2. Object Oriented Software Development Process

IT 3106– Object Oriented Analysis and Design

Level II - Semester 3

Overview

In this section students will be exposed to the

- Object Oriented Systems Analysis
- Phases and Workflows of Unified Process

Intended Learning Outcomes

At the end of this lesson students will be able to

- define Object Oriented Systems Analysis and explain its benefits
- Identify the fundamental principles and characteristics of object-oriented systems analysis and design
- Explain the Unified Process, and its extensions
- describe the phases and workflows of Unified Process

List of Subtopics

2.1 Introduction to Object Oriented Systems Analysis and Design [Ref 1: Pg. 23-25]

2.2 Benefits of OOAD [Ref 1: Pg. 25]

2.3 The Unified Process [Ref 1: Pg. 25-34]

- 2.3.1 Phases
- 2.3.2 Workflows

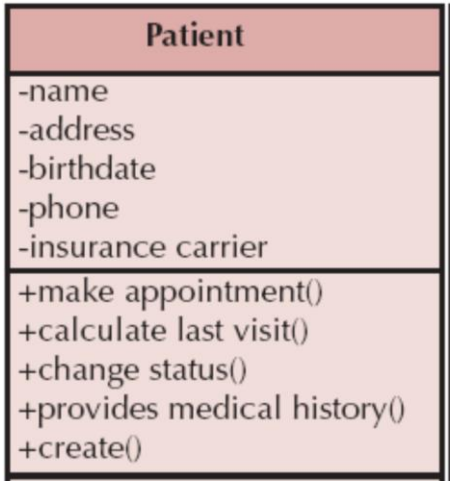
Ref 1: Alan Dennis, Barbara Haley, David Tegarden, Systems analysis design, An Object Oriented Approach with UML : an object oriented approach, 5th edition, John Wiley & Sons, 2015, ISBN 978-1-118-80467-4

2.1 Introduction to Object Oriented Systems Analysis and Design (OOAD)

- The primary difference between a traditional approach like structured design and an object-oriented approach is how a problem is decomposed.
- In traditional approaches, the problem-decomposition process is either process-centric or data-centric.
- In the object-oriented approach, the focus is on capturing the structure and behavior of information systems into small modules that combines both data and process.

An Example of a Class in OOAD

Contains both data and processes



Patient

-name
-address
-birthdate
-phone
-insurance carrier

+make appointment()
+calculate last visit()
+change status()
+provides medical history()
+create()

2.1 Introduction to Object Oriented Systems Analysis and Design (OOAD)

- According to the creators of the Unified Modeling Language (UML), Grady Booch, Ivar Jacobson, and James Rumbaugh, any modern object-oriented approach to developing information systems must be
  - ➢ use-case driven,
  - ➢ architecture-centric, and
  - ➢ iterative and incremental.

Use-Case driven,

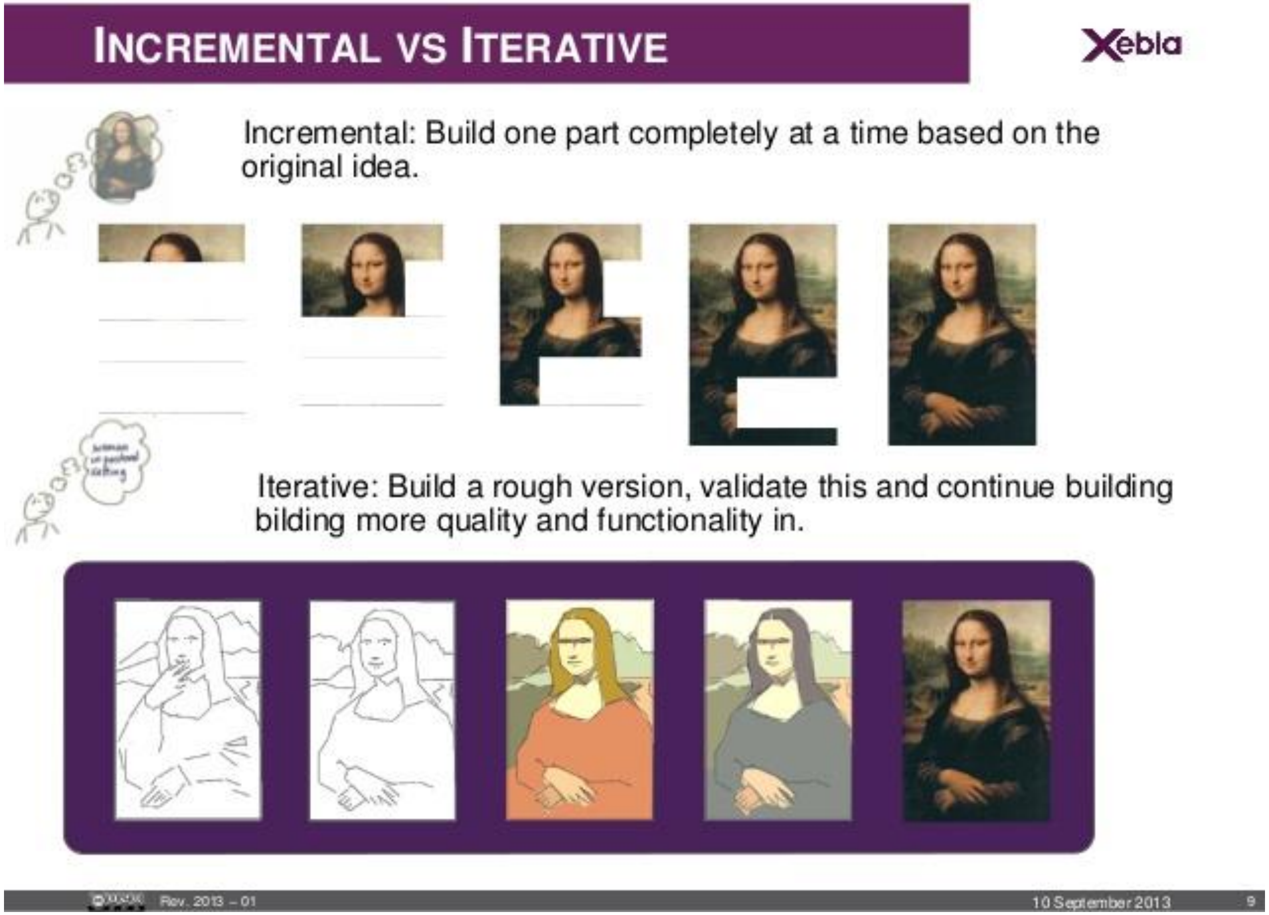
- Use Cases are the primary modeling tools defining the behavior of the system.
- A use case describes how the user interacts with the system to perform some activity, such as placing an order or making a reservation.
- The use cases are used to identify and to communicate the requirements for the system to the programmers. (Use Case Modeling is described in Section 4)
- Use cases are inherently simple because they focus on only one business process at a time. In contrast, the process model diagrams used by traditional structured methodologies are far more complex because they require the systems analyst and user to develop models of the entire system.
- In contrast to structured methodologies, a Use Case focuses on only one business process at a time, so developing models is much simpler.

Architecture-Centric

- *Architecture-centric* means that the underlying software architecture of the system drives the specification, construction, and documentation of the system.
- Modern object-oriented systems analysis and design approaches should support at least three separate but interrelated architectural views of a system: functional, static, and dynamic.
  - The **functional**, or external, view describes the behavior of the system from the perspective of the user. (eg Use Case Diagram)
  - The **structural**, or static, view describes the system in terms of attributes, methods, classes, and relationships. (eg. Class Diagram)
  - The **behavioral**, or dynamic, view describes the behavior of the system in terms of messages passed among objects and state changes within an object. (eg. Sequence, State diagrams)

Iterative and Incremental

- Object-oriented systems analysis and design approaches emphasize **iterative** and **incremental** development involving continuous testing and refinement throughout the life cycle.



INCREMENTAL VS ITERATIVE

Xebia

Incremental: Build one part completely at a time based on the original idea.

Iterative: Build a rough version, validate this and continue building bilding more quality and functionality in.

Rev. 2013 – 01

10 September 2013

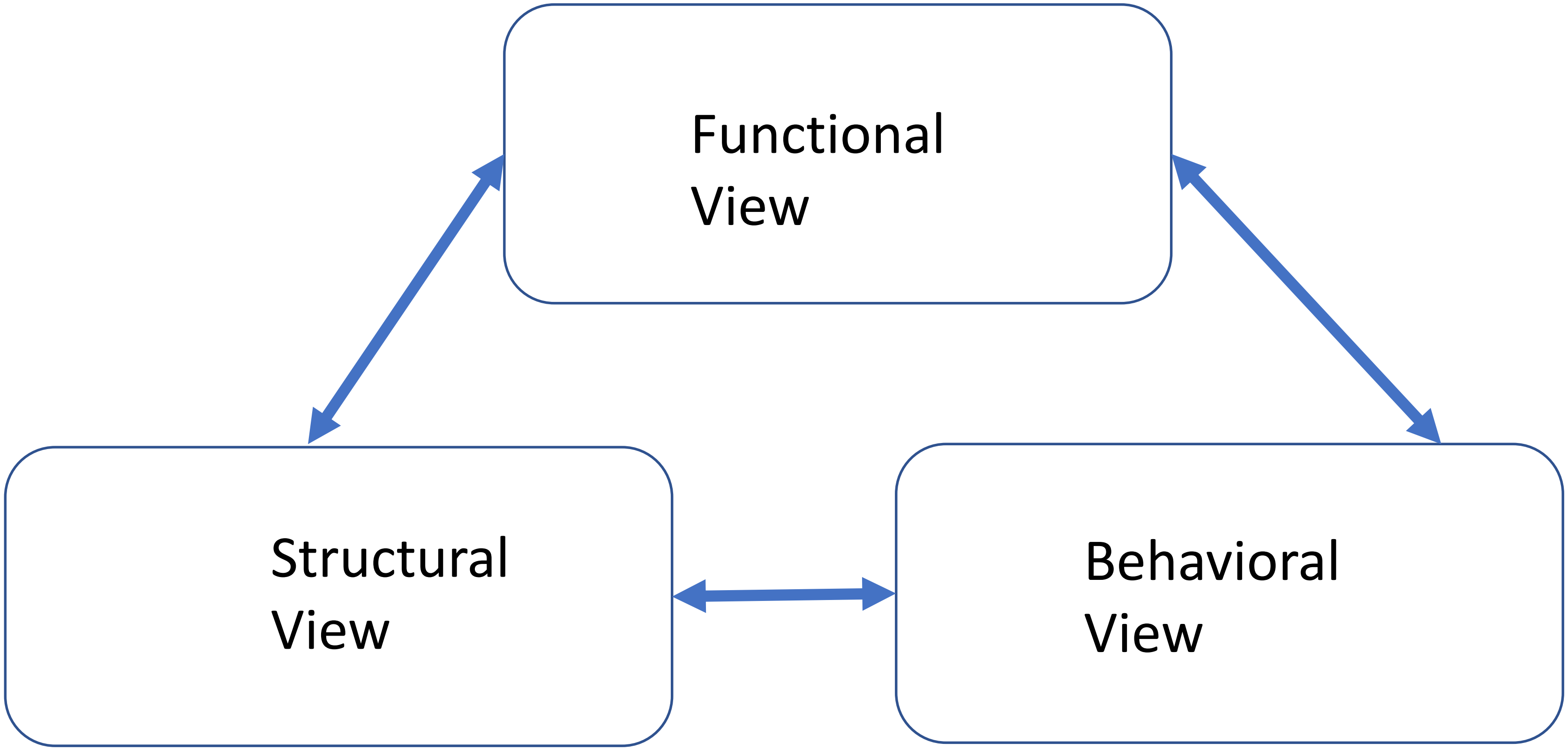
9

Iterative and Incremental cont…

- In other words Systems analysts(SA) develop their understanding of a user's problem by building up the three architectural views little by little.
- Here the approach is
    - SA will first work with the user to create a functional representation of the system. (Use Case model)
    - Next SA will attempt to build a structural representation of the evolving system. (Class Model)
    - Thereafter SA will distribute the functionality of the system over the evolving structure to create a behavioural representation of the evolving system. (Sequence diagram-or an Interaction model))

Iterative and Incremental cont…

Object Oriented



Functional View

Structural View

Behavioral View

2.2 Benefits of Object-Oriented Analysis Design

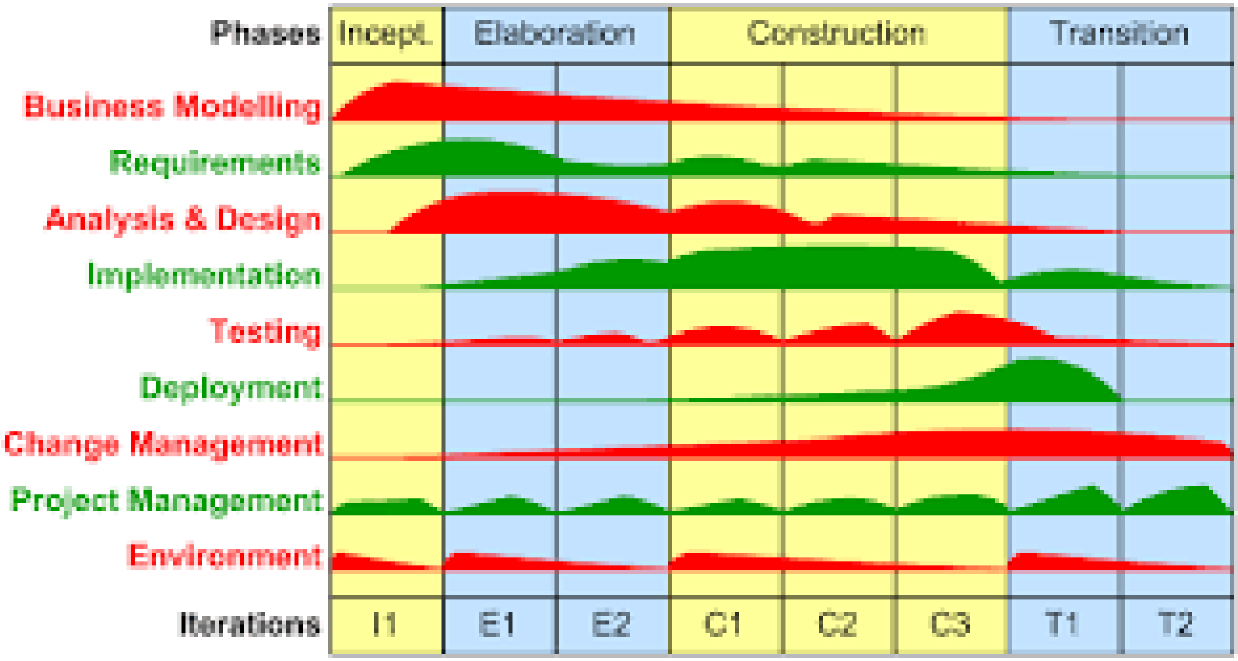
- OO concepts enable analysts to break a complex system into smaller, more-manageable modules, and later combine the modules to form an Information System.
- This modularity makes systems development easier to grasp, easier to share among members of a project team, and easier to communicate to users.
- By modularizing systems development, the project team actually is creating reusable pieces that can be plugged into other systems.
- When there are reusable components, new projects don’t have to start completely from scratch, hence can save development time.
- It improves the quality of the system due to program reuse. The principles of encapsulation and data hiding help the developer to develop systems that cannot be tampered by other parts of the system.

2.3 The Unified Process

- The Unified Process is a specific methodology that maps out when and how to use the various Unified Modeling Language (UML) techniques for object-oriented analysis and design
- The primary contributors were Grady Booch, Ivar Jacobsen, and James Rumbaugh (UML Authors).
- UML provides structural support for developing the structure and behavior of an information system, the Unified Process provides the behavioral support.
- The Unified Process is use-case driven, architecture-centric, and iterative and incremental.
- The Unified Process is a two-dimensional systems development process described by a set of phases and workflows.
- It is Popular with large scale projects

The Unified Process

- The phases Unified process are **inception**, **elaboration**, **construction**, and **transition**.
- The workflows include business modeling, requirements, analysis, design, implementation, test, deployment, configuration and change management, project management, and environment.

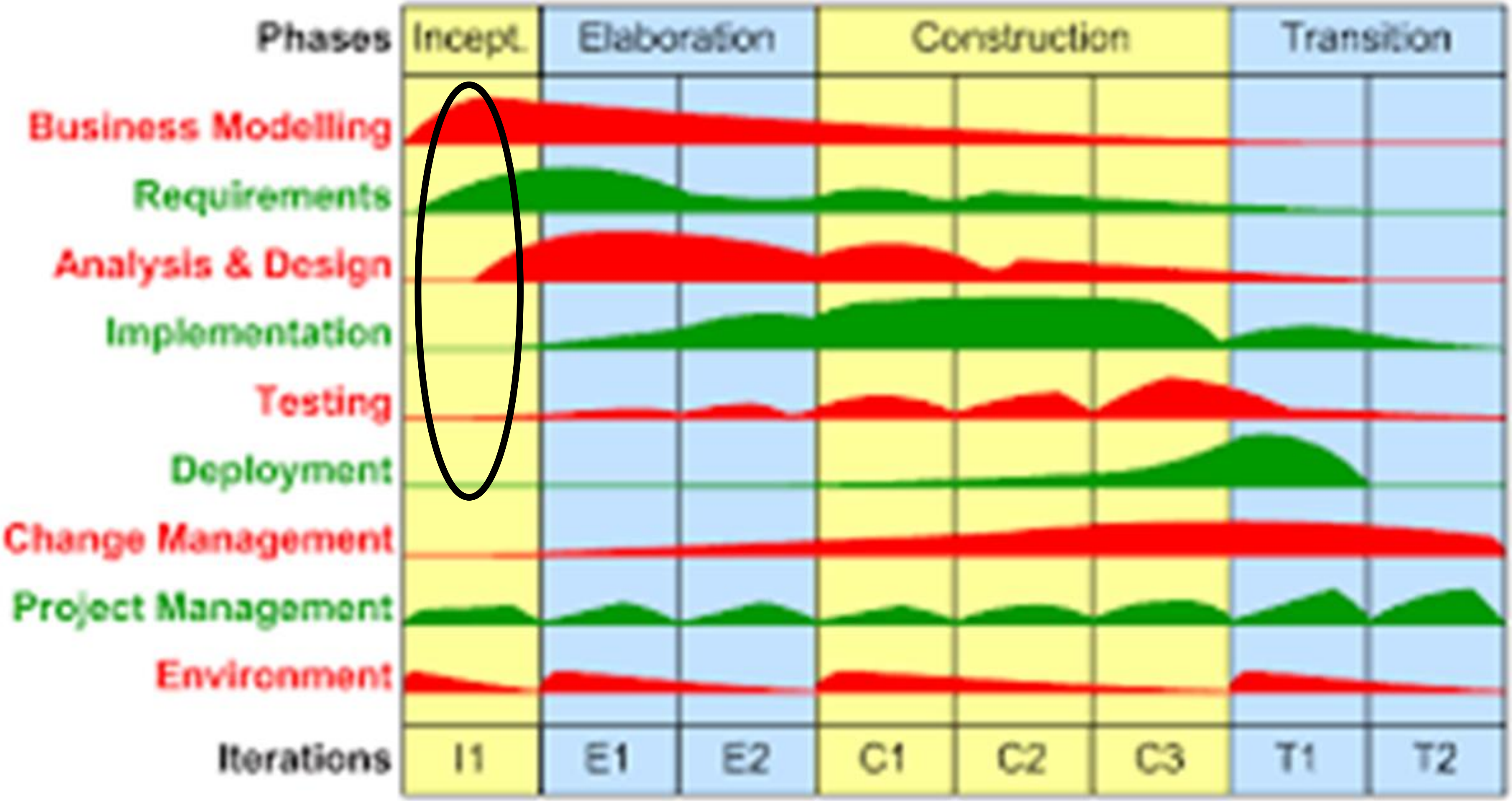


2.3.1 Phases of Unifies Process

- The *phases* of the Unified Process support an analyst in developing information systems in an iterative and incremental manner.
- The phases describe how an information system evolves through time.
- Depending on which development phase the evolving system is currently in, the level of activity varies over the *workflows.*
- Each workflow approximates the amount of activity that takes place during the specific phase. (see previous figure)

  Eg. The inception phase initially involves more on the Engineering work flows such as business modeling and requirements workflows while the test and deployment workflows are overlooked.

Phases of Unifies Process cont…



Inception phase initially involves more on business modeling and requirements workflows

© e-Learning Centre, UCSC

Phases of Unifies Process cont.…

- Each phase contains a set of iterations, and each iteration uses the various workflows to create an incremental version of the evolving system.
- As the system evolves through the phases, it improves and becomes more complete.
- Each phase has objectives, activities, and incremental deliverables.

Four Phases of Unifies Process

- **Inception** - Do we understand what we are building? Establish the business case for the project.
- **Elaboration** - Do we understand how to build it? Establish a project plan and a sound architecture
- **Construction** - Are we ready to release in the near future? Grow the system
- **Transition** - Can we release now? Supply the system to its end users

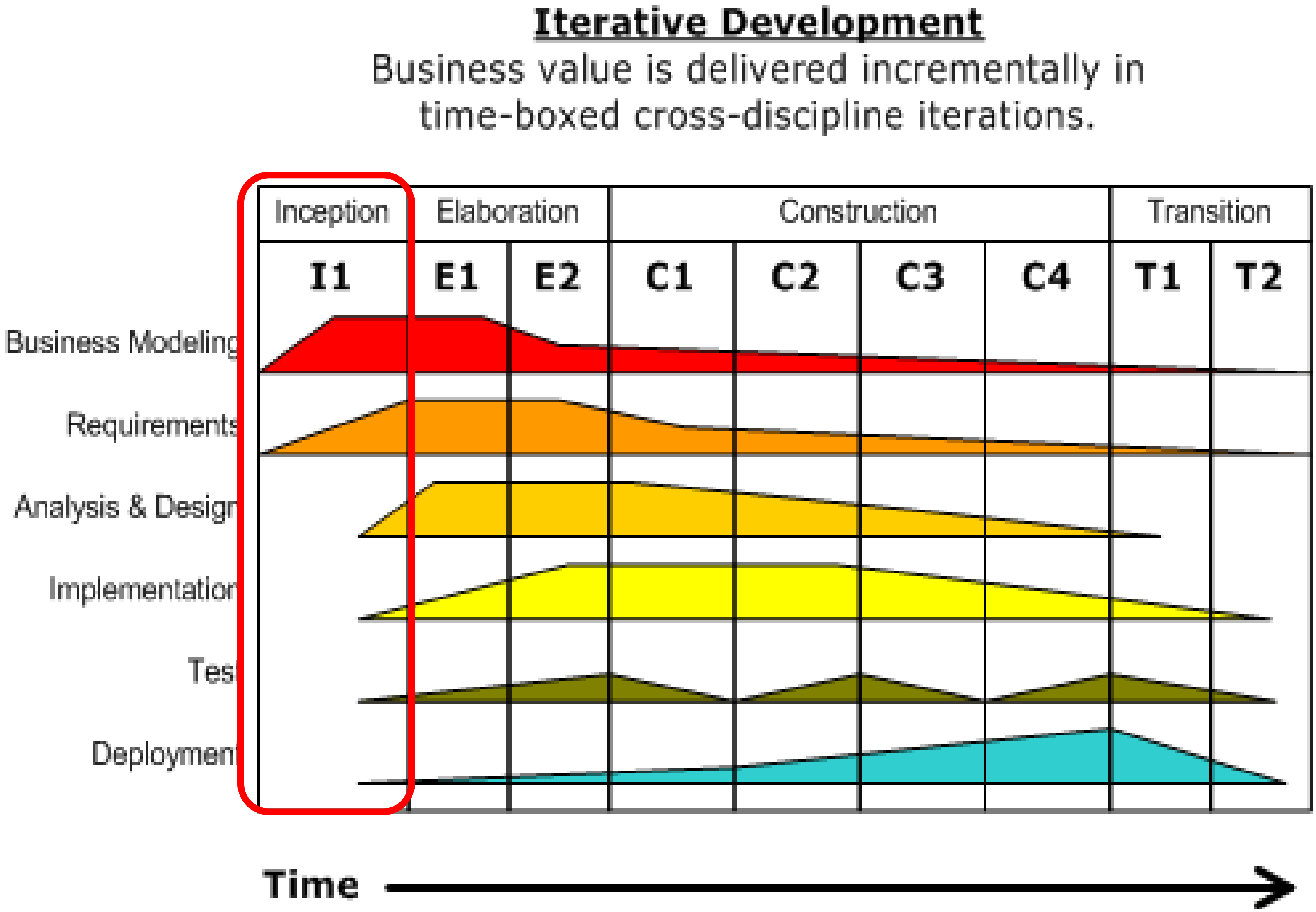
Inception Phase

- *Inception phase* is very similar to the planning phase of a traditional SDLC approach.
- In this phase, a business case is made for the proposed system. This includes feasibility analysis that should answer questions such as the following:
  - Do we have the technical capability to build it (technical feasibility)?
  - If we build it, will it provide business value (economic feasibility)?
  - If we build it, will it be used by the organization (organizational feasibility)?
- To answer these questions, the development team mainly performs work related primarily to the business modeling, requirements, and analysis workflows.

Inception Phase cont….

- Supporting workflows relevant to these flows are project management and environment (see figure given in slide 16)
- Important primary deliverables from the inception phase :
  - ➢a vision document that sets the scope of the project;
  - ➢primary requirements and constraints, Initial Use Case model presented to users
  - ➢the feasibility and risks associated with the project;
  - ➢necessary environment to develop the system;
  - ➢Initial Project Plan and major milestones with cost estimates;
  - ➢One or several prototypes serving as a proof of concept.
  - ➢some aspects of the problem domain classes being implemented and tested.

Inception Phase cont….



© e-Learning Centre, UCSC

Elaboration Phase

- When we typically think about object-oriented systems analysis and design, the activities related to the *elaboration phase* of the Unified Process are the most relevant.
- The analysis and design workflows are the primary emphasis during this phase.
- Continues with developing the vision document, while finalizing the business case.
- Revise the risk assessment and eliminate highest risk elements of the project.
- Establish and validate the System Architecture.
- Complete a project plan in sufficient detail to allow the stakeholders to be able to agree with constructing the actual final system.
- Gather the requirements, build the UML structural and behavioral models.

Elaboration Phase

- The primary deliverables of this phase include
  - the UML structure and behavior diagrams.
  - description of the software architecture
  - an executable of a baseline version of the evolving information system.
  - devised Business case and Risk list
- The baseline version serves as the foundation for all later iterations.
- By providing a solid foundation at this point, the developers have a basis for completing the system in the construction and transition phases.

Construction Phase

- The *construction phase* focuses heavily on programming the evolving information system.
- This phase is mainly concerned with the implementation workflow. However, the requirements workflow and the analysis and design workflows also are involved with this phase.
- Missing requirements are identified, and the analysis and design models are completed.
- During the last iteration, the deployment workflow uses intensely.
- The configuration and change management workflow, with its version-control activities, becomes extremely important during the construction phase.
- The primary deliverable of this phase is an implementation of the system that can be released for beta and acceptance testing.

Transition Phase

- Like the construction phase, the *transition phase* addresses aspects typically associated with the implementation phase of a traditional SDLC approach.
- Its primary focus is on the testing and deployment workflows. Essentially, the business modeling, requirements, and workflows should have been completed in earlier iterations of the evolving information system.
- Depending on the results from the testing workflow, some redesign and programming activities on the design and implementation workflows could be necessary, but they should be minimal at this point.
- From a managerial perspective, the project management, configuration and change management, and environment are involved.
- Some of the activities that take place are beta and acceptance testing, fine-tuning the design and implementation, user training, and rolling out the final product onto a production platform.
- The primary deliverable is the actual executable information system. The other deliverables include user manuals, a plan to support the users, and a plan for upgrading the information system in the future.

2.3.2 Workflows

- The workflows describe the tasks or activities that a developer performs to evolve an information system over time.
- The workflows of the Unified Process are grouped into two broad categories: engineering and supporting.

Engineering Workflows:

➢Engineering workflows include business-modeling, requirements, analysis, design, implementation, test, and deployment workflows.

➢The engineering workflows deal with the activities that produce the technical product (i.e., the information system).

Business Modeling Workflow

- Uncovers problems and identifies potential projects within a user organization.
- Aids management in understanding the scope of the projects.
- The primary purpose of business modeling is to ensure that both developer and user organizations understand where and how the to-be-developed information system fits into the business processes of the user organization.
- Primarily executed during the inception phase
- The activities that take place are most closely associated with the planning phase of the traditional SDLC; however, requirements gathering, and use-case and business process modeling techniques also help us to understand the business situation.

Requirements Workflow

- Includes extracting both functional and nonfunctional requirements.
- Typically, requirements are gathered from project stakeholders, such as end users, managers within the end user organization, and even customers.
- The requirements workflow is used mostly during the inception and elaboration phases.
- The identified requirements are very helpful for developing the vision document and the use cases used throughout the development process.
- Additional requirements tend to be discovered throughout the development process.

Analysis Workflow

- The analysis workflow is predominantly associated with the elaboration phase, but like the requirements workflow, it is possible that additional analysis will be required throughout the development process.
- Primarily addresses the creation of an analysis model of the problem domain.
- The analyst begins designing the architecture associated with the problem domain; using UML.
- the analyst creates structural and behavior diagrams that depict a description of the problem domain classes and their interactions.
- The primary purpose of the analysis workflow is to ensure that both the developer and user organizations understand the underlying problem and its domain without overanalyzing.
- A second purpose of the analysis workflow is to identify useful reusable classes for class libraries. By reusing predefined classes, you can speed up the development process.

Design Workflow

- The design workflow transitions the analysis model into a design model that can be used to implement the system
- The analysis workflow concentrated on understanding the problem domain, whereas the design workflow focuses on developing a solution that will execute in a specific environment.
- Basically, the design workflow simply enhances the description of the evolving system by adding classes that address the environment (boundary classes) of the system to the evolving analysis model.
- The design workflow uses activities such as detailed problem domain class design, optimization of the evolving information system, database design, user-interface design, and physical architecture design.
- The design workflow is associated primarily with the elaboration and construction phases of the Unified Process.

Implementation Workflow

- The primary purpose of the implementation workflow is to create an executable solution based on the design model.
- This includes not only writing new classes but also incorporating reusable classes from executable class libraries into the evolving solution.
- The new classes and their interactions with the incorporated reusable classes must be tested.
- In the case of multiple groups performing the implementation, the programmers also must integrate individually tested modules to create an executable version of the system.
- The implementation workflow is associated primarily with the elaboration and construction phases.

Testing Workflow

- The primary purpose of the testing workflow is to increase the quality of the evolving system.
- Testing includes unit testing , integration testing, user acceptance testing, and the actual alpha testing of the soft ware.
- Testing should go on throughout the development of the system.
- Testing of the analysis and design models occurs during the elaboration and construction phases, whereas implementation testing is performed primarily during the construction and, to some degree, transition phases.
- Basically, at the end of each iteration during the development of the information system, some type of test should be performed.

Deployment Workflow

- Mostly associated with the transition phase of the Unified Process.
- Includes activities such as soft ware packaging, distribution, installation, and beta testing.
- When deploying the new system into a user organization, the developers might have to convert the current data, interface the new soft ware with the existing soft ware, and train the end user to use the new system.

2.3.2 Workflows cont….

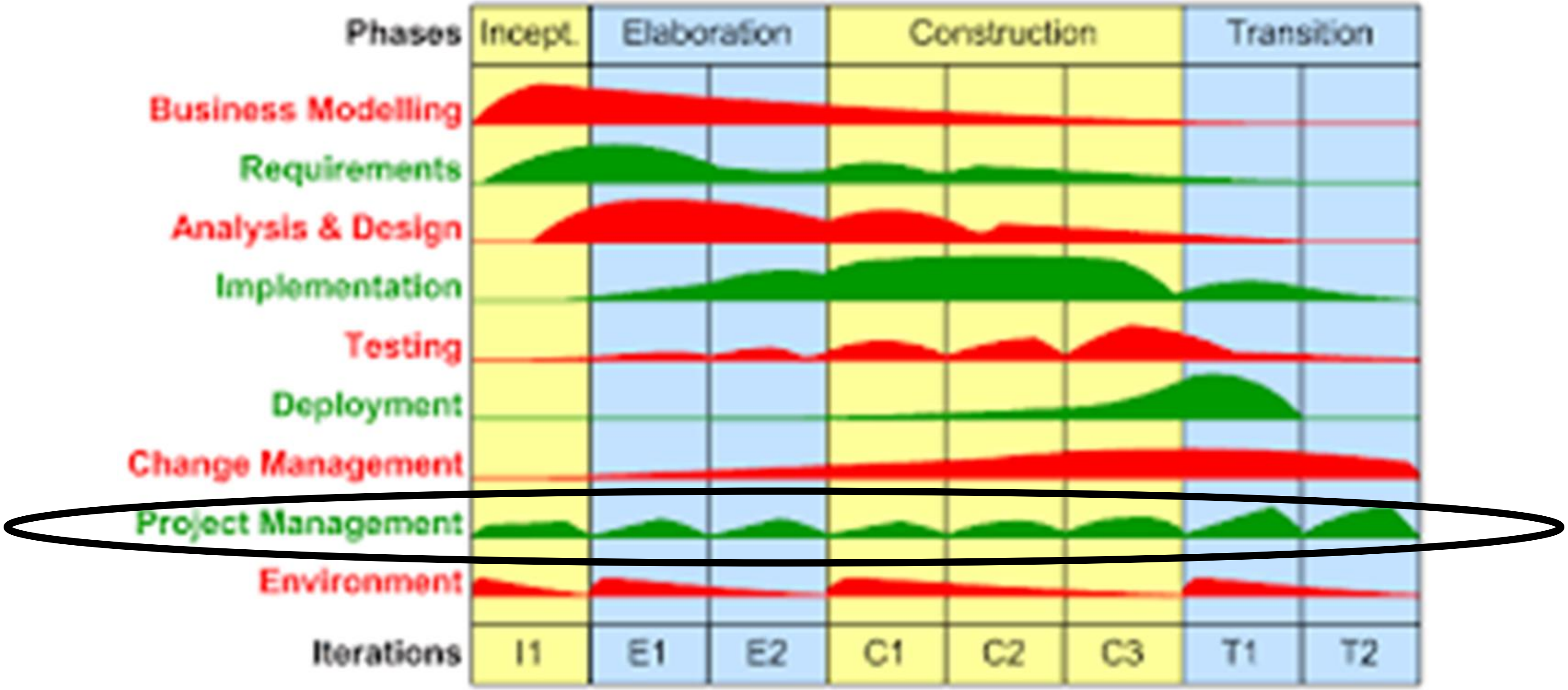
- The workflows describe the tasks or activities that a developer performs to evolve an information system over time.
- The workflows of the Unified Process are grouped into two broad categories: engineering and supporting.
- Already looked at **engineering workflows**

Supporting Workflows

- Include the project management, configuration and change management, and environment workflows.
- The supporting workflows focus on the managerial aspects of information systems development.

Project Management Workflow

- Whereas the other workflows associated with the Unified Process are technically active during all four phases, the project management workflow is the only truly cross-phase workflow.



Project Management Workflow

- Information systems tend to grow or evolve over time. At the end of each iteration, a new incremental version of the system is ready for delivery.
- The project management workflow is quite important owing to the complexity of the two-dimensional development model of the Unified Process (workflows and phases).
- This activities of the workflow include
  - ➢ identifying and managing risks, managing scope,
  - ➢ estimating the time to complete each iteration and the entire project,
  - ➢ estimating the cost of the individual iteration and the whole project, and
  - ➢ tracking the progress being made toward the final version of the evolving information system.

Configuration and Change Management Workflow

- The configuration and change management workflow are associated mostly with the construction and transition phases.
- The primary purpose of the configuration and change management workflow is to keep track of the state of the evolving system.
- The evolving information system comprises a set of artifacts (e.g., diagrams, source code, and executables).
- During the development process, these artifacts are modified.
- The artifacts themselves should be handled as any expensive asset would be handled—access controls must be put into place to safeguard the artifacts from being stolen or destroyed.
- The artifacts are modified on a regular basis; therefore, a good version control mechanism should be established.
- Finally, a good deal of project management , information needs to be captured (e.g., author, time, and location of each modification).

Environment Workflow

- During the development of an information system, the development team needs to use different tools and processes.
- The environment workflow addresses these needs. For example, a CASE tool that supports the development of an object-oriented information system via the UML could be required.
- Other necessary tools include programming environments, project management tools, and configuration management tools.
- The environment workflow involves acquiring and installing these tools.
- Even though this workflow can be active during all the phases of the Unified Process, it should be involved primarily with the inception phase.

Extension to Unified Process

❖Some weaknesses of Unified Process

- It does not address staffing, budgeting, or contract management issues.
- It does not address issues relating to maintenance, operations, or support of the product once it has been delivered.
- does not address cross- or inter-project issues.

❖Therefore it is not a complete software process; it is only a development process.

❖To address these omissions, Ambler and Constantine suggest adding a **production phase** and two workflows: the **operations and support workflow** and the **infrastructure management workflow**.

❖In addition to these new workflows, the test, deployment, and environment workflows are modified, and the project management and the configuration and change management workflows are extended into the production phase.

Production Workflow

- Concerned primarily with issues related to the software product after it has been successfully deployed.
- Focuses on issues related to updating, maintaining, and operating the soft ware.
- Unlike the previous phases, there are no iterations or incremental deliverables.
- Based on the activities that take place during this phase, no engineering workflows are relevant.
- The supporting workflows that are active during this phase include
  - ➢configuration and change management workflow,
  - ➢project management workflow,
  - ➢new operations and support workflow, and
  - ➢infrastructure management workflow.

Operations and Support Workflow

- Addresses issues related to supporting the current version of the software and operating the software daily.
- Activities include
  - creating plans for the operation and support of the soft ware product once it has been deployed,
  - creating training and user documentation,
  - putting into place necessary backup procedures,
  - monitoring and optimizing the performance of the soft ware, and
  - performing corrective maintenance on the soft ware.
- This workflow becomes active during the construction phase; its level of activity increases throughout the transition and, finally, the production phase.

Infrastructure Management Workflow

- Primary purpose is to support the development of the infrastructure necessary for the development.
- When the development and maintenance of a problem-domain architecture model goes beyond the scope of a single project and reuse is going to occur, the infrastructure management workflow is essential.

Existing Workflow Modifications and Extensions

- In addition to the workflows that were added to address deficiencies contained in the Unified Process, existing workflows had to be modified and/or extended into the production phase.
- These workflows include the test, deployment, environment, project management, and configuration and change management workflows.

Refinement of the Unified Process

- Rational Unified Process (RUP) is a refinement of the Unified Process that was created by Rational Software (now owned by IBM).
- It uses a series of software tools along with a process framework to define how to carry out the activities needed to run a software project,
- A modern generic process derived from the work on the UML and associated process
- RUP is included in the IBM Rational Method Composer (RMC) product which allows customization of the process.
- Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP) developed by Scott Ambler. It describes a simple, easy to understand approach to developing business application software using agile techniques
- The Open Unified Process (OpenUP) is a part of the Eclipse Process Framework (EPF), an open source process framework developed within the Eclipse Foundation. Its goals are to make it easy to adopt the core of the Rational Unified Process (RUP) / Unified Process.