UNIT-1

Analysis and Design

Analysis emphasizes an *investigation* of the problem and requirements, rather than a solution. For example, if a new computerized library information system is desired, how will it be used?

"Analysis" is a broad term, best qualified, as in *requirements analysis* (an inves-tigation of the requirements) or *object analysis* (an investigation of the domain objects).

Design emphasizes a *conceptual solution* that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Ultimately, designs can be implemented.

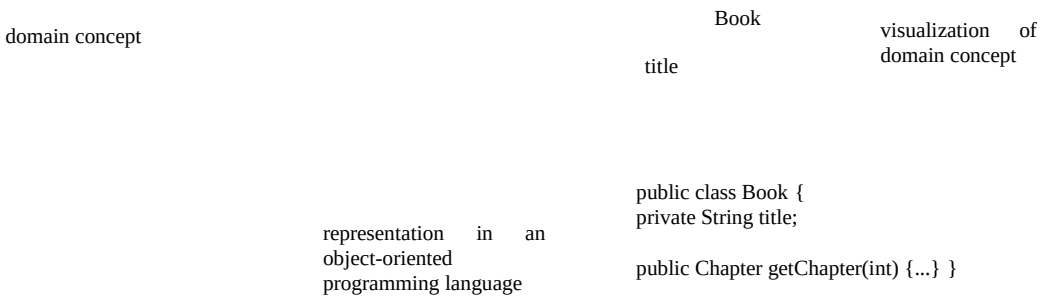
Analysis and design have been summarized in the phase *do the right thing (analysis), and do the thing right (design).*

Object-Oriented Analysis and Design

During **object-oriented analysis,** there is an emphasis on finding and describ-ing the objects—or concepts—in the problem domain. For example, in the case of the library information system, some of the concepts include *Book*, *Library*, and *Patron.*

During **object-oriented design,** there is an emphasis on defining software objects and how they collaborate to fulfill the requirements. For example, in the library system, a *Book* software object may have a *title* attribute and a *getChap-ter* method

Finally, during implementation or object-oriented programming, design objects are implemented, such as a *Book* class in Java.



TYPICAL ACTIVITIES / WORKFLOWS / DISCIPLINES IN OOAD

The **Unified Process** has emerged as a popular software development process for building object-oriented systems.

UP Phases

A UP project organizes the work and iterations across four major phases:

1. **Inception**— approximate vision, business case, scope, vague estimates.
2. **Elaboration**—refined vision, iterative implementation of the core architec ture, resolution of high risks, identification of most requirements and scope, more realistic estimates.
3. **Construction**—iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.
4. **Transition—**beta tests, deployment**.**

Inception is not a requirements phase; rather, it is a kind of feasibility phase, where just enough investigation is done to support a decision to continue or stop.

Elaboration is not the requirements or design phase; rather, it is a phase where the core architecture is iteratively implemented, and high risk issues are mitigated.

Schedule-oriented terms in the UP

development cycle

iteration *phase*

inc. *ela borat ion* *con struc tion* *trans ition*

milestone	*release*	*increment*	*final production release*
An iteration end-point when some significant decision or evaluation occurs.	A stable executable subset of the final product. The end of each iteration is a minor release.	The difference (delta) between the releases of 2 subsequent iterations.	At this point, the system is released for production

UP Disciplines

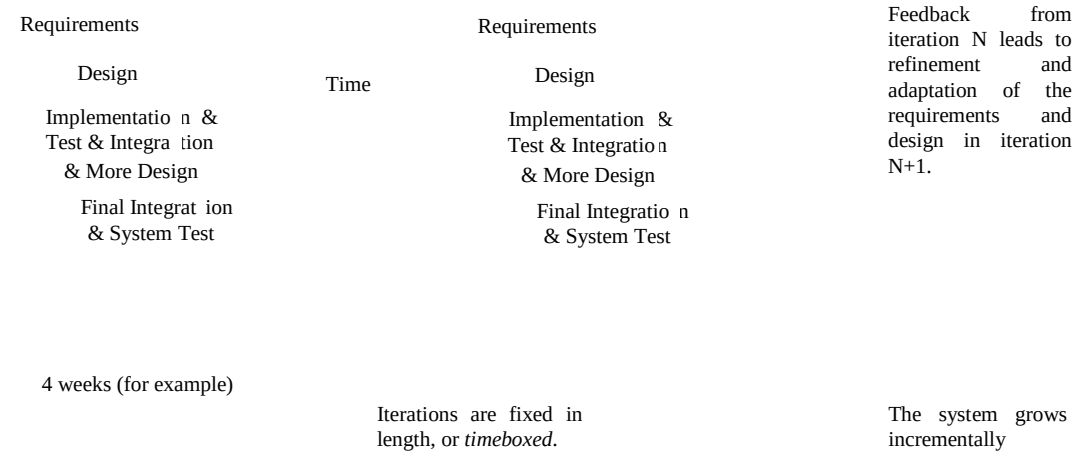
There are several disciplines in the UP

- **Business Modeling**: The Domain Model artifact,to visualize noteworthy concepts in the application domain.
- **Requirements**:The UseCase Model and Supplementary Specification Artifacts to capyure functional and non-functional requirements.
- **Design**: The Design Model artifact,to design the software objects.
- **Implementation**:Programming and building the system,not deploying it.
- **Environment**:Refers to establishing the tools and customizing the process for the project.

Iterative Development

Development is organized into a series of short, fixed-length (for example, four week) mini-projects called **iterations;** the outcome of each is a tested, integrated, and executable system. Each iteration includes its own requirements analysis, design, implementation, and testing activities.

Early iterative process ideas were known as spiral development and evolution-ary development [Boehm.88, Gilb88].



Benefits of Iterative Development

Benefits of iterative development include:

- early rather than late mitigation of high risks (technical, requirements, objectives, usability, and so forth)
- early visible progress
- early feedback, user engagement, and adaptation, leading to a refined sys tem that more closely meets the real needs of the stakeholders
- managed complexity; the team is not overwhelmed by "analysis paralysis" or very long and complex steps
- the learning within an iteration can be methodically used to improve the development process itself, iteration by iteration

The UML

The Unified Modeling Language (UML) is a language for speci-fying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems

Three Ways To Apply UML:

- **UML AS SKETCH:**Informal and incomplete Diagrams created to explore difficult parts of the problem or solution space,exploiting the power of visual languages.

- **UML AS BLUEPRINT:** Detailed Design Diagrams are used for Reverse Engineering to visualize and better understanding existing code in UML Diagrams or for Forward Engineering (Code Generation)

- **UML AS PROGRAMMING LANGUAGE:** Complete execution specification of a software system in UML.Executable Code will be automayically generated but is not normally seen or modified by developers;one works only in UML programming language.

THREE PERSPECTIVES TO APPLY UML:

- **CONCEPTUAL PERSPECTIVE**:The diagrams are interpreted as describing things in a situation of the real world or domain of interest.
- **SPECIFICATION(S/W)PERSPECTIVE**:The diagrams describe software abstractions or components with specifications and interfaces but no commitment to a particular implementation.
- **IMPLEMENTATION(S/W) PERSPECTIVE:** The diagrams describe software implementations in aparticular technology.

MAPPING DISCIPLINES TO UML ARTIFACTS

Sample Development Case of UP artifacts, s - start; r - refine

Discipline	Artifact Iteration-*	Incep. 11	Elab. El. .En	Const. CL.Cn	Trans. T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Implementation	Implementation Model		s	r	r
Project Management	SW Development Plan	s	r	r	r
Testing	Test Model		s	r	
Environment	Development Case	s	r		

INTRODUCTION TO DESIGN PATTERNS

DESIGN PATTERN: A template for how to solve a problem that can be used in many situations.

Design Patterns systematically names , explains and evaluate an important and recurrent design in OOdesign.

The goal of the Design Pattern is to capture the Design experience in a form that people can use effectively.

Goals Of A Good Design

Flexibility: Actions for Change

- Identify
- Change
- Test

Extensibility: The ability to add new functionality with ease.

Maintainability: The togetherness of flexibility , extensibility , fixing of bugs and refactorings.

MVC Architecture:

MVC consists of three kinds of objects :

M: Model is the application object.

V: View is the screen presentation.

C: Controller is the way the user interface reacts to user input.

- MVC decouples to increase flrxibility and reuse.
- MVC decouples views and models by establishing a subscribe or notify protocol between them.
- A view must ensure that its appearance must reflect state of the model.
- Whenever the model's data changes the model notifies views that depends on it.
- We can also create new views for a model without re-writing it.
- The model contains some data values and the views defining a spread sheet,histogram and the pie chart displays these data in various ways.
- The model communicates with its value changes and views communicate with the model to access these values.
- Feature of MVC is that views can be nested.

MVC Architecture:

·	a	b	C
x	60	30	10
y	50	30	20
z	80	10	10