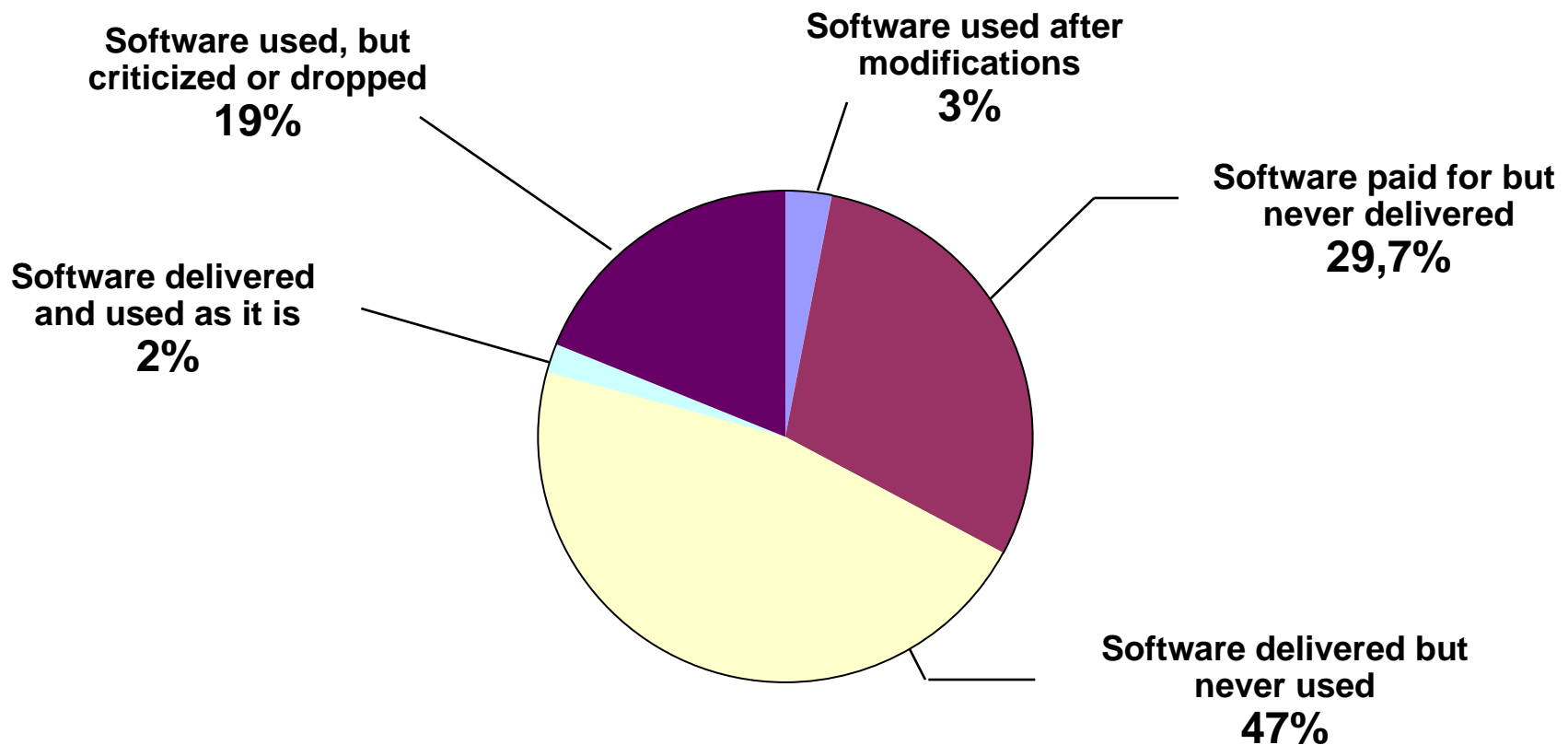

System Development Life Cycle (SDLC)

SI-215

Analisa & Desain Sistem Informasi I

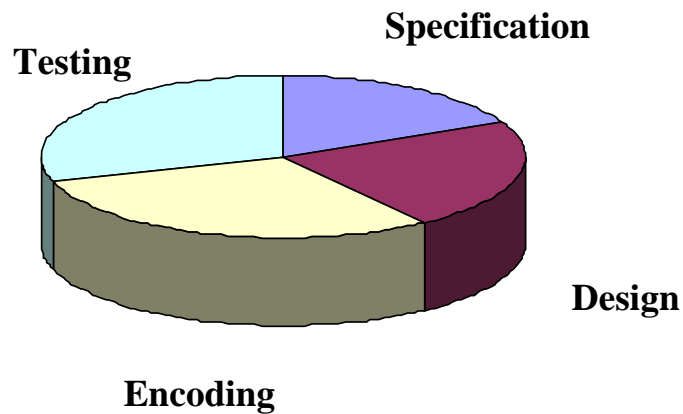
Rosa Ariani Sukamto

Permasalahan Perangkat Lunak

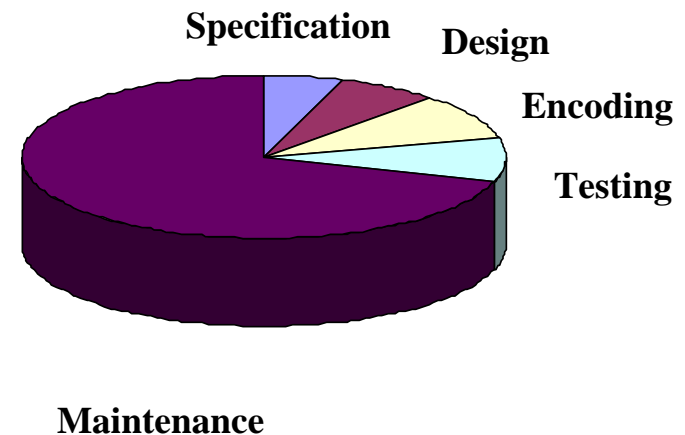


Permasalahan Perangkat Lunak (2)

**Distribution of effort :
what is believed**



**Distribution of effort:
what happens**



Fase SDLC

- Identifikasi, seleksi, dan perencanaan sistem
- Analisis sistem
- Desain sistem
- Implementasi sistem
- Pemeliharaan sistem (maintenance)

Identifikasi, Seleksi, dan Perencanaan Sistem

- Mengidentifikasi kebutuhan user
- Menyeleksi kebutuhan user dari proses identifikasi dengan melihat kapasitas teknologi dan efisiensi
- Merencanakan sistem

Kebutuhan

- Kebutuhan fungsional and non-fungsional
 - Non-fungsional : Sistem bisa menjadi tidak digunakan jika tidak dipenuhi
 - Menggunakan CASE (Computer Aided Software Engineering) tools, bahasa pemrograman tertentu
 - Menggunakan bahasa tertentu
- Kebutuhan User (customer)
- Kebutuhan Sistem (kontrak dengan klien)
- Kebutuhan dokumen dan perangkat lunak (developer)

Analisis Sistem

- Permodelan data
 - Entity Relationship Diagram (ERD)
 - Conceptual Data Model (CDM)
 - Physical Data Model (PDM)
- Permodelan proses
 - Unified Modeling Language (UML)

Desain Sistem

- Desain form dan laporan (*report*)
- Desain Antarmuka dan dialog (*message*)
- Desain basis data dan file (*framework*)
- Desain proses (struktur proses)

Implementasi Sistem

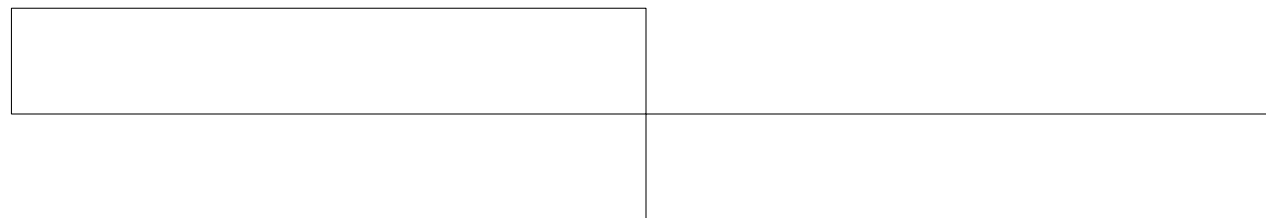
- Pemrograman dan pengetesan perangkat lunak (*software*)
 - Developmental (*error testing* per modul oleh *programmer*)
 - Alpha testing (*error testing* ketika sistem digabungkan dengan antarmuka *user* , oleh *software tester*)
 - Beta testing (*testing* dengan lingkungan dan data sebenarnya)
- Konversi sistem
 - Mengaplikasikan perangkat lunak pada lingkungan yang sebenarnya untuk digunakan oleh organisasi
- Dokumentasi
- Pelatihan

Metode Konversi Sistem

- Konversi paralel



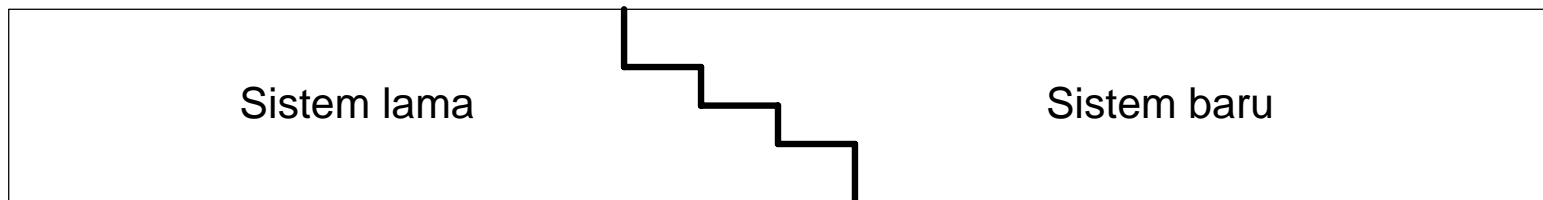
- Konversi langsung



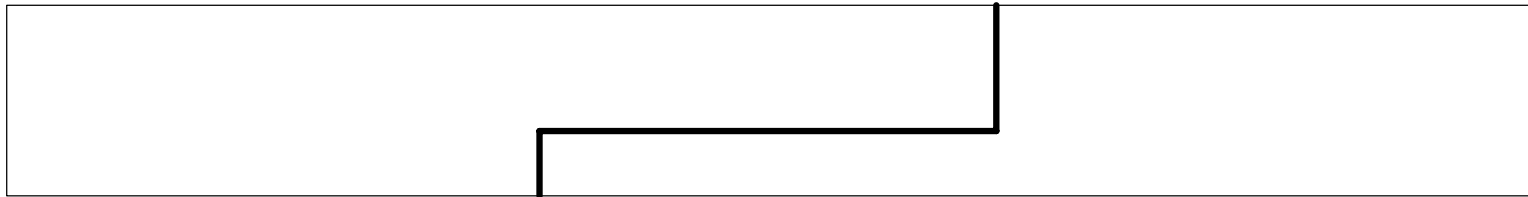
Sistem lama

Metode Konversi Sistem (2)

- Konversi per fase



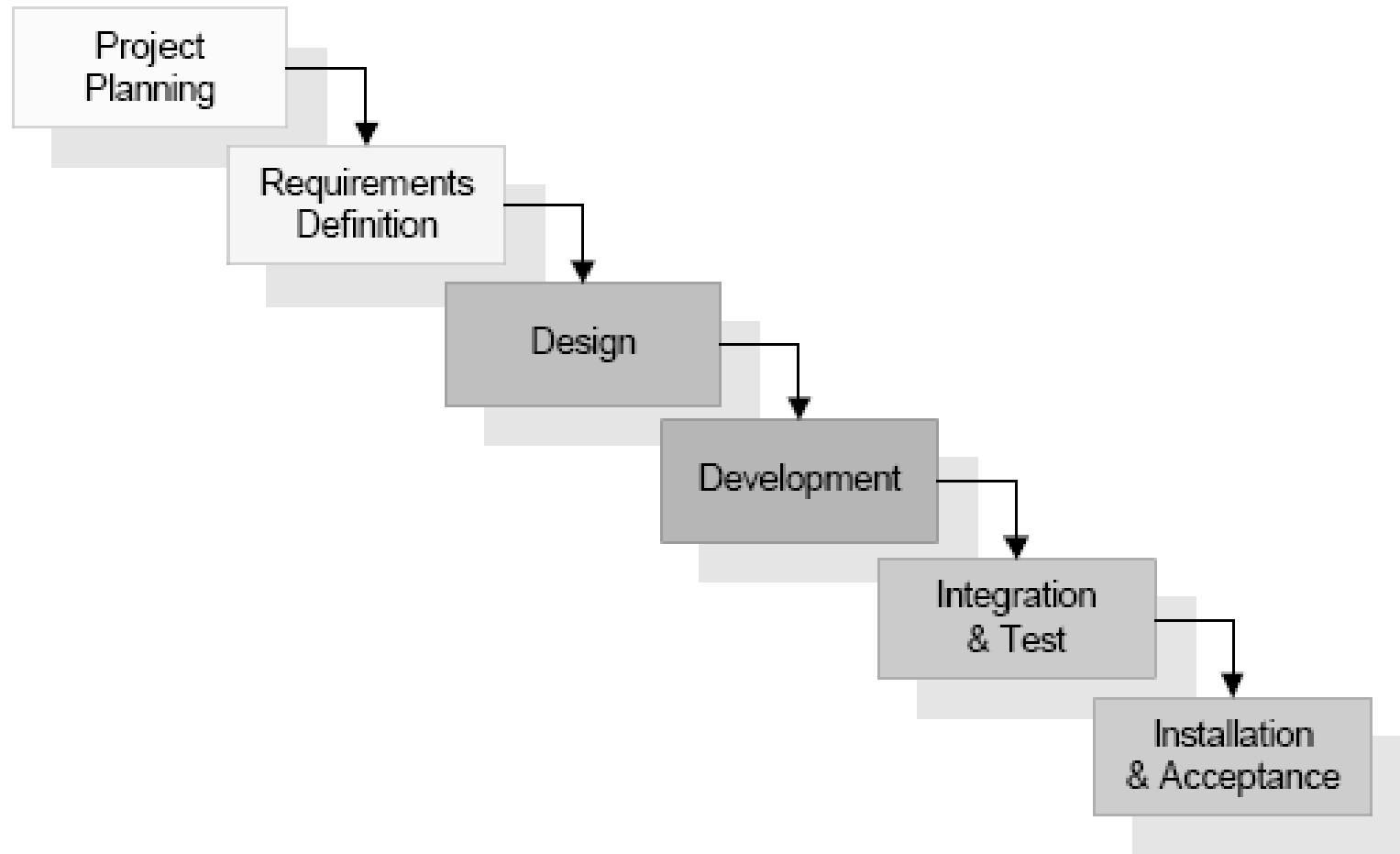
- Konversi pilot (*single location*)



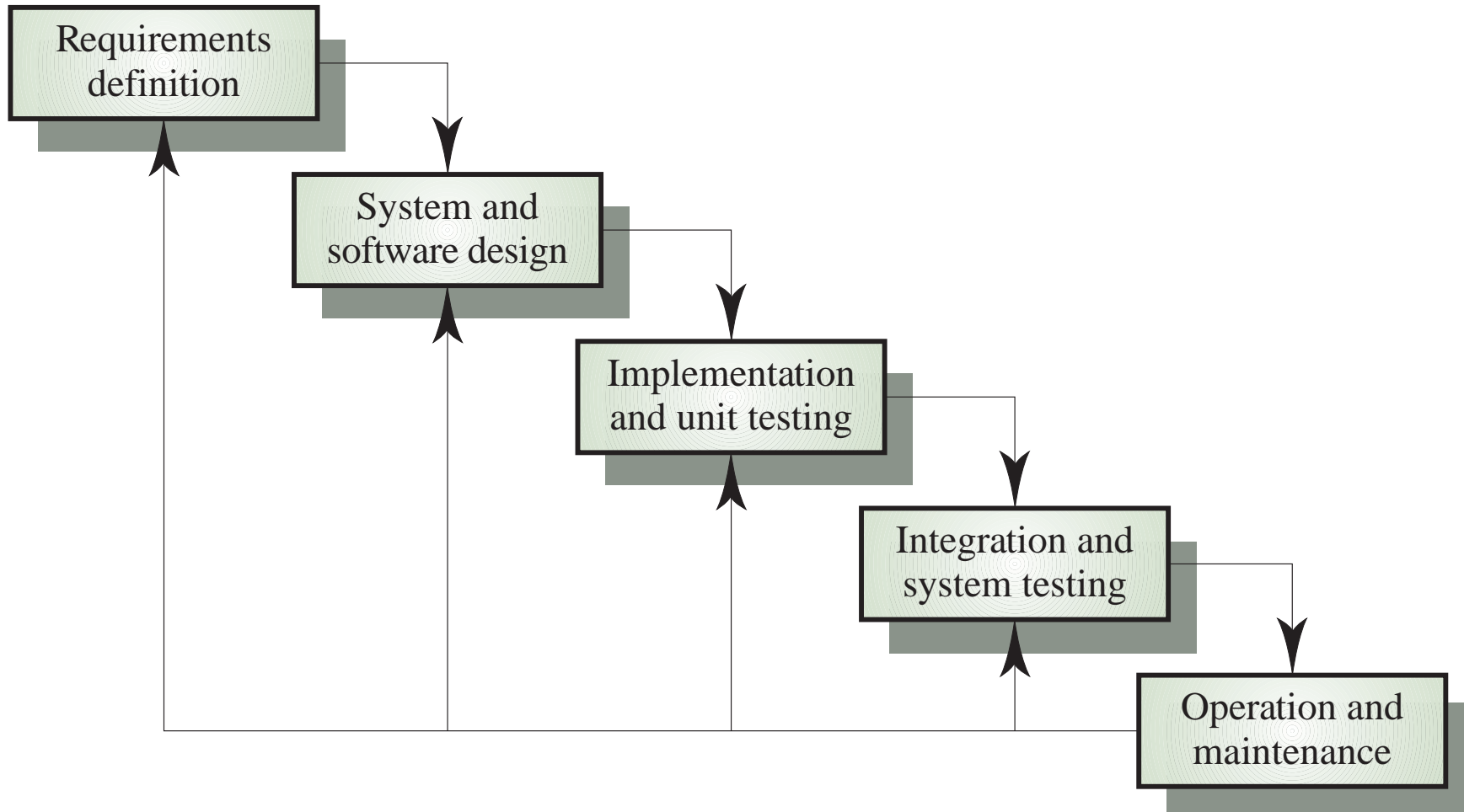
Pemeliharaan Sistem

- **Corrective** – memperbaiki desain dan *error* pada program
- **Adaptive** – memodifikasi sistem untuk beradaptasi dengan perubahan lingkungan
- **Perfective** – Melibatkan sistem untuk menyelesaikan masalah baru atau mengambil kesempatan (penambahan fitur)
- **Preventive** – Menjaga sistem dari kemungkinan masalah di masa yang akan datang

SDLC Waterfall



SDLC Waterfall (2)



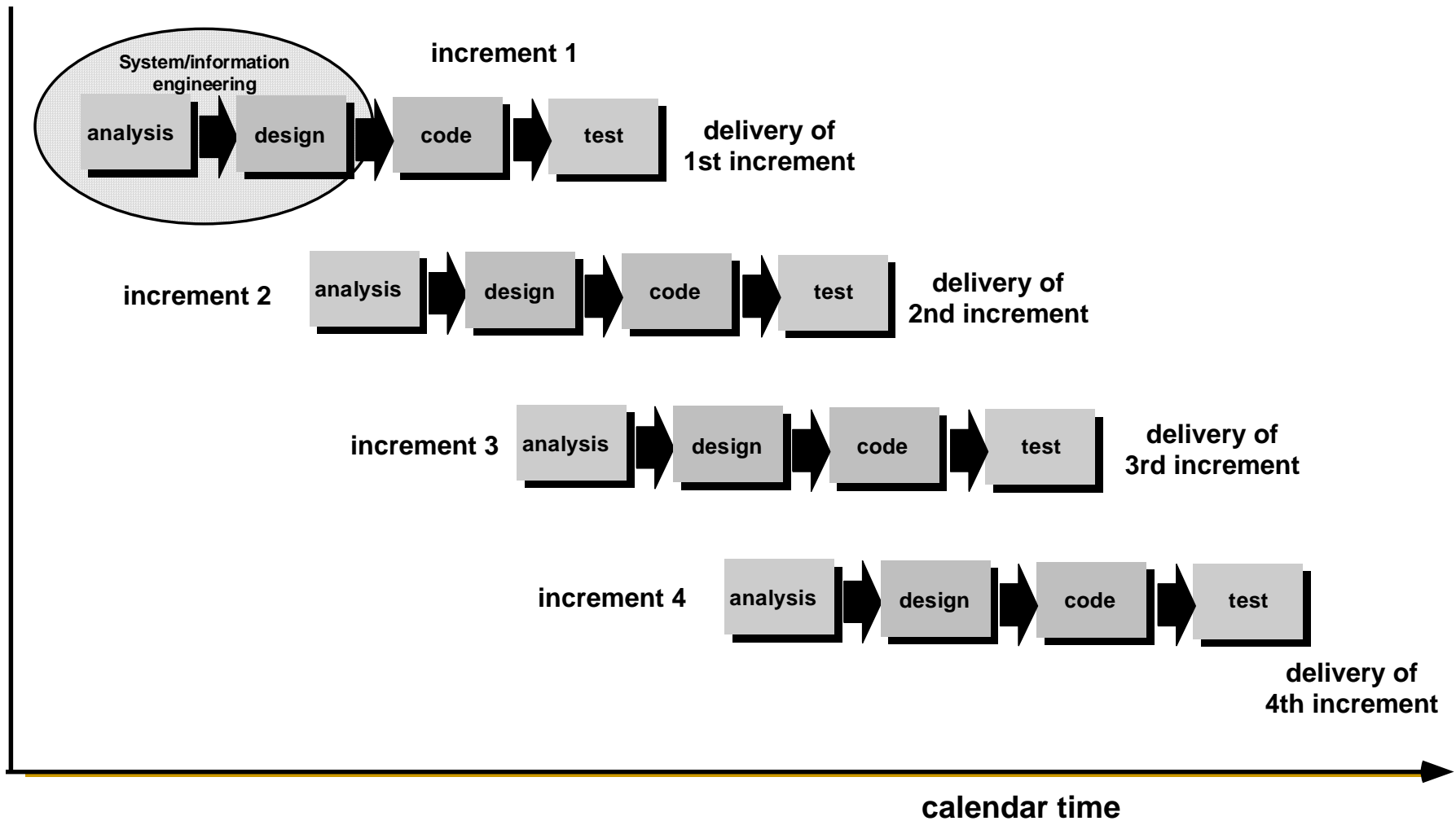
SDLC Waterfall(3)

- Kelebihan
 - Proses-prosesnya mudah dipahami dan jelas
 - Mudah dalam pengelolaan proyek
 - Dokumen dihasilkan setiap akhir fase
 - Sebuah fase dijalankan setelah fase sebelumnya selesai
 - Struktur sistem jelas
 - Kondisi tepat SDLC Waterfall
 - Kebutuhan user telah sangat dipahami
 - Kemungkinan terjadinya perubahan kebutuhan user kecil
- Kelemahan
 - Proyek dunia nyata jarang mengikuti alur proses
 - Kesulitan jika terjadi perubahan kebutuhan
 - Waktu pengerjaan bertambah
 - Ada anggota tim yang harus menunggu pekerjaan pekerja lain
 - Kesabaran customer/klien

SDLC - Incremental Model

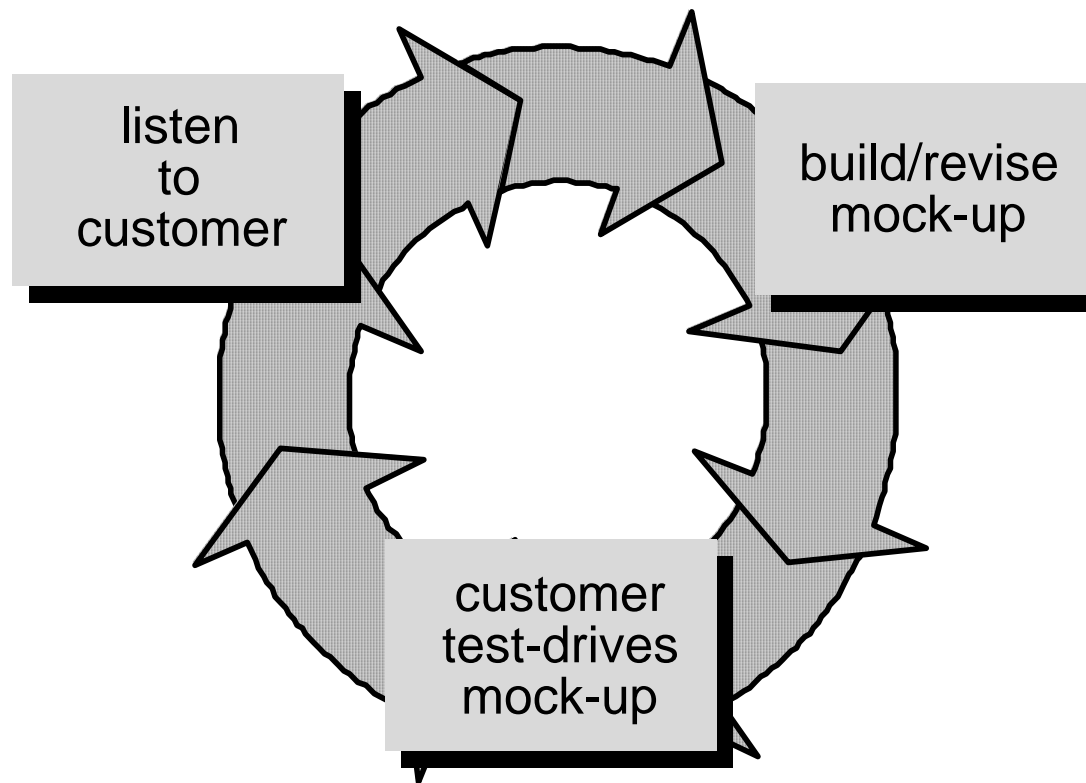
- Memberikan implementasi sebagai inisial proses kepada user untuk mendapatkan masukan balik
- Pengembangan dengan terus melakukan eksplorasi
- Per prototipe
- Permasalahan
 - Batasan proses tidak jelas
 - Sistem kurang terstruktur
- Kemampuan aplikasi
 - Untuk sistem dengan interaksi skala kecil dan medium
 - Untuk antarmuka user
 - Untuk sistem dengan masa penggunaan pendek

SDLC - Incremental Model (2)



SDLC Prototypes

- Membuat sebuah contoh prototipe untuk menunjukkan kebutuhan dan desain ke pemakai



SDLC Prototypes

■ Kelemahan

- ❑ Harus ada versi yang dapat dijalankan sebagai prototipe sebelum sistem dikembangkan (bisa berupa contoh sistem lain)
- ❑ Harus ada implementasi sistem yang dikembangkan sebelum dibuat sebuah sistem final

SDLC Spiral Life Cycle

- Mendefinisikan kebutuhan dengan sedetail mungkin
- Pembuatan desain untuk sistem yang baru
- Pembuatan prototipe dari pembuatan desain, pembuatan prototipe selanjutnya berdasarkan evaluasi prototipe sebelumnya
- Proses prototipe dilakukan berulang-ulang sampai *customer* puas
- Sistem dibuat berdasarkan prototipe yang memuaskan *customer*
- Sistem di tes dan dievaluasi

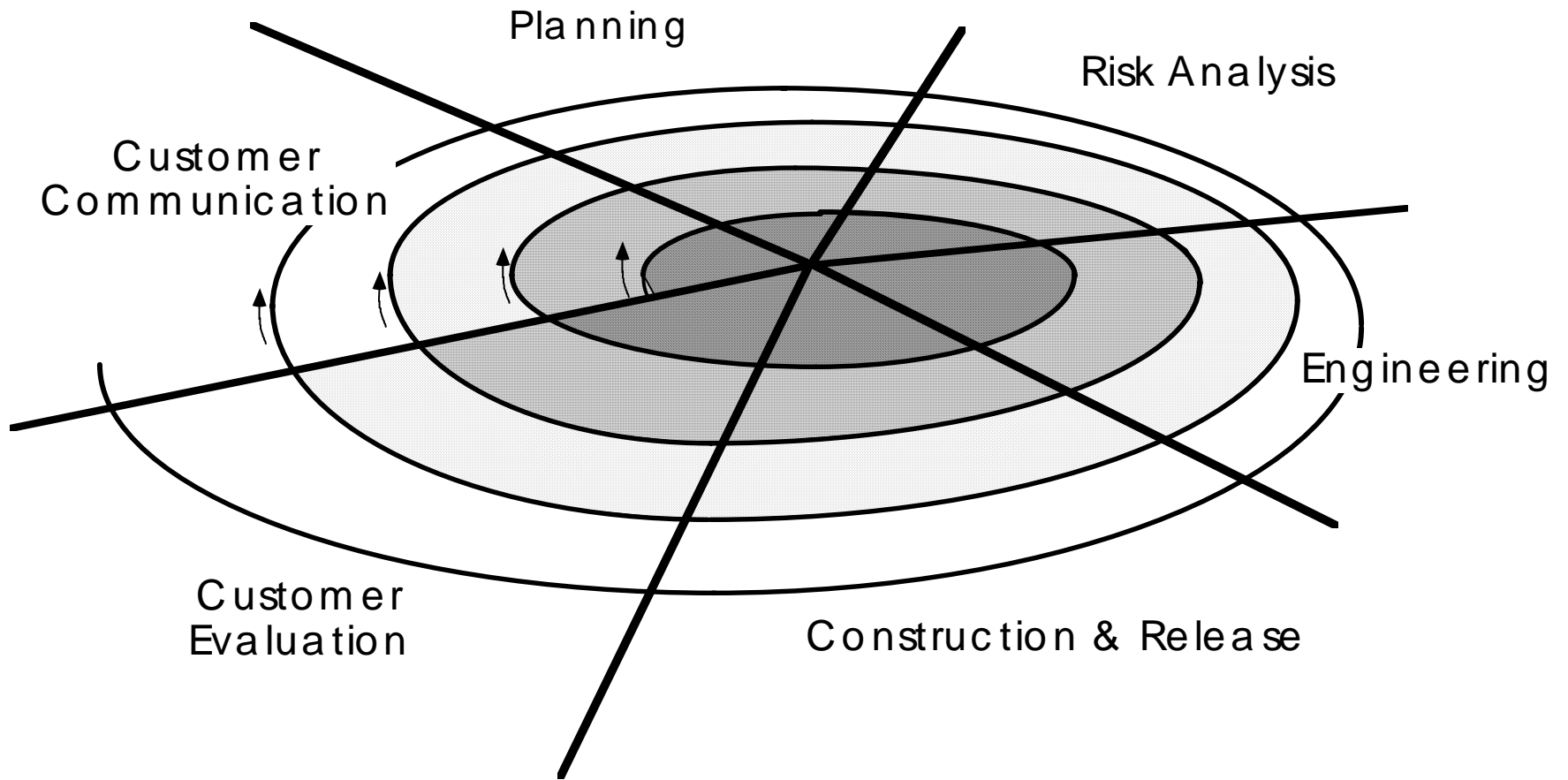
Kelebihan

- Dapat digunakan untuk sistem yang besar
- Sangat cocok sebagai mekanisme mengurangi resiko

Kelemahan

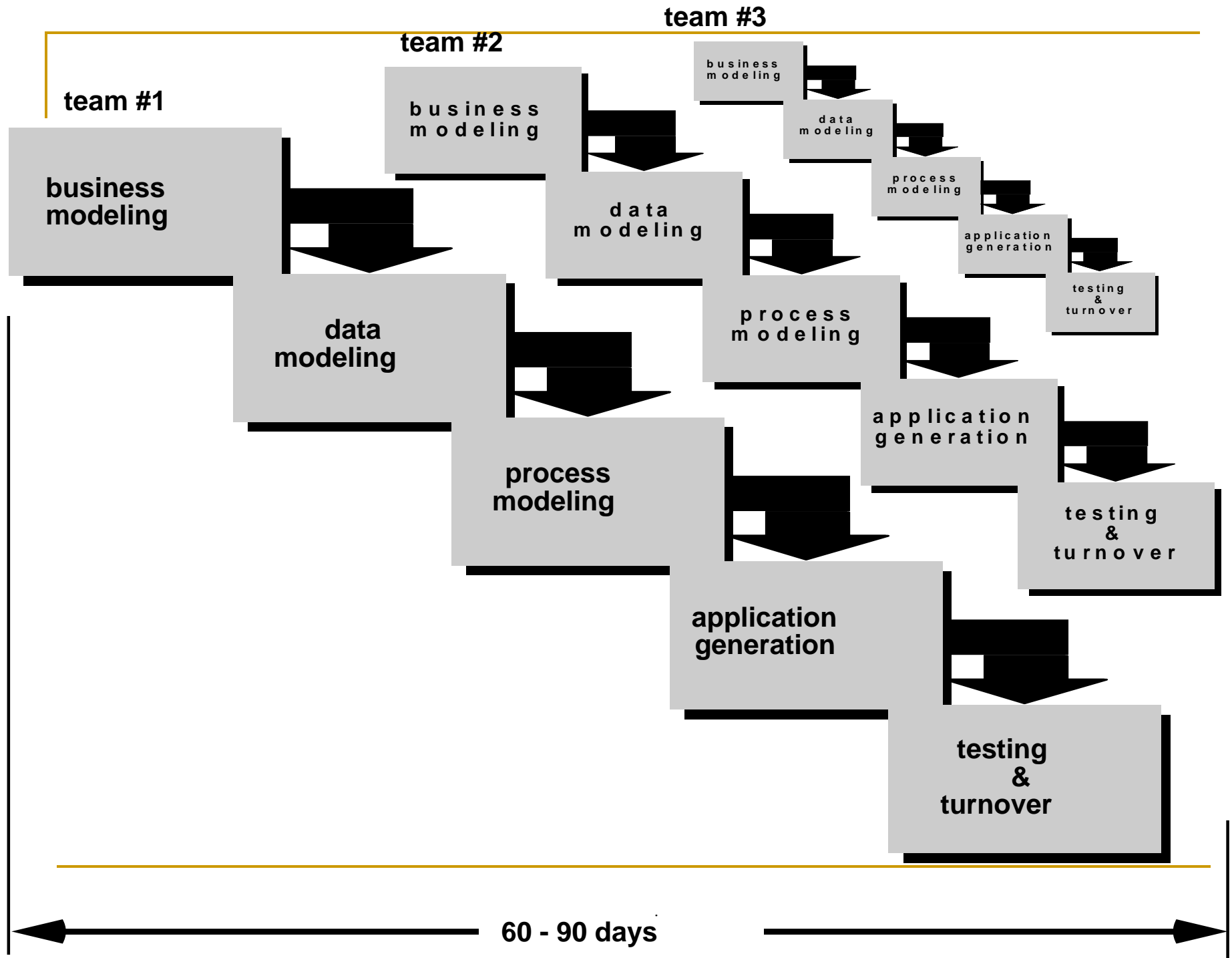
- Terlalu banyak memikirkan resiko yang akan terjadi
- Masih jarang digunakan

SDLC Spiral Life Cycle



SDLC Rapid Application Development (RAD)

- Mengumpulkan spesifikasi menggunakan workshop atau group khusus
- Melakukan tes berulang-ulang oleh user terhadap desain yang diawali dengan prototipe
- Menggunakan kembali komponen perangkat lunak yang ada
- Jadwal yang ketat terhadap perbaikan desain produk versi selanjutnya
- Komunikasi yang tidak terlalu formal antar anggota tim
- Kelebihan
 - Waktu pengembangan singkat
- Kelemahan
 - Untuk proyek besar memerlukan lebih banyak sumber daya
 - Sangat memerlukan kerjasama antara customer dan developer
 - Tidak cocok untuk kebutuhan yang tidak dapat dimodulkan
 - Tidak cocok untuk sistem yang memerlukan banyak perbaikan
 - Tidak sesuai untuk pengembangan sistem dengan resiko tinggi (aplikasi dengan teknologi baru)



Sebab Gagalnya Perangkat Lunak

- Perencanaan yang tidak realistik karena terlalu banyak kasus dan pemikiran optimis
- Penelusuran yang tidak efektif
- Terlalu terpaku pada kebutuhan sementara
- Resiko