

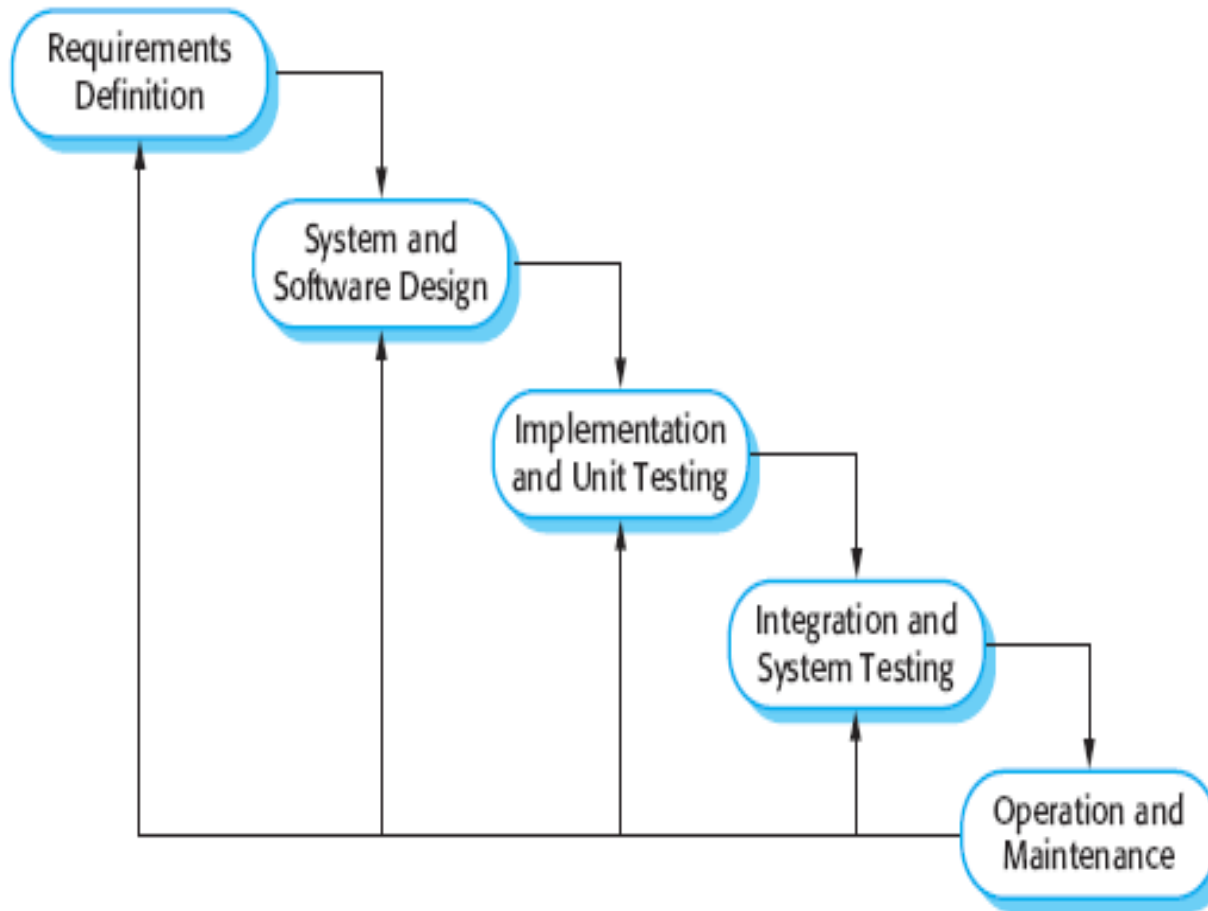
## Pertemuan 2

# MODEL PROSES PENGEMBANGAN PERANGKAT LUNAK

# 1. MODEL WATERFALL

- Model *Waterfall* juga disebut siklus hidup klasik (*Classic Life Cycle*).
- Merupakan pendekatan sistematis dan berurutan (*sequential*) pada PL yang dimulai dengan spesifikasi kebutuhan *user* dan berlanjut melalui beberapa tahapan, dan diakhiri dengan penyerahan sistem/ PL kepada pelanggan.
- Pada prinsipnya, hasil dari setiap tahap adalah satu atau lebih dokumen yang disetujui (*Sign off*).
- Tahap berikutnya tidak boleh dimulai sampai tahap sebelumnya selesai.

# Model *Waterfall*



Sumber: **Sommerville (2011: 30)**

# Tahapan Model Waterfall

## ***1. Requirements Analysis and Definition***

Langkah ini merupakan analisa kebutuhan sistem. Berisi layanan-layanan sistem, kendala, dan tujuan yang ditetapkan melalui konsultasi dengan pengguna sistem, kemudian didefinisikan secara rinci yang berfungsi sebagai spesifikasi sistem.

## ***Requirements Analysis and Definition (Lanjutan)***

### **Misalkan untuk aplikasi web:**

Maka analisa kebutuhannya ditinjau dari kebutuhan Admin dan User. Jelaskan secara singkat kebutuhan apa yang bisa dilakukan oleh admin dan user.

Contoh untuk web: PT. Telkom Property ini membutuhkan suatu sistem terkomputerisasi untuk mempermudah proses pengelolaan barang pada departement housekeeping agar proses keluar masuk barang beserta laporanlaporannya lebih tertib, teratur dan efisien

### **Misalkan untuk aplikasi desktop:**

Pada program inventory, kebutuhannya adalah pencatatan barang.

# Tahapan Model Waterfall

## 2. *System and Software Design*

Proses *design* mengalokasikan kebutuhan *hardware* dan *software* untuk membangun arsitektur sistem secara keseluruhan (struktur data, arsitektur PL, *interface*, dan detail/algoritma prosedural). Proses *design* akan menerjemahkan syarat kebutuhan perancangan PL yang dapat diperkirakan sebelum dibuat *coding*.

Contoh:

rancangan yang dibuat seperti usecase sistem housekeeping inventory, activity login, dll. Tools program yang digunakan secara umum yaitu ERD, LRS dan UML.

# Tahapan Model Waterfall

## 3. *Implementation and Unit Testing*

Perancangan PL direalisasikan sebagai satu set program atau unit program (*coding*). *Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer yang dilakukan oleh *programmer*.

Kemudian dilanjutkan dengan testing terhadap pengujian unit dengan melibatkan verifikasi setiap unit agar memenuhi spesifikasinya.

Contoh:

perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program yang dibuat dengan menggunakan bahasa pemrograman Java dan menggunakan database MySQL serta Netbeans IDE sebagai tools editornya.

# Tahapan Model Waterfall

## 4. *Integration and System Testing*

Program-program diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa kebutuhan/persyaratan PL telah dipenuhi. Setelah pengujian, PL sistem dikirim ke pelanggan.

Contoh:

Setelah tahap pembuatan code program dilakukan pengujian program untuk menemukan kesalahan-kesalahan program. Pengujian program menggunakan *black box testing* untuk memastikan bahwa seluruh input yang memerlukan verifikasi data sudah benar.



# Tahapan Model Waterfall

## 5. *Operation and Maintenance*

Operasi dan pemeliharaan adalah siklus hidup terlama. Sistem ini dipasang dan digunakan oleh *user*. Perawatan melibatkan koreksi kesalahan yang tidak ditemukan sebelumnya, meningkatkan implementasi sistem dan meningkatkan layanan sistem saat persyaratan baru ditemukan.

Contoh:

Program dipasang dan digunakan secara nyata. Maintenance melibatkan pembetulan kesalahan yang tidak di temukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai suatu kebutuhan sistem yang baru.

## Kelebihan model *waterfall*:

- a. Memiliki proses yang urut dan bertahap, sehingga kualitas sistem/PL yang dihasilkan akan baik.
- b. Setiap proses memiliki spesifikasinya sendiri, karena setiap tahap harus terselesaikan dengan lengkap sebelum melanjutkan ke tahap berikutnya.
- c. Setiap proses tidak dapat saling tumpang tindih.
- d. Metode ini akan lebih baik digunakan jika kebutuhan-kebutuhan sudah diketahui.

## Kekurangan model *waterfall*:

- a. Proses yang dilakukan cenderung panjang dan lama, karena proses pengembangan tidak dapat dilakukan berulang sebelum menghasilkan produk.
- b. Kesalahan kecil pada satu tahapan akan menimbulkan masalah besar jika tidak diketahui sejak awal pengembangan, berakibat pada tahapan selanjutnya
- c. Biaya penggunaan metode yang cenderung mahal
- d. Membutuhkan banyak riset dan penelitian pendukung untuk mengembangkan sistem sehingga pelanggan harus sabar karena pembuatan PL baru dimulai pada tahap perancangan.
- e. Kenyataannya sulit untuk mengikuti aturan *sequential*, karena iterasi sulit dilakukan dan menyebabkan masalah baru.

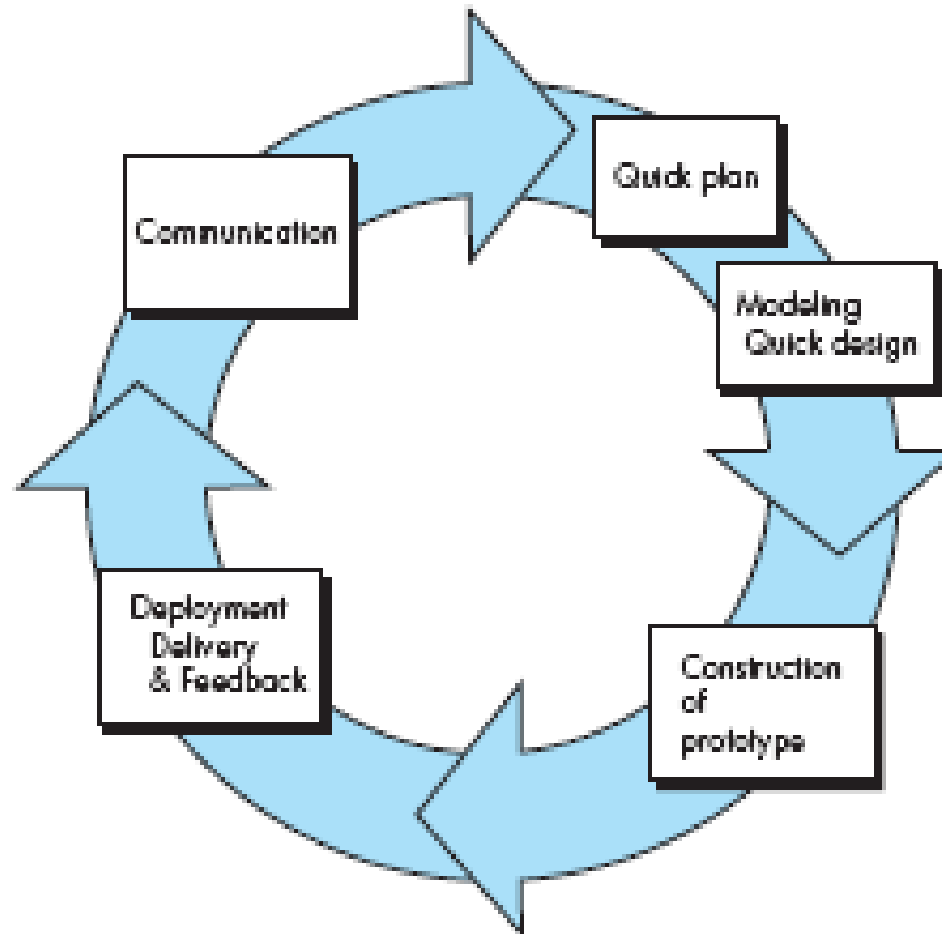
## 2. MODEL *PROTOTYPE*

- *Prototype* adalah pendefinisian sejumlah sasaran PL berdasarkan kebutuhan dan pemahaman secara umum, tetapi tidak bisa mengidentifikasi kebutuhan secara rinci untuk beberapa fungsi dan fitur-fitur.
- Tujuannya adalah untuk membantu dalam tahap analisis dan desain yang memungkinkan pengguna untuk melihat lebih awal apa yang akan dilakukan sistem, yaitu untuk memfasilitasi validasi.
- *Prototype* dapat digunakan sebagai model proses yang berdiri sendiri.

## Model *Prototype* (Lanjutan)

- Pembuatan prototipe biasanya digunakan sebagai teknik yang dapat diimplementasikan di dalam konteks setiap model proses PL, sehingga membantu *stakeholder* untuk lebih memahami apa yang akan dikembangkan ketika spesifikasi kebutuhan belum jelas.

# Model *Prototype*



Sumber: Pressman (2015: 45)

## Tahapan dalam Model *Prototype*

1. Dimulai dengan dilakukannya komunikasi antara tim pengembang PL dengan pelanggan.
2. Tim pengembang bertemu dengan *stakeholder* untuk mendefinisikan sasaran keseluruhan PL, mengidentifikasi spesifikasi kebutuhan yang diketahui, dan menggambarkan definisi lebih jauh pada iterasi selanjutnya.
3. Pembuatan prototipe direncanakan dengan cepat, dan pemodelan dilakukan
4. Prototipe diserahkan kepada *stakeholder* untuk dievaluasi, dan memberikan umpan balik yang digunakan untuk persyaratan lebih lanjut
5. Iterasi akan terjadi saat prototipe diperbaiki

## Tujuan Pengembangan Model *Prototype*:

- a. Membuat antarmuka pengguna yang dapat diterima
- b. Membuat sistem yang dapat berfungsi, meskipun terbatas, tetapi tersedia dengan cepat untuk menunjukkan kelayakan dan kegunaan dari aplikasi
- c. Dapat digunakan untuk melatih pengguna sebelum sistem yang lengkap dikirim ke pelanggan
- d. Untuk menjelaskan bahwa beberapa teknologi baru akan menyediakan fasilitas yang dibutuhkan



# Manfaat Model *Prototype*

Model *prototype* dikembangkan dan didemonstrasikan pada awal proses pembangunan PL, sehingga dapat bermanfaat untuk:

- a. Menghindari kesalahpahaman antara pengembang PL dan pengguna
- b. Beberapa fasilitas yang hilang mungkin dapat terungkap
- c. Fasilitas yang sulit digunakan/membingungkan dapat diidentifikasi dan disempurnakan
- d. Pengembang PL mungkin menemukan persyaratan yang tidak lengkap atau tidak konsisten.

## Masalah pada Model *Prototype*

1. *Stakeholder* hanya melihat tampilan PL yang akan dipakai tanpa mempedulikan bagaimana kerja sistem, dan pemeliharaan jangka panjang.
2. Perubahan yang dibuat selama pengembangan PL mungkin akan mengubah struktur arsitektur. Oleh karena itu mungkin sulit dan mahal untuk pemeliharaannya.
3. Karakteristik sistem yang penting seperti kinerja, keamanan dan keandalan, mungkin akan diabaikan selama pengembangan PL.
4. Selama tahap pengembangan, *prototype* akan diubah untuk memenuhi kebutuhan pengguna. kemungkinan perubahan yang dibuat akan tidak terkontrol dan tidak didokumentasikan dengan baik.

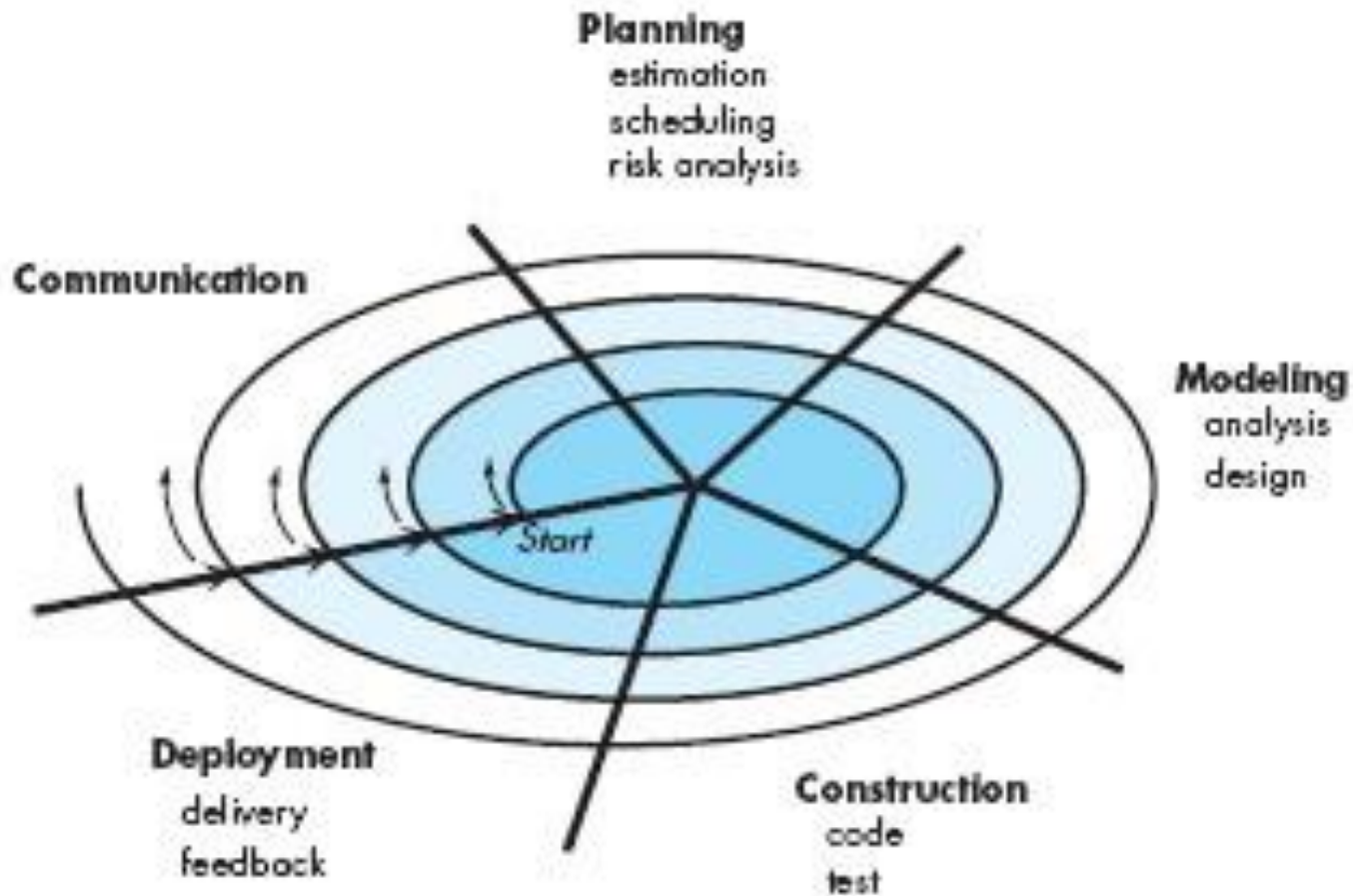
### 3. MODEL *SPIRAL*

- Model Spiral merupakan suatu model proses PL evolusioner yang menggabungkan pendekatan *prototyping* yang bersifat iteratif dengan aspek yang terkontrol dan sistematis pada model *waterfall*.
- Model pengembangan spiral adalah model proses PL yang dikendalikan risiko yang digunakan untuk memandu para *stakeholder* untuk secara bersamaan merekayasa sistem yang bernuansa PL.

## Model *Spiral* (Lanjutan)

- Model spiral menggunakan prototipe sebagai mekanisme pengurangan risiko.
- Model spiral menggunakan pendekatan langkah demi langkah (*waterfall*) yang sistematis tetapi menggabungkannya ke dalam kerangka iteratif yang lebih realistis mencerminkan dunia nyata.
- Model spiral adalah pendekatan realistis untuk pengembangan sistem berskala besar.

# Model *Spiral*



Sumber: Pressman (2015: 48)

## Penjelasan gambar:

- a. Proses evolusioner ini dimulai dari titik tengah, ke bagian luar spiral searah jarum jam.
- b. Risiko PL akan dipertimbangkan saat masing-masing gerakan dibuat dan titik pengukuran dicatat setiap saat langkah-langkah evolusioner dilewati.
- c. Lintasan pertama di sekitar spiral dapat menghasilkan spesifikasi produk, putaran berikutnya di sekitar spiral mungkin digunakan untuk mengembangkan suatu prototipe dan pada lintasan berikutnya secara progresif bergerak ke versi PL yang semakin canggih.

## Penjelasan gambar (lanjutan)

- d. Setiap melewati lintasan menghasilkan penyesuaian pada perencanaan proyek.
- e. Biaya dan jadwal disesuaikan berdasarkan umpan balik yang berasal dari pelanggan setelah pengiriman produk.
- f. Selain itu, dilakukan penyesuaian jumlah iterasi yang direncanakan untuk menyelesaikan produk PL.

## Keuntungan Model *Spiral*

- a. Pendekatan yang dikendalikan risiko menghindari banyak kesulitan.
- b. Mengakomodasi persiapan untuk evolusioner siklus hidup, pertumbuhan, dan perubahan pada produk PL.
- c. Menyediakan mekanisme untuk tujuan kualitas dan PL gabungan ke pengembangan produk PL
- d. Mempunyai fokus untuk mengeliminasi kesalahan (*error*)
- e. Menyediakan pendekatan terpisah untuk pengembangan dan pemasangan PL
- f. Menyediakan kerangka kerja aktif untuk pengembangan sistem *hardware* dan *software* yang terintegrasi.



## Kerugian Model *Spiral*

- a. Memerlukan penyesuaian dengan PL yang menitik beratkan pada kontrol dan titik permasalahan yang merupakan keunggulan model *waterfall*.
- b. Berdasarkan keahlian manajemen risiko yang memerlukan penaksiran risiko yang masuk akal, akan menimbulkan masalah yang lebih besar jika risiko mayor tidak ditemukan
- c. Memerlukan kebutuhan untuk penelitian lebih lanjut terhadap langkah-langkah spiral khususnya untuk area analisis risiko.

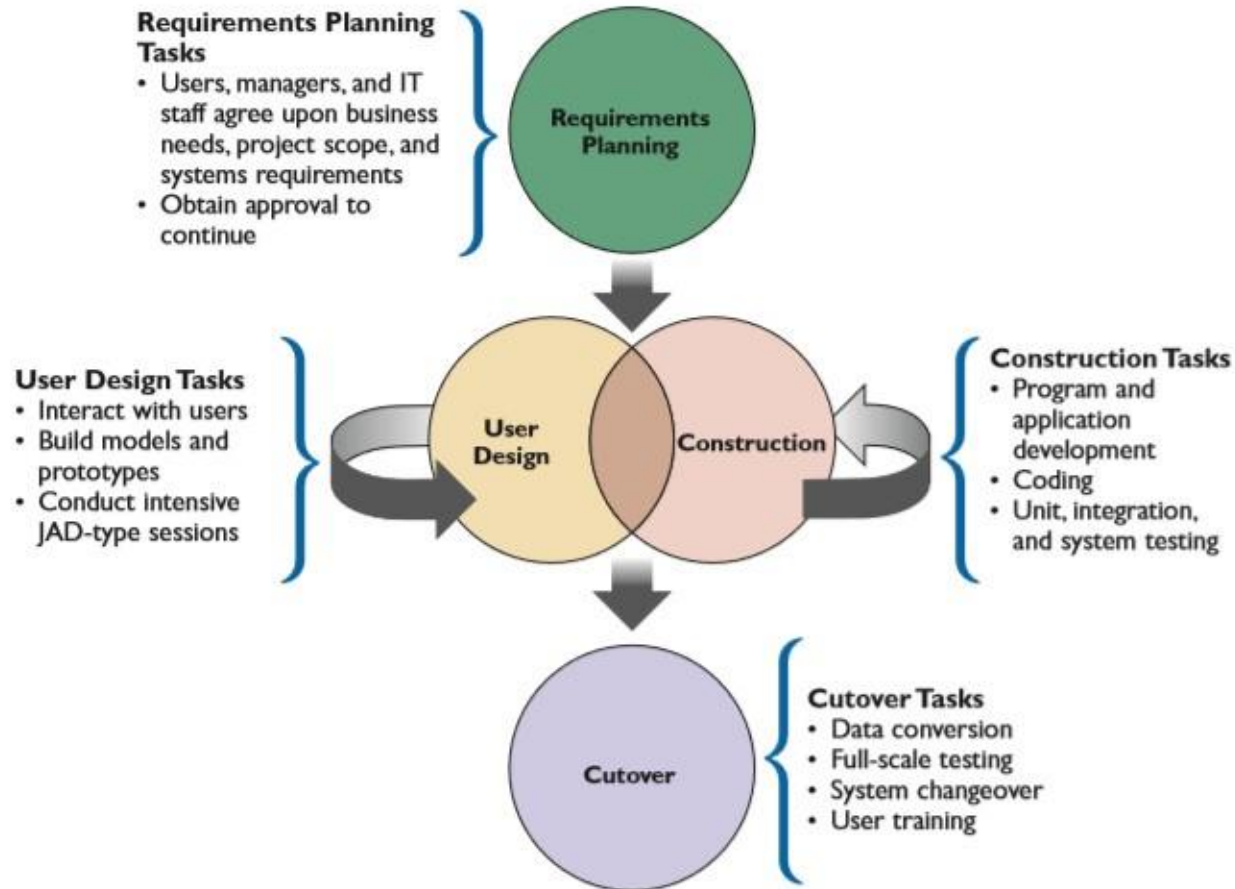
## 4. **MODEL *Rapid Application Development* (RAD)**

- RAD adalah teknik berbasis tim yang mempercepat pengembangan SI dan menghasilkan fungsi-fungsi SI.
- RAD menggunakan pendekatan kelompok
- Produk akhir RAD adalah sistem informasi baru.
- RAD adalah metodologi yang lengkap, dengan 4 fase siklus hidup yang sejajar dengan fase SDLC tradisional.
- Penggunaan RAD untuk mengurangi biaya dan waktu pengembangan, dan meningkatkan probabilitas keberhasilan

## Model RAD (Lanjutan)

- RAD sangat bergantung pada prototipe dan keterlibatan pengguna.
- Berdasarkan input pengguna, prototipe dimodifikasi dan proses interaktif berlanjut sampai sistem benar-benar dikembangkan dan pengguna puas.
- Interaksi berkelanjutan antara fase desain dan konstruksi pengguna
- Tim proyek menggunakan *CASE tools* untuk membangun prototipe dan membuat aliran dokumentasi yang berkelanjutan.

# Fase dan Aktivitas RAD



Sumber: Shelly (2012: 148)

# Fase RAD (Lanjutan)

## 1. REQUIREMENTS PLANNING

- Fase ini menggabungkan elemen-elemen perencanaan sistem dan fase analisis SDLC.
- Pengguna, manajer, dan anggota staf IT mendiskusikan dan menyetujui kebutuhan bisnis, ruang lingkup proyek, kendala, dan persyaratan sistem.
- Fase ini berakhir ketika tim menyetujui masalah-masalah utama dan mendapatkan izin manajemen untuk melanjutkan.

# Fase RAD (Lanjutan)

## 2. USER DESIGN

- Selama fase ini, pengguna berinteraksi dengan sistem analis kemudian mengembangkan model dan prototipe yang mewakili semua input, proses, output.
- Tim/subkelompok RAD biasanya menggunakan kombinasi teknik JAD (*Joint Application Development*) dan *CASE tools* untuk menerjemahkan kebutuhan pengguna ke dalam model.
- Desain pengguna adalah kontinyu, proses interaktif memungkinkan pengguna untuk memahami, memodifikasi, dan akhirnya menyetujui model kerja sistem yang memenuhi kebutuhan mereka.

## Fase RAD (Lanjutan)

### 3. CONSTRUCTION

- Fase konstruksi berfokus pada tugas pengembangan program dan aplikasi yang mirip dengan SDLC.
- Pengguna terus berpartisipasi dan masih dapat menyarankan perubahan atau peningkatan saat tampilan atau laporan aktual dikembangkan.

# Fase RAD (Lanjutan)

## 4. CUTOVER

- Merupakan fase peralihan, termasuk konversi data, pengujian, pergantian ke sistem baru, dan pelatihan pengguna.
- Dibandingkan dengan metode tradisional, seluruh proses dikompresi. Akibatnya, sistem baru dibangun, dikirim, dan ditempatkan dalam operasi yang lebih cepat (*agile*).



# Tujuan RAD

- a. Mengurangi waktu dan biaya pengembangan dengan melibatkan pengguna dalam setiap fase pengembangan sistem.
- b. Membuat modifikasi yang diperlukan dengan cepat, seiring dengan perkembangan desain.
- c. Membatasi biaya perubahan yang biasanya terjadi dalam jadwal pengembangan yang berlarut-larut.
- d. Membutuhkan sistem informasi untuk mendukung fungsi bisnis baru, karena digerakkan oleh pengguna.
- e. Dengan input pengguna, membantu tim pengembangan merancang sistem yang membutuhkan antarmuka pengguna yang interaktif atau kompleks.

# Keuntungan dan Kerugian RAD

## Keuntungan utama:

- sistem dapat dikembangkan lebih cepat dengan penghematan biaya yang signifikan

## Kerugian:

- Menekankan mekanisme sistem itu sendiri dan tidak menekankan kebutuhan strategis bisnis perusahaan.
- Baik untuk jangka pendek.
- Memungkinkan lebih sedikit waktu untuk mengembangkan kualitas, konsistensi, & standar desain.
- RAD dapat menjadi alternatif yang menarik, jika suatu organisasi memahami risiko yang mungkin terjadi

## 5. MODEL SCRUM

- *Scrum* adalah sebuah proses yang *agile* untuk menangani produk yang kompleks.
- Scrum digunakan untuk memandu kegiatan pengembangan dalam suatu proses yang mencakup kerangka kerja seperti: kebutuhan, analisis, desain, evolusi, dan pengiriman.
- Scrum menekankan penggunaan seperangkat pola proses PL yang telah terbukti efektif untuk proyek dengan jadwal yang ketat, perubahan kebutuhan, dan kekritisian bisnis

# Tahapan kegiatan pengembangan Scrum

## 1. *Backlog*

- Daftar prioritas kebutuhan proyek/fitur yang menyediakan nilai bisnis bagi pelanggan.
- Item dapat ditambahkan ke *backlog* kapan saja (diperkenalkan adanya perubahan).

# Tahapan kegiatan pengembangan Scrum (Lanjutan)

## 2. *Sprints*

- Unit kerja diperlukan untuk mencapai kebutuhan yang didefinisikan dalam *backlog* yang harus sesuai dengan *time box* yang telah didefinisikan sebelumnya (biasanya 30 hari).
- Time box adalah waktu yang telah dialokasikan untuk menyelesaikan beberapa tugas
- Tidak diperkenankan adanya perubahan, sehingga anggota tim bekerja dalam lingkungan jangka pendek tetapi stabil.

## Tahapan kegiatan pengembangan Scrum (Lanjutan)

### 3. *Scrum Meetings*

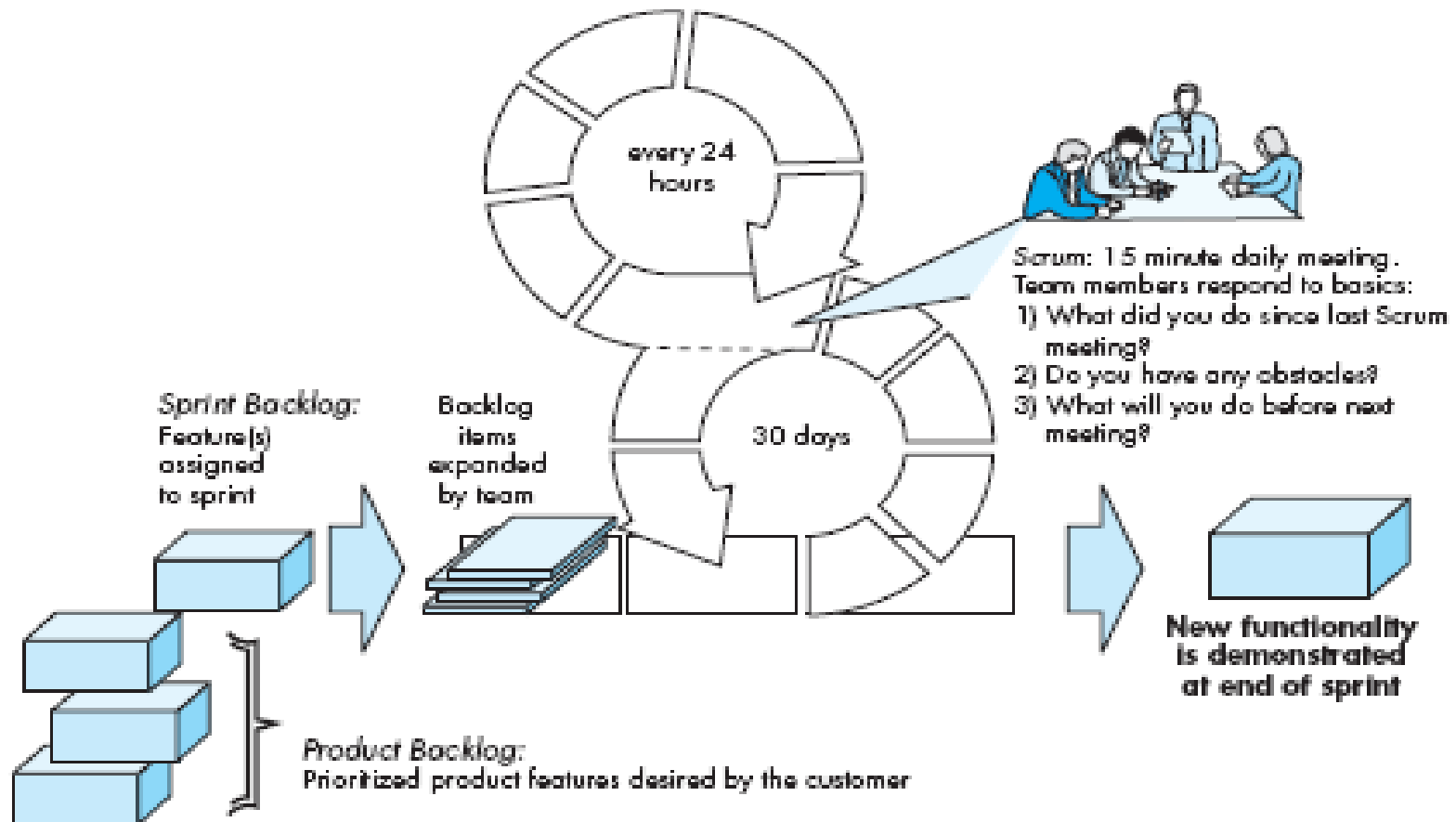
Waktunya pendek (biasanya 15 menit) tiap pertemuan yang diadakan setiap hari.

Pertemuan ini membantu tim untuk mengungkap potensi masalah sedini mungkin.

### 4. *Demo*

Memberikan peningkatan PL untuk pelanggan sehingga fungsionalitas yang telah diterapkan dapat didemonstrasikan dan dievaluasi oleh pelanggan.

# Model Scrum



Sumber: Pressman (2015: 78)

## Penjelasan gambar

- a. Gagasan awal adalah bahwa seluruh tim harus diberdayakan untuk membuat keputusan sehingga istilah 'Manajer Proyek', telah dengan sengaja dihindari.
- b. **Scrum Master** adalah seorang fasilitator yang mengatur
  - pertemuan harian
  - melacak *backlog* dari pekerjaan yang harus dilakukan
  - mencatat keputusan
  - mengukur kemajuan terhadap *backlog*
  - berkomunikasi dengan pelanggan dan manajemen di luar tim



## Penjelasan gambar (lanjutan)

- c. Seluruh tim menghadiri pertemuan harian, agar tetap fokus pada tugas masing-masing.
- d. Selama pertemuan, semua anggota tim berbagi informasi, melaporkan kemajuan mereka sejak pertemuan terakhir, menjelaskan masalah yang muncul, dan apa yang direncanakan untuk hari berikutnya.
- e. Ini berarti bahwa semua orang di tim tahu apa yang sedang terjadi, dan jika masalah muncul, dapat saling membantu.
- f. Semua orang berpartisipasi dalam perencanaan jangka pendek ini.

## Keuntungan Model *Scrum*

- a. Produk dipecah menjadi beberapa sub produk sehingga dapat dikelola dan dimengerti.
- b. Persyaratan yang tidak stabil tidak menghambat kemajuan.
- c. Seluruh tim memiliki visibilitas sehingga dapat meningkatkan komunikasi tim.
- d. Pelanggan melihat pengiriman tepat waktu dan mendapatkan umpan balik tentang bagaimana produk tersebut bekerja.
- e. Kepercayaan antara pelanggan dan pengembang akan terjalin dimana semua orang mengharapkan proyek untuk berhasil.

# Tugas

- Mahasiswa membuat rancangan pengembangan PL menggunakan salah satu model proses yang telah dibahas, dengan menjelaskan tahapan-tahapan SDLC seperti pada contoh slide sebelumnya
- Studi kasus ditentukan oleh Dosen dan tidak diperkenankan mengutip atau menulis ulang dari jurnal/artikel yang ada di internet
- Dikumpulkan pada pertemuan berikutnya