

11S3132- Software Design and Analysis

Unified Software Development Process

Present By: Ranty Deviana Siahaan, S.Kom, M.Eng.

**Week-2, Session-1, 2
T.A.25/26**



Institut Teknologi Del

Jl. Sisingamangaraja
Sitoluama, Laguboti 22381
Toba – SUMUT
<http://www.del.ac.id>

**Lecturer: RDS
- APPL 11S3132–
- Sem. Gassal 2025/2026 -**

Outline

- Intro
- Process vs Method vs Methodology
- Why Methodology
- Elements of Methodology
- Unified Software Development Process (USDP)

Intro

Unified Software Development Process (USDP) adalah kerangka kerja pengembangan perangkat lunak yang berorientasi objek, iteratif, dan berfokus pada *use case* untuk menghasilkan sistem yang terstruktur dan berkualitas tinggi.

Process, Method and Methodology

PROCESS

- Urutan aktivitas/tugas yang dilakukan dalam sebuah proyek.
- **Misalnya:** requirement gathering → design → coding → testing → deployment.

METHOD

- Cara atau teknik spesifik untuk melaksanakan suatu tugas.
- **Misalnya:** Brainstorming untuk elicitation, UML untuk desain.

METHODOLOGY

- Kerangka kerja menyeluruh (framework dan prinsip) untuk mengorganisasi proses dan metode.
- **Misalnya:** Agile, Waterfall, Unified Process.

Process, Method and Methodology

Aspek	Process	Method	Methodology
Definisi	Urutan aktivitas atau langkah dalam proyek	Teknik spesifik untuk melaksanakan suatu aktivitas	Kerangka kerja menyeluruh yang mengatur proses dan metode
Fokus	Apa yang dikerjakan.	Bagaimana cara mengerjakannya.	Mengapa dan dengan prinsip apa dikerjakan.
Contoh	Requirement → Design → Coding → Testing.	Brainstorming, Interview, UML, TDD	Agile, Waterfall, Unified Process
Level	Operasional (task)	Taktis (teknik)	Strategis (framework)

Process, Method or Methodology?

Method/Process

- Langkah-langkah *step-by-step* untuk menyelesaikan suatu pekerjaan.
- Biasanya spesifik untuk **satu proyek** (karena setiap proyek unik).

Methodology

- Kumpulan prinsip umum yang menjadi **panduan memilih metode** yang tepat untuk suatu tugas atau proyek.

Why Methodology?

- **Alasan utama:** Teknik dalam pengembangan sistem harus terorganisasi agar saling mendukung.

- **Contoh situasi:** Analis sudah membuat collaboration diagram untuk use case utama. **Muncul Pertanyaan:**

- Apakah harus lanjut ke sequence diagram dan menulis spesifikasi operasi?
- Atau fokus ke class diagram dengan inheritance dan composition?

Methodology adalah kombinasi pendekatan, teknik, siklus hidup, dan prinsip kerja → sebagai panduan menyeluruh dalam pengembangan sistem.

- **Keterbatasan UML:** UML hanya menyediakan **bahasa pemodelan, bukan aturan urutan kerja** → Tidak membantu menentukan apa langkah selanjutnya.
- **Peran Methodology:** Memberikan **panduan, alur, dan prinsip** agar semua teknik (UML, tools, praktik) bekerja selaras menuju tujuan proyek.

Why Methodology?

Isi dari sebuah methodology:

1. Pendekatan pengembangan → misalnya *object-oriented*.
2. Teknik dan notasi → misalnya: UML untuk pemodelan.
3. Model siklus hidup → misalnya: *spiral*, *incremental* untuk mengatur tahapan, dsb.
4. Prosedur dan filosofi terpadu → aturan dan prinsip kerja bersama.

Karakteristik Methodology:

- Lebih **abstrak** → meta process yang bisa diterapkan ke **banyak proyek**.
- Dapat di-**instantiate** → artinya **diterapkan spesifik** pada satu proyek.

Unified Software Development Process (USDP)

Apa itu USDP?

- Metodologi publik untuk pengembangan perangkat lunak berorientasi objek.
- Dikembangkan pertama kali oleh **Rational team**.

Evolusi

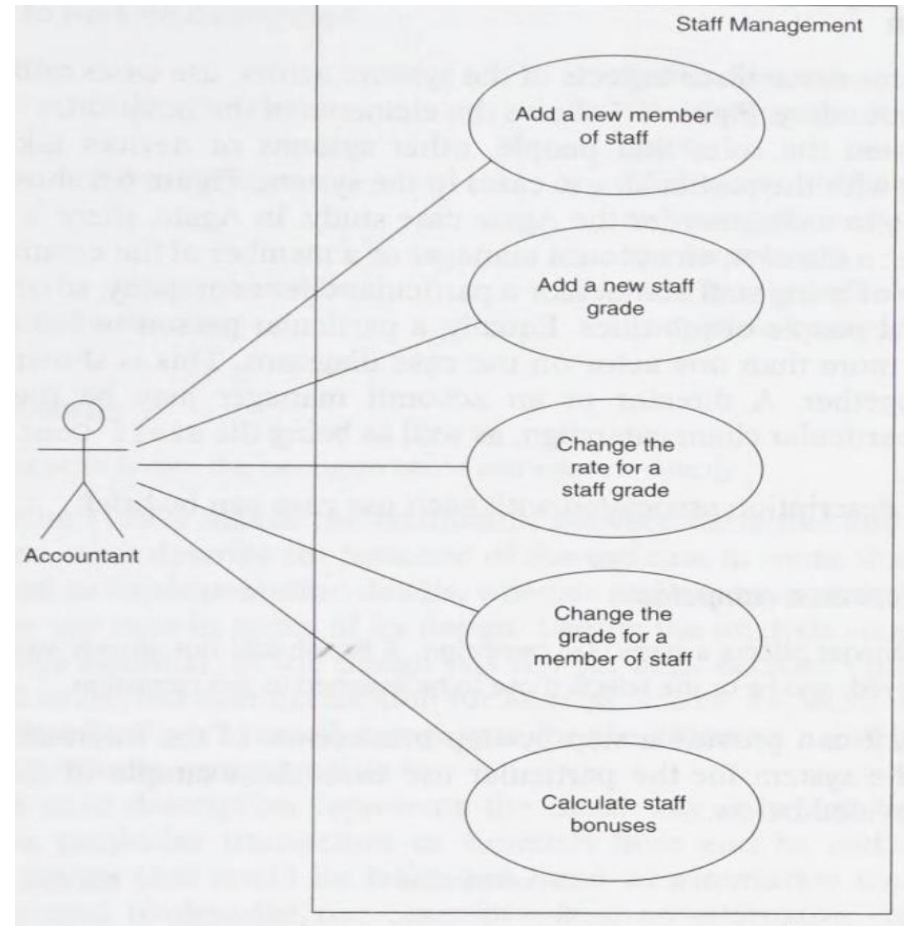
- Sekarang banyak digantikan oleh **Rational Unified Process (RUP)**, yang mirip dalam aspek utama.

Prinsip Utama:

- **Use-case driven** → kebutuhan digambarkan lewat use case.
- **Architecture-centric** → desain berpusat pada arsitektur sistem.
- **Iterative development** → dikerjakan bertahap dan berulang.
- **Incremental development** → hasil dibangun sedikit demi sedikit sampai lengkap.

Use-Case Driven

- Use case **merupakan** cerita singkat tentang apa yang dilakukan pengguna dengan sistem.
- Jadi titik awal desain sistem → dari use case, lahir diagram dan model lain.
- Setiap use case **merupakan** jembatan antara kebutuhan pengguna dan kode program.
- Membantu memastikan sistem benar-benar sesuai dengan kebutuhan nyata.



cara paling sederhana menjelaskan fungsi sistem dari sudut pandang pengguna

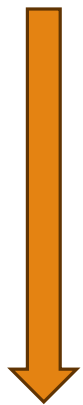
Architecture-Centric

- Dalam USDP, arsitektur sistem sudah dipikirkan sejak awal proyek.
- Arsitektur jadi kerangka utama untuk semua model dan aktivitas pengembangan.
- **Biasanya terlihat pada:** Boundary, Control, Entity classes → memisahkan bagian antarmuka, logika, dan data.
- Packages → mengelompokkan model dan kode agar rapi dan mudah dikelola.
- USDP menekankan arsitektur sebagai fondasi sejak awal, supaya sistem terstruktur dan mudah dikembangkan.

Iterative Development

Iterative artinya membangun sistem secara bertahap sambil terus memperbaiki hasil di tiap putaran.

- Siklus USDP itu berulang (*cyclic*):



- Analisis sebagian kecil kebutuhan
- Desain bagian itu.
- Koding sesuai desain.
- Uji hasil koding.
- Perbaiki dan ulangi siklusnya.



- Sehingga, kesalahan bisa ditemukan lebih cepat.
- Sistem berkembang sedikit demi sedikit sampai lengkap.

Incremental Development

- Sistem dibangun sedikit demi sedikit dalam bentuk bagian (*increment*) yang sudah bisa dipakai.
- Setiap *increment* adalah potongan software yang berfungsi penuh.
- Sederhananya, satu *use case* **adalah** satu increment yang dikirimkan ke pengguna.

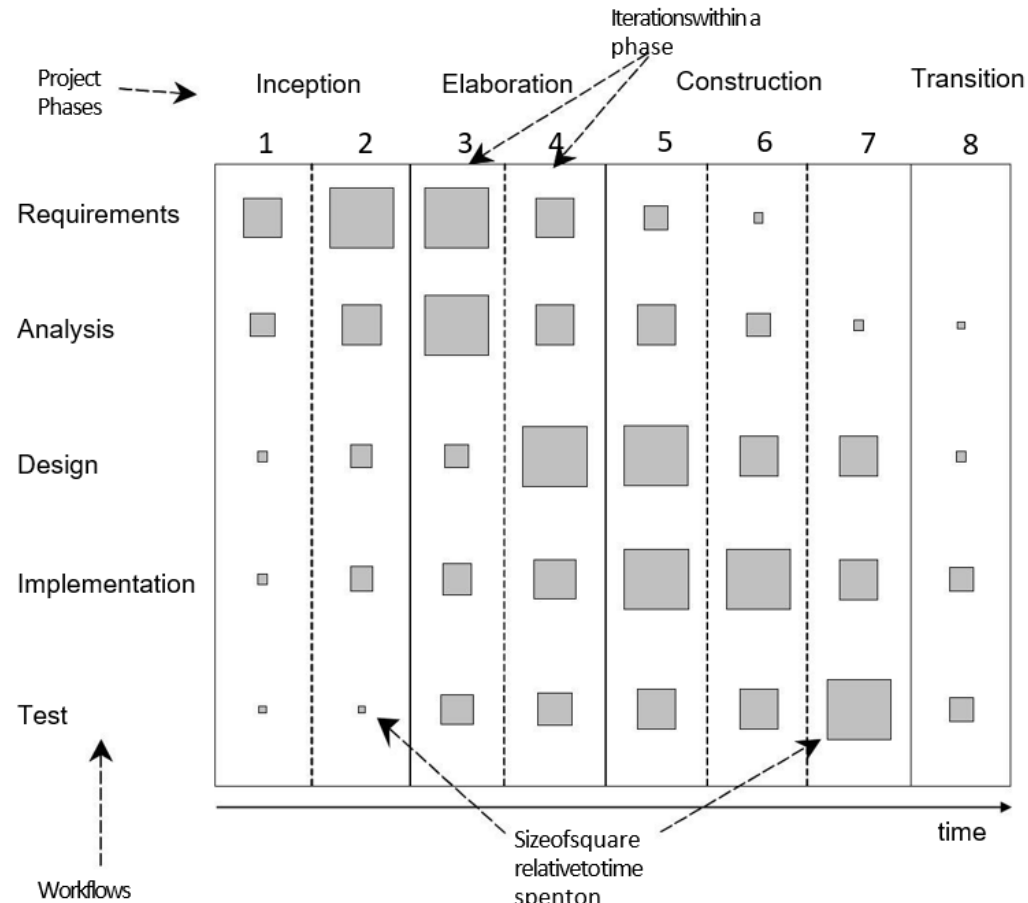
Contoh:

- Iterasi 1: fitur Login.
- Iterasi 2: fitur Input Data Mahasiswa.
- Iterasi 3: fitur Cetak Laporan.

membangun software per bagian yang langsung bisa digunakan, lalu digabung hingga sistem lengkap

Four Phases

Empat tahapan ini
(**Inception**, **Elaboration**,
Construction, **Transition**)
adalah fase utama dalam
USDP yang menggambarkan
perjalanan proyek dari **ide**
awal → **perincian**
kebutuhan →
pembangunan sistem →
penerapan ke pengguna.



Inception Phase

memahami kebutuhan dan
membuktikan ide proyek lewat
analisis dan percobaan kecil.

- Tahap studi kelayakan: menilai apakah manfaat proyek lebih besar daripada risikonya.
- Proyek diuji dengan membuat versi kecil dari *software* untuk membuktikan ide bisa berjalan.

Main Activity:

- Mengumpulkan kebutuhan (requirements capture).
- Analisis awal untuk memahami ruang lingkup proyek.
- Sedikit desain, implementasi, dan testing sebagai percobaan awal.
- Iterasi kecil mungkin sudah dilakukan meski masih tahap awal.
- Pendekatan Object-Oriented (OO) mendukung iterasi ini.

Elaboration Phase

memperdalam analisis dan desain agar proyek lebih jelas, risiko berkurang, dan biaya lebih pasti

- Membuat desain sistem yang sesuai.
- **Tujuan:** mengurangi ketidakpastian biaya dan risiko.
- Menunjukkan bahwa sistem bisa dibangun sesuai waktu dan anggaran.
- Lebih banyak waktu dipakai untuk desain.
- Ada sedikit peningkatan pada implementasi dan testing, tapi fokus masih di analisis dan desain.

Construction phase

fase membangun dan menguji sistem
sampai benar-benar berfungsi

- Membangun sistem lewat beberapa iterasi hingga siap dipakai di lingkungan nyata.
- Implementasi dan testing jadi aktivitas utama.
- Setiap iterasi makin sedikit fokus ke desain, lebih banyak ke pengujian dan penyempurnaan.

Transition phase

fase menyerahkan sistem ke pengguna,
memperbaiki masalah, dan mengganti
sistem lama bila perlu.

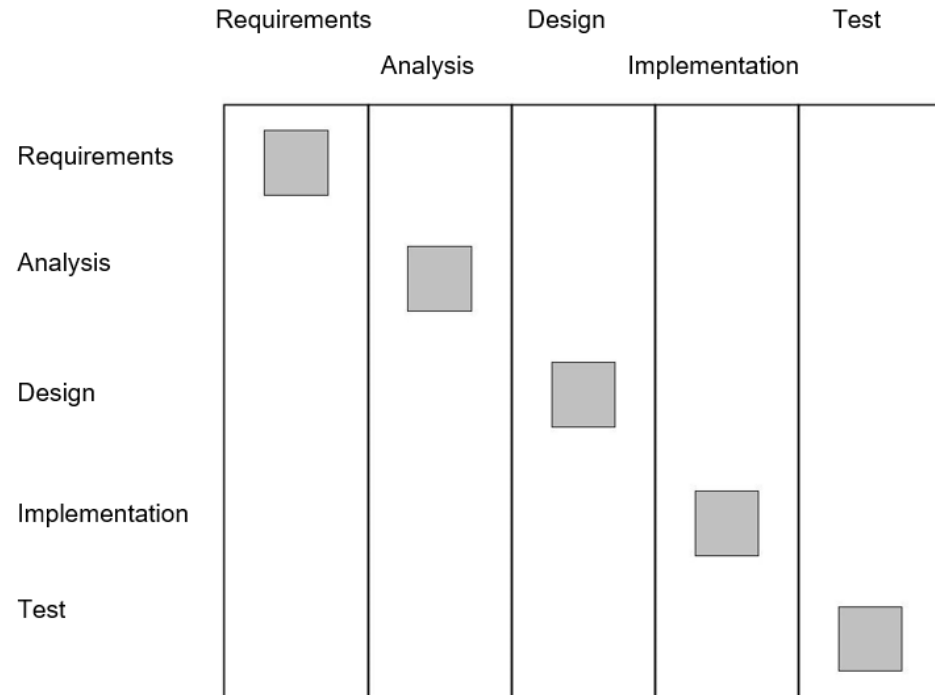
- Sistem mencapai kapasitas penuh sesuai tujuan.
- Menangani bug atau masalah yang muncul.
- Bisa termasuk konversi sistem lama ke sistem baru.

Phases, Workflows and Iterations

- Setiap fase dalam USDP berisi aktivitas yang dikelompokkan dalam workflow, dengan porsi usaha berbeda di tiap fase, dan tiap fase bisa dijalankan berulang (iterasi).
- Hal ini menjelaskan bahwa empat fase USDP (Inception, Elaboration, Construction, Transition) tidak berjalan sekali jalan, tapi terdiri dari workflow dan iterasi yang berulang untuk menyempurnakan sistem.

Difference from Waterfall Life Cycle

- Waterfall → tiap fase dikerjakan berurutan penuh.
- Contoh: semua kebutuhan selesai dulu → baru analisis → baru desain → baru implementasi → terakhir testing.
- Iterative (seperti USDP) → fase bisa berjalan paralel dan berulang.
- Contoh: sebagian kebutuhan masih bisa ditambah/diperbaiki sambil analisis atau desain berjalan.



Waterfall itu kaku dan berurutan, sedangkan Iterative itu fleksibel dan bisa paralel/berulang.

Major Activities of the Development Process

Activity	Techniques	Key Deliverables Use
Requirements Capture and Modelling	Requirements Elicitation Use Case Modelling Architectural Modelling Prototyping	Case Model Requirements List InitialArchitecture Prototypes Glossary

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Requirements Analysis	Communication Diagrams Class and Object Modelling Analysis Modelling	Analysis Models

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
System Architecture and Design	Deployment Modelling Component Modelling Package Modelling Architectural Modelling Design Patterns	Overview Design and Implementation Architecture

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Class Design	Class and Object Design Models Modelling Interaction Modelling State Modelling Design Patterns	

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
User Interface Design	Class and Object Modelling Interaction Modelling State Modelling Package Modelling Prototyping Design Patterns	Design Models with Interface Specification

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Data Management Design	Class and Object Modelling Interaction Modelling State Modelling Package Modelling Design Patterns	Design Models with Database Specification

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Construction	Programming ComponentRe-use Database DDL Programming Idioms ManualWriting	Constructed System Documentation

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Testing	Programming TestPlanning and Design Testing	TestPlans TestCases TestedSystem

Major Activities of the Development Process

Activity	Techniques	Key Deliverables
Implementation	Planning Training Data Conversion	Installed System

References

1. Bennet, Simmon. et al. 2002.Object-Oriented System Analysis and Design Using UML. 2 nd Edition.McGraw-Hill.
2. Booch, Grady.et al. 2007. Object-Oriented Analysis and Design With applications.3 rd Edition. Addison-Wesley.
3. Valacich, Joseph S. 2017. Modern System Analysis and Design. 8th Edition. Pearson.
4. Budgen, David.2003. Software Design. 2 nd Edition. Pearson
5. Sierra, Kathy. Baters, Bert. 2005.Head First OOAD. O'Reilly.

Thank you😊
