

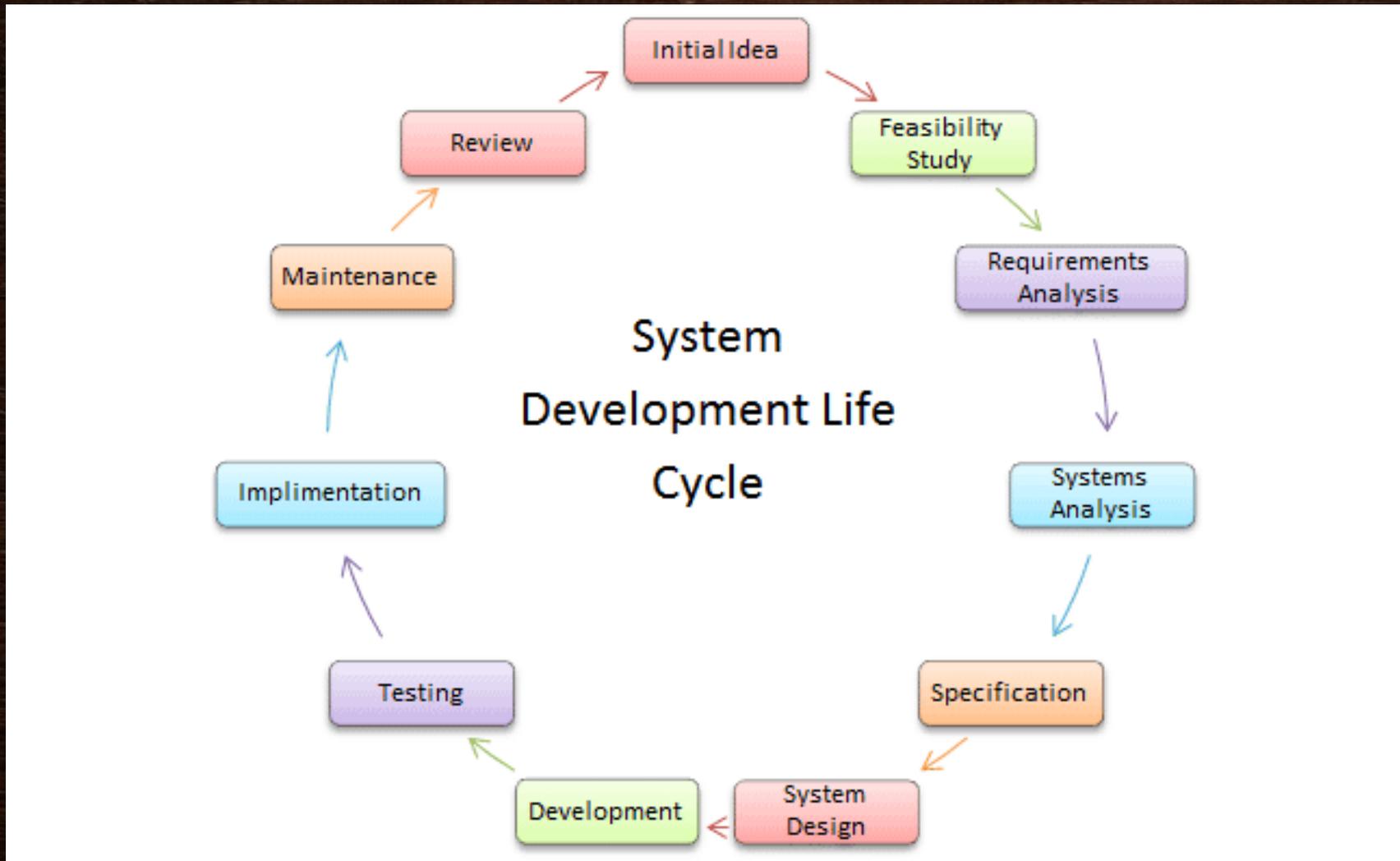
SDLC  
*(System Development  
Life Cycle)*

OLEH; RAHMAT ROBI WALIYANSYAH, M.KOM

# DEFINISI SDLC

- Menurut Gustafson (dalam buku “theory and problems of software engineering”, 2002) adalah urutan dari kegiatan yang ada di dalam sebuah pengembangan perangkat lunak.

# SDLC (*System Development Life Cycle*)



# SDLC (SystemDevelopment Life Cycle)

- SDLC ini tidak hanya penting untuk proses **produksi** software saja, namun juga sangat penting untuk proses **maintenance** software itu sendiri.
- Tanpa pengarsipan data-data development suatu software, maka akan sangat menyulitkan perusahaan dalam maintenance software tersebut dikemudian hari.

# Kegunaan SDLC

- Kegunaan utama dari SDLC adalah mengakomodasi beberapa kebutuhan.
- Kebutuhan-kebutuhan itu berasal dari kebutuhan pengguna akhir dan juga perbaikan sejumlah masalah yang terkait dengan pengembangan perangkat lunak.

# AGILE SOFTWARE DEVELOPMENT

- Agile merupakan jenis pengembangan sistem jangka pendek yang memerlukan **adaptasi cepat** dan pengembang terhadap **perubahan dalam bentuk apapun**.
- Dalam *Agile Software Development* **interaksi** dan **personel** lebih penting dari pada **proses** dan **alat**, **software** yang berfungsi lebih penting daripada **dokumentasi yang lengkap**, **kolaborasi** dengan **klien** lebih penting dari pada **negosiasi kontrak**, dan **sikap tanggap** terhadap perubahan lebih penting daripada **mengikuti rencana**.

# Menurut Agile Alliance, ada 12 prinsip yang mendorong keberhasilan dalam penerapan Agile Software Development, yaitu :

- Kepuasan klien
- Menerima perubahan kebutuhan.
- Penyerahan hasil/software dalam hitungan waktu beberapa minggu sampai beberapa bulan.
- Pihak bisnis dan pengembang harus bekerja sama setiap hari selama pengembangan berjalan.
- Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan dipercaya untuk dapat menyelesaikan proyek.
- Komunikasi dengan berhadapan langsung.

Menurut Agile Alliance, ada 12 prinsip yang mendorong keberhasilan dalam penerapan Agile Software Development, yaitu:

- Software yang berfungsi.
- Dukungan yang stabil dari sponsor.
- Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile.
- Kesederhanaan penting.
- Arsitektur, kebutuhan dan desain yang bagus.
- Secara periodik tim evaluasi diri.

# Kelebihan dari Agile Software Development yaitu :

- Meningkatkan kepuasan kepada klien.
- Pembangunan system dibuat lebih cepat.
- Mengurangi resiko kegagalan implementasi software dari segi non-teknis.
- Jika pada saat pembangunan system terjadi kegagalan, kerugian dari segi materi cenderung relative kecil.

# Kekurangan dari Agile Software Development yaitu :

- Agile jarang dipraktekkan secara langsung.
- Interksi dengan customers yang berlebihan.
- Agile sulit diimplementasikan dalam proyek yang berskala besar.
- Membutuhkan manajemen tim yang terlatih.
- Lemah dalam perencanaan arsitektur (Scrum dan Extreme Programming).
- Keterbatasan waktu dalam perencanaan Proyek.

**Berikut beberapa model proses yang terdapat pada model Proses Agile :**

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dinamic System Development Method (DSDM)
- SCRUM

# Extreme Programming (XP)

- Extreme Programming adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel.
- XP bukan hanya berfokus pada coding tetapi meliputi seluruh area pengembangan perangkat lunak.

# Extreme Programming (XP)

- XP Pertama kali diusulkan oleh Kent Beck dan Ward Cunningham pada bulan Maret 1996 dengan menggunakan pendekatan OOP (Object Oriented Programming), terdiri dari : aktivitas perencanaan, aktivitas desain, aktivitas pengkodean dan aktivitas pengujian.
- Asal mula XP digunakan karena pada saat itu permintaan dari customer yang sering berubah dengan cepat sehingga mengakibatkan perputaran kehidupan metode pengembangan perangkat lunak menjadi lebih pendek dan tidak selaras.
- Tujuan XP adalah meminimalisir biaya yang diperlukan jika ada perubahan dalam pengembangan perangkat lunak.

# XP tepat digunakan saat kondisi :

- Keperluan berubah dengan cepat
- Resiko tinggi dan ada proyek dengan tantangan yang baru
- Tim programmer sedikit, yaitu 2-10 orang
- Mampu mengotomatiskan tes
- Ada peran serta pelanggan secara langsung

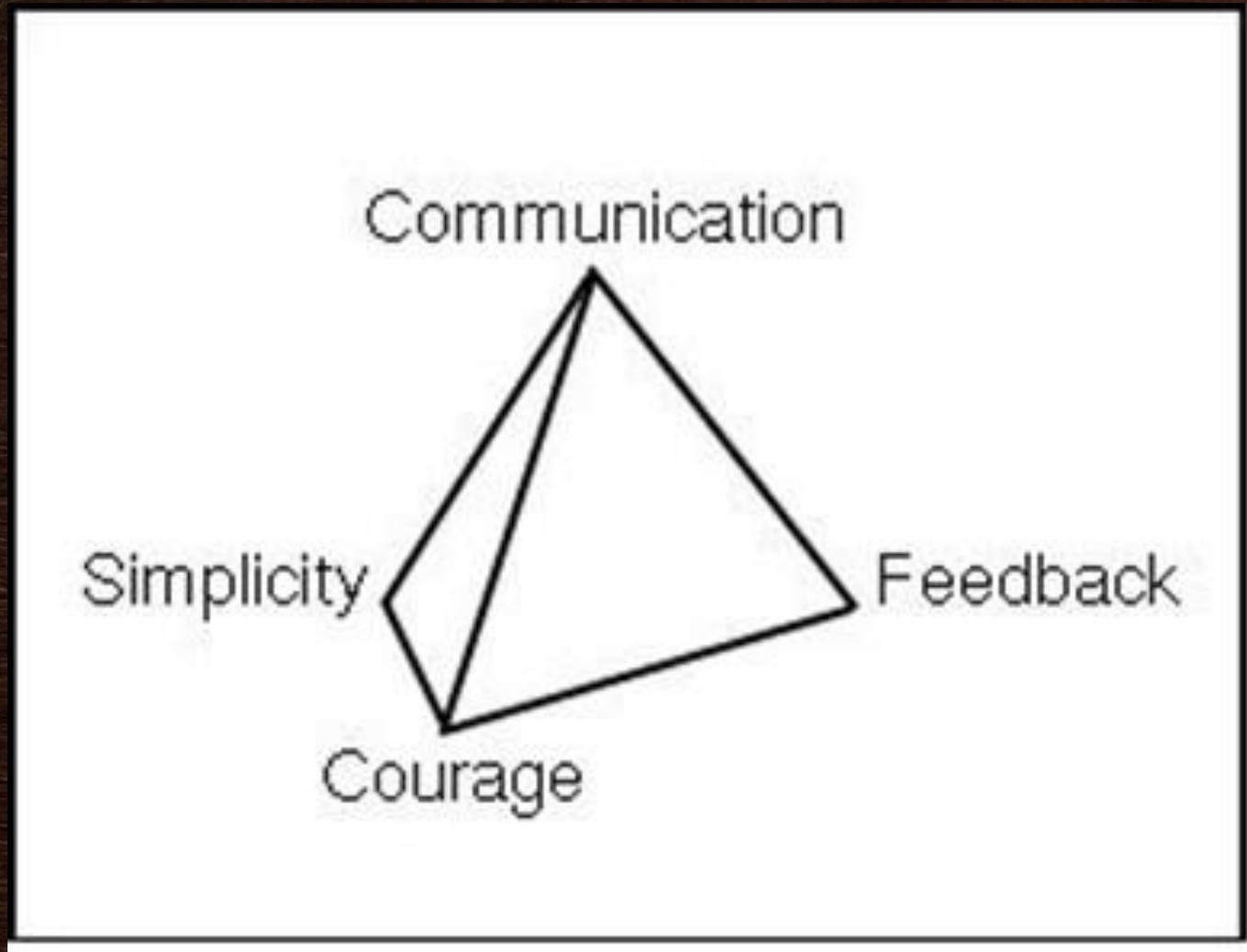
# Kelebihan Extreme Programming, yaitu :

- Meningkatkan kepuasan kepada klien.
- Pembangunan system dibuat lebih cepat.
- Menjalinkan komunikasi yang baik dengan client.
- Meningkatkan komunikasi dan sifat saling menghargai antar developer.

# Kelemahan XP :

- Cerita-cerita yang menunjukkan requirements kemungkinan besar tidak lengkap sehingga Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
- Tidak bisa membuat kode yang detail di awal (prinsip simplicity dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).
- XP tidak memiliki dokumentasi formal yang dibuat selama pengembangan. Satu-satunya dokumentasi adalah dokumentasi awal yang dilakukan oleh user.

# Core Value Extreme Programming



# Core Value Extreme Programming

## Komunikasi (Communication)

- Kurangnya komunikasi merupakan penyebab utama kegagalan pengembangan software. Oleh karena itu Extreme Programming (XP) memfokuskan diri pada hubungan komunikasi yang baik antar tim-klien, anggota tim, dan manajer proyek.
- Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (pair programming).
- Klien harus dilibatkan dalam proses pengembangan perangkat lunaknya dengan tujuan untuk memberikan pandangan pengembang sesuai dengan pandangan pengguna sistem yang dibangun.

# Core Value Extreme Programming

## Kesederhanaan (Simplicity)

- Extreme Programming (XP) melakukan semua pekerjaan dengan sederhana dan praktis tanpa mengurangi fungsi utamanya.
- Dalam pengerjaan, metode yang dipilih adalah metode yang pendek dan simpel.
- Jangan terlalu rumit dalam membuat desain, hilangkan fitur yang tidak ada gunanya atau hapus fungsi yang tidak terpakai.

# Core Value Extreme Programming

## Umpan balik (Feedback)

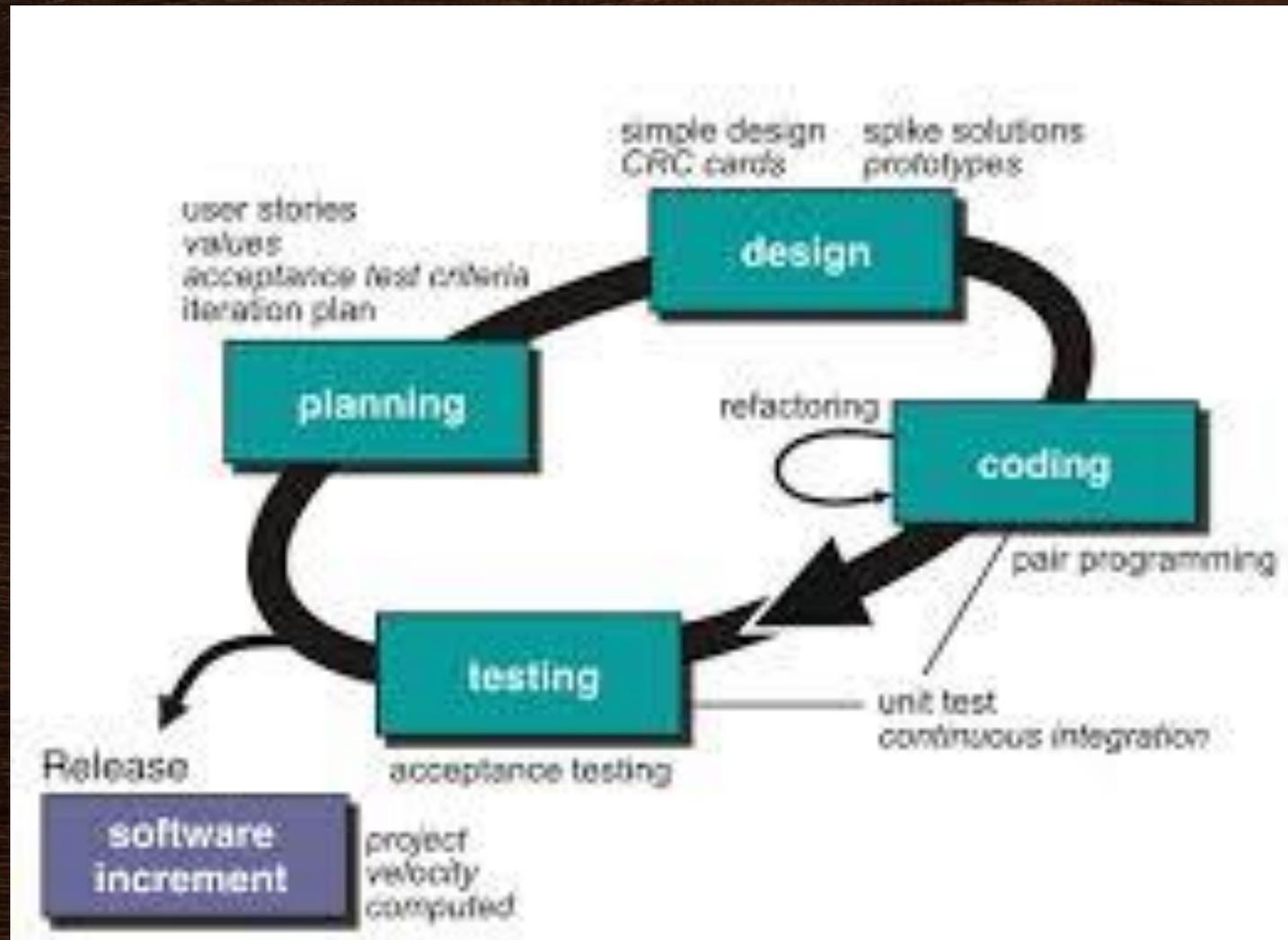
- Selalu evaluasi perkembangan perangkat lunak yang sedang dikerjakan.
- Segala informasi harus dikumpulkan setiap interval waktu yang konsisten dan kesalahan-kesalahan yang muncul selama proses pengembangan harus dibahas dan dicari solusinya.
- Umpan balik tersebut berfungsi sebagai indikator kemajuan proyek dan menginformasikan pemimpin proyek apabila perubahan perlu dibuat.

# Core Value Extreme Programming

## Keberanian (Courage)

- Programmer Extreme Programming (XP) didorong untuk berani bereksperimen dan menulis ulang kode jika mereka tidak puas dengan kode atau desain yang sudah ada.

# Berikut Siklus Hidup dalam XP Programming.



# Tahapan Dalam Extreme (XP) Programming, yaitu:

- **Planning.** Customer dan XP team bekerja bersama untuk memutuskan bagaimana grup story untuk dibangun oleh XP team.
- Jika komitmen telah dibuat, XP team akan membangun story-story dengan cara :
  1. Story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama.
  2. Story dengan resiko paling tinggi akan diimplementasikan terlebih dulu. Setelah project pertama direlease dan didelivery, XP team memperhitungkan kecepatan project. Selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.

# Tahapan Dalam Extreme (XP) Programming, yaitu:

- **Design.** XP menggunakan CRC card, untuk mengenali dan mengatur object oriented class yang sesuai dengan software increment.
- **Coding.** Sebelum membuat code, lebih baik membuat unit test tiap story untuk dimasukkan dalam software increment. XP menyarankan agar dua orang bekerja bersama pada satu komputer workstation untuk membuat code dari satu story (pair programming), untuk menyediakan real time problem solving dan jaminan real time quality. Setelah pair programming selesai, code diintegrasikan dengan kerja lainnya (continuous integration).

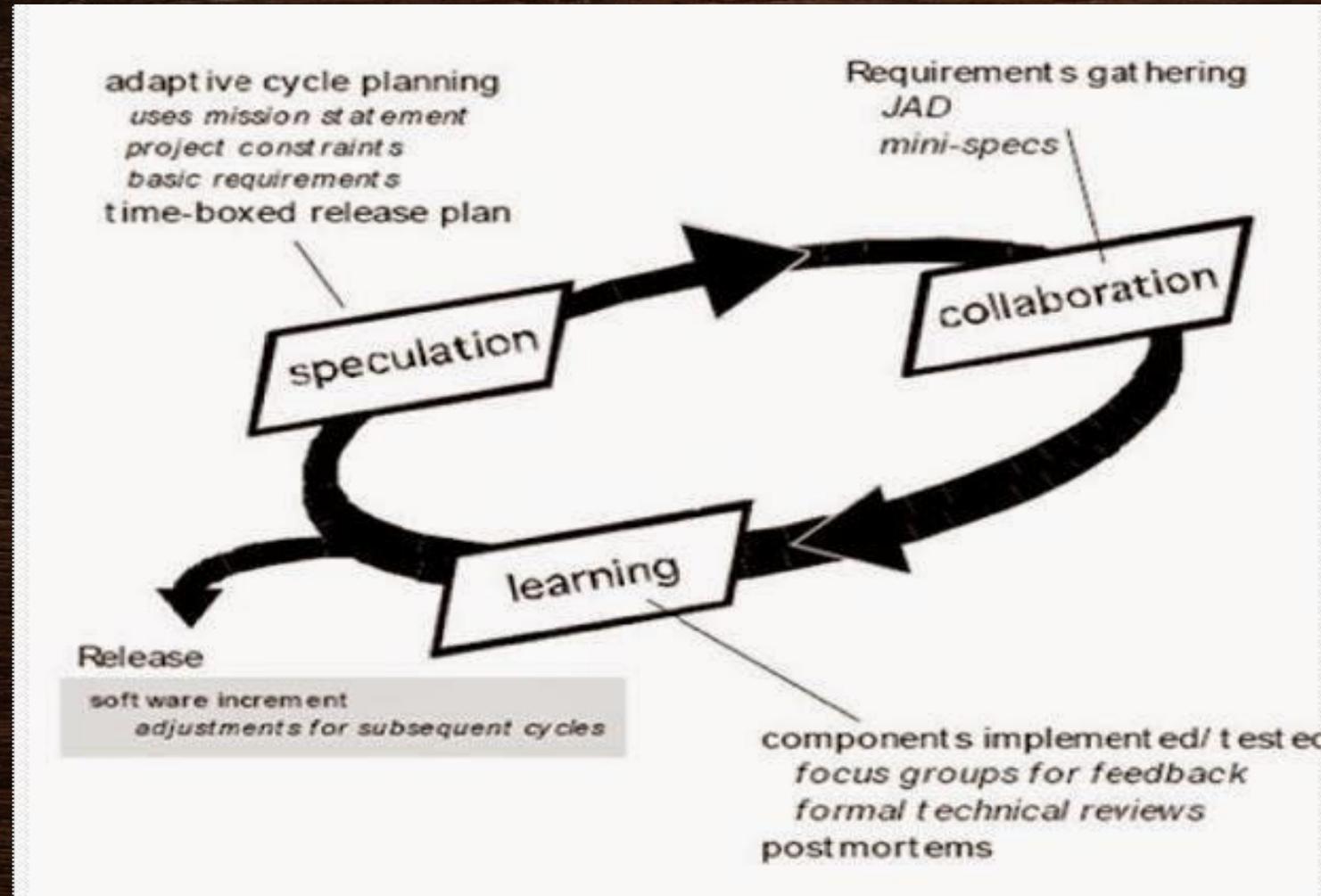
# Tahapan Dalam Extreme (XP) Programming, yaitu:

- **Testing.** Unit test yang telah dibuat harus diimplementasikan menggunakan suatu framework dan diatur ke dalam universal testing suite, integrasi dan validasi sistem dapat dilakukan setiap hari. Customer test (acceptance test) dilakukan oleh customer dan fokus pada keseluruhan fitur dan fungsional sistem. Acceptance test diperoleh dari customer stories yang telah diimplementasikan sebagai bagian dari software release.

# Adaptive Software Development (ASD)

- Diusulkan oleh Jim Highsmith sebagai teknik untuk membangun software dan sistem yang kompleks.
- Filosofi dari ASD adalah kolaborasi manusia dan tim yang mengatur diri sendiri.
- Aktivitas pada proses ASD adalah speculation, collaboration & learning.

# Adaptive Software Development (ASD)



# Tahapan Adaptive Software Development (Pressman, 2005) :

- **SPECULATION** adalah aktivitas adaptive cycle planning yaitu menggunakan informasi awal seperti misi dari klien, batasan proyek dan kebutuhan dasar untuk mendefinisikan rangkaian software increment (produk software yang secara berkala diserahkan).

# Tahapan Adaptive Software Development (Pressman, 2005) :

- **COLLABORATION** adalah aktifitas orang-orang yang bermotivasi tinggi saling bekerja sama : saling melengkapi, rela membantu, kerja keras, terampil dibidangnya dan mengkomunikasikan masalah untuk menghasilkan penyelesaian yang efektif.

# Tahapan Adaptive Software Development (Pressman, 2005) :

## • LEARNING

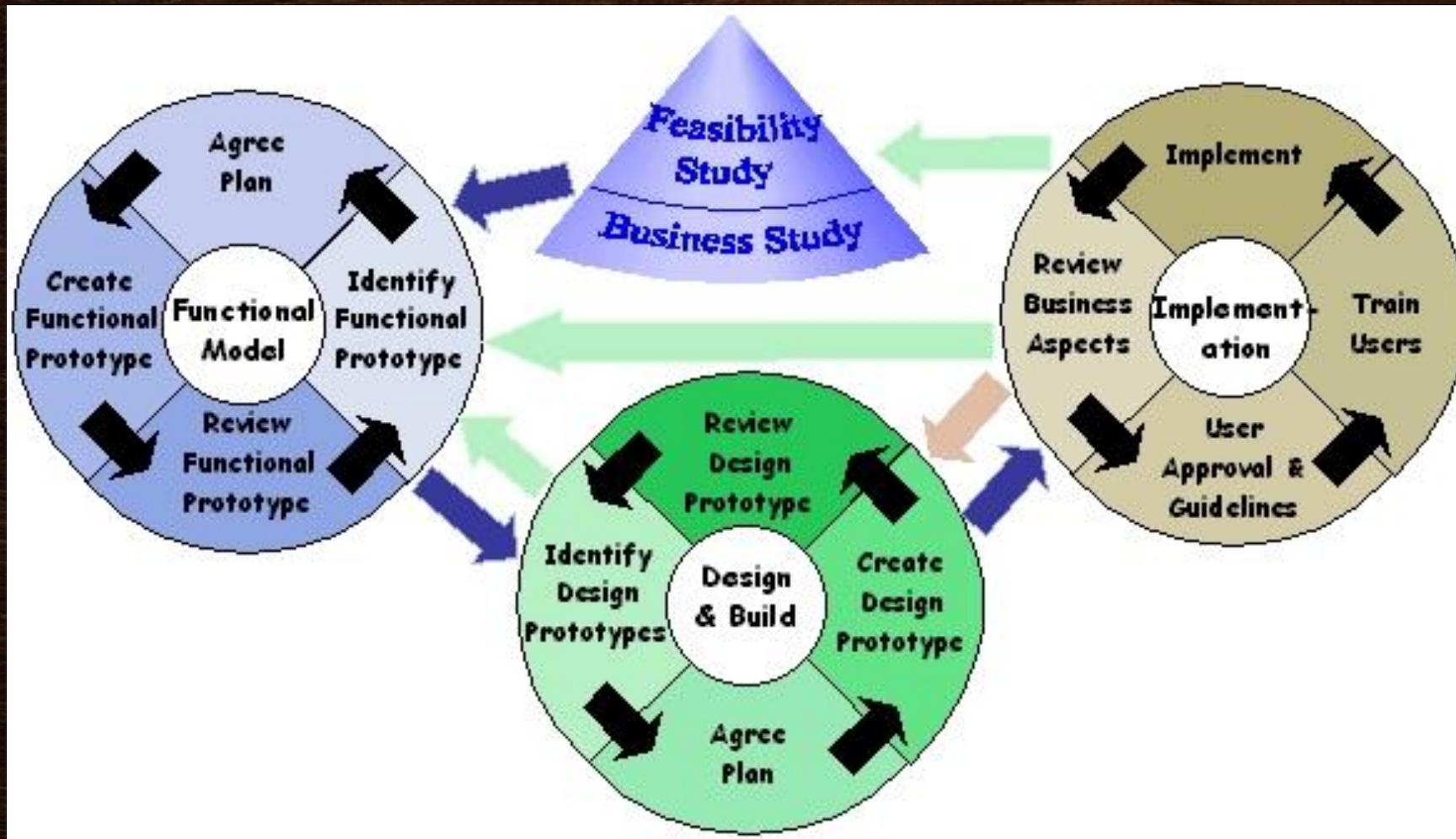
Proses pembelajaran proyek ini bisa dilakukan dengan 3 cara, yaitu sebagai berikut :

1. Focus Group adalah klien dan pengguna memberi masukan terhadap software.
2. Formal Technique Reviews adalah tim ASD secara lengkap melakukan review.
3. Postmortems adalah tim ASD melakukan introspeksi pada kinerja dan proses.

# Dynamic System Development Method

- Menyajikan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan.
- Aktifitas pada Dynamic System development method adalah Feasibility Study, Business Study, Functional Model Iteration, Desain & Build Iteration, Implementation.

# Siklus hidup DSDM



# Beberapa karakteristik DSDM yaitu sebagai berikut :

- Menyajikan kerangka kerja (Framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan.
- Membangun software dengan cepat yaitu 80% dari proyek diserahkan dalam 20% dari waktu total untuk menyerahkan proyek secara utuh.
- Aktifitas Feasibility Study yaitu dengan melihat kebutuhan, lalu diuji apakah sesuai digunakan untuk proses DSDM.
- Aktifitas Business Study yaitu menyusun kebutuhan fungsional dan informasi, menentukan arsitektur aplikasi dan identifikasi kebutuhan pemeliharaan untuk aplikasi.
- Aktifitas Functional model iteration yaitu menghasilkan incremental prototype yang memperlihatkan fungsi software ke client untuk mendapatkan kebutuhan yang lebih jelas dan terkonfirmasi.

# Beberapa karakteristik DSDM yaitu sebagai berikut, (Lanjutan) :

- Aktifitas Design and Build Iteration yaitu melakukan cek ulang prototype yang dibangun untuk memastikan bahwa prototype yang dibangun dengan cara tersebut dipastikan semua fungsi benar-benar bekerja.
- Aktifitas Implementation yaitu menempatkan software pada lingkungan sebenarnya sekalipun belum lengkap atau masih ada perubahan.
- DSDM dapat dikombinasikan dengan XP yang menghasilkan kombinasi model proses mengikuti metode DSDM dan praktek yang sejalan dengan XP.

# Tahapan Dynamic Software Development Method (DSDM) :

- Feasibility study (Studi Kelayakan)
- Business study – prioritized requirements (Studi Bisnis - Persyaratan Diprioritaskan)
- Functional model iteration (Fungsional Model Iterasi)
- Risk analysis (Analisis Resiko)
- Time-box plan (Rencana Waktu)
- Design and build iteration (Desain dan Membangun Iterasi)
- Implementation (Implementasi)

# Kelebihan Dynamic Software Development Method (DSDM) :

- Menyajikan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan.
- Membangun software dengan cepat.
- DSDM dapat dikombinasikan dengan XP.

# Kelemahan Dynamic Software Development Method (DSDM)

- Setiap iterasi bergantung pada prototype sebelumnya.
- Menentukan scope dari suatu prototype proyek tidak pernah selesai.
- Dokumentasi sering kali tidak lengkap pada pembuatan prototype.
- Isu-isu mengenai system backup and recovery, system performance dan system security kurang/tidak diperhatikan dan sering terlupakan.

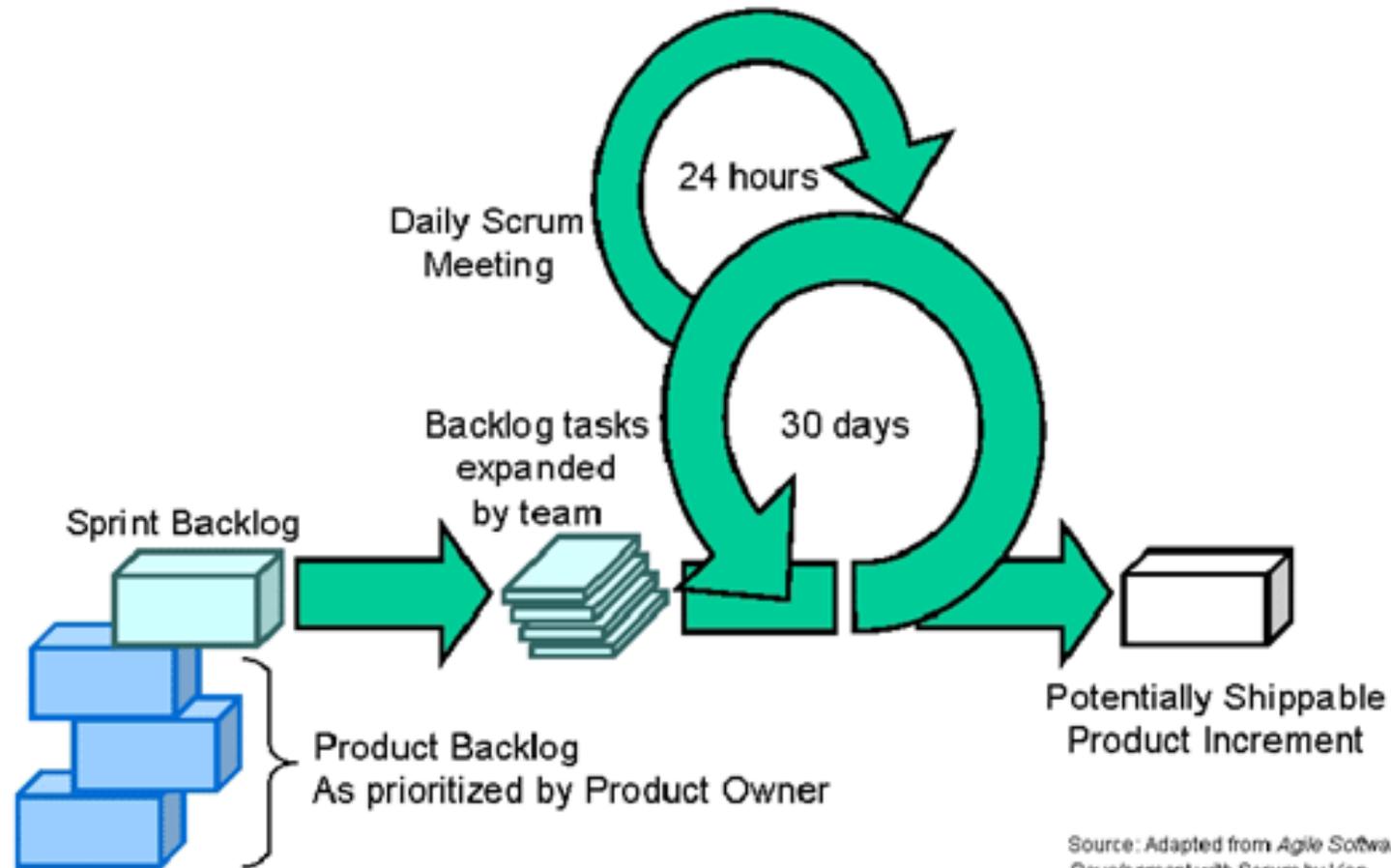
# Waktu Penggunaan Dynamic Software Development Method (DSDM)

- Dynamic Software Development Method (DSDM) digunakan untuk proyek yang membutuhkan waktu singkat dan tidak memerlukan kinerja tinggi, ini karena DSDM berdasarkan pada model RAD.

# SCRUM

- Diperkenalkan oleh Jeff Sutherland awal tahun 1990-an.
- Pengembangan berikutnya dilakukan oleh Schwaber dan Beedle.
- Pada metode SCRUM terdapat aktivitas yang dijalankan sebagai berikut : Backlog, Sprints, Scrum Meetings, Demo.

Skema dari SCRUM adalah sebagai berikut :



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

# Karakteristik SCRUM

- Ukuran tim yang kecil melancarkan komunikasi, mengurangi biaya, dan memberdayakan satu sama lain.
- Proses dapat beradaptasi terhadap perubahan teknis dan bisnis.
- Proses menghasilkan beberapa software increment.
- Pembangunan dan orang yang membangun dibagi dalam tim yang kecil.
- Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun.

# Scrum tepat digunakan saat kondisi :

- Keperluan berubah dengan cepat
- Tim programmer sedikit, yaitu 5-9 orang
- Pelanggan tidak terlalu paham dengan apa yang diinginkan

# Kelebihan Scrum antara lain :

- Keperluan berubah dengan cepat.
- Tim berukuran kecil sehingga melancarkan komunikasi, mengurangi biaya dan memberdayakan satu sama lain.
- Pekerjaan terbagi-bagi sehingga dapat diselesaikan dengan cepat.
- Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun.
- Proses Scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan.

# Kelemahan Scrum antara lain :

- Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.