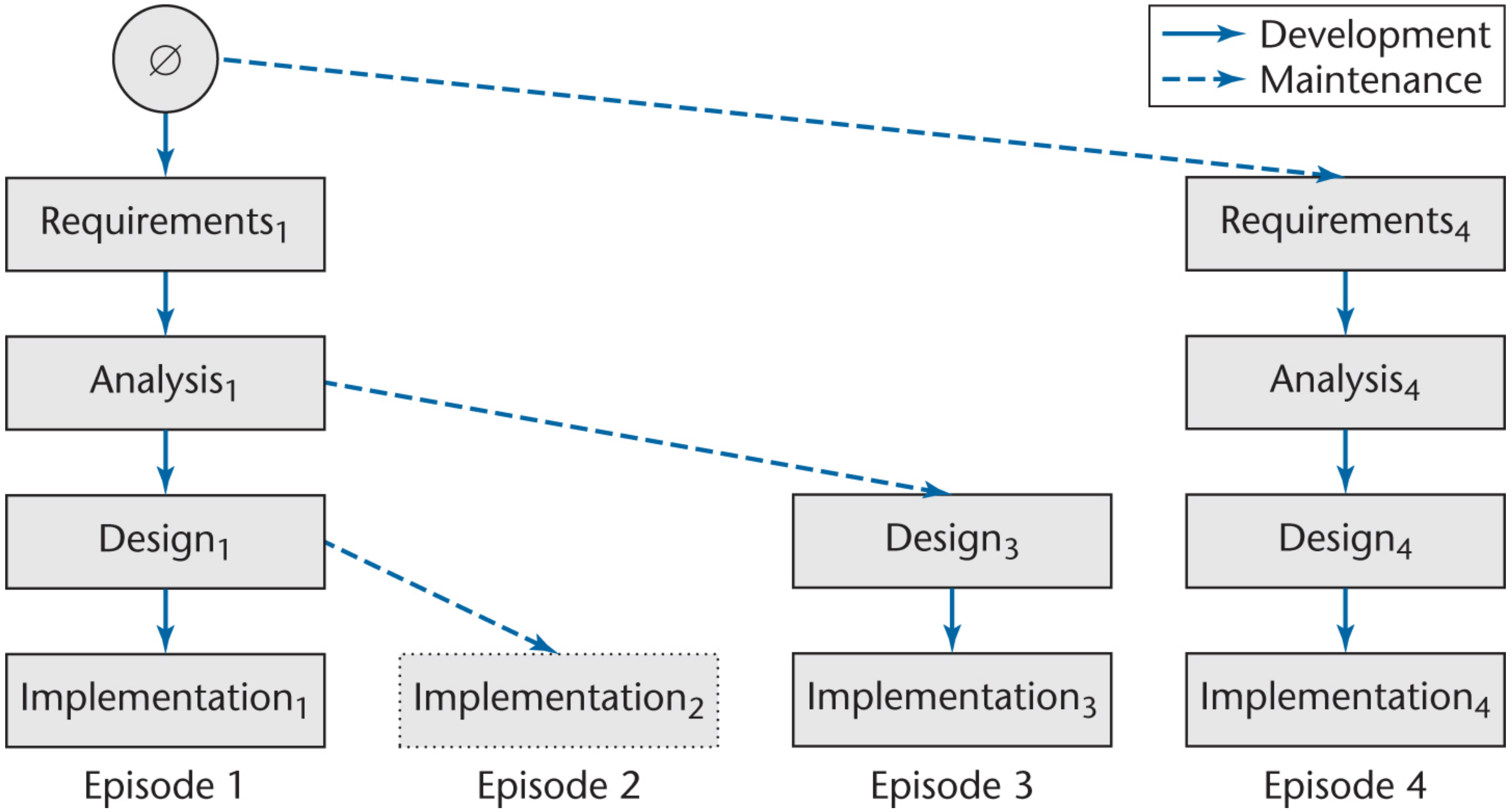
One-Dimensional Model

A traditional life cycle is a one-dimensional model represented by the single axis.

Example: Waterfall model



Evolution-Tree Model

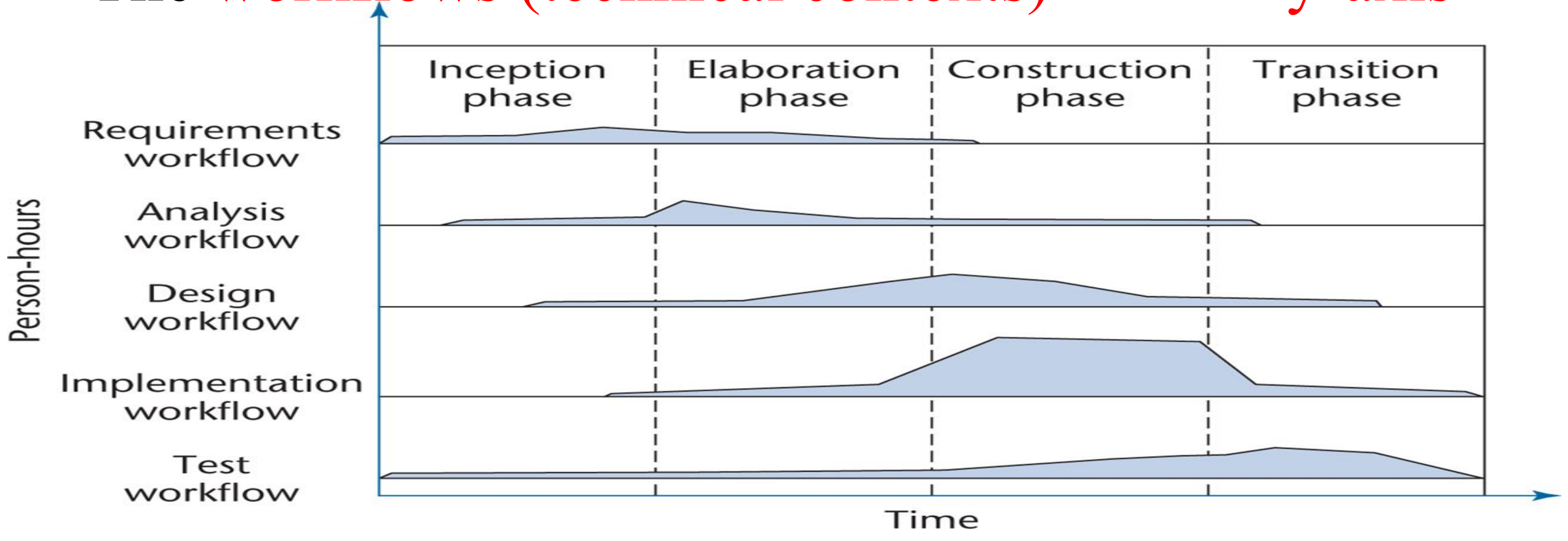


The Evolution-Tree Model

- The explicit order of events is described as episodes (periods).
- At the end of each episode
  - We have a *baseline*, a complete set of *artifacts* (essential components of the software)
- Example:
  - Baseline at the end of Episode 3:
    - Requirements$_1$, Analysis$_1$, Design$_3$, Implementation$_3$

Two-Dimensional Life-Cycle Model

- A dynamic life cycle is a two-dimensional model represented by two axis, phases and workflows Example: Unified Processes model
- The Unified Processes is a two-dimensional model
  - The phases (business contexts) $\longrightarrow$ x-axis
  - The workflows (technical contexts) $\longrightarrow$ y-axis



Why Two-Dimensional Model?

- Not all additional complications of the one -dimensional model are necessary.
- In an ideal world, each workflow would be completed before the next workflow is started
- In real world, the development task is too big to fit one phase.
- Therefore, Miller's Law is applied
  - The development task is divided into increments (phases)
  - Within each increment, iteration is performed until the task is completed.

Two-Dimensional Model

- At the beginning of the process, there is no enough information about the software product to carry out the requirements and other core workflows.
- A software product then is divided into subsystems.
- If subsystem is too large, then it is divided into components.

Two-Dimensional Model

- The Unified Processes is the best solution for managing a large problem as a set of smaller, independent sub-problems.
  - It provides a framework for increment and iteration.

Sequential Phases vs. Workflows

- Sequential phases do not exist in the real world
- Instead, the five core overlapped ***workflows*** (activities) are performed over the entire product life cycle periods:
  - Requirements workflow
  - Analysis workflow
  - Design workflow
  - Implementation workflow
  - Testing workflow
- At each life cycle period, one workflow dominates.

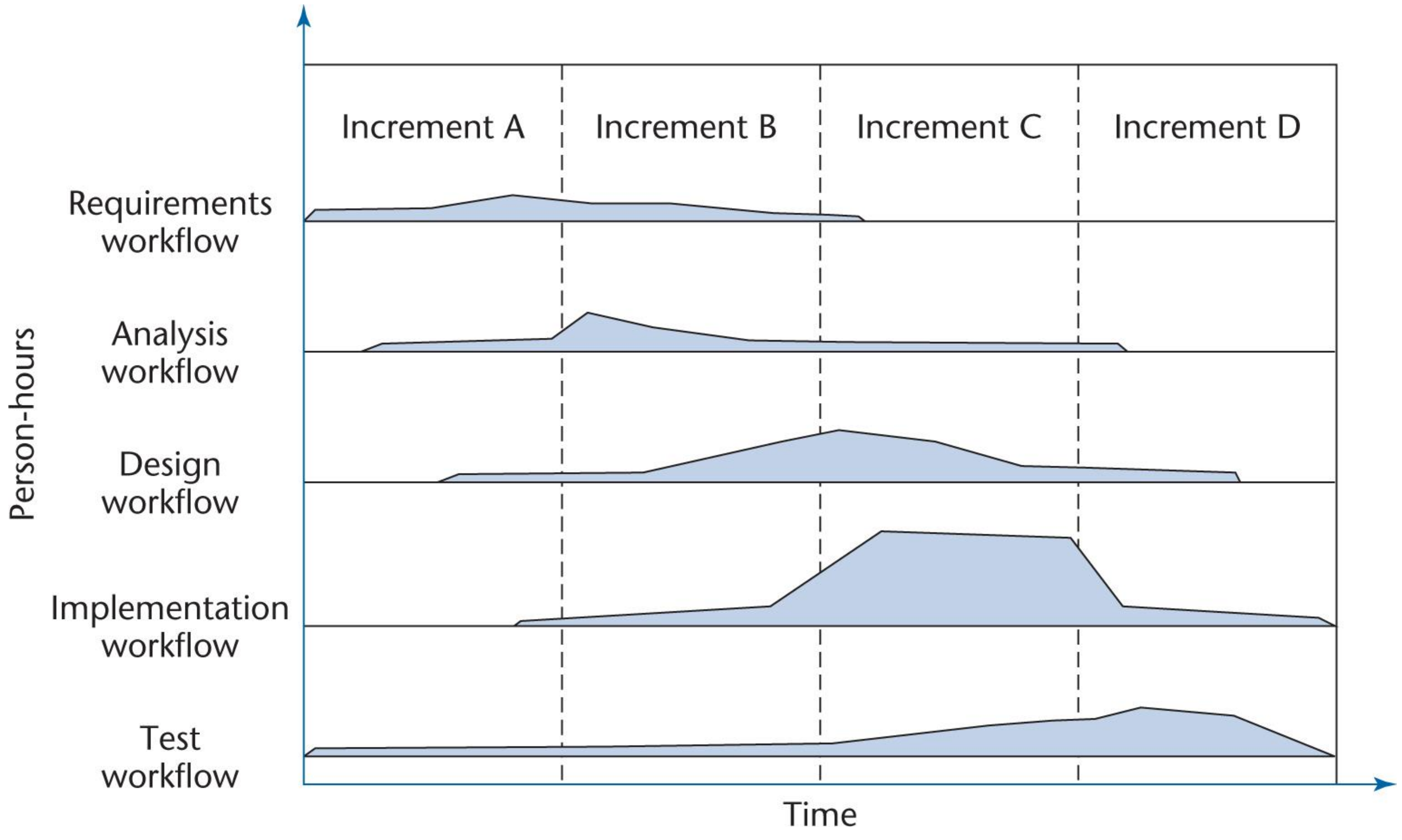
Iterative and Incremental Life-Cycle Model

- In real life, there is no single-phase at each episode. Instead, the operations of each phase is (***incremental***) spread out over the episode.
- The basic software development process is ***iterative.***
  - Each successive software version is intended to be closer to its target than its predecessor

Incremental Software Development - Miller's Law

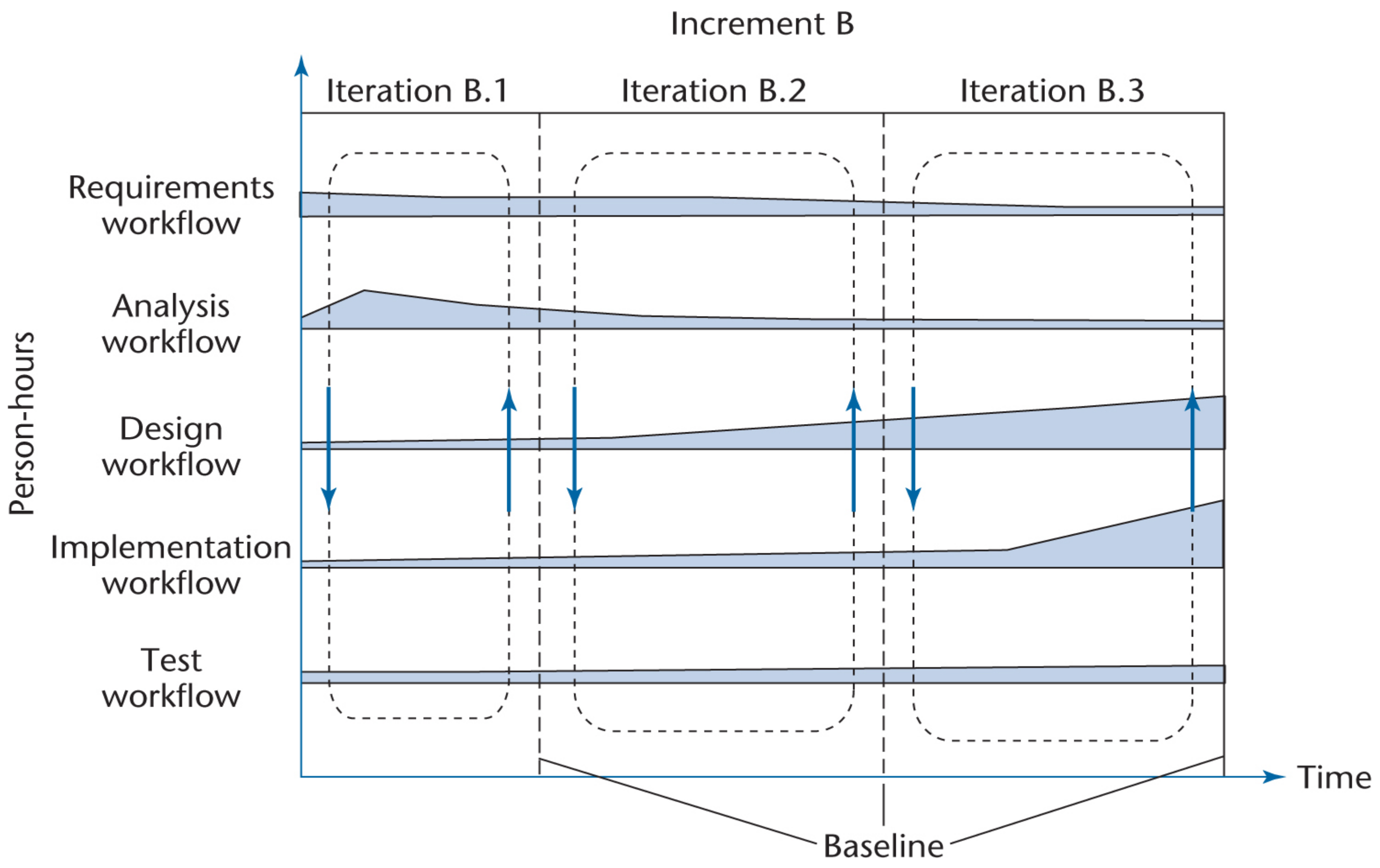
- At one time, we can concentrate on approximately seven *chunks* (units of information)
- To handle larger amounts of information, use ***stepwise refinement***
  - Every aspect is eventually handled in order of current importance.
    - Start with aspects that are currently the most important
    - Postpone aspects that are currently less important.
- This ***incremental*** process produces an **incremental** software development

Iterative and increment Life Cycle Model



Iterative and increment Life Cycle Model

- Iterative and increments are used in conjunction with one another
  - There are multiple workflows of each phase

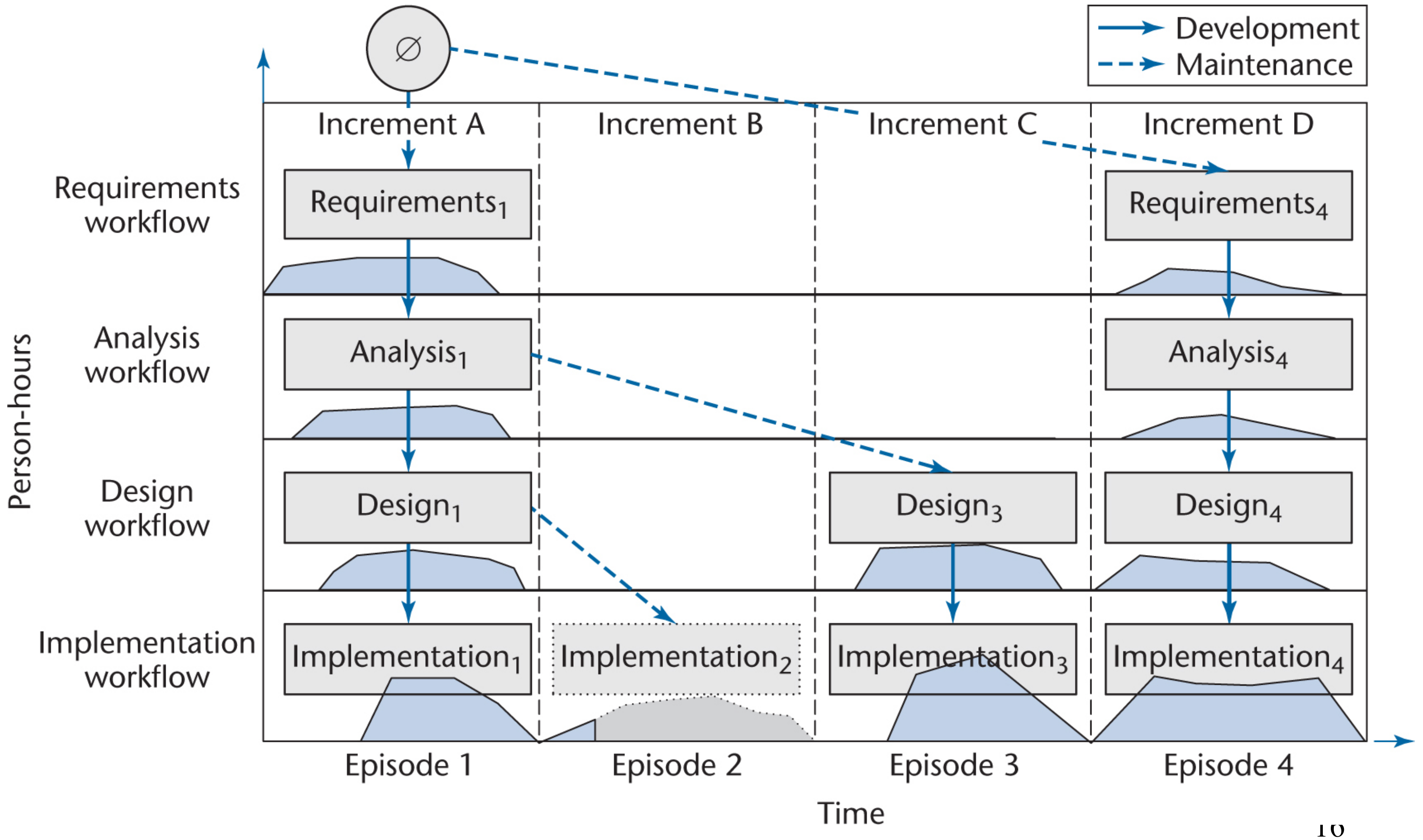


Iterative and increment Life Cycle Model

- Examples:
- At the beginning of the life cycle
    - The requirements workflow predominates
- At the end of the life cycle
    - The implementation and test workflows predominate
- Planning and documentation activities are performed throughout the life cycle

Evolution-Tree vs. Iterative-and-Increment

The evolution-tree model has been applied on the iterative-and-incremental life-cycle model

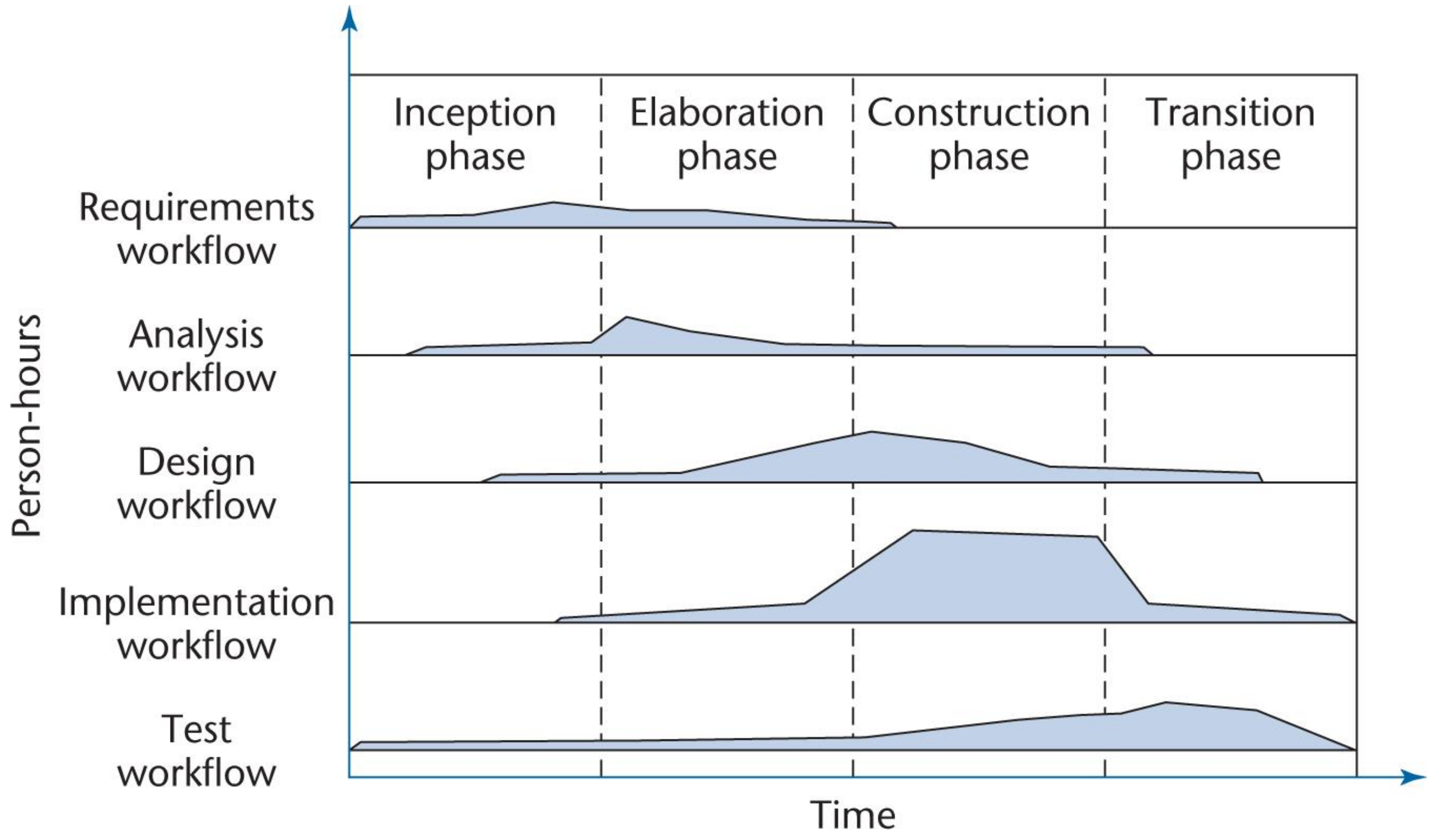


Evolution-Tree vs. Iterative-and-Increment

- Each episode corresponds to an increment
- Not every increment includes every workflow for example, Increment B is not completed
- Dashed lines denote maintenance
  - Episodes 2, 3: Corrective maintenance
  - Episode 4: Perfective maintenance
- Each increment is a waterfall mini-project

The Phases of the Unified Process (Increment Model)

The increments are identified as phases



The Phases of the Unified Process

- In practice, the phases of the Unified Processes are represented by the following increments:
  1. Inception (starting) phase
  2. Elaboration (preparation) phase
  3. Construction (design and implementation) phase
  4. Transition (installation and testing) phase

The Phases of the Unified Process

- Every step performed in the Unified Process falls into:
  - One of the five core workflows (requirements, analysis, design, implementation, testing)

  and

  - One of the four increments (inception, elaboration, construction, or transition)

1. The Inception Phase

- The inception phase aims to determine whether the proposed software product is feasible
  - ➢ Understand the domain
  - ➢ Build the business model
  - ➢ Define the scope of the proposed project and focus on the subset of the business model that is covered by the proposed software product
  - ➢ Start working with the initial business case

The Inception Phase - The Initial Business Case

Answers to the following questions are needed by the end of the inception phase so that the initial business case can be made.

- Is the proposed software product cost effective?
- How long will it take to obtain a return on investment?
- Alternatively, what will be the cost if the company doesn't develop the proposed software product?

The Inception Phase - The Initial Business Case

- If the software product is sold in the marketplace, do the necessary marketing studies been performed?
- Can the proposed software product be delivered in time?
- What will be the impact if the proposed software product is delivered late?

The Inception Phase - The Initial Business Case

- What are the risks involved in developing the software product
- How can these risks be reduced?
  - Does the team who will develop the proposed software product have the necessary experience?
  - Do we need a new hardware for this software product?
  - Is there a risk for not delivered the product in time?
  - Is there a way to reduce a risk by ordering back-up hardware from another supplier?
  - Any software tools needed?
  - Are the software tools currently available?
  - Do they have all the necessary functionalities?

The Inception Phase - Risks

There are three major risk categories:

1. Technical
2. Requirements

   The risk of getting the requirements incorrect.
3. Architecture

   The risk of getting the architecture incorrect. The architecture may not be sufficiently robust.

The Inception Phase – Reduce Risks

- To reduce all types of risks
  - The risks need to be ranked so that the critical risks are reduced first.

The Inception Phase: Analysis Workflow

- A small amount of the analysis may be performed during the inception phase.
- Information needed for the design of the architecture is extracted

The Inception Phase: Design Workflow

- A small amount of the design may be performed.
- Information needed for functional processes implementation is extracted

The Inception Phase: Implementation Workflow

- Coding is generally not performed during the inception phase. However, a *proof-of-concept prototype* is sometimes built to test the feasibility of constructing part of the software product.

The Inception Phase: Test Workflow

- The test workflow starts almost at the start of the inception phase
  - The aim is to ensure that the requirements have been accurately determined.

The Inception Phase: Planning (Future Trends)

- There is insufficient information at the beginning of the inception phase to plan the entire development
  - The only planning required at the start of the project is for the inception phase itself
- The planning that can be done at the end of the inception phase is the plan for only the elaboration phase

The Inception Phase: Documentation

The deliverables of the inception phase include:

- The initial version of the domain model
- The initial version of the business model (business case)
- The initial version of the requirements artifacts
- A preliminary version of the analysis artifacts
- A preliminary version of the architecture (design)
- The initial list of risks
- The initial ordering of the use cases
- The plan for the elaboration phase

The Inception Phase: The Initial Business Case

- Obtaining the initial version of the business case is the overall aim of the inception phase
- This initial version incorporates
  - A description of the scope of the software product
  - Financial details
  - If the proposed software product needs to be marketed, the business case will also include:

    Revenue projections, market estimates, initial cost estimates
  - If the software product needs to be used in-house, the business case will include
    - The initial cost–benefit analysis

2. Elaboration (Preparation) Phase

- The aim of the elaboration phase is to refine the initial requirements.
- The major activities of the elaboration phase:
  - Refine the architecture
  - Monitor the risks and refine their priorities
  - Refine the business case
  - Produce the project management plan

The Tasks of the Elaboration Phase

- The tasks of the elaboration phase correspond to:
  - Completing the requirements workflow
  - Performing virtually the entire analysis workflow
  - Starting the design of the architecture

The Elaboration Phase: Documentation

- The deliverables of the elaboration phase include:
  - The completed domain model
  - The completed business (business case)
  - The completed requirements artifacts
  - The completed analysis artifacts
  - An updated version of the architecture (design)
  - An updated list of risks
  - The updated project management plan (for the rest of the project)

3. Construction Phase

- The aim of the construction phase is to produce the first operational-quality version of the software product
  - This is sometimes called the **beta release**
- The tasks of this phase are focused on
  - Implementation
  - Testing
    - Unit testing of modules
    - Integration testing of subsystems
    - Product testing of the overall system

The Construction Phase: Documentation

- The deliverables of the construction phase include:
  - The initial user manual and other manuals, as appropriate
  - All the artifacts (beta release versions)
  - The completed architecture (design)
  - The updated risks list
  - The updated project management plan (for the remainder of the project)
  - If necessary, the updated business case

4. The Transition Phase

- The aim of the transition phase is to ensure that the client’s requirements have been met
  - Faults in the software product are corrected
  - All the manuals are completed
  - Discover any previously unidentified risks
- This phase is driven by feedback from the site(s) at which the beta release has been installed
- The deliverables (documentation) of the transition phase include:
  - All the artifacts (final versions)
  - The completed manuals