Course Name: Object Oriented Software Engineering Course Code: CS107

BATCH: 2020

SECTION: 3L

SESSION: JAN-JUNE 2023

FACULTY: MS. RAVITA CHAHAR

EMAIL ID: ravita.chahar@chitkara.edu.in

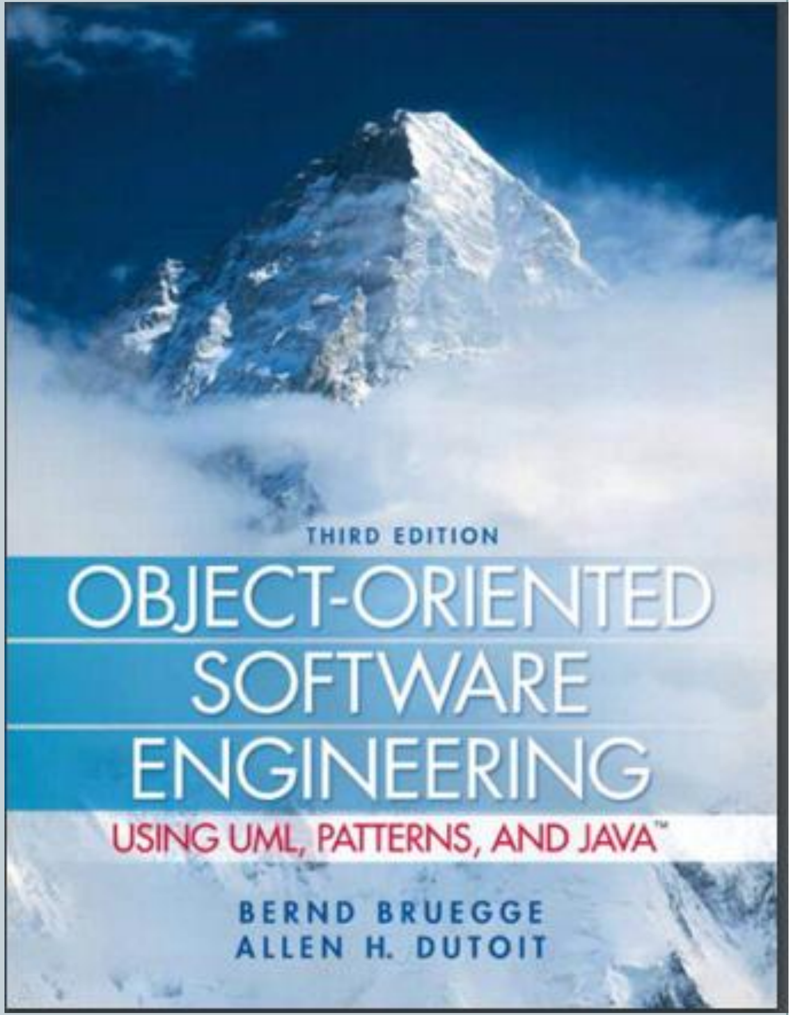
CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Object Oriented Software Engineering

LECTURE 1

Readings

- Required:
- **Bernd Bruegge, Allen Dutoit: "Object-Oriented Software Engineering: Using UML, Patterns, and Java", Prentice Hall, 2003.**


THIRD EDITION
OBJECT-ORIENTED SOFTWARE ENGINEERING
USING UML, PATTERNS, AND JAVA™
BERND BRUEGGE
ALLEN H. DUTOIT

Readings

- Recommended:
  - Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2nd ed., C. Larman
  - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: "Design Patterns", Addison-Wesley, 1996.
  - Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley, 1999.
  - K. Popper, "Objective Knowledge, an Evolutionary Approach, Oxford Press, 1979.
- Additional books may be recommended during individuals lectures

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Why Software Engineering?

- There is a growing need for talented software developers across every industry. As technology advances, the ability to build quality software while considering design, development, security, and maintenance is sought after amongst all kinds of companies, from finance and banking to healthcare and national security.
- Software Engineering applies the knowledge and theoretical understanding gained through computer science to building high-quality software products. As a maturing discipline, software is becoming more and more important in our everyday lives. Our software development and engineering professional program is Pace University's response to the tremendous growth of the software development industry.

Software Engineering:
A Problem Solving Activity

- **Analysis**: Understand the nature of the problem and break the problem into pieces
- **Synthesis**: Put the pieces together into a large structure

For problem solving we use

- Techniques (methods):
  - Formal procedures for producing results using some well-defined notation
- Methodologies:
  - Collection of techniques applied across software development and unified by a philosophical approach
- Tools:
  - Instrument or automated systems to accomplish a technique

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Software Engineering: Definition

Software Engineering is a collection of techniques, methodologies and tools that help
with the production of

- a high quality software system
- with a given budget
- before a given deadline

while change occurs.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Scientist vs Engineer

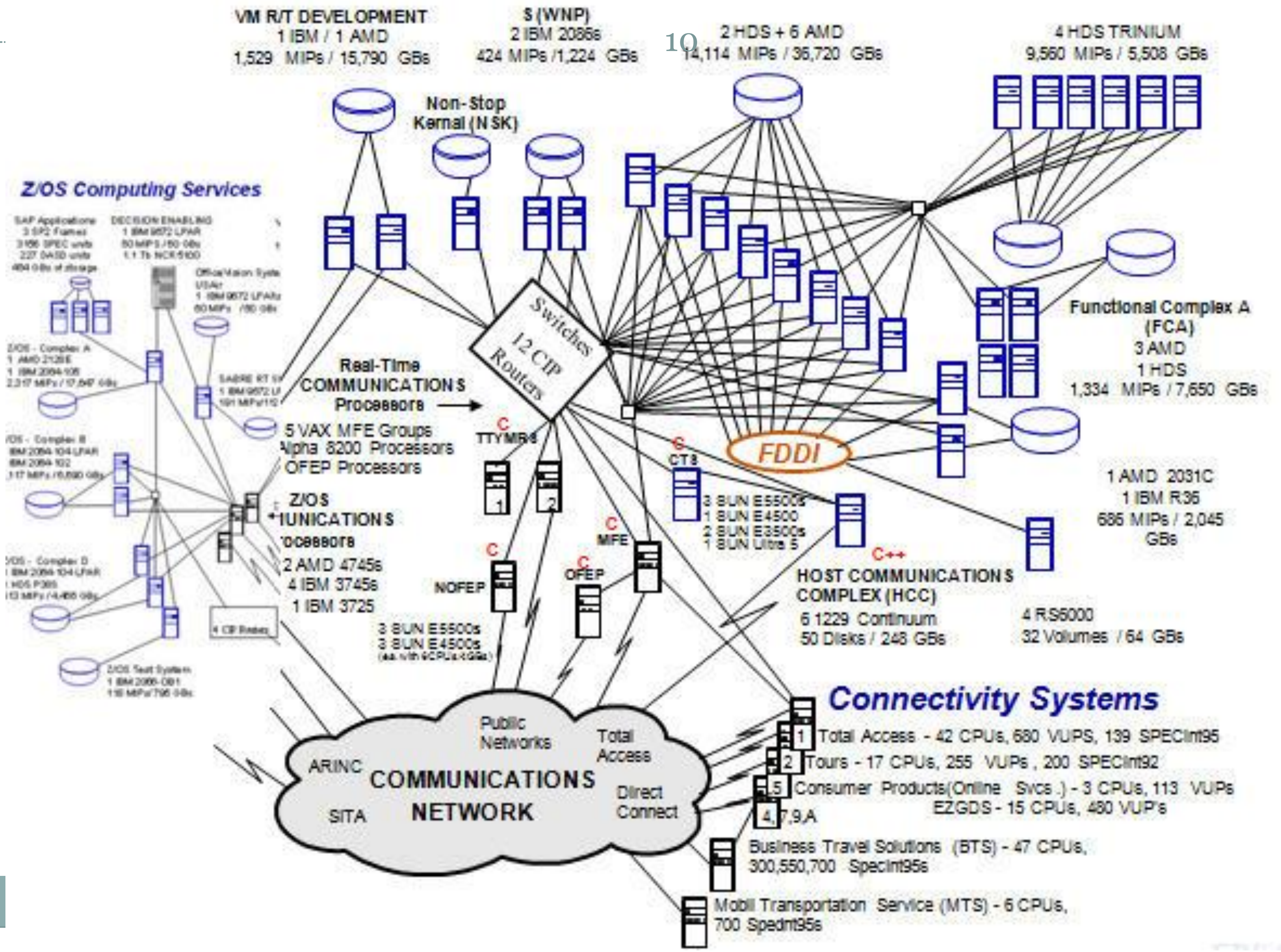
- Computer Scientist
  - Proves theorems about algorithms, designs languages, defines knowledge representation schemes
  - Has infinite time…
- Engineer
  - Develops a solution for an application-specific problem for a client
  - Uses computers & languages, tools, techniques and methods
- Software Engineer
  - Works in multiple application domains
  - Has only 3 months…
  - …while changes occurs in requirements and available technology

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Factors affecting the quality of a software system

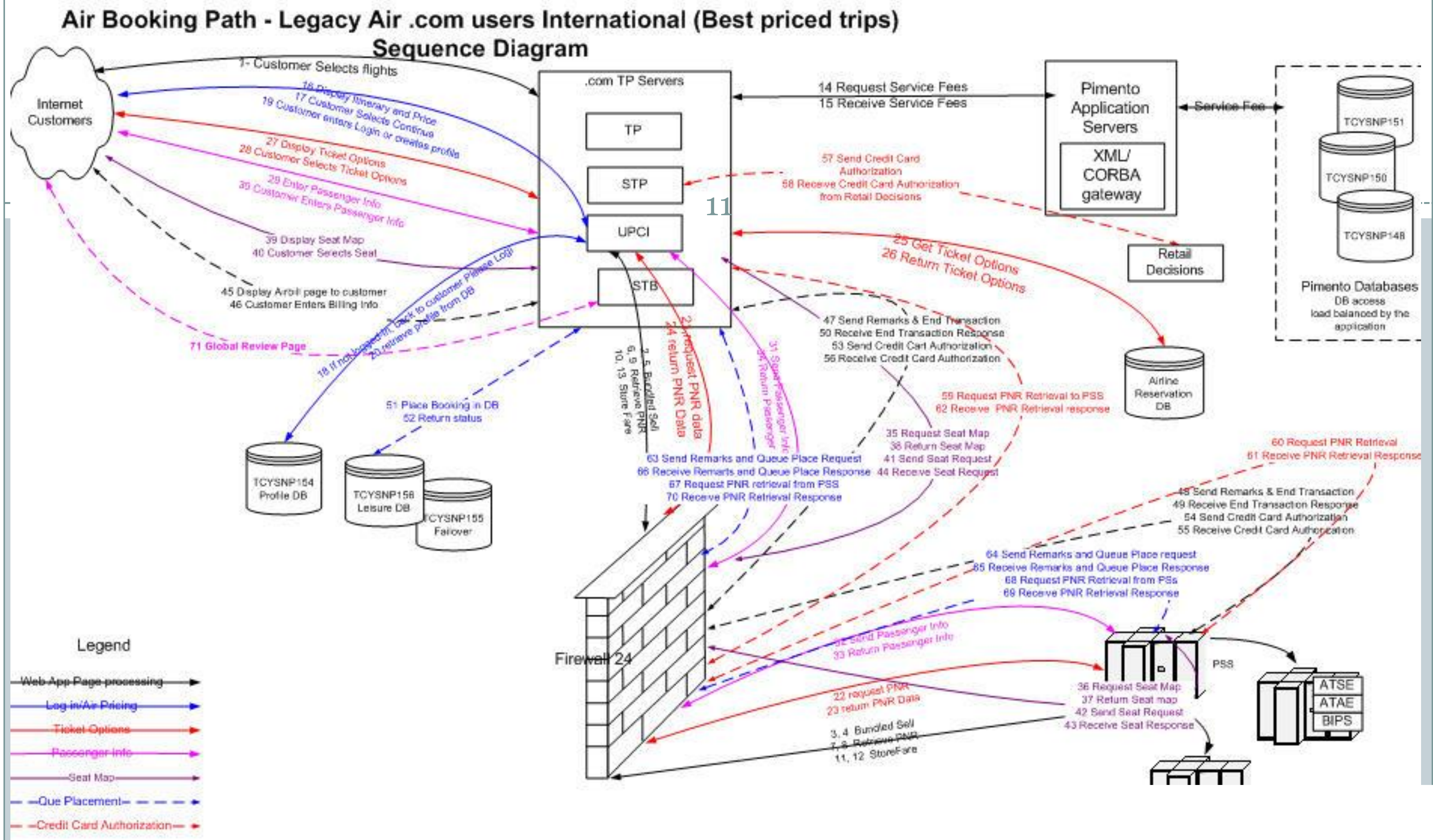
- **Complexity:**
  - The system is so complex that no single programmer can understand it anymore
  - The introduction of one bug fix causes another bug
- **Change:**
  - The "Entropy" of a software system increases with each change: Each implemented change erodes the structure of the system which makes the next change even more expensive ("Second Law of Software Dynamics").
  - As time goes on, the cost to implement a change will be too high, and the system will then be unable to support its intended task. This is true of all systems, independent of their application domain or technological base.

Complex Server Connections



CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Complex Message Flow


Air Booking Path - Legacy Air .com users International (Best priced trips)
Sequence Diagram
Internet Customers
1- Customer Selects flights
16 Display Itinerary and Price
17 Customer Selects Continue
19 Customer enters Login or creates profile
27 Display Ticket Options
28 Customer Selects Ticket Options
29 Enter Passenger Info
35 Customer Enters Passenger Info
39 Display Seat Map
40 Customer Selects Seat
45 Display Airbill page to customer
46 Customer Enters Billing Info
71 Global Review Page
18 If not logged in, back to customer Please Login
20 retrieve profile from DB
.com TP Servers
TP
STP
UPCI
STB
14 Request Service Fees
15 Receive Service Fees
Pimento Application Servers
XML/ CORBA gateway
Service Fee
TCYSNP151
TCYSNP150
TCYSNP148
Pimento Databases
DB access load balanced by the application
57 Send Credit Card Authorization
58 Receive Credit Card Authorization from Retail Decisions
Retail Decisions
25 Get Ticket Options
26 Return Ticket Options
Airline Reservation DB
47 Send Remarks & End Transaction
50 Receive End Transaction Response
53 Send Credit Cart Authorization
56 Receive Credit Card Authorization
59 Request PNR Retrieval to PSS
62 Receive PNR Retrieval response
35 Request Seat Map
38 Return Seat Map
41 Send Seat Request
44 Receive Seat Request
60 Request PNR Retrieval
61 Receive PNR Retrieval Response
48 Send Remarks & End Transaction
49 Receive End Transaction Response
54 Send Credit Card Authorization
55 Receive Credit Card Authorization
64 Send Remarks and Queue Place request
65 Receive Remarks and Queue Place Response
68 Request PNR Retrieval from PSs
69 Receive PNR Retrieval Response
2, 5 Bundled Sell
6, 9 Retrieve PNR
10, 13 Store Fare
21 request PNR data
24 return PNR Data
31 Send Passenger Info
34 Return Passenger Info
63 Send Remarks and Queue Place Request
66 Receive Remarks and Queue Place Response
67 Request PNR retrieval from PSS
70 Receive PNR Retrieval Response
51 Place Booking in DB
52 Return status
TCYSNP154 Profile DB
TCYSNP156 Leisure DB
TCYSNP155 Failover
Firewall 24
32 Send Passenger Info
33 Return Passenger Info
22 request PNR
23 return PNR Data
3, 4 Bundled Sell
7, 8 Retrieve PNR
11, 12 StoreFare
36 Request Seat Map
37 Return Seat map
42 Send Seat Request
43 Receive Seat Response
PSS
ATSE
ATAE
BIPS
Legend
Web App Page processing
Log in/Air Pricing
Ticket Options
Passenger Info
Seat Map
Que Placement
Credit Card Authorization

So where are we right now?

- Three ways to deal with complexity:
  - Abstraction
  - Decomposition
  - Hierarchy
- Object-oriented decomposition is a good methodology
  - Unfortunately, depending on the purpose of the system, different objects can be found
- How can we do it right?
  - Many different possibilities
  - Our current approach: Start with a description of the functionality (Use case model), then proceed to the object model
  - This leads us to the software lifecycle

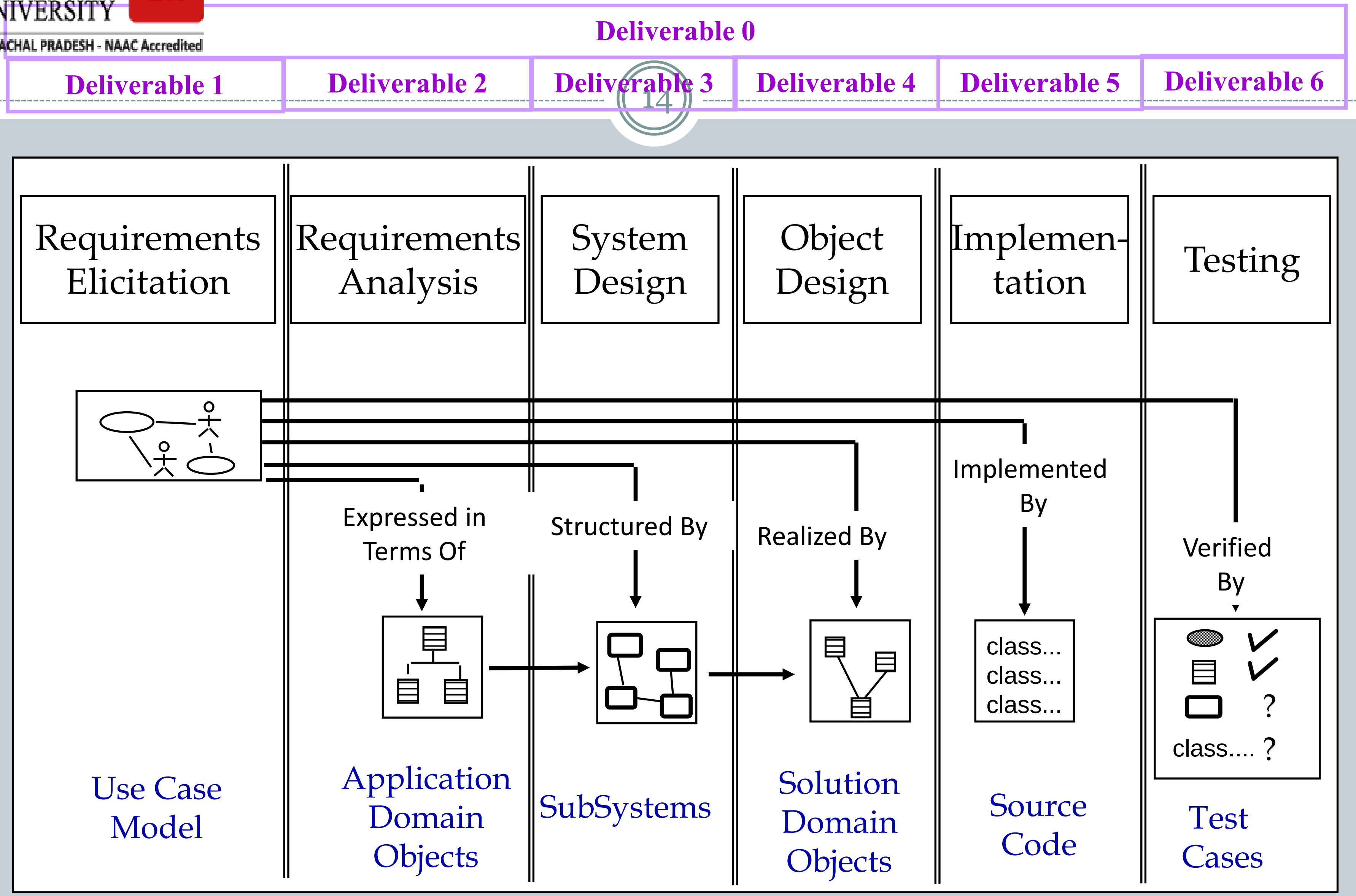
CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Software Lifecycle Definition

- **Software lifecycle:**
  - Set of activities and their relationships to each other to support the development of a software system

- **Typical Lifecycle questions:**
  - Which activities should I select for the software project?
  - What are the dependencies between activities?
  - How should I schedule the activities?

Software Lifecycle Activities



Each activity produces one or more models

Reusability: Design Patterns and Frameworks

- Design Pattern:
  - A small set of classes that provide a template solution to a recurring design problem
  - Reusable design knowledge on a higher level than data structures (link lists, binary trees, etc)
- Framework:
  - A moderately large set of classes that collaborate to carry out a set of responsibilities in an application domain.
    - Examples: User Interface Builder
- Provide architectural guidance during the design phase
- Provide a foundation for software components industry

Teams and Specifications

- Do we really need to write specifications?

- A typical software team will in general do the following:
  - Discuss what to do
  - Divide up the work
  - Implement incompatible components
  - Be surprised when it doesn’t all just work together

Summary

- Software engineering is a problem solving activity
  - Developing quality software for a complex problem within a limited time while things are changing
- There are many ways to deal with complexity
  - Modeling, decomposition, abstraction, hierarchy
  - Issue models: Show the negotiation aspects
  - System models: Show the technical aspects
  - Task models: Show the project management aspects
  - Use Patterns: Reduce complexity even further
- Many ways to deal with change
  - Tailor the software lifecycle to deal with changing project conditions
  - Use a nonlinear software lifecycle to deal with changing requirements or changing technology
  - Provide configuration management to deal with changing entities

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

THANK YOU