PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
REKAYASA PERANGKAT LUNAK LANJUT

Rekayasa Perangkat Lunak Lanjut

Analisis Berorientasi Objek

UNIFIED MODELING LANGUAGE™

Disusun Oleh:

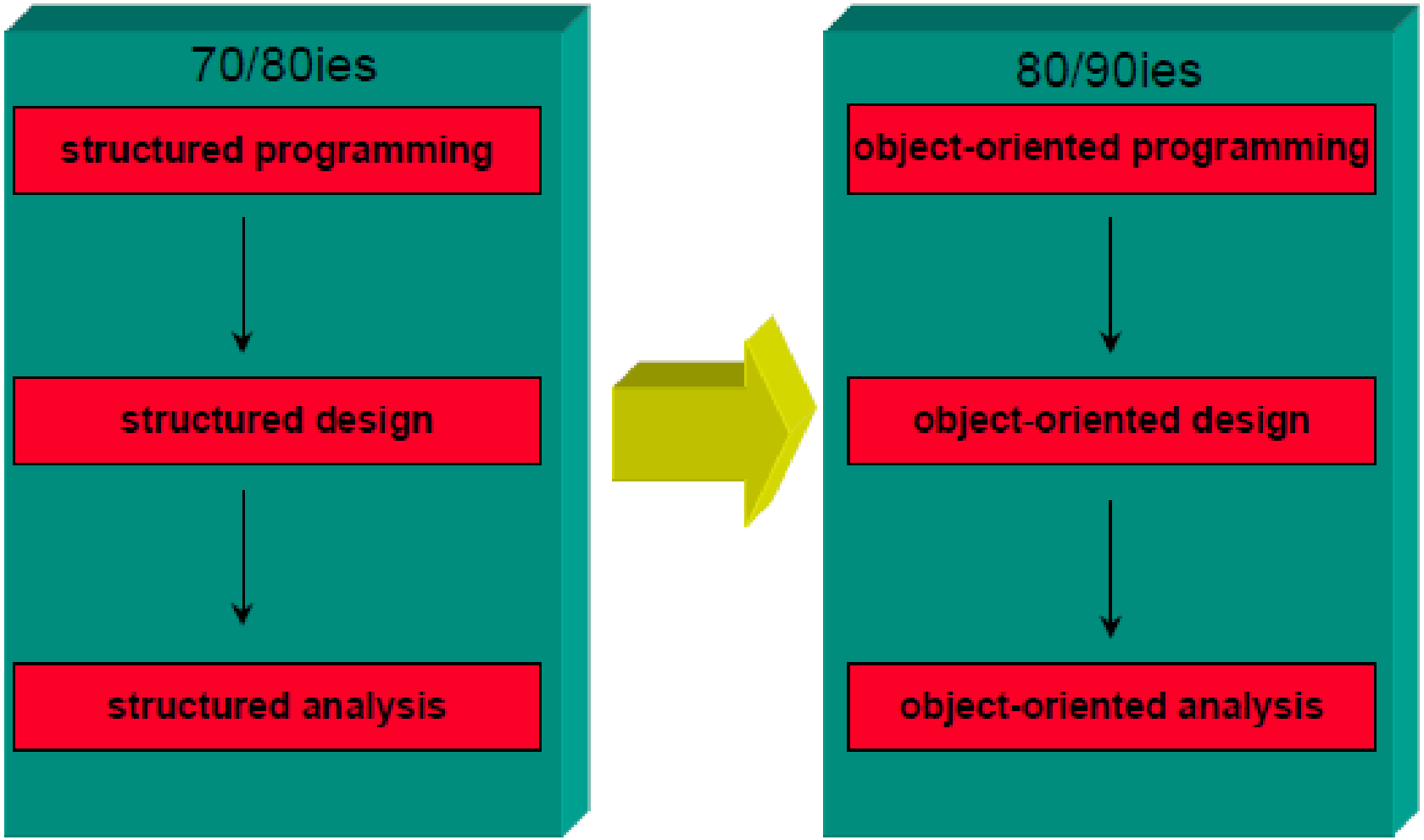
Tim RPLL

Silabus Mata Kuliah

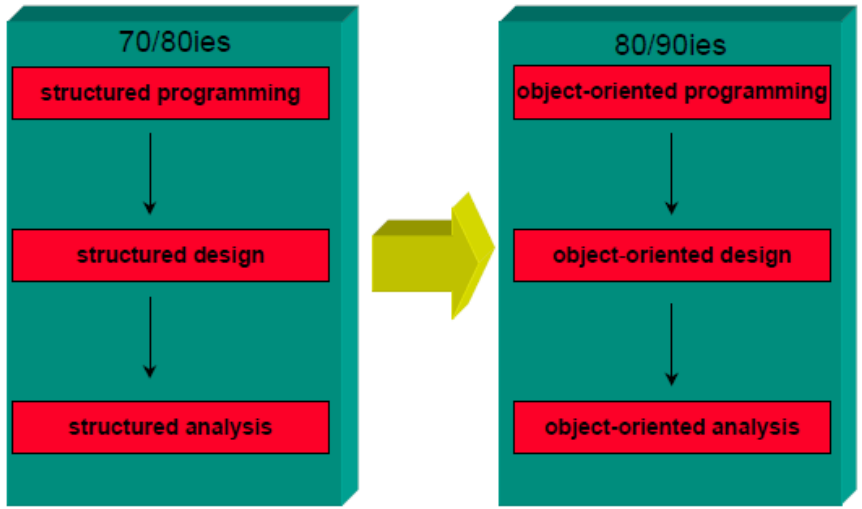
1. Pendahuluan
2. *Overview: Analisis Terstruktur*
3. Overview: Perancangan Terstruktur – Arsitektur, Interface, Data
4. **Analisis Berorientasi Objek**
5. Perancangan Berorientasi Objek
6. Pengenalan Web App. + Requirement Web App.
7. Konsep Web Engineering

Evolusi Metode Berorientasi Objek

Evolution of OO Development Methods



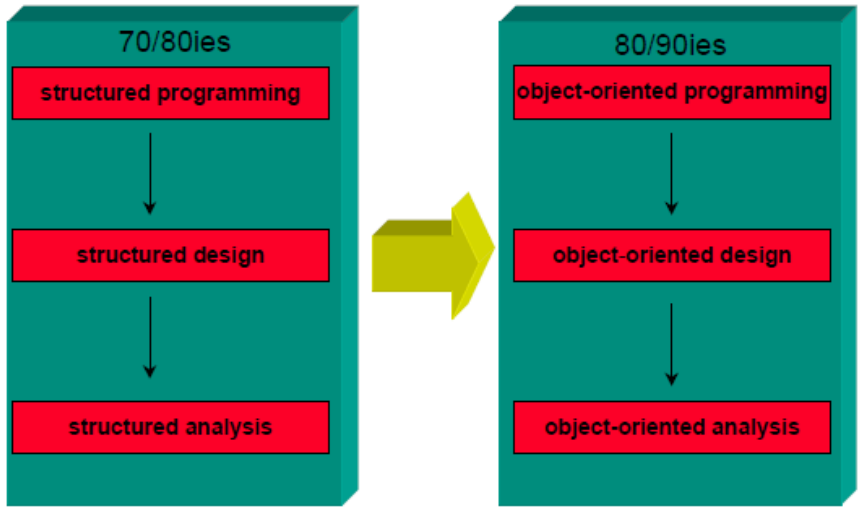
Evolution of OO Development Methods



- ✓ Metode beorientasi objek mulai berkembang ketika Grady Booch pada tahun 80-an mempublikasikan suatu paper bagaimana melakukan perancangan untuk bahasa ADA namun memberi judul paper tersebut dengan : Object-Oriented Design.

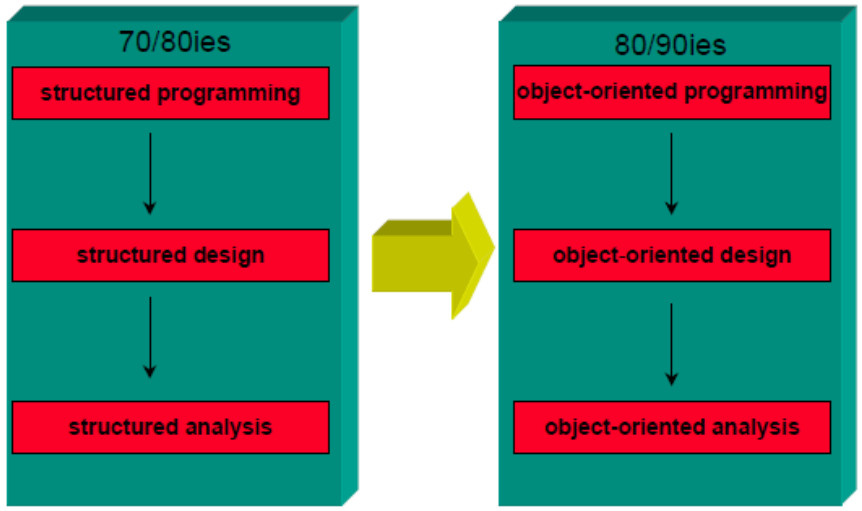
- ✓ Selanjutnya ide tersebut terus ia kembangkan sampai tahun 90 an.

Evolution of OO Development Methods



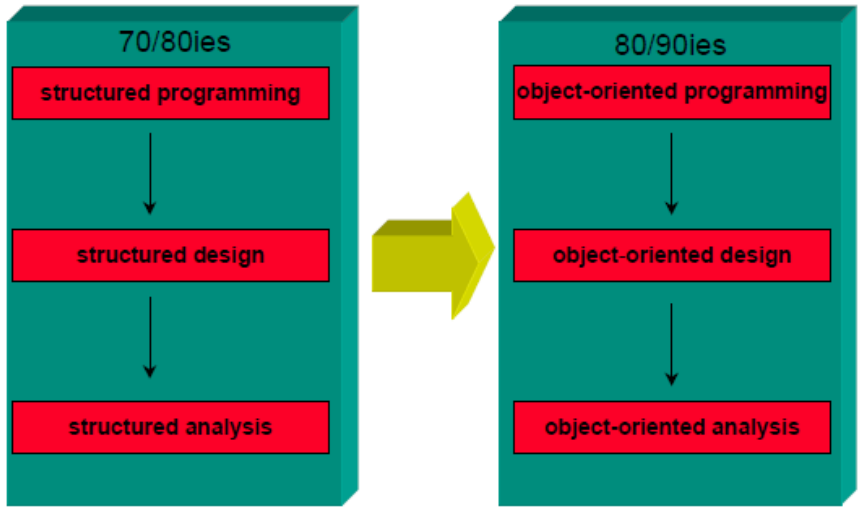
- ✓ Pada tahun 1991 Peter Coad dan Yourdon memperkenalkan metode berorientasi objek yang lebih sederhana dibandingkan Booch.
- ✓ Metode ini menjadi cepat populer karena mendukung layanan-layanan yang terdapat pada C++.
- ✓ Pada waktu itu C++ merupakan bahasa pemrograman berorientasi objek yang paling populer .

Evolution of OO Development Methods



- ✓ Pada tahun 1994 Ivar Jacobson memperkenalkan konsep use case dan object oriented software engineering.
- ✓ Pada tahun 1994 itu juga yaitu bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, mempelopori usaha untuk penyatuan notasi pendekatan berorientasi objek.
- ✓ Pada tahun 1995 dihasilkan draft pertama dari UML (versi 0.8).
- ✓ Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – http://www.omg.org).

Evolution of OO Development Methods



- ✓ Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003.
- ✓ Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999.
- ✓ Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek

Metode Berorientasi Objek

Analisis Berorientasi Objek

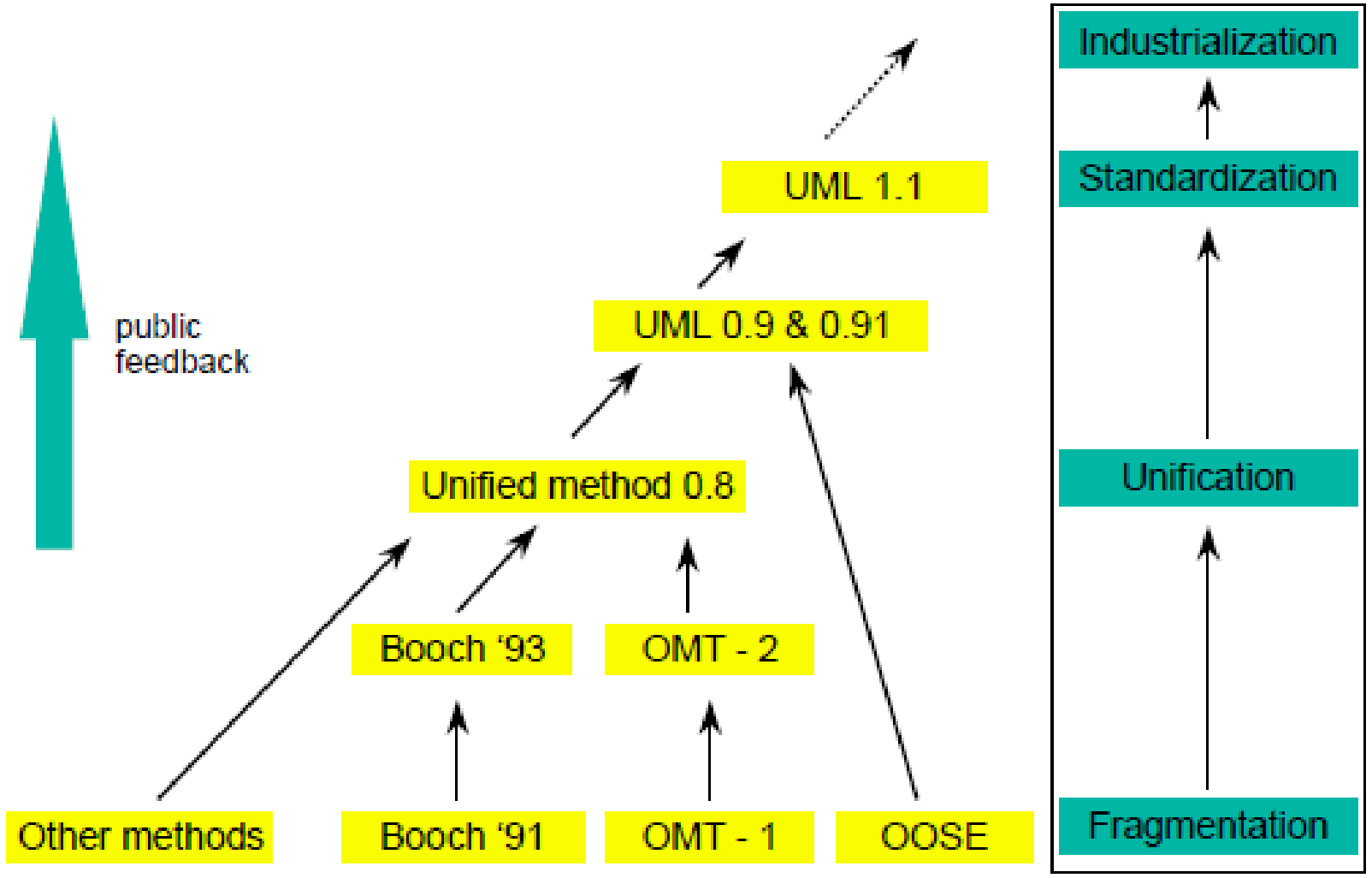
Analisis Berorientasi Objek

- Berfokus pada pendefinisian kelas-kelas dan cara bagaimana mereka saling bekerjasama satu dengan yang lainnya untuk memenuhi kebutuhan para pelanggan.
- Pada Paradigma Analysis Design dan Diagram, *Unified Modeling Language* (UML) merupakan perkakas (*tools*) yang digunakan untuk melakukan pemodelan berorientasi objek

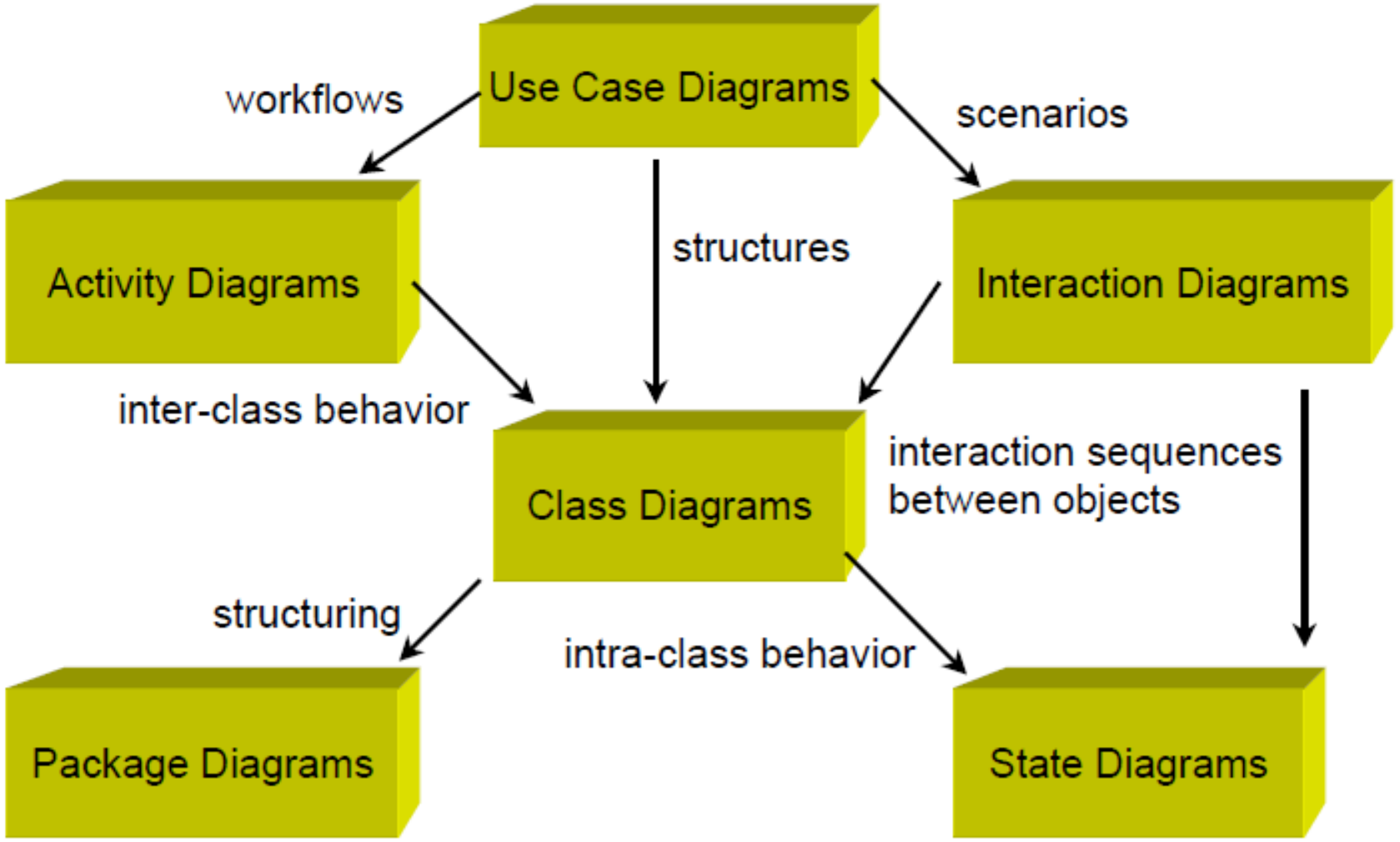
UML

- UML: Unified Modeling Language
- UML dapat digunakan untuk memodelkan semua proses dalam siklus hidup pengembangan dan seluruh teknologi implementasi yang berbeda
- UML adalah bahasa standar untuk memvisualisasikan,menspesifiksi, konstruksi, dan mendokumentasikan artifak dari sistem perangkat lunak
- UML adalah suatu alat komunikasi untuk team dan para *stakeholders*

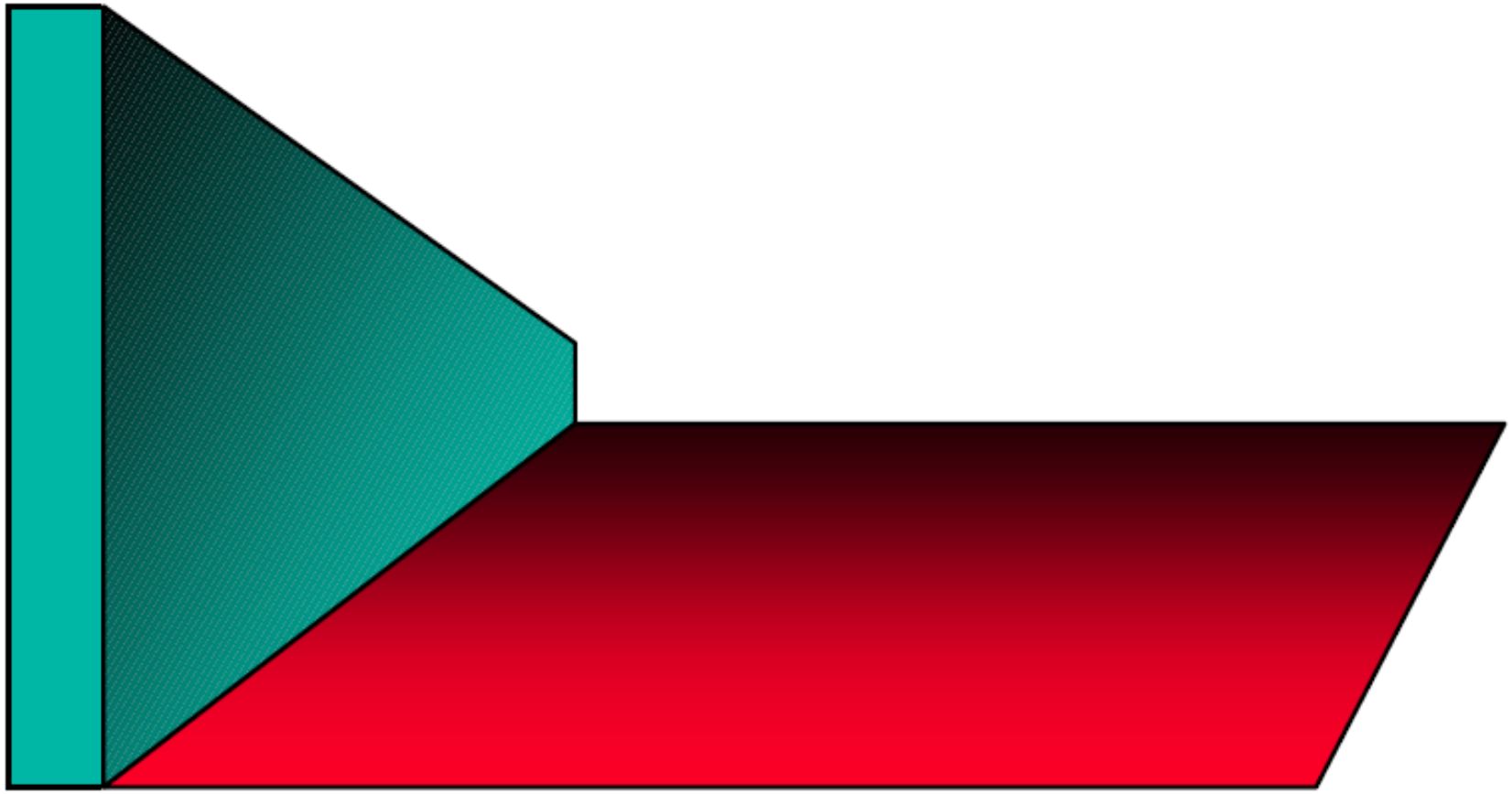
History of UML



UML Diagrams

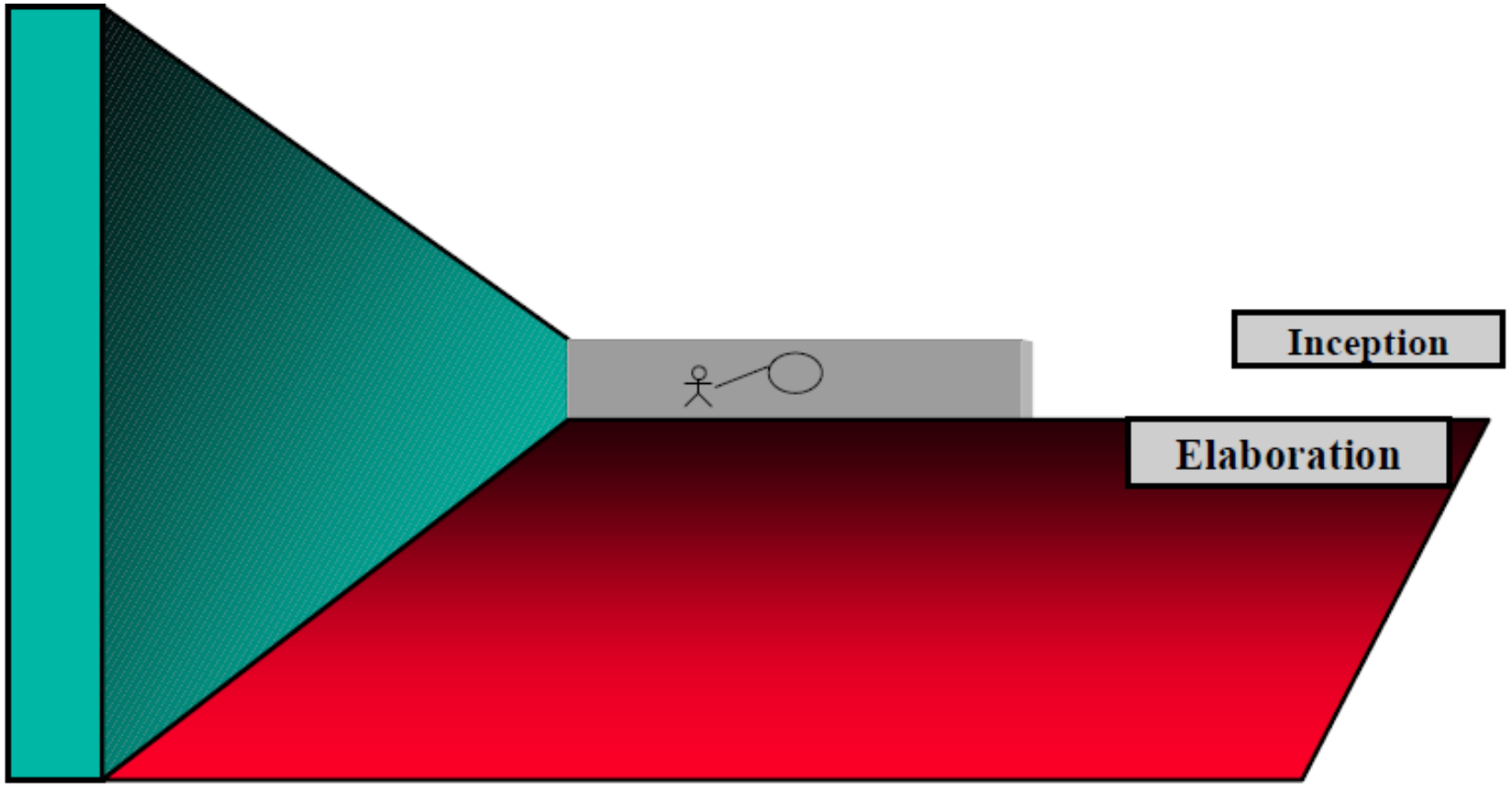


Diagrams and Process



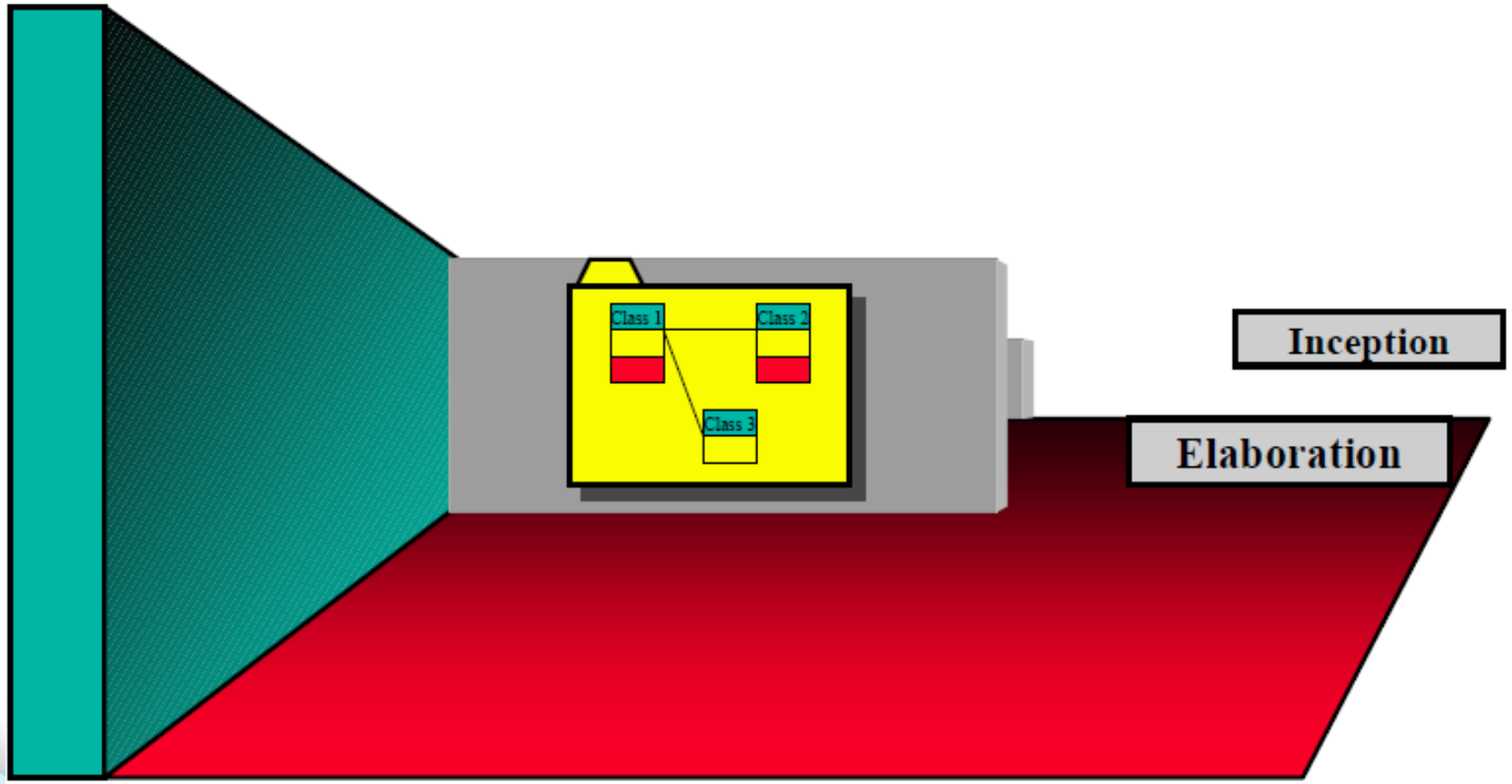
Diagrams and Process

Use Case Diagrams



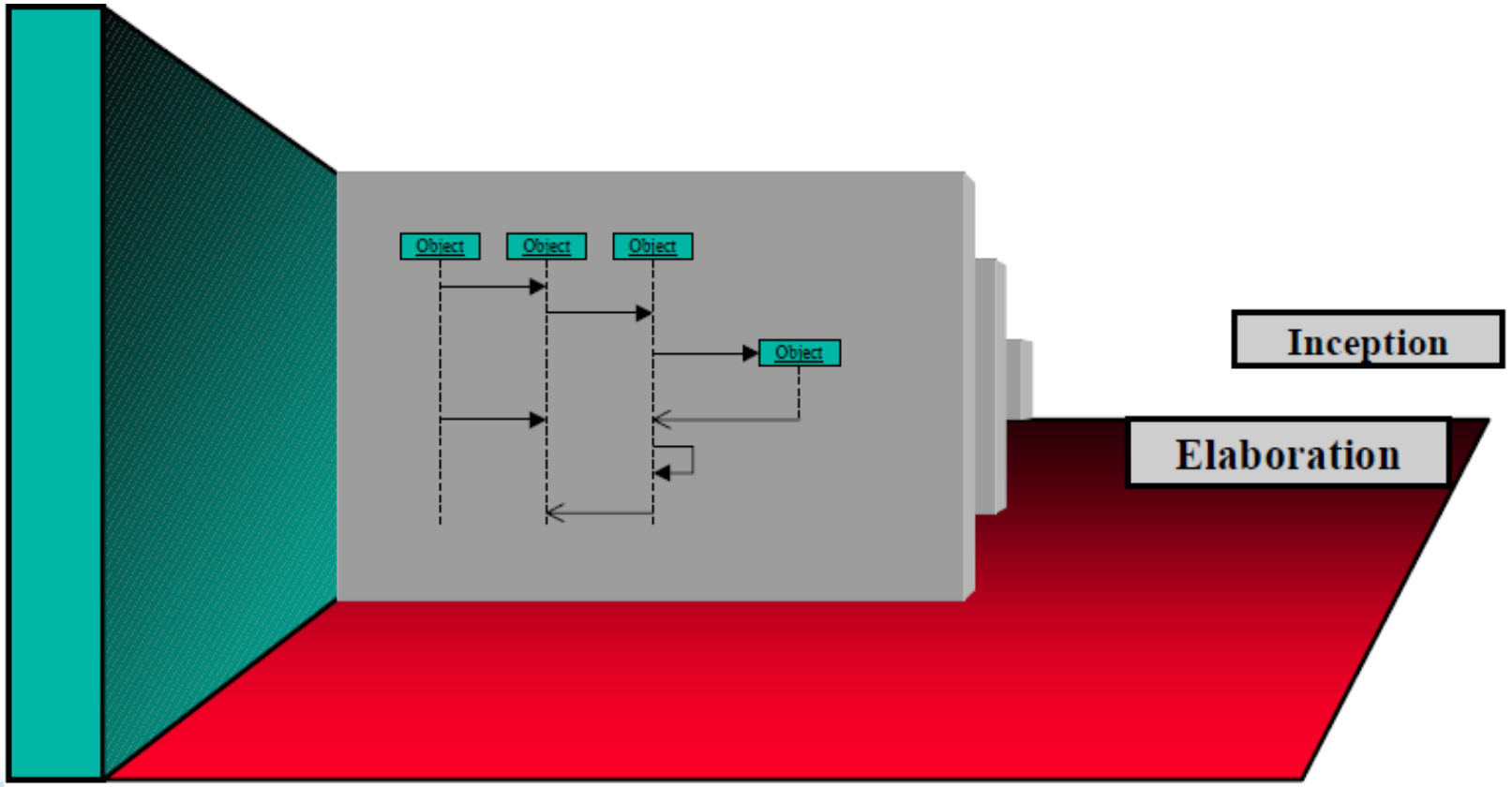
Diagrams and Process

Class & Package Diagrams



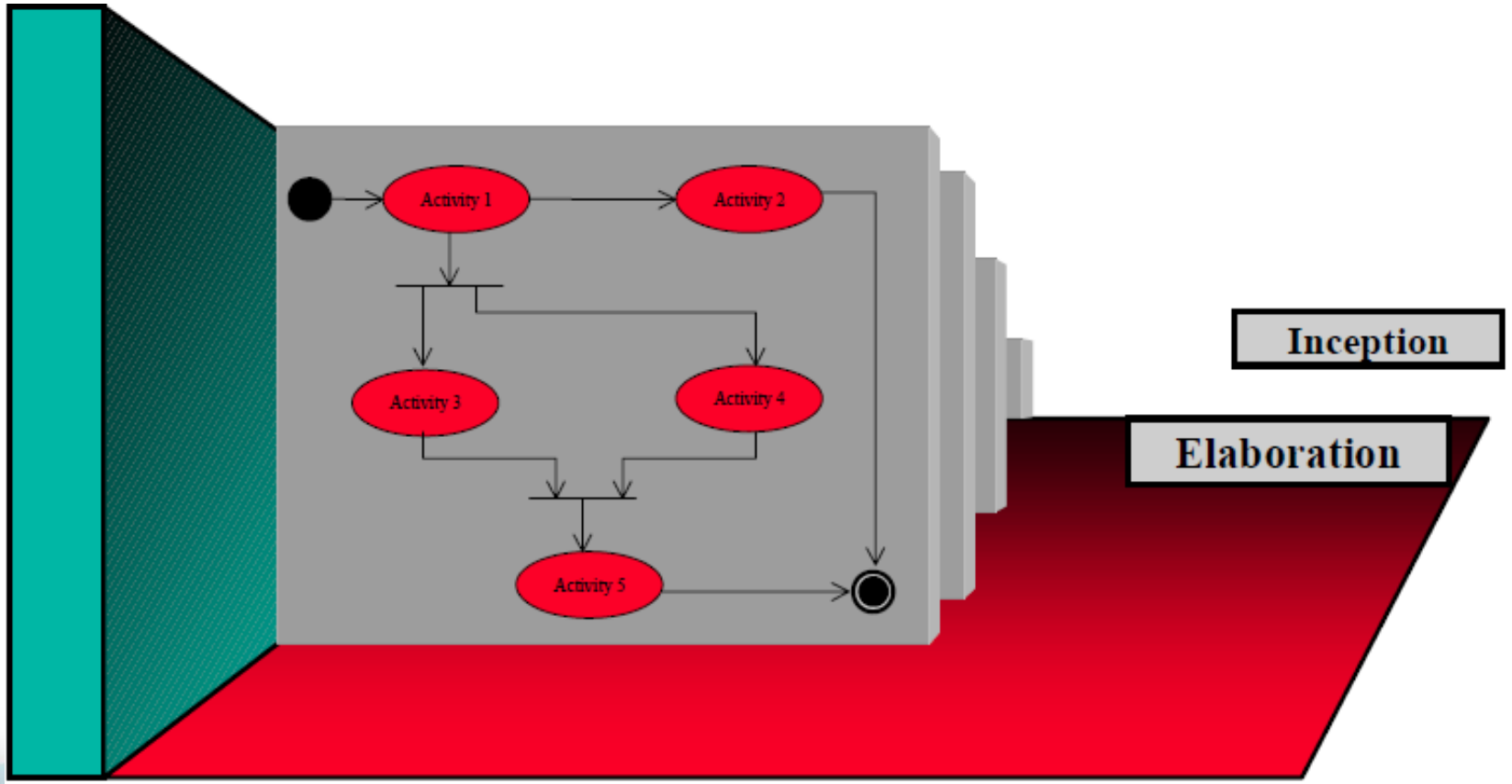
Diagrams and Process

Interaction Diagrams (Scenarios)



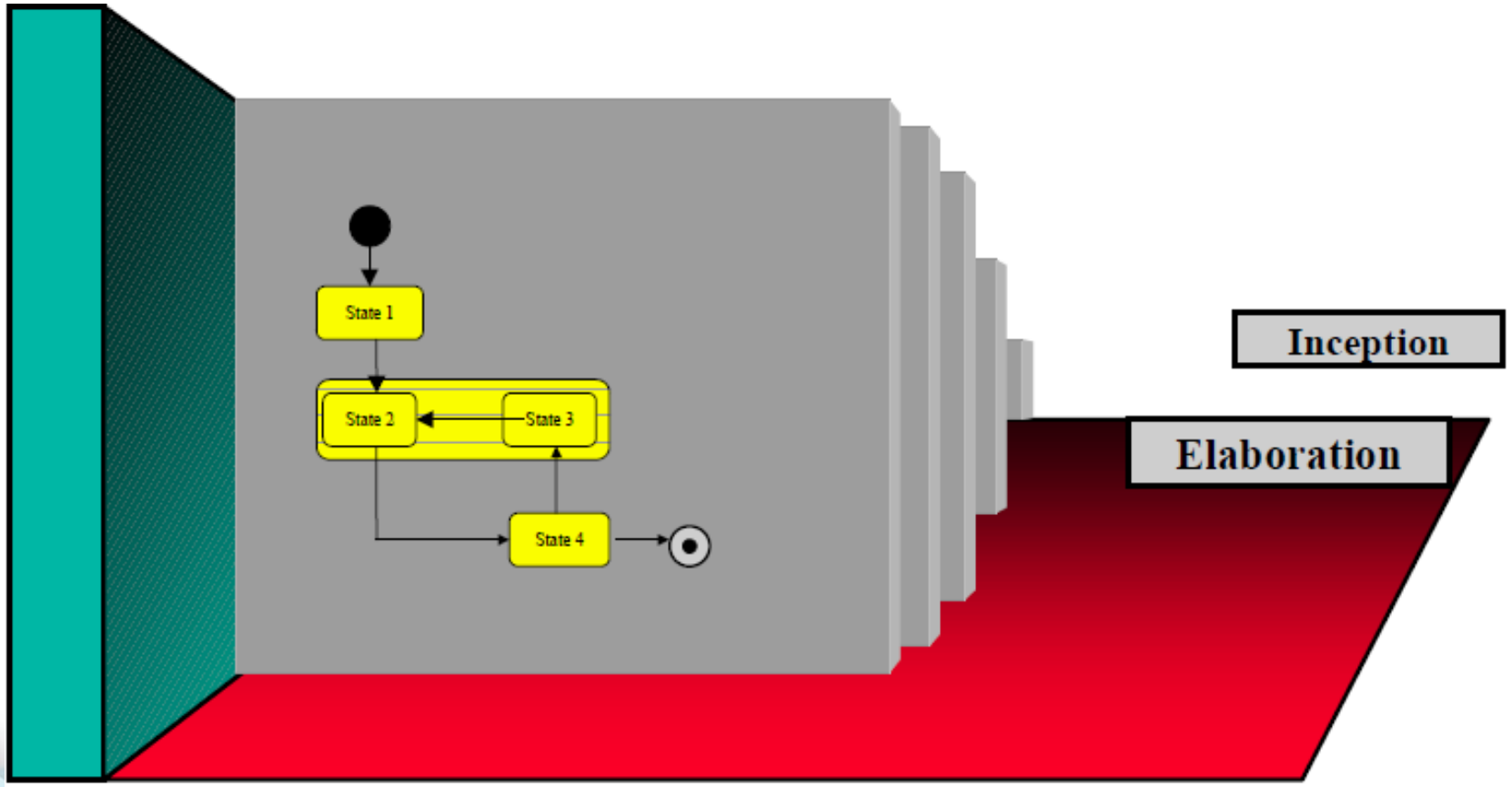
Diagrams and Process

Activity Diagrams (Workflow, Interclass Behavior)



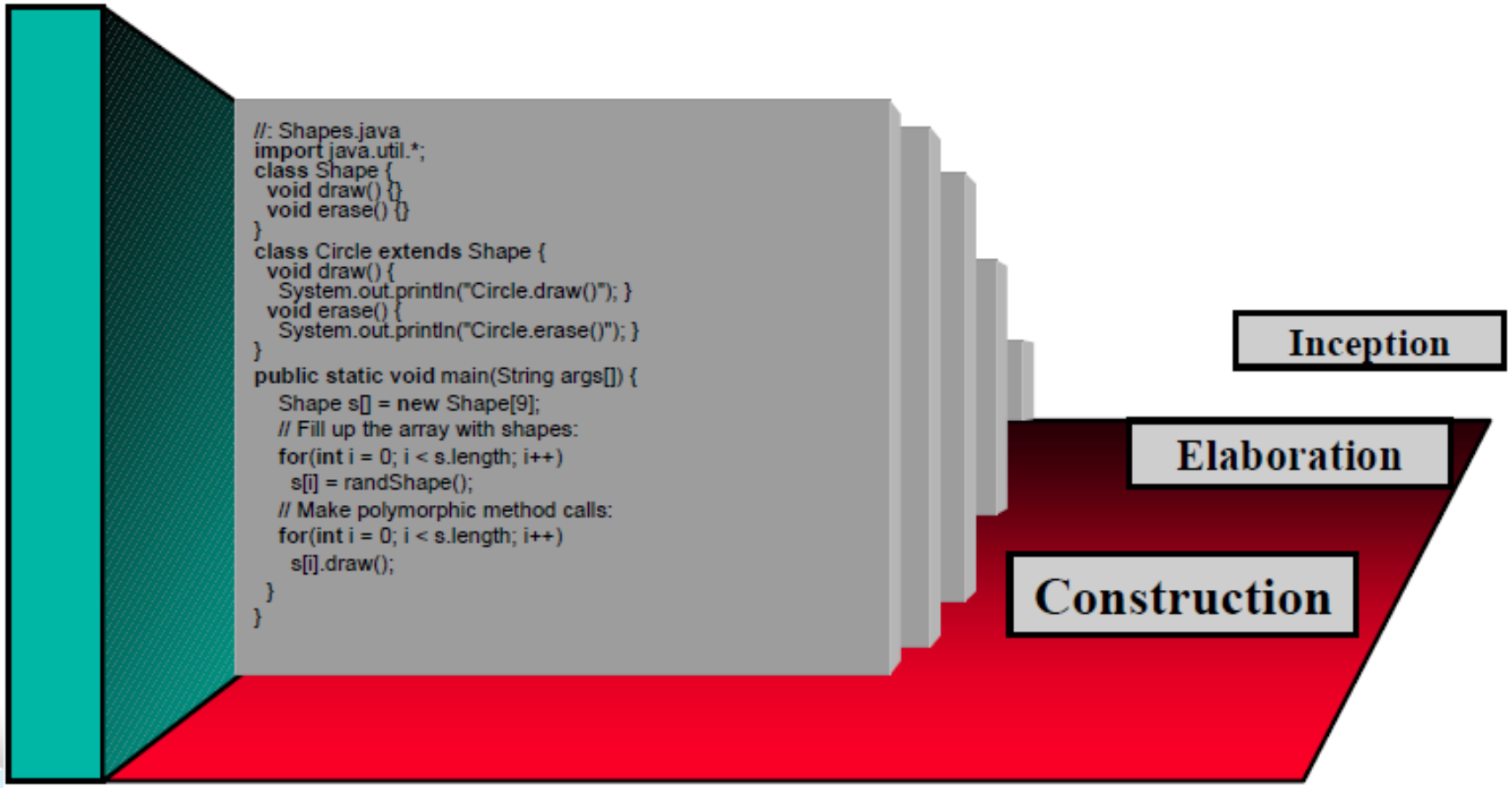
Diagrams and Process

State Transition Diagrams (Intraclass Behavior)



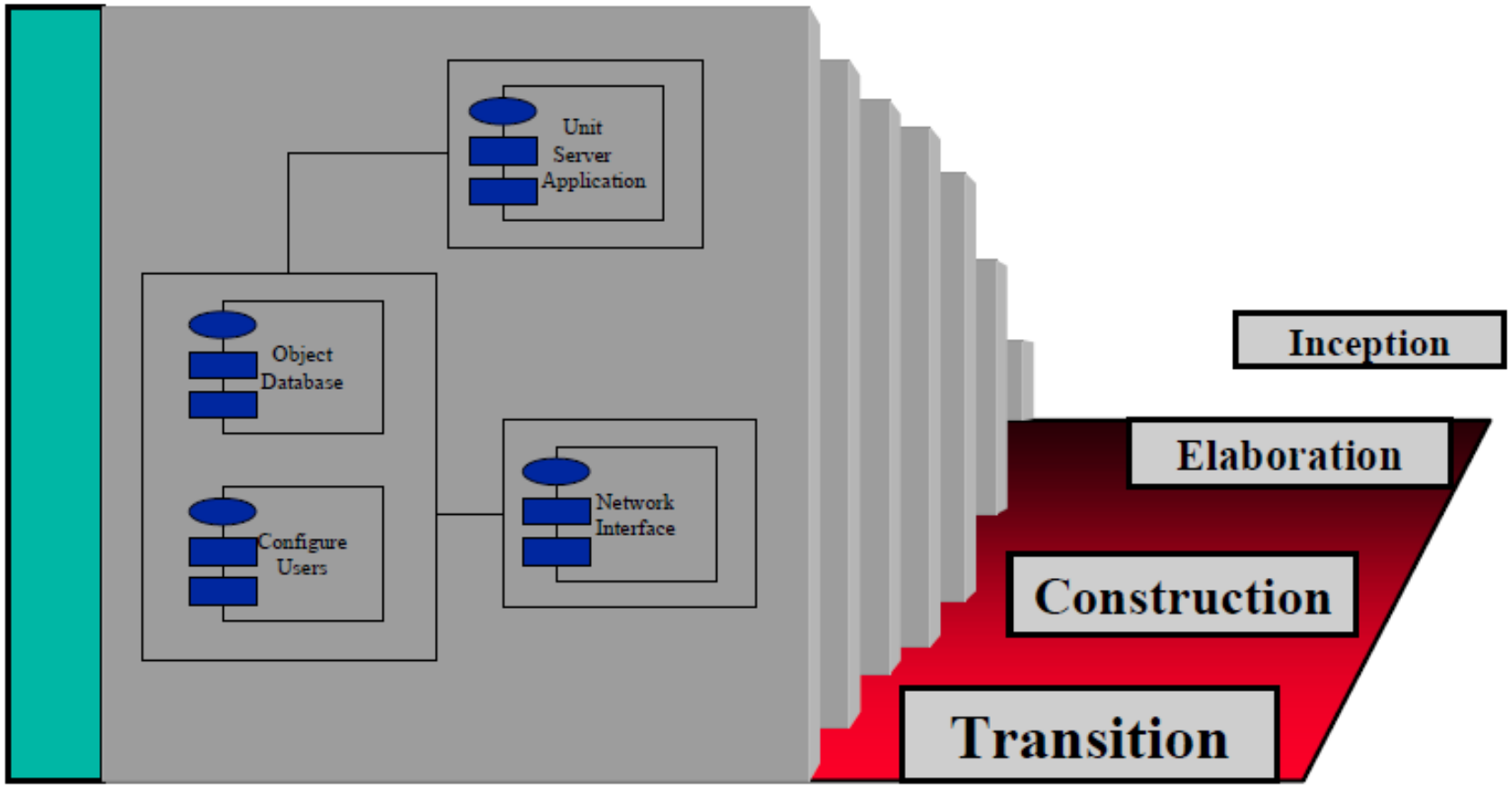
Texts and Process

Source Code



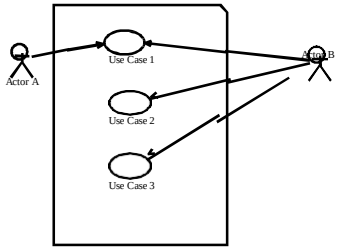
Diagrams and Process

Deployment Diagrams

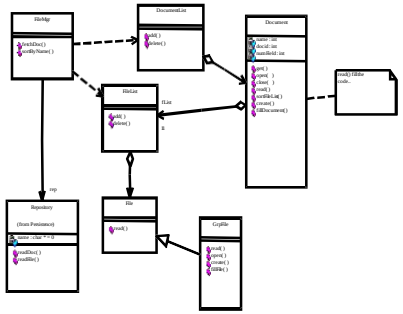


UML

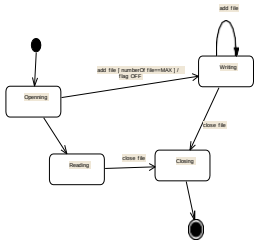
Use-Case Diagram



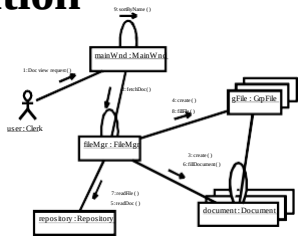
Class Diagram



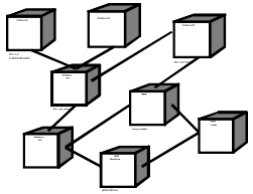
Statechart Diagram



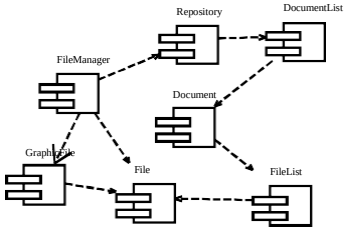
Collaboration Diagram



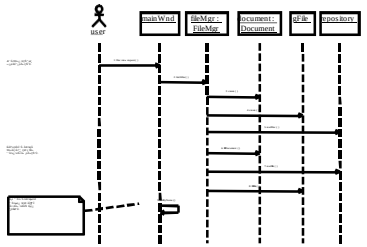
Deployment Diagram



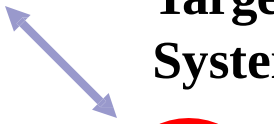
Component Diagram



Sequence Diagram



Forward and Reverse Engineering



Target System

UML 2.0

UML version 2.0 memiliki 14 diagram yang terbagi pada 2 kelompok besar:

1. Structure Diagrams
2. Behavior Diagrams

UML 2.0

Diagram Name	Used to	Primary Phase
Structure Diagrams		
Class	Illustrate the relationships between classes modeled in the system.	Analysis, Design
Object	Illustrate the relationships between objects modeled in the system. Used when actual instances of the classes will better communicate the model.	Analysis, Design
Package	Group other UML elements together to form higher level constructs.	Analysis, Design, Implementation
Deployment	Show the physical architecture of the system. Can also be used to show software components being deployed onto the physical architecture.	Physical Design, Implementation
Component	illustrate the physical relationships among the software components.	Physical Design, Implementation
Composite Structure	Illustrate the internal structure of a class, i.e., the relationships among the parts of a class.	Analysis, Design
Behavioral Diagrams		
Activity	Illustrate business workflows independent of classes, the flow of activities in a use case, or detailed design of a method.	Analysis, Design
Sequence	Model the behavior of objects within a use case. Focuses on the time-based ordering of an activity.	Analysis, Design
Communication	Model the behavior of objects within a use case. Focuses on the communication among a set of collaborating objects of an activity.	Analysis, Design
Interaction Overview	Illustrate an overview of the flow of control of a process.	Analysis, Design
Timing	Illustrate the interaction that takes place among a set of objects and the state changes in which they go through along a time axis.	Analysis, Design
Behavioral State Machine	Examine the behavior of one class.	Analysis, Design
Protocol State Machine	Illustrates the dependencies among the different interfaces of a class.	Analysis, Design
Use-Case	Capture business requirements for the system and to illustrate the interaction between the system and its environment.	Analysis

FIGURE 2-6 UML 2.0 Diagram Summary

UML Structure Diagrams

Mewakili data dan hubungan statis pada sistem informasi

1. Class Diagram
2. Object Diagram
3. Package Diagram
4. Deployment Diagram
5. Component Diagram
6. Composite Structure Diagram

UML Structure Diagrams

1. Class Diagrams
   - Kosakata umum yang digunakan oleh analis dan pengguna
   - Mewakili sesuatu/benda (*employee, paycheck,…*)
   - Menunjukkan hubungan antar kelas
2. Object Diagrams
   - Mirip dengan Class Diagram
   - Gambaran tentang objek-objek dalam sistem
   - Hubungan antar objek
3. Package Diagrams
   - Kelompok elemen-elemen UML digunakan untuk membentuk tingkat konstruksi yang lebih tinggi

UML Structure Diagrams

4. Deployment Diagrams
   - Menunjukkan arsitektur fisik dan komponen perangkat lunak sistem
   - For example, network nodes
5. Component Diagrams
   - Hubungan fisik di antara komponen perangkat lunak
   - Example – Client/Server (Mesin mana yang berjalan pada *software* yang mana)
6. Composite Structure
   - Menggambarkan struktur internal dari kelas yang kompleks

UML Behavior Diagrams

Menggambarkan hubungan dinamis antara objek yang mewakili sistem informasi bisnis

1. Activity Diagram
2. Sequence Diagram
3. Communication Diagram
4. Interaction Diagram
5. Timing Diagram
6. Behavior State Machine
7. Protocol State Machine
8. Use Case Diagrams

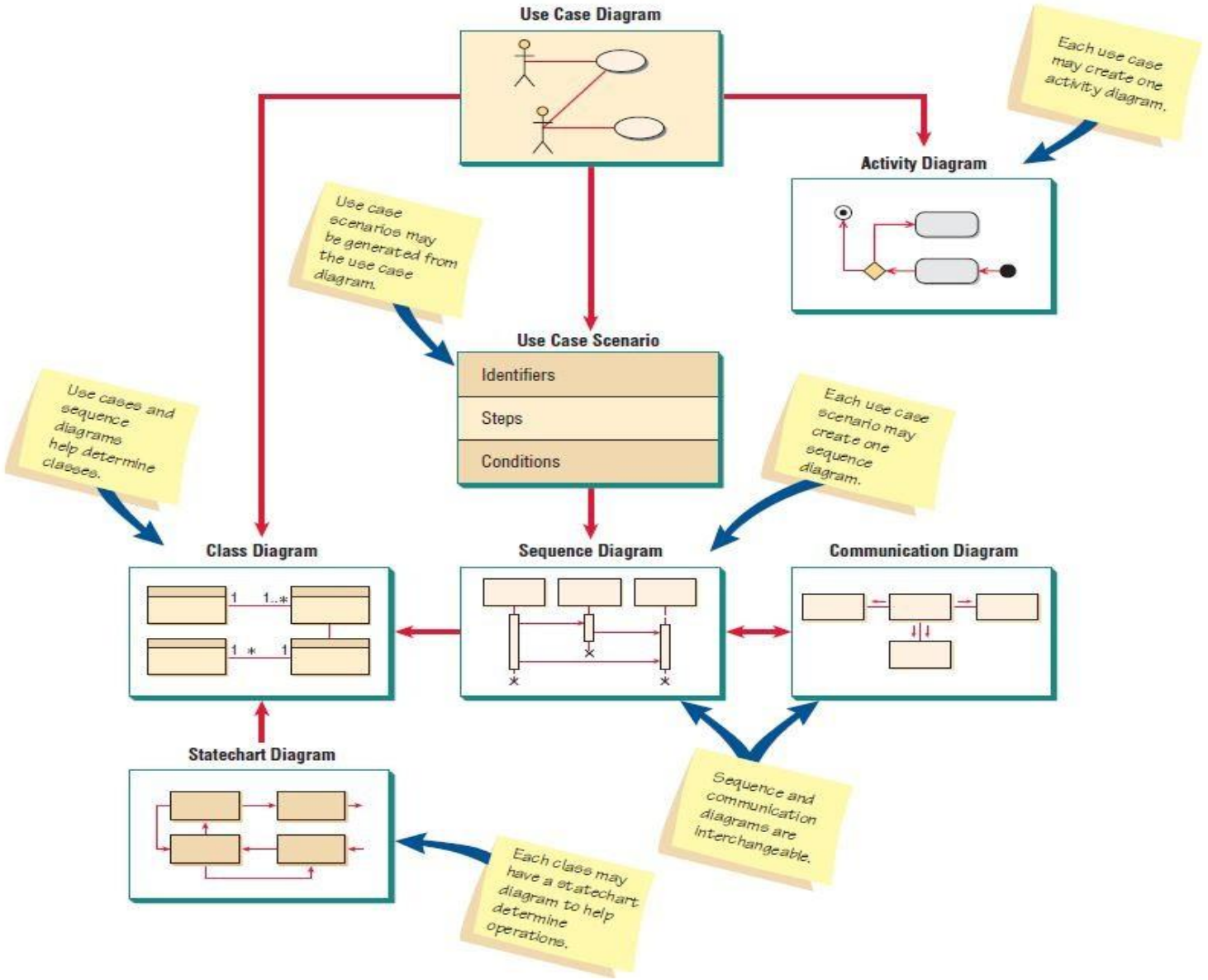
UML Behavior Diagrams

1. Activity Diagrams
   - Model proses pada suatu sistem informasi
   - Example: Business workflows, business logic
2. Interaction Diagrams
   - Menunjukkan interaksi anatar objek
3. Sequence Diagrams
   - Urutan berdasarkan waktu interaksi
4. Communication Diagrams
   - Komunikasi antara sekumpulan objek yang berkolaborasi dari suatu aktivitas

untuk mengekspresikan protokol penggunaan atau siklus hidup beberapa classifier

UML Behavior Diagrams

5. Protocol State Machine
   - Untuk mengkespresikan protocol penggunaan atau siklus hidup beberapa classifier
6. Timing Diagrams
   - Menunjukkan bagaimana suatu objek berubah dari waktu ke waktu
7. Behaviour State Machines
   - Memeriksa perilaku dari suatu kelas
   - Menunjukkan model keadaan-keadaan yang berbeda dan transisi keadaan dari suatu objek
8. Use-Case Diagrams
   - Menunjukkan interaksi antara sistem dan lingkungan
   - Menangkap kebutuhan bisnis


Use Case Diagram
Each use case may create one activity diagram.
Activity Diagram
Use case scenarios may be generated from the use case diagram.
Use Case Scenario
Identifiers
Steps
Conditions
Use cases and sequence diagrams help determine classes.
Each use case scenario may create one sequence diagram.
Class Diagram
Sequence Diagram
Communication Diagram
Statechart Diagram
Sequence and communication diagrams are interchangeable.
Each class may have a statechart diagram to help determine operations.

Tools pembuatan UML

- **Enterprise Architect**
- Rational Rose
- Visual Paradigm
- Microsoft Visio
- Star UML
- Netbeans UML Plugin, dll

Terimakasih

ANY QUESTIONS?