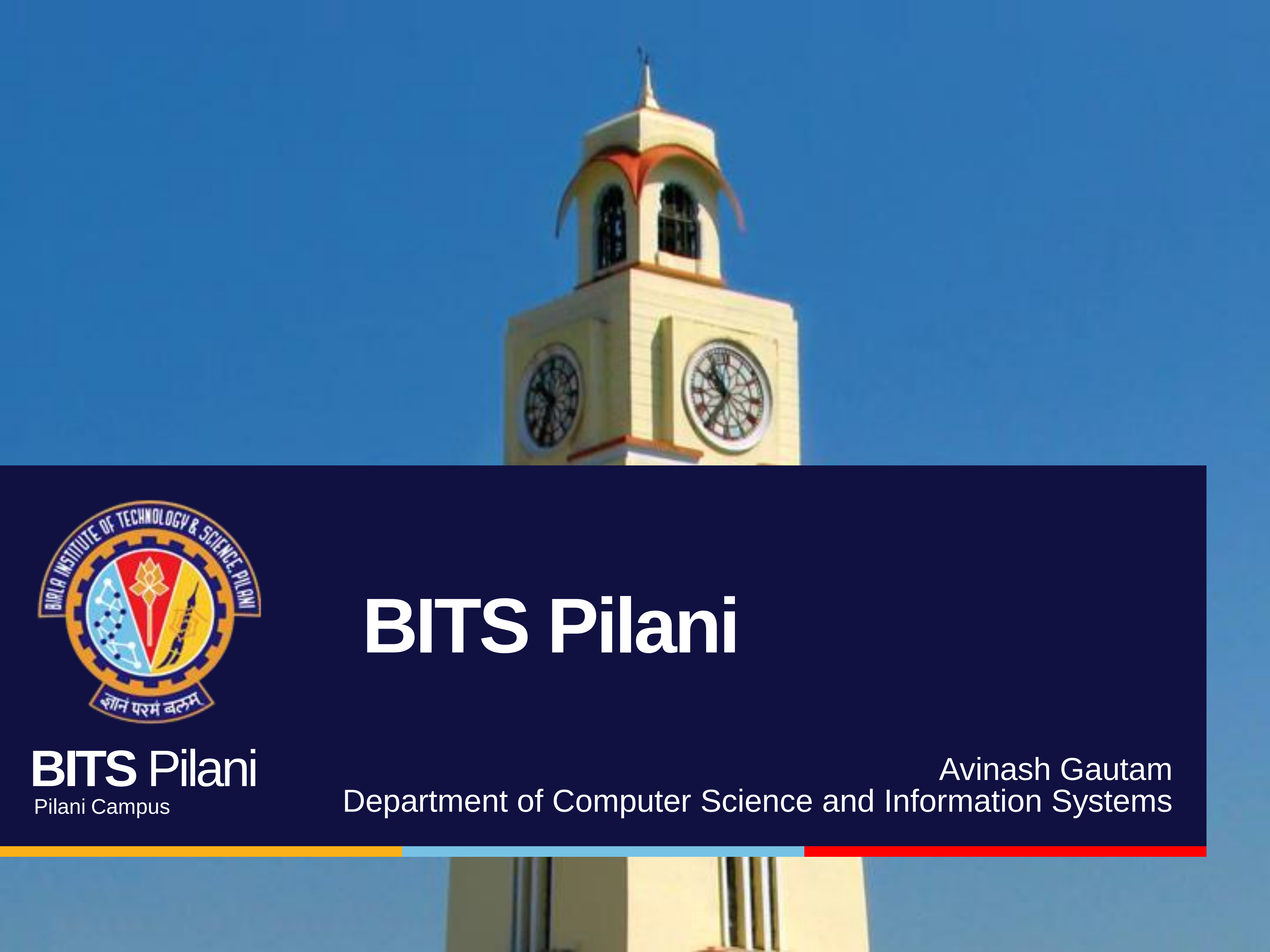


BITS Pilani

BITS Pilani
Pilani Campus

Avinash Gautam
Department of Computer Science and Information Systems

Lecture - 2

Object Oriented Analysis and Design

Chapter 1

Applying UML and Patterns

Craig Larman

Applying UML

- UML is just a standard diagramming notation. It is just a tool, not a skill that is valuable in itself. Knowing UML helps you communicate with others in creating software, but the real work in this course is learning Object-Oriented Analysis and Design, not how to draw diagrams.

Assigning Responsibilities

- The most important skill in Object-Oriented Analysis and Design is assigning responsibilities to objects. That determines how objects interact and what classes should perform what operations.

Requirements Analysis

- All Software Analysis and Design is preceded by the analysis of requirements.
- One of the basic principles of good design is to defer decisions as long as possible. The more you know before you make a design decision, the more likely it will be that the decision is a good one.
- TFCL: ***Think First, Code Later!***

Use Cases

- Writing Use Cases is not a specifically Object Oriented practice. But it is a best practice for elaborating and understanding requirements. So we will study Use Cases.

The Unified Process

- A standardized approach to analysis and design helps to ensure that all necessary tasks are understood and completed in software development.
- This text, and the course, will focus on the Unified Process developed at Rational Software by Ivar Jacobsen, Grady Boch, Jim Rumbaugh, and others.

Other Necessary Skills

- Requirements Analysis, Object-Oriented Analysis and Object-Oriented Design are not a complete toolkit for a software developer. There are many other skills necessary in Software development, including programming. This course only covers a subset of the necessary skills.

What is Object Oriented Analysis?

- The emphasis is on finding and describing the objects (or concepts) in the problem domain.
- In a Library Information System, some of the concepts include *Book*, *Library*, and *Patron*.

What is Object Oriented Design?

- The emphasis is defining software objects and how they collaborate to fulfill the requirements.
- In a Library Information System, a *Book* software object may have a *title* attribute and a *get Chapter* method.

Implementation

- During *Implementation*, or *Object-Oriented Programming*, design objects are implemented, such as a book class in Java.
- Implementation is also known as *Coding* or *Construction*.

Example Tasks

- Define Use Cases
- Define a Domain Model
- Define Interaction Diagrams
- Define Design Class Diagrams

Iterative development and The Unified process

Chapter 2

Applying UML and Patterns

-Craig Larman

The Unified Process

- The Unified Process has emerged as a popular and effective software development process.
- In particular, the Rational Unified Process, as modified at Rational Software, is widely practiced and adopted by industry.

The Most Important Concept

- The critical idea in the Rational Unified Process is *Iterative Development*.
- Iterative Development is successively enlarging and refining a system through multiple iterations, using feedback and adaptation.
- Each iteration will include requirements, analysis, design, and implementation.
- Iterations are *timeboxed*.

Why a new methodology?

- The philosophy of process-oriented methods is that the requirements of a project are completely frozen before the design and development process commences. As this approach is not always feasible, there is also a need for flexible, adaptable and agile methods, which allow the developers to make late changes in specifications.

What is Rational Unified Process (RUP)?

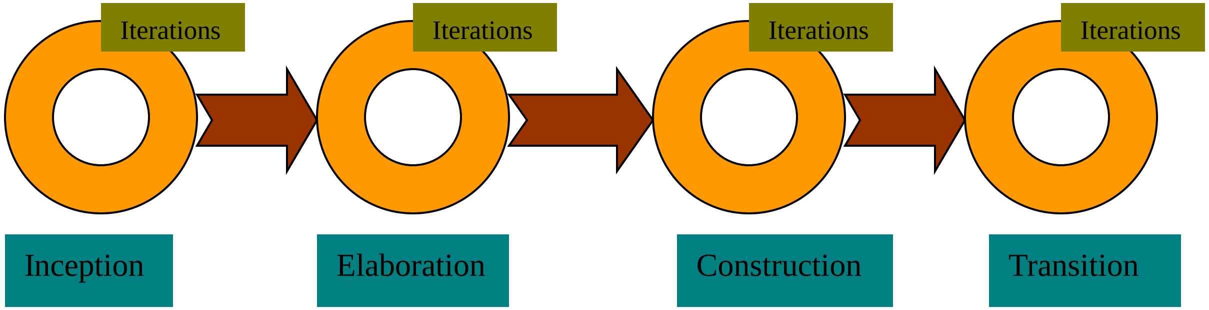
- RUP is a complete software-development process framework , developed by Rational Corporation.
- It’s an iterative development methodology based upon six industry-proven best practices.
- Processes derived from RUP vary from lightweight—addressing the needs of small projects —to more comprehensive processes addressing the needs of large, possibly distributed project teams.

Phases in RUP

- RUP is divided into four phases, named:
- Inception
- Elaboration
- Construction
- Transition

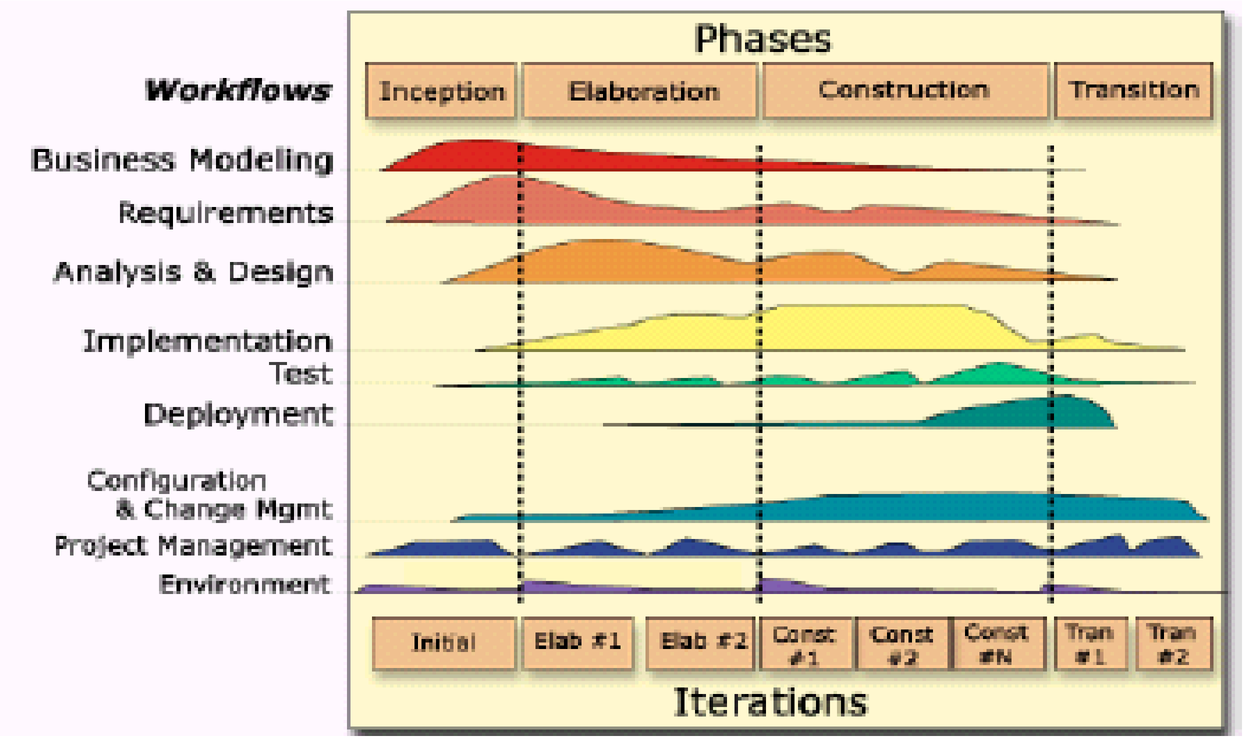
Iterations

EEach phase has iterations, each having the purpose of producing a demonstrable piece of software. The duration of iteration may vary from two weeks or less up to six months.



The iterations and the phases fig 1

Resource Histogram



RUP Overview Diagram

The iterations and the phases fig 2

Unified Process best practices

- Get high risk and high value first
- Constant user feedback and engagement
- Early cohesive core architecture
- Test early, often, and realistically
- Apply use cases where needed
- Do some visual modeling with UML
- Manage requirements
- Manage change requests and configuration

Inception

- The life-cycle objectives of the project are stated, so that the needs of every stakeholder are considered. Scope and boundary conditions, acceptance criteria and some requirements are established.

Inception - Activities

- **Formulate the scope of the project.**
- Needs of every stakeholder, scope, boundary conditions and acceptance criteria established.
- **Plan and prepare the business case.**
- Define risk mitigation strategy, develop an initial project plan and identify known cost, schedule, and profitability trade-offs.
- **Synthesize candidate architecture.**
- Candidate architecture is picked from various potential architectures
- **Prepare the project environment.**

Inception - Exit criteria

- An initial business case containing at least a clear formulation of the product vision - the core requirements - in terms of functionality, scope, performance, capacity, technology base.
- Success criteria (example: revenue projection).
- An initial risk assessment.
- An estimate of the resources required to complete the elaboration phase.

Elaboration

- An analysis is done to determine the risks, stability of vision of what the product is to become, stability of architecture and expenditure of resources.

Elaboration - Entry criteria

- The products and artifacts described in the exit criteria of the previous phase.
- The plan approved by the project management, and funding authority, and the resources required for the elaboration phase have been allocated.

Elaboration - Activities

- **Define the architecture.**
- Project plan is defined. The process, infrastructure and development environment are described.
- **Validate the architecture.**
- **Baseline the architecture.**
- To provide a stable basis for the bulk of the design and implementation effort in the construction phase.

Elaboration - Exit criteria

- A detailed software development plan, with an updated risk assessment, a management plan, a staffing plan, a phase plan showing the number and contents of the iteration , an iteration plan, and a test plan
- The development environment and other tools
- A baseline vision, in the form of a set of evaluation criteria for the final product
- A domain analysis model, sufficient to be able to call the corresponding architecture ‘complete’.
- An executable architecture baseline.

Construction

- The Construction phase is a manufacturing process. It emphasizes managing resources and controlling operations to optimize costs, schedules and quality. This phase is broken into several iterations.

Construction - Entry criteria

- The product and artifacts of the previous iteration. The iteration plan must state the iteration specific goals
- Risks being mitigated during this iteration.
- Defects being fixed during the iteration.

Construction - Activities

- **Develop and test components.**
- Components required satisfying the use cases, scenarios, and other functionality for the iteration are built. Unit and integration tests are done on Components.
- **Manage resources and control process.**
- **Assess the iteration**
- Satisfaction of the goal of iteration is determined.

Construction - Exit Criteria

- The same products and artifacts, updated, plus:
- A release description document, which captures the results of an iteration
- Test cases and results of the tests conducted on the products,
- An iteration plan, detailing the next iteration
- Objective measurable evaluation criteria for assessing the results of the next iteration(s).

Transition

- The transition phase is the phase where the product is put in the hands of its end users. It involves issues of marketing, packaging, installing, configuring, supporting the user-community, making corrections, etc.

Transition - Entry criteria

- The product and artifacts of the previous iteration, and in particular a software product sufficiently mature to be put into the hands of its users.

Transition - Activities

- Test the product deliverable in a customer environment.
- Fine tune the product based upon customer feedback
- Deliver the final product to the end user
- Finalize end-user support material

Transition - Exit criteria

- An update of some of the previous documents, as necessary, the plan being replaced by a "post-mortem" analysis of the performance of the project relative to its original and revised success criteria;
- A brief inventory of the organization's new assets as a result this cycle.

Advantages of RUP

- The RUP puts an emphasis on addressing very early high risks areas.
- It does not assume a fixed set of firm requirements at the inception of the project, but allows to refine the requirements as the project evolves.
- It does not put either a strong focus on documents
- The main focus remains the software product itself, and its quality.

Drawbacks of RUP

- RUP is not considered particularly “agile” However, recent studies have shown that by adopting the right essential artifacts RUP is agile.
- It fails to provide any clear implementation guidelines.
- RUP leaves the tailoring to the user entirely.

THANK YOU