



REKAYASA PERANGKAT LUNAK I

Proses Pembangunan Perangkat Lunak

Disusun Oleh:

TIM MK RPL

Teknik Informatika UNIKOM

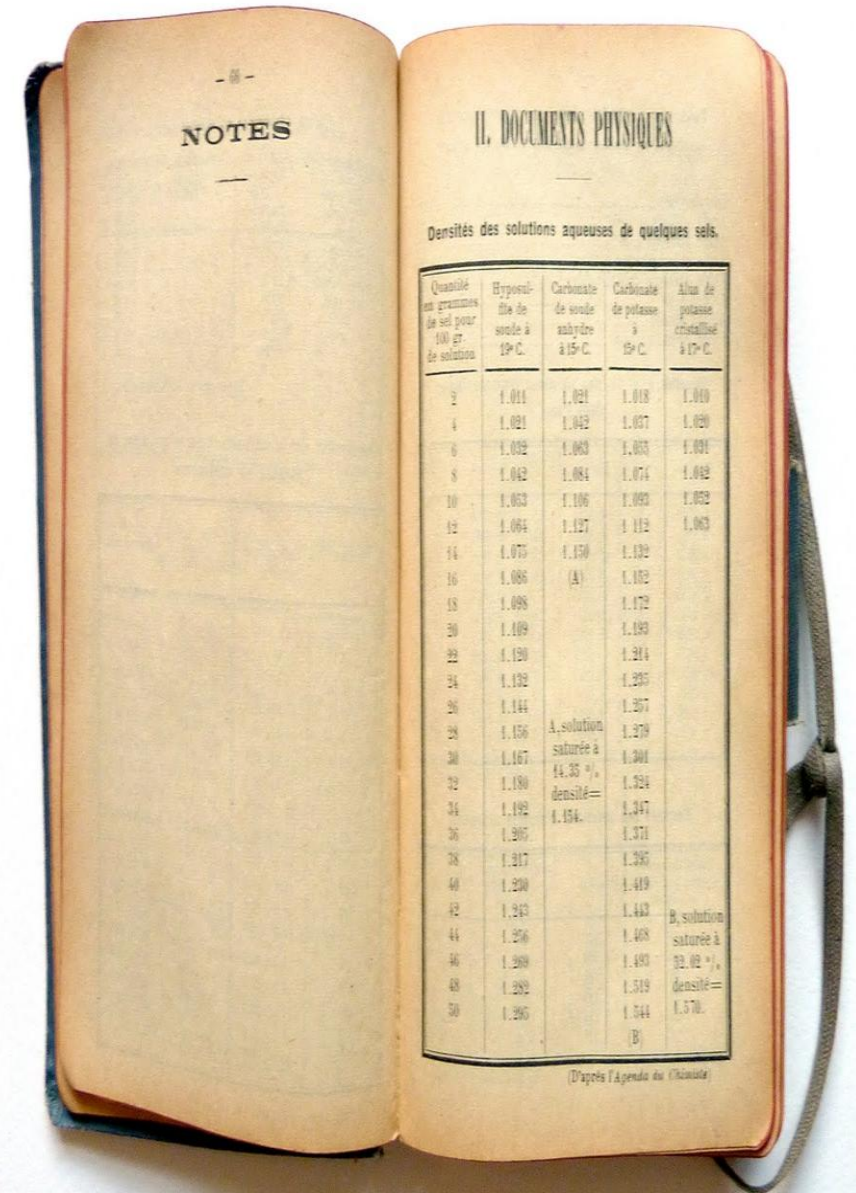


AGENDA PERKULIAHAN

➡ Software Development Life Cycle

➡ Generic Process Model

➡ Prescriptive Process Model



Software Development Life Cycle

PENGERTIAN SOFTWARE DEVELOPMENT LIFE CYCLE

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

AKTIFITAS PEMBANGUNAN PERANGKAT LUNAK

- ➊ Gathering Requirements
- ➋ Team Management (incl. Analysis)
- ➌ Software Design
- ➍ Coding
- ➎ Testing
- ➏ Documentation
- ➐ Software Maintenance



PERTANYAAN SEPUTAR SDLC

❶ Aktivitas mana yang harus dipilih dalam pembangunan perangkat lunak?

SDLC berisi proses yang bukan untuk dipilih tapi untuk dilakukan secara keseluruhan.

❷ Hubungan antar aktivitas dalam SDLC?

Hasil dari kegiatan sebelumnya mempunyai pengaruh yang sangat besar untuk kegiatan berikutnya.

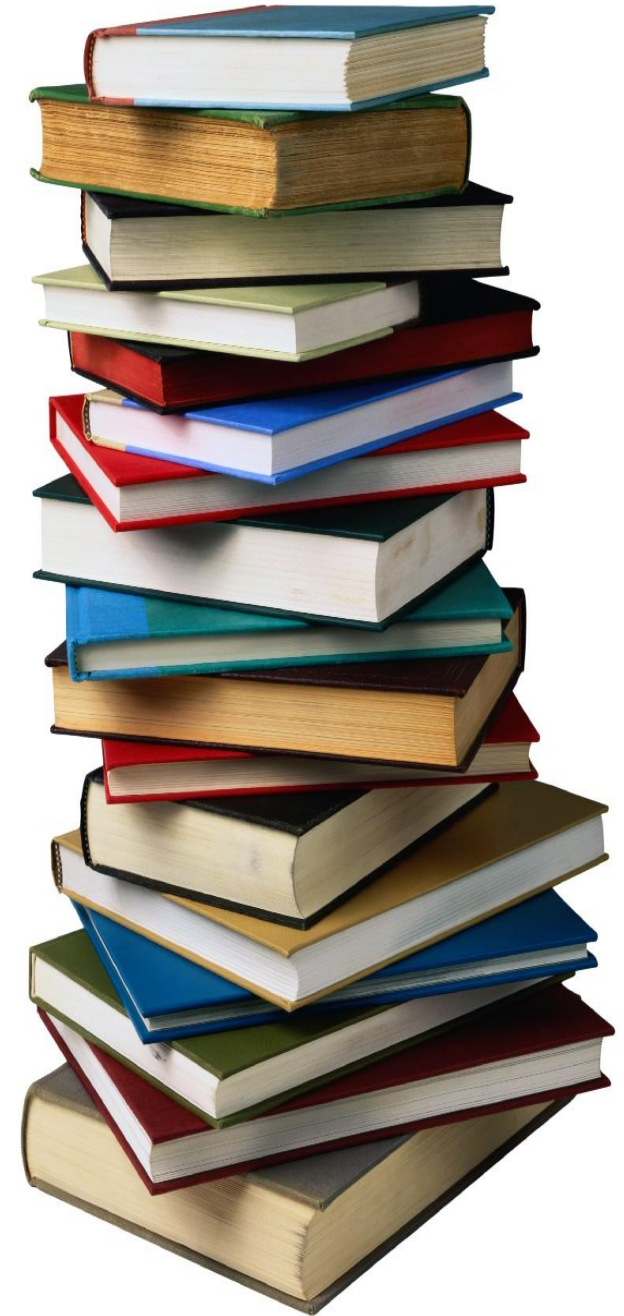
❸ Bagaimana cara menjadwalkan kegiatan pada SDLC?

Penjadwalan kegiatan SDLC dibahas dalam ilmu Manajemen Proyek (Time Management) dan tidak dibahas pada RPL.

Generic Process Model

KONTEN MATERI

- ➡ Definisi Generic Process Model
- ➡ Process Flow
- ➡ Linear Process Flow
- ➡ Iterative Process Flow
- ➡ Evolutionary Process Flow



DEFINISI GENERIC PROCESS MODEL

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

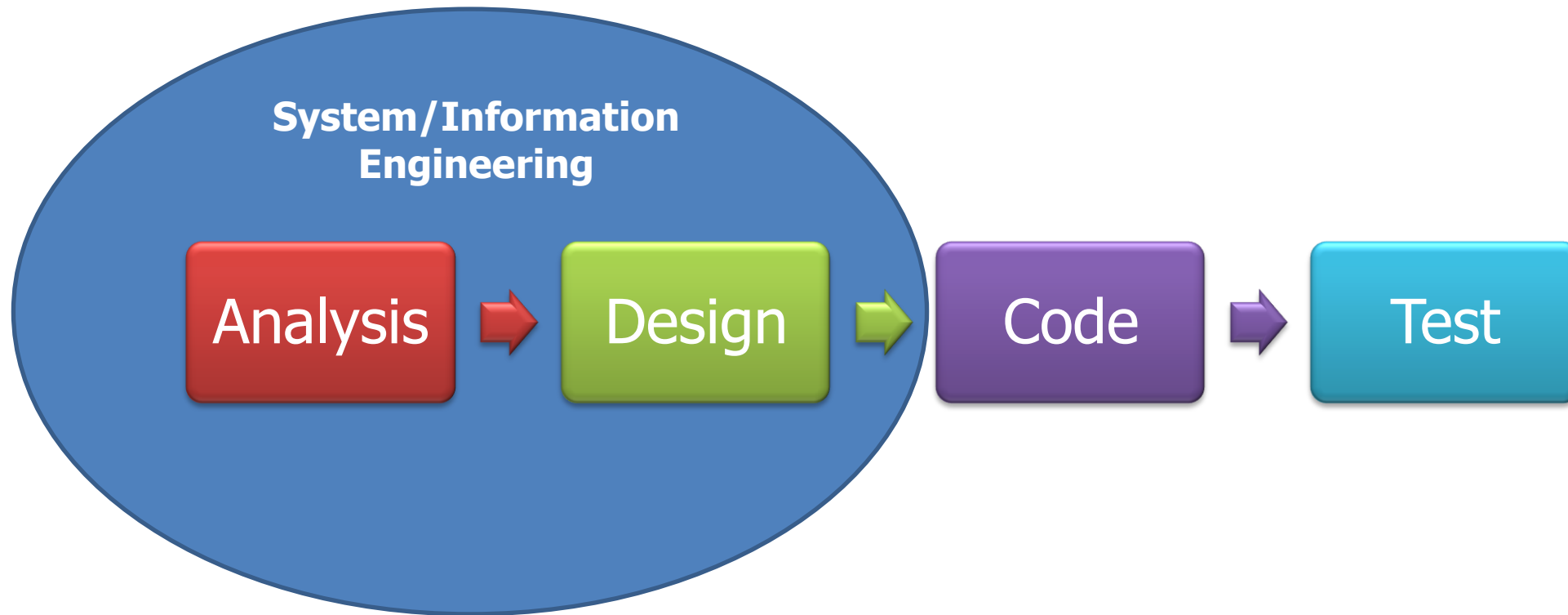
- ➊ Communication
- ➋ Planning
- ➌ Modeling
- ➍ Construction
- ➎ Deployment.

PROCESS FLOW

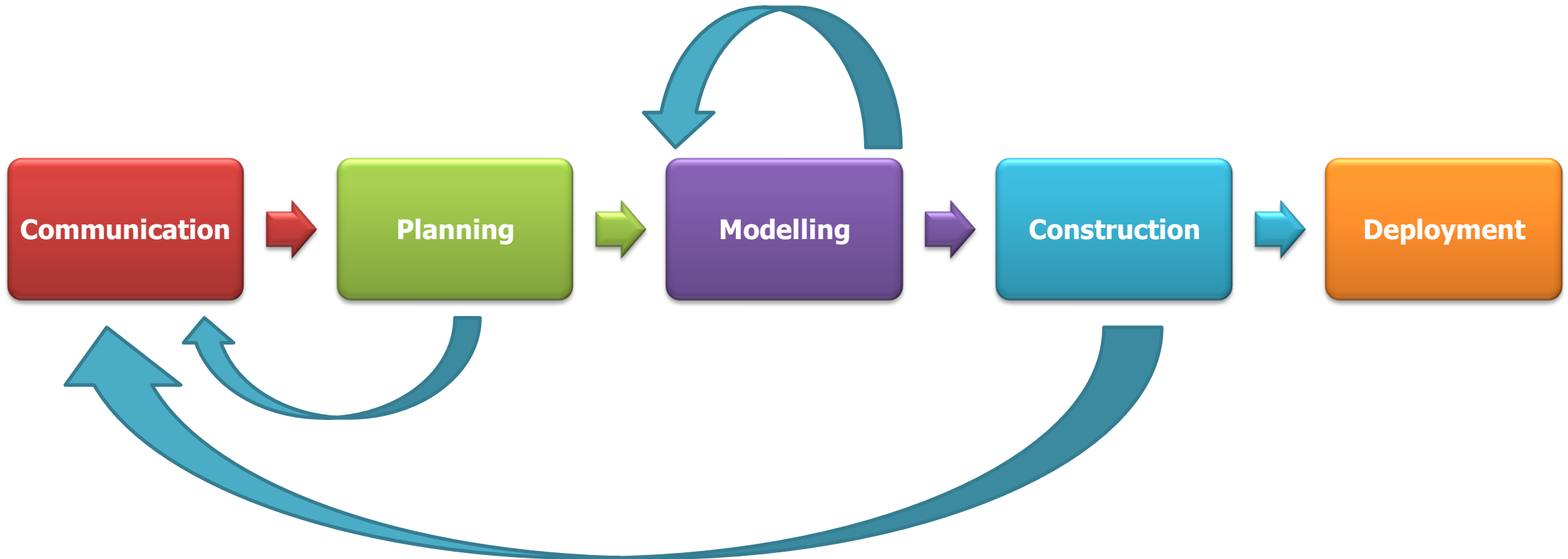
- ❶ **Linear Process Model**
- ❷ **Iterative Process Model**
- ❸ **Evolutionary Process Model**



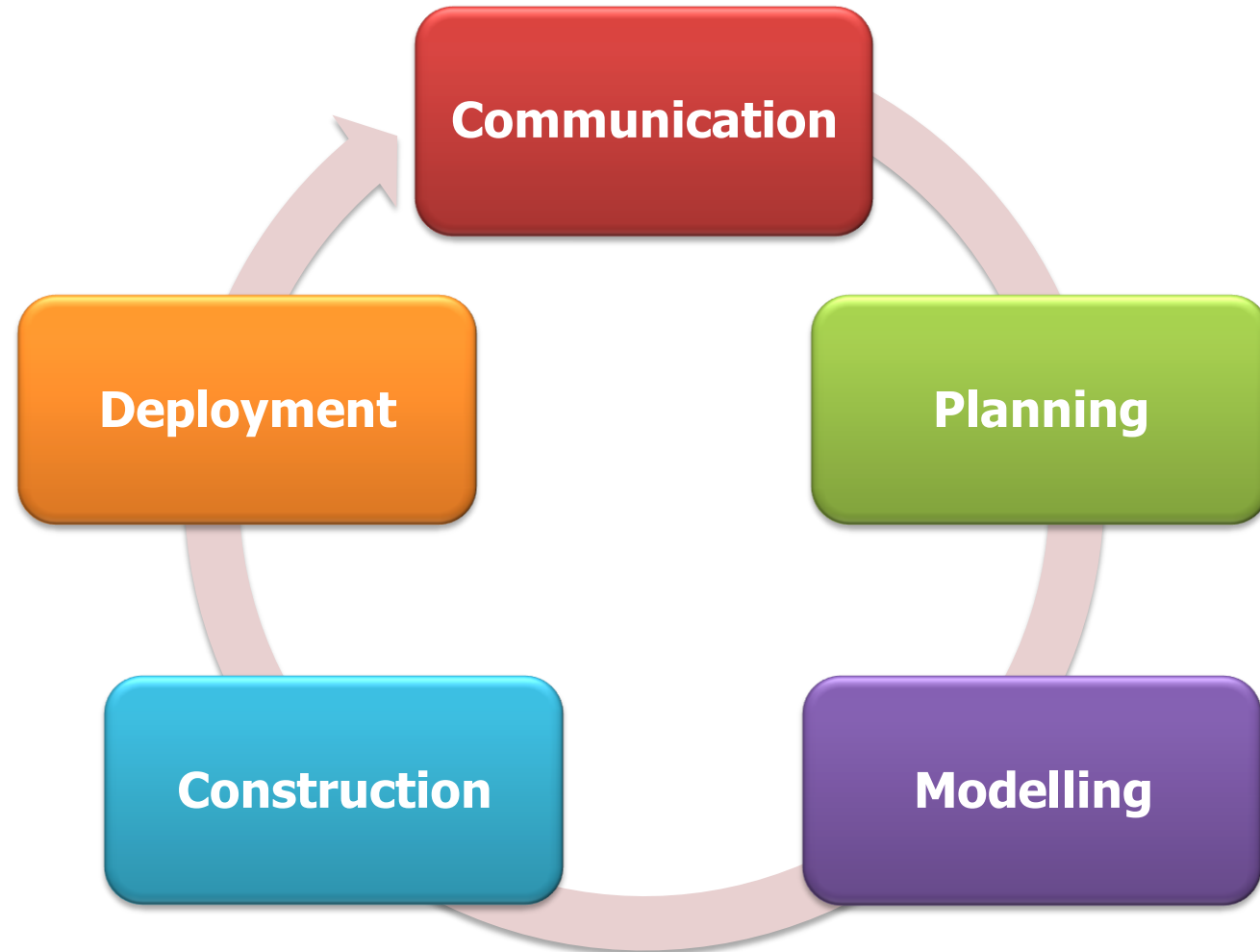
LINEAR PROCESS FLOW



ITERATIVE PROCESS FLOW



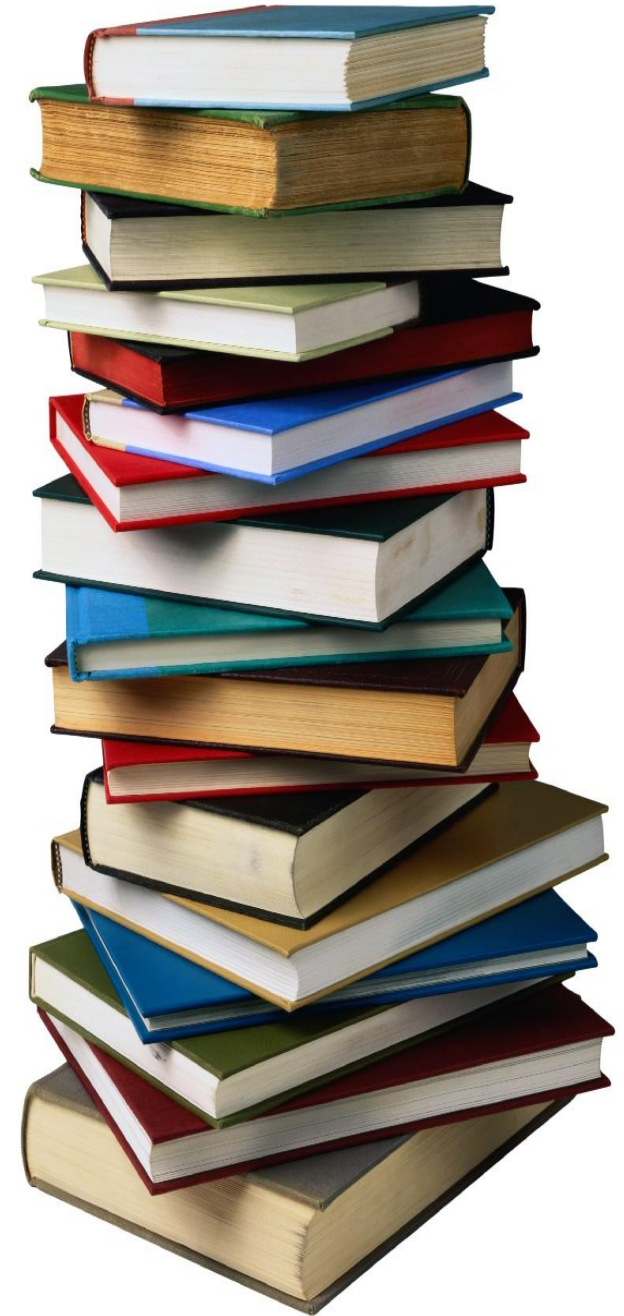
EVOLUTIONARY PROCESS FLOW



Prescriptive Process Model

KONTEN MATERI

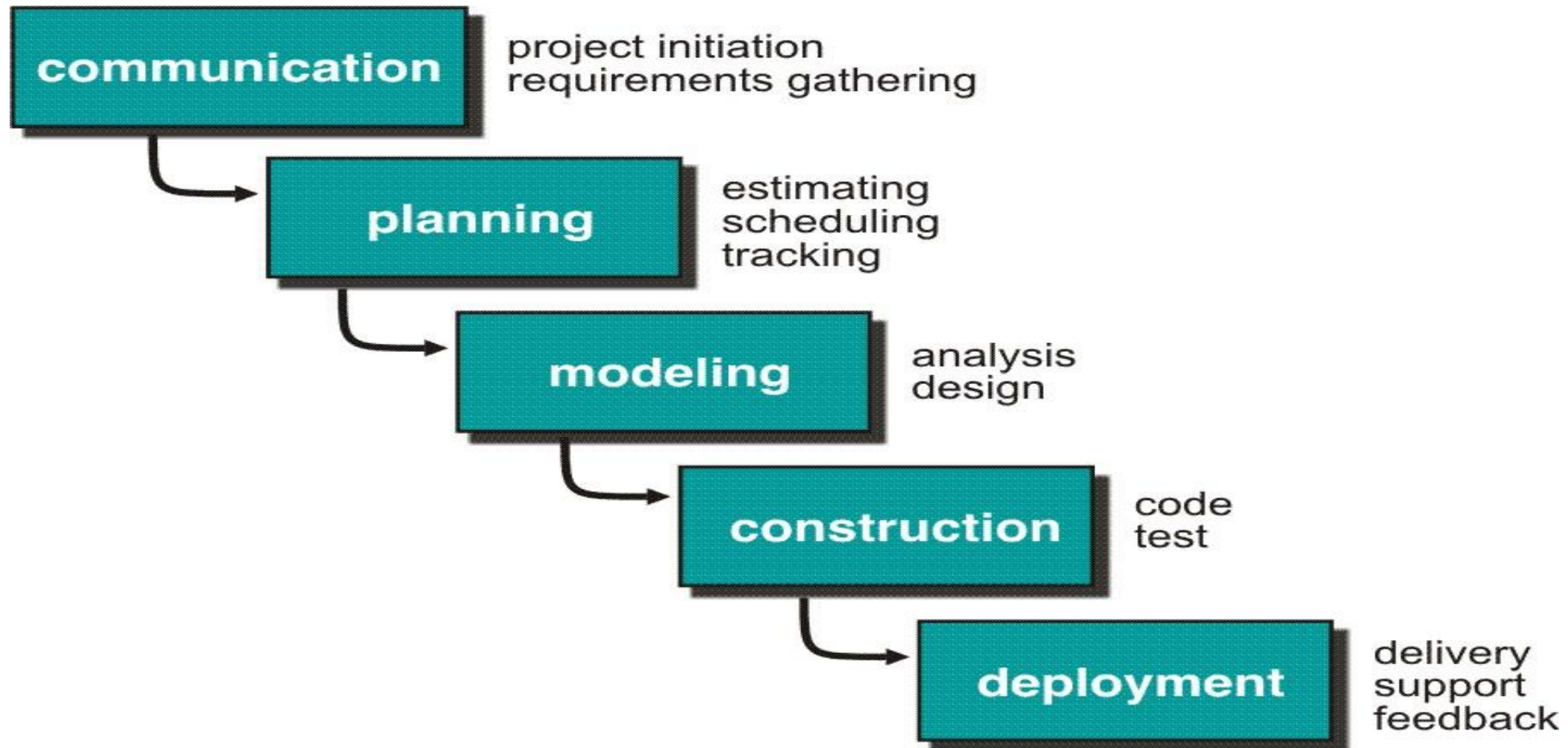
- ⇒ Waterfall Model
- ⇒ V Model
- ⇒ Incremental Process Model
- ⇒ Evolutionary Process Model
- ⇒ Specialized Process Model
- ⇒ Unified Process
- ⇒ Agile Methods (example: XP)



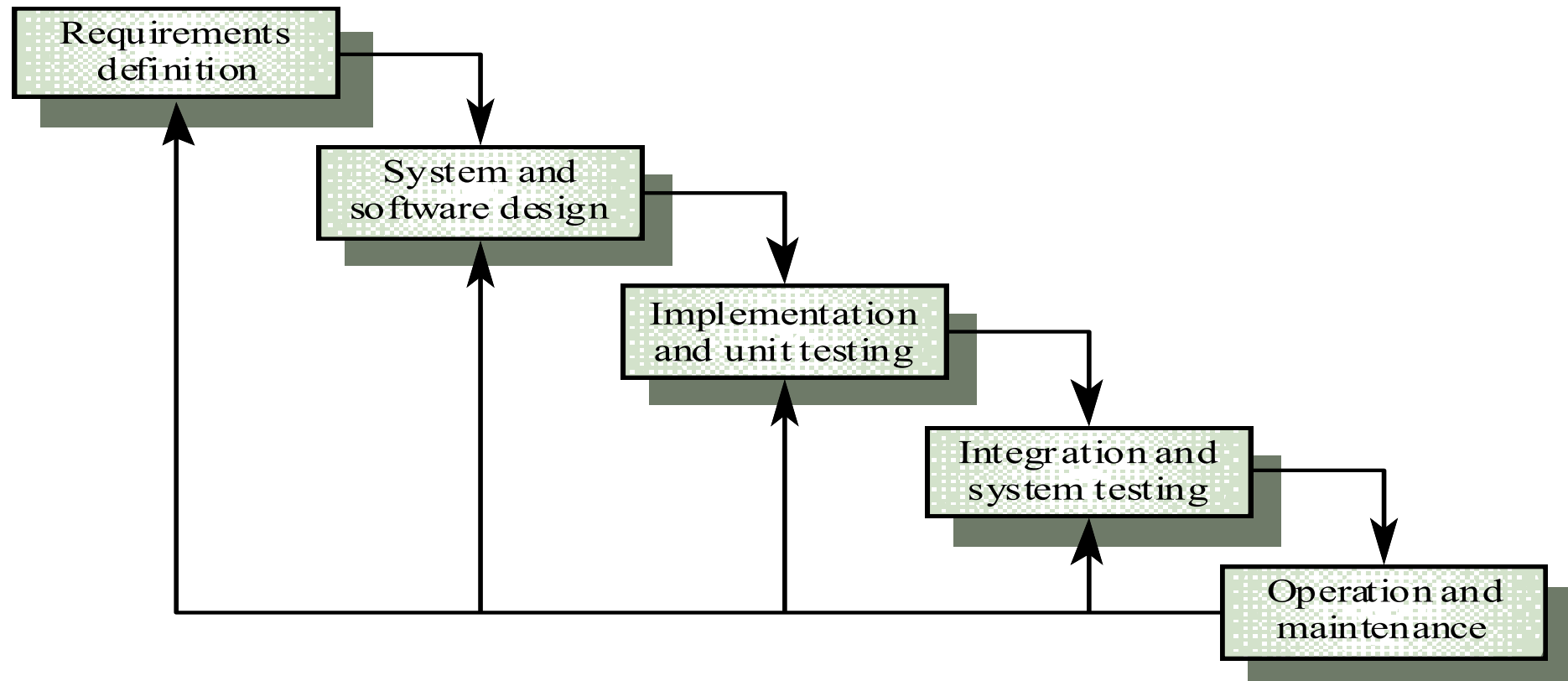
KARAKTERISTIK WATERFALL MODEL

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

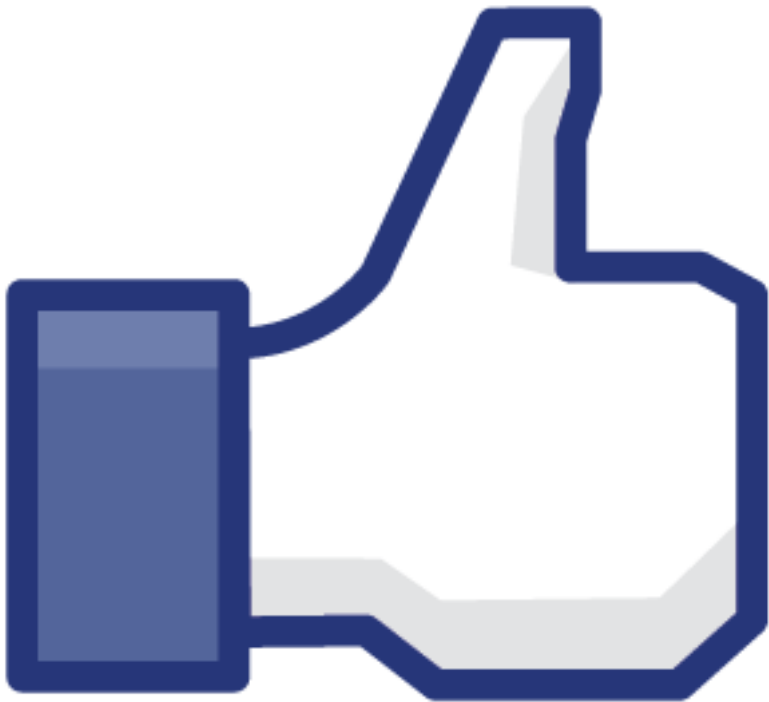
GAMBAR WATERFALL MODEL (ROGER S, PRESSMAN)



GAMBAR WATERFALL MODEL (IAN SOMMERVILLE)

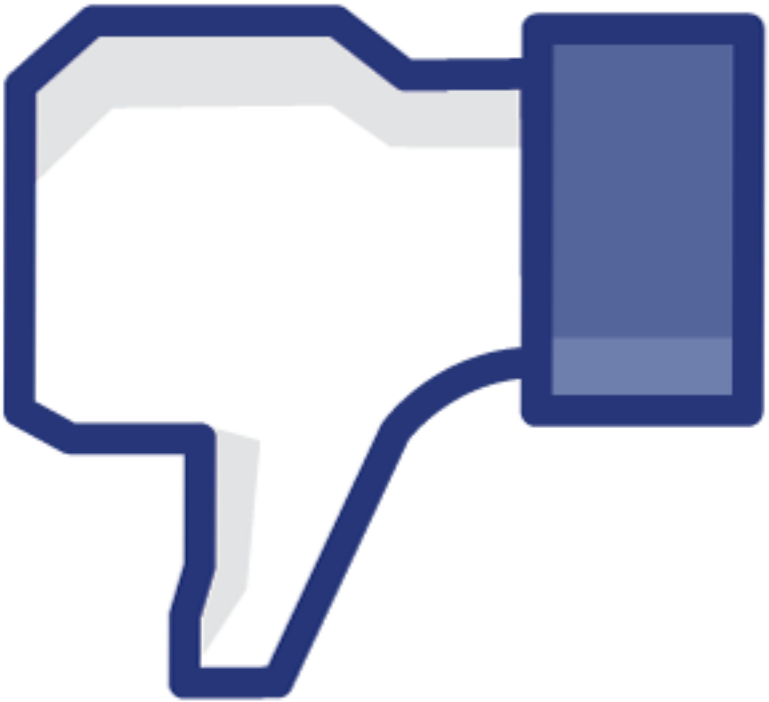


KELEBIHAN WATERFALL MODEL



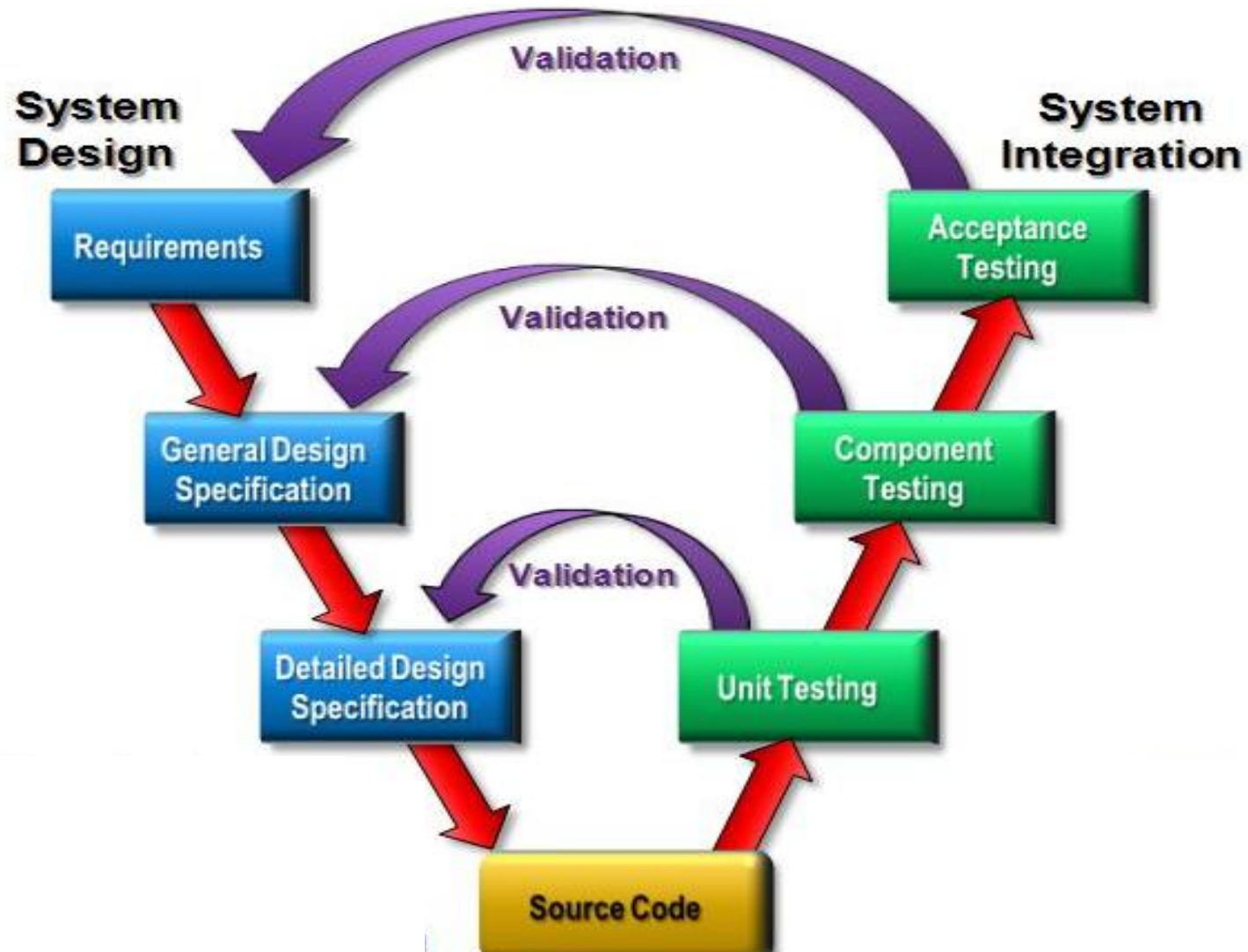
- ➊ Menghasilkan mature process pada setiap tahapnya.
- ➋ Mudah untuk diaplikasikan pada sebuah proyek.
- ➌ Menghasilkan sistem yang terstruktur dengan baik.
- ➍ Memiliki tingkat visibilitas yang tinggi (setiap tahap mempunyai dokumen yang jelas).

KEKURANGAN WATERFALL MODEL



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

GAMBAR V MODEL



INCREMENTAL PROCESS MODEL

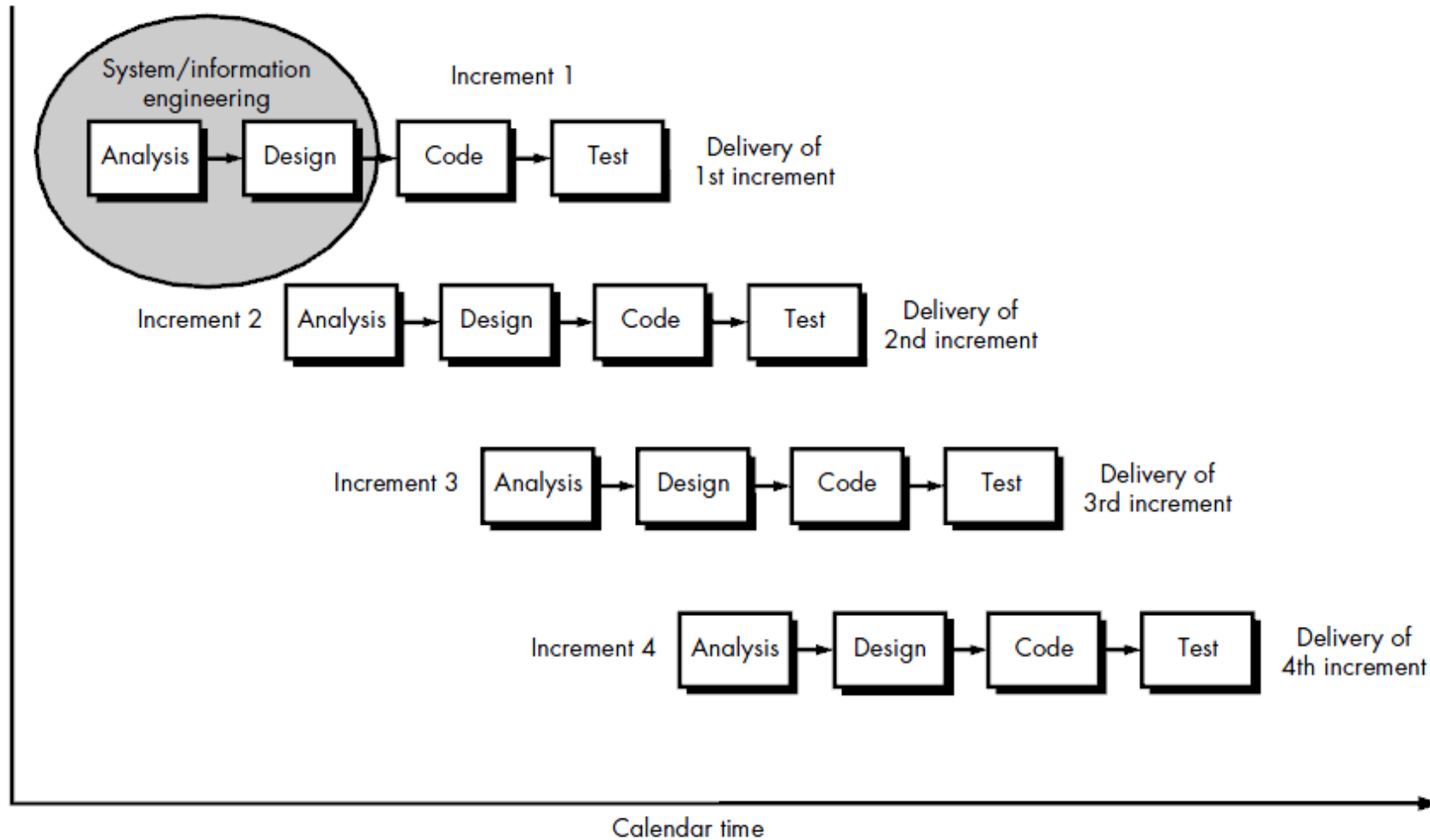
① Incremental Model

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

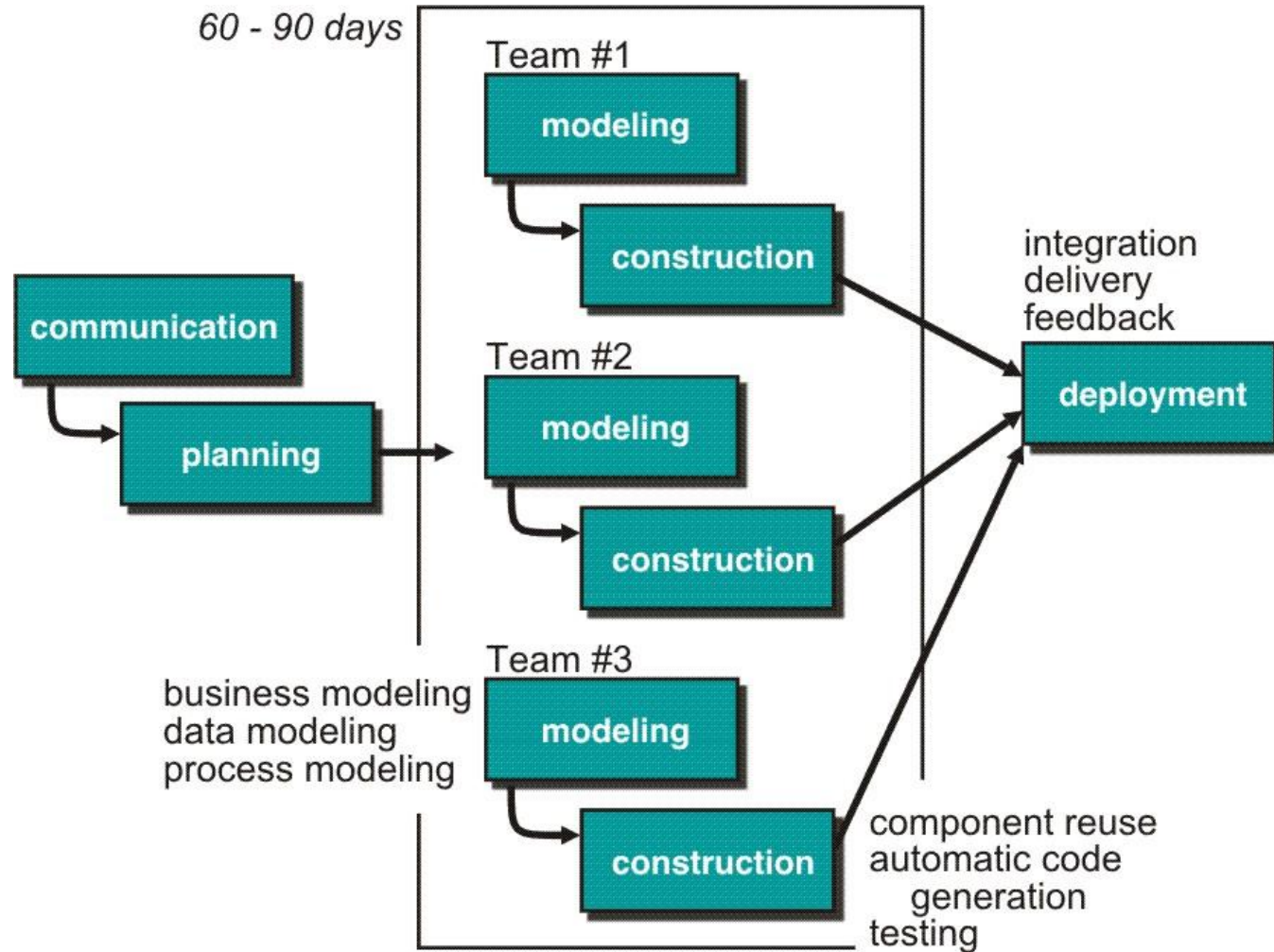
② Rapid Application Development (RAD)

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

GAMBAR INCREMENTAL MODEL



GAMBAR RAPID APPLICATION MODEL



EVOLUTIONARY PROCESS MODEL

① Prototyping

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

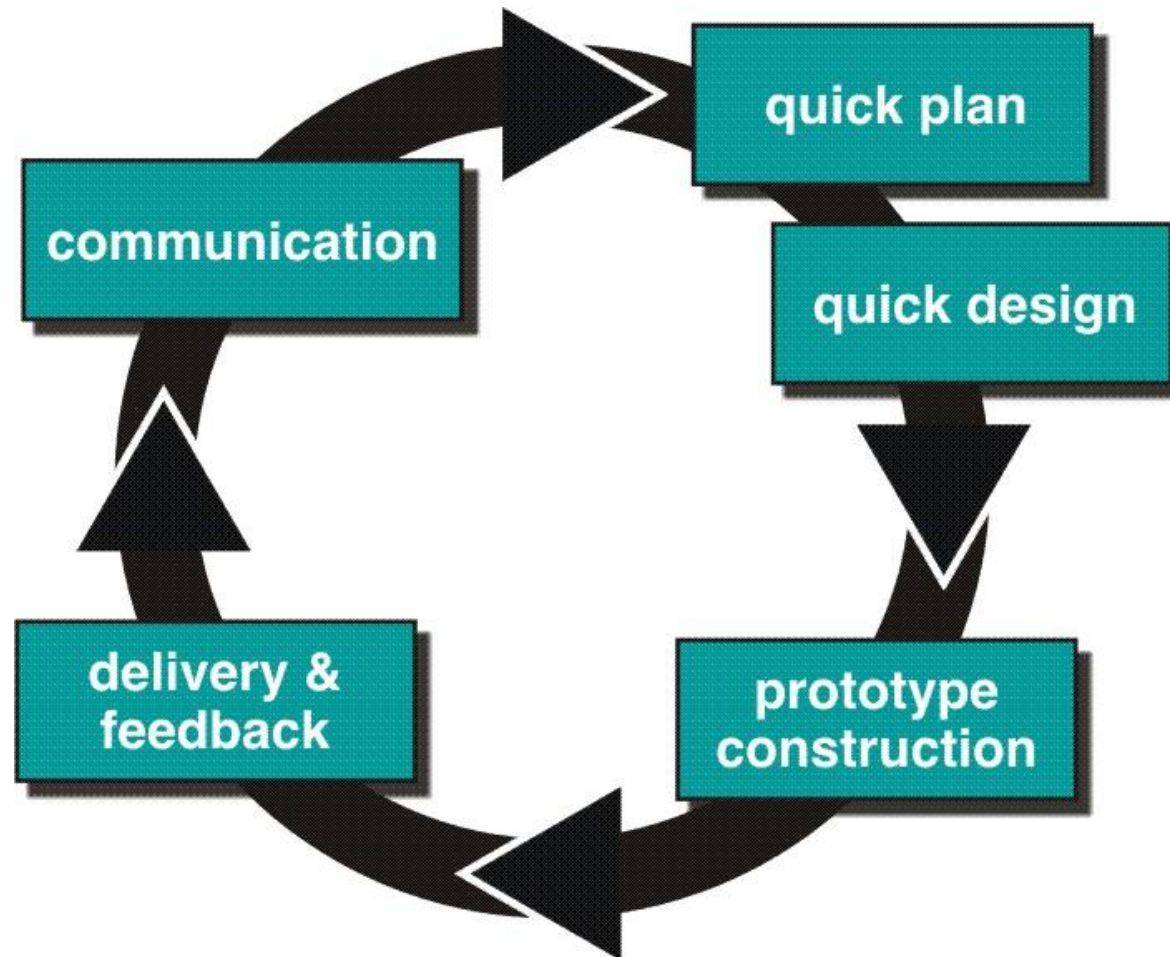
② Spiral Model

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

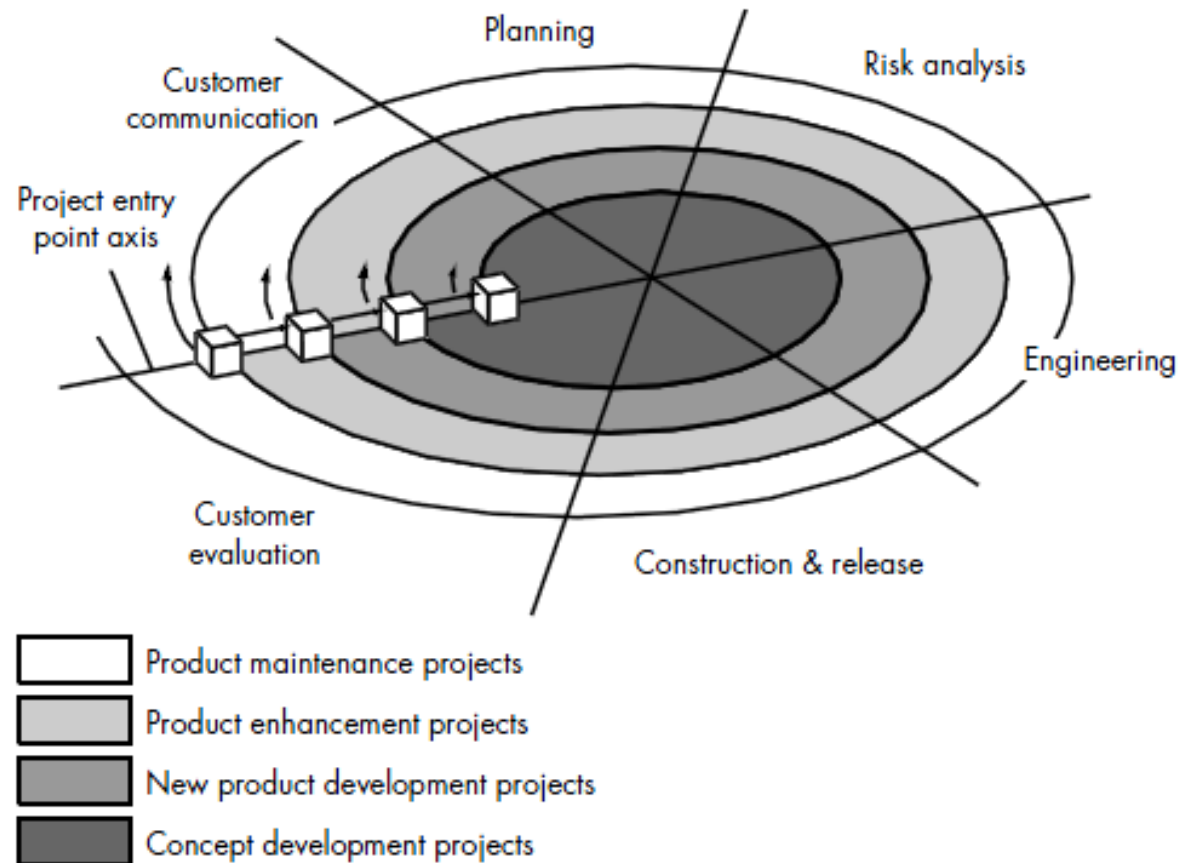
③ Concurrent Model

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

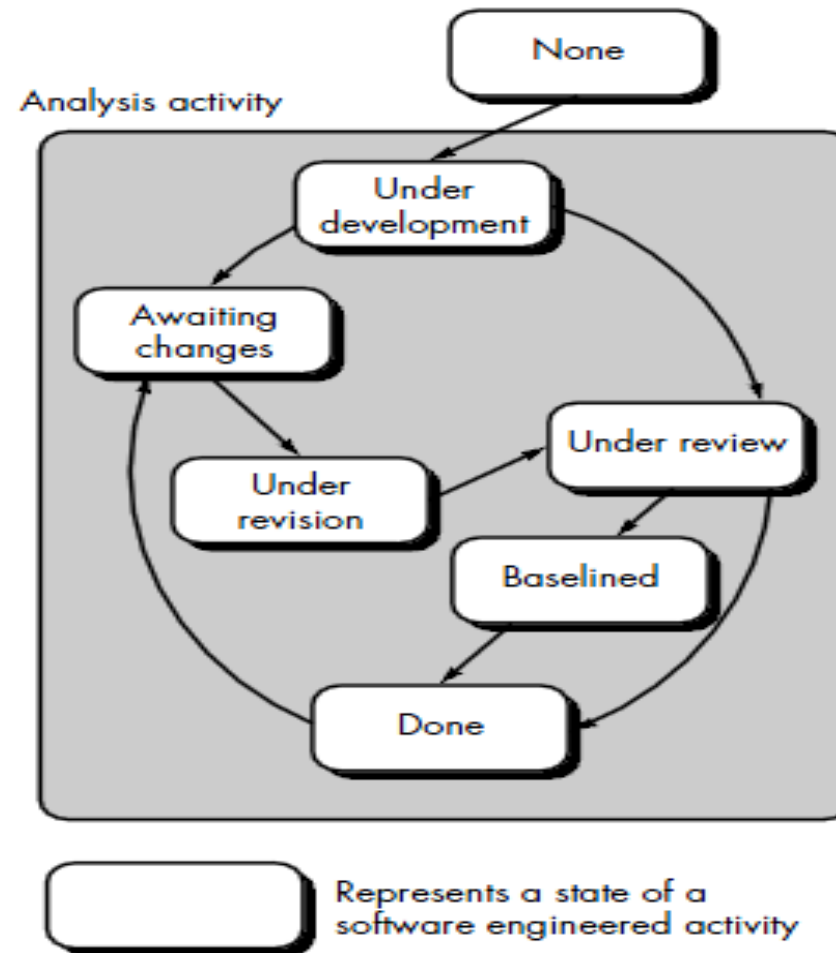
GAMBAR PROTOTYPING MODEL



GAMBAR SPIRAL MODEL



GAMBAR CONCURRENT PROCESS MODEL



SPECIALIZED PROCESS MODEL

① Component Based Development

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

② Formal Method

“Model proses yang menggunakan model matematika sebagai spesifikasi kebutuhan”

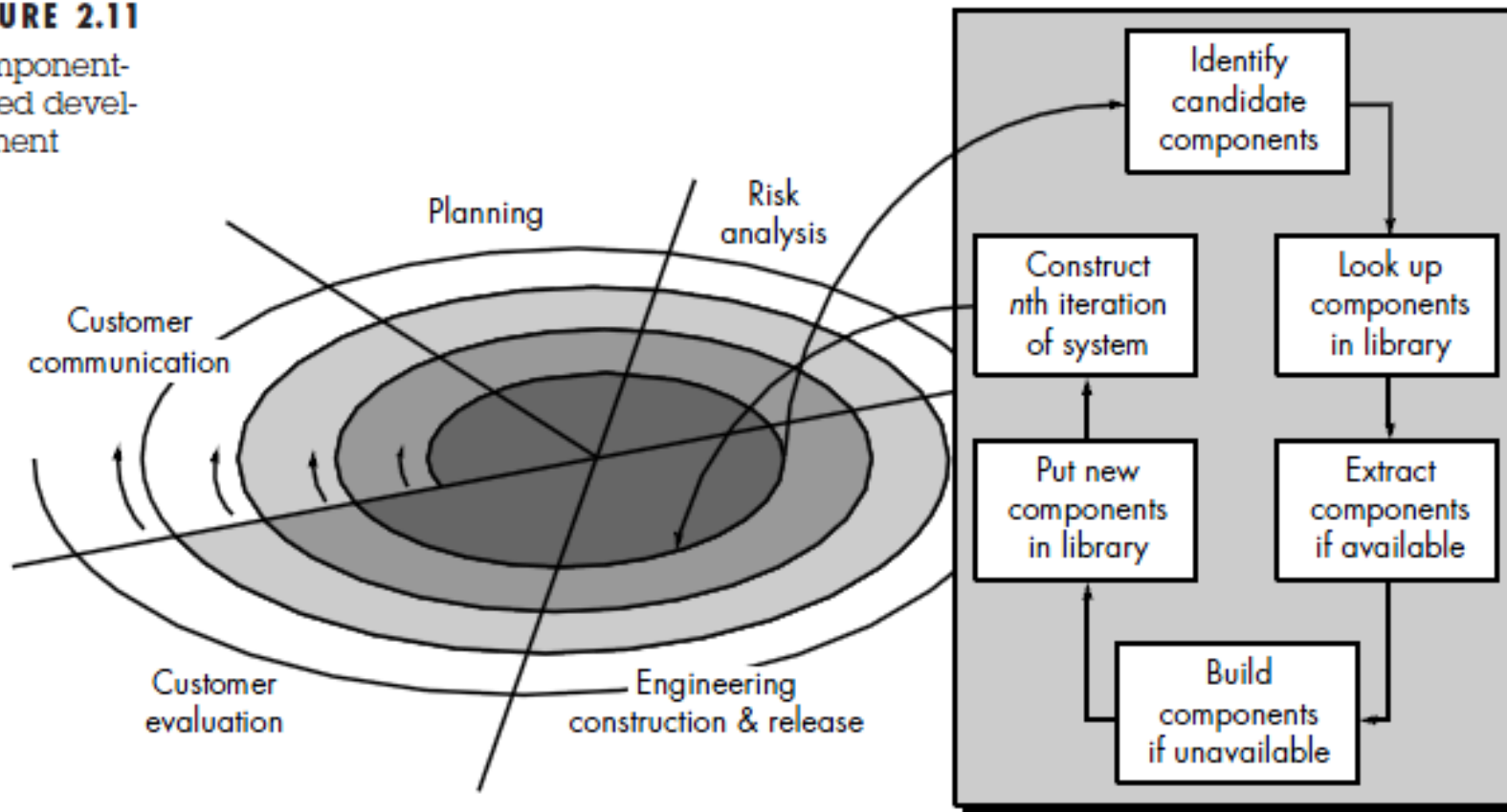
③ Aspect Oriented Model

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

GAMBAR COMPONENT BASED DEVELOPMENT

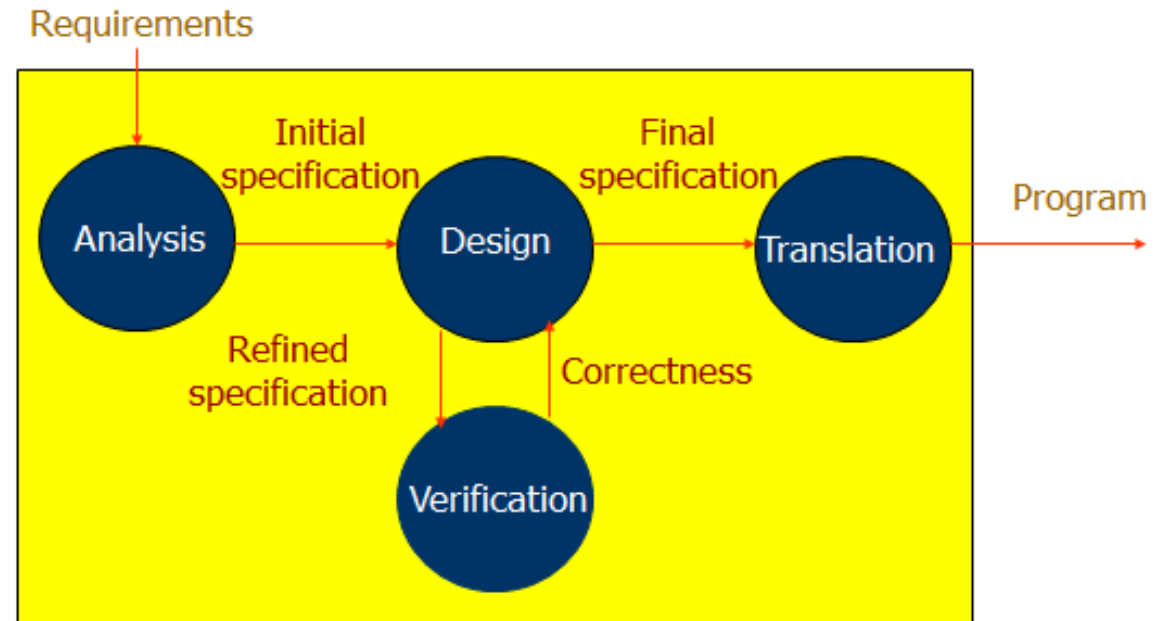
FIGURE 2.11

Component-based development

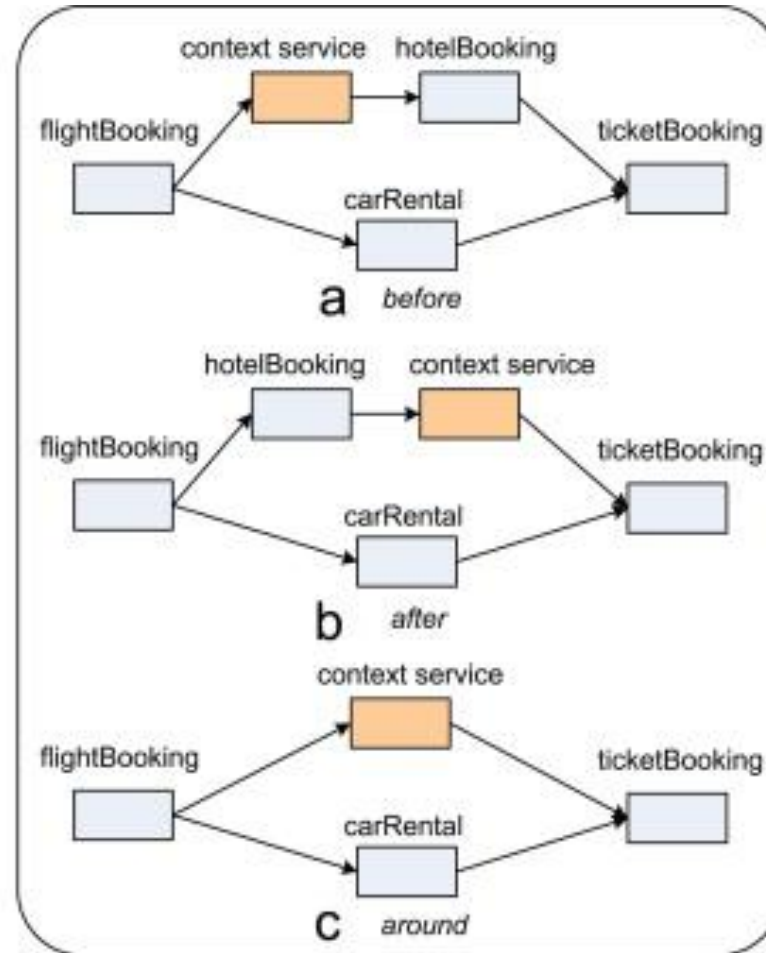
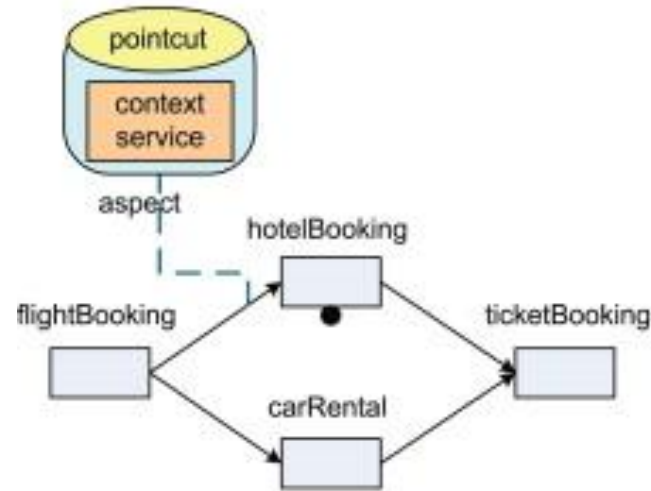


GAMBAR FORMAL METHOD

```
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
```



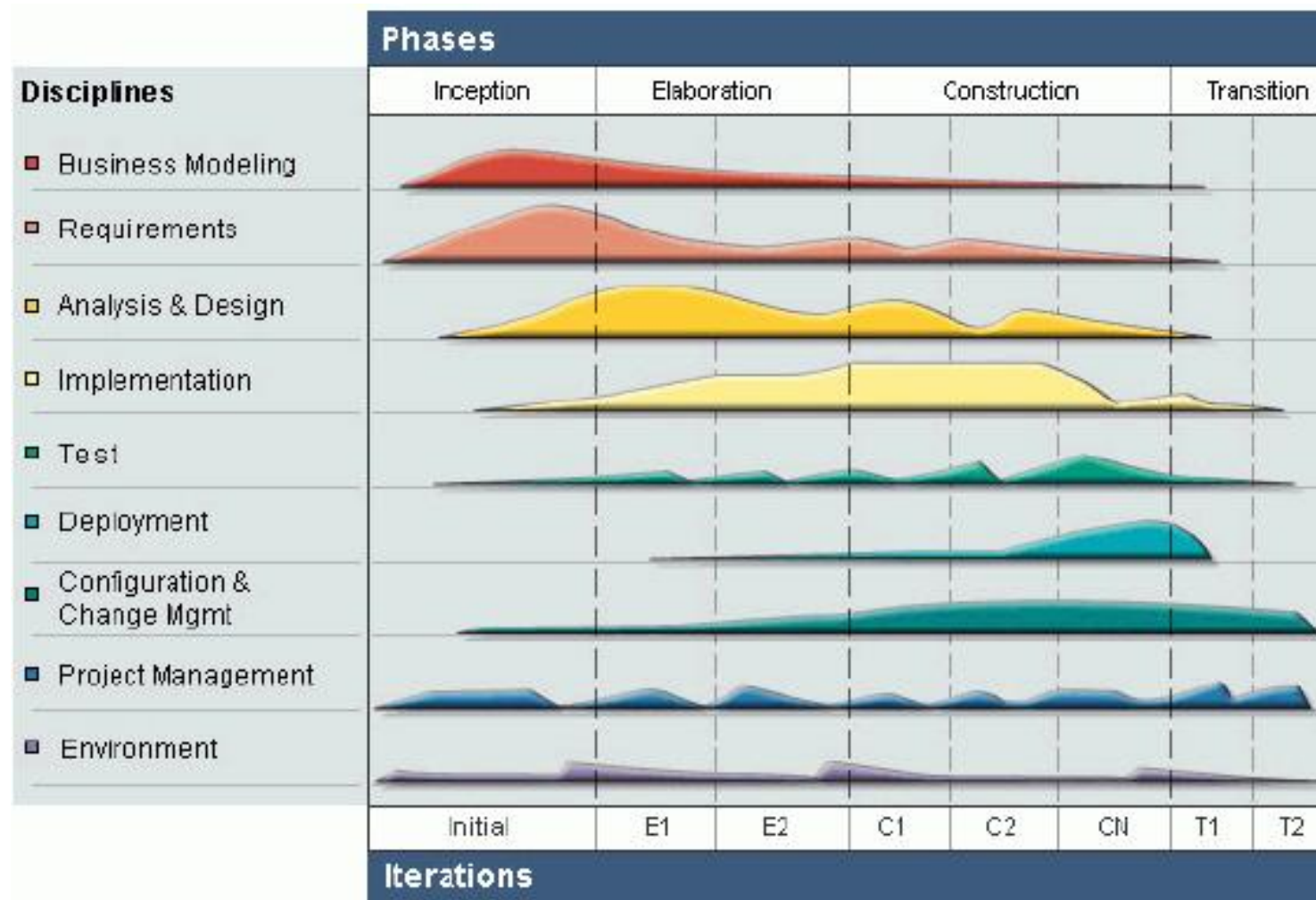
GAMBAR CONTOH ASPECT ORIENTED MODEL



PENGERTIAN UNIFIED PROCESS

- ❶ Model proses yang menggabungkan beberapa konsep unggulan dari model proses lainnya.
- ❷ 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.

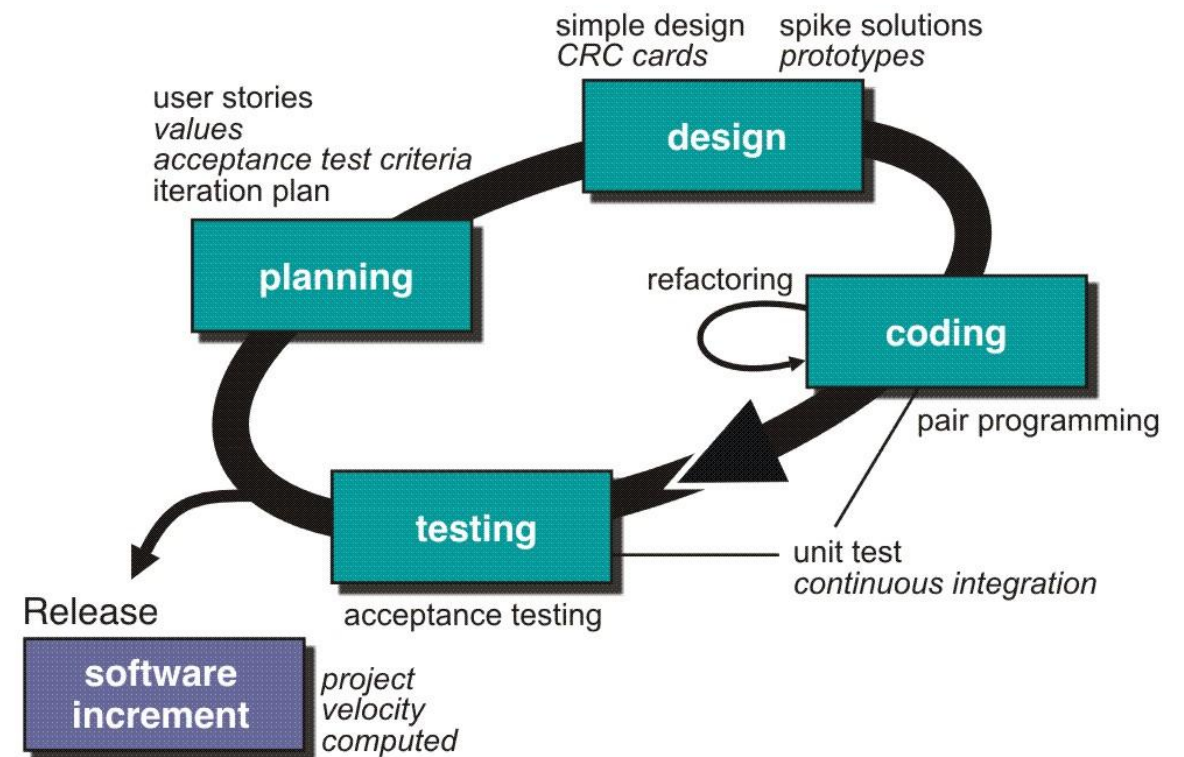
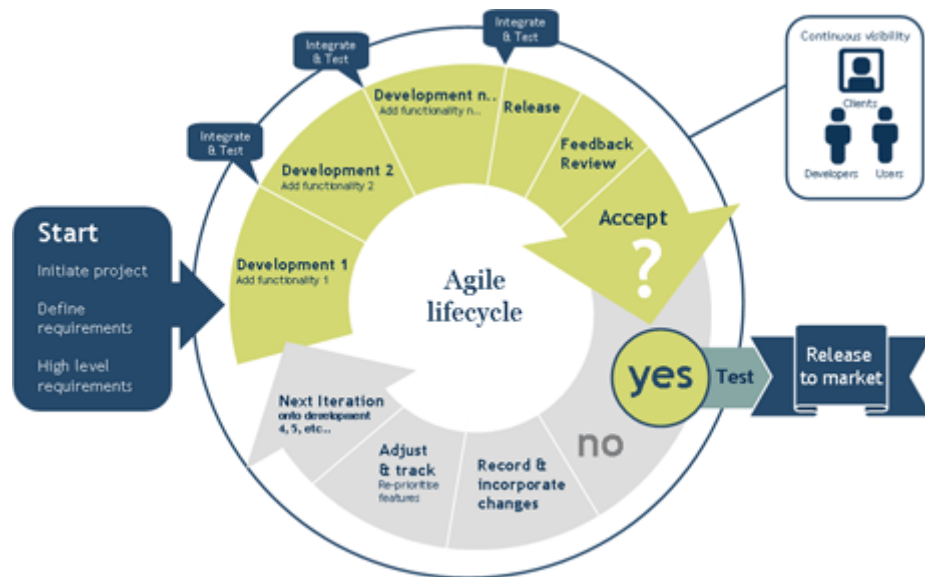
GAMBAR UNIFIED PROCESS



PENGERTIAN AGILE METHOD

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

GAMBAR AGILE METHOD



Terima Kasih