

Model Proses Perangkat Lunak

Ratna Wardani

Department of Electronic Engineering
Yogyakarta State University

Materi

- S/W Process Model
- Tahapan S/W Process
- Model Proses S/W
 - ✓ Model Waterfall
 - ✓ Model Prototype
 - ✓ Model Rapid Application Development (RAD)
 - ✓ Model Evolusioner
 - Pertambahan/Incremental
 - Spiral
 - ✓ Formal Method
 - ✓ Re-usable Model

Definisi Proses Perangkat Lunak

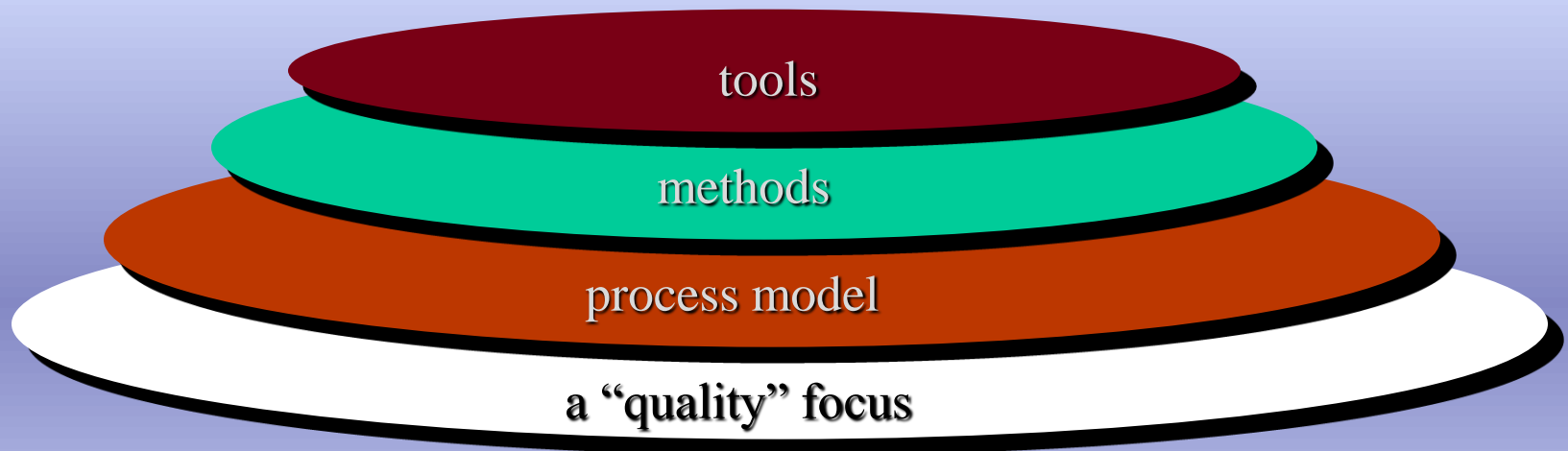
- Fritz Bauer
 - ✓ Pembangunan dan Penggunaan prinsip-prinsip rekayasa dalam rangka mendapatkan perangkat lunak yang ekonomis yang handal dan bekerja efisien pada komputer yang nyata
- IEEE (*Institute of Electrical and Electronics Engineers*)
 - ✓ **Aplikasi** pendekatan sistematis, disiplin, terkuantifikasi pada pengembangan, operasi, perawatan perangkat lunak, yaitu aplikasi rekayasa pada perangkat lunak
 - ✓ **Studi** pendekatan sistematis, disiplin, terkuantifikasi pada pengembangan, operasi, perawatan perangkat lunak

Proses Perangkat Lunak

- Tujuan:
 - ✓ Memodelkan tahapan atau aktivitas yang harus dilakukan dalam proyek pengembangan perangkat lunak
- Aplikasi:
 - ✓ Dengan mengikuti model proses, proyek pengembangan perangkat lunak harus dapat meningkatkan kualitas proses:
 - Dapat mengulang sukses terdahulu
 - Dapat di-manage → dokumentasi, distandarkan dan diorganisasikan
 - Dapat diukur → dapat dikontrol dengan pengukuran secara detail

RPL sebagai Teknologi Berlapis

Rekayasa Perangkat Lunak



- Kualitas sebagai bangunan dasar
- Proses sebagai "perekat" dan kerangka kerja
- Metode sebagai teknik pengembangan
- Tools sebagai pendukung metode dan proses

Pandangan Umum tentang RPL

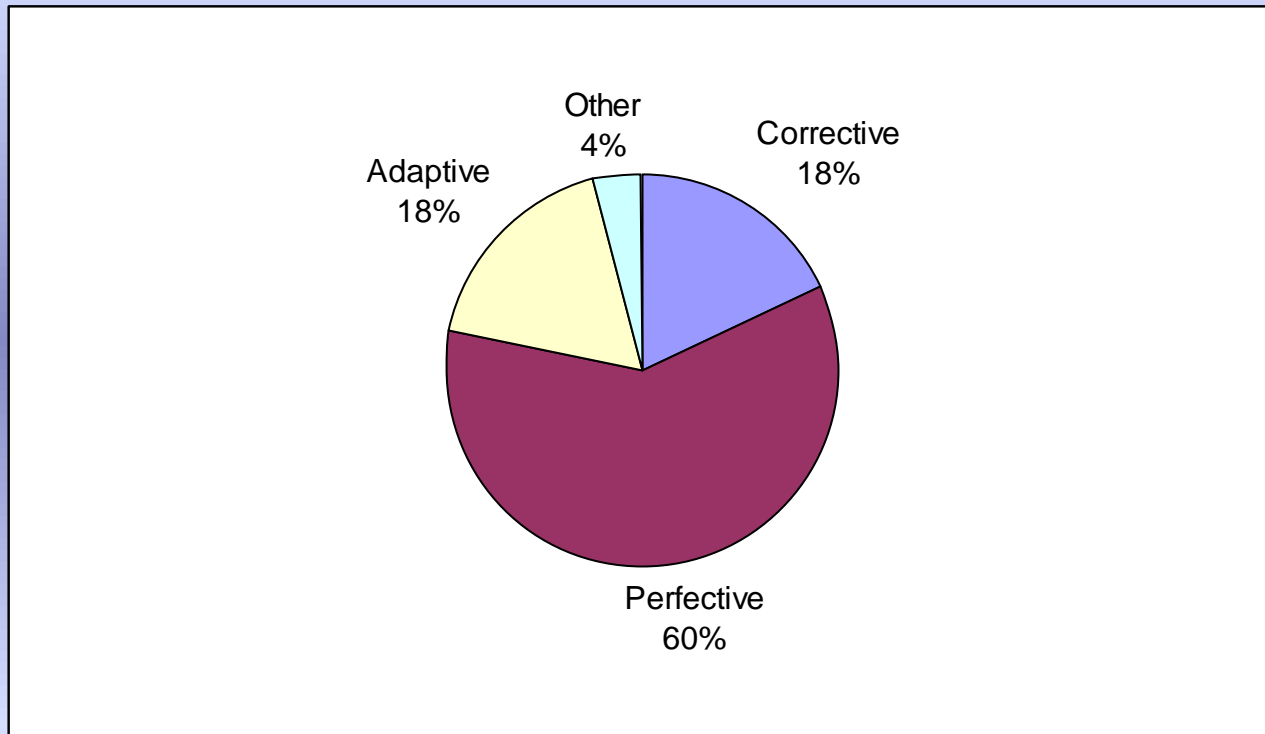
- **Rekayasa : analisis, desain, konstruksi, verifikasi, dan manajemen entitas teknis (dan sosial)**
 - ✓ Problem apa yang harus diselesaikan ?
 - ✓ Karakteristik entitas apa yang digunakan untuk menyelesaikan masalah ?
 - ✓ Bagaimana entitas (dan solusinya) direalisasikan ?
 - ✓ Bagaimana entitas di konstruksi ?
 - ✓ Pendekatan apa yang digunakan untuk menemukan kesalahan yang dibuat pada desain dan konstruksi entitas?
 - ✓ Bagaimana entitas didukung dalam jangka panjang, dimana koreksi, adaptasi, dan peningkatan selalu diminta pengguna pada entitas

Tahapan Umum Model Proses

- Fase definisi, fokus pada pertanyaan "apa"
- Fase pengembangan, fokus pada pertanyaan "bagaimana"
- Fase pemeliharaan, fokus pada "perubahan" :
 - ✓ Koreksi / **corrective maintenance** → memperbaiki kerusakan yang ditemukan
 - ✓ Adaptasi / **Adaptive maintenance** → beradaptasi terhadap perubahan lingkungan eksternal
 - ✓ Peningkatan / **Prefective maintenance** → perluasan kebutuhan fungsional original
 - ✓ Pencegahan / **preventive maintenance** → perubahan S/W agar mudah dikoreksi, disesuaikan dan dikembangkan

Pemeliharaan / Maintenance

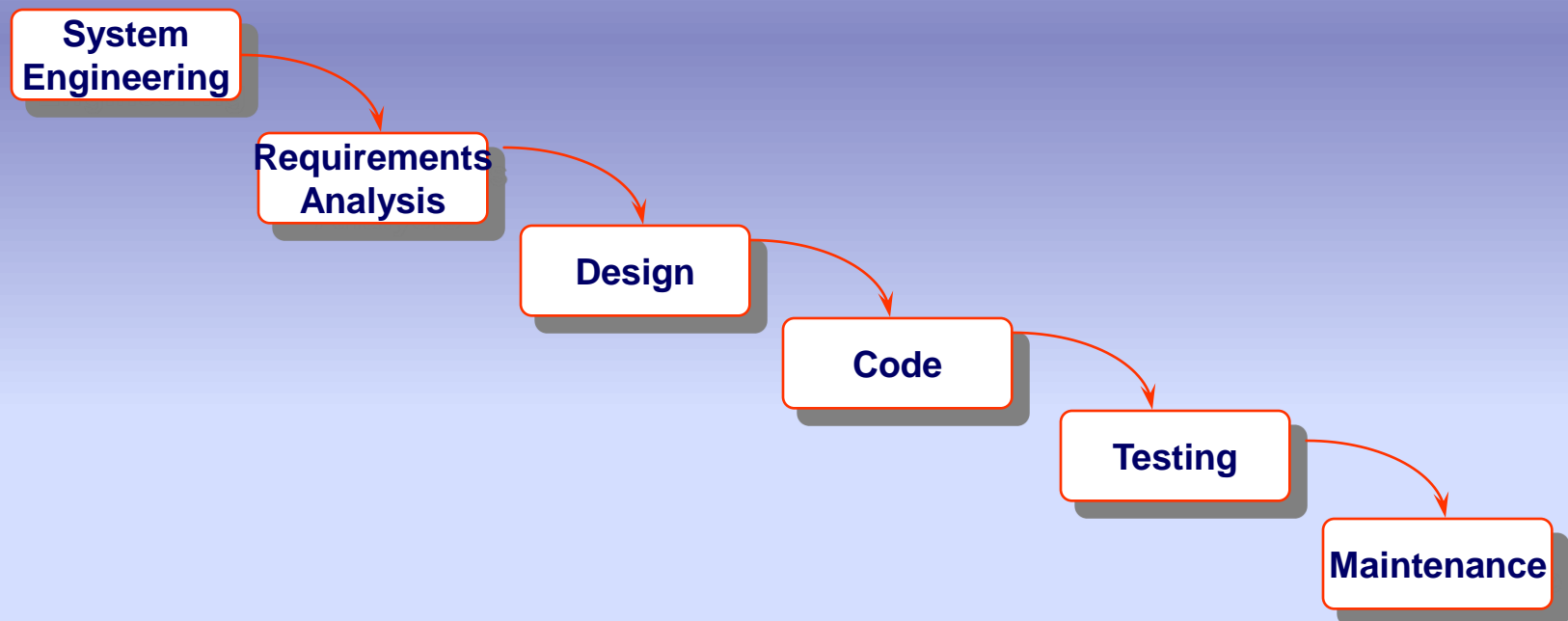
- Effort untuk pengembangan S/W dalam konteks Pemeliharaan



Model Proses RPL

■ Waterfall

- ✓ Sekuensial linier, SDLC (System Development Life Cycle)
- ✓ Pendekatan tradisional dan paling luas dipakai
- ✓ Meski ada kelemahan, tapi menjadi dasar model proses RPL



Model Proses RPL

- Waterfall
 - ✓ **System Engineering**: membangun persyaratan semua elemen sistem
 - ✓ **Requirement Analysis** : identifikasi kebutuhan perangkat lunak
 - ✓ **Design** : menerjemahkan kebutuhan ke representasi perangkat lunak (struktur data, arsitektur PL, representasi interface, detail prosedural/algoritma)
 - ✓ **Code** : menerjemahkan desain dalam bahasa mesin
 - ✓ **Testing** : pengujian fungsionalitas perangkat lunak
 - ✓ **Maintenance** : perbaikan maupun perubahan yang diperlukan untuk mengakomodasi kebutuhan pengguna

Model Proses RPL

■ Waterfall

✓ Kelemahan:

- Jarang sebuah proyek nyata mengikuti aliran sekuensial
- Kesulitan pengguna dalam menyatakan kebutuhan di awal proses, sementara model ini membutuhkan spesifikasi kebutuhan di awal proses
- Perlu kesabaran pengguna hingga proses selesai secara keseluruhan
- Blocking state → tim proyek harus menunggu tim lain menyelesaikan tugasnya yang saling ketergantungan

Model Proses RPL

■ Prototyping

- ✓ Kesulitan pengguna mendefinisikan input, proses, output yang diminta secara detail
- ✓ Developer tidak yakin terhadap efisiensi algoritma, kemampuan adaptasi terhadap sistem operasi, atau bentuk interaksi mesin dengan orang

Prototyping (Tahapan)

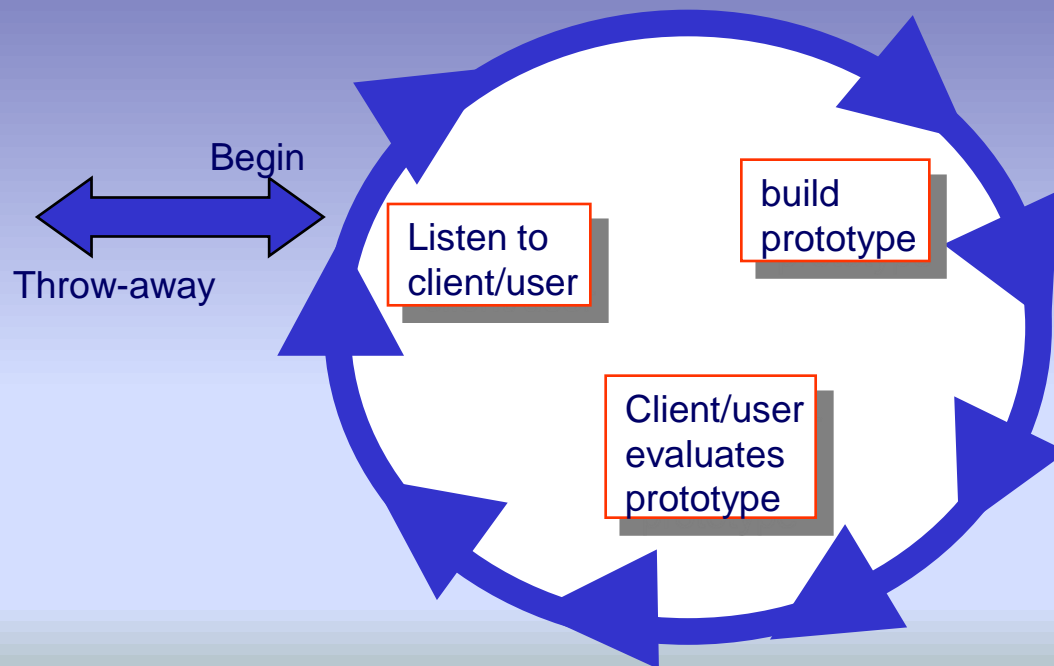
- Pengumpulan kebutuhan dan perbaikan (awal)
 - Quick Design
 - Pembentukan Prototype
 - Evaluasi Pelanggan
 - Perbaikan Prototype
 - Produk S/W (Akhir)
- } (Iterasi)

Model Proses RPL

■ Prototyping

✓ Kelemahan:

- Customer melihat prototipe tersebut sebagai versi dari software.
- Developer membuat implemetasi yang kompromitas dengan tujuan untuk memperoleh prototipe pekerjaan secara cepat



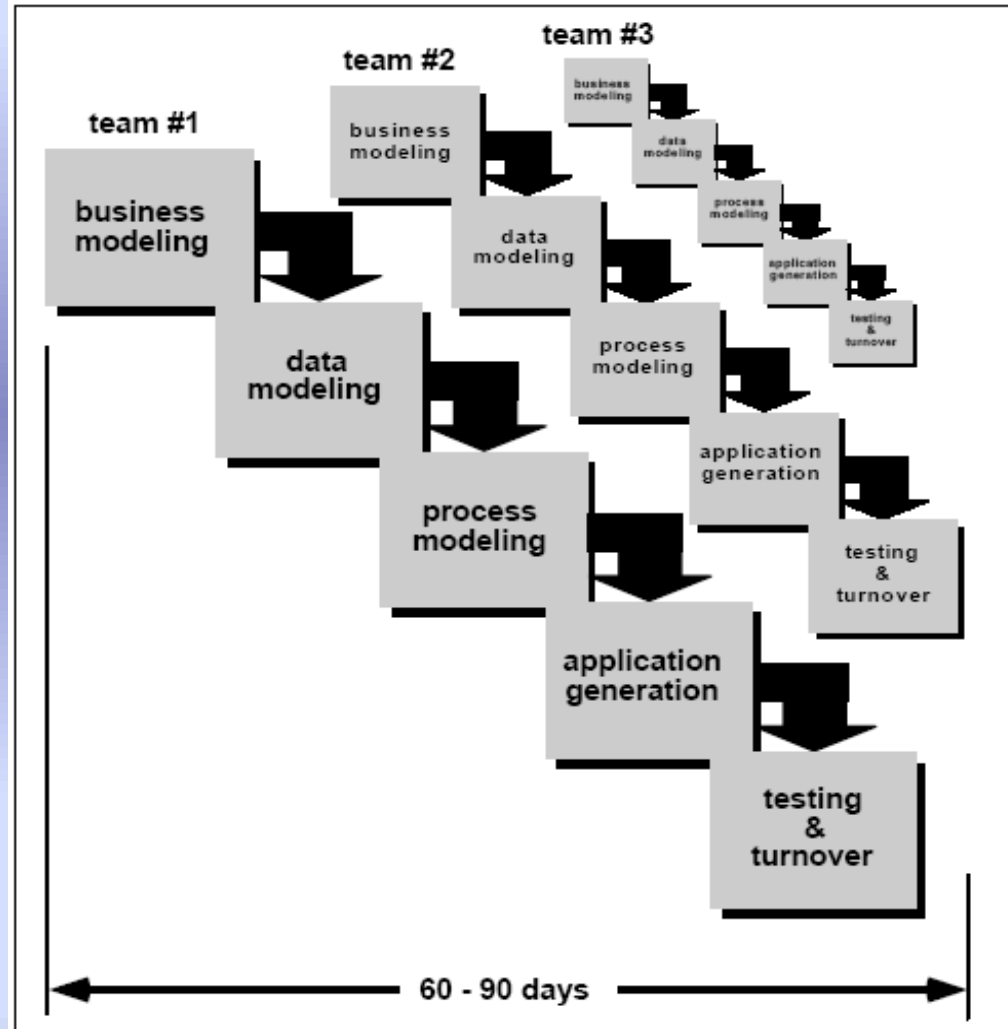
Model Proses RPL

▪ RAD

- ✓ model proses pembangunan PL yang incremental.
- ✓ menekankan pada siklus pembangunan yang pendek/singkat.
- ✓ mengadopsi model waterfall dan pembangunan dalam waktu singkat dicapai dengan menerapkan component based construction.
- ✓ waktu yang singkat adalah batasan yang penting untuk model ini.
- ✓ jika kebutuhan lengkap dan jelas maka waktu yang dibutuhkan untuk menyelesaikan secara komplit software yang dibuat adalah misalnya 60 sampai 90 hari

Model Proses RPL

- RAD
 - ✓ Model:



Model Proses RPL

▪ RAD

✓ Kelemahan:

- Tidak cocok untuk proyek skala besar
- Proyek bisa gagal karena waktu yang disepakati tidak dipenuhi
- Sistem yang tidak bisa dimodularisasi tidak cocok untuk model RAD
- Proyek dengan resiko teknis yang tinggi kurang cocok untuk model RAD

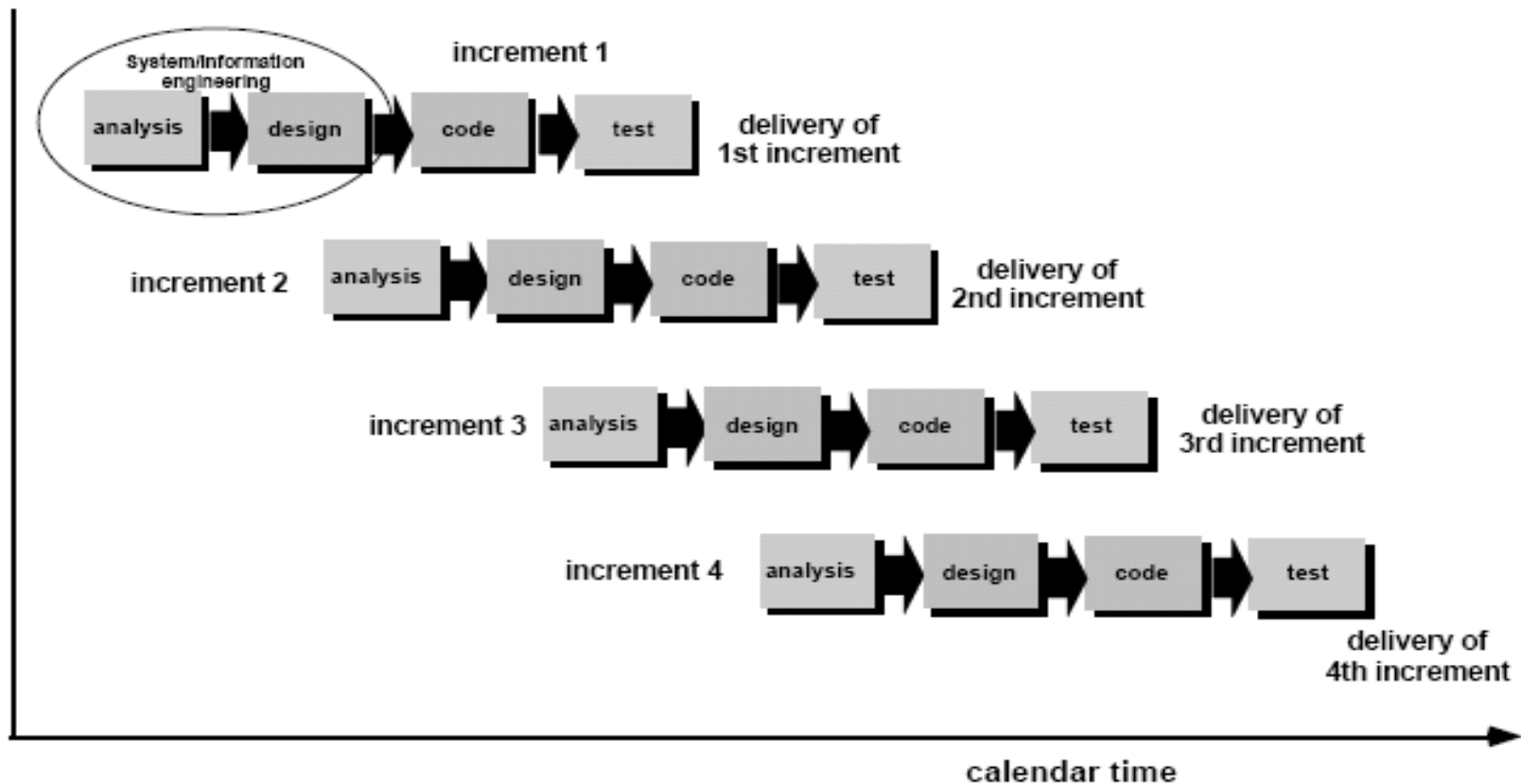
Model Proses RPL

▪ Incremental

- ✓ Pengembangan sistem berdasarkan model sistem yang dipecah sehingga model pengembangannya secara increment/bertahap.
- ✓ Kebutuhan pengguna diprioritaskan dan prioritas tertinggi dimasukkan dalam awal increment
- ✓ Mengkombinasikan elemen-elemen dari waterfall dengan sifat iterasi/perulangan.
- ✓ Produk yang dihasilkan pada increment pertama bukanlah prototype, tapi produk yang sudah bisa berfungsi dengan spesifikasi dasar.
- ✓ Mampu mengakomodasi perubahan secara fleksibel

Model Proses RPL

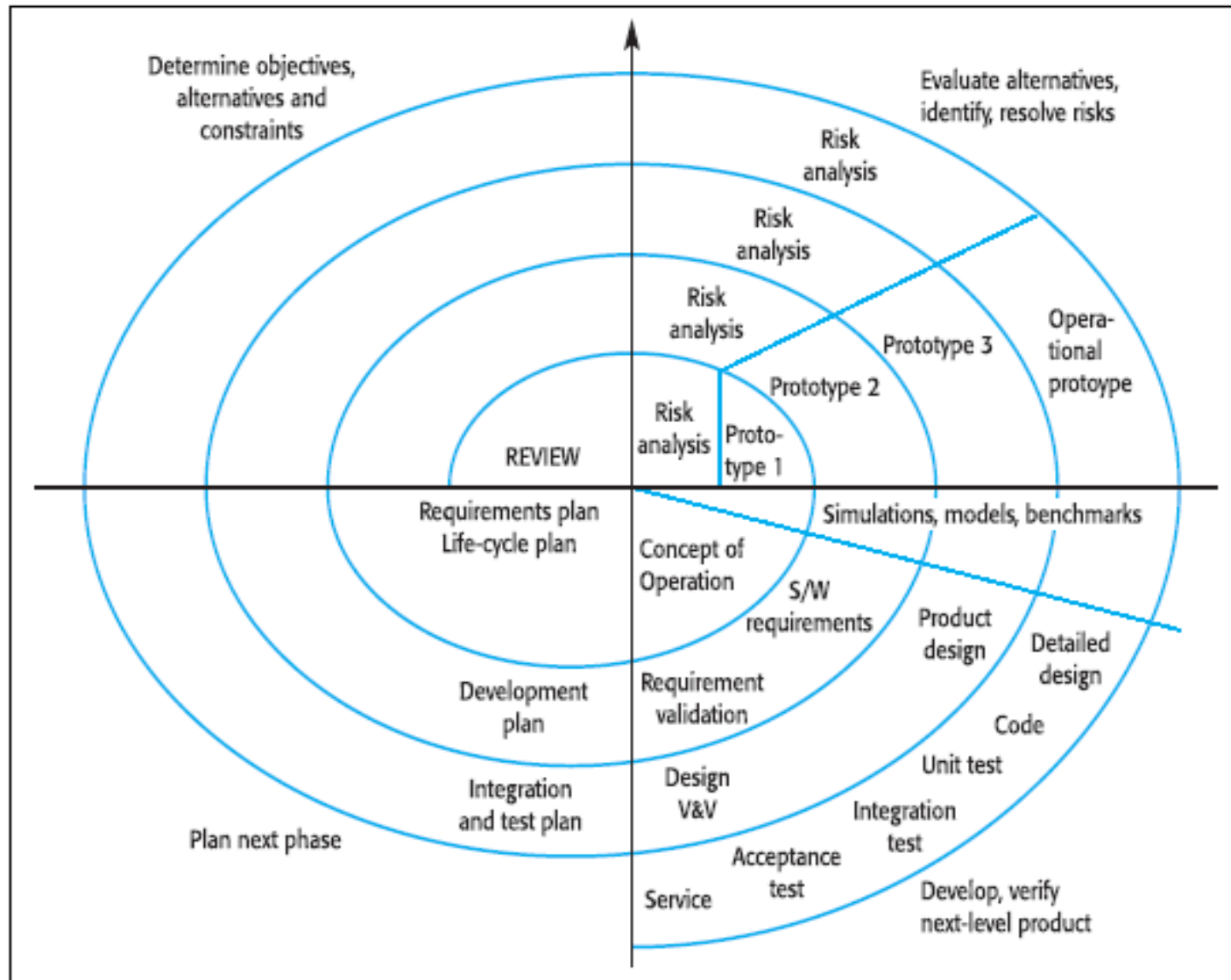
Evolutionary Model (Incremental)



Model Proses RPL

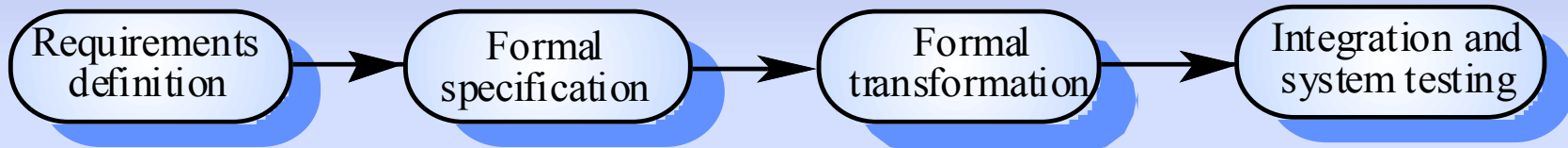
- Spiral
 - ✓ Proses digambarkan sebagai spiral.
 - ✓ Setiap loop mewakili satu fase dari software process.
 - ✓ Loop paling dalam berfokus pada kelayakan dari sistem, loop selanjutnya tentang definisi dari kebutuhan, loop berikutnya berkaitan dengan desain sistem dan seterusnya

Model Proses RPL



Model Proses RPL

- Pengembangan Sistem Formal
 - ✓ Spesifikasi matematis perangkat lunak yang secara formal diterjemahkan ke dalam implementasi
 - ✓ Memungkinkan pereayasa PL mengkhususkan, mengembangkan dan memverifikasi sistem dengan notasi matematis yang tepat
 - ✓ Mengurangi ambiguitas, ketidaklengkapan dan ketidak-konsistenan
 - ✓ Bahasa Z adalah salah satu tools untuk spesifikasi formal



Model Proses RPL

▪ Pengembangan Sistem Formal

✓ Kelemahan:

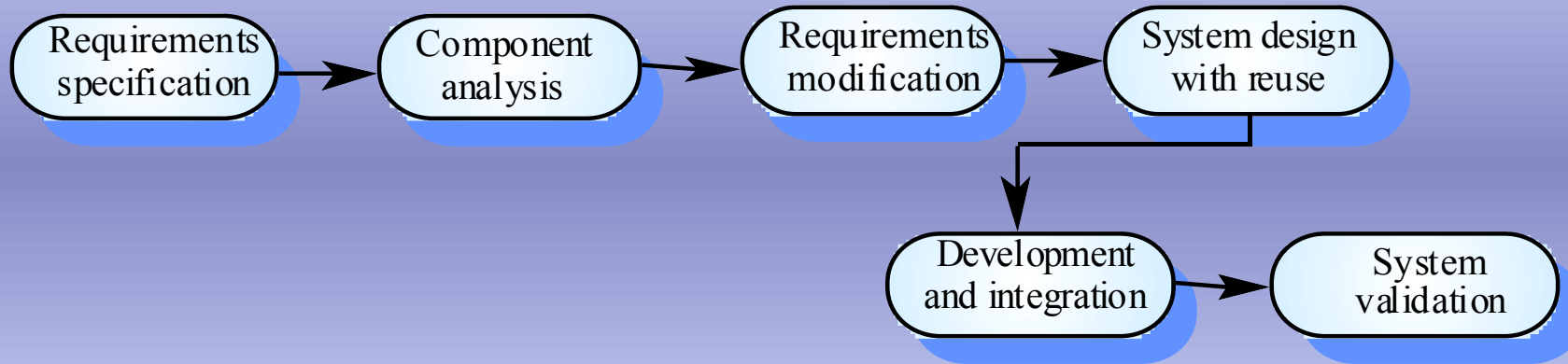
- Memerlukan waktu yang lama dan mahal
- Memerlukan keahlian khusus dan pelatihan untuk mengaplikasikannya
- Sulit untuk mengkomunikasikan dengan pemakai
- Untuk sebagian besar sistem, metode ini tidak memberikan keuntungan biaya atau kualitas yang signifikan dibandingkan dengan pendekatan yang lain.

Model Proses RPL

- Pengembangan Berorientasi Re-Usable
 - ✓ Sistem dibangun dari komponen yang sudah ada
 - ✓ Bergantung pada sejumlah besar komponen perangkat lunak yang dapat dipakai ulang, yang bisa didapat, dan berapa kerangka kerja integrasi untuk komponen-komponen ini.
 - ✓ Komponen-komponen ini disebut *COTS (Commercial Off-The-Shelf Systems/Sistem Siap Beli Komersial)* yang dapat digunakan untuk memberikan fungsionalitas khusus seperti format teks, perhitungan numerik,dll.

Model Proses RPL

- Pengembangan Berorientasi Re-Usable



Model Proses RPL

- Pengembangan Berorientasi Re-Usable
 - ✓ Keuntungan :
 - Mengurangi besarnya perangkat lunak yang akan dikembangkan
 - Memperkecil biaya dan resiko
 - Memungkinkan penyelesaian perangkat lunak dengan cepat