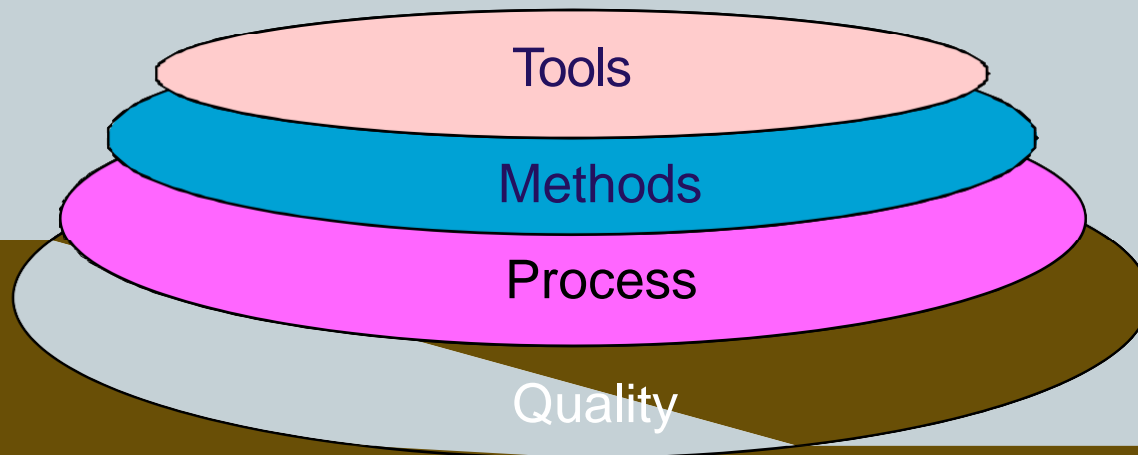


3. The Software Process

3.1 Software Engineering Layers

2



3.3 General Phase to Software Engineering

3

- Defiuitiou phase ⇒ berfokus pada 'apa' (what):
 - informasi yang akan diproses
 - fungsi dan perfomauce yang dibutuhkan
 - tingkah laku sistem yang diharapkan
 - iuteface yang akan dibangun
 - batasan sistem yang sukses
- Developmeut phase ⇒ berfokus pada 'bagaimana' (how):
 - data dikonstruksikan
 - fungsi-fungsi diimplementasikan
 - detail prosedur akan diimplementasikan
 - iuteface dikarakterisasi
 - rancangan akan diterjemahkan ke dalam pemrograman
 - pengujian dilakukan
- Maiuteuauce phase ⇒ berfokus pada 'perubahan' (change):
 - dihubungkan dengan koreksi kesalahan
ketika lingkungan perangkat lunak berkembang
 - sehubungan dengan perubahan kebutuhan pelanggan

3.4 Changes in Phase Development

4

- Correction (Koreksi)
 - membetulkan cacat atau kerusakan
- Adaptation (Adaptasi)
 - modifikasi perangkat lunak karena perubahan kebutuhan fungsional original (CPU, OS, aturan bisnis, karakteristik produk eksternal, dll)
- Enhancement (Perkembangan)
 - memperluas perangkat lunak sehingga melampaui kebutuhan fungsi originalnya
- Prevention (Pencegahan)
 - pencegahan sebagai antisipasi perubahan karena usia perangkat lunak

3.5 Umbrella Activities

5

- Software project tracking and control
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Document preparation and production
- Reusability management
- Measurement
- Risk management

3.6 Software Development Stages

6

- Requirements Analysis & Specification
- Conceptual/System Design
- Detailed/Program Design
- Implementation/Coding
- Unit & Integration Testing
- System Testing
- System Delivery
- Maintenance

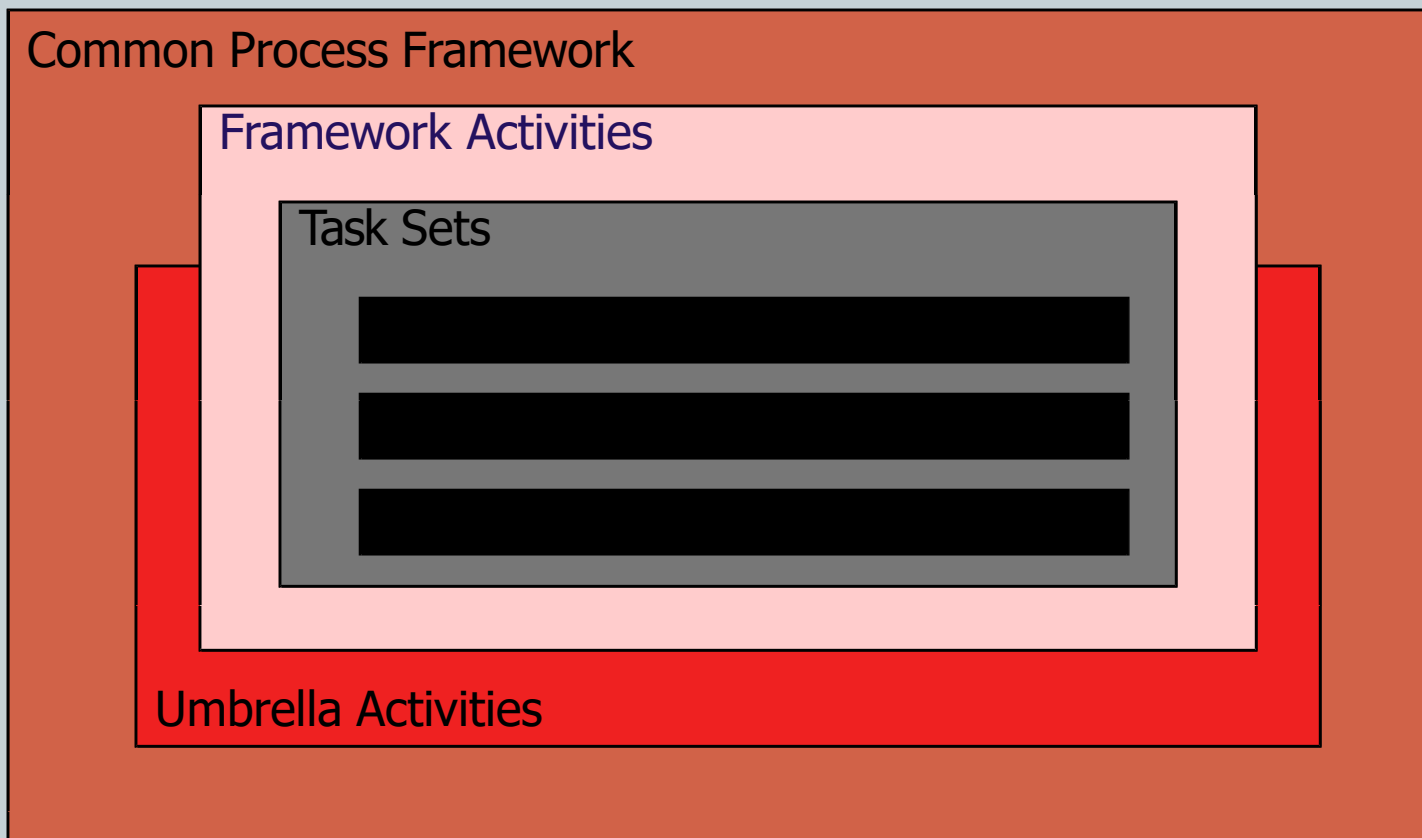
3.7. The Software Process

7

- Kerangka kerja untuk tugas-tugas yang dibutuhkan untuk membangun perangkat lunak dengan kualitas tinggi
- Strategi pengembangan PL yang melingkupi lapisan proses, metode dan alat bantu yang digunakan

3.7. The Software Process

8



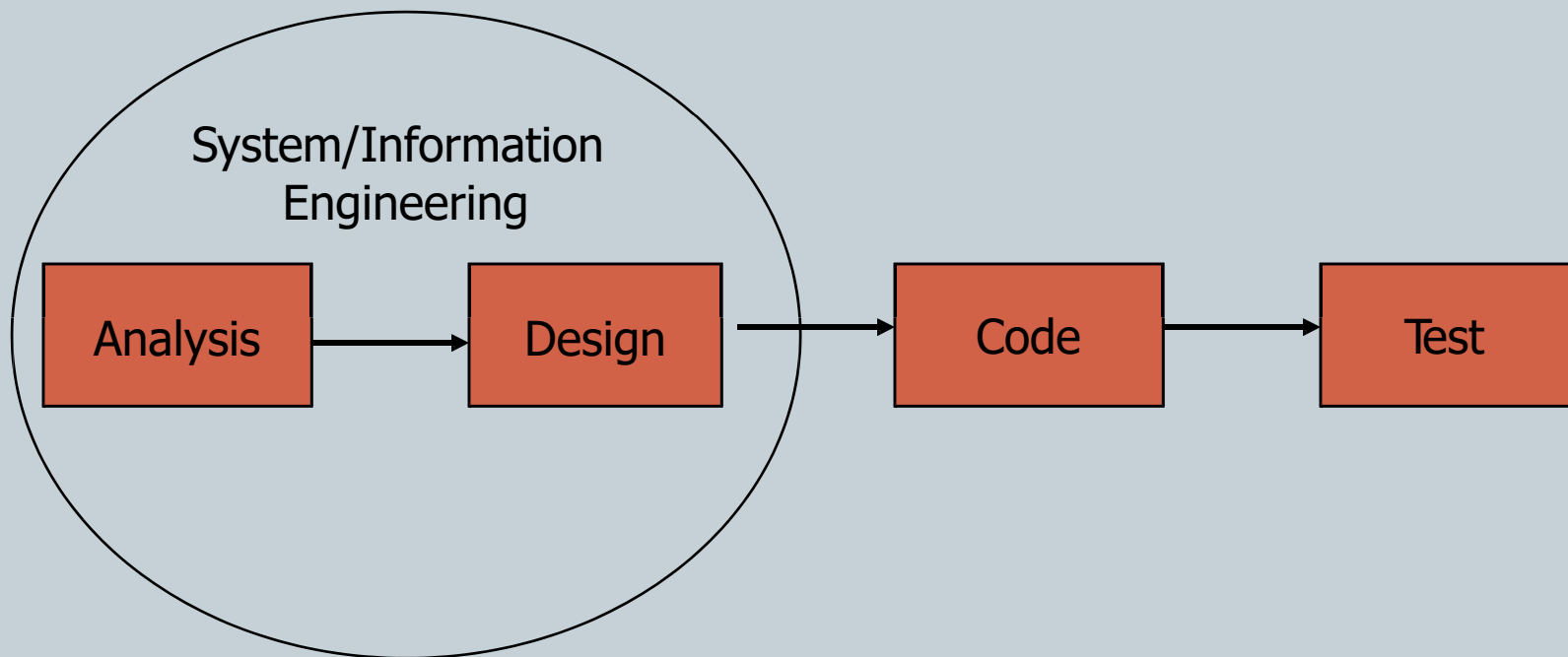
3.8. Software Process Models

9

- **Linear Sequential Model**
 - Waterfall Model
 - V Model
 - RAD Model
- **Prototyping Model**
- **Evolutionary Model**
 - Incremental Model
 - Spiral Model
 - Component Assembly Model
 - Concurrent Development Model
- **Formal Model**
- **Fourth Generation Techniques**
 - Extreme Programming (XP) Model

3.8.1. Linier Sequential Model

10



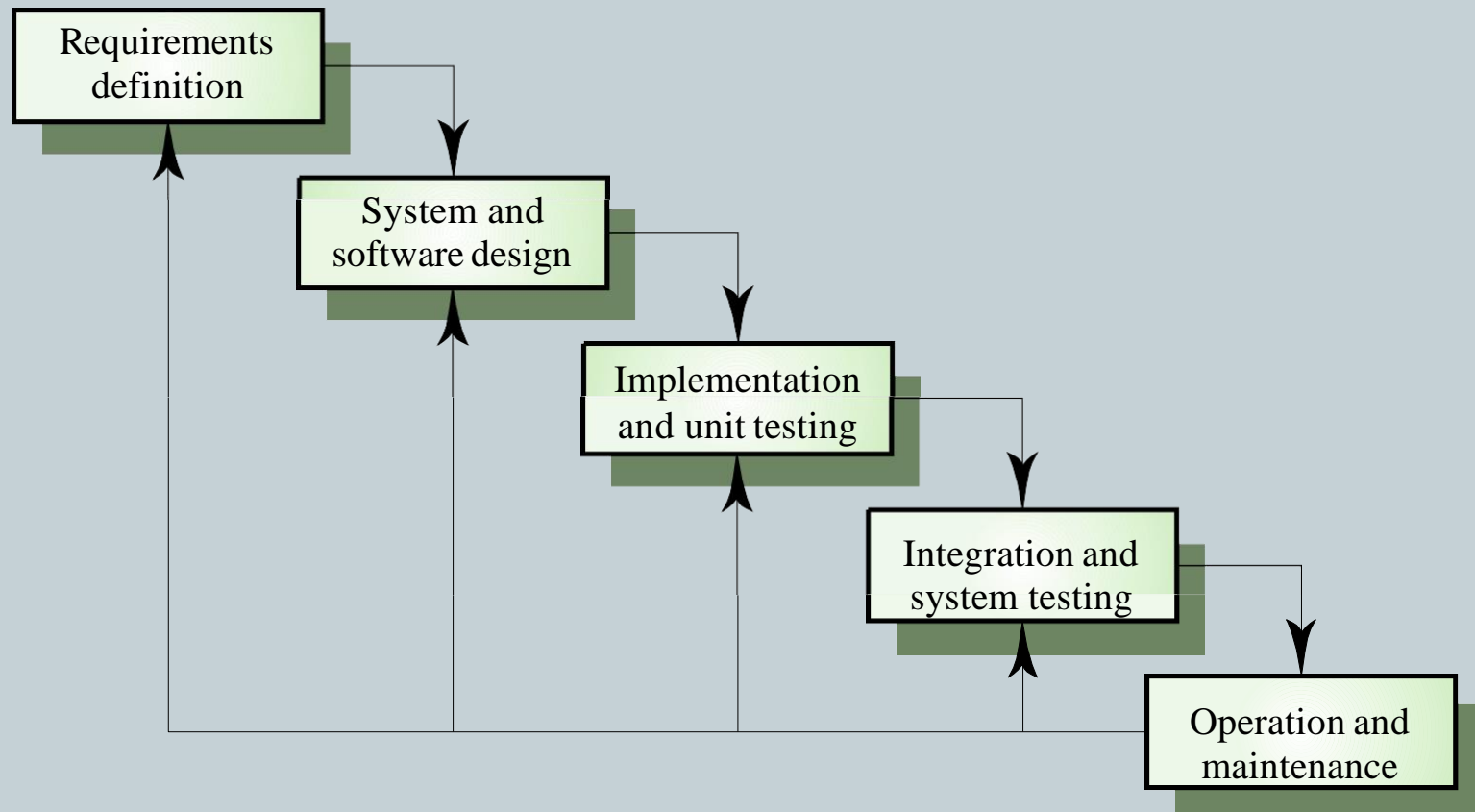
3.8.1.1 Waterfall Model

11

- Sebuah pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial.
- Disebut juga 'Classic Life Cycle'.
Paradigma yang sudah lama sekali, tetapi masih banyak yang memakai karena dianggap masih sesuai dengan keadaan sekarang.

Waterfall Model Diagram

12



Aktifitas Waterfall Model

13

- Requirements analysis and definition : Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun.
- System and software design : Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
- Implementation and unit testing : desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji.
- Integration and system testing : Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).
- Operation and maintenance : mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Kelemahan Waterfall Model

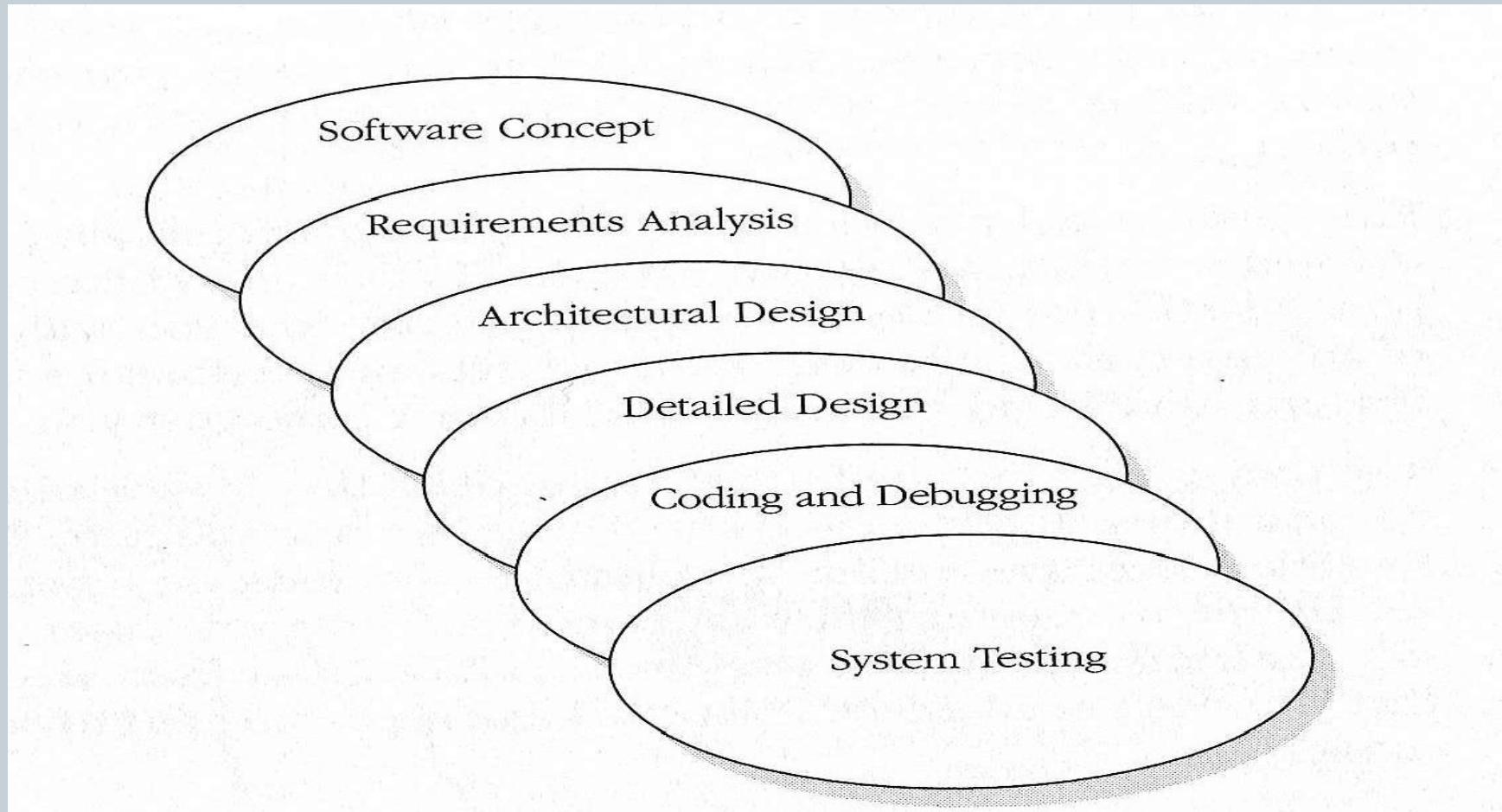
14

- Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani.
- Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya.

Modified Waterfall Model (M.Kochanski)

15

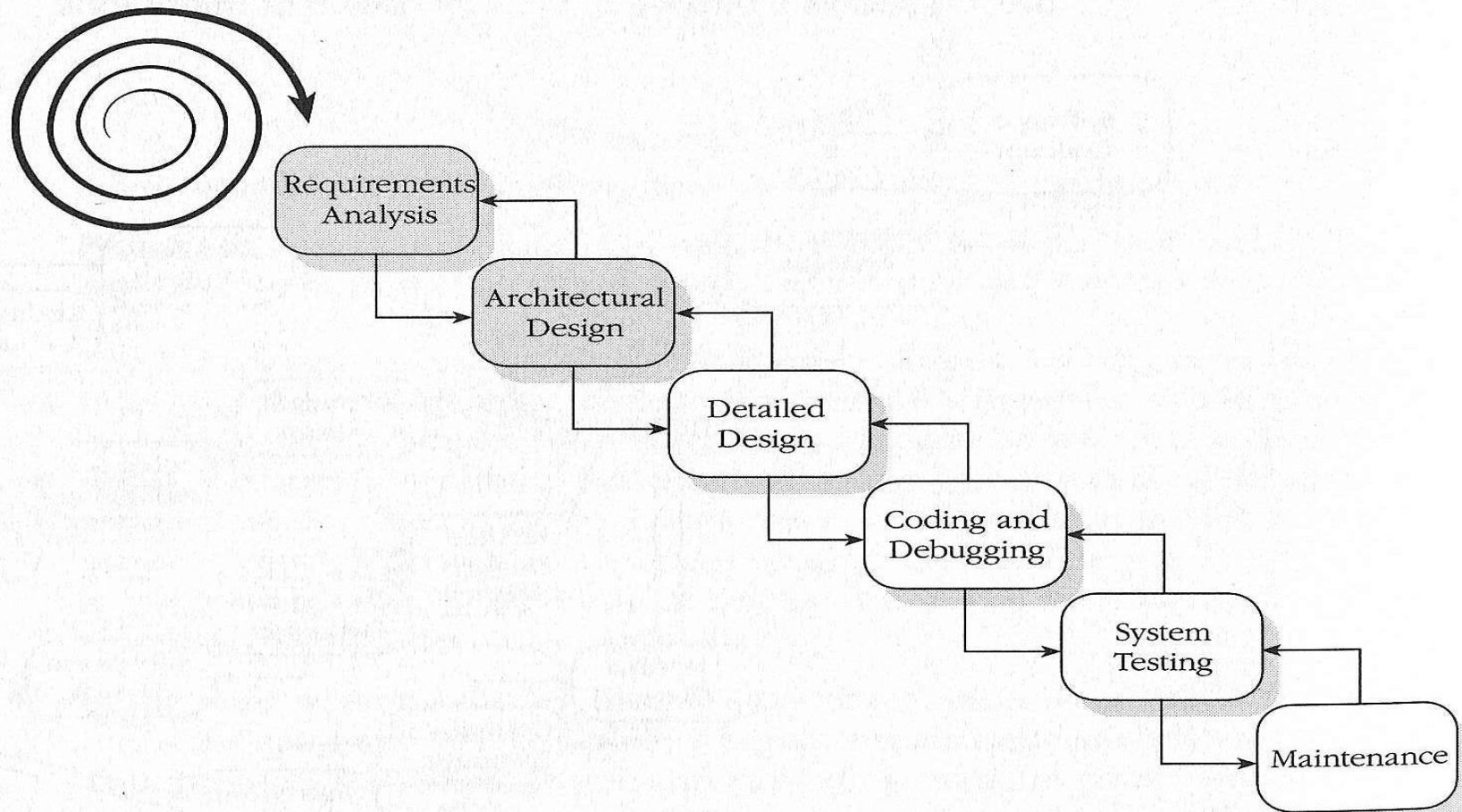
Sashimi



Modified Waterfall Model (M.Kochanski)

16

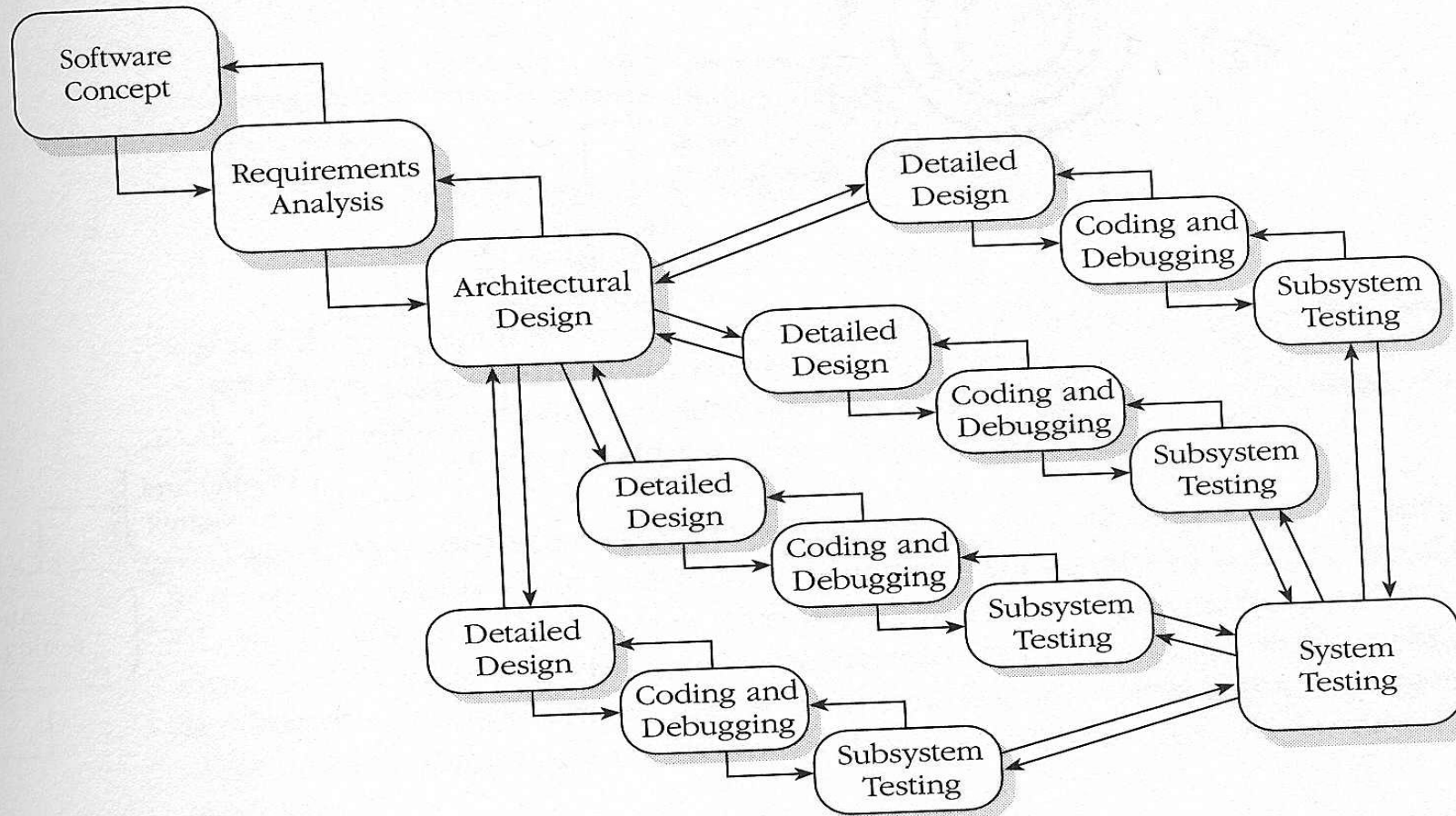
Waterfall with Spiral Introduction



Modified Waterfall Model (M.Kochanski)

17

Waterfall with Subprojects



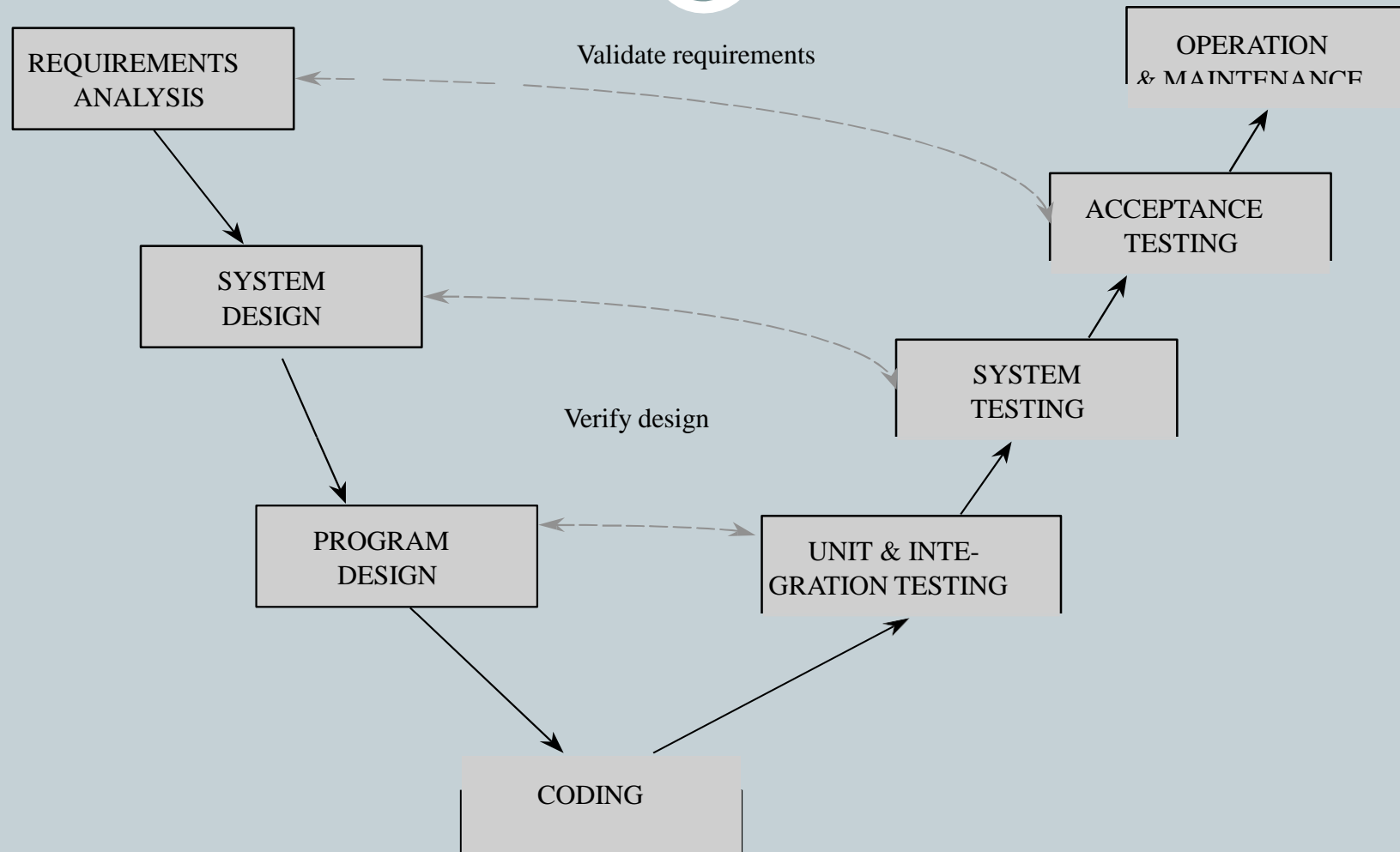
3.8.1.z V Model

18

- V model adalah metode pengembangan perangkat lunak yang mengijinkan pada setiap prosesnya untuk dilakukan testing dan validasi. Jadi proses baru menggunakan hasil dari proses lama sebagai acuannya. Ini memungkinkan meminimalisasikan kesalahan pada prosesnya.

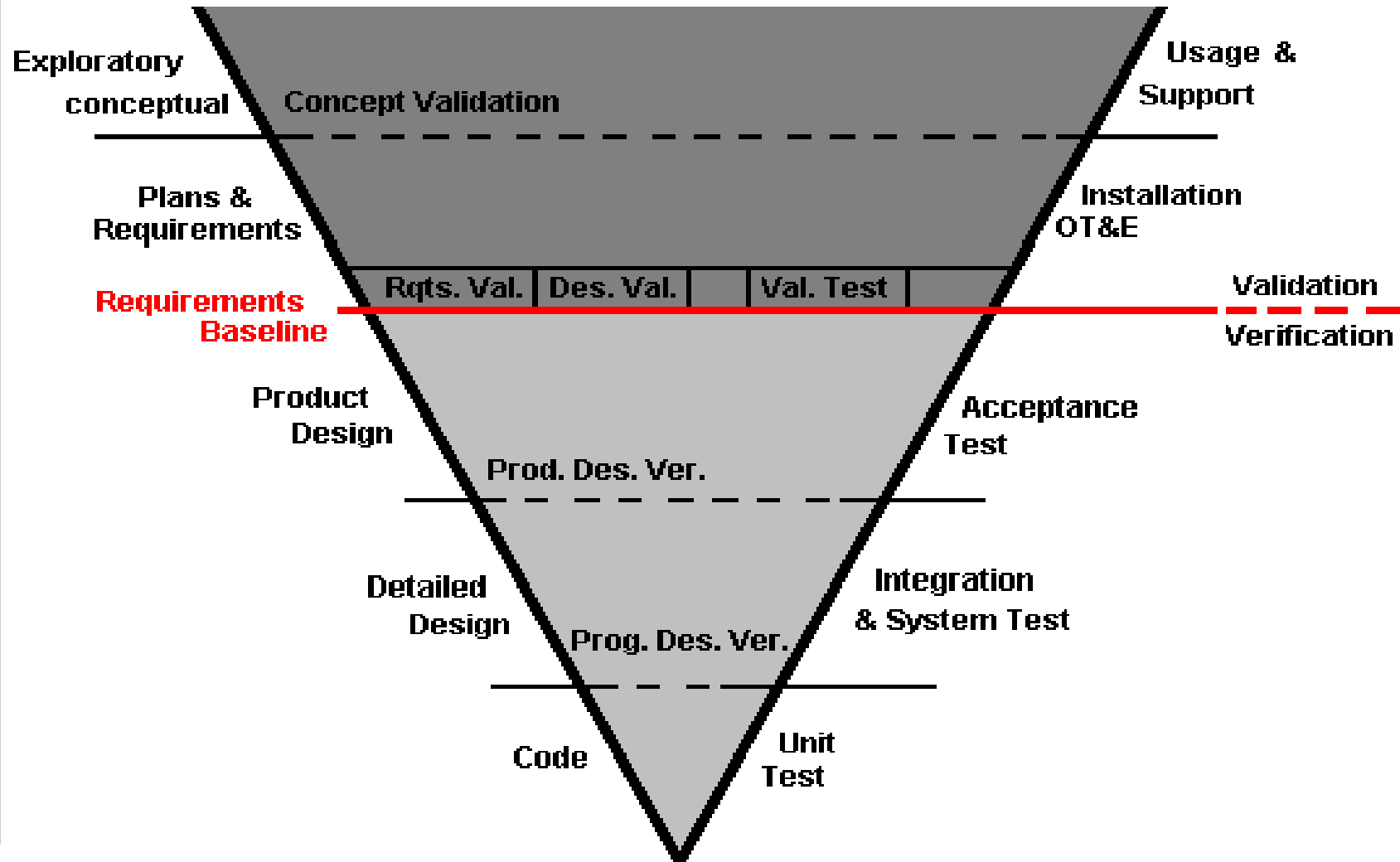
V Model Diagram (Pict-1)

19



V Model Diagram (Pict-z)

20



Keuntungan V Model

21

- Bahasa yang digunakan untuk merepresentasikan konsep V model menggunakan bahasa formal. Contoh: dengan menggunakan objek model ataupun frame-frame
- Meminimalisasikan kesalahan pada hasil akhir karena ada test pada setiap prosesnya
Penyesuaian yang cepat pada proyek yang baru
- Memudahkan dalam pembuatan dokumen proyek
- Biaya yang murah dalam perawatan dan modifikasinya

Kerugian V Model



- Aktivitas V-Model hanya difokuskan pada projectnya saja, bukan pada keseluruhan organisasi. V-Model adalah proses model yang hanya dikerjakan sekali selama project saja, bukan keseluruhan organisasi.
- Prosesnya hanya secara sementara. Ketika project selesai, jalannya proses model dihentikan. Tidak berlangsung untuk keseluruhan organisasi.

Metod yang ditawarkan terbatas Sehingga kit tidak memiliki cara pandang dari metode yang lain. Kita tidak memiliki kesempatan untuk mempertimbangkan jika ada tools lain yang lebih baik.

- Toolnya tidak selengkap yang dibicarakan. SDE (Software Development Environment).Tidak ada tools untuk hardware di V-Model. Tool yang dimaksud adalah “software yang mendukung pengembangan atau pemeliharaan / modifikasi dari system IT.

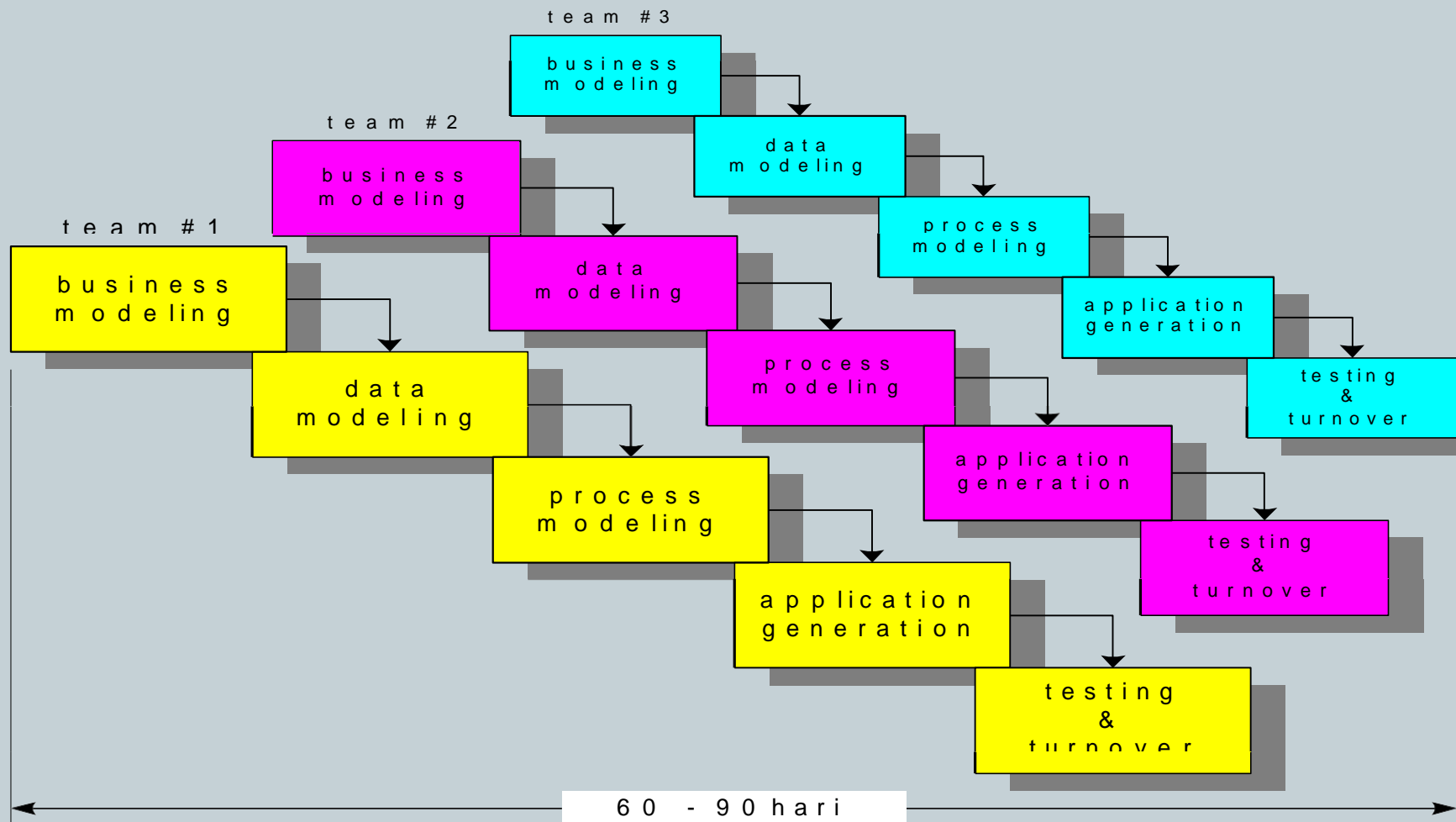
3.8.1.3 RAD Model

23

- RAD = Rapid Application Development
 - Adaptasi dari waterfall model yang:
 - menekankan siklus pengembangan perangkat lunak yang sangat pendek;
 - menggunakan pendekatan konstruksi berbasis komponen.
- Menciptakan sistem fungsional yang utuh dalam waktu 60-90 hari.

RAD Model Diagram

24



RAD Model Phases

25

- **Business Modelling**
Memodelkan fungsi-fungsi bisnis untuk menjawab pertanyaan-pertanyaan: Informasi apa yang mengendalikan proses bisnis ? Informasi apa yang dimunculkan? Ke mana informasi itu pergi? Siapa yang memprosesnya?
- **Data Modelling**
Aliran informasi yang didefinisikan pada fase business modelling ditransformasikan kedalam serangkaian obyek data.
- **Process Modelling**
Mentransformasikan obyek data pada suatu fungsi yang menghasilkan aliran informasi yang dibutuhkan.
- **Application Generation**
Mengkonstruksi perangkat lunak dengan memakai komponen yang ada (bila memungkinkan) atau menciptakan komponen yang dapat dipakai lagi.
- **Testing and Turnover**
Menguji komponen baru.

RAD Model Weakness

26

- Tidak cocok untuk proyek skala besar
- Proyek bisa gagal karena waktu yang disepakati tidak dipenuhi
- Sistem yang tidak bisa dimodularisasi tidak cocok untuk model ini
- Resiko teknis yang tinggi juga kurang cocok untuk model ini

3.8.z Prototyping Model

27

Dipakai jika:

- Sistem mempunyai resiko tinggi
 - tidak jelas permasalahannya
- Lebih fokus pada perancangan dialog user - komputer
 - bagaimana membuat dialog yang baik, ramah, mudah ?
- Sistem diminati oleh banyak pemakai
 - mencari kesepakatan (dasar untuk menyamakan persepsi)
- User ingin cepat selesai
 - user tidak sabar menunggu
 - prototipe segera memperlihatkan bentuk kerja sistem
- Masa pakai singkat
 - sistem hanya dipakai beberapa kali saja
- Ingin menunjukkan inovasi
 - pengembang dapat menunjukkan kecanggihan
- Kebutuhan berubah-ubah
 - user sulit menjelaskan kebutuhan

Types of Prototyping

28

- Evolutionary prototyping
Dimulai dari model, kemudian dikembangkan dan akhirnya dipakai.
- Throw-away prototyping
Hanya dikembangkan sebagai model untuk mencari blue-print.

Evolutionary Prototyping

29

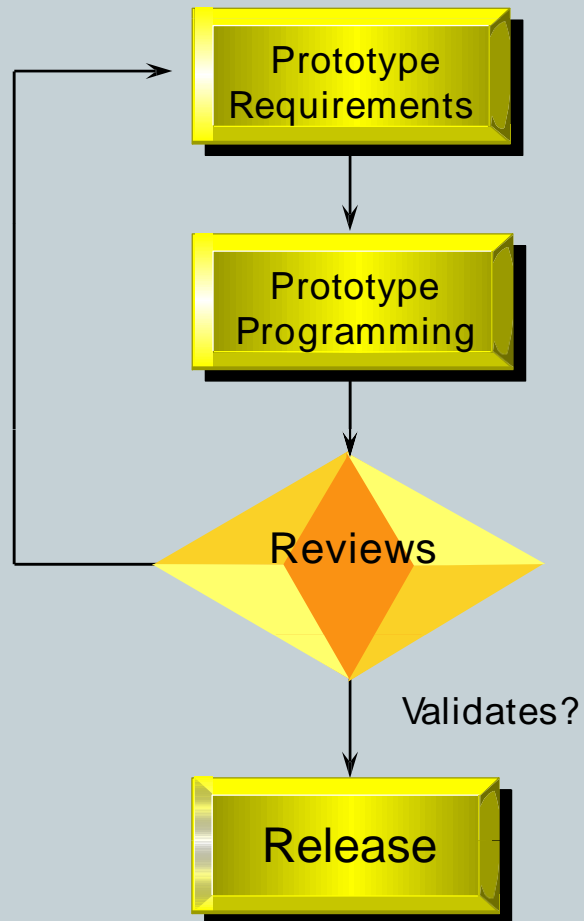
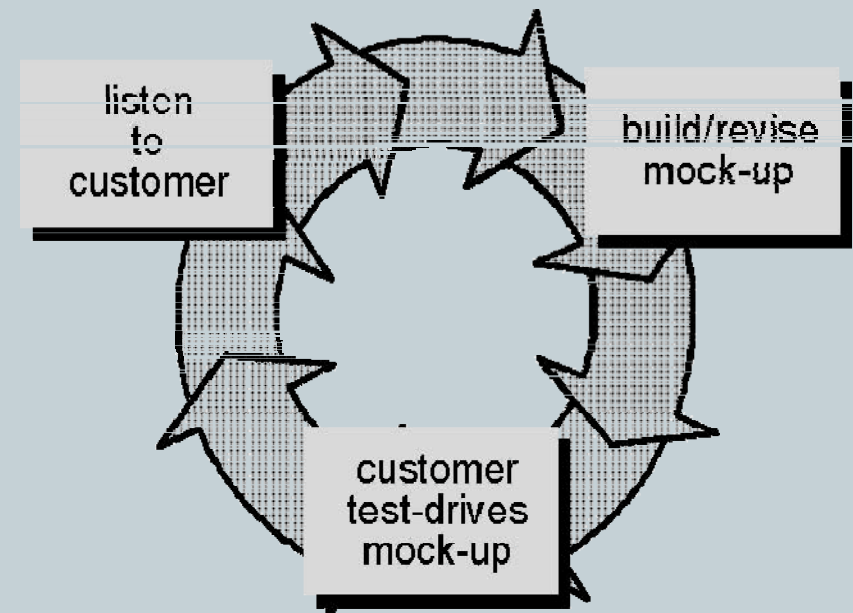


Diagram Lain



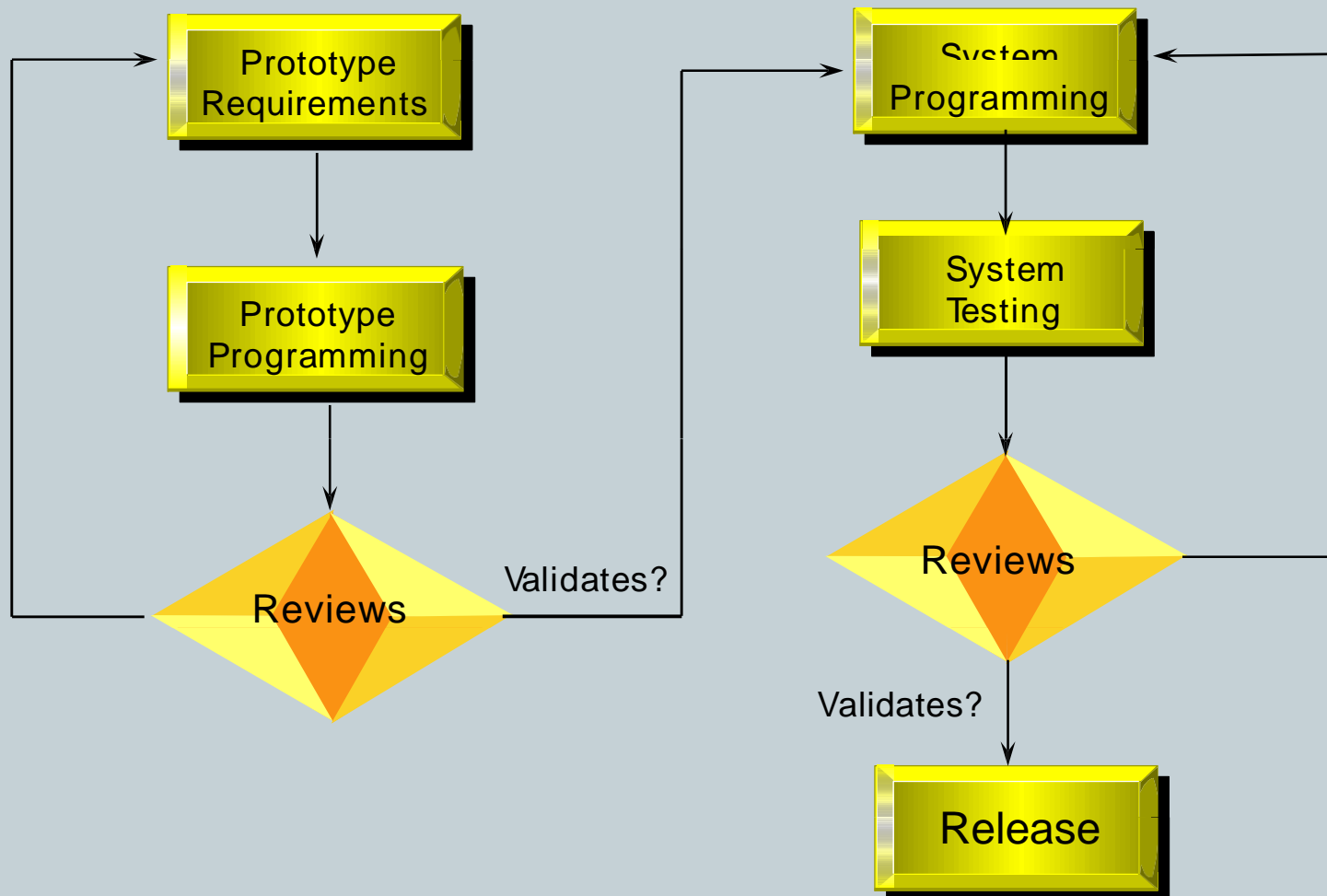
Phases Description

30

- Requirements : developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
- Design : perancangan dilakukan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan prototype.
- Evaluasi prototype : klien mengevaluasi prototype yang dibuat dan digunakan untuk memperjelas kebutuhan software.

Throw-away prototyping

31



Prototyping Speciality

32

- Frekuensi komunikasi user – developer meningkat
 - pengembang akan selalu meminta pendapat user
- Membantu analis dalam
 - menentukan kebutuhan user yang sebenarnya meminimalkan salah persepsi
- Peran user meningkat
 - evaluasi oleh user berkali-kali user bisa memberikan masukan setiap saat
- Pengembangan lebih cepat
 - program bisa langsung dibuat
 - user melihat perkembangan tahap demi tahap
- Implementasi mudah
 - user sudah mengenal perangkat lunak yang dikembangkan
 - user tidak akan merasa asing
 - sejak awal user sudah merasa memiliki

Prototyping Weakness

33

- User sibuk
 - user & pengembang harus sama-sama memiliki komitmen menyediakan waktu untuk bertemu.
- User sulit melakukan evaluasi
 - bentuk prototipe sering berubah, disesuaikan dengan kebutuhan user.
- User ingin cepat selesai
 - bentuk program sudah terlihat sejak awal.
 - user merasa tidak akan lama lagi selesai.
 - pengembang sering mengabaikan dokumentasi.
- User berharap terlalu banyak
 - keseringan evaluasi & komunikasi membuat user menjadi berubah keinginan dan tidak pasti dengan kebutuhan.
- Prototipe bekerja tidak efisien
 - lebih mementingkan keberhasilan.

3.8.3 Evolutionary Model

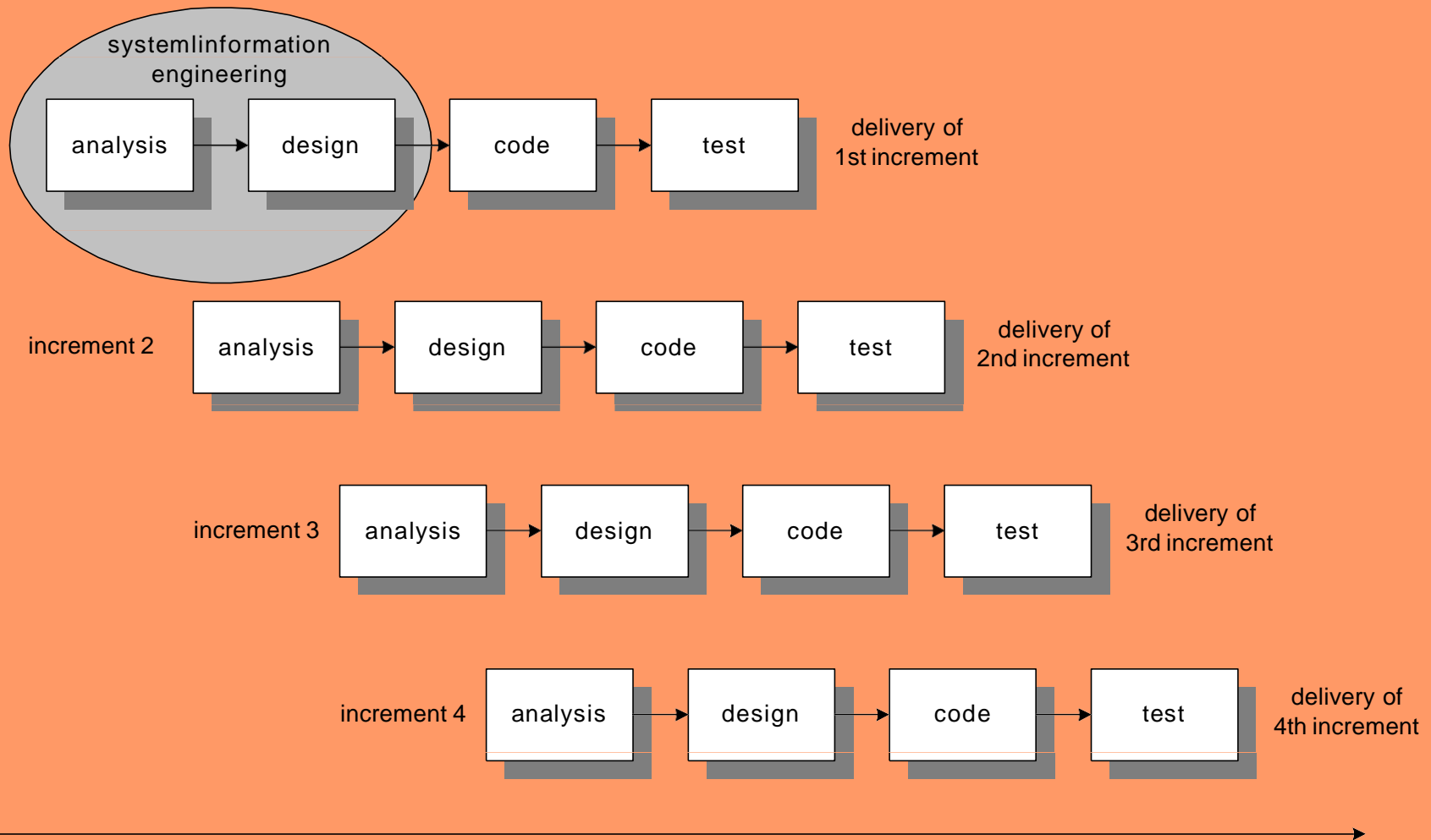
34

3.8.3.1 Incremental Model

- Incremental Model merupakan gabungan antara model linier sekuensial dan prototyping.
- Setiap linier sekuen menghasilkan produk yang deliveriabies.
- Increment pertama merupakan produk inti (core), yang mengandung persyaratan/kebutuhan dasar.
- Penambahan dilakukan pada increment-increment berikutnya.

Incremental Model Diagram

35



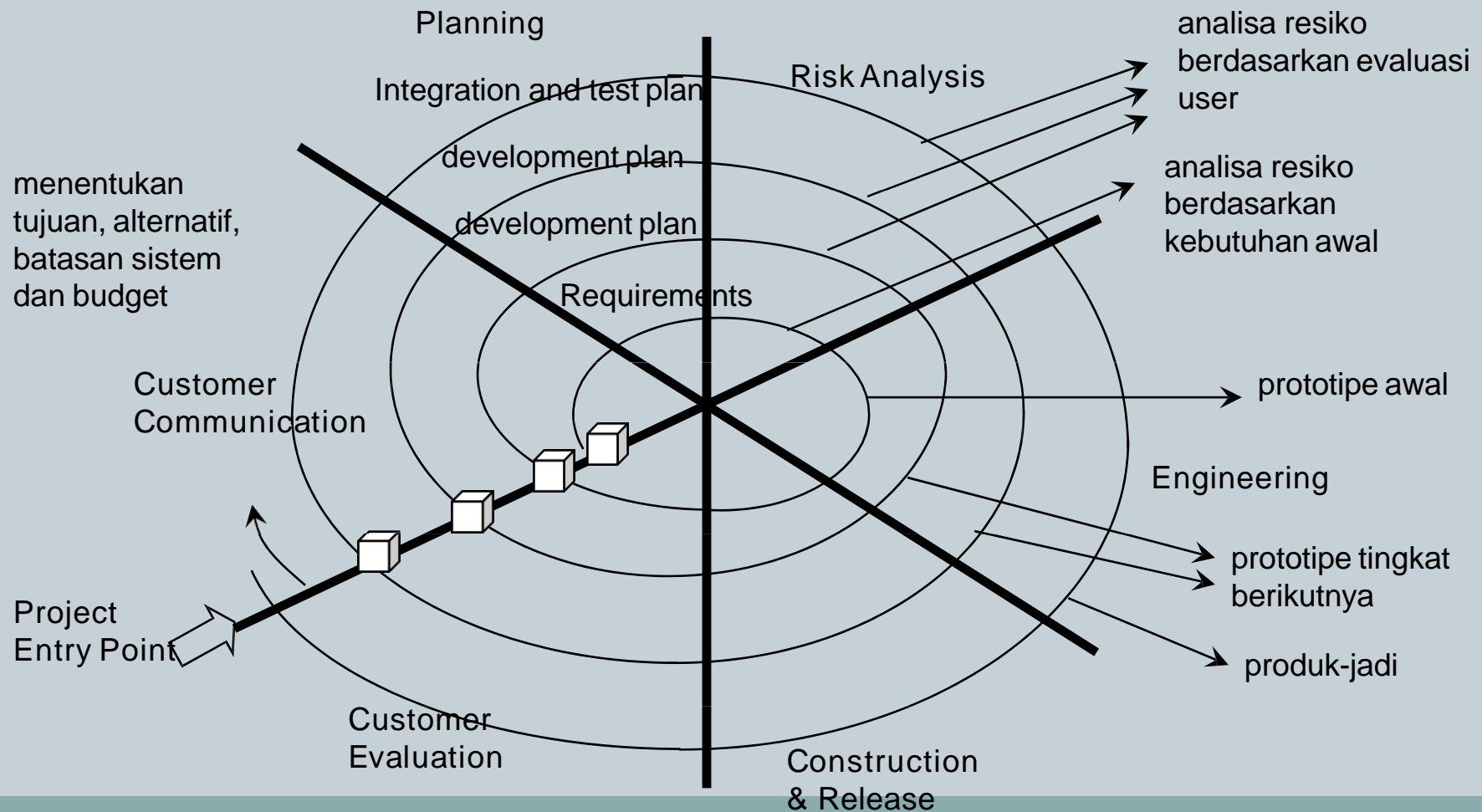
3.8.3.z Spiral Model

36

- Evolutionary process (pengembangan bertingkat)
- Menggabungkan keunggulan prototyping dan waterfall
- Memungkinkan dikembangkan perangkat lunak secara bertahap (incremental) dan cepat.
- Terbagi atas 6 tahapan
 - customer communication
 - planning
 - risk analysis
 - engineering
 - construction & release
 - customer evaluation

Spiral Model Diagram

37



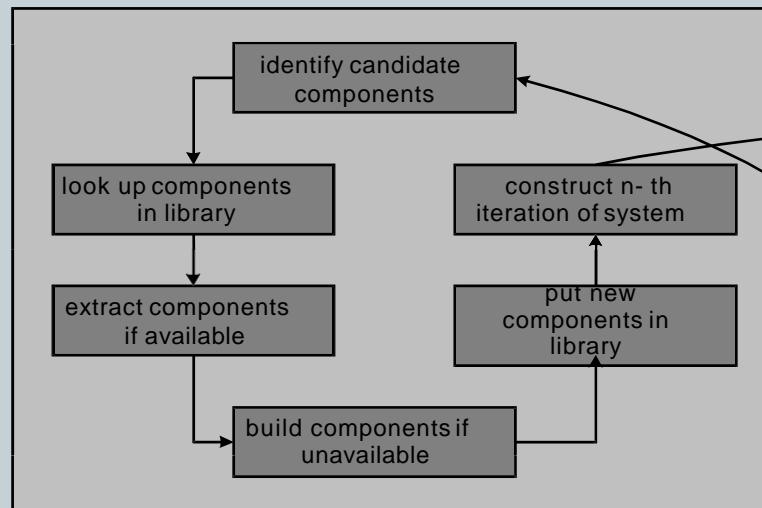
Spiral Model Phases

38

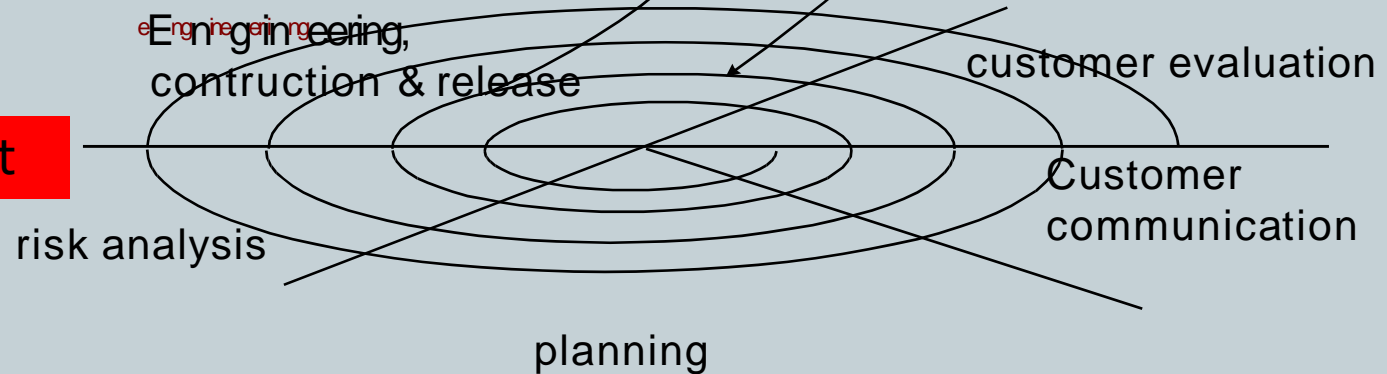
- **Customer communication:** membangun komunikasi yang baik dengan pengguna/customer.
- **Planning:** mendefinisikan resources, batas waktu, informasi-informasi lain seputar proyek
- **Risk Analysis:** identifikasi resiko manajemen dan teknis
- **Engineering:** pembangunan contoh-contoh aplikasi, misalnya prototype
- **Construction and Release:** pembangunan, test, install dan support.
- **Customer Evaluation:** mendapatkan feedback dari pengguna berdasarkan evaluasi PL pada fase engineering dan fase instalasi.

3.8.3.3 Component Assembly Model

39

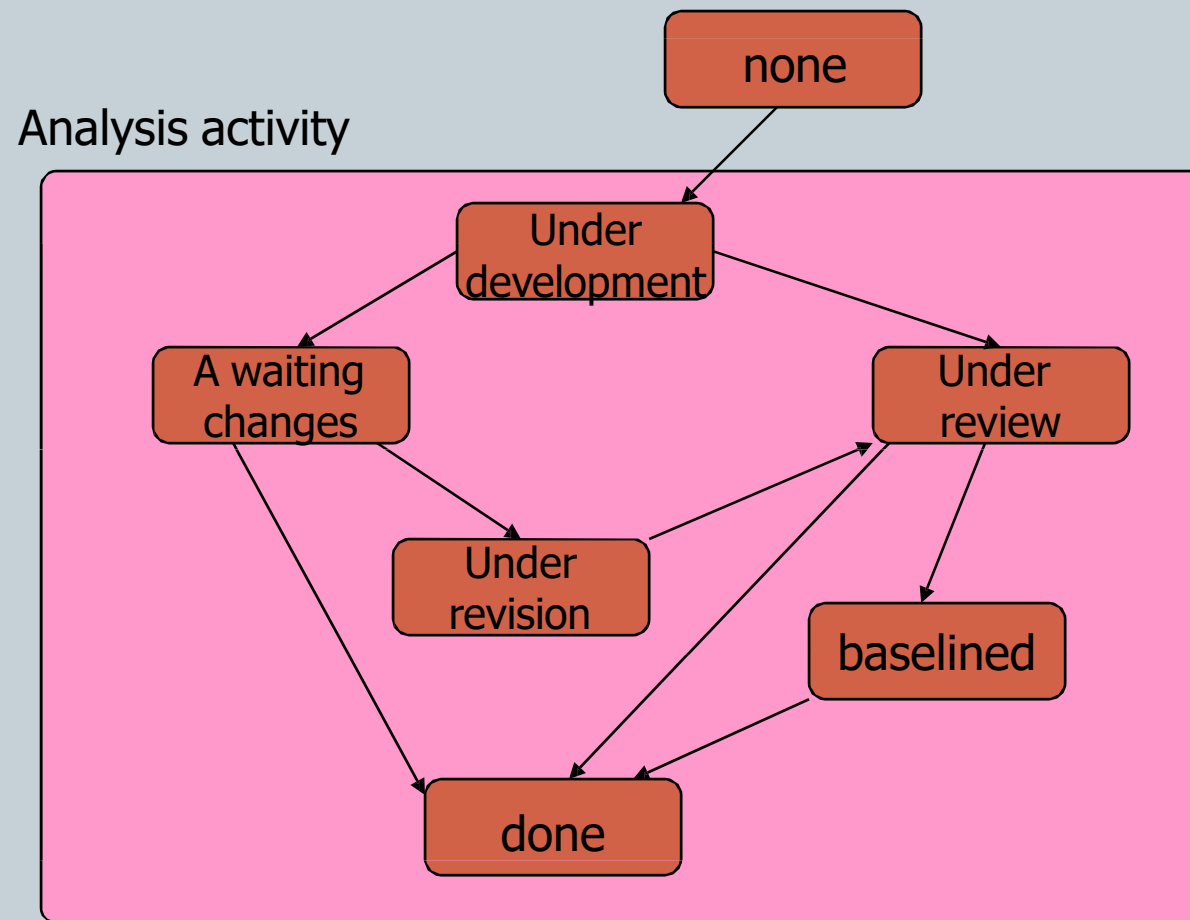


entry point



3.8.3.0 Concurrent Development Model

40



Konkurensi Tercapai dengan Cara:

41

- Aktivitas sistem dan komponen yang berlangsung secara simultan dan dapat dimodelkan dengan menggunakan pendekatan orientasi keadaan;
- Aplikasi klien/server khusus yang diimplementasikan dengan banyak komponen yang masing-masing bisa dirancang dan direalisasikan secara konkuren.

3.8.4 Formal Model

42

mencakup sekumpulan aktivitas yang membawa kepada spesifikasi matematis perangkat lunak komputer;

- memungkinkan software engineer untuk mengkhususkan, mengembangkan, dan mem-verifikasi sistem berbasis komputer dengan menggunakan notasi matematis yang tepat;
- Variasi dari pendekatan ini disebut clean-room software engineering.
- (silahkan dibaca sendiri – untuk pengayaan)

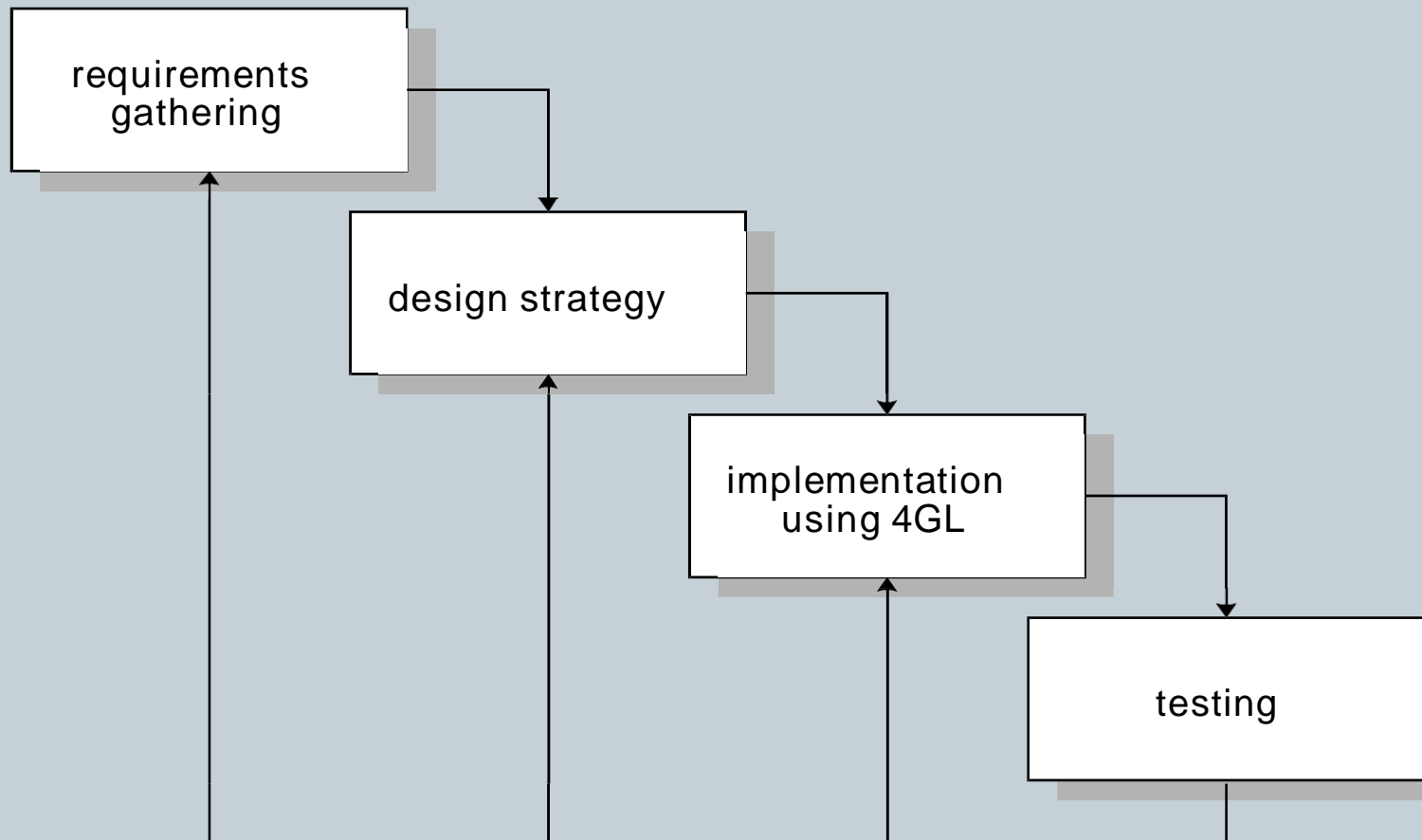
3.8.5 Fourth Generation Techniques (4GT)

43

- Terkait dengan penggunaan tools.
- Pengembang software mendefinisikan karakteristik software secara 'high level'; tool secara otomatis akan membangkitkan kode.
- 4GT mempercepat proses pengembangan perangkat lunak.
- Proses perancangan dan dokumentasi baik.
- Masih dipertanyakan beberapa pihak: efisiensi kode yang

4GT Techniques Diagram

44



3.8.5.j Extreme Programming (XP) Model

45

- Model proses ini diciptakan dan dikembangkan oleh Kent Beck. Model ini adalah model proses yang terbaru dalam dunia rekayasa perangkat lunak dan mencoba menjawab kesulitan dalam pengembangan software yang rumit dan sulit dalam implementasi. Menurut Kent Beck XP adalah : “A lightweight, efficient, low-risk, flexible, predictable, scientific and fun way to develop software”.

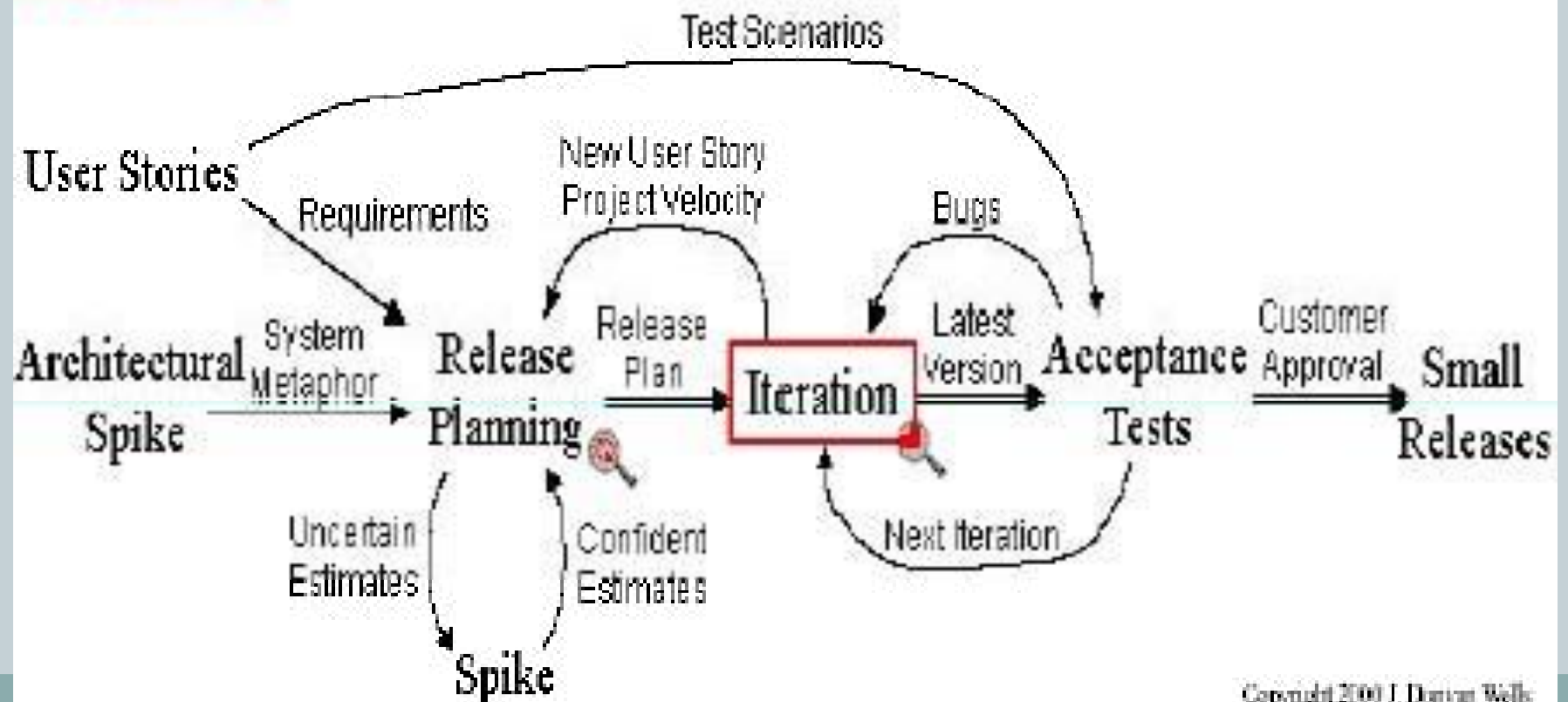
Suatu model yang menekankan pada keterlibatan user secara langsung dalam requirement, design, coding dan testing.

XP Model

46



Extreme Programming Project



Copyright 2000 J. Dorian Wells

Karakteristik XP Model

47

- Communication (Komunikasi): komunikasi antara developer dan klien sering menjadi masalah. Karena itu komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (pair programming). Developer didampingi oleh pihak klien dalam melakukan coding dan unit testing sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan developer. Selain itu perkiraan beban tugas juga diperhitungkan.
- Simplicity (Sederhana): Menekankan pada kesederhanaan dalam pengkodean : “What is the simplest thing that could possibly work?” Komunikasi yang lebih banyak mempermudah, dan rancangan yang sederhana mengurangi penjelasan.
- Feedback (Tanggapan): Setiap feedback ditanggapi dengan melakukan tes, unit test atau system integration.
- Courage (Berani): Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.