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

1. Gathering Requirements
2. Team Management (incl. Analysis)
3. Software Design
4. Coding
5. Testing
6. Documentation
7. 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

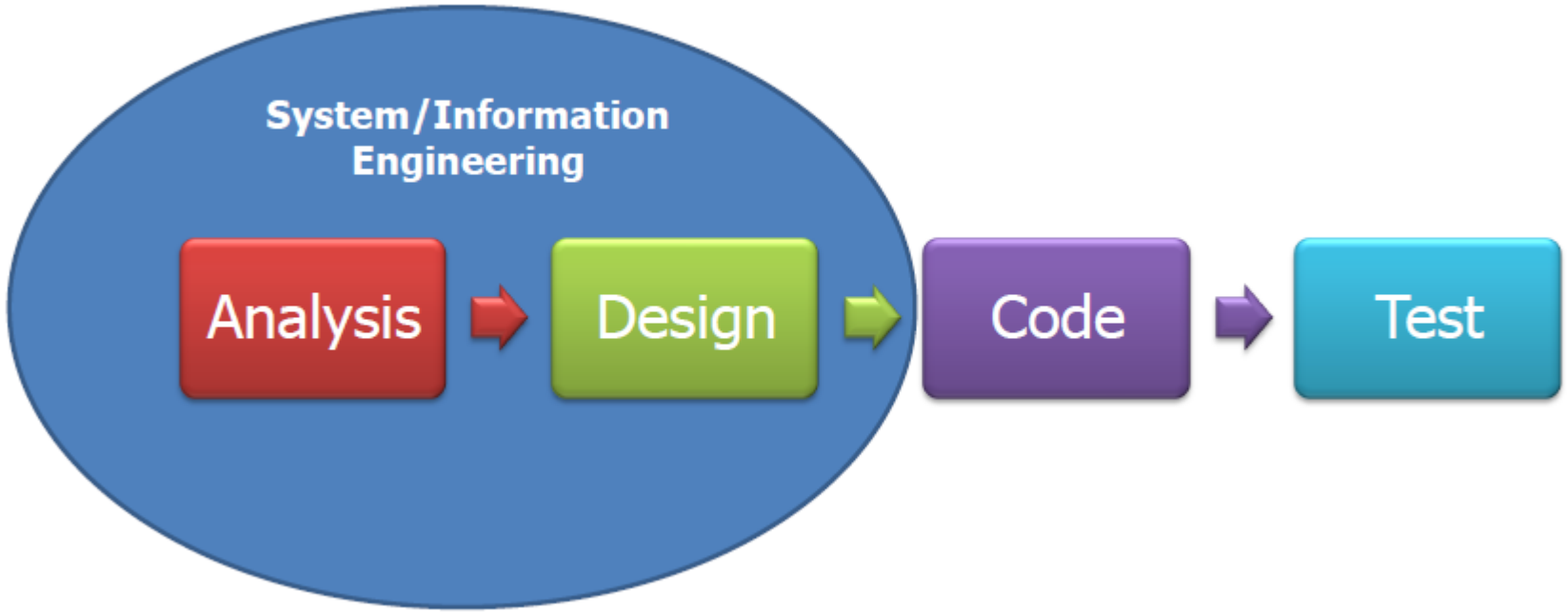
Terdiri dari 5 aktifitas umum dalam pembangunan perangkat lunak, yaitu:

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

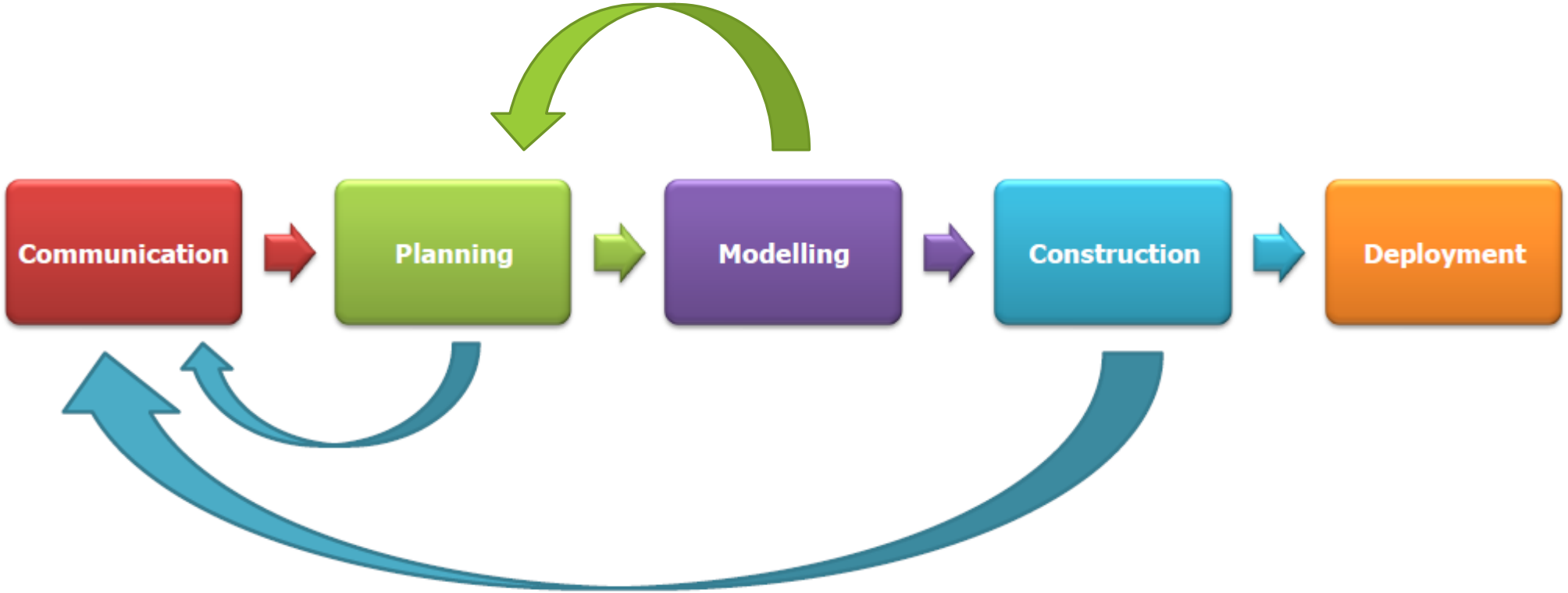
Process Flow

- Linear Process Model
- iterative Process Model
- Evolutionary Process Model

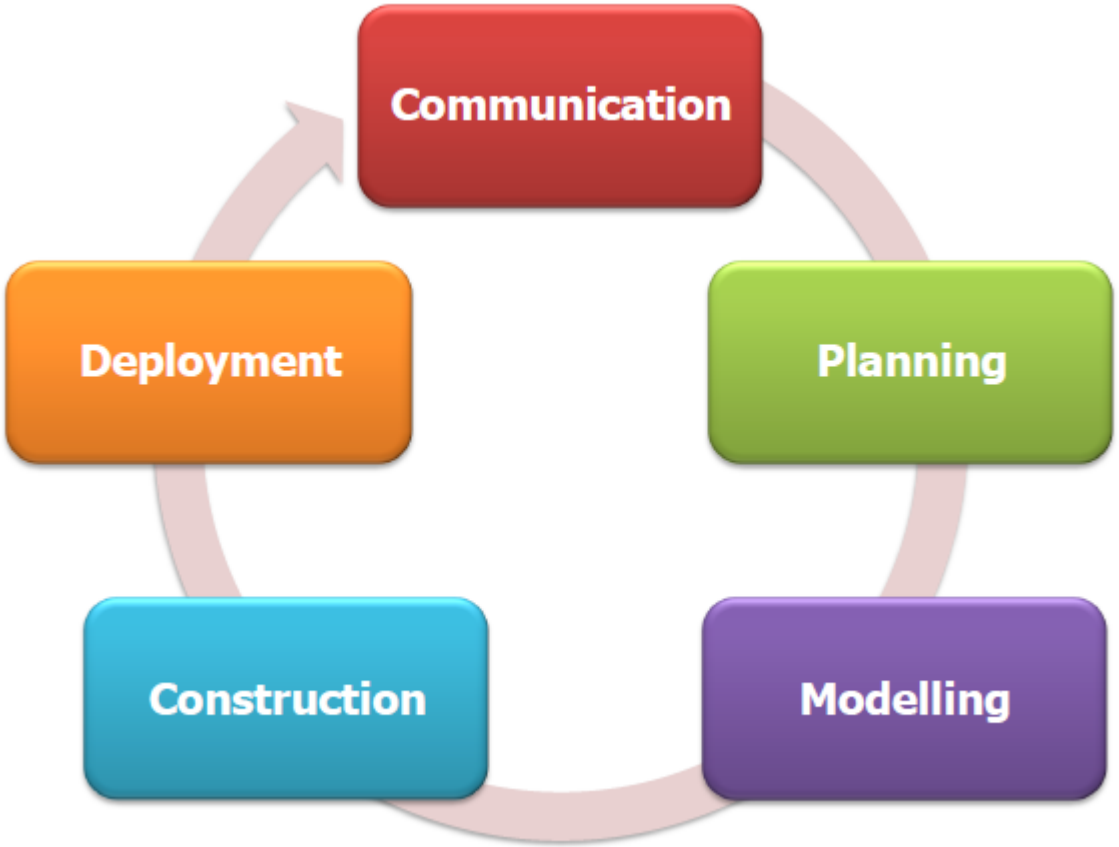
Linier Process Model



Iterative Process Model



Evolutionary Process Model



Presciptive Process Model

Presciptive Process Model

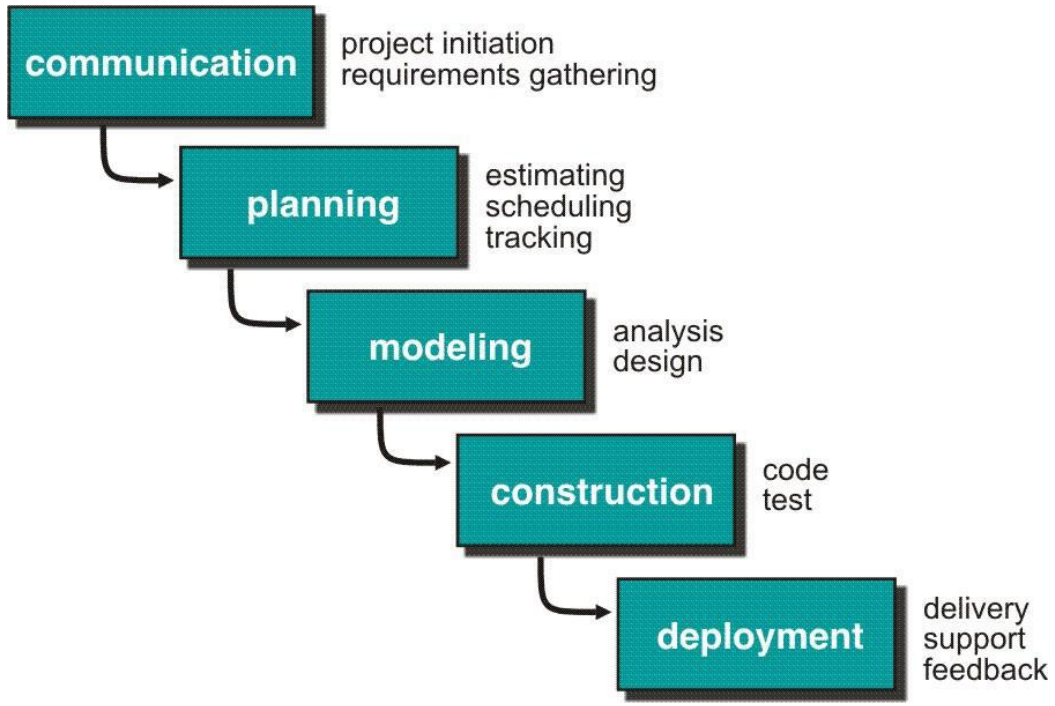
- Waterfall Model
- V Model
- Incremental Process Model
- Evolutionary Process Model
- Specialized Process Model
- Unified Process
- Agile Methods

1. Waterfall Model

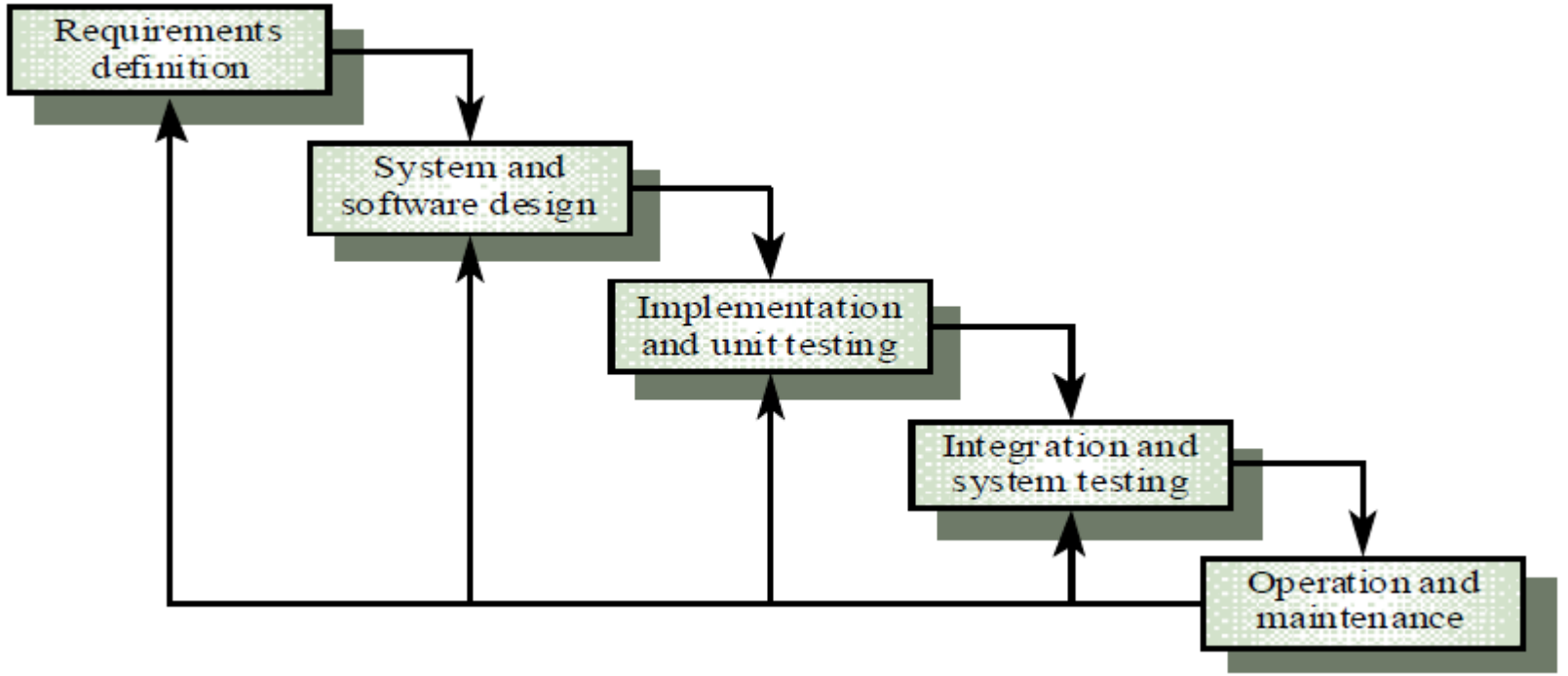
KARAKTERISTIK WATERFALL MODEL

- 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.

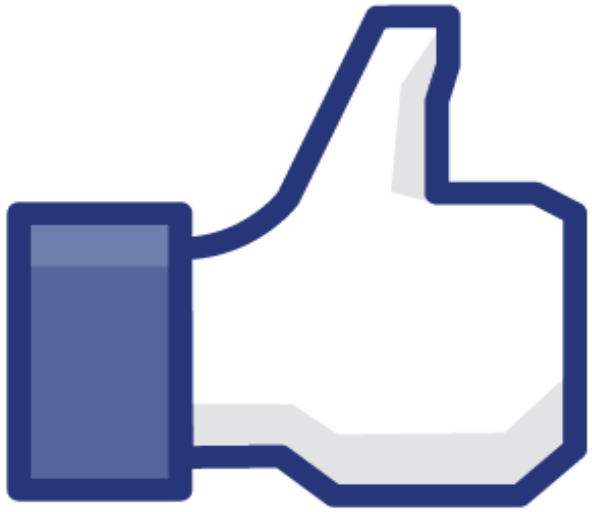
GAMBAR WATERFALL MODEL (ROGER S, PRESSMAN)



GAMBAR WATERFALL MODEL (IAN SOMMERFILLE)

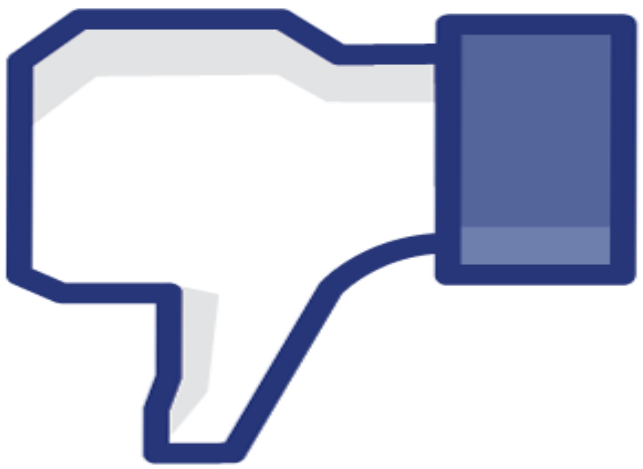


KELEBIHAN METODE WATERFALL



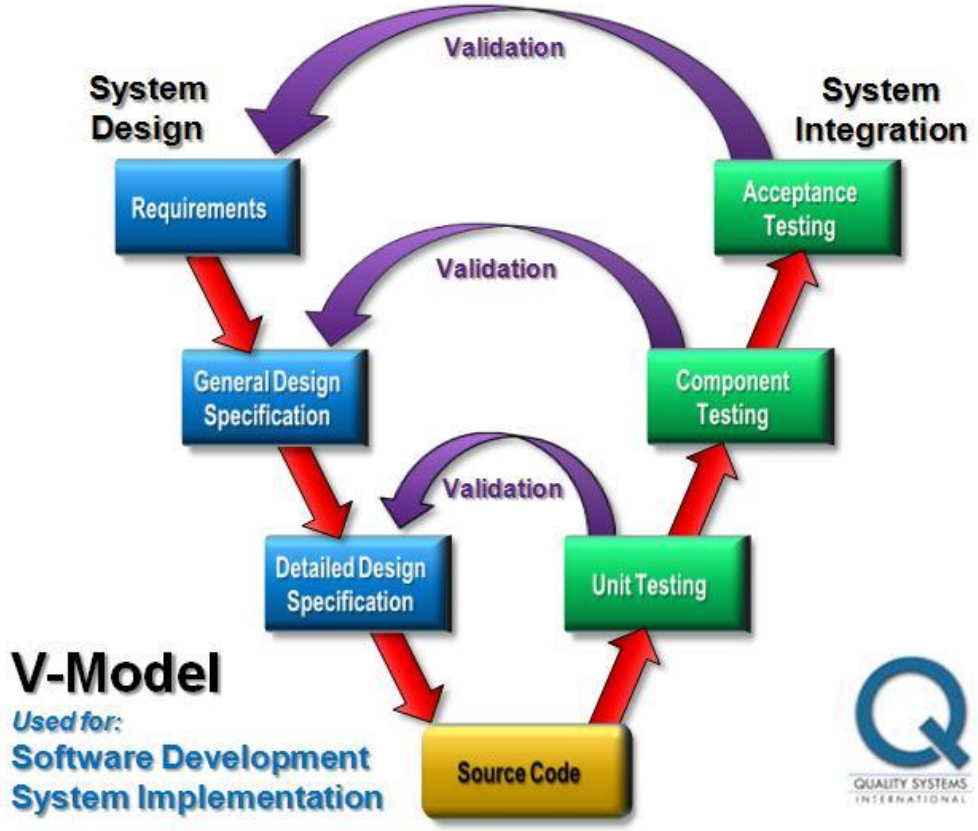
- Menghasilkan mature process pada setiap tahapnya.
- Mudah untuk diaplikasikan pada sebuah proyek.
- Menghasilkan sistem yang terstrukturdengan baik.
- Memiliki tingkat visibilitas yang tinggi (setiap tahap mempunyai dokumen yang jelas).

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


Validation
System Design
System Integration
Requirements
Acceptance Testing
Validation
General Design Specification
Component Testing
Validation
Detailed Design Specification
Unit Testing
V-Model
Used for:
Software Development
System Implementation
Source Code
QUALITY SYSTEMS INTERNATIONAL

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 PROCESS MODEL

Model Incremental merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang.

Elemen-elemen tersebut dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan memenuhi kebutuhan pemakai.

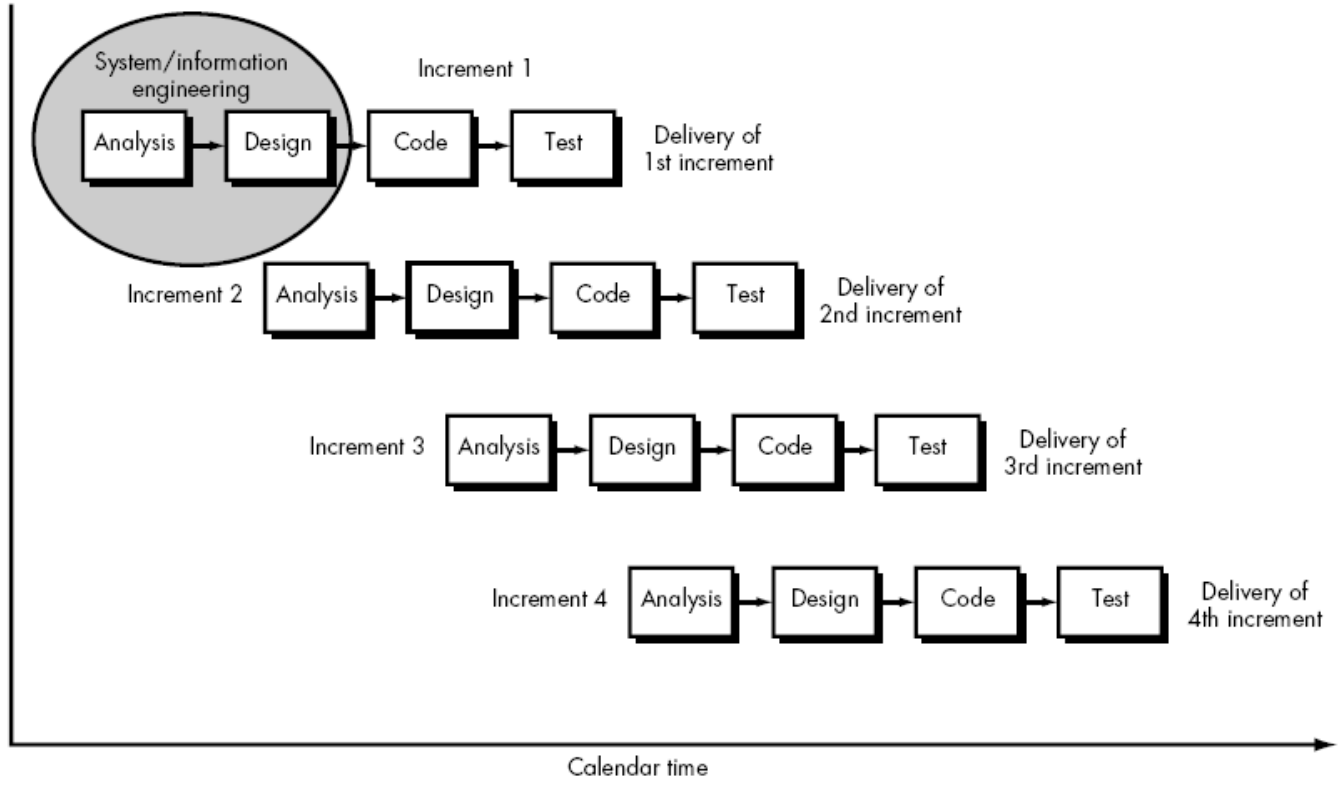
INCREMENTAL PROCESS MODEL

Model ini berfokus pada penyampaian produk operasional dalam Setiap pertambahanya.

Pertambahan awal ada di versi stripped down dari produk akhir, tetapi memberikan kemampuan untuk melayani pemakai dan juga menyediakan platform untuk evaluasi oleh pemakai.

Model ini cocok dipakai untuk proyek kecil dengan anggota tim yang sedikit dan ketersediaan waktu yang terbatas.

Incremental Model



RAD Model

Rapid application development(RAD) adalah model proses pengembangan perangkat lunak tambahan yang menekankan siklus perkembangan yang sangat pendek.

Model RAD adalah adaptasi dari model sekuensial linier (Waterfall models) "kecepatan tinggi" berbasis komponen.

Metode RAD akan berjalan maksimal jika pengembang aplikasi telah merumuskan kebutuhan dan ruang lingkup pengembangan aplikasi dengan baik.

RAD Model

Business modelling

- Menjawab pertanyaan-pertanyaan seperti informasi apa yang mengendalikan proses bisnis? Informasi apa yang dihasilkan? Siapa yang menghasilkan informasi? Kemana informasi itu diberikan? Siapa yang mengolah informasi?.

Data modelling

- aliran informasi yang sudah didefinisikan, disusun menjadi sekumpulan objek data. karakteristik/atribut dan hubungan antar objek-objek tersebut analisis kebutuhan dan data.

RAD Model

Process Modelling

- objek data yang sudah didefinisikan diubah menjadi aliran informasi yang diperlukan untukmenjalankan fungsi-fungsi bisnis.

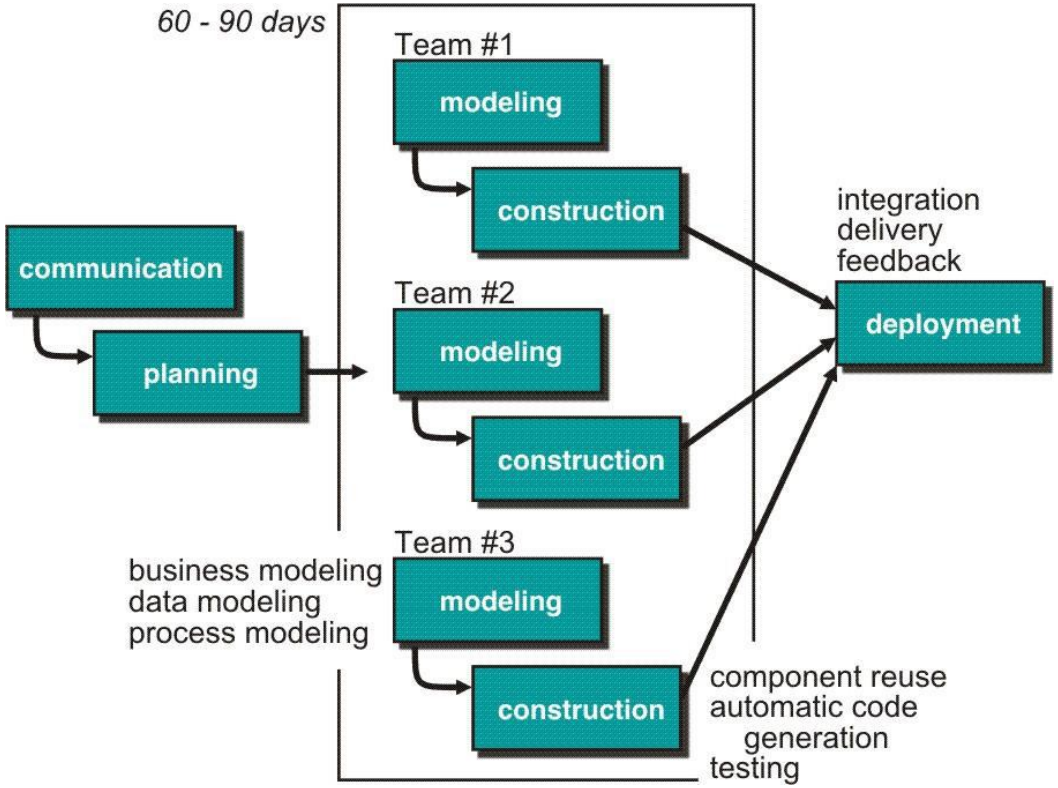
Application Generation

- RAD menggunakan component program yang sudah ada atau membuat component yang bisa digunakan lagi, selama diperlukan.

Testing and Turnover

- karena menggunakan component yang sudah ada, maka kebanyakan component sudah melalui uji atau testing. Namun component baru dan interface harus tetap diuji.

RAD Model


60 - 90 days
Team #1
modeling
construction
communication
planning
Team #2
modeling
construction
integration
delivery
feedback
deployment
Team #3
business modeling
data modeling
process modeling
modeling
construction
component reuse
automatic code
generation
testing

EVOLUTIONARY PROCESS MODEL

- Software system **evolves over time** as requirements often change as development proceeds. Thus, a straight line to a complete end product **is not possible**. However, **a limited version must be delivered** to meet competitive pressure.
- Usually a set of core product or system requirements is well understood, but the details and extension have yet to be defined.

EVOLUTIONARY PROCESS MODEL

- You need a process model that has been explicitly designed to accommodate a product that evolved over time.
- It is iterative that enables you to develop increasingly more complete version of the software.
- Two types are introduced, namely **Prototyping** and **Spiral** models.

PROTOTYPING: When to Use?

Customer defines a set of general objectives but **does not identify detailed** requirements for functions and features. Or **Developer may be unsure** of the efficiency of an algorithm, the form that human computer interaction should take.

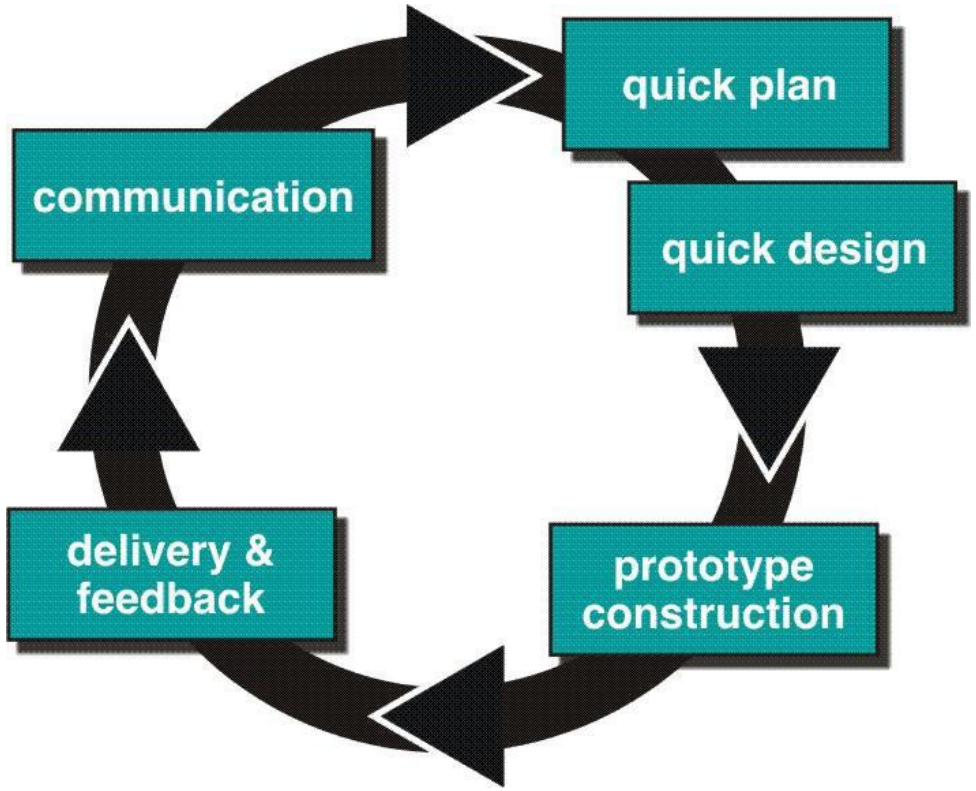
PROTOTYPING: What Step?

- Begins with communication by **meeting with stakeholders** to **define the objective**, identify whatever requirements are known, outline areas where further definition is mandatory.
- **A quick plan** for prototyping and modeling (quick design) occur.
- **Quick design** focuses on a representation of those aspects the software that will be visible to end users. (interface and output).
- Design leads to the construction of a prototype which will be **deployed and evaluated**. Stakeholder's comments will be used to **refine requirements**.

PROTOTYPING: Conclusion

- Both stakeholders and software engineers like the prototyping paradigm.
- **Users get a feel for the actual system, and developers get to build something immediately.**
- However, engineers may make compromises in order to get a prototype working quickly. The less-than-ideal choice may be adopted forever after you get used to it.

PROTOTYPING: Model



THE SPIRAL

Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model **prototyping** dengan pengulangannya dan model **waterfall** dengan pengendalian dan sistematikanya.

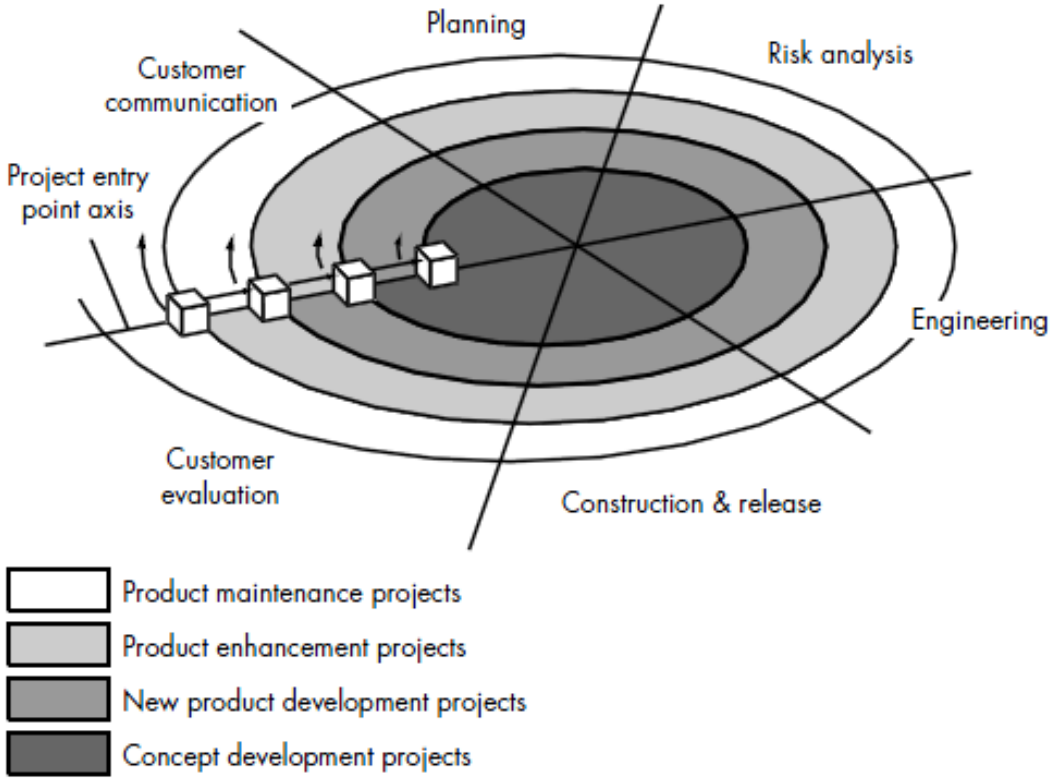
Model ini dikenal dengan sebutan **Spiral Boehm**.

Pengembang dalam model ini memadukan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.

THE SPIRAL

Liason	• membangun komunikasi yang baik dengan calon pengguna/pemakai.
Planning (perencanaan)	• menentukan sumber-sumber informasi, batas waktu dan informasi-informasi yang dapat menjelaskan proyek.
Analisis Risiko	• mendefinisikan risiko, menentukan apa saja yang menjadi risiko baik teknis maupun manajemen.
Rekayasa (engineering)	• pembuatan prototipe.
Konstruksi dan Pelepasan (release)	• melakukan pembangunan perangkat lunak yang dimaksud, diuji, diinstall dan diberikan dukungan tambahan untuk keberhasilan proyek.
Evaluasi Pengguna	• memberikan masukan berdasarkan hasil yang didapat dari tahap engineering dan instalasi.

THE SPIRAL: Model



CONCURENT PROCESS MODEL (CPM)

Pada model ini aktifitas kerja dilakukan secara bersamaan, setiap proses kerja memiliki beberapa pemicu kerja dari aktifitas.

Pemicu dapat berasal dari awal proses kerja maupun dari pemicu yang lain karena setiap pemicu akan saling berhubungan.

Sebagai contoh proses desain akan berubah atau dihentikan sementara karena ada perubahan permintaan kebutuhan dari customer.

CONCURENT PROCESS MODEL (CPM)

CPM dapat digambarkan secara skematik sebagai rangkaian dari kegiatan teknis utama, tugas dan hubungan antar bagian.

Metode ini merupakan suatu skema model yang mengimplementasikan suatu proses kerja yang dilakukan cepat namun dikerjakan secara bersama-sama dan tetap efektif dalam menyelesaikan berbagai penyelesaian masalah sesuai permintaan customer.

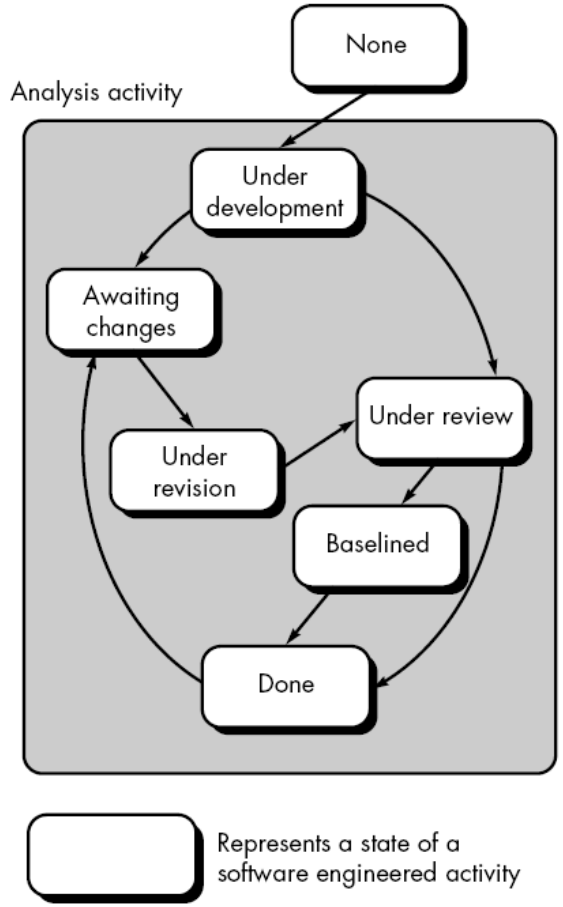
CONCURENT PROCESS MODEL (CPM)

Diagram Modeling/Analysis Activity menunjukkan skematik dari satu aktivitas dengan Concurrent Process Model.

Aktivitas analisa pada setiap orang mencatat bagian-bagian di setiap waktu sesuai jadwal.

Dengan cara yang sama, aktivitas yang lain seperti komunikasi antara customer dapat digambarkan dengan cara yang sama.

CONCURENT PROCESS MODEL (CPM)



CONCURENT PROCESS MODEL (CPM)

CPM sering digunakan sebagai paradigma untuk pengembangan aplikasi Client/Server.

Sistem Client/Server terdiri atas satu set komponen yang fungsional.

Ketika diaplikasikan untuk Client/Server, CPM menggambarkan aktivitas di dua dimensi yaitu dimensi sistem dan dimensi komponen.

- **Dimensi Sistem** ditujukan menggunaan tiga aktivitas : **Design, Perakitan** (Assembly) dan **Penggunaan** (Use).
- **Dimensi Komponen** ditujukan dengan dua aktivitas : **Design** dan **Realisasi**.

CONCURENT PROCESS MODEL (CPM)

Concurrency dicapai dalam jalan dua arah yaitu sebagai berikut :

- Sistem dan komponen aktivitas terjadi secara simultan dan dapat diperagakan menggunakan pendekatan yang berorientasi status sebelumnya.
- Kekhasan aplikasi Client/Server adalah diterapkan dengan banyak komponen, masing-masing dapat dirancang dan direalisasi secara bersamaan.

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

Component-based development sangat berkaitan dengan teknologi berorientasi objek.

Pada pemrograman berorientasi objek, banyak class yang dibangun dan menjadi komponen dalam suatu software.

Class-class tersebut bersifat reusable artinya bisa digunakan kembali.

Model ini bersifat iteratif atau berulang-ulang prosesnya.

COMPONENT BASED DEVELOPMENT

Identifikasi kelas-kelas yang akan digunakan kembali dengan menguji kelas tersebut dengan data yang akan dimanipulasi dengan aplikasi / software / algoritma yang baru

Kelas yang dibuat pada proyek sebelumnya disimpan dalam kelas library, sehingga bisa langsung diambil dari library yang sudah ada. Jika ada kebutuhan kelas baru, maka kelas dibuat dengan metode berorientasi objek.

Bangun software dengan kelas-kelas yang sudah ditentukan atau kelas baru yang dibuat, integrasikan.

COMPONENT BASED DEVELOPMENT

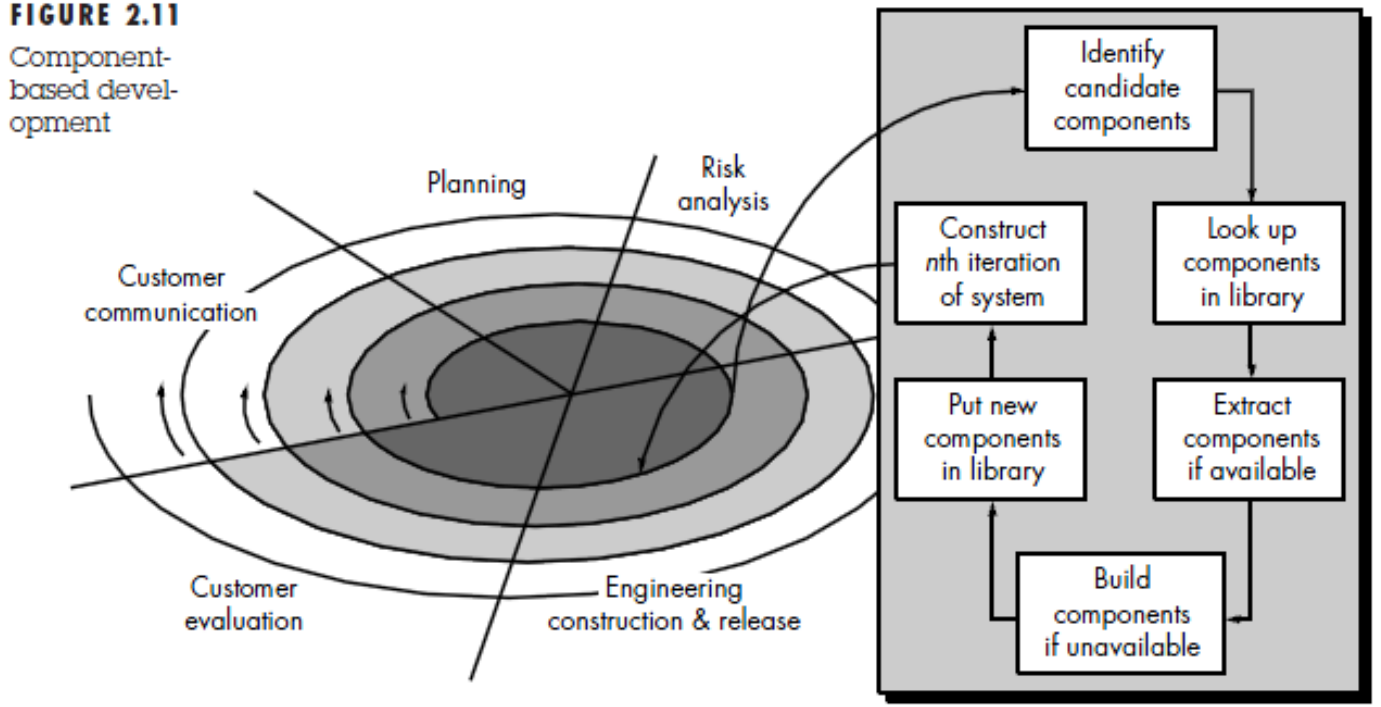


FIGURE 2.11
Component-based development

FORMAL METHOD

Teknik formal method adalah teknik yang mengandalkan perhitungan matematika dalam setiap prosesnya.

Hanya digunakan pada sistem yang sangat memperhatikan **keamanan atau keselamatan dari pengguna**.

Contoh penggunaan teknik ini adalah **aerospace engineering.**

FORMAL METHOD

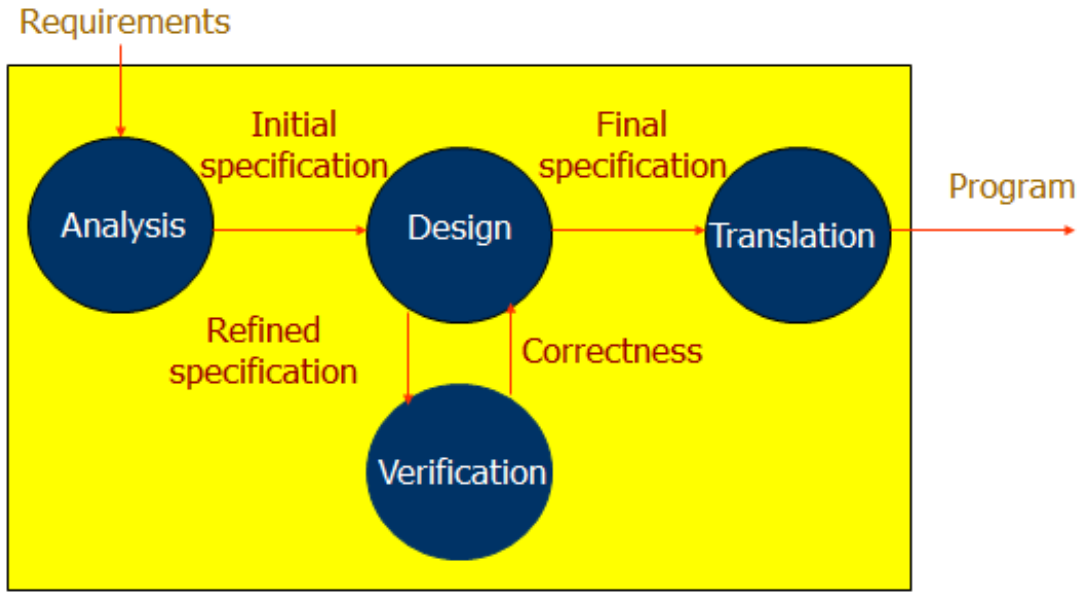
Pertama
- persyaratan informal dianalisis dan fungsi ditentukan secara resmi.

Kedua
- proses pembangunan membutuhkan spesifikasi formal ini dan merubahnya menjadi lebih rinci.

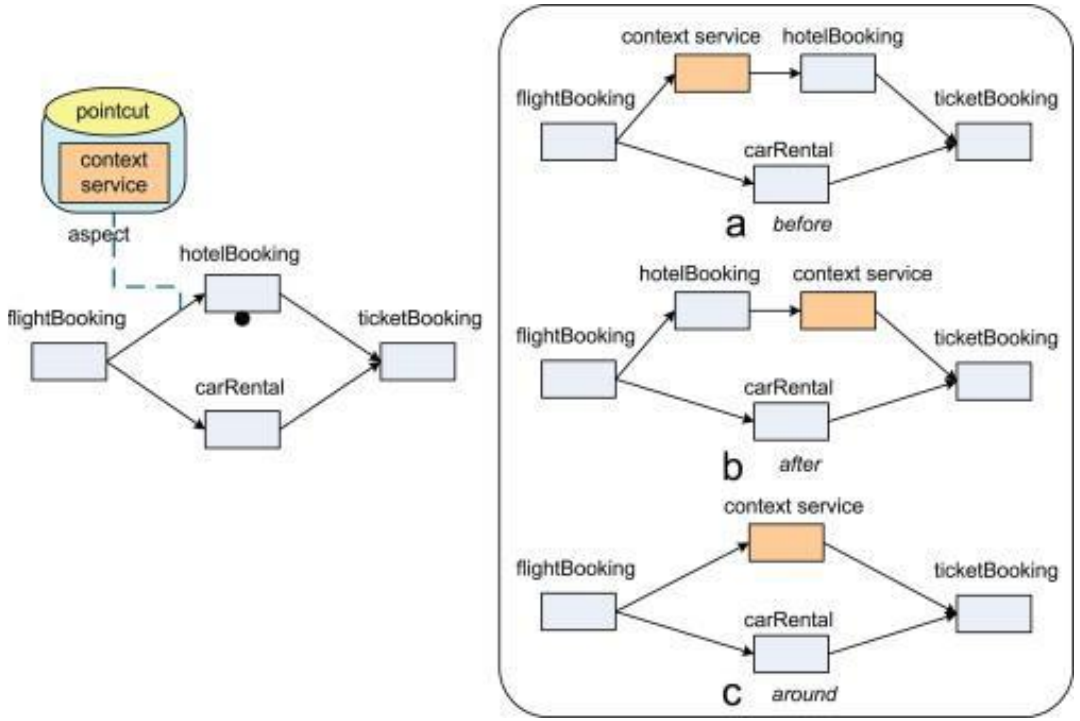
Ketiga
- deskripsi dapat dieksekusi oleh beberapa prosesor.

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
```



ASPECT ORIENTED MODEL



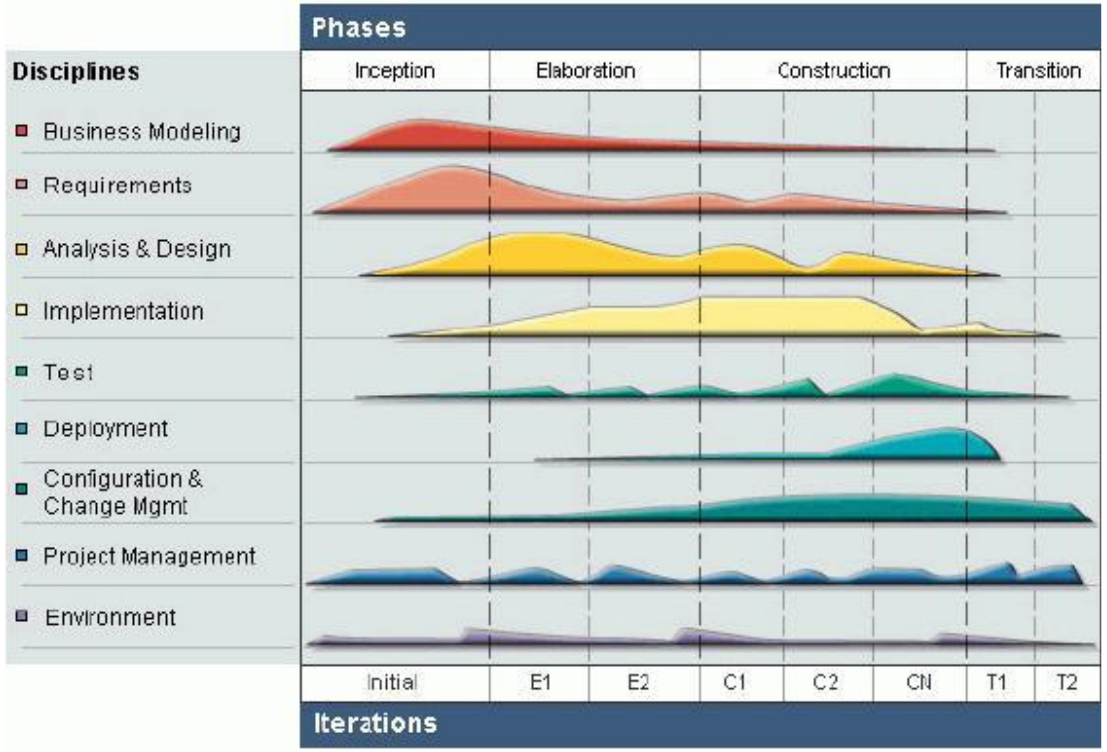
UNIFIED PROCESS

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

Penekanan pada model proses ini, yaitu:

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

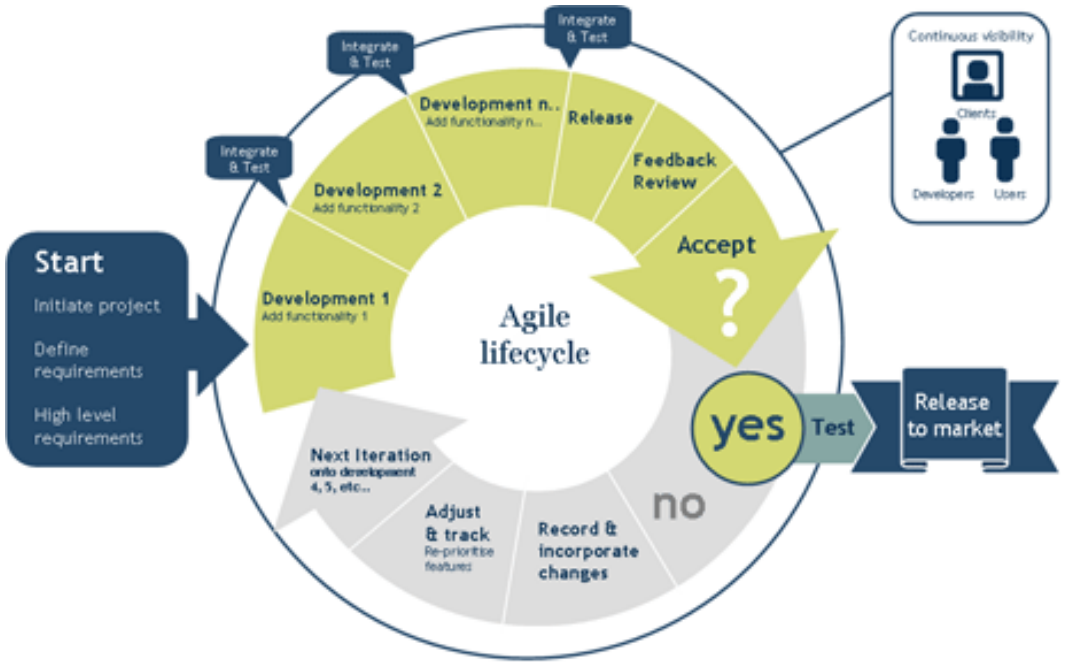
UNIFIED PROCESS

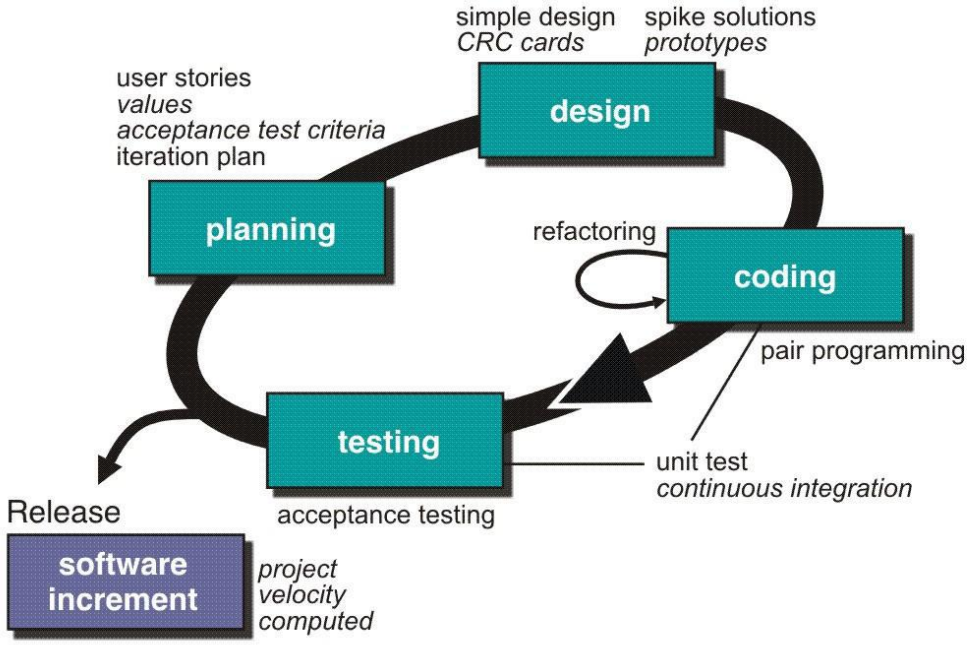


AGILE METHOD

"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**