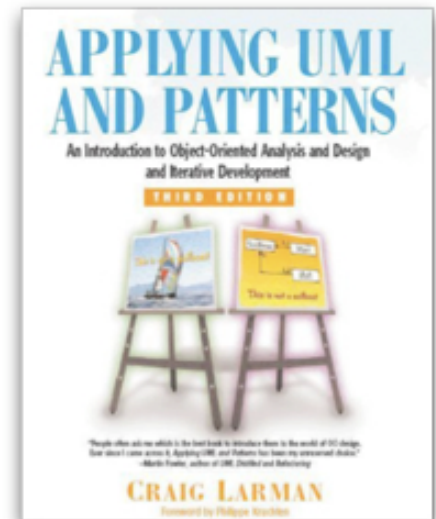




Object-Oriented Software Engineering

Pendahuluan

Slide untuk melengkapi buku Applying UML and Patterns
Versi 1.0



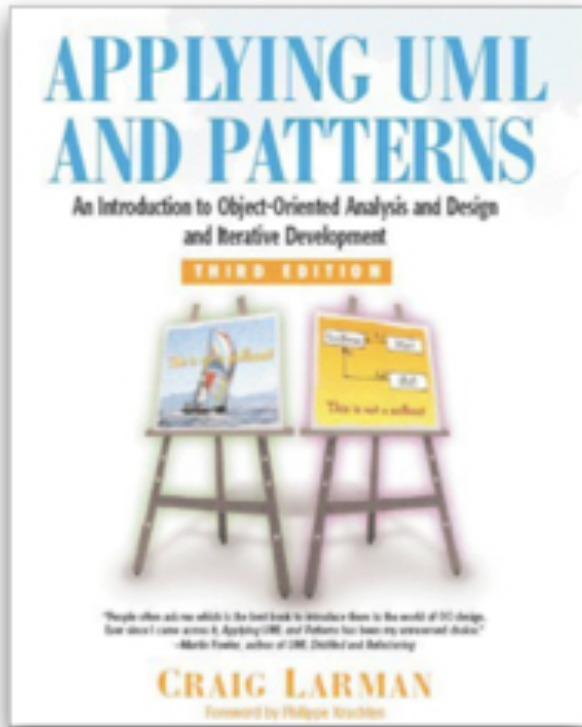
Program Studi Teknik Informatika

Jurusan Teknik Informatika
Politeknik Negeri Batam

Jalan Ahmad Yani, Batam Center, Batam 29461
www.polibatam.ac.id

Pustaka

- Materi kuliah diambil dari buku berikut:



Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development

Craig Larman

Addison Wesley Professional

ISBN 0-13-148906-2

Materi

- *Object Oriented Software Engineering*
- *Object Oriented Analysis and Design*
- *Unified Process*
- *Unified Modeling Language (UML)*
- Sejarah singkat

Catatan: materi kuliah ini menggunakan istilah-istilah dalam bahasa aslinya yaitu Bahasa Inggris untuk menghindari kesalahan arti

Object-oriented Software Engineering

- *Software engineering* merupakan sebuah disiplin ilmu yang berfokus pada riset, edukasi, dan penerapan proses dan metode engineering untuk meningkatkan produktivitas dan kualitas software
- *Object-oriented software engineering* (OOSE) adalah bagian khusus dari *software engineering*. OOSE melihat dunia dan semua sistem di dalamnya terdiri dari berbagai object yang saling terkait dan berinteraksi
 - *OO modeling and design languages* untuk berkomunikasi mengenai analisis dan desain
 - *OO software development processes* untuk membimbing proses pembangunan software
 - *OO software development methodologies* tentang cara melakukan aktivitas dalam software process
 - *OO development tools and environment* untuk mendukung proses dan metodologi pembangunan software

Object-oriented Analysis and Design

- Analisis menekankan pada investigasi mengenai masalah dan kebutuhan, bukan pada solusi. Sebagai contoh, jika sebuah sistem informasi akademik ingin dibangun, bagaimana sistem tersebut akan digunakan? Apa saja fungsi-fungsi di dalamnya?
- "Analisis" merupakan sebuah istilah yang luas, seperti juga istilah analisis kebutuhan (*requirements analysis*) – yaitu investigasi mengenai kebutuhan atau analisis berorientasi objek (object-oriented analysis) -- yaitu investigasi mengenai *domain objects*.

Object-oriented Analysis and Design

- Desain menekankan pada solusi konseptual (baik dalam software maupun hardware) untuk memenuhi kebutuhan, bukan mengenai implementasinya. Sebagai contoh, deskripsi skema basisdata dan object dalam software. Pada akhirnya, desain dapat diimplementasikan, dan implementasi (berupa kode) merupakan hasil penerapan desain.
- Sama seperti analisis, penggunaan istilah desain juga dirasa paling tepat, seperti dalam *object-oriented design* atau *database design*.
- Secara singkat kedua istilah tersebut dirangkum sebagai

do the right thing (analisis), and do the thing right (desain)

Object-oriented Analysis and Design

- Dalam *object-oriented analysis*, penekanan pada menemukan dan mendeskripsikan *problem domain*. Contohnya pada sistem informasi penerbangan, konsep yang ada di antaranya adalah *Plane*, *Flight*, dan *Pilot*.

domain concept



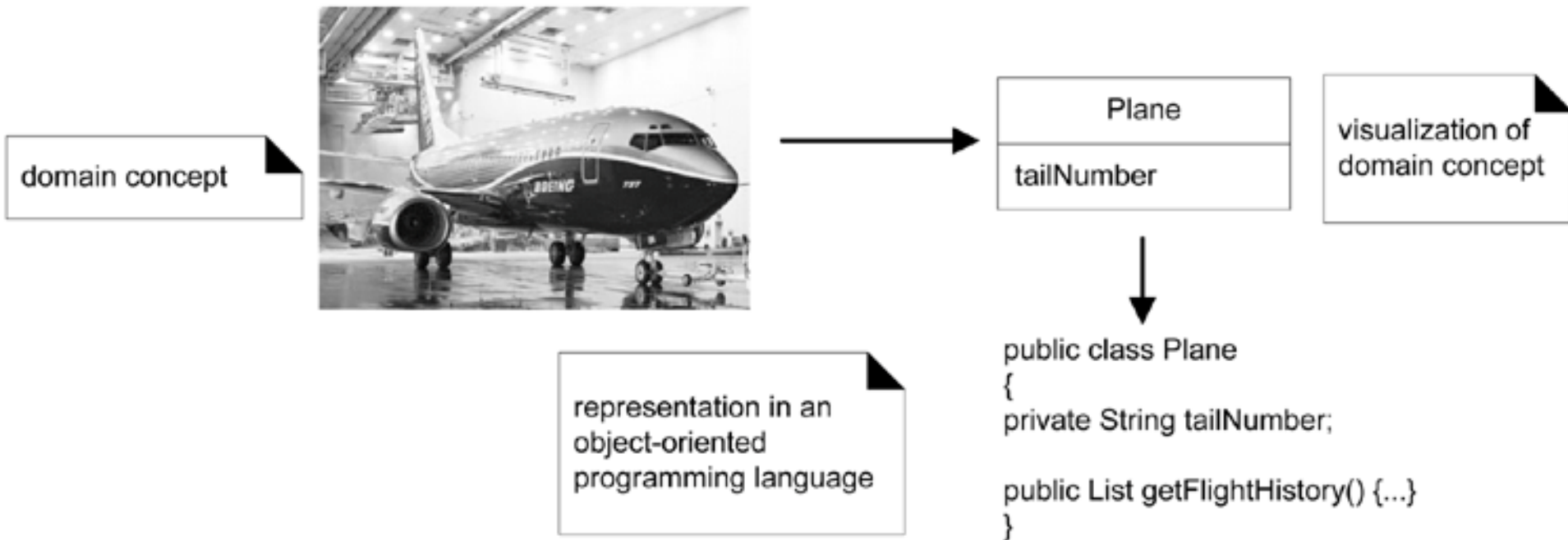
Object-oriented Analysis and Design

- Dalam *object-oriented design* (atau *object design*), penekanan pada mendefinisikan *software objects* dan bagaimana objek-objek tersebut bekerjasama untuk memenuhi kebutuhan. Contohnya sebuah *Plane* object memiliki atribut *tailNumber* dan method *getFlightHistory*.



Object-oriented Analysis and Design

- Terakhir, dalam implementasi atau pemrograman berorientasi objek, desain objects diimplementasikan, misalnya menjadi *Plane* class dalam Java.



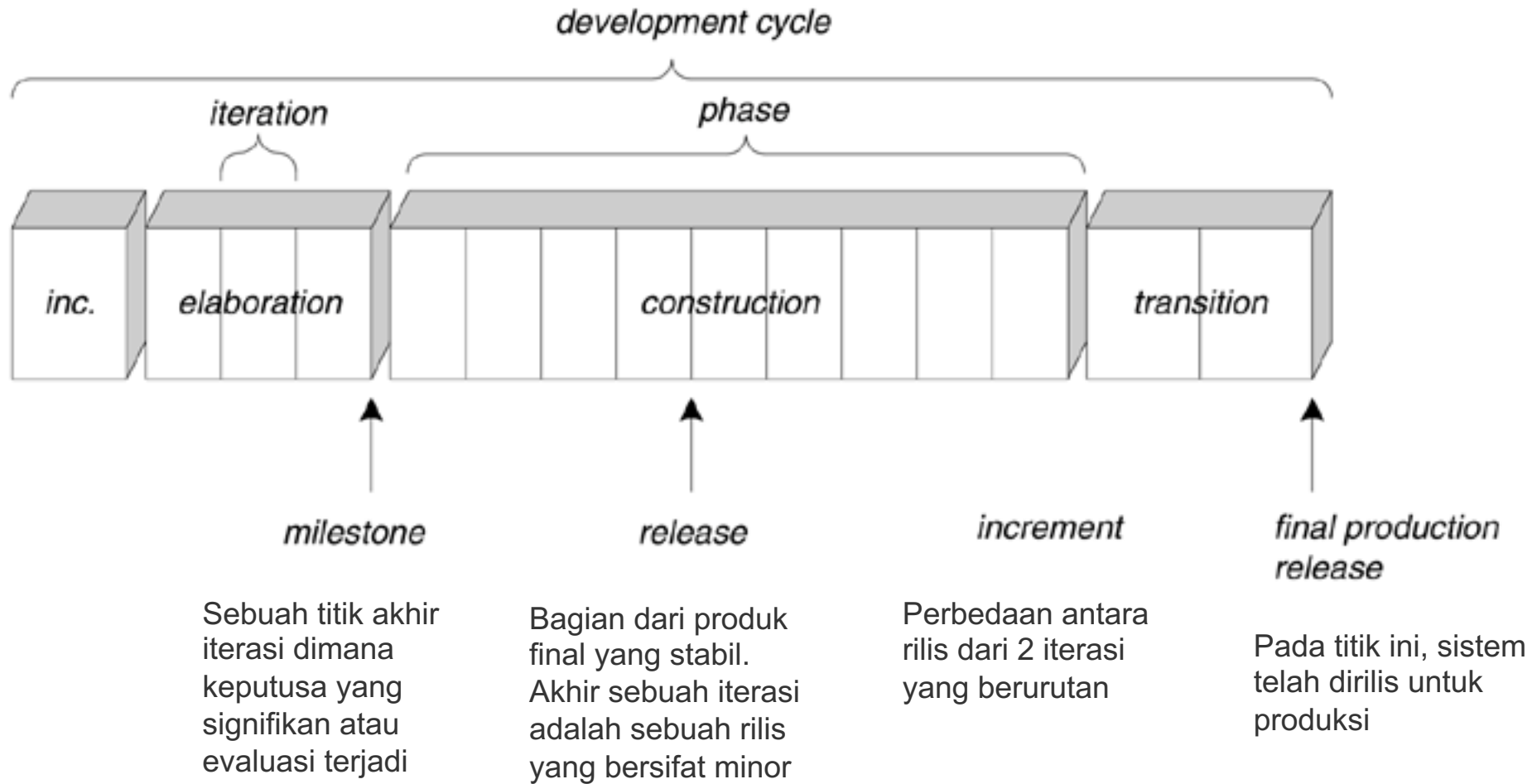
Unified Process (Jacobson, I., Booch, G., and Rumbaugh, J. 1999.)

- *Software development process* mendeskripsikan pendekatan untuk membangun, melakukan deployment, dan jika memungkinkan melakukan pemeliharaan software.
- **Unified Process** muncul sebagai salah satu software development process yang populer untuk membangun sistem berorientasi objek. Secara khusus, **Rational Unified Process** atau **RUP** (Kruchten, 2000.), penerapan lebih detail dari Unified Process, telah digunakan secara luas.

Unified Process

- UP project mengelola pekerjaan dan iterasi/pengulangan dalam empat fase besar:
 - *Inception* memperkirakan visi, proses bisnis, scope, perkiraan kasar.
 - *Elaboration* secara iteratif mendefinisikan arsitektur utama, memperhitungkan resiko, mengidentifikasi kebutuhan dan scope, perkiraan yang lebih realistis.
 - *Construction* secara iteratif melakukan implementasi berbagai elemen, dan mempersiapkan deployment.
 - *Transition* beta tests, deployment.

Unified Process

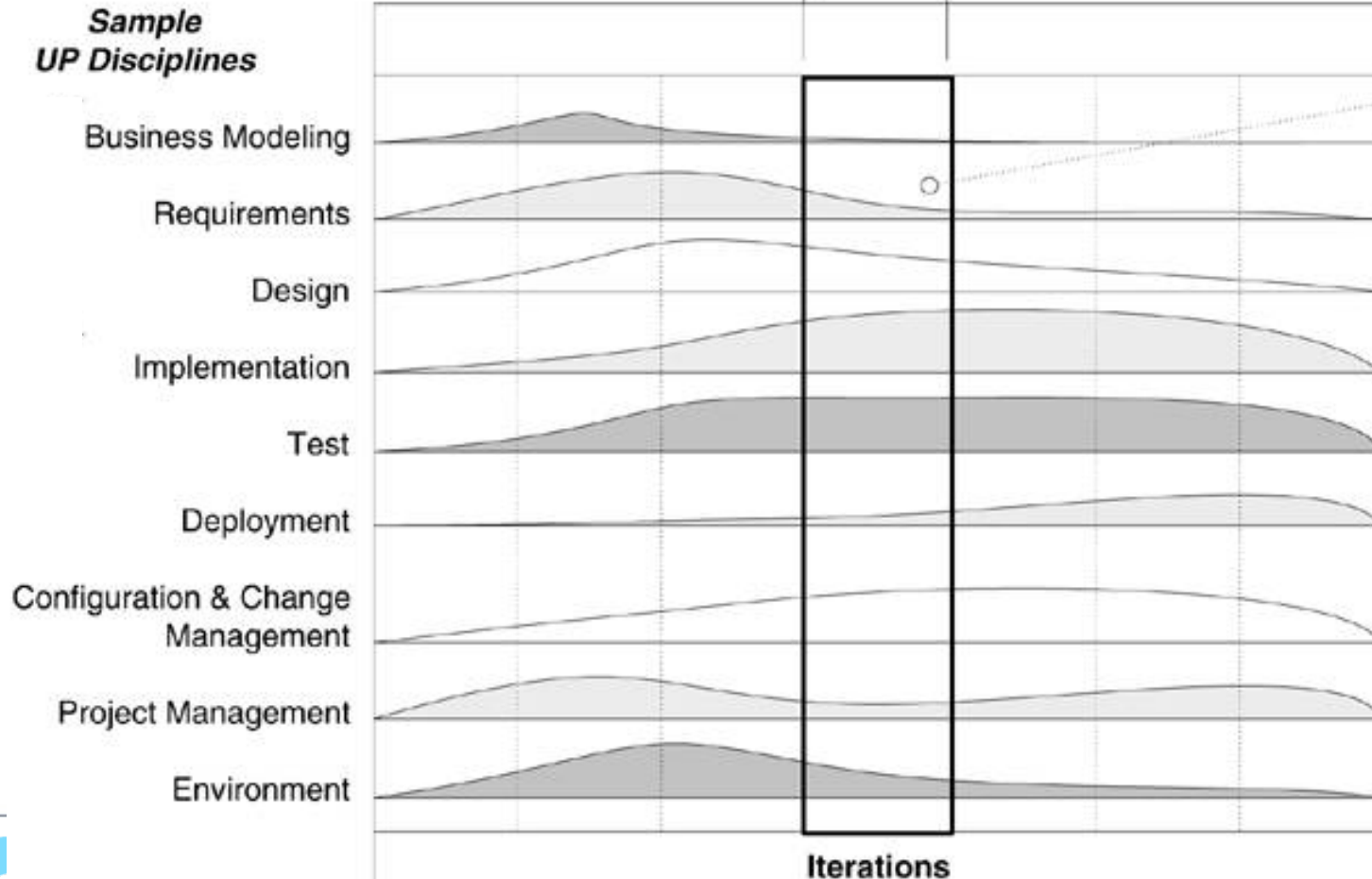


Unified Process

- UP mendeskripsikan aktivitas pekerjaan, seperti menuliskan use case, dalam **disciplines**, yaitu sekumpulan aktivitas (dan artifacts) dalam sebuah bidang, seperti aktivitas dalam *requirements analysis*. Dalam UP, **artifact** adalah istilah umum untuk produk yang dihasilkan: kode, Web graphics, database schema, text documents, diagrams, models, dan sebagainya.
- There are several disciplines in the UP

Unified Process

A four-week iteration (for example).
A mini-project that includes work in most disciplines, ending in a stable executable.



Note that although an iteration includes work in most disciplines, the relative effort and emphasis change over time.

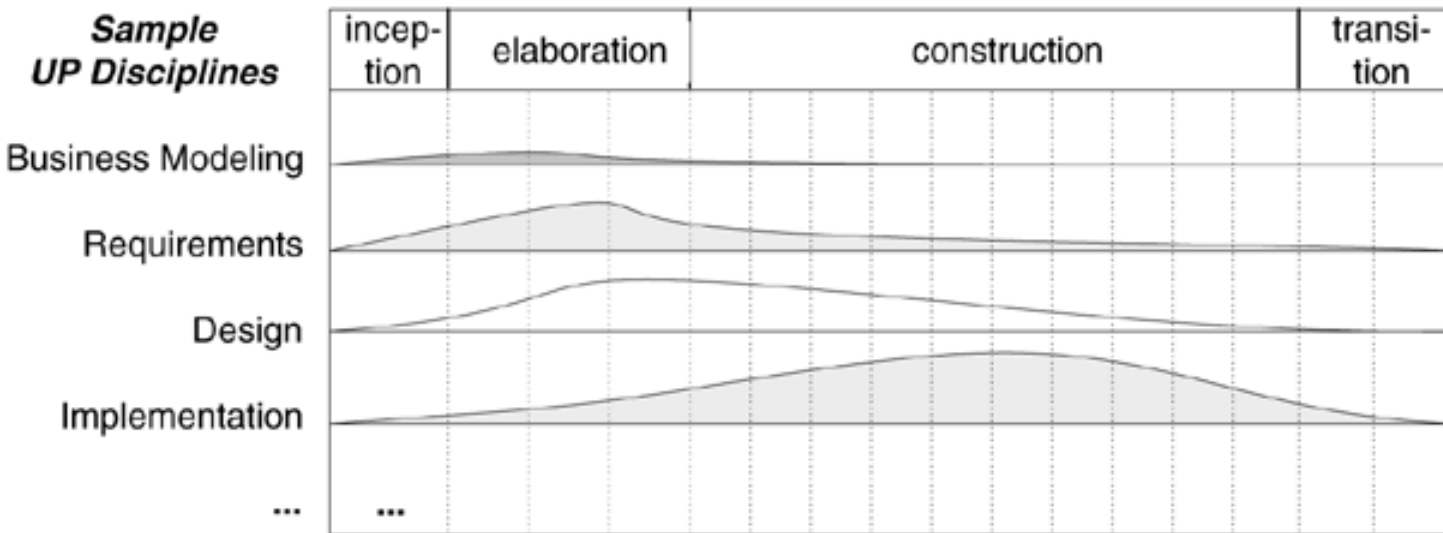
This example is suggestive, not literal.

Unified Process

Apa hubungan antara Discipline dengan Phase?

- Sebuah iterasi biasanya meliputi berbagai discipline. Namun, effort atau beban dari berbagai discipline akan berubah seiring waktu. Iterasi awal biasanya lebih menekankan pada *requirements* dan *design*, dan iterasi-iterasi selanjutnya tidak, seiring stabilnya requirements dan design melalui proses *feedback* dan *adaptation*.
- Menghubungkan hal tersebut dengan UP phases (*inception*, *elaboration*, ...), berikut adalah ilustrasi perubahan beban terhadap phase. Sebagai contoh, dalam *elaboration*, iterasi cenderung memiliki pekerjaan *requirements* dan *design* dalam jumlah tinggi, walaupun implementasi juga dimulai. Selama *construction*, penekanan ada pada implementasi dan bukan pada *requirements analysis*.

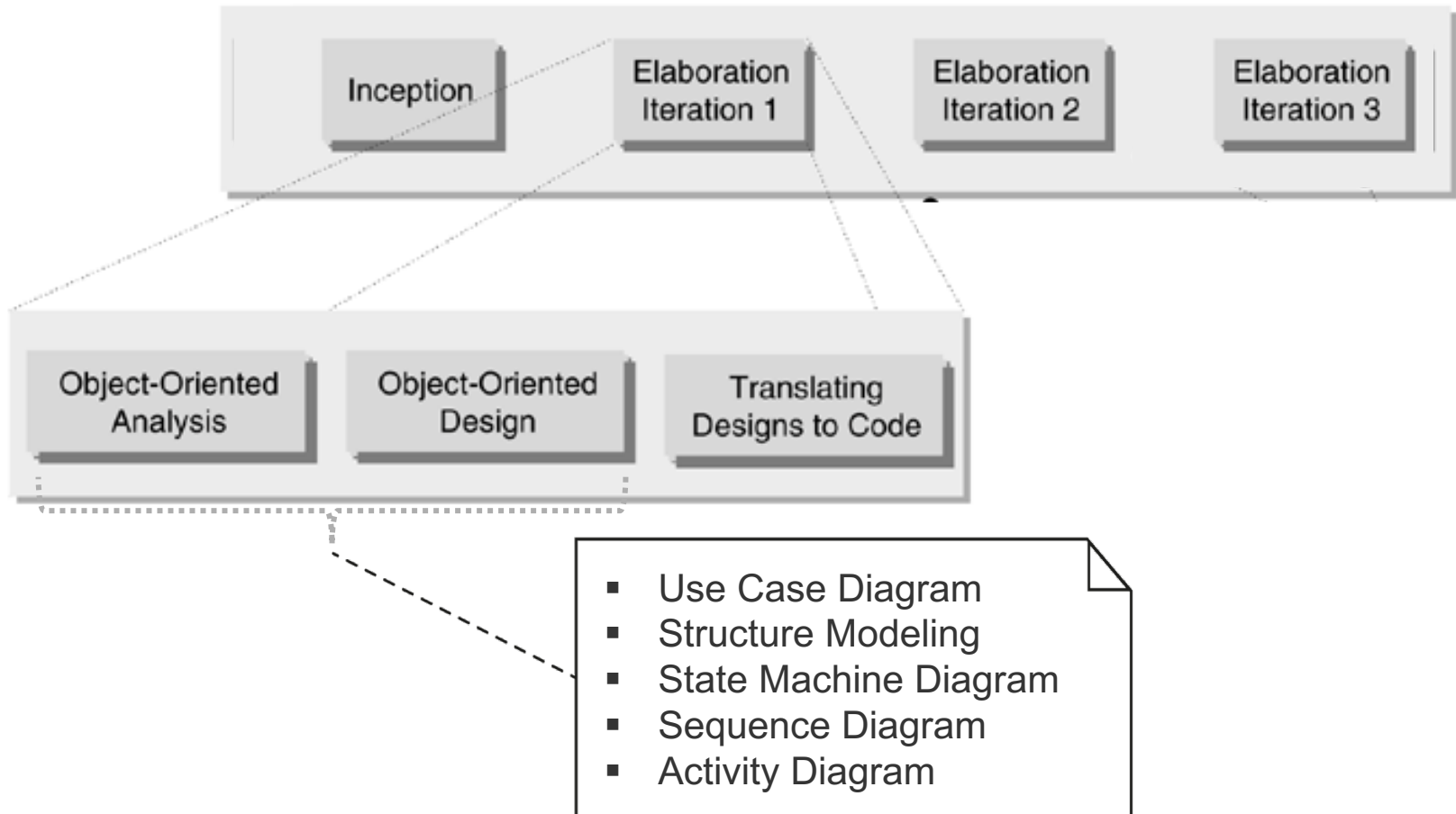
Unified Process



The relative effort in disciplines shifts across the phases.

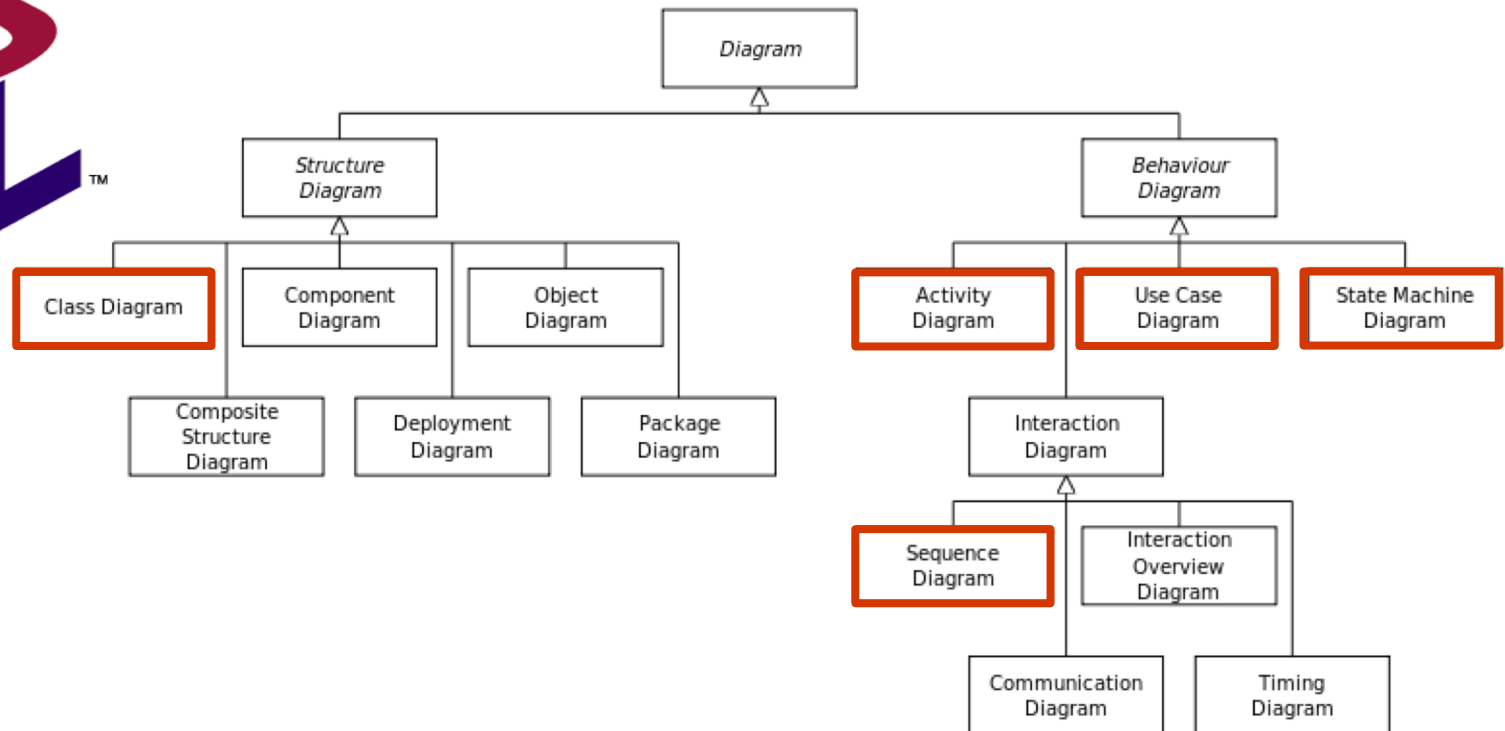
This example is suggestive, not literal.

OOAD vs UP



Unified Modeling Language (UML)

- Unified Modeling Language (UML) adalah bahasa visual untuk menspesifikasi, membangun, dan mendokumentasikan artifact sistem



Sejarah singkat

- Tahun 1960-an dan 1970-an muncul bahasa pemrograman OO, seperti Simula dan Smalltalk
- 1982 – *Grady Booch* (juga pencipta UML) menulis paper pertama dengan judul *Object-Oriented Design*
 - Pioner OOA/D lain juga menyumbangkan idenya dalam kurun 1980an: Kent Beck, Peter Coad, Don Firesmith, Ivar Jacobson (pencipta UML), Steve Mellor, Bertrand Meyer, Jim Rumbaugh (pencipta UML), dan Rebecca Wirfs-Brock.
- 1991 – dua buku utama tentang OOA/D books diterbitkan
 - Yang pertama mendeskripsikan metode OMT – object modeling technique (*Object-Oriented Modeling and Design*) oleh Rumbaugh et al.
 - Yang kedua mendeskripsikan metode Booch (*Object-Oriented Design with Applications*) oleh Booch.

Sejarah singkat

- 1994 – UML dimulai oleh Booch and Rumbaugh untuk mengkombinasikan kedua metode, metode Booch dan metode OMT serta mendefinisikan notasi yang sama.
 - Draft pertama UML dinamakan sebagai *Unified Method*
- Selanjutnya bergabung bersama Ivar Jacobson (pencipta metode Objectory) dalam Rational Corporation dan dikenal sebagai *three amigos*
- 1997 – UML 1.0 muncul, didefinisikan oleh Object Management Group (OMG, badan standar industri untuk berbagai standar yang terkait OO)
 - OMG task force dipimpin oleh Mary Loomis dan Jim Odell
- 1999 – Three amigos menerbitkan buku mengenai *software development methodology*
 - Mula-mula dinamakan *Rational Unified Process* (RUP), tapi kemudian disebut sebagai *Unified Software Development Process* (USDP) atau disingkat sebagai *Unified Process* (UP)