Design Defects and Restructuring

Engr. Abdul-Rahman Mahmood

abdulrahman@nu.edu.pk

alphapeeler.sf.net/pubkeys/pkey.htm

pk.linkedin.com/in/armahmood

www.twitter.com/alphapeeler

www.facebook.com/alphapeeler

abdulmahmood-sss   alphasecure

armahmood786

http://alphapeeler.sf.net/me

alphapeeler#9321

reddit.com/user/alphapeeler

www.flickr.com/alphapeeler

http://alphapeeler.tumblr.com

armahmood786@jabber.org

alphapeeler@aim.com

mahmood_cubix   48660186

alphapeeler@icloud.com

pinterest.com/alphapeeler

www.youtube.com/user/AlphaPeeler

Unified Process Introduction

Basic Characteristics of the Unified Process

- Object-oriented
- Use-case driven
- Architecture centric
- Iteration and incrementation

Basic Characteristics of the Unified Process

- ## Object-oriented
  - Utilizes object oriented technologies.
  - Classes are extracted during <u>OOA</u> and designed during <u>OOD</u>.

Use-case driven

- Utilizes use case model to describe complete functionality of the system

Req.ts	Analysis	Design	Impl.	Test

Use Cases bind these workflows together

Basic Characteristics of the Unified Process

- ## Architecture centric
  - Focus core <u>architecture</u> in the <u>early iterations</u>
  - In earliest iterations, get high valued requirements
  - View of the whole design with the <u>important characteristics made more visible</u>
  - Expressed with <u>class diagram</u>

Basic Characteristics of the Unified Process

Iteration and incrementation

- Way to divide the work

- Iterations are steps in the process, and increments are growth of the product

- The basic software development process is iterative

  - Each successive version is intended to be closer to its target than its predecessor

The Rational Unified Process

- RUP is a method of managing OO Software Development
- It can be viewed as a <u>Software Development Framework</u> which is extensible and <u>features:</u>
  - Iterative Development
  - Requirements Management
  - Component-Based Architectural Vision
  - Visual Modeling of Systems
  - Quality Management
  - Change Control Management

The Unified Process is Engineered
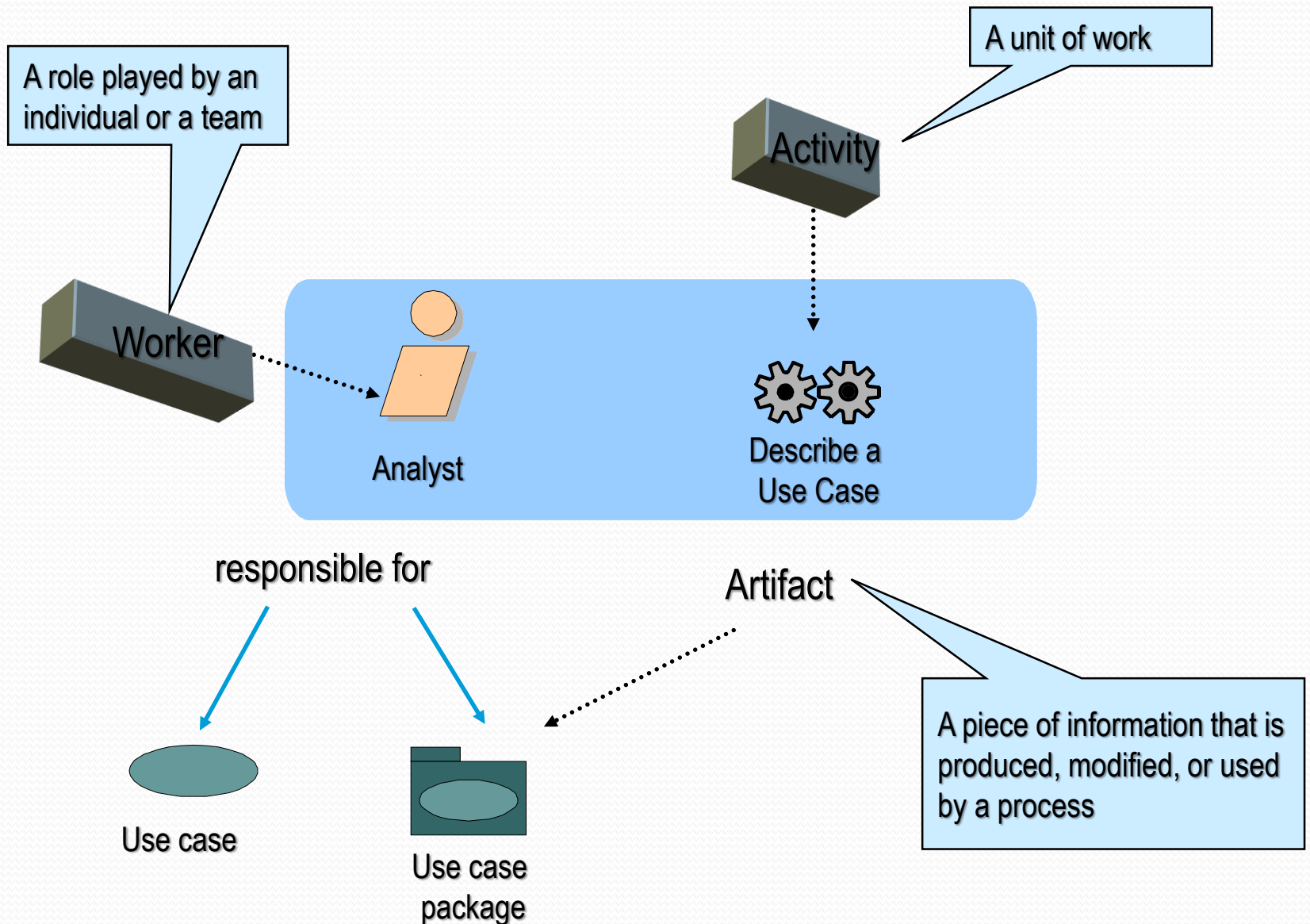
The Unified Process is a Process Framework
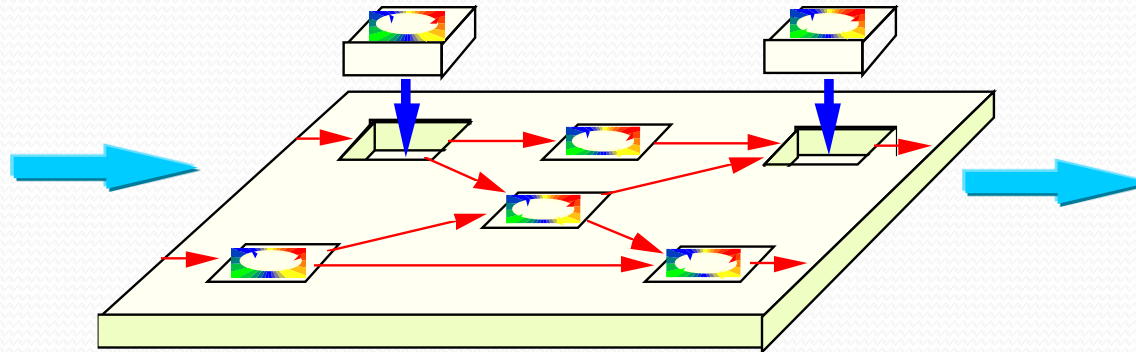
While the Unified Process is widely used, there is NO Universal Process!

- The Unified Process is designed for flexibility and extensibility
    - » allows a variety of lifecycle strategies
    - » selects what artifacts to produce
    - » defines activities and workers
    - » models concepts
    - » IT IS A PROCESS FRAMEWORK for development

Unified Process Model
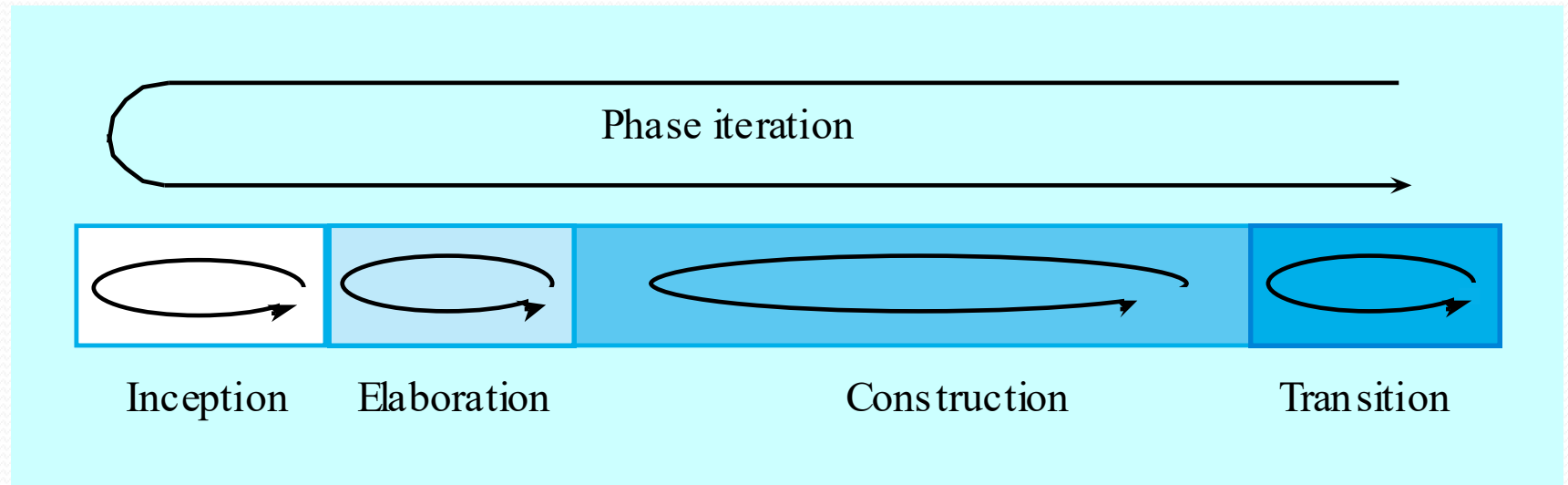
Goals and Features of Each Iteration

- Slowly chip away at the project risks:
    - performance risks
    - integration risks (different vendors, tools, etc.)
    - conceptual risks (hunt out analysis and design flaws)
- Perform a "miniwaterfall" project that ends with a delivery of something tangible in code.
- Each iteration is risk-driven.
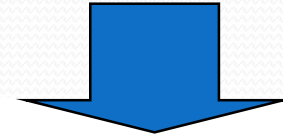- The result of a single iteration is an incremental improvement.

Unified Process Phases

Inception	Elaboration	Construction	Transition

- Inception
  - Define **business case**, **risks**, **10% requirements** identified, estimate next phase effort.
- Elaboration
  - **Understanding of problem / architecture**, **risk significant** units are coded/tested, **80%** requirements identified.
- Construction
  - System design, **programming and testing**.
- Transition
  - **Deploy** the system in its operating environment.

The Phases/Workflows of the Unified Process

.Phase is Business context of a step

Workflow is Technical context of a step

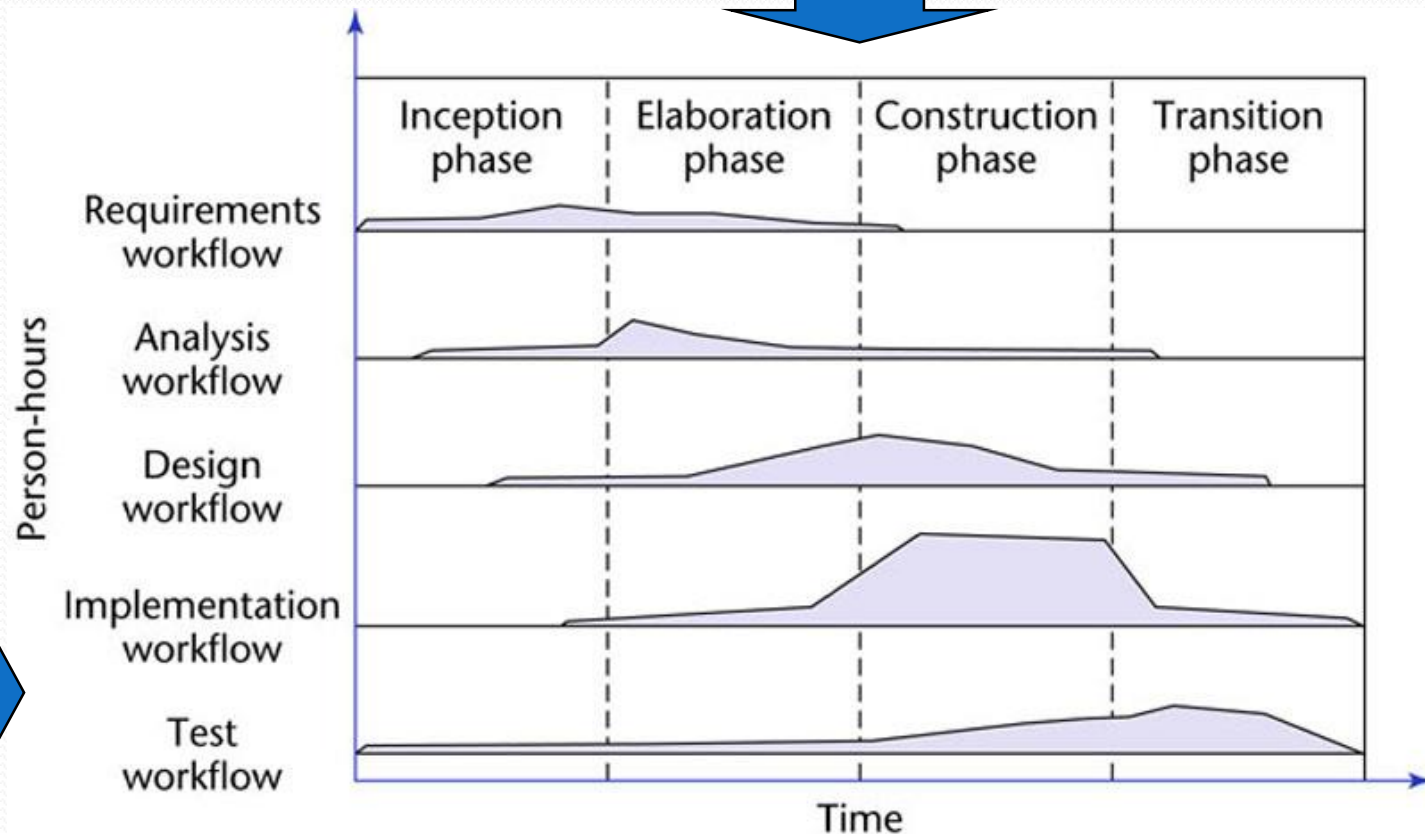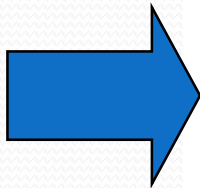|	Inception phase	Elaboration phase	Construction phase	Transition phase
Requirements workflow				
Analysis workflow				
Design workflow				
Implementation workflow				
Test workflow				

Person-hours

Time

Figure 3.1

The Phases/Workflows of the Unified Process

- **NOTE: Most of the requirements work or workflow is done in the inception phase.**

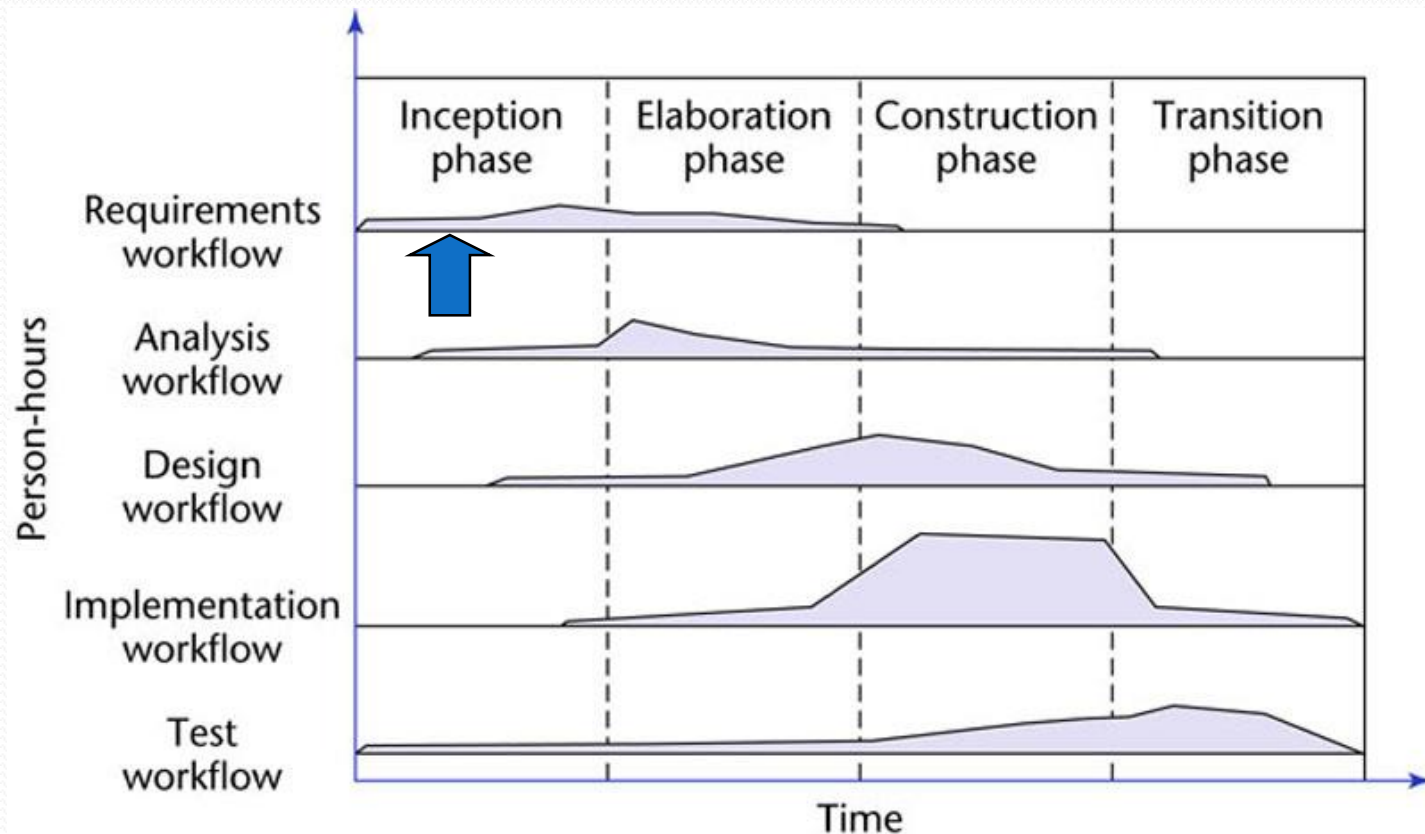- **However some is done later.**

Figure 3.1

The Phases/Workflows of the Unified Process

- **<u>NOTE:</u> Most of the implementation work or workflow is done in construction**
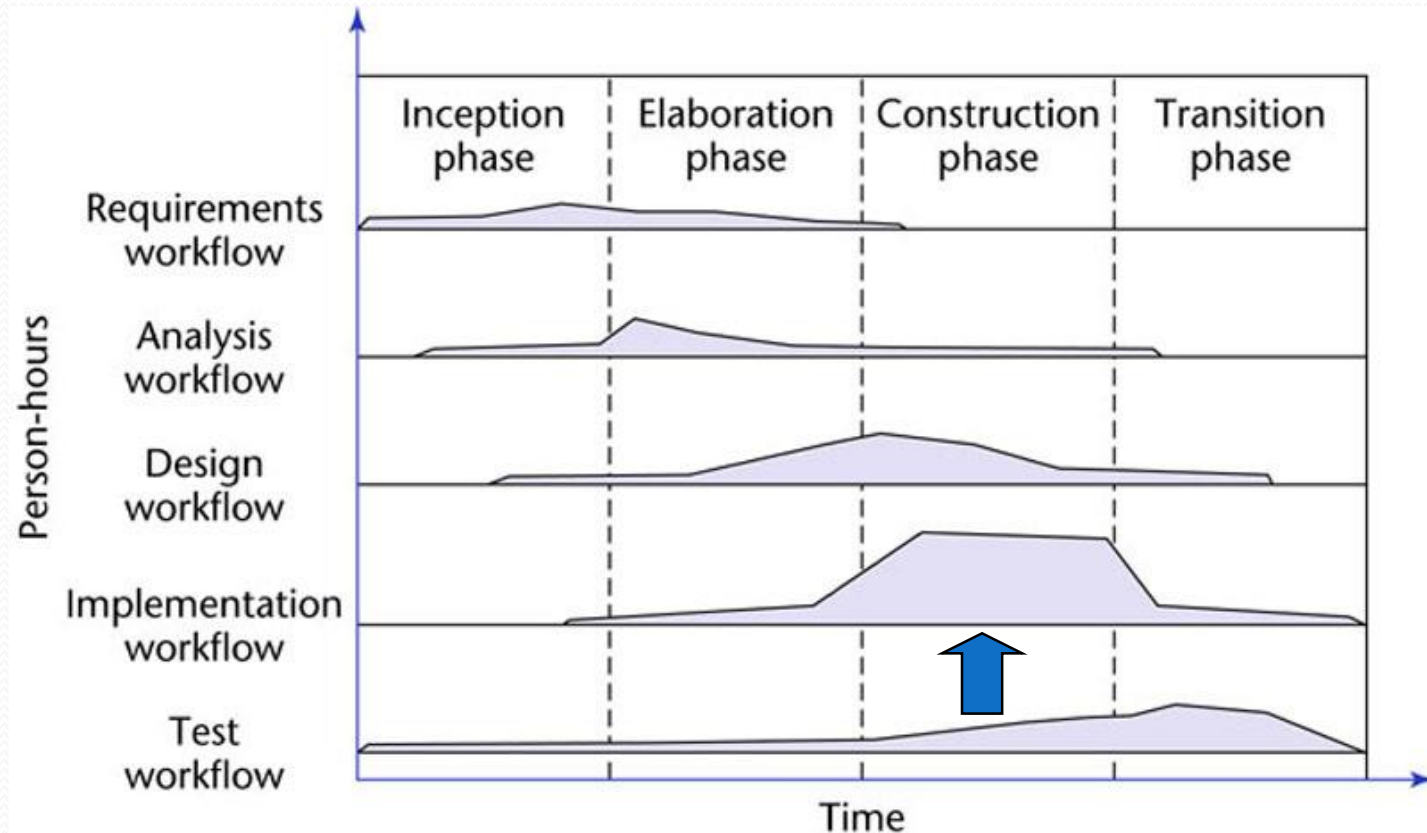
- **However some is done earlier and some later.**

Figure 3.1

Phase Deliverables

Inception Phase	Elaboration Phase	Construction Phase	Transition Phase
• The initial version of the domain model • The initial version of the business model • The initial version of the requirements artifacts • A preliminary version of the analysis artifacts • A preliminary version of the architecture • The initial list of risks • The initial ordering of the use cases • The plan for the elaboration phase • The initial version of the business case	• The completed domain model • The completed business model • The completed requirements artifacts • The completed analysis artifacts • An updated version of the architecture • An updated list of risks • The project management plan (for the rest of the project) • The completed business case	• The initial user manual and other manuals, as appropriate • All the artifacts (beta release versions) • The completed architecture • The updated risk list • The project management plan (for the remainder of the project) • If necessary, the updated business case	• All the artifacts (final versions) • The completed manuals

UP Life cycle in four phases

- Inception
- Elaboration
- Construction
- Transition

The Enterprise Unified Process (EUP) adds two more phases to this:

- *Production:* keep system useful/productive after deployment to customer

- *Retirement:* archive, remove, or reuse etc.

Example roles in UP

- *Stake Holder:* customer, product manager, etc.
- *Software Architect:* established and maintains architectural vision
- *Process Engineer:* leads definition and refinement of Development Case
- *Graphic Artist:* assists in user interface design, etc.

Some UP guidelines

- *Attack risks early* *on* and continuously so, before they will attack you
- Stay focused on *developing executable software* in early iterations
- Prefer *component-oriented* architectures and *reuse existing components*
- Quality is a way of life, not an afterthought

Six best "must" UP practices

1. Time-boxed iterations: _avoid speculative powerpoint architectures"_

2. _Strive for cohesive architecture_ and reuse existing components:

- e.g. core architecture developed by small, co-located team

- then early team members divide into sub-project leaders

Six best "must" UP practices

3. Continuously verify quality: _test early & often_, and realistically by integrating all software at each iteration

4. _Visual modeling_: prior to programming, do at least some visual modeling to explore creative design ideas

Six best "must" UP practices

5. *Manage requirements:* find, organize, and track requirements through skillful means

6. *Manage change:*
-   disciplined configuration management protocol, version control,
-   change request protocol
-   baselined releases at iteration ends

Unified Process Workflows

The Unified Process

- The Unified Process IS A
- 2-dimensional systems development process described by a
  - set of phases and (dimension one)
  - Workflows (dimension two)

The Unified Process

- Phases
  - Describe the business steps needed to develop, buy, and pay for software development.
  - The business increments are identified as phases

- Workflows
  - Describe the tasks or activities that a developer performs to evolve an information system over time

Process Overview

Workflow (tasks)	Phases (time)			
	Inception	Elaboration	Construction	Transition
Requirements				
Analysis				
Design				
Implementation				
Test				

workflows

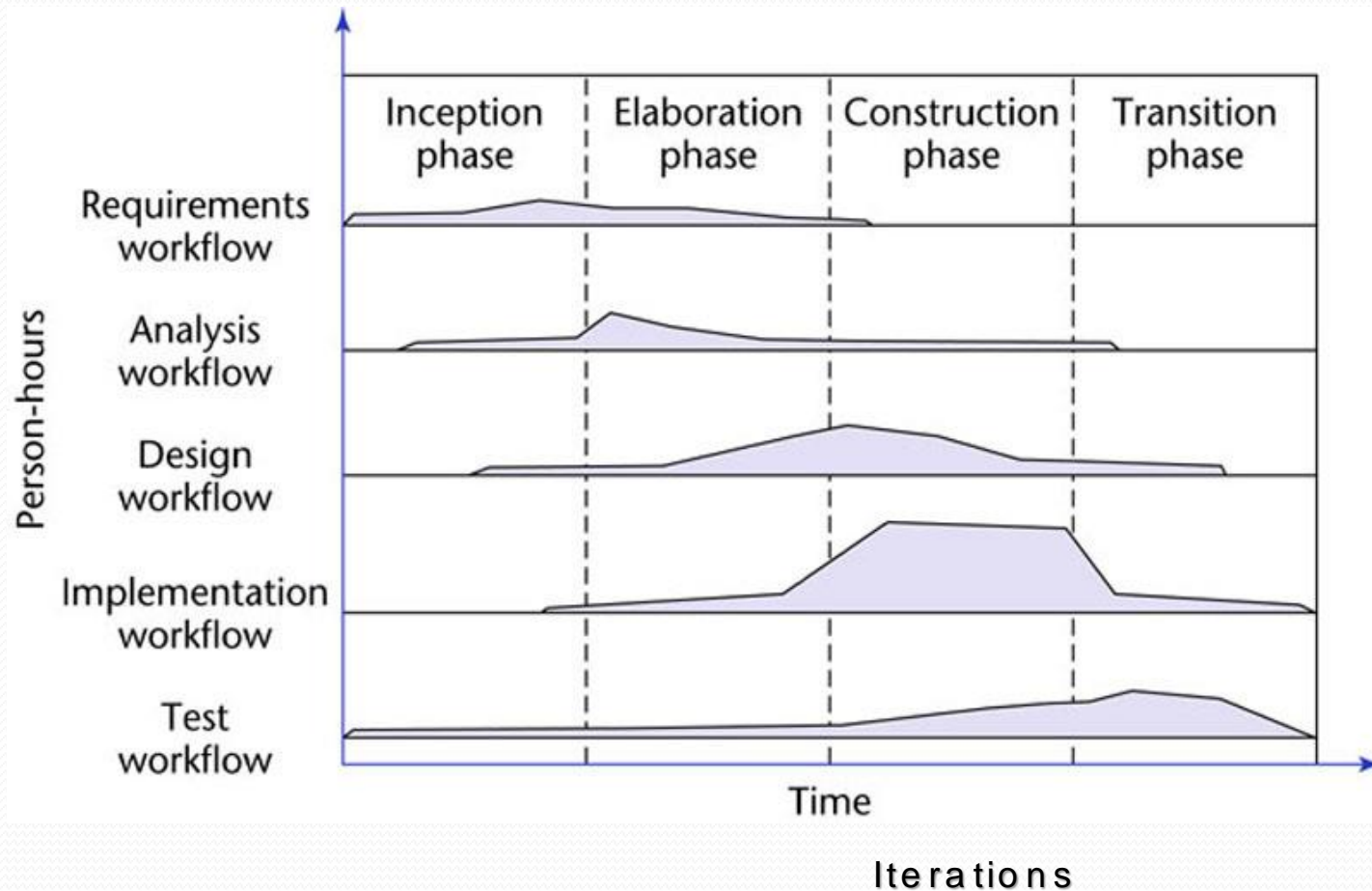| Workflow	Description
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.
Test	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and change management	This supporting workflow managed changes to the system (see Chapter 29).
Project management	This supporting workflow manages the system development (see Chapter 5).
Environment	This workflow is concerned with making appropriate software tools available to the software development team.

Primary Workflows

- The Unified Process

- PRIMARY WORKFLOWS
  - **Requirements workflow**
  - **Analysis workflow**
  - **Design workflow**
  - **Implementation workflow**
  - **Test workflow**
  - **Post delivery maintenance workflow**
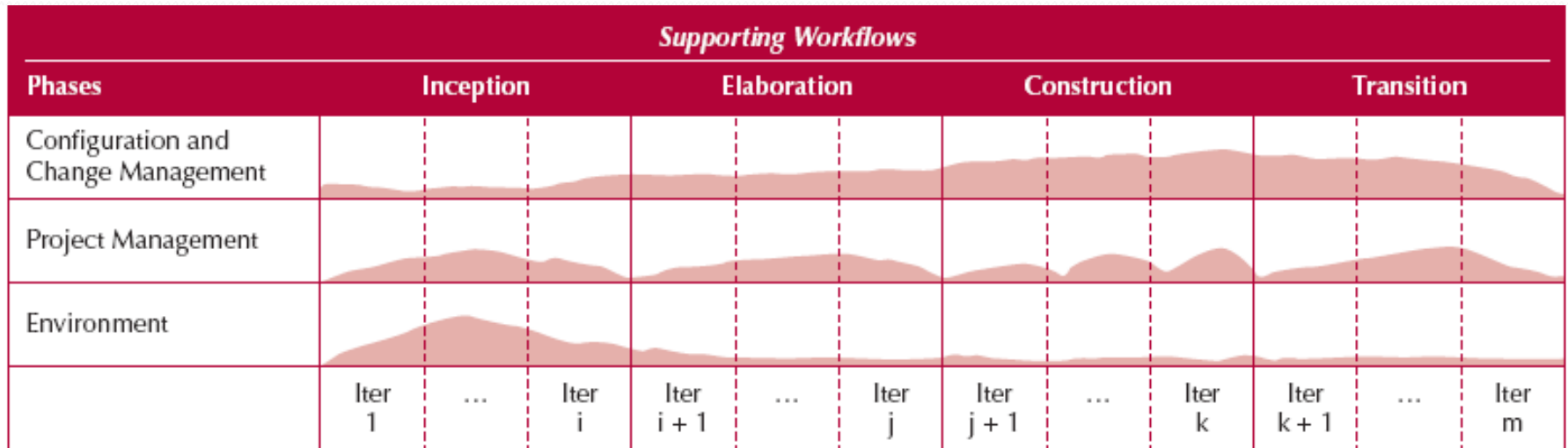
- Supplemental Workflows
  - Planning Workflow

Iterations and Workflow

Supporting Workflows of The Unified Process

Phases	Supporting Workflows											
	Inception			Elaboration			Construction			Transition		
Configuration and Change Management												
Project Management												
Environment												
	Iter 1	…	Iter i	Iter i + 1	…	Iter j	Iter j + 1	…	Iter k	Iter k + 1	…	Iter m

Software Project Management Plan

- Once the client has signed off the specifications, detailed planning and estimating begins

- We draw up the software project management plan, including
  - Cost estimate
  - Duration estimate
  - Deliverables
  - Milestones
  - Budget

- This is the earliest possible time for the SPMP

Post delivery Maintenance

- Post delivery maintenance is an essential component of software development
  - More money is spent on post delivery maintenance than on all other activities combined

- Problems can be caused by
  - Lack of documentation of all kinds

- Two types of testing are needed
  - Testing the changes made during post delivery maintenance
  - Regression testing

- All previous test cases (and their expected outcomes) need to be retained

Retirement

- Software is can be made unmaintainable because
  - A drastic change in design has occurred
  - The product must be implemented on a totally new hardware/operating system
  - Documentation is missing or inaccurate
  - Hardware is to be changed—it may be cheaper to rewrite the software from scratch than to modify it

- These are instances of maintenance (rewriting of existing software)
- True retirement is a rare event

- It occurs when the client organization no longer needs the functionality provided by the product