

Proses Pengembangan Perangkat Lunak (1)

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi
Politeknik Negeri Malang

Dosen Pengampu: Wilda Imama Sabilla, S.Kom., M.Kom.

Outline

- Proses Pengembangan Perangkat Lunak
- Model Proses Pengembangan Perangkat Lunak
- Contoh Model Proses Pengembangan Perangkat Lunak

Tujuan

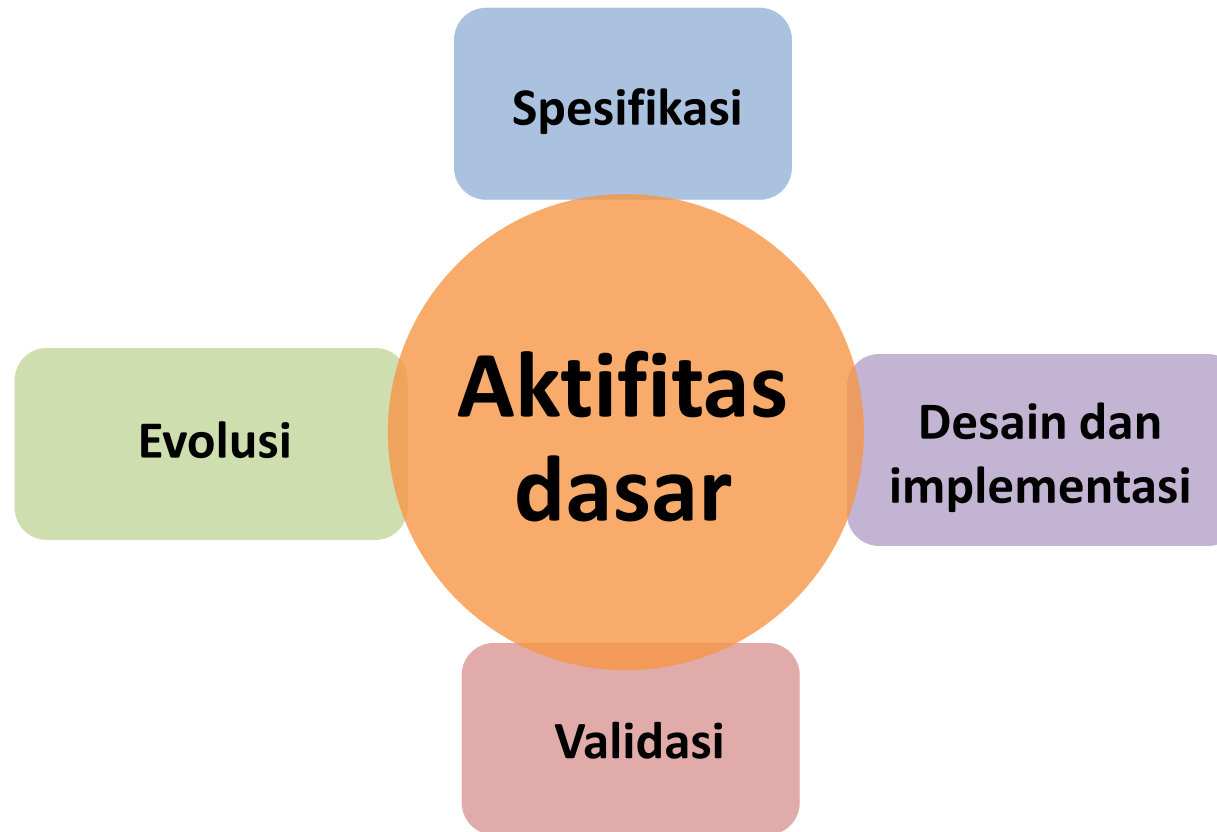
- Memahami konsep proses pengembangan perangkat lunak dan permodelannya
- Memahami proses pengembangan perangkat lunak beserta penerapannya
- Memahami dasar aktivitas dalam proses pengembangan perangkat lunak

Proses Pengembangan Perangkat Lunak

- Definisi:
 - **Rangkaian aktivitas dalam rangka memproduksi produk perangkat lunak**
- Istilah lain:
Software Development Life Cycle (SDLC)



Proses Pengembangan Perangkat Lunak



Proses Pengembangan Perangkat Lunak

Spesifikasi

- Fungsionalitas perangkat lunak dan batasan operasinya harus didefinisikan.
- Tahap di mana kebutuhan dan persyaratan perangkat lunak dirinci dan didokumentasikan secara jelas.
- Berfungsi sebagai landasan untuk seluruh proses pengembangan.



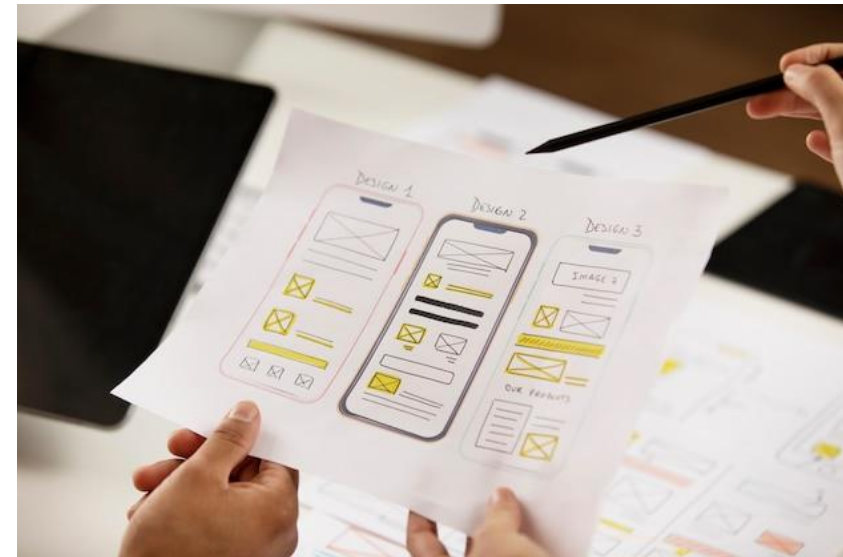
Aktivitas Proses: Software Specification

- Software specification atau requirements engineering proses identifikasi, analisis, dokumentasi, dan validasi kebutuhan atau persyaratan sistem yang akan dikembangkan.
- Proses ini menjadi tahap awal dan kritis dalam siklus pengembangan perangkat lunak karena kesuksesan proyek bergantung pada pemahaman yang tepat terhadap kebutuhan pengguna dan pemangku kepentingan.
- Tahap:
 1. Feasibility study (Tahap penilaian pada kebutuhan sistem).
 2. Elicitation and analysis (Tahapan pengumpulan kebutuhan apa saja yang di perlukan).
 3. Specification (Tahapan standarisasi kebutuhan yang harus ada dalam software).
 4. Validation (Tahapan pengecekan kebutuhan kepada costumer dan persetujuan).

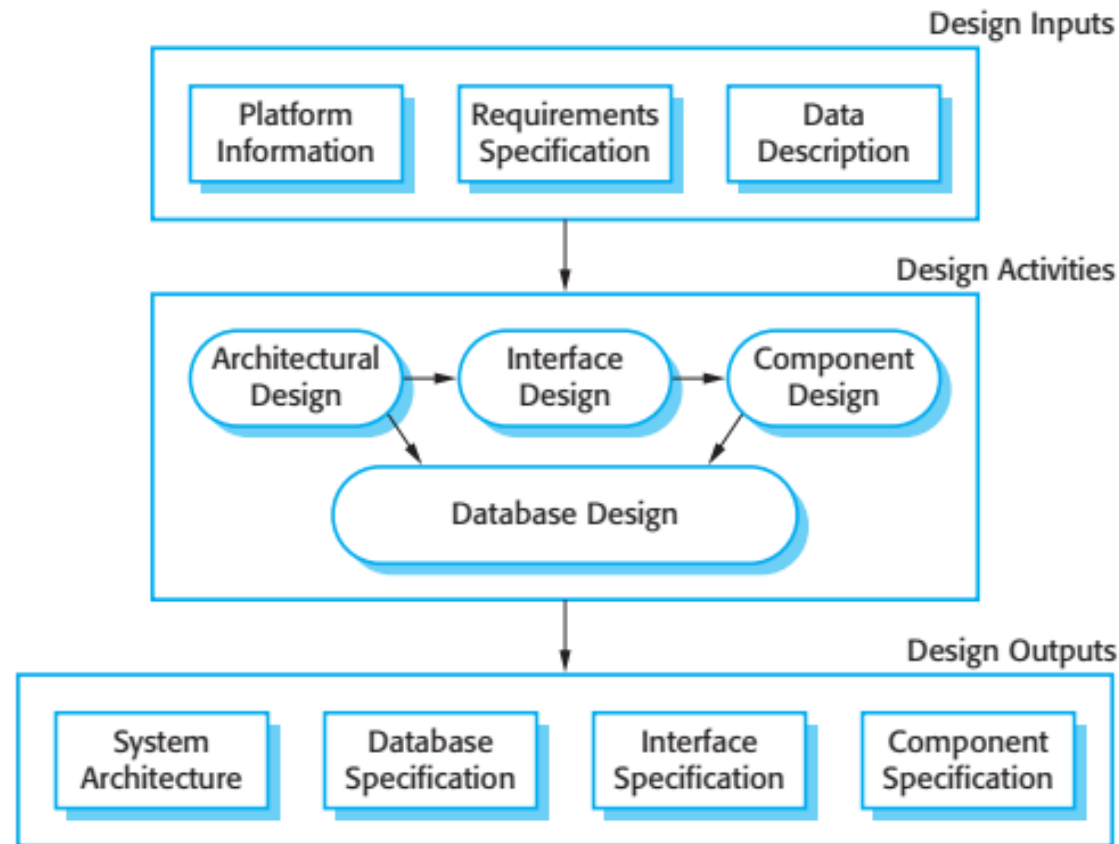
Proses Pengembangan Perangkat Lunak

Desain dan implementasi

- Memastikan agar dapat menghasilkan perangkat lunak sesuai dengan spesifikasi.
- Proses desain adalah tahap di mana arsitektur dan detail desain perangkat lunak dibuat berdasarkan spesifikasi yang telah ditentukan.
- Proses implementasi adalah tahap di mana desain yang telah dibuat diimplementasikan menjadi kode yang dapat dijalankan.



Aktivitas Proses: Software Design and Implementation



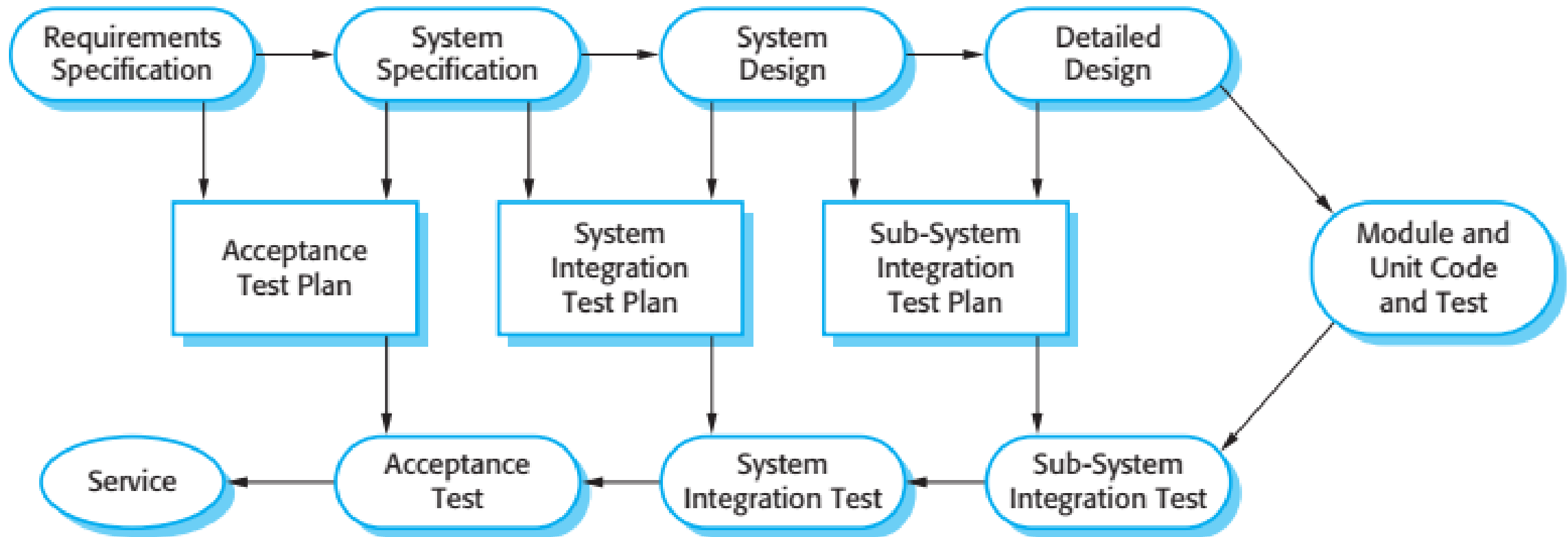
Proses Pengembangan Perangkat Lunak

Validasi

- Perangkat lunak harus divalidasi untuk memastikan sesuai dengan keinginan pengguna/klien.
- Tahap di mana perangkat lunak yang telah dikembangkan diperiksa dan diuji untuk memastikan bahwa ia memenuhi kebutuhan dan harapan pengguna.
- Agar dihasilkan perangkat lunak berkualitas, memenuhi kebutuhan, siap untuk produksi.



Aktivitas Proses: Software Validation



Proses Pengembangan Perangkat Lunak

Evolusi

- Perangkat lunak harus divalidasi untuk memastikan sesuai dengan keinginan pengguna/klien.
- Tahap di mana perangkat lunak yang telah dikembangkan diperiksa dan diuji untuk memastikan bahwa ia memenuhi kebutuhan dan harapan pengguna.
- Agar dihasilkan perangkat lunak berkualitas, memenuhi kebutuhan, siap untuk produksi.



Aktivitas Proses: Software Evolution

- Evolusi perangkat lunak mengacu pada perubahan sistem perangkat lunak sepanjang waktu dalam menanggapi perubahan kebutuhan, lingkungan operasional, teknologi, dan faktor-faktor lainnya.
- Proses ini seringkali tidak terhindarkan karena lingkungan di sekitar perangkat lunak terus berubah, dan karena itu, perangkat lunak harus beradaptasi agar tetap relevan dan berfungsi sebagaimana mestinya.

Aktivitas Proses: Software Evolution

1.Perubahan Kebutuhan: Kebutuhan untuk perangkat lunak dapat berubah seiring waktu karena faktor-faktor seperti perubahan bisnis, kebijakan perusahaan, atau umpan balik pengguna. Evolusi perangkat lunak memerlukan pemahaman yang baik tentang perubahan ini dan kemampuan untuk menyesuaikan fungsionalitas dan fitur sesuai dengan kebutuhan baru.

2.Perubahan Teknologi: Teknologi perangkat lunak juga terus berkembang, dengan munculnya bahasa pemrograman baru, framework, dan teknologi infrastruktur. Evolusi perangkat lunak memungkinkan penggunaan teknologi terbaru dan integrasi dengan sistem yang sudah ada untuk meningkatkan kinerja dan efektivitas perangkat lunak.

3.Perbaikan dan Peningkatan: Evolusi perangkat lunak juga melibatkan perbaikan bug, peningkatan kinerja, dan penambahan fitur baru untuk meningkatkan pengalaman pengguna dan memberikan nilai tambah kepada pengguna.

4.Perubahan Lingkungan Operasional: Perubahan dalam lingkungan operasional seperti infrastruktur perangkat keras, platform, atau sistem operasi juga dapat mempengaruhi evolusi perangkat lunak. Perangkat lunak harus dapat beradaptasi dengan perubahan ini untuk tetap berjalan secara efisien.

5.Manajemen Evolusi: Manajemen evolusi perangkat lunak melibatkan proses yang terstruktur untuk mengelola perubahan dan memastikan bahwa perangkat lunak tetap memenuhi kebutuhan pengguna serta tetap relevan dalam lingkungan yang berubah. Ini termasuk pemantauan kinerja, pemeliharaan kode, dan pengelolaan versi perangkat lunak.

Kategori Proses Pengembangan Perangkat Lunak

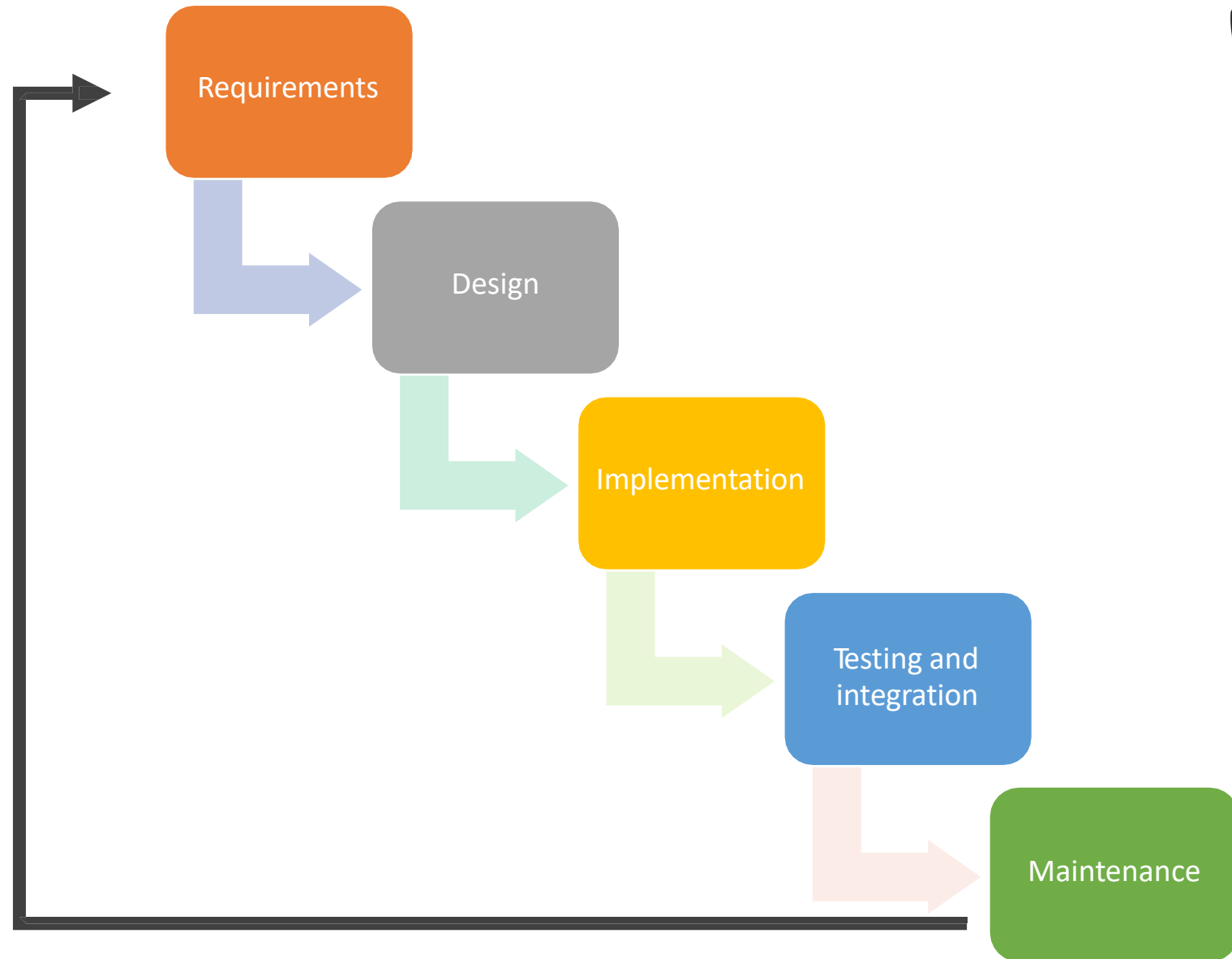
- Proses perangkat lunak itu rumit, seperti semua proses intelektual dan kreatif, bergantung pada orang yang membuat keputusan dan penilaian.
- Tidak ada proses yang ideal, sebagian besar organisasi telah mengembangkan proses pengembangan perangkat lunak mereka sendiri.
- Proses Pengembangan Perangkat Lunak dapat dikategorikan menjadi:
 - 1. Plan-driven**
 - Semua aktivitas pada proses direncanakan
 - Progress diukur sesuai rencana.
 - 2. Agile**
 - Perencanaan bersifat bertahap.
 - Lebih mudah untuk mengubah prosesnya (adaptif) agar sesuai dengan perubahan kebutuhan pelanggan.

Model Proses Pengembangan Perangkat Lunak

- Permodelan aktivitas pengembangan perangkat lunak.
- Setiap model proses merepresentasikan proses secara **umum**.
- Dapat digunakan sebagai *framework* yang dikembangkan menjadi proses pengembangan perangkat lunak yang spesifik.
- Model proses:
 - *Waterfall*
 - *Incremental development*
 - *Reuse-oriented software engineering*

Waterfall

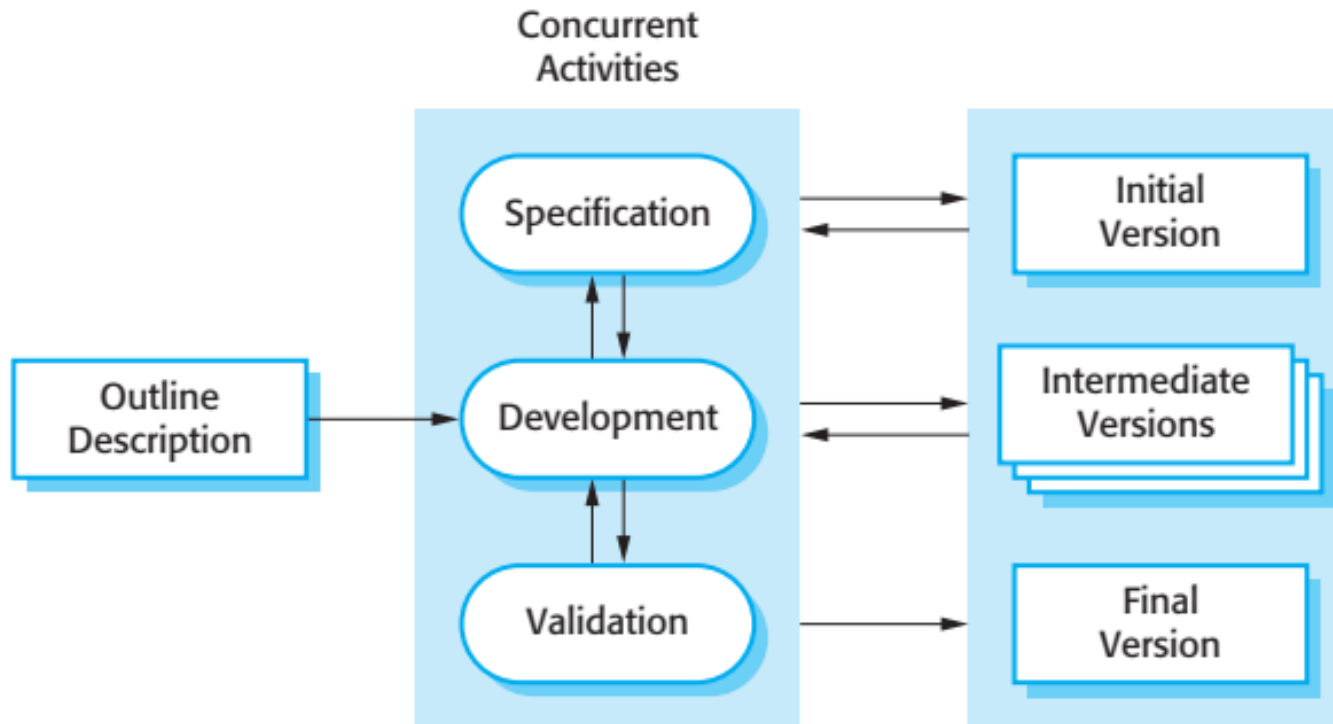
- Dibuat tahun 1970 oleh Winston W. Royce
- Setiap proses harus dilakukan sebelum menuju proses selanjutnya



Waterfall

- Kelebihan:
 - Setiap fase terdapat dokumentasi
 - Mudah untuk dimonitor oleh seorang project manager
- Kekurangan:
 - Kurang fleksibel, spesifikasi kebutuhan di awal harus 'sudah pasti'

Incremental development



- Proses pengembangan perangkat lunak yang dibagi menjadi beberapa versi hingga perangkat lunak menjadi satu produk yang utuh
- Menjadi dasar pendekatan Agile

Incremental development

- Kelebihan:
 - Dapat menekan biaya atas perubahan kebutuhan perangkat lunak
 - User dapat memberikan feedback pada setiap versi perangkat lunak
 - Perangkat lunak dapat dikembangkan secara cepat
- Kekurangan:
 - Lebih sulit untuk memonitor proses karena tidak ada dokumen yang *rigid* pada setiap prosesnya
 - Memungkinkan perubahan yang masif pada versi perangkat lunak selanjutnya, sehingga waktu dan biaya pengembangan dapat meningkat

Reuse-oriented Software Engineering

- Menggunakan pendekatan aspek *reusable* pada perangkat lunak
- Istilah lain: **Component Based Software Engineering**
- Komponen pada perangkat lunak yang telah ada, digunakan kembali sesuai kebutuhan



Reuse-oriented Software Engineering

- Beberapa contoh komponen yang bisa digunakan kembali:
 - Web service (layanan web) yang dikembangkan sesuai dengan standar layanan dan tersedia untuk pemanggilan jarak jauh.
 - Kumpulan objek yang dikembangkan sebagai paket untuk diintegrasikan dengan komponen framework.
 - Sistem perangkat lunak mandiri yang dikonfigurasi untuk digunakan dalam lingkungan tertentu.

Reuse-oriented Software Engineering

Component Analysis

- Berdasarkan spesifikasi kebutuhan, dilakukan pencarian komponen untuk mengimplementasikan spesifikasi tersebut.
- Biasanya, tidak ada kecocokan yang sempurna, dan komponen yang mungkin digunakan hanya menyediakan sebagian dari fungsionalitas yang dibutuhkan.

Reuse-oriented Software Engineering

Requirements modification

- Kebutuhan dianalisis menggunakan informasi tentang komponen yang telah ditemukan.
- Kemudian, kebutuhan tersebut dimodifikasi untuk mencerminkan komponen yang tersedia.
- Jika modifikasi tidak memungkinkan, aktivitas analisis komponen dapat diulangi untuk mencari solusi alternatif.

Reuse-oriented Software Engineering

System design with reuse

- Kerangka sistem dirancang atau kerangka yang sudah ada digunakan kembali.
- Para desainer mempertimbangkan komponen yang digunakan kembali dan mengatur kerangka tersebut agar sesuai dengan komponen tersebut.
- Beberapa perangkat lunak baru mungkin perlu dirancang jika komponen yang dapat digunakan kembali tidak tersedia.

Reuse-oriented Software Engineering

Development and integration

- Perangkat lunak yang tidak dapat diperoleh secara eksternal dikembangkan.
- Komponen serta sistem COTS (Commercial Off-The-Shelf) diintegrasikan untuk menciptakan sistem baru.
- Integrasi (integration) sistem dalam model ini, mungkin menjadi bagian dari proses pengembangan (development) bukan aktivitas yang terpisah.

Reuse-oriented Software Engineering

- Kelebihan:
 - Dapat mengurangi biaya pembuatan modul perangkat lunak dengan model yang sama
 - Waktu pengembangan bisa jadi lebih cepat
- Kekurangan:
 - Harus dilakukan kompromi antara kebutuhan pengguna dengan produk yang ada, jika tidak, memungkinkan ketidaksesuaian produk yang eksisting dengan kebutuhan pengguna

Diskusi

STUDI KASUS 1

Sebuah perusahaan ritel ingin mengembangkan sistem manajemen inventaris untuk melacak stok barang, mengelola pemesanan, dan menghasilkan laporan. Proyek ini memiliki spesifikasi yang jelas dan tidak diharapkan ada perubahan besar selama pengembangan.

STUDI KASUS 2

Sebuah lembaga pendidikan ingin mengembangkan platform e-learning yang memungkinkan siswa untuk mengikuti kursus online, berinteraksi dengan instruktur, dan mengerjakan tugas. Proyek ini memiliki beberapa fitur yang diinginkan tetapi spesifikasinya dapat berubah seiring waktu karena umpan balik dari pengguna awal.

Diskusi

- Bagaimana Anda akan menerapkan model Waterfall dalam proyek ini?
- Apakah model Waterfall cocok untuk proyek ini?
- Bagaimana Anda akan menerapkan model Incremental dalam proyek ini?
Jelaskan bagaimana platform dapat dikembangkan dalam beberapa iterasi
- Apakah model Incremental cocok untuk proyek ini?



Any questions?