



REKAYASA PERANGKAT LUNAK I

PROSES PEMBANGUNAN PERANGKAT LUNAK

Agenda Perkuliahan

Software Development Life Cycle

Generic Process Model

Prescriptive Process Model

Software Development Life Cycle (SDLC)

PENGERTIAN SDLC

“Sekumpulan kegiatan dan keterhubungannya satu sama lain untuk mendukung pembangunan dari sebuah perangkat lunak”

AKTIFITAS PEMBANGUNAN PERANGKAT LUNAK

- ❖ Gathering Requirements
- ❖ Team Management (incl. Analysis)
- ❖ Software Design
- ❖ Coding
- ❖ Testing
- ❖ Documentation
- ❖ Software Maintenance

PERTANYAAN SEPUTAR SDLC

- ❖ **Aktifitas mana yang harus dipilih dalam pembangunan perangkat lunak?**
SDLC berisi proses yang bukan untuk dipilih tapi untuk dilakukan secara keseluruhan.
- ❖ **Hubungan antar aktifitas dalam SDLC?**
Hasil dari kegiatan sebelumnya mempunyai pengaruh yang sangat besar untuk kegiatan berikutnya.
- ❖ **Bagaimana cara menjadwalkan kegiatan pada SDLC?**
Penjadwalan kegiatan SDLC dibahas dalam ilmu Manajemen Proyek (Time Management) dan tidak dibahas pada RPL.

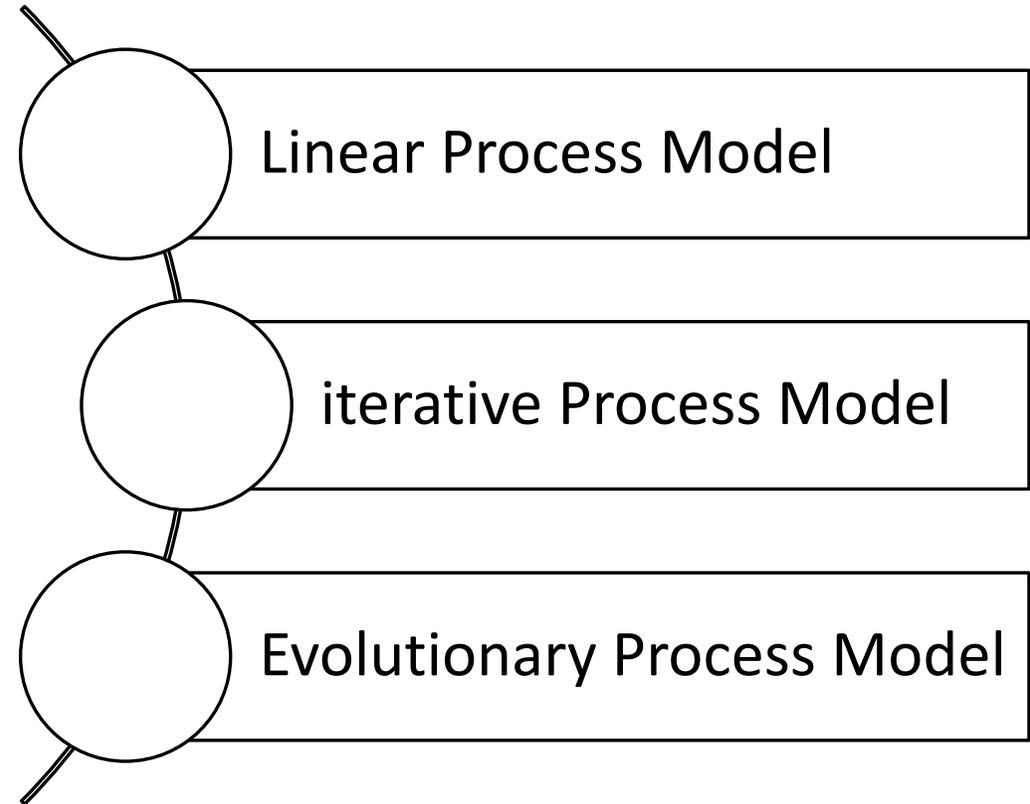
Generic Process Model

DEFINISI GENERIC PROCESS MODEL

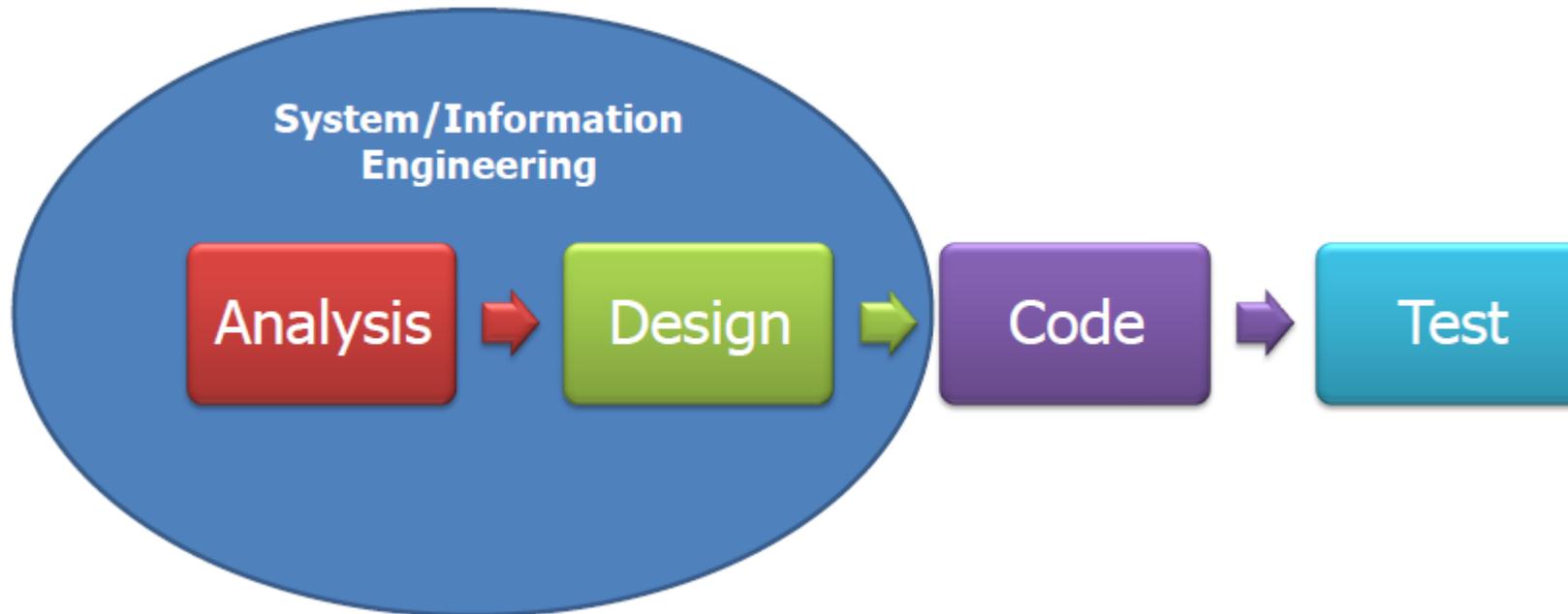
Terdiri dari 5 aktifitas umum dalam pembangunan perangkat lunak, yaitu:

- ❖ **Communication**
- ❖ **Planning**
- ❖ **Modeling**
- ❖ **Construction**
- ❖ **Deployment.**

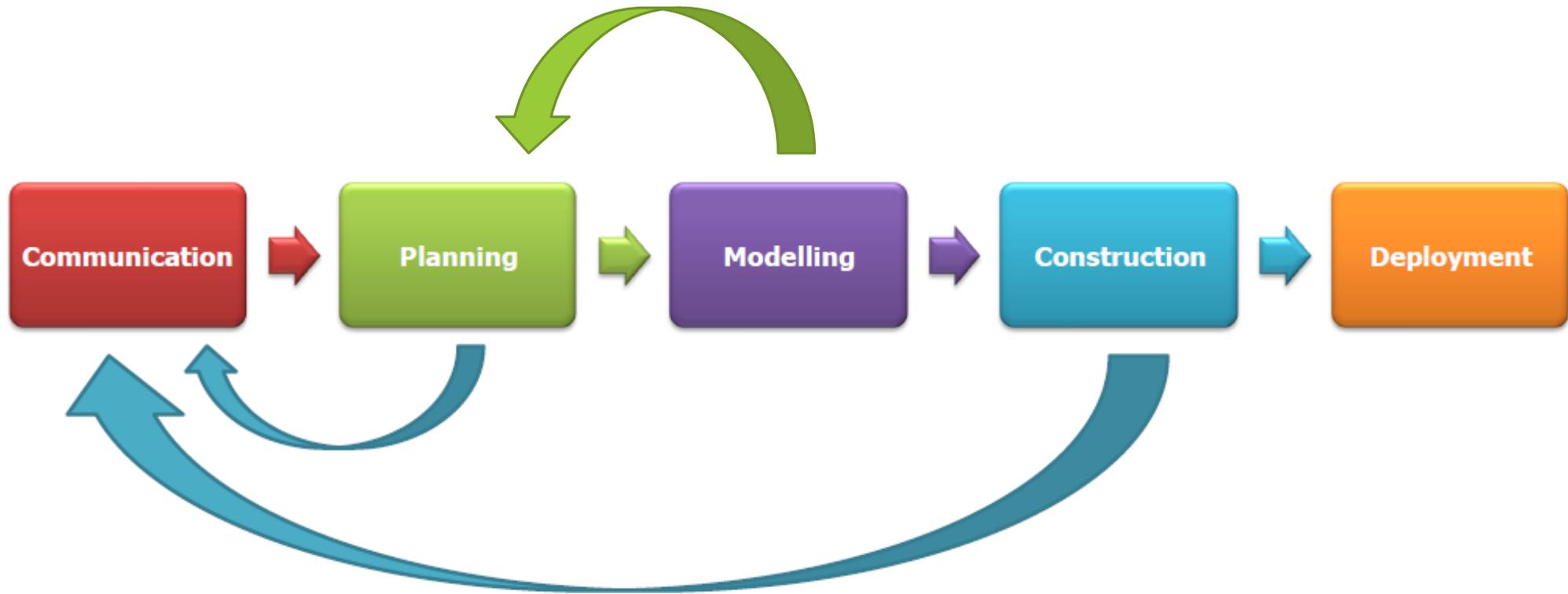
Process Flow



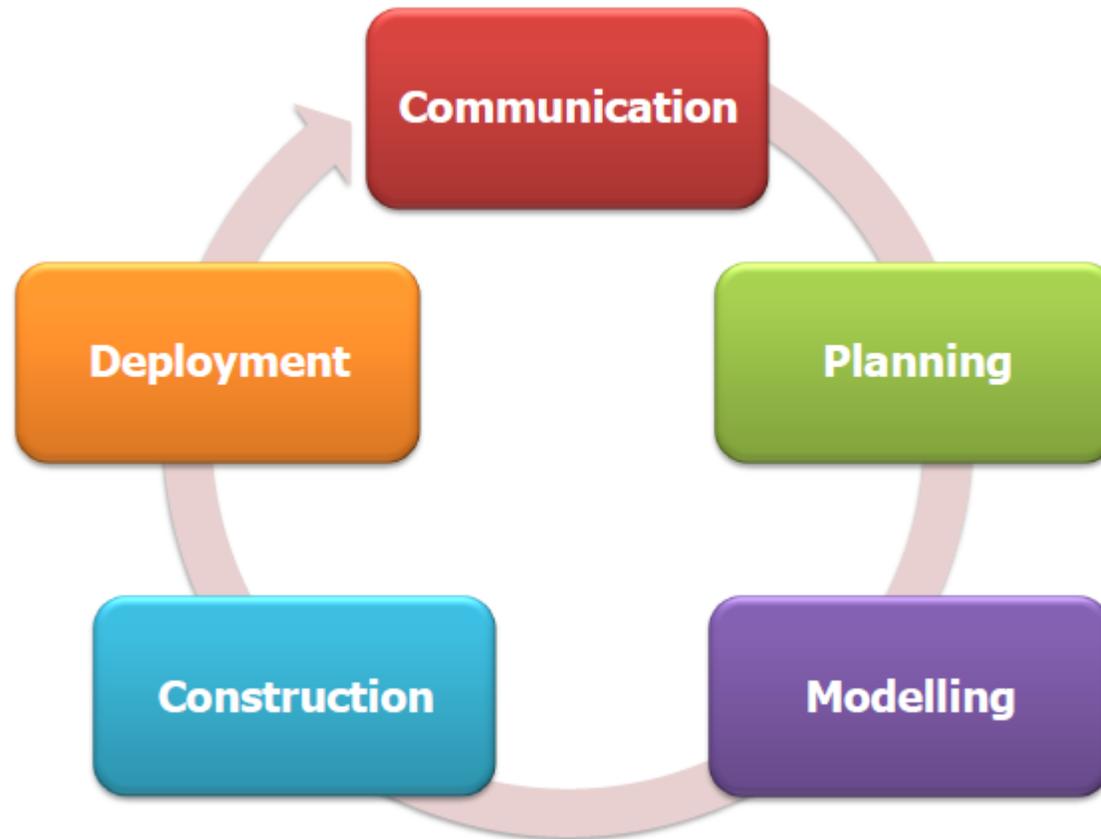
Linier Process Model



Iterative Process Model



Evolutionary Process Model



Prescriptive Process Model

Prescriptive Process Model

Waterfall Model

V Model

Incremental Process Model

Evolutionary Process Model

Specialized Process Model

Unified Process

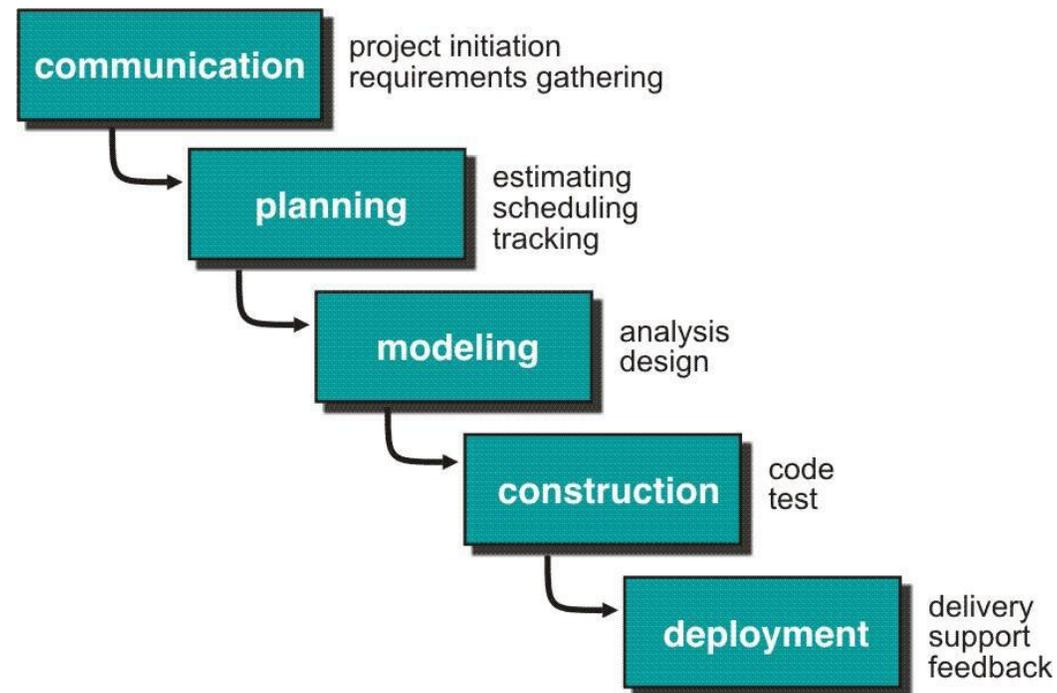
Agile Methods

1. Waterfall Model

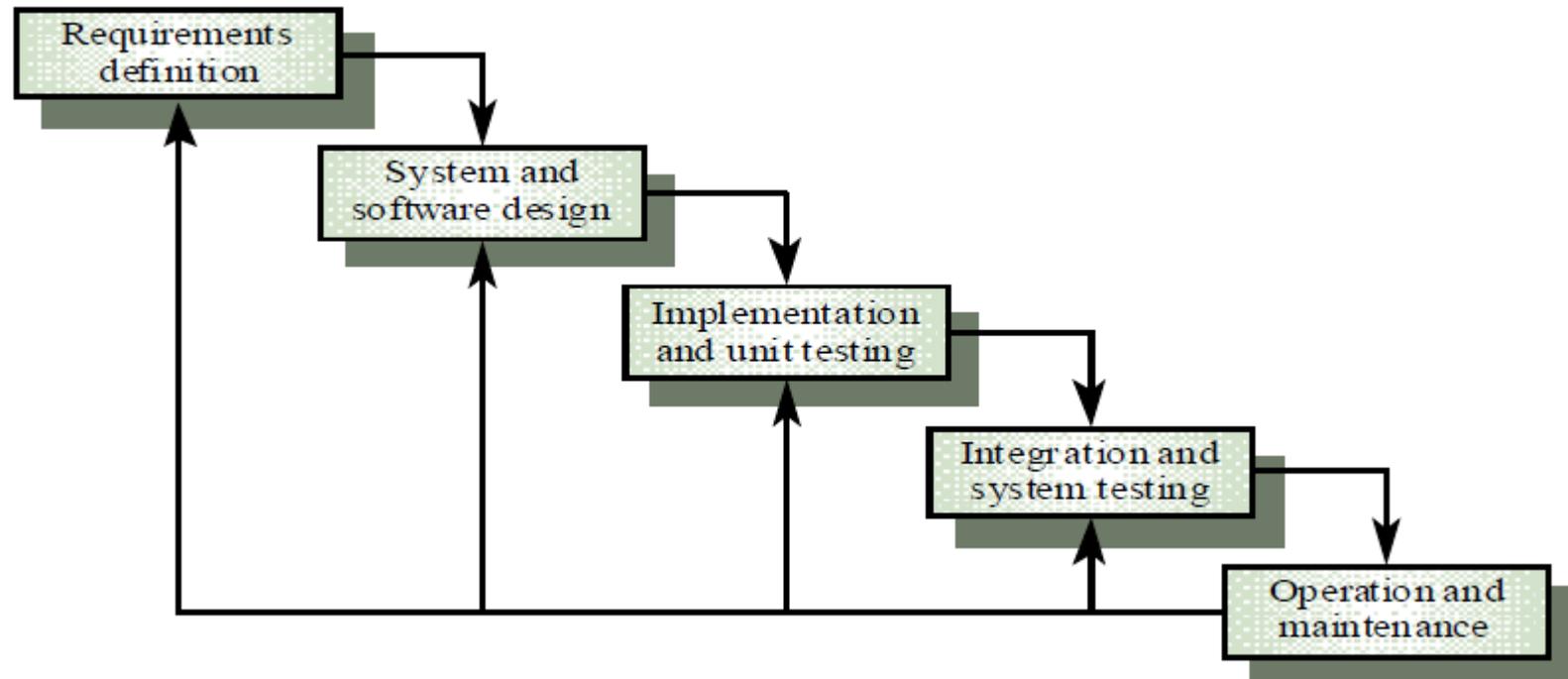
KARAKTERISTIK WATERFALL MODEL

- ❖ Setiap tahap menghasilkan dokumen di akhir tahapnya.
- ❖ Tidak ada overlapping pada setiap tahapnya.
- ❖ Setiap tahapan akan punya pengaruh besar pada hasil di tahap berikutnya.
- ❖ Memerlukan biaya besar jika melakukan rework.

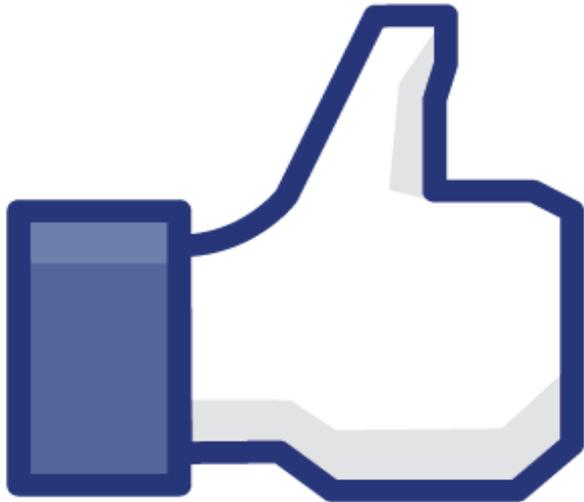
GAMBAR WATERFALL MODEL (ROGER S, PRESSMAN)



GAMBAR WATERFALL MODEL (IAN SOMMERFILLE)

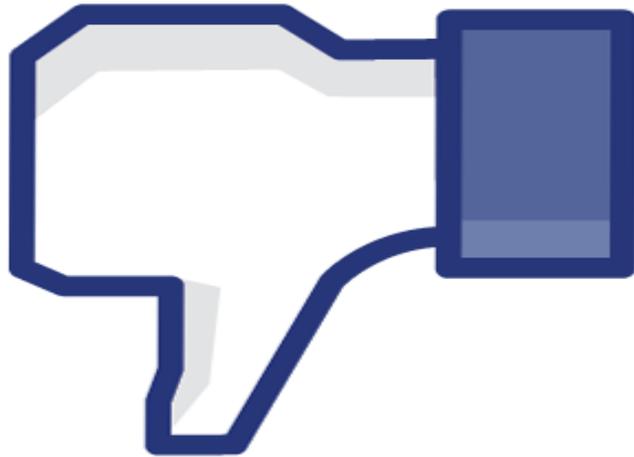


KELEBIHAN METODE WATERFALL



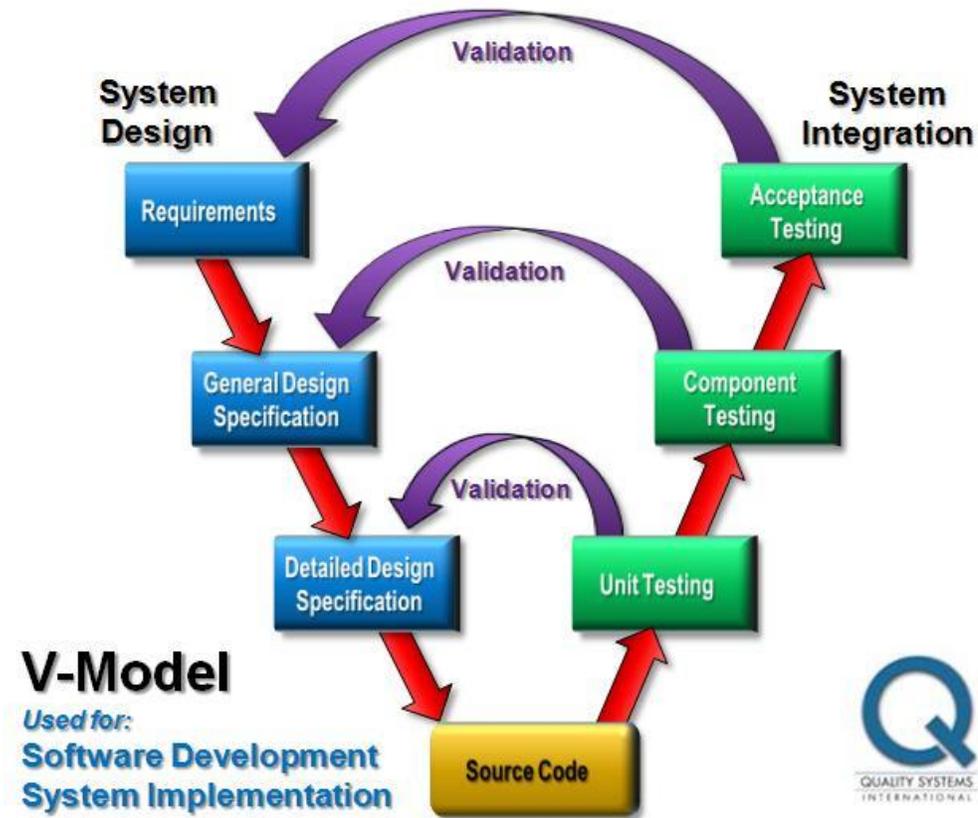
- ❖ Menghasilkan mature process pada setiap tahapnya.
- ❖ Mudah untuk diaplikasikan pada sebuah proyek.
- ❖ Menghasilkan sistem yang terstruktur dengan baik.
- ❖ Memiliki tingkat visibilitas yang tinggi (setiap tahap mempunyai dokumen yang jelas).

KEKURANGAN METODE WATERFALL



- ❖ Ketidak fleksibelan antar tahapan.
- ❖ Susah dalam merespon perubahan kebutuhan customer.
- ❖ Model ini hanya cocok jika:
 - a) Kebutuhan customer sudah sangat jelas
 - b) Perubahan kebutuhan dibatasi.

V Model



INCREMENTAL PROCESS MODEL

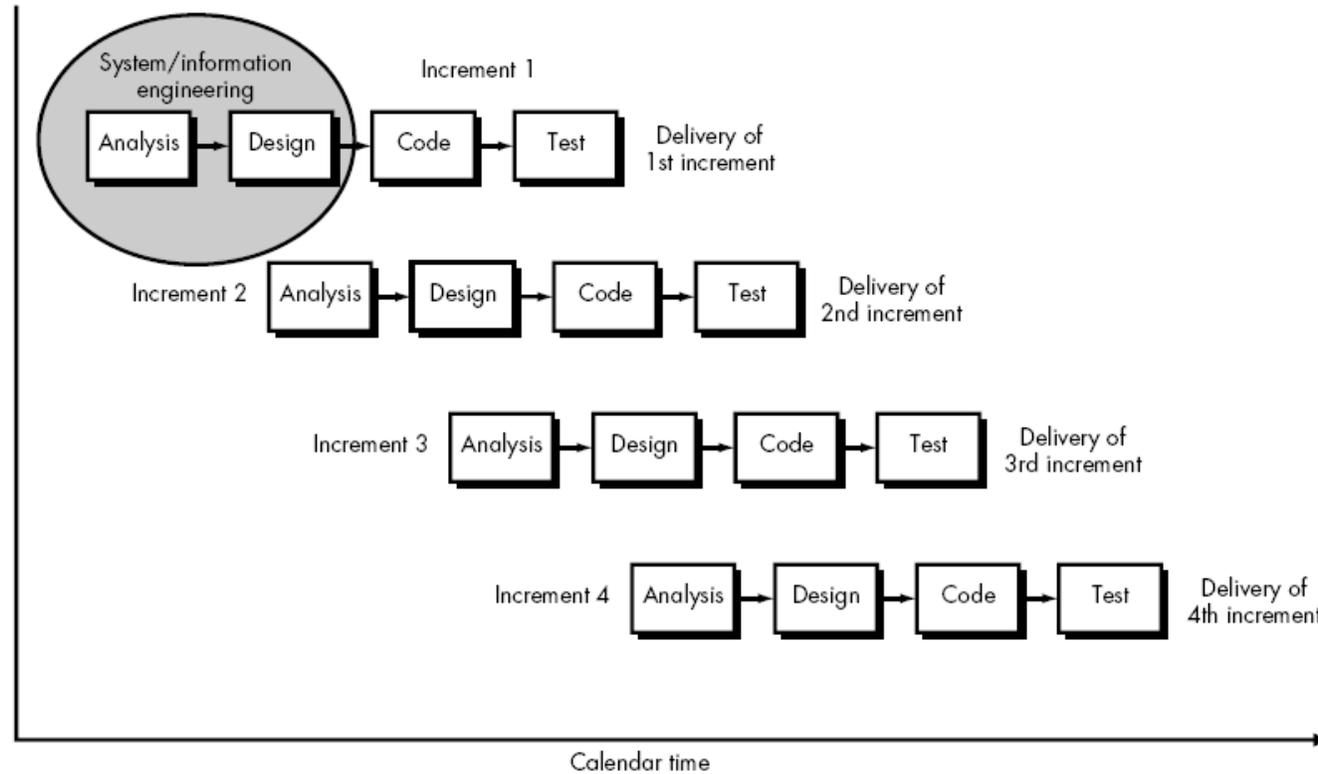
❖ Incremental Model

“Rework tidak harus menunggu satu siklus selesai. Satu siklus dianggap sebagai satu increment“

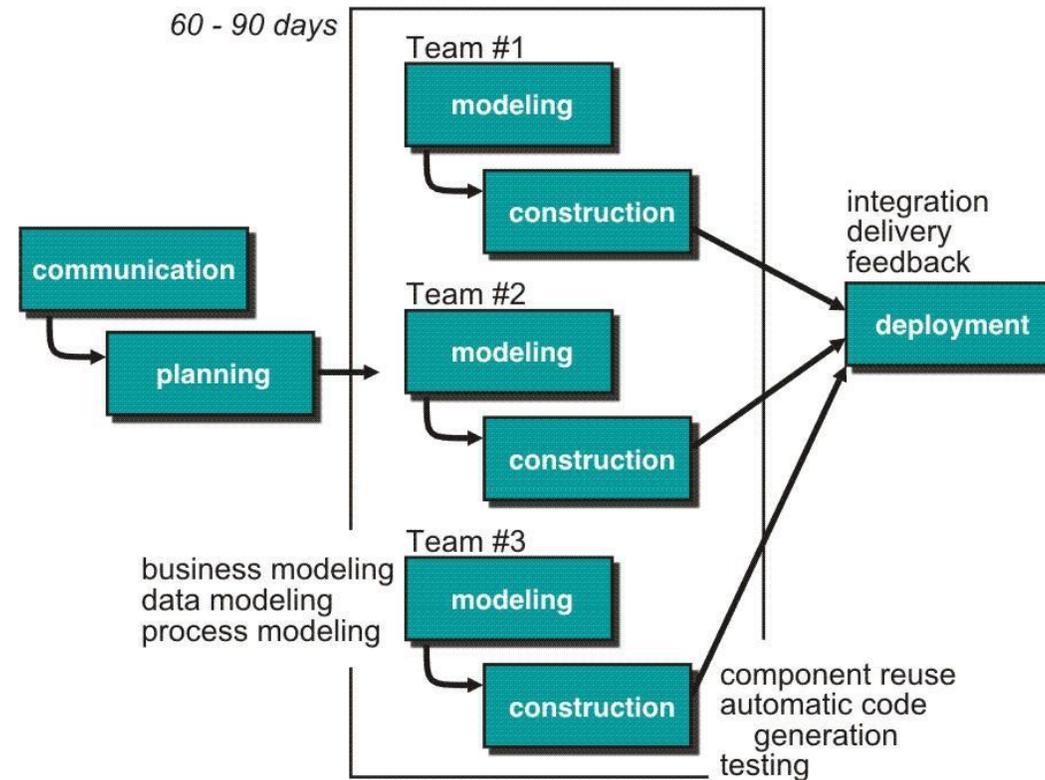
❖ Rapid Application Development (RAD)

“Ada pembagian tim dan pekerjaan yang jelas pada tahap modelling dan construction (berulang dalam kurun waktu tertentu)”

Incremental Model



RAD Model



EVOLUTIONARY PROCESS MODEL

❖ Prototyping

“Membantu software engineer untuk mendapatkan kebutuhan berupa feedback cepat dari customer”

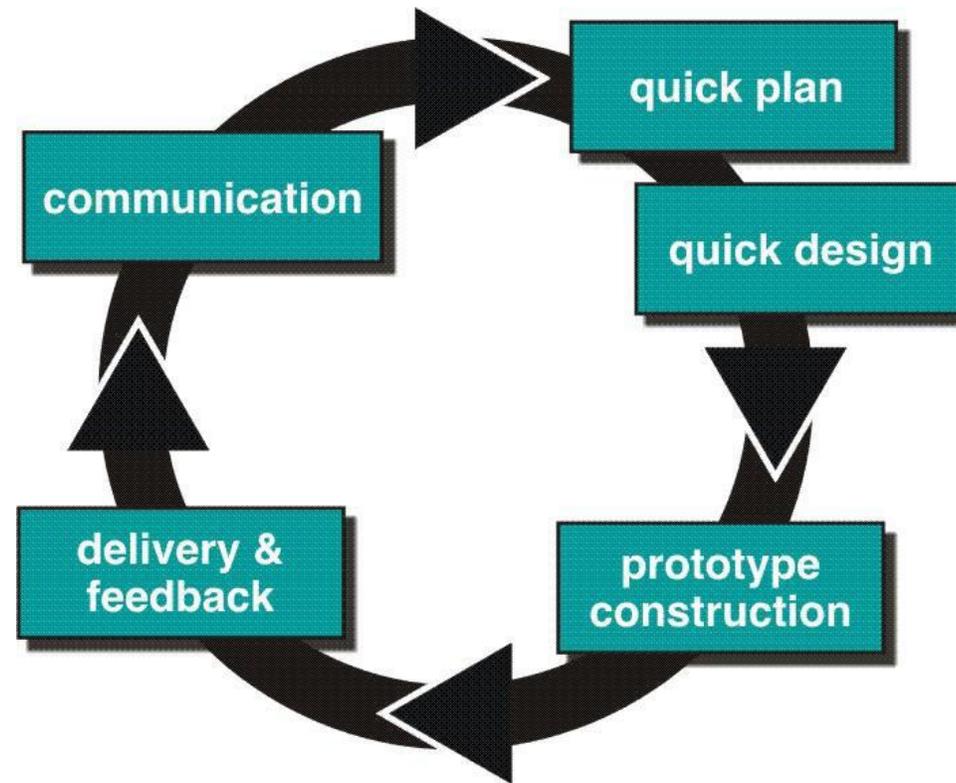
❖ Spiral Model

“Digunakan pada software engineering skala besar. Growing of software menjadi ide dari model proses ini”

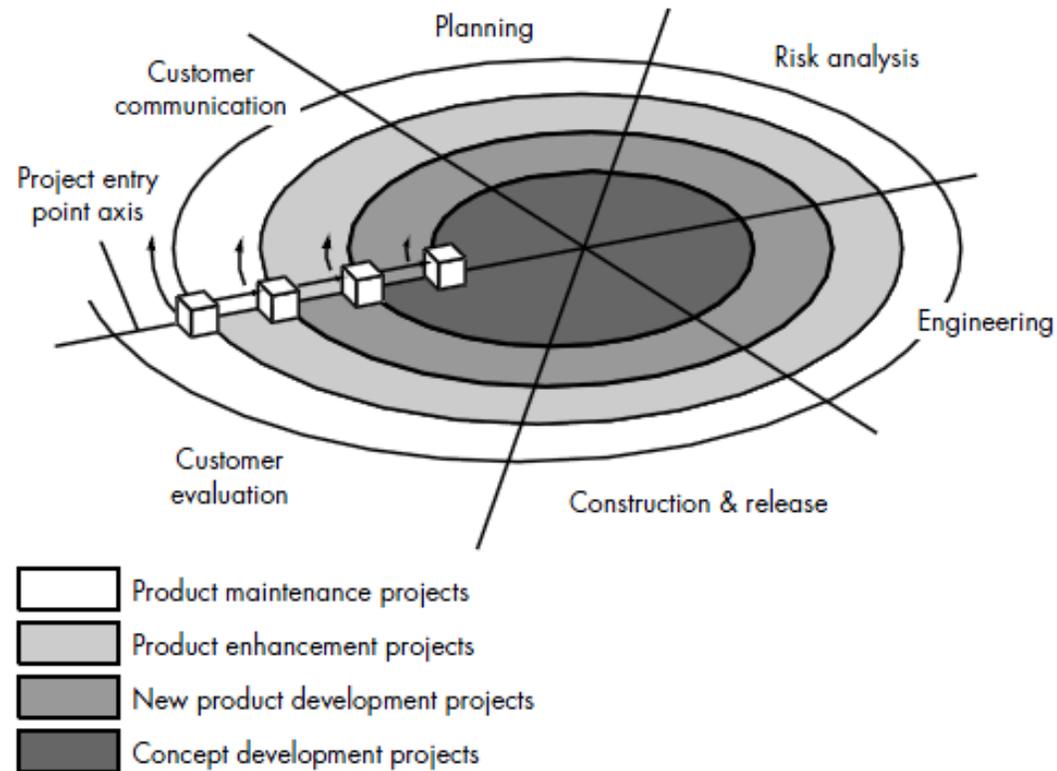
❖ Concurrent Model

“Model proses yang memungkinkan perubahan keadaan dari under development ke awaiting change (berhenti sementara) ketika terjadi perubahan kebutuhan”

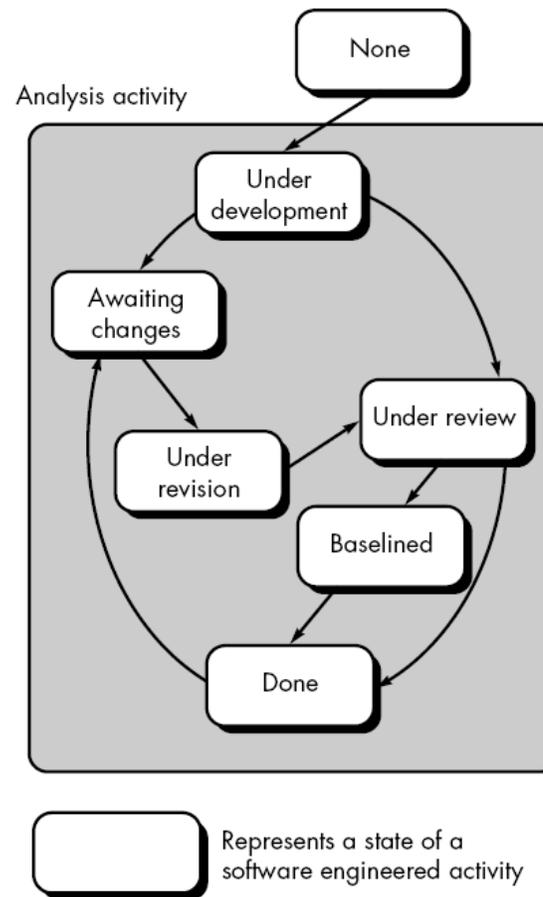
Prototyping



Spiral Model



Concurrent Model



SPECIALIZED PROCESS MODEL

❖ Component Based Development

“Model proses yang digunakan ketika konsep reuse menjadi tujuan utama dalam pembangunan perangkat lunak. Arsitektur perangkat lunak dibentuk dalam komponen-komponen”

❖ Formal Method

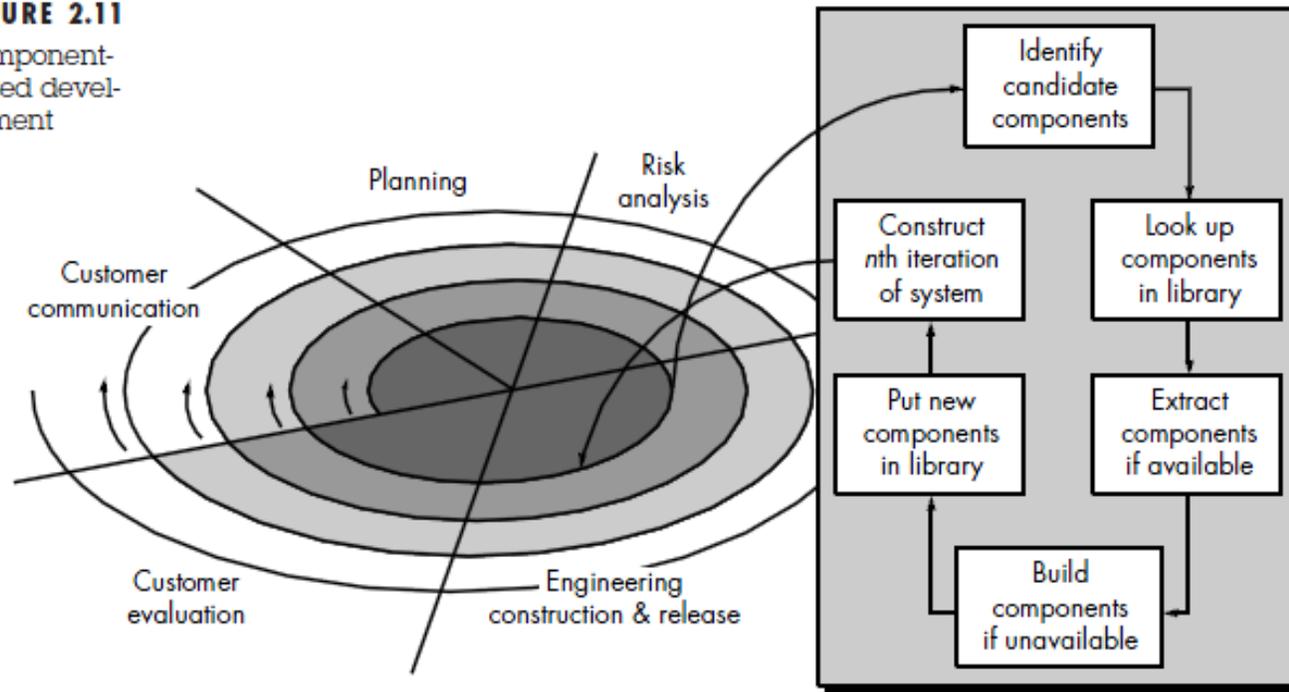
“Model proses yang menggunakan model matematika sebagai spesifikasi kebutuhan”

❖ Aspect Oriented Model

“Paradigma yang menekankan pada pendefinisian, penspesifikasian, dan pembangunan aspek (function, fitur, dan konten informasi)”

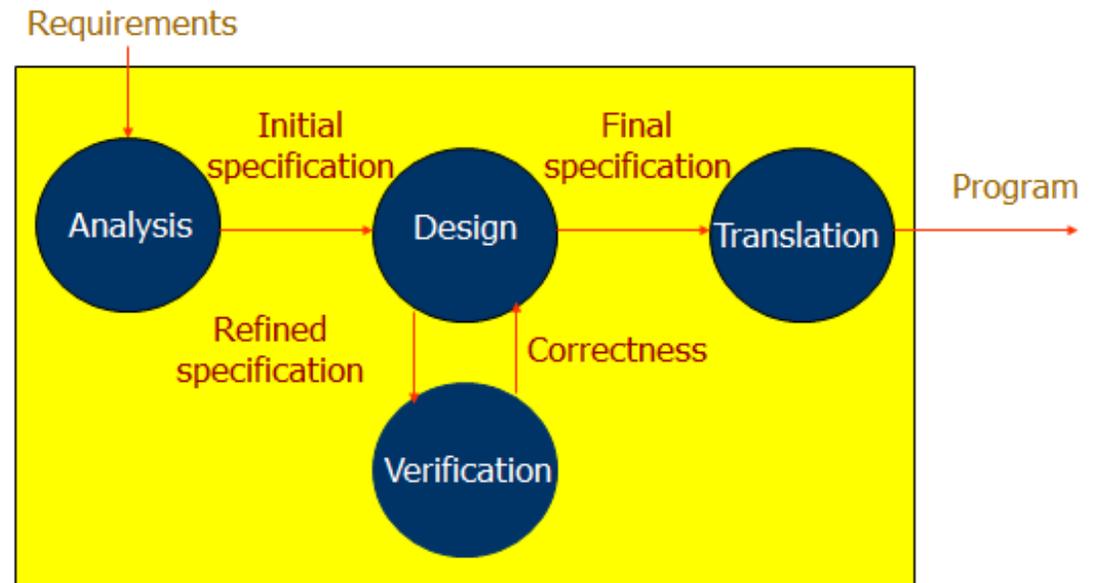
COMPONENT BASED DEVELOPMENT

FIGURE 2.11
Component-based development

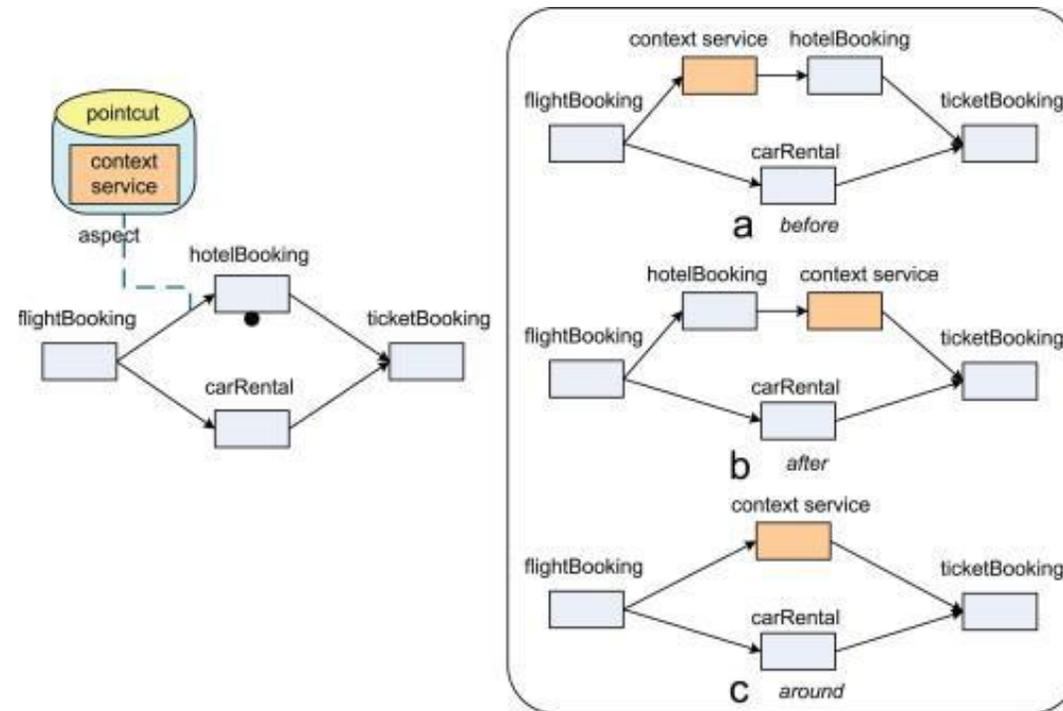


FORMAL METHOD

```
scheme DATABASE =  
  class  
    type  
      Person,  
      Database= Person-set  
    value  
      empty: Database,  
      register: Person x Database → Database,  
      check: Person x Database → Bool  
    axiom  
      empty ≡ {},  
      ∀p:Person, db: Database •  
        register(p,db) ≡ {p} ∪ db,  
      ∀p:Person, db: Database •  
        check(p,db) ≡ p ∈ db  
  end
```



CONTOH ASPECT ORIENTED MODEL



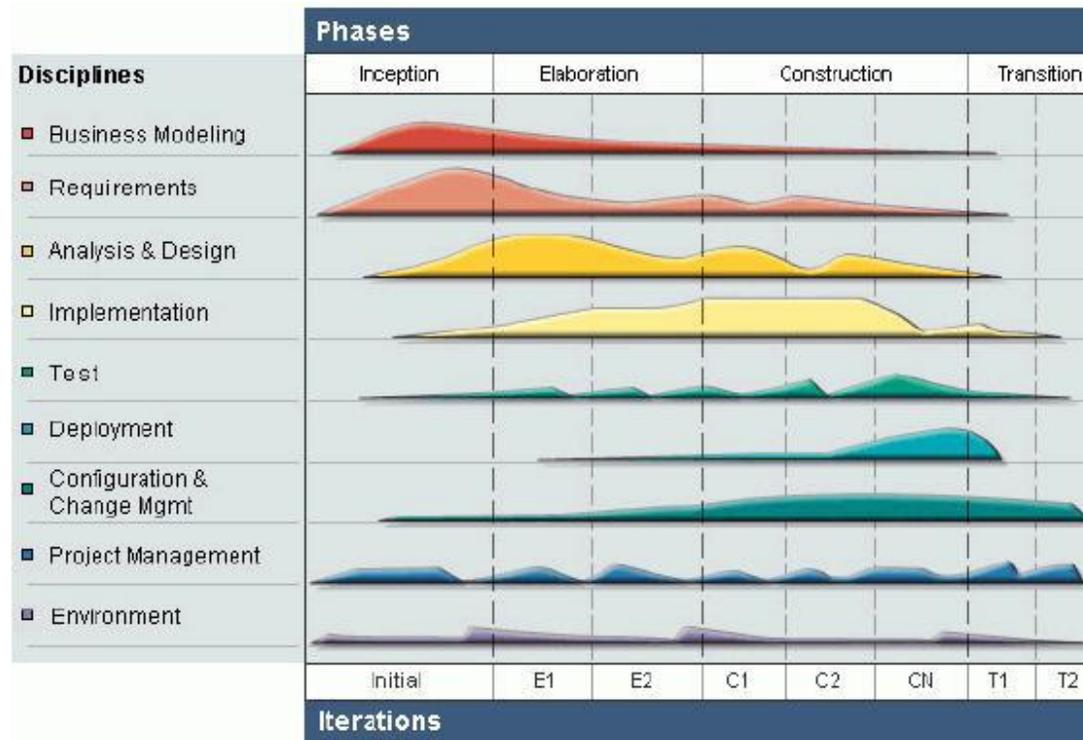
UNIFIED PROCESS

Model proses yang menggabungkan beberapa konsep unggulan dari model proses lainnya.

Penekanan pada model proses ini, yaitu:

- A. Komunikasi dengan customer secara intens (streamlined).
- B. Arsitektur yang reusable dan terbuka pada perubahan kebutuhan.
- C. Alur proses yang iterative atau incremental.

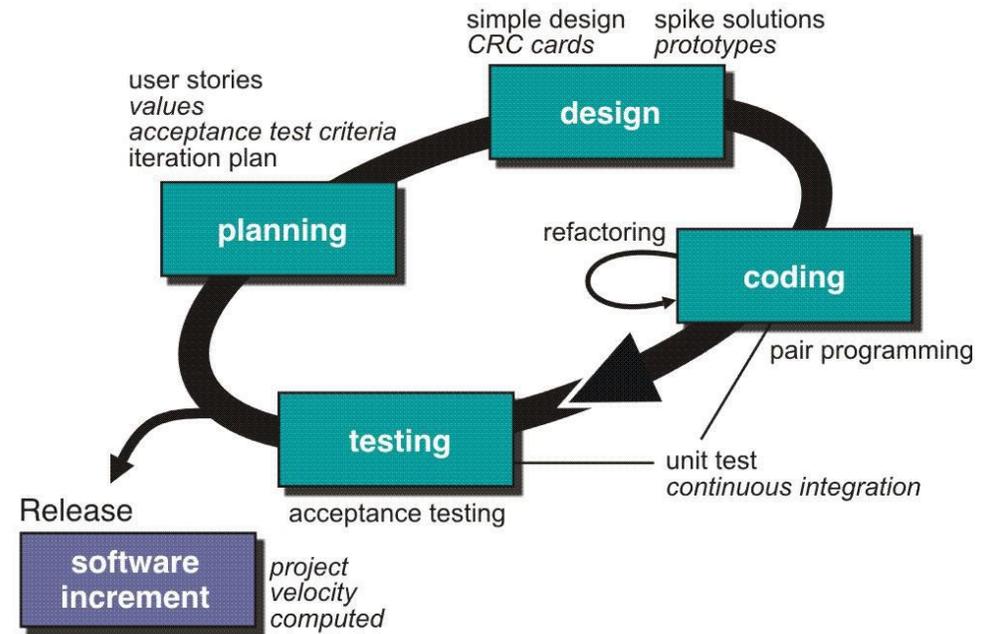
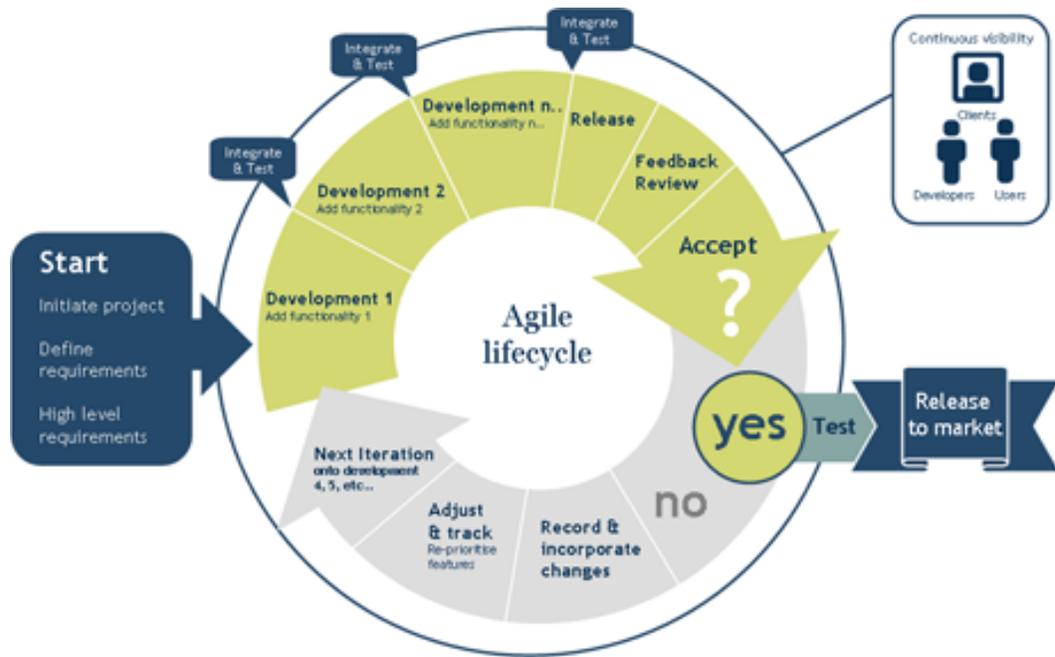
UNIFIED PROCESS



AGILE METHOD

“Model proses yang menekankan pada **fast delivery** dari setiap poin aktifitas dalam rangka **memperpendek** jangka waktu proyek pembangunan perangkat”

AGILE METHOD



Memilih Metodologi ???

- 1) Kejelasan kebutuhan pengguna
- 2) Penguasaan teknologi
- 3) Tingkat kerumitan sistem yang akan dibangun
- 4) Tingkat kehandalan system
- 5) Waktu pelaksanaan pengembangan