

# Information System Development Life Cycle

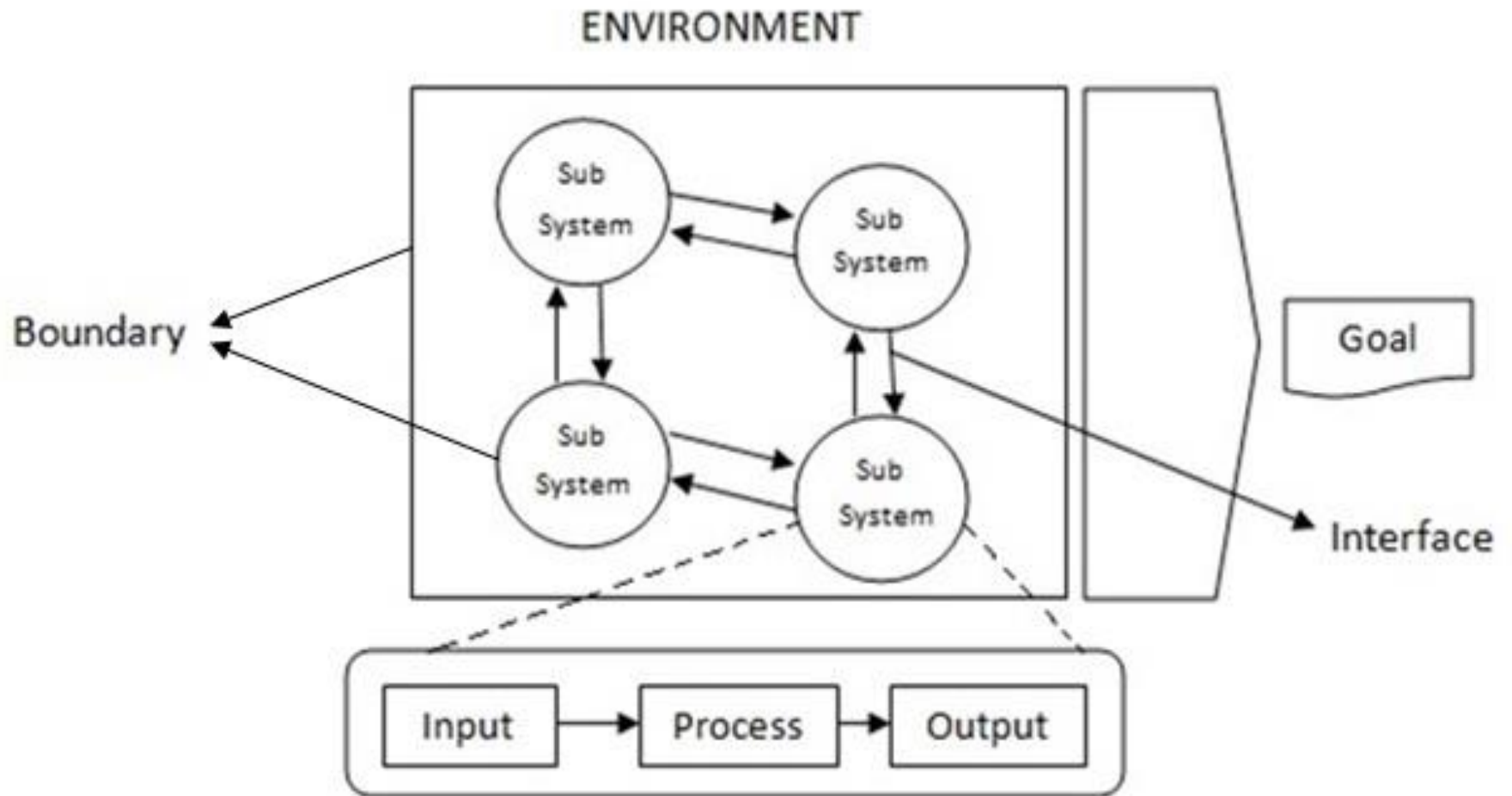
## Pengantar Sistem Informasi



# System

- System is a collection of related, interacting subsystems (components) combined and organized in such a fashion as to responsive to external stimuli and the influence and constraints of its environment, to accomplish a general or specific purpose

# System





How the customer explained it



How the Project Leader understood it



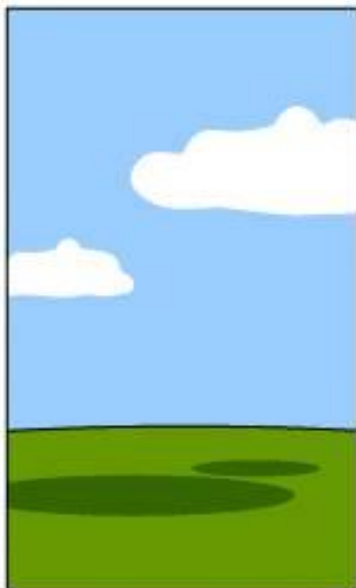
How the Analyst designed it



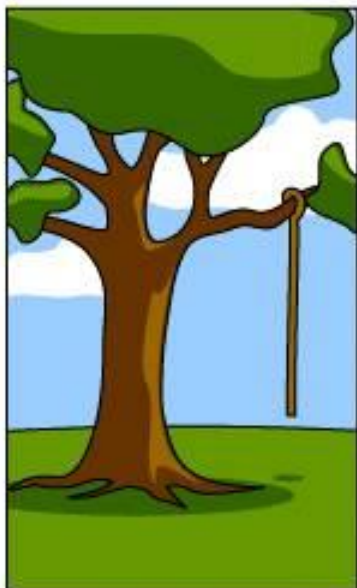
How the Programmer wrote it



How the Business Consultant described it



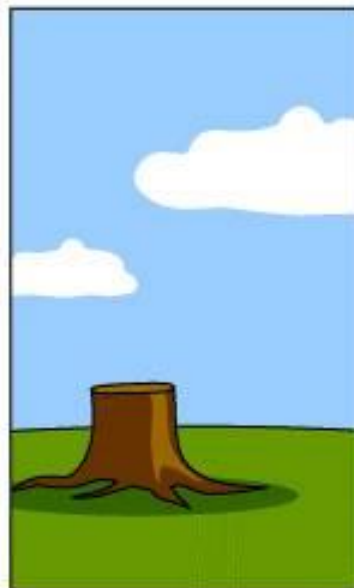
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed



# Cara mendapatkan Sistem Informasi

- Turn-key system : membeli sistem dan program jadi
- Taylor-made/customized system : memesan kepada tim pembangun sistem (software house)
- Out-sourcing system : pihak luar mengolah data perusahaan.



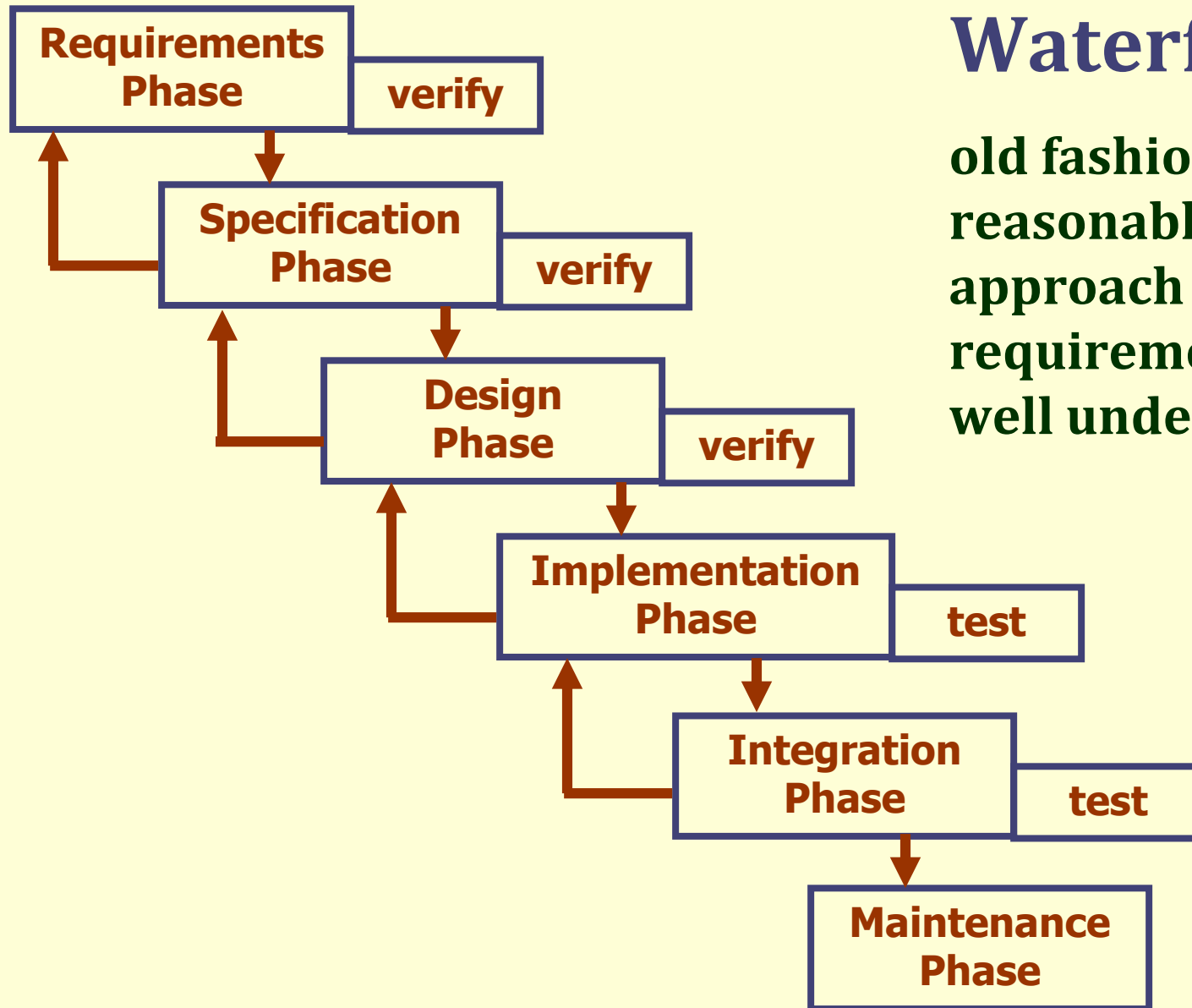
# Cara mendapatkan Program Komputer

- Shareware : mencoba menggunakan program komputer yang diminati, bila cocok dibeli.
- Download software : mengambil program gratis, atau membayar di awal
- Rental software : menyewa, agar selalu mutakhir



# IS Development Life Cycle

- Waterfall
- Prototyping
- Incremental
- Spiral
- Agile



# Waterfall

old fashioned but  
reasonable  
approach when  
requirements are  
well understood





# The Waterfall Model Problem

- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
- Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.



# Requirement Phase

- Tawaran pekerjaan : Term of Reference (Kerangka Acuan Kerja)
- Tim mempelajari dokumen TOR dan membuat dokumen tanggapan terhadap TOR.
- Tim menyusun dokumen Proposal yang dilengkapi dengan Rencana Anggaran Biaya
- Proposal memuat kebutuhan pembangunan sistem secara lengkap untuk setiap sumber daya : manusia, dana, waktu/jadwal (Gantt chart), sarana perangkat keras, bahan data/informasi
- Manajemen proyek : struktur tim kerja, deskripsi kerja, Work Breakdown Structure (WBS)
- Penjadwalan sesuai WBS



# Specification Phase

Proposal Proyek menjadi dasar untuk :

- Menyusun spesifikasi sistem, spesifikasi pengguna, spesifikasi perangkat lunak, spesifikasi perangkat keras
- Menyusun spesifikasi produk PL. Menguraikan secara detail fitur-fitur yang akan dibangun, dan semua format keluaran yang dapat dihasilkan sistem PL.
- Analisis proses bisnis antara as-is system (sistem kini) dengan to-be system (system usulan)
- Menyusun prosedur dan aturan sesuai hak akses dan jenis pengguna
- Menyusun kebutuhan basisdata, spesifikasi entitas data dan relasi antar entitas data



# Specification Phase

- Memodelkan kebutuhan sistem dan kebutuhan perangkat lunak serta kebutuhan basisdata menggunakan diagram-diagram yang tepat.
- Menyusun kandidat objek, dilanjutkan dengan memodelkan kebutuhan kelas-objek
- Jika sudah berpengalaman, phase ini akan lebih cepat, karena dapat memanfaatkan modul2 sub-sistem yang pernah dibangun sebelumnya
- Dokumen yang dihasilkan : Spesifikasi Kebutuhan Sistem
- System Requirement Specification (SRS)
- Dokumen dilengkapi sejumlah diagram pemodelan kebutuhan sistem



# Design Phase

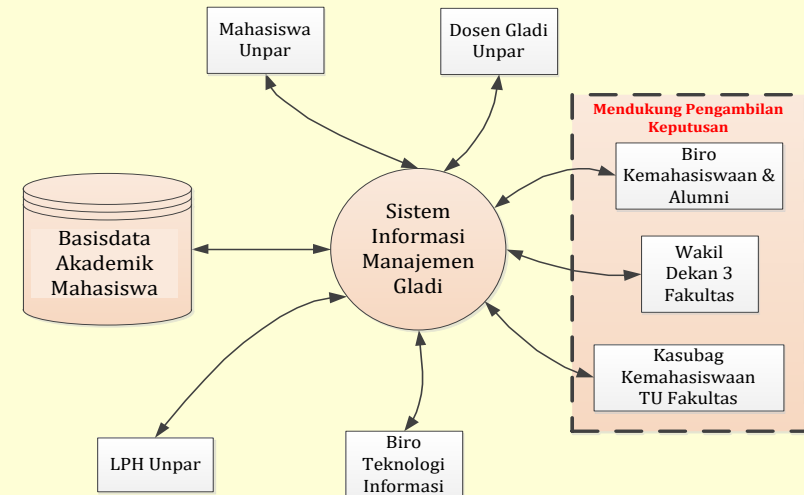
SRS menjadi dasar untuk :

- Merancang antar muka dengan pengguna
- Merancang tabel-tabel basisdata
- Merancang kelas-objek rinci dilanjutkan dengan merancang algoritma metode-metode yang dibutuhkan setiap kelas-objek
- Merancang struktur modul dilanjutkan dengan merancang algoritma setiap modul
- Dokumen yang dihasilkan : Design Software System

# Design Phase

SRS menjadi dasar untuk :

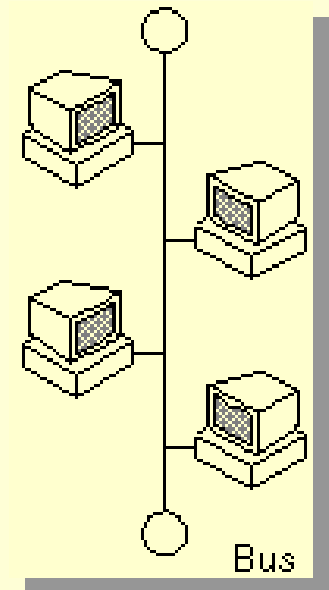
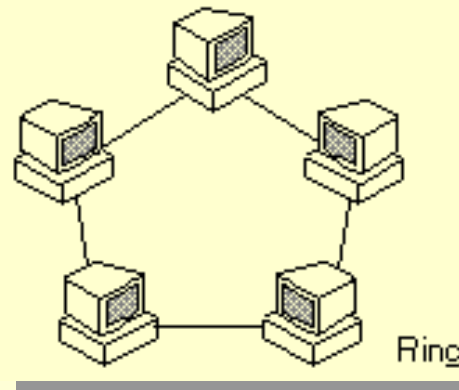
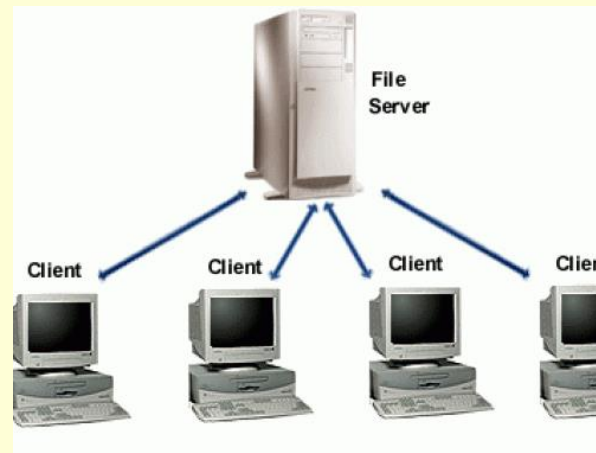
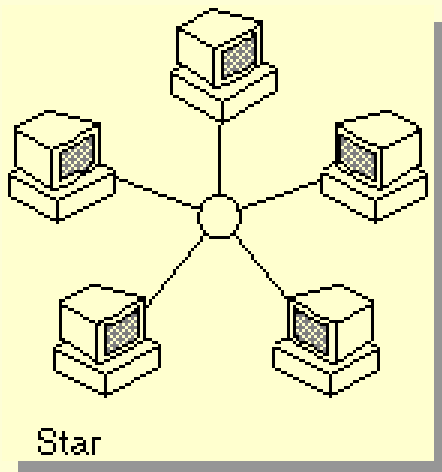
- Merancang antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi



# Design Phase

SRS menjadi dasar untuk :

- Jika dibutuhkan : merancang antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi





# Implementation Phase

DSS menjadi dasar untuk :

- Menyediakan perangkat keras, perangkat lunak, media perantara, sistem eksternal, dll
- Koding/membangun program
- Koding/membangun basisdata
- Pengujian modul/unit
- Dokumen yang dihasilkan : Source Code Document
- Test-Case : Uji : modul, unit, integrasi, black box, white box, alpha, beta
- Dokumen yang dihasilkan : Testing Document





# Integration Phase

Seluruh dokumen menjadi dasar :

- Instalasi server
- Instalasi sistem
- Instalasi basisdata
- Dokumen yang dihasilkan : Acuan Teknis : panduan instalasi, panduan/modul pelatihan, petunjuk operasional
- Ujicoba
- Pelatihan bagi pengguna tahap-1
- Peralihan dari as-is sistem ke to-be sistem
- Pelatihan bagi pengguna tahap-2

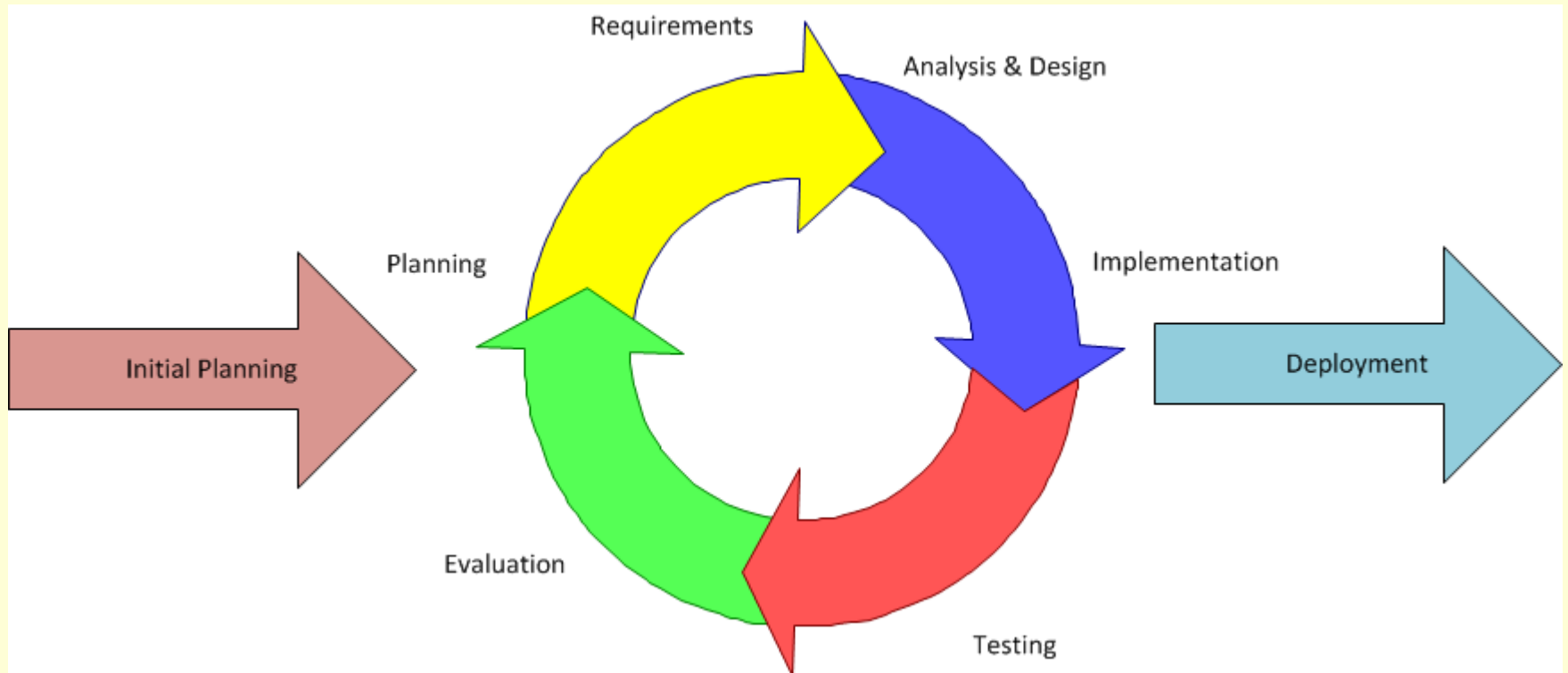


# Maintenance Phase

Dokumen Acuan Teknis menjadi dasar :

- Memperbaiki tampilan antarmuka agar semakin menyamankan pengguna
- Memperbaiki panduan penggunaan agar semakin mudah dipahami pengguna
- Memperbaiki kualitas jaringan apabila terjadi keluhan akses ke sistem
- Memperbaiki format keluaran (hasil cetak) apabila diperlukan
- Mengakomodasi media komunikasi dengan para pengguna, agar proses pemeliharaan sistem menjadi efisien
- Pekerjaan proyek sistem berakhir.

# Prototyping Model

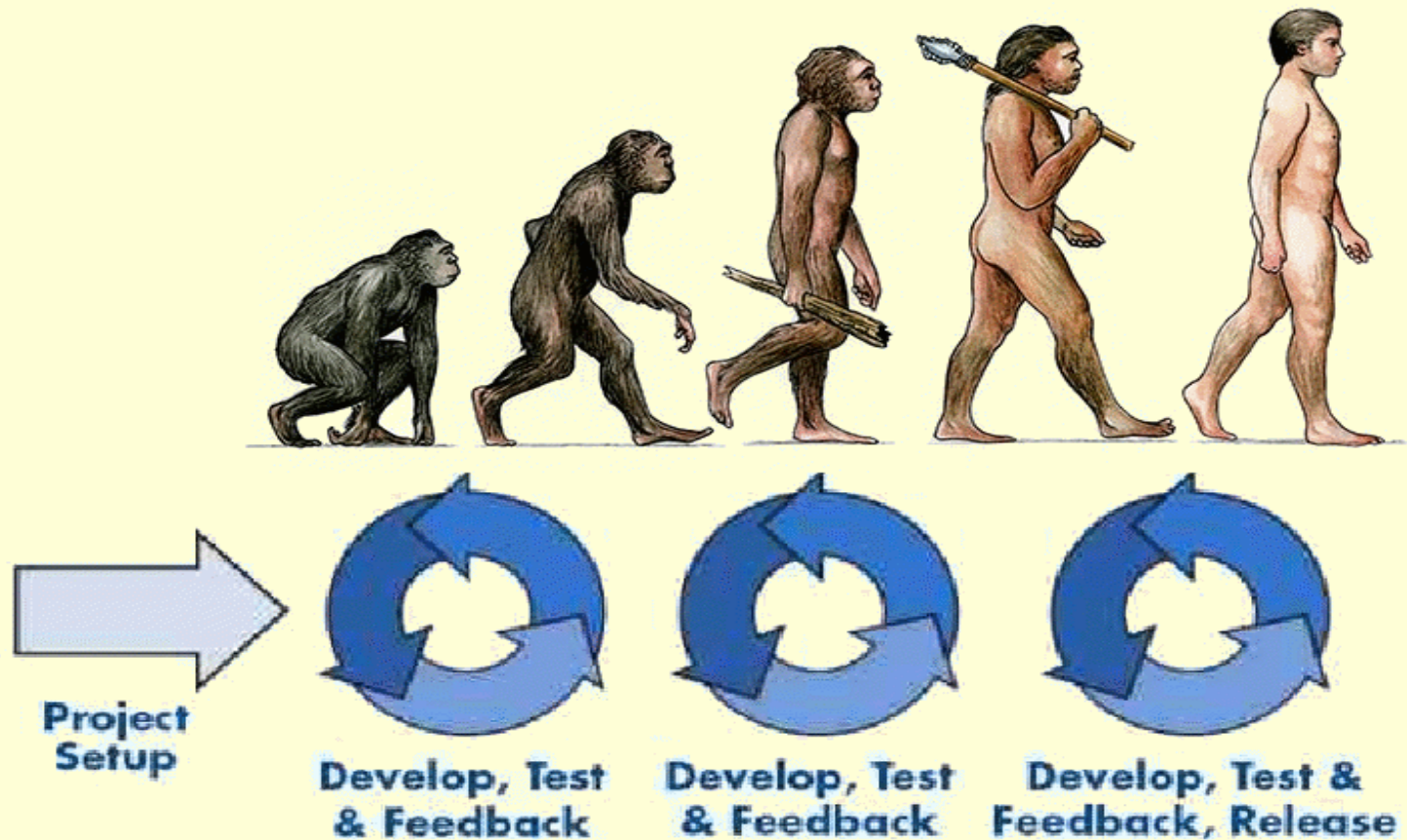




# Tahap-tahap Model Prototyping

- Mengidentifikasi kebutuhan dasar dari pengguna, yang melingkup data/informasi pada input/output PL. Detil kebutuhan lainnya, seperti keamanan data, dapat diabaikan.
- Mengembangkan prototipe awal PL dengan cepat. Kelengkapan PL diutamakan pada antar-muka dengan pengguna (berbentuk form dan laporan).
- Pengguna melakukan uji-coba terhadap PL dengan mencoba menggunakan PL tersebut, lalu memberikan umpan-balik kepada tim pembangun PL.

# Prototyping Model





# Model Prototyping

- Memberikan kesempatan kepada para pengguna PL untuk mengevaluasi produk PL (dengan mencoba menggunakannya sendiri) sebelum dihasilkan produk final. Dengan pendekatan ini, kepuasan pengguna terhadap PL dapat lebih dijamin
- Proses ini diulang-ulang sampai para pengguna puas dan dapat menerima PL, yang lalu dinyatakan sebagai PL final



# Karakteristik Prototyping

- Cocok untuk tingkat interaktif yang cukup rendah dengan pengguna, dan lebih banyak melakukan komputasi.
- Cocok untuk pengguna berkarakteristik khusus misalnya dalam hal selera dan filosofi dalam antarmuka yang mungkin sulit dipahami oleh tim pembangun PL



# Kekuatan dan Kelemahan Prototyping

- Mengurangi waktu dan biaya pembangunan PL.
- Melibatkan pengguna dalam proses pembangunan, sehingga kepuasan pengguna dapat lebih dijamin.
- Analisis pada pembangunan prototipe tidak lengkap, sehingga dimungkinkan adanya kesalahan-kesalahan.
- Menimbulkan “kebingungan”, pengguna sulit membedakan prototipe PL yang belum jadi dengan produk PL final.
- Tim pembangun PL sulit “melepaskan diri” dari prototipe dan cenderung untuk mempertahankannya meski ada kesalahan-kesalahan pada prototipe.
- Jika tim pembangun PL tidak mampu untuk mengendalikan bertambahnya kebutuhan, maka waktu pembangunan dapat menjadi lama.

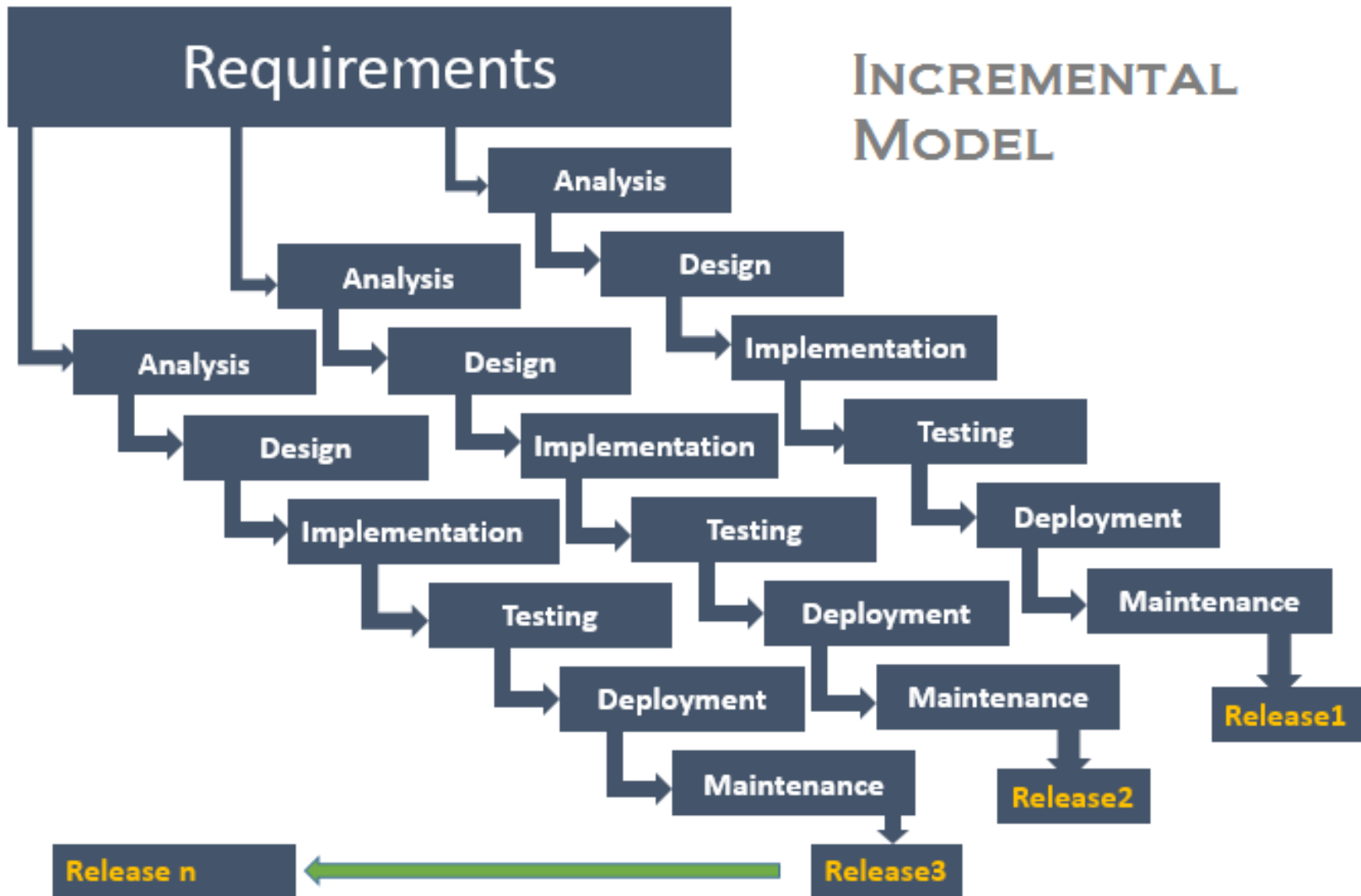




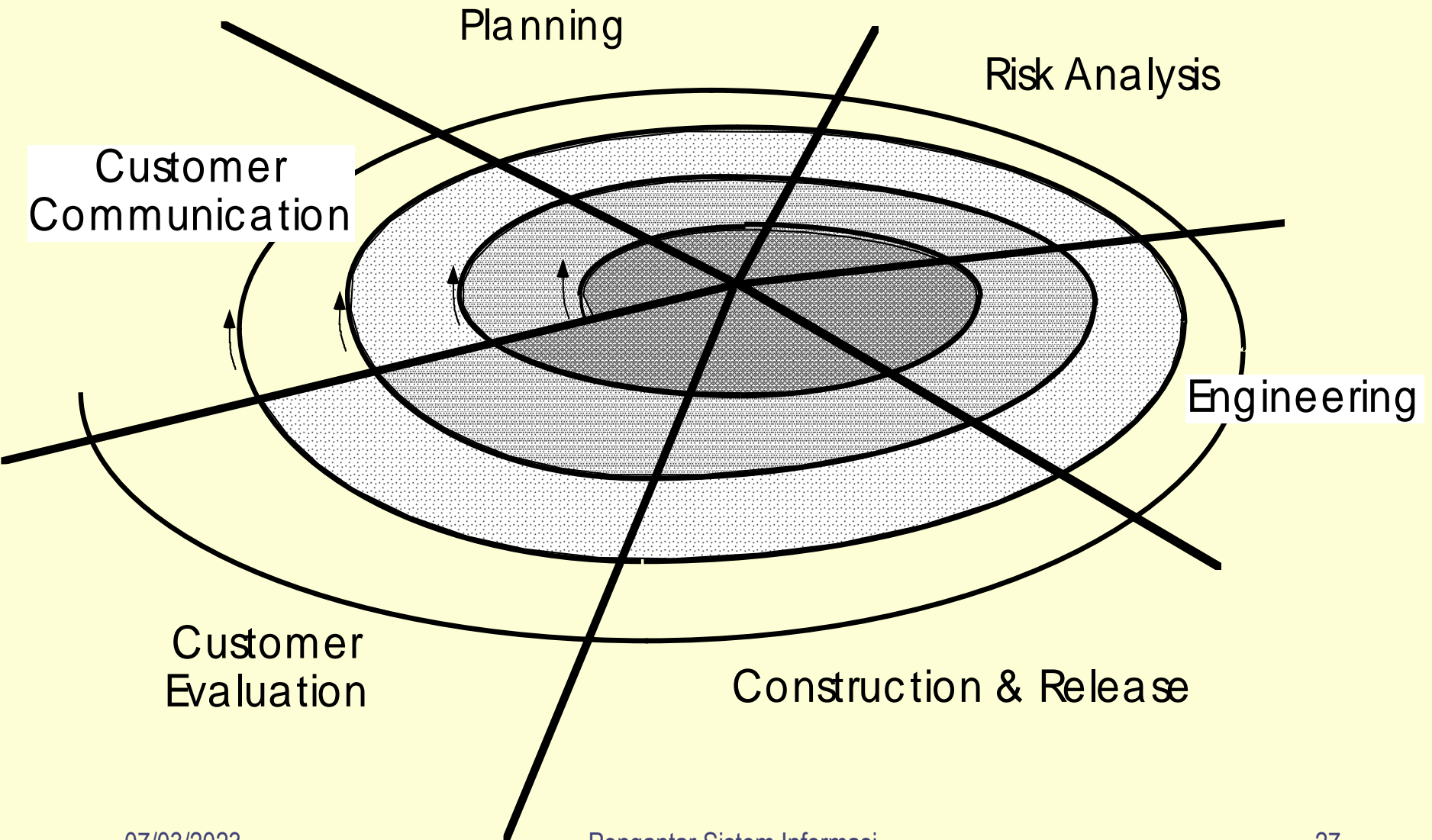
# Incremental Model

- A process of software development where requirements are broken down into multiple standalone modules of software development cycle
- This model combines the elements of the waterfall model with the iterative philosophy of prototyping
- The product is decomposed into a number of components, each of which is designed and built separately
- Each component is delivered to the client when it is complete. This allows partial utilization of the product and avoids a long development time. It also avoids a large initial capital outlay and subsequent long waiting period.
- Helps ease the traumatic effect of introducing a completely new system all at once

# Incremental Model



# Spiral Model



# Spiral Model

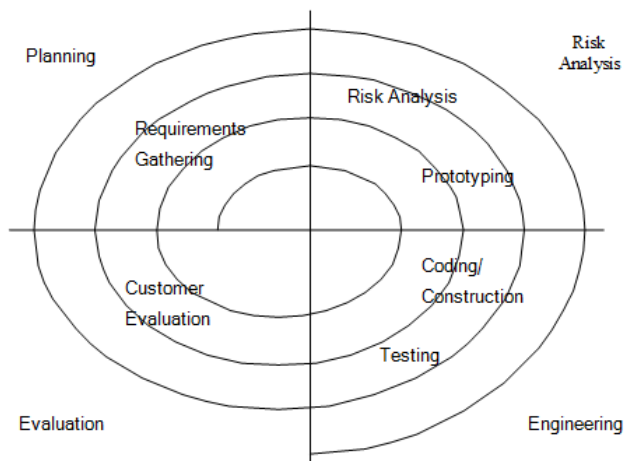
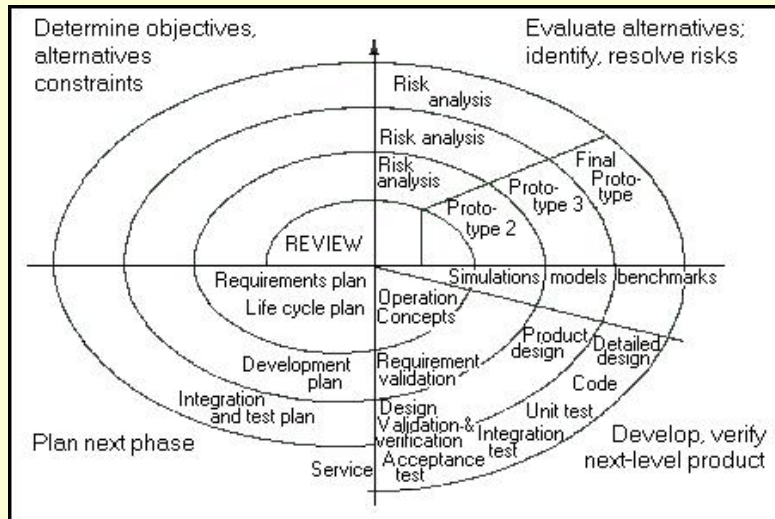
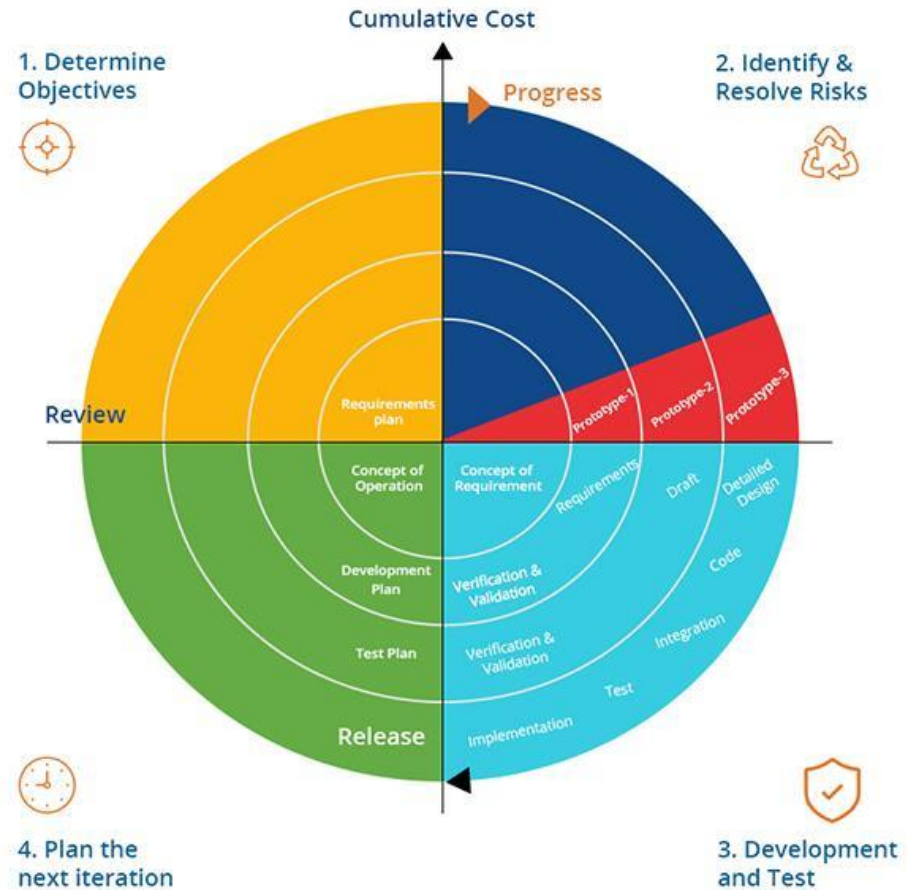
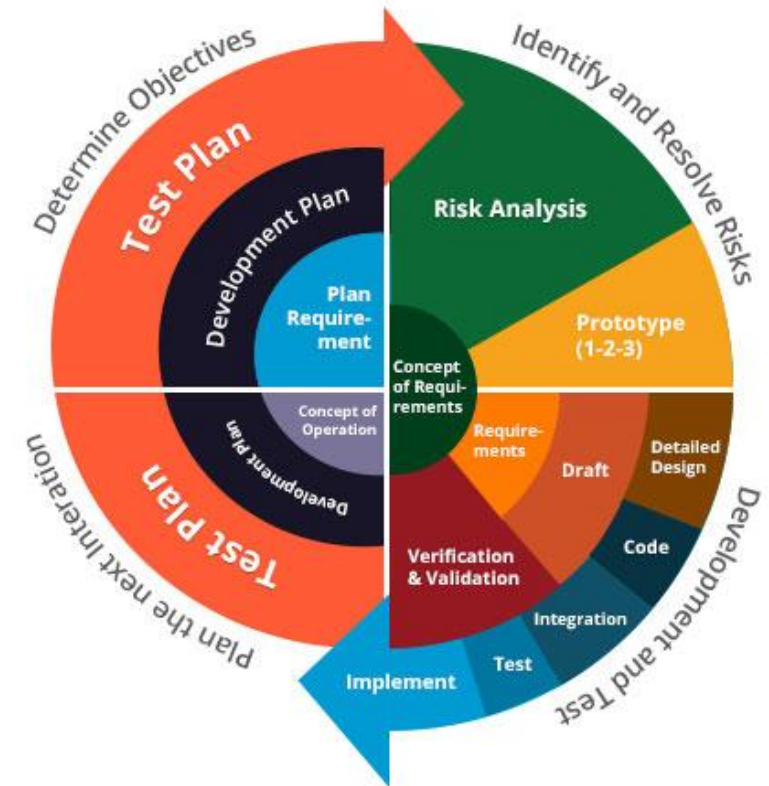
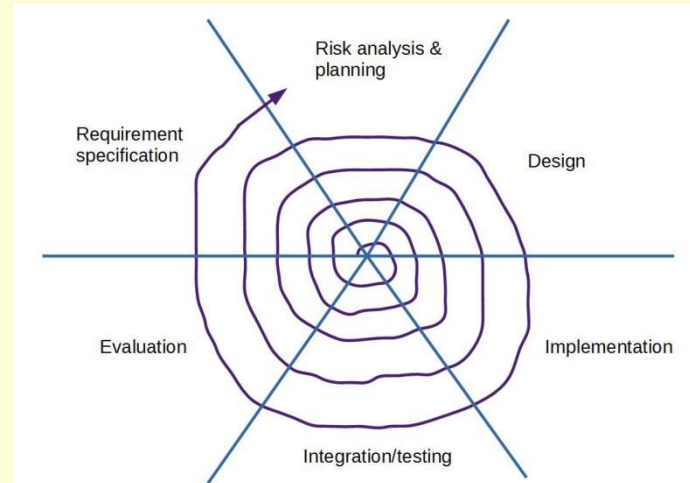
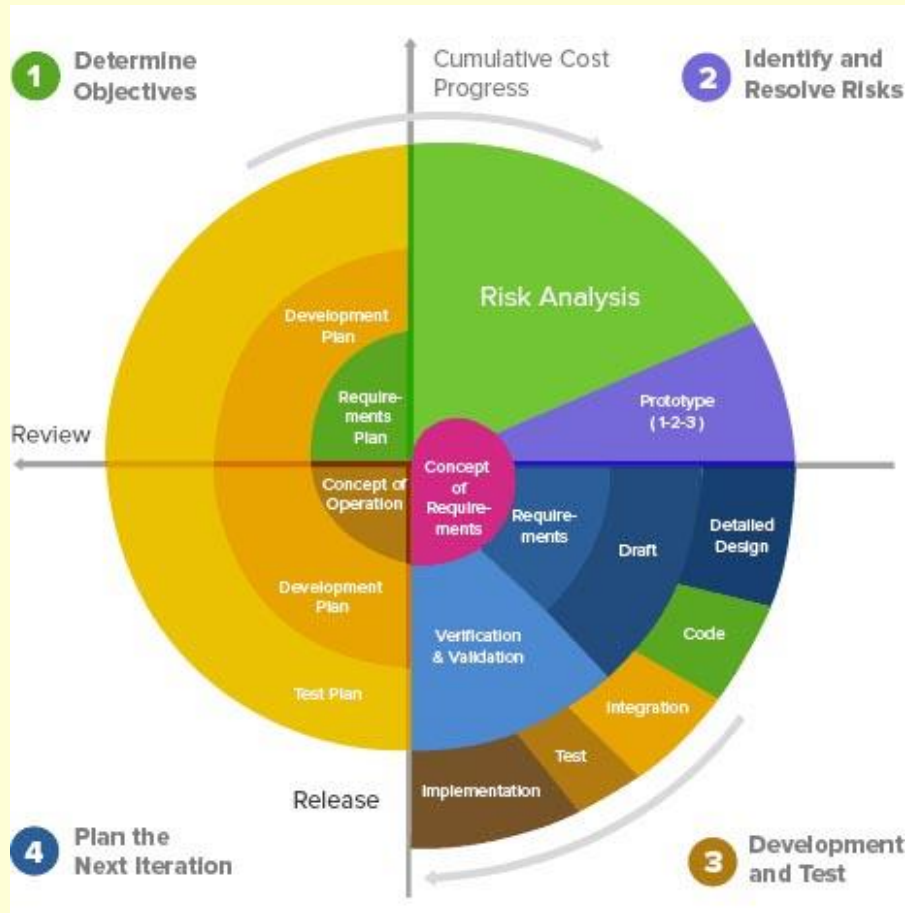


Fig. Spiral Model



# Spiral Model



# Agile Model

Not like this....



1



2



3



4

Like this!



1



2



3



4



5

Henrik Kniberg



Agile is iterative, incremental but  
iterative waterfall is not agile!

07/03/2023

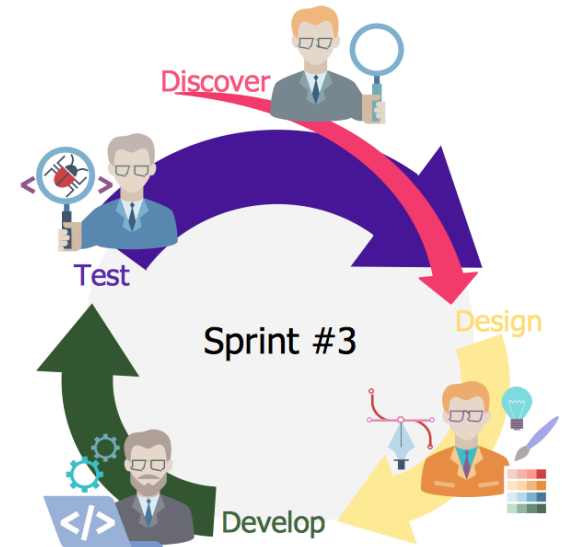
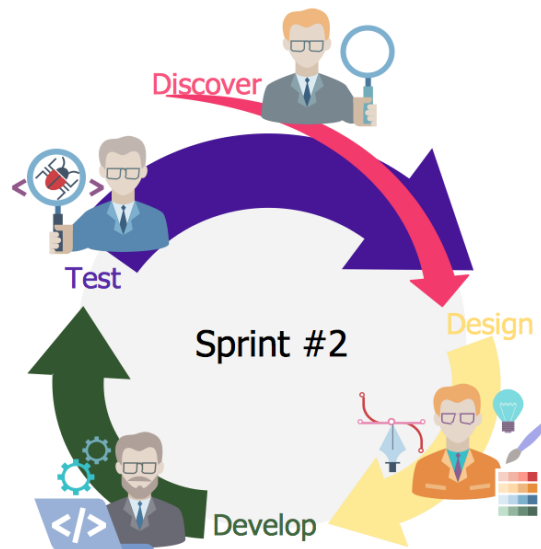
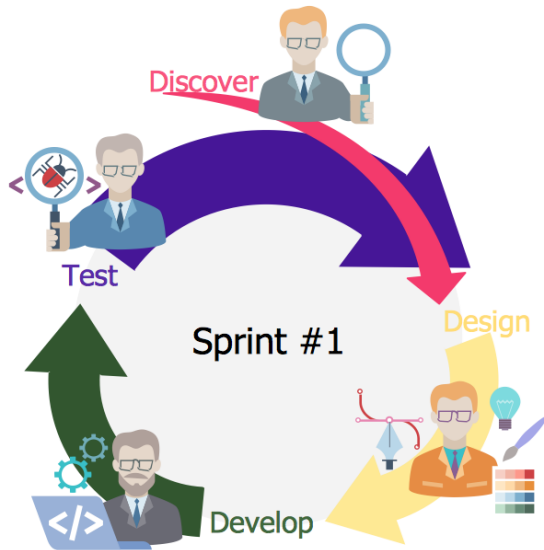




# Agile Model

- Pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun
- Diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.
- Kebutuhan digali dan solusi dibangun, keduanya dilakukan bersama-sama oleh tim pembangun dan pemilik pekerjaan
- Perencanaan adaptif, pengembangan secara evolusi, pendekatan iteratif berbasis waktu.
- Cermat dan fleksibel terhadap kebutuhan baru (perubahan)

# Agile methodologie







# Agile Alliance's Manifesto

- **Interaksi dan personel** lebih penting daripada proses dan alat, di dalam agile interaksi antar anggota tim sangatlah penting, karena tanpa adanya interaksi yang baik maka proses pembuatan perangkat lunak tidak akan berjalan sesuai rencana.
- **Perangkat lunak yang berfungsi** lebih penting daripada dokumentasi yang lengkap, saat melakukan demo dan pelatihan kepada klien, perangkat lunak yang berfungsi dengan baik akan lebih berguna daripada dokumentasi yang lengkap.



# Agile Alliance's Manifesto

- **Kolaborasi dengan klien** lebih penting daripada negosiasi kontrak, salah satu ciri dari agile adalah klien menjadi bagian dari tim pengembangan perangkat lunak. Kolaborasi yang baik dengan klien saat proses pembuatan perangkat lunak sangatlah penting ketika menggunakan agile. Karena fungsi-fungsi dari perangkat lunak yang dikembangkan harus terus menerus dibicarakan dan diimprovisasi disesuaikan dengan keinginan klien.
- **Respon terhadap perubahan** lebih penting daripada mengikuti rencana, *agile development methods* berfokus terhadap kecepatan respon tim ketika klien menginginkan perubahan saat proses pembuatan perangkat lunak.

# Karakteristik Agile

- Kebutuhan perangkat lunak sulit diprediksi dari awal dan selalu akan berubah. Selain itu, prioritas klien juga sering berubah seiring berjalannya proyek.
- Desain dan pembangunan sering tumpang tindih. Sulit diperkirakan seberapa jauh desain yang diperlukan sebelum pembangunan.
- Analisis, desain, pembangunan dan testing tidak dapat diperkirakan seperti yang diinginkan.

## Peran :

- Klien : Terlibat dalam menentukan fungsi dari perangkat lunak yang akan dibuat, melakukan pengujian dan memberikan umpan balik.
- Manajer Tim : mengkolaborasikan developer dengan klien, membuat dan mengevaluasi target pengerjaan perangkat lunak
- Analis sistem & tim pengembang proyek

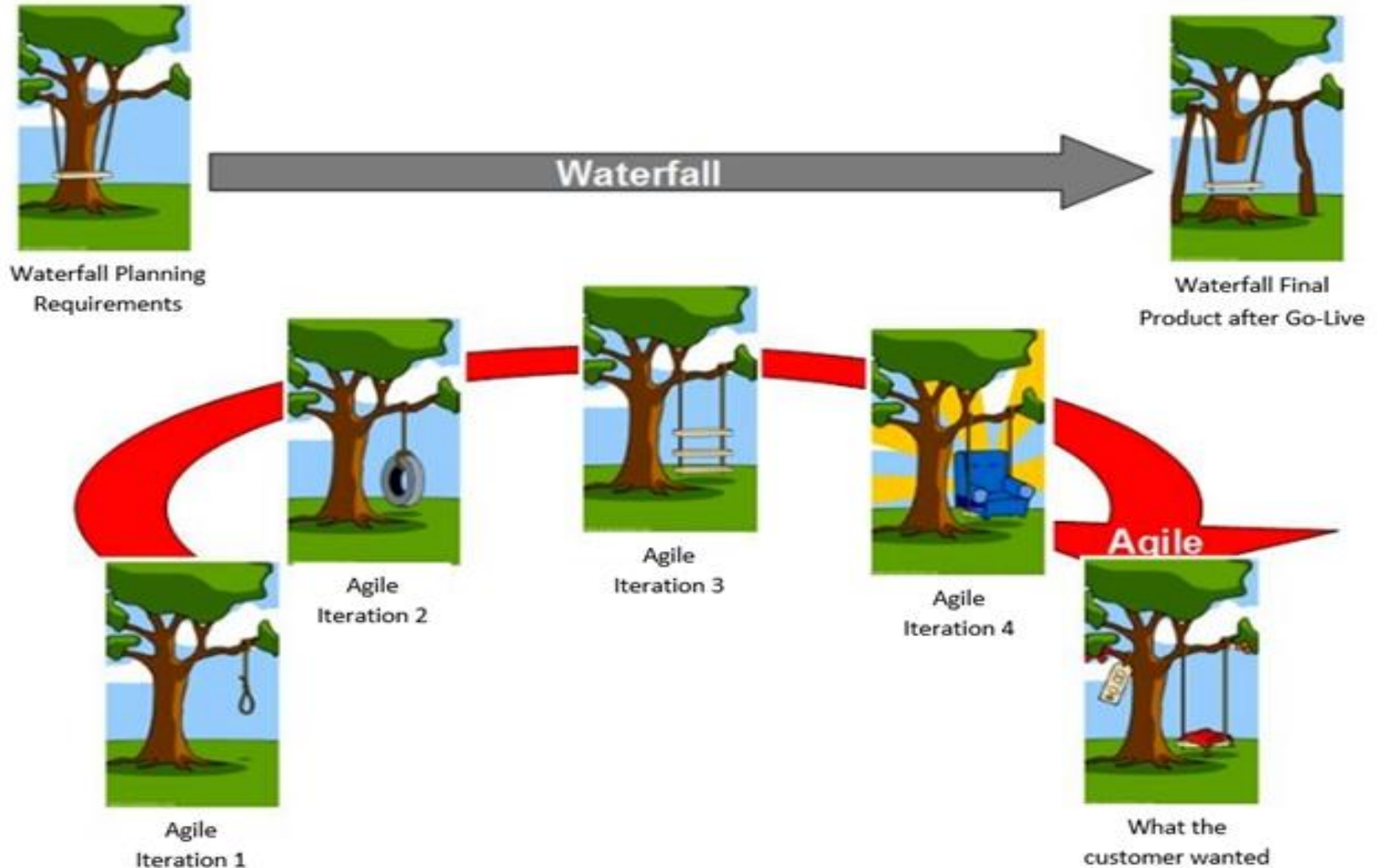


## +/- Agile

- 82% Menambah produktivitas tim.
  - 77% Menambah kualitas perangkat lunak.
  - 78% Menambah kepuasan klien.
  - 37% Menghemat biaya.
- 
- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
  - Tidak cocok dalam skala tim yang besar (>20 orang).
  - Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

<b>Agile</b>	<b>Waterfall</b>
Architecture is informal and incremental	Architecture is very well documented and completed before coding starts
Developers share ownership of code	Each developer is responsible for one area
<a href="#">Continuous integration</a>	Integration performed at the end or after milestones
Focus on completing stories (functionality) in short iterations	Focus on completing modules (parts of the architecture) at different large milestones
Relies on engineering practices (TDD, refactoring, design patterns...)	Doesn't necessarily rely on engineering practices.
Light process and documentation	Heavy process and documentation
Requires cross-trained developers, knowledgeable in all required technologies	Relies on a small group of architects/designers to overview the complete code, the rest of the team can be very specialized.
Main roles: Developer	Main roles: Architect, Developer
Open door policy. Developers are encouraged to talk directly with business, QA and management at any time. Everyone's point of view is considered.	Only a few developers, and some architects can contact business people. Communication mainly only happens at the beginning of the project and at milestones.

# Agile vs Waterfall



# Selamat Belajar

## Pengantar Sistem Informasi