

IF208 - Dasar Rekayasa Perangkat Lunak  
Tahun ajaran 2023 - 2024

# *Process Model (model process)*



**Program Studi Teknik Informatika**

Jurusan Teknik Informatika  
Politeknik Negeri Batam

Jalan Ahmad Yani, Batam Center, Batam 29461  
[www.polibatam.ac.id](http://www.polibatam.ac.id)

# Dasar pemikiran

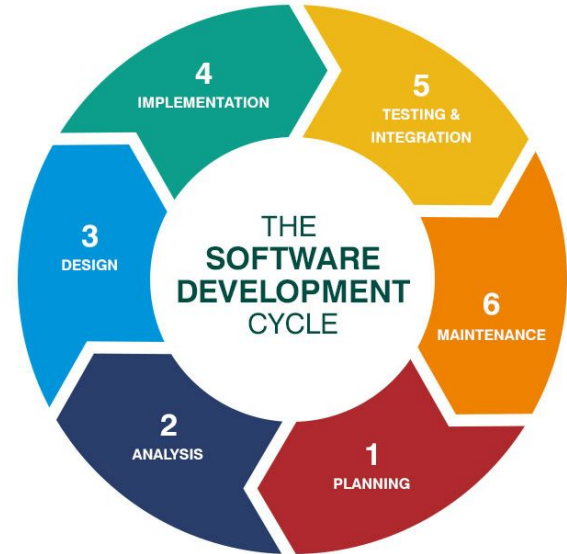
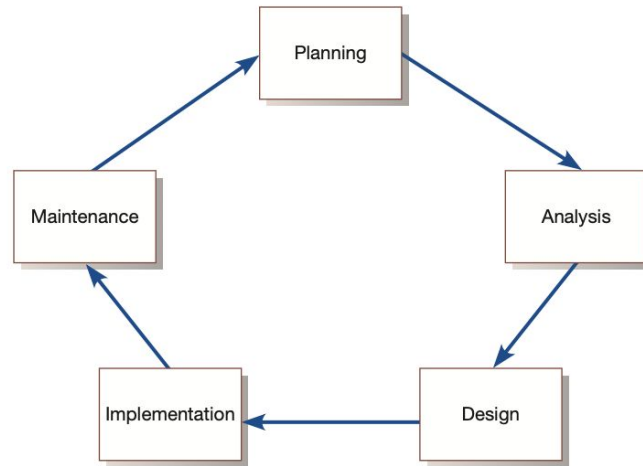
1. Apa yang dimaksud dengan model proses ?
2. Mengapa kita harus memahami model proses ?
3. Model proses apa yang paling bagus ?
4. Apa akibat pemilihan model proses yang *salah*?

# Software Process

- Merupakan sekumpulan aktivitas yang harus dilakukan untuk membangun sebuah perangkat lunak
- *Software process model* (model proses perangkat lunak) merupakan penggambaran dari *software process*
- *Plan-driven process* adalah proses dimana semua aktivitas direncanakan di awal dan progress diukur berdasarkan rencana tersebut
- *Agile process* merupakan proses dimana perencanaan dilakukan secara bertahap sehingga lebih mudah mengubah proses tersebut sesuai kebutuhan customer
- Tidak ada software process yang **benar** atau **salah**

# Software Process Activities

- Planning
- Analysis
- Design
- Implementation
- Testing
- Maintenance



# Model Proses

- Secara umum, sebuah proses terdiri dari lima aktivitas—perencanaan, analisis, desain, implementasi, pengujian. Sebagai tambahan, beberapa aktivitas pendukung—*project tracking and control, risk management, quality assurance, configuration management, technical reviews*, dan sebagainya—juga dilakukan selama proses berlangsung.
- Setiap aktivitas terdiri dari serangkaian aksi

### Process framework

#### Umbrella activities

##### framework activity # 1

software engineering action #1.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

software engineering action #1.k

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

##### framework activity # n

software engineering action #n.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

software engineering action #n.m

Task sets

work tasks  
work products  
quality assurance points  
project milestones

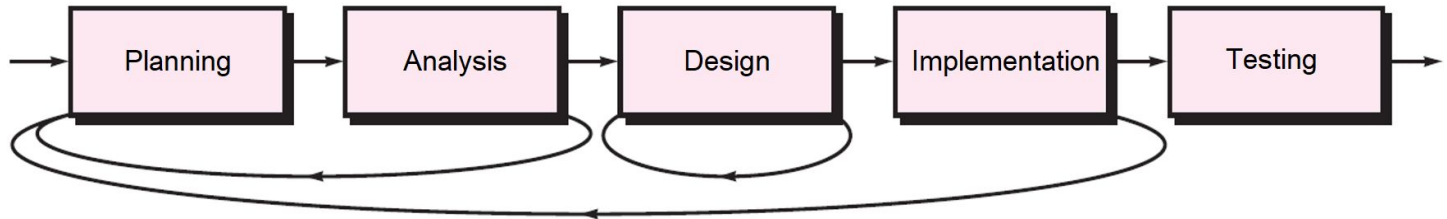
# Umbrella Activities (Pressman, 2009)

- *Software project tracking and control*—untuk tim pengembang melacak progres/kemajuan proyek dan agar tim pengembang dapat mengambil tindakan yang diperlukan agar pengerjaan sesuai dengan jadwal.
- *Risk management*—menentukan semua risiko yang mungkin dapat mempengaruhi kinerja proyek atau kualitas produk.
- *Software quality assurance*—mendefinisikan dan menjalankan semua aktivitas untuk menjamin kualitas produk.
- *Software configuration management*—mengelola efek atau akibat yang terjadi karena adanya perubahan selama berlangsungnya *software process*.
- *Reusability management*—mendefinisikan kriteria untuk produk yang dapat digunakan kembali (termasuk *software components*) dan menentukan mekanisme untuk menghasilkan *software component* yang dapat digunakan kembali.
- dsb...

# Process Flow

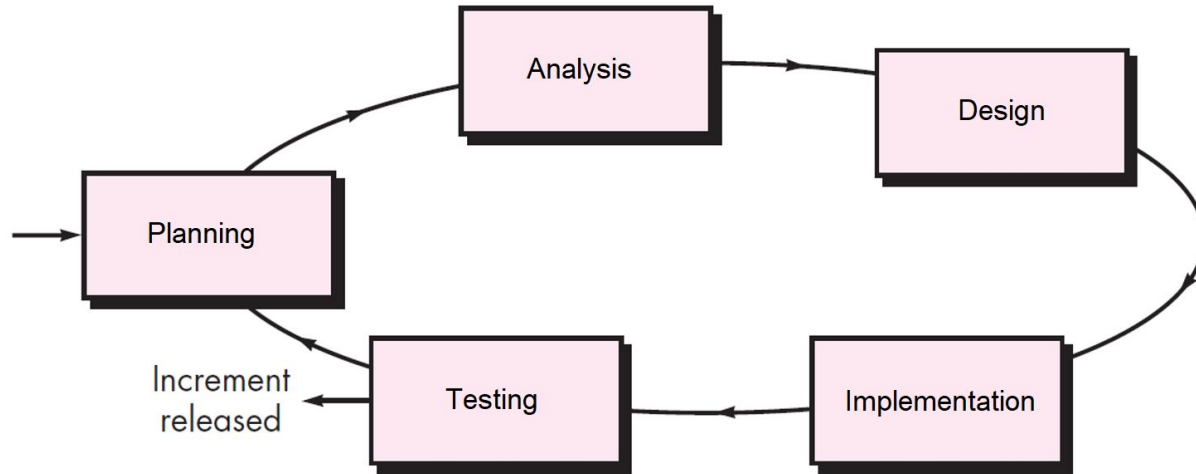


(a) Linear process flow



(b) Iterative process flow

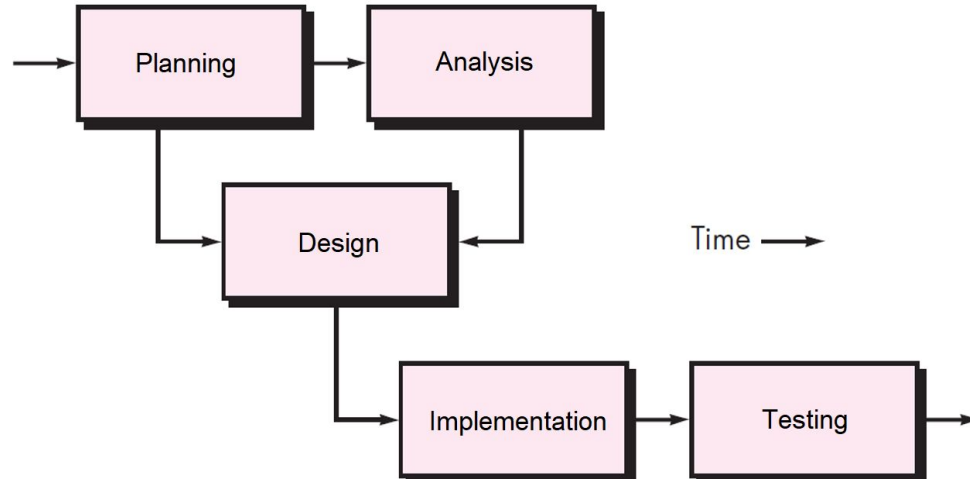
# Process Flow



(c) Evolutionary process flow

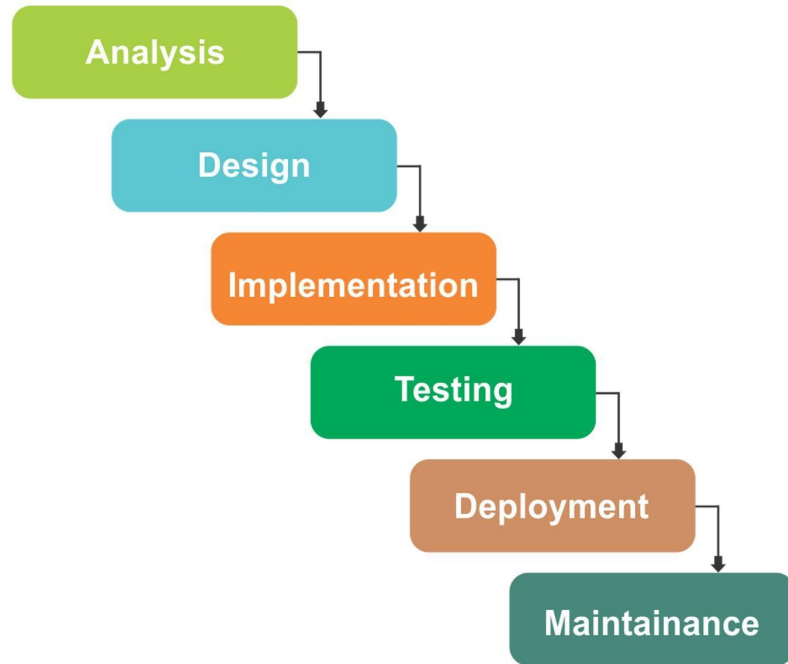


# Process Flow

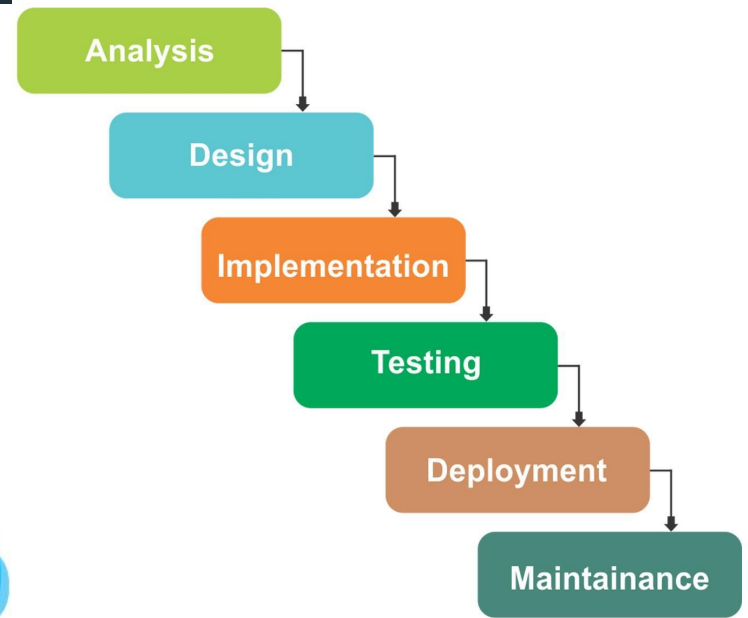
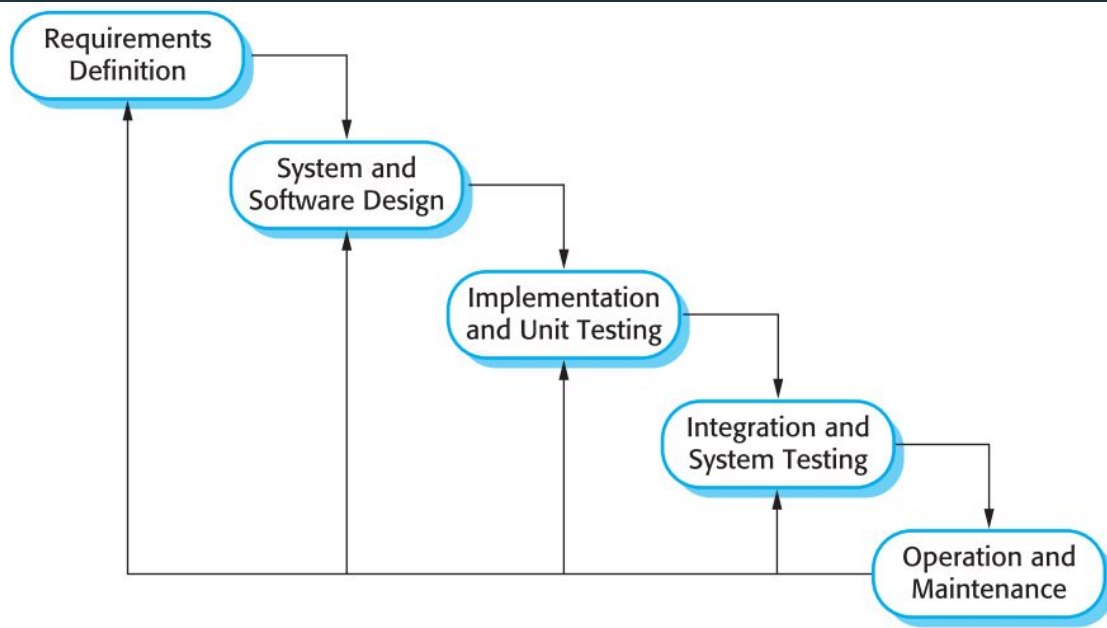


(d) Parallel process flow

# Model Waterfall (Royce, 1970)



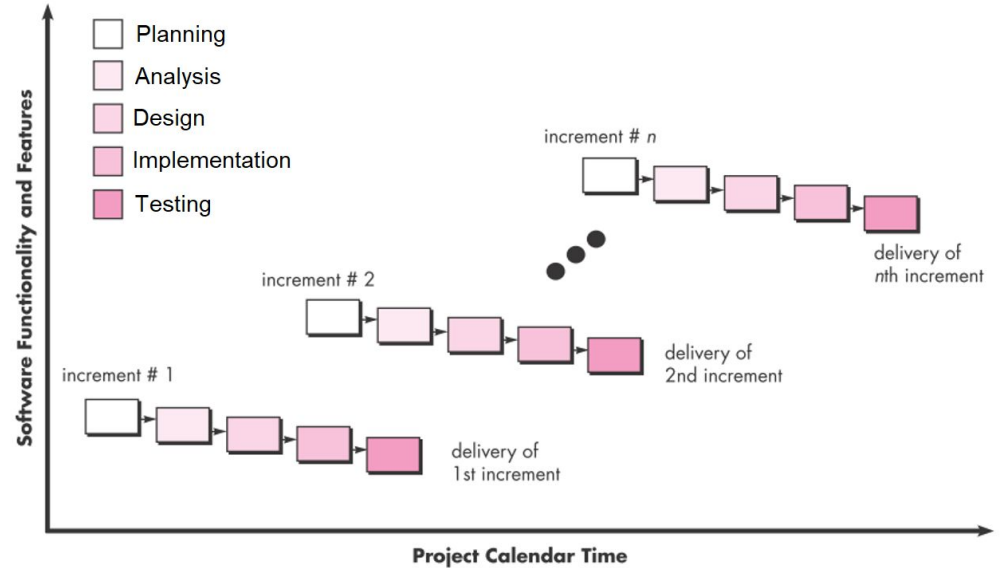
- Sebuah tahap harus diselesaikan sebelum maju ke tahap berikutnya
- Model ini hanya sesuai pada software dengan *requirements* yang sudah jelas terdefinisi dan tidak banyak perubahan pada proses desain
- Kelemahan:
  - Tidak mudah bagi customer untuk mendefinisikan semua requirements
  - Versi software yang dapat dijalankan baru akan terbangun di tahap akhir proyek



Walaupun dalam versi asli model waterfall yang ditemukan oleh Winston Royce (1970) terdapat “*feedback loops*” (gambar kiri), namun biasanya implementasi dari model ini dilakukan secara linear saja (gambar kanan)

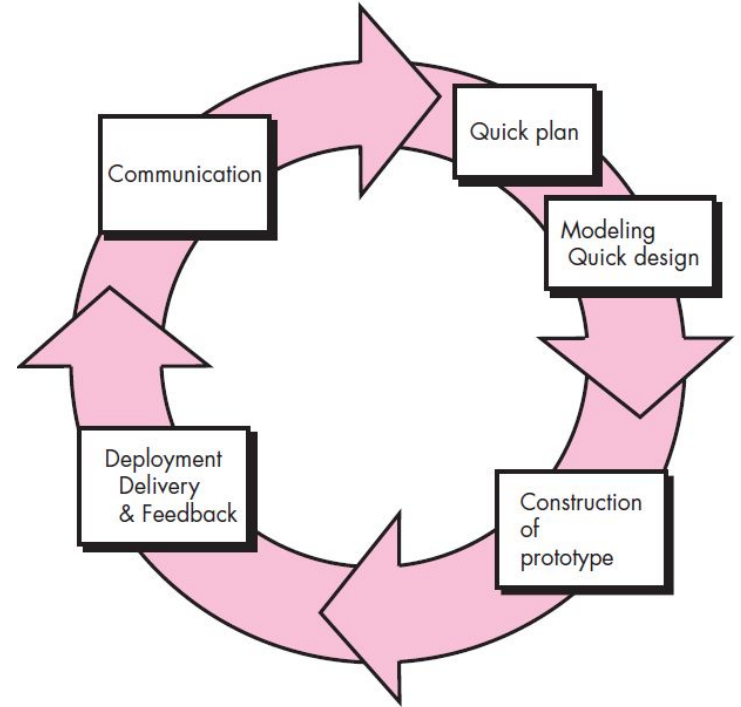
# Incremental Process Model

- Membangun versi awal, kemudian menerima komentar dari user dan mengubah versi awal tersebut sampai selesai
- Setiap versi menambahkan fungsi yang dibutuhkan user



# Prototyping

- Prototype adalah versi awal sebuah sistem yang digunakan untuk mendemonstrasikan konsep dan mencoba berbagai pilihan desain, dan mendalami masalah serta pemecahannya
- Dalam beberapa kasus, prototype ini tidak dapat digunakan lagi sehingga untuk sistem yang sebenarnya, developer membuat kembali dari awal ⇒ *'throw-away prototypes'*
- Prototipe juga dapat bersifat *evolutionary* dimana prototype akan terus dikembangkan/berevolusi menjadi produk yang sebenarnya

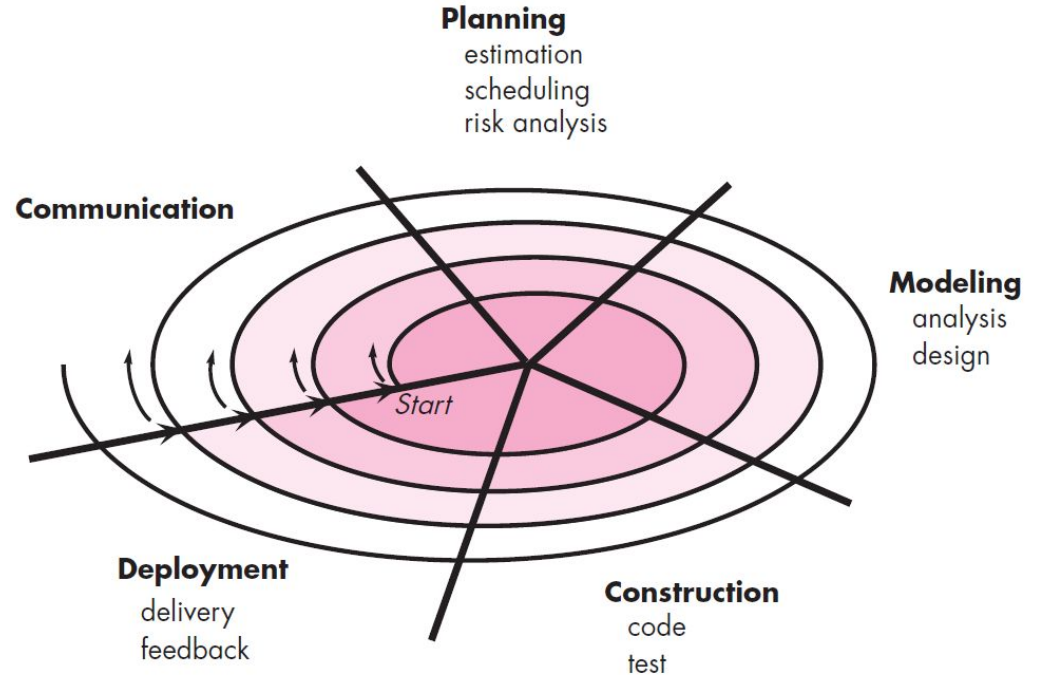


# Prototyping

- Kelemahan:
  - Sebagai pengembang perangkat lunak, kita seringkali melakukan beberapa kompromi dalam implementasi untuk mendapatkan prototipe secara cepat.
  - Customer melihat perangkat lunak yang terlihat sudah jadi dan berjalan, padahal prototype biasanya dibuat sejadinya, tanpa memperhitungkan kualitas secara keseluruhan atau aspek kinerja perangkat lunak yang lain secara jangka panjang.

# The Spiral Model (Barry Boehm, 1998)

- Menggabungkan sifat iterative dari prototyping dengan aspek sistematis dari model waterfall
- Merupakan model proses berbasis risiko yang digunakan untuk memandu pembangunan perangkat lunak



# The Spiral Model

- Dengan spiral model, perangkat lunak dikembangkan dengan serangkaian rilis yang bersifat evolutionary
  - Dalam iterasi-iterasi awal, rilis dapat berupa sebuah model atau prototype. Dalam iterasi-iterasi berikutnya, dihasilkan versi perangkat lunak yang lebih lengkap.
- Kelemahan:
  - Sulit untuk meyakinkan customers (terutama terkait kontrak) bahwa pendekatan yang bersifat evolutionary dapat dikendalikan dengan baik.
  - Diperlukan keahlian dalam manajemen risiko dan kesuksesan penggunaan spiral model sangat bergantung pada keahlian tersebut. Jika ada risiko besar yang tidak terungkap dan tidak dikelola dengan baik, maka pasti akan muncul masalah.