

Proses Pengembangan Perangkat Lunak Agile Development Methods

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi
Politeknik Negeri Malang

Dosen Pengampu: Wilda Imama Sabilla, S.Kom., M.Kom.

Tujuan

- Untuk menjelaskan Agile Development Methods yang merupakan salah satu metodologi pengembangan perangkat lunak
- Proses dan model-model pengembangan perangkat lunak Agile Development Methods

Agile Development Methods

- Agile Model merupakan proses pengembangan software yang berkembang pada tahun 1990.
- Metodologi yang dikenal sebagai agile development methods ini mengutamakan fleksibilitas terhadap perubahan-perubahan yang terjadi selama pengembangan.
- Model-model dari agile diantaranya Rational Unified Process (1994), Scrum (1995), Crystal, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995).
- Pada akhirnya terbentuklah pada tahun 2001 proses pengembangan Agile Modeling (AM).

Proses Pengembangan Agile Modeling (AM)

- Dilakukan secara iterasi atau perulangan.
- Jika suatu proyek pengembangan software dikerjakan dengan menggunakan metode agile, maka selama waktu pengerjaannya akan selalu dijumpai proses pengembangan yang dilakukan berulang.

12 Prinsip-prinsip Agile Development Methods



1. Prioritas utama adalah **memuaskan klien** dengan menghasilkan **perangkat lunak yang bernilai** secara **cepat dan rutin**.
2. **Siap terhadap perubahan kebutuhan**. Proses Agile memanfaatkan perubahan untuk keuntungan klien.
3. Menghasilkan perangkat lunak yang bekerja secara rutin, dari jangka waktu beberapa minggu sampai beberapa bulan, dengan **mengutamakan jangka waktu yang pendek**.
4. **Rekan bisnis dan pengembang** perangkat lunak harus **bekerja sama** sepanjang proyek.
5. **Lingkungan pengembang proyek memiliki suasana yang motivatif**. Berikan mereka lingkungan dan dukungan yang dibutuhkan, dan percayai mereka untuk dapat menyelesaikan pekerjaan dengan baik
6. Metode yang paling efisien dan efektif untuk **bertukar informasi** dari dan dalam tim pengembang adalah dengan **komunikasi secara langsung**.

12 Prinsip-prinsip Agile Development Methods



7. **Perangkat lunak yang bekerja** adalah **ukuran utama kemajuan** suatu tim.
8. Proses Agile mendukung **pengembangan yang berkelanjutan** dengan **kecepatan pengembangan yang konsisten**
9. Perhatian terhadap **detail-detail teknis** dan **desain** akan meningkatkan **agility**
10. **Kesederhanaan** (memaksimalkan jumlah pekerjaan yang belum dilakukan) adalah hal yang sangat penting
11. **Self-organizing team** mendukung arsitektur, kebutuhan, dan rancangan perangkat lunak yang baik
12. Secara berkala, **tim pengembang berefleksi** tentang bagaimana agar pengembangan lebih efektif, kemudian menyesuaikan cara bekerja mereka.

Manfaat Agile development methods

- High-value & working App system
- Iterative, incremental, evolutionary
- Cost control & value-driven development
- High-quality production
- Flexible & risk management
- Collaboration
- Self-organizing, self-managing teams

High-value & working App system

- Dapat dihasilkan perangkat lunak yang mempunyai nilai jual yang tinggi
- Biaya pembuatan bisa ditekan
- Perangkat lunak bisa berjalan dengan baik

Iterative, incremental, evolutionary

- Tim harus bekerja dalam waktu yang singkat (biasanya 1-3 minggu).
- Selalu menambah fungsionalitas dari perangkat lunak sesuai dengan kebutuhan klien.

Cost control & value-driven development

- Pengembangan perangkat lunak disesuaikan dengan kebutuhan pengguna.
- Tim bisa dengan cepat merespon kebutuhan yang diinginkan pengguna sehingga waktu dan biaya pembuatan perangkat lunak bisa dikontrol.

High-quality production

- Biaya pembuatan perangkat lunak bisa ditekan dan proses pembuatan bisa dipercepat , tetapi kualitas dari perangkat lunak yang dibuat harus tetap dijaga
- Dengan melakukan tes setiap fungsionalitas perangkat lunak setelah selesai dibuat berarti agile juga mengakomodir kebutuhan ini

Flexible & Risk Management

- Apabila menggunakan metode pembuatan yang biasanya dipakai, ketika ingin mengubah fungsionalitas dari wireframe yang telah dibuat biasanya dibutuhkan proses yang rumit.
- Aktivitas dilakukan dengan fleksibel agar kegagalan perangkat lunak pun bisa diminimalisir, antara lain:
 - pertemuan dengan sistem analis untuk mengubah sistem perangkat lunak
 - perubahan rencana rilis produk
 - perubahan biaya produksi sehingga fungsionalitas perangkat lunak mudah diubah.

Collaboration

- Tim pengembang diharuskan sering bertemu untuk membahas:
 - perkembangan proyek
 - feedback dari klien
 - tambahan pada perangkat lunak
- Pertemuan dilakukan agar tim bisa berkolaborasi dengan maksimal

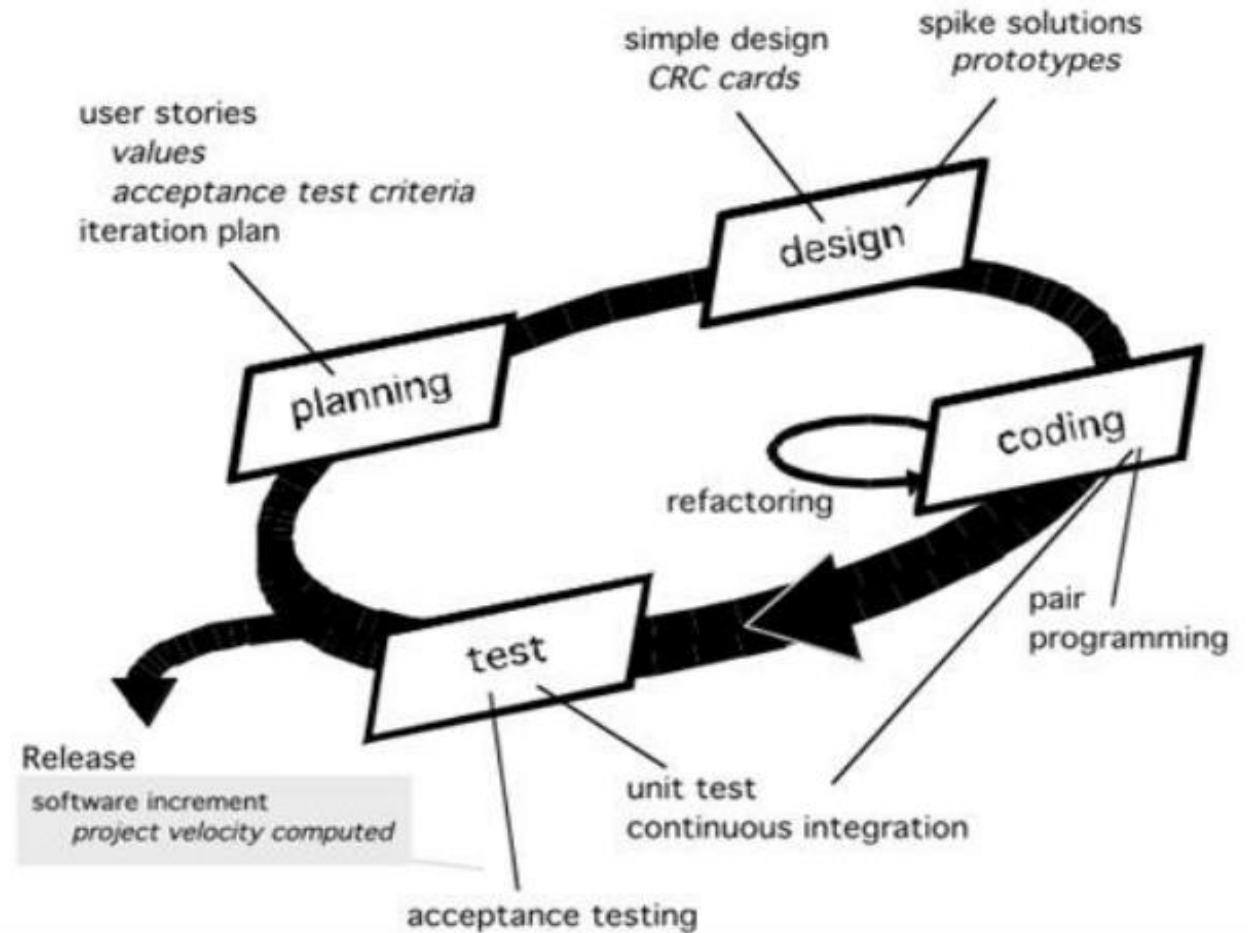
Self-organizing, self-managing teams

- Rekrut orang terbaik, beri dan dukung kebutuhan mereka lalu biarkan mereka bekerja.
- Itulah perbedaan agile dan SDM lainnya.
- Dengan agile, developer dapat manajemen dirinya sendiri, sedangkan manajer tim hanya bertugas mengkolaborasikan developer perangkat lunak dengan klien.
- Sehingga terciptalah tim yang solid

Model Agile XP (eXtreme Programming)



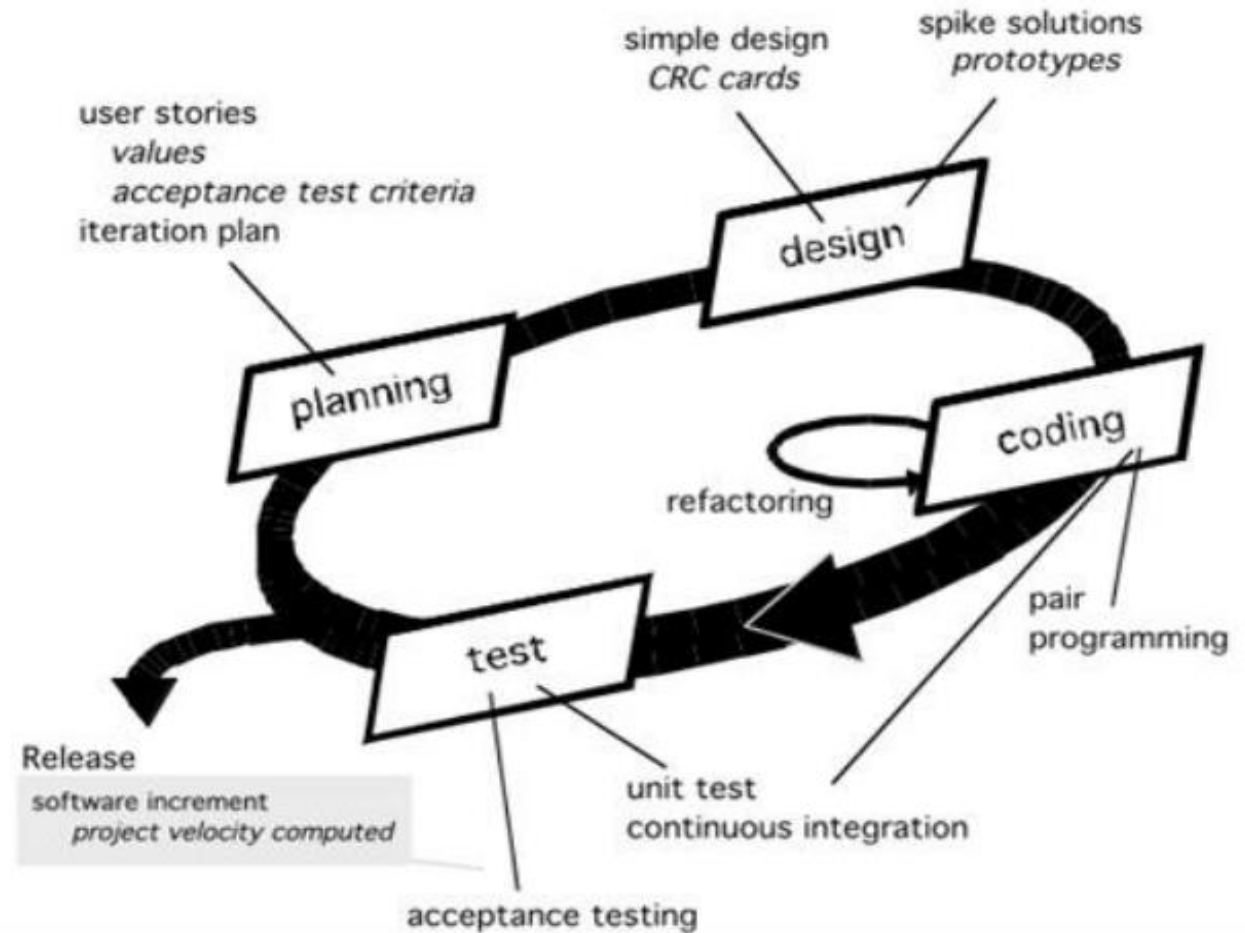
- Model ini tidak hanya melakukan proses code dan testing saja, tetapi juga melakukan proses pemetaan arsitektur sistem agar proses coding dan testing bisa dilaksanakan tanpa mengganggu sistem yang ada sebelumnya.



Model Agile XP (eXtreme Programming)



- Metodologi ini berfokus pada peningkatan kualitas perangkat lunak dan kemampuan tim untuk merespons kebutuhan yang berubah dengan cepat.
- XP menekankan kolaborasi yang kuat antara tim pengembang dan pelanggan, serta siklus pengembangan yang cepat dan berulang.

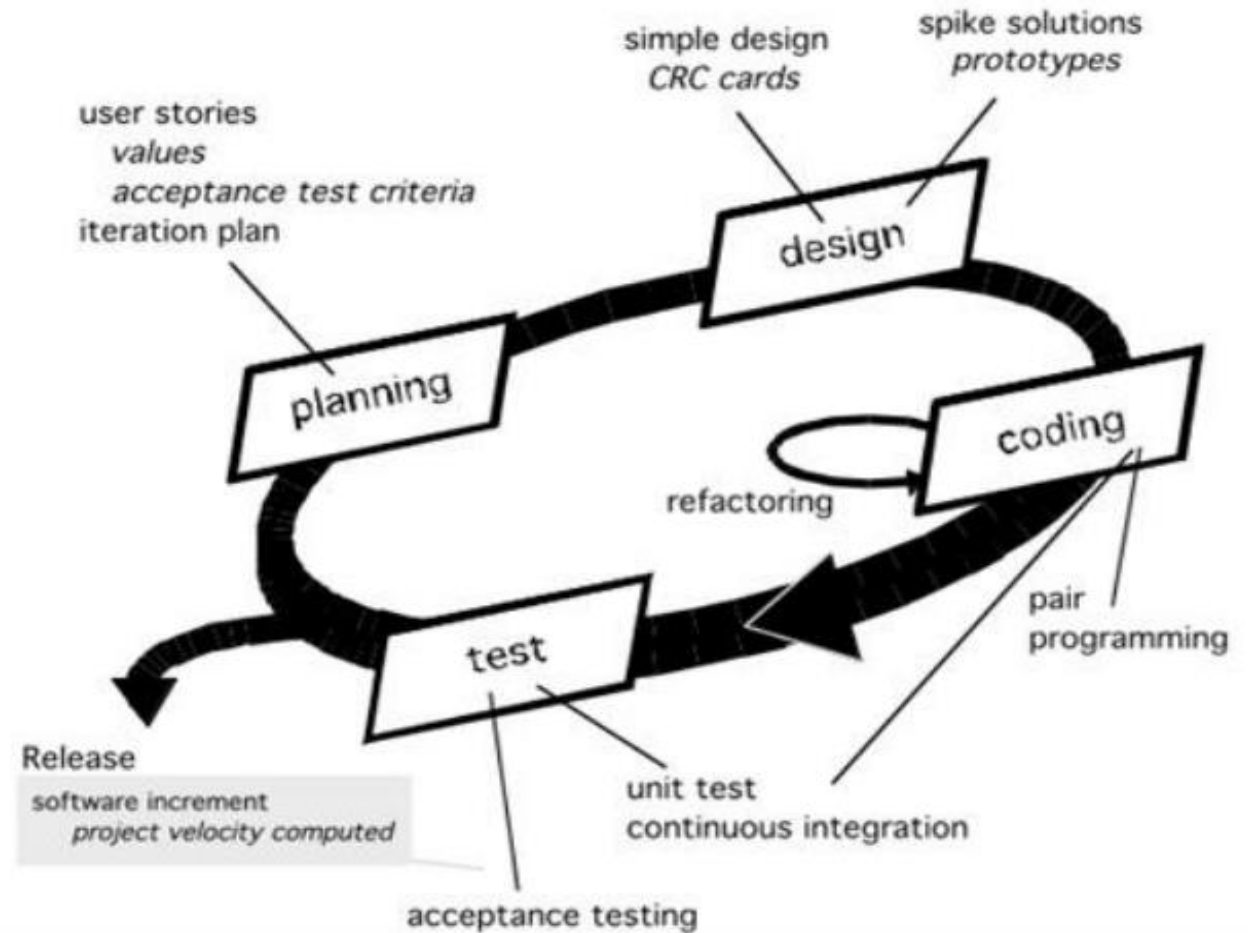


Tahapan - Model Extreme Programming

1. Planing

Perencanaan adalah proses metodis yang dirancang untuk mencapai tujuan tertentu dan pengambilan keputusan untuk mencapai hasil yang diinginkan. Kebutuhan yang dibutuhkan pada tahap ini teknik pengumpulan data:

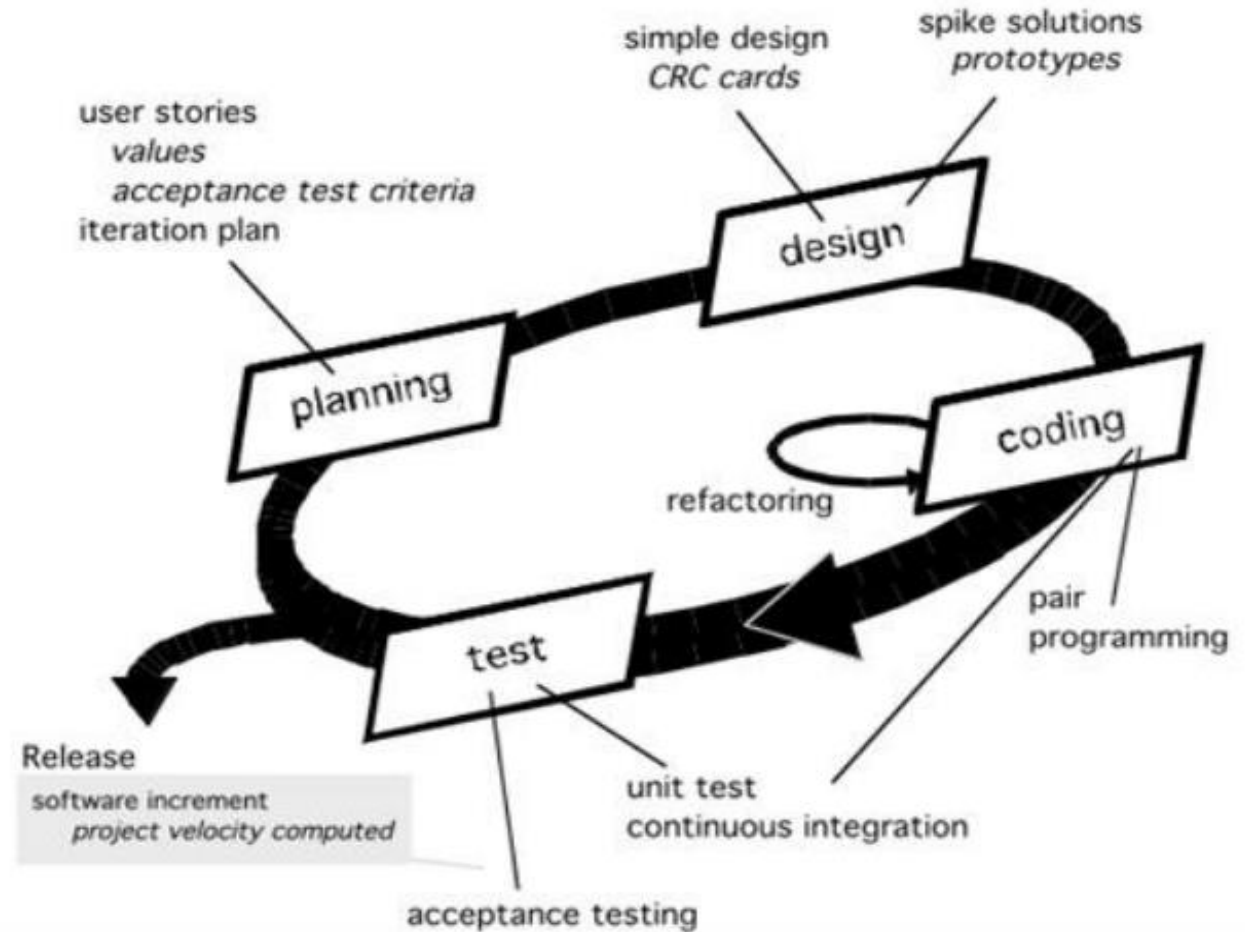
- Analisis kebutuhan sistem
- Identifikasi actor
- Identifikasi use case



Tahapan - Model Extreme Programming

2.Design

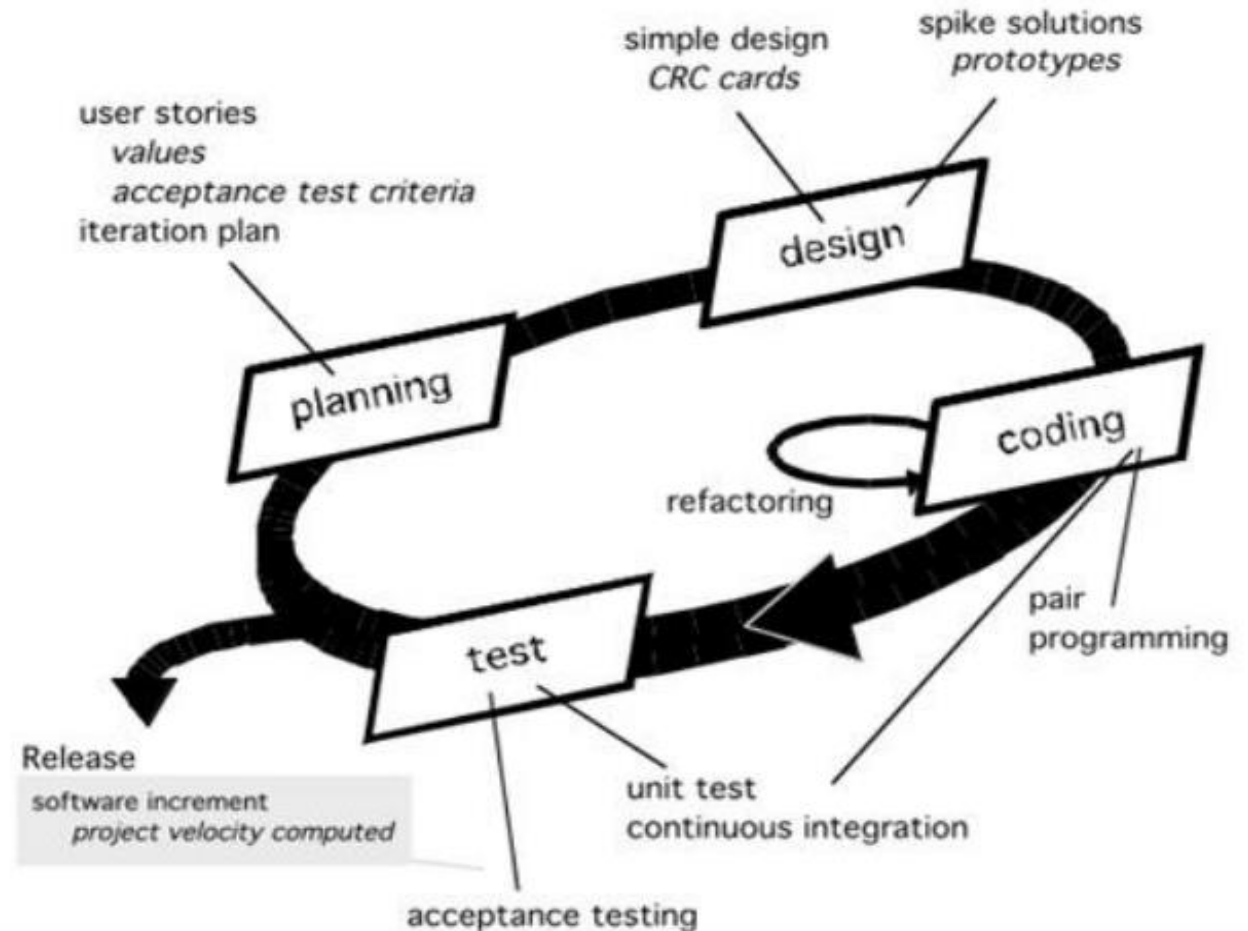
- Aktivitas design dalam pengembangan aplikasi bertujuan untuk mengatur pola logika dalam sistem.
- Sebuah design yang baik, dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Dengan begitu, jika salah satu fitur pada sistem mengalami kerusakan, tidak akan mempengaruhi sistem secara keseluruhan
- XP mengutamakan desain yang sederhana dan hanya menambahkan fitur jika memang benar-benar diperlukan agar perangkat lunak tetap ramping dan mudah dipelihara.



Tahapan - Model Extreme Programming

3.Coding

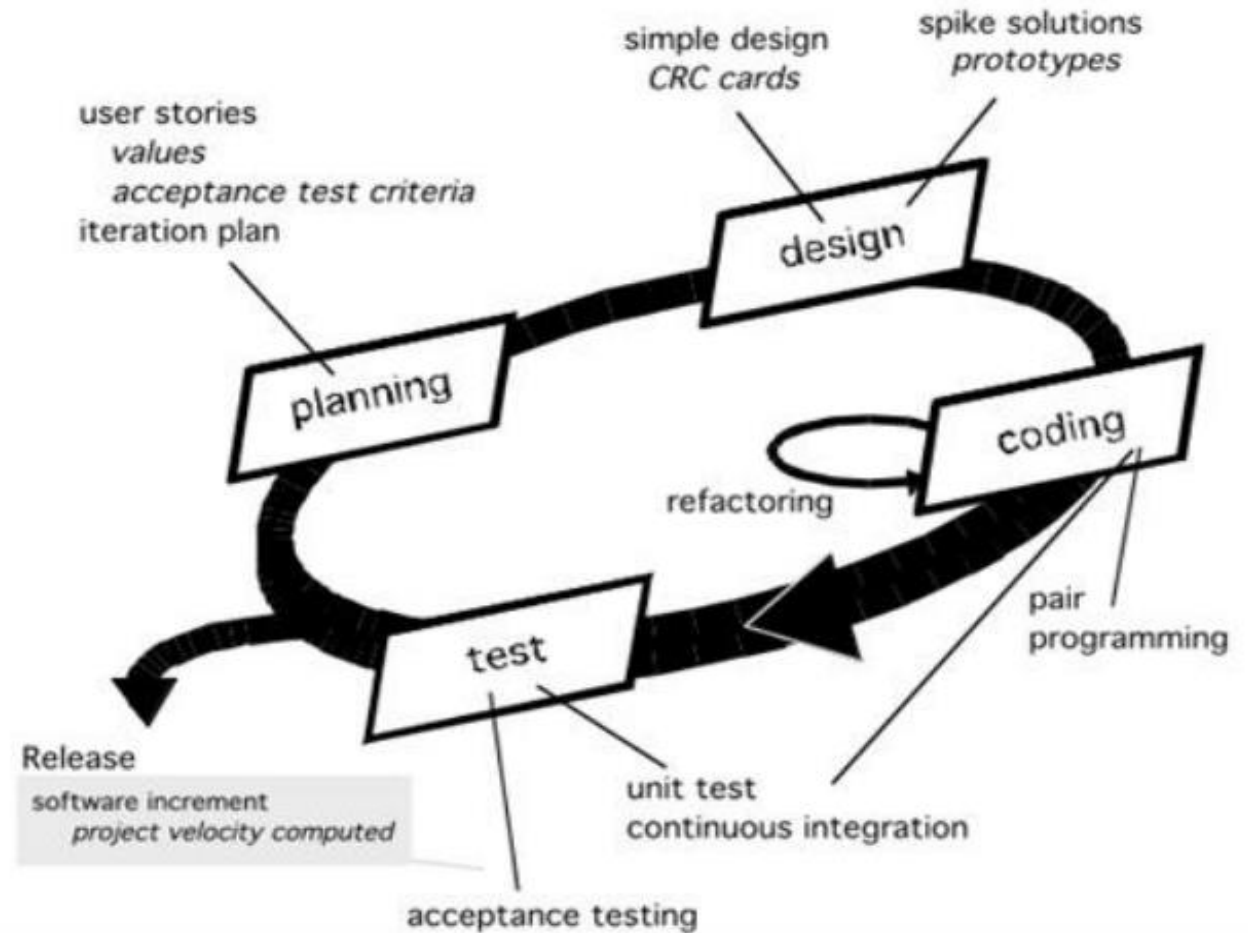
- Setelah berbagai unit test selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan coding aplikasi.
- XP menerapkan konsep pair programming dimana setiap tugas sebuah modul dikembangkan oleh 2 orang programmer.
- Satu orang menulis kode, sementara yang lain meninjau kode tersebut. Ini meningkatkan kualitas kode dan memastikan lebih banyak ide masuk ke dalam solusi.



Tahapan - Model Extreme Programming

3.Coding

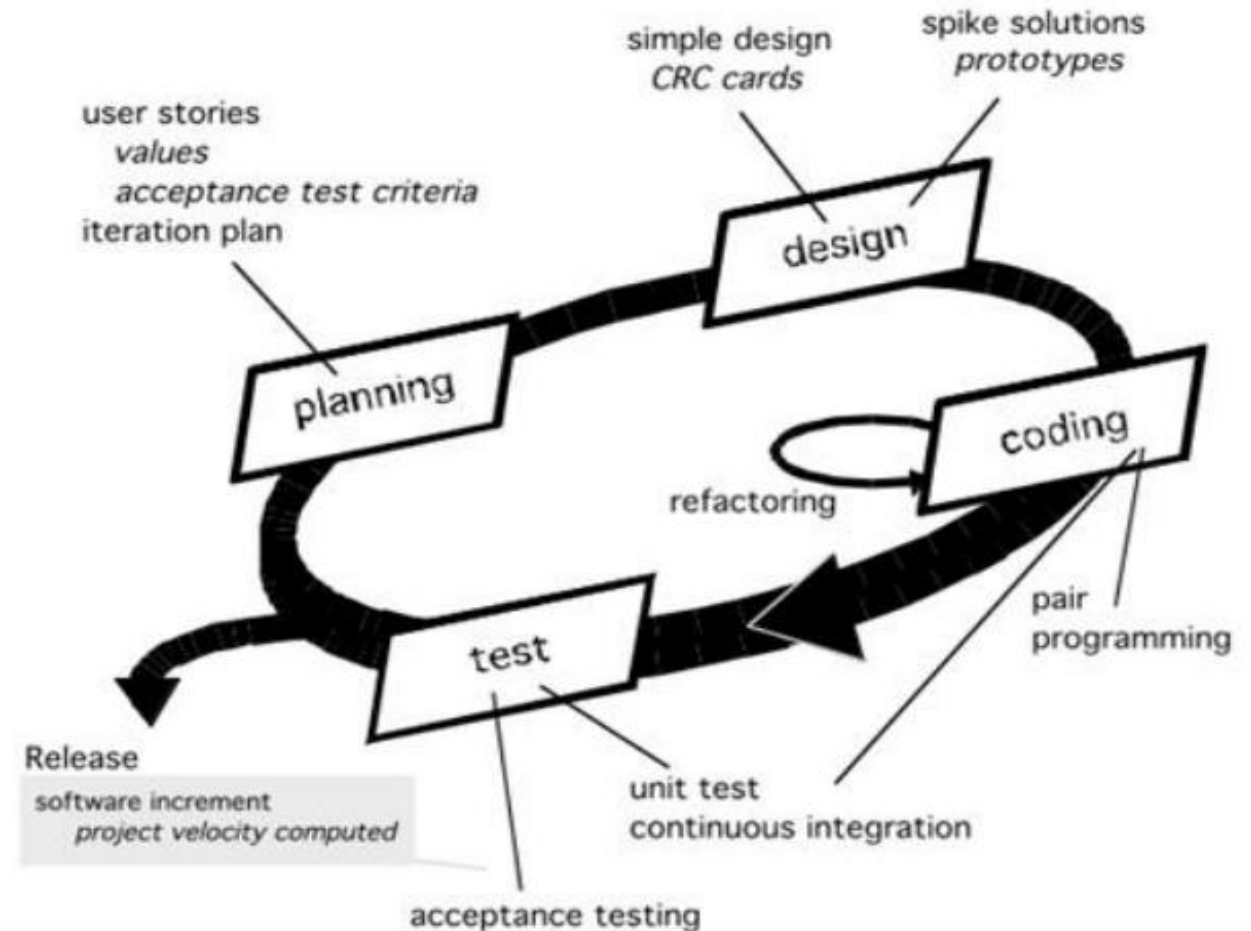
- **Refactoring:** XP mendorong pengembang untuk terus-menerus memperbaiki struktur kode tanpa mengubah fungsionalitasnya.
- Bertujuan menjaga kode tetap bersih dan mudah dimodifikasi.



Tahapan - Model Extreme Programming

4. Testing

- Pada tahapan pengujian ini aplikasi langsung diuji coba oleh pengguna atau klien
- Mendapat tanggapan langsung mengenai penerapan yang telah digambarkan sebelumnya



Kelebihan XP

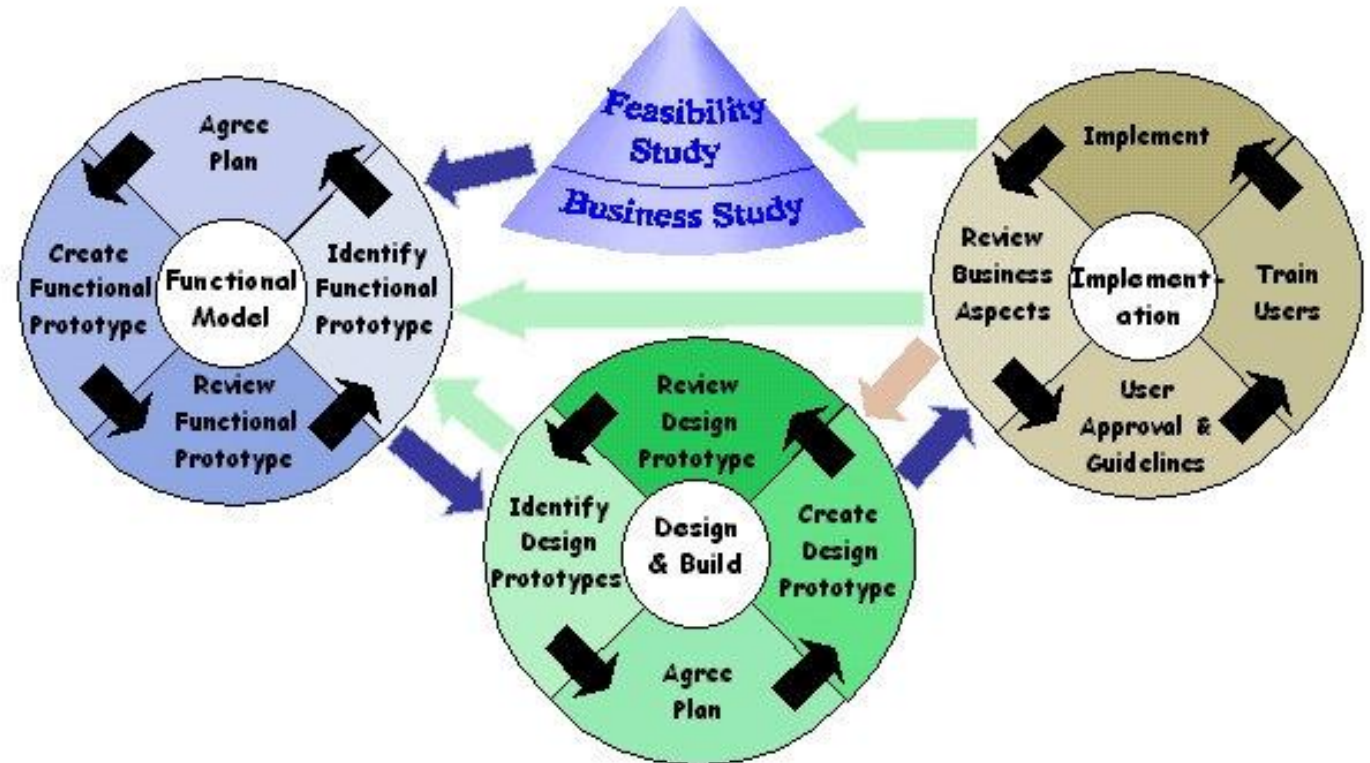
- Umpan balik yang segera setelah mendapat respon dari user.
- Collective ownership yang tidak menggantungkan pekerjaan pada satu orang saja
- Perubahan dan penambahan requirements dapat direspon meskipun proses pengembangan sudah hampir selesai
- Proses pengembangan yang menyertakan satu orang dari pihak user menjadi on-site customer memudahkan komunikasi selama proses pengembangan
- Metode yang populer karena lebih santai dan non-restriktif
- Biaya lebih murah
- Mampu mengotomatiskan tes
- Setiap feedback ditanggapi dengan melakukan tes, unit tes atau system integration dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
- Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

Kekurangan XP

- 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.
- Komunikasi yang selalu bersifat oral tanpa dokumentasi formal.

Model Agile DSDM (Dynamic Systems Development Method)

- Dynamic System Development Method menyediakan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan



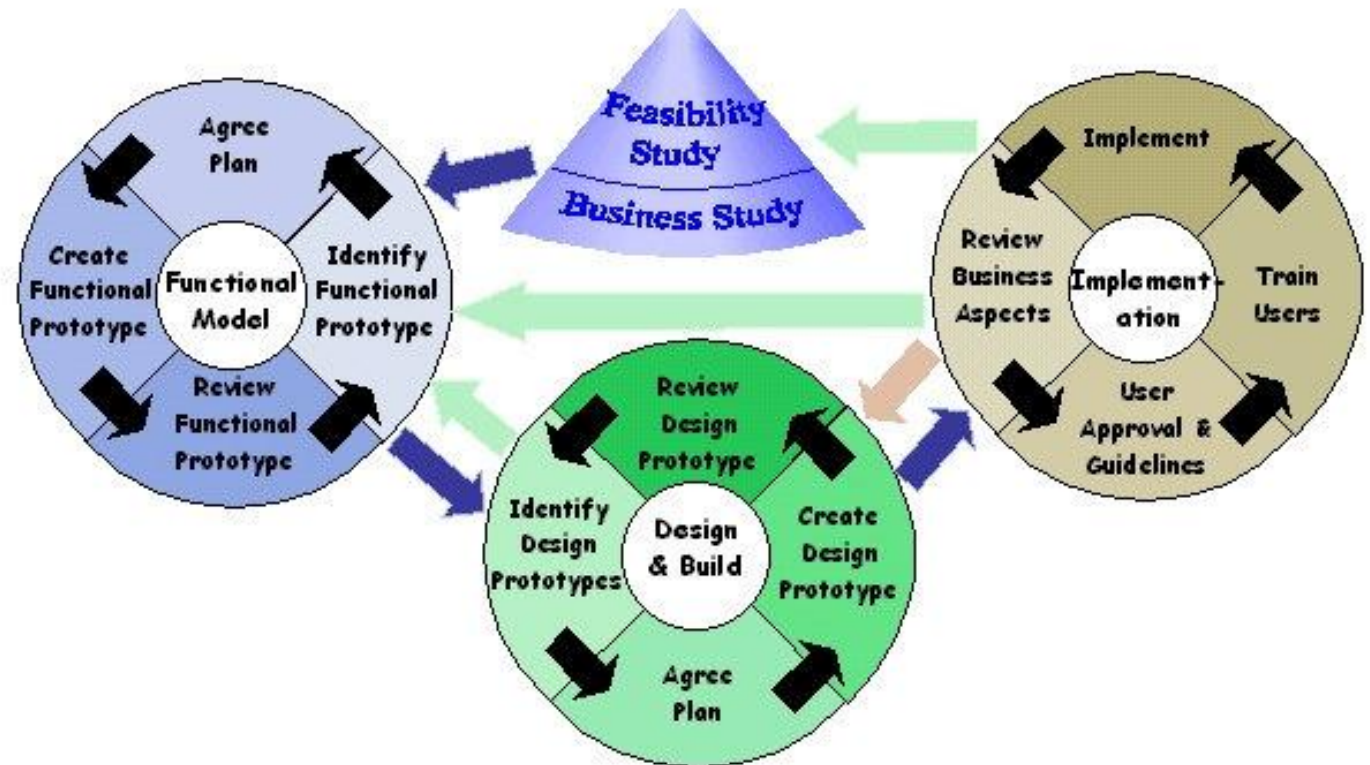
Model Agile DSDM (Dynamic Systems Development Method)

- Prinsip Prioritas: DSDM memiliki prinsip yang disebut MoSCoW untuk memprioritaskan fitur atau pekerjaan:
 - Must have (harus ada) – Fitur yang sangat penting.
 - Should have (sebaiknya ada) – Fitur penting, tapi tidak mendesak.
 - Could have (bisa ada) – Fitur tambahan yang berguna.
 - Won't have (tidak perlu ada sekarang) – Fitur yang tidak akan dikerjakan untuk saat ini.
- Dengan menggunakan MoSCoW, DSDM memastikan bahwa proyek tetap fokus pada fitur yang paling penting dan dapat disesuaikan dengan keterbatasan waktu serta sumber daya.

Model Agile DSDM (Dynamic Systems Development Method)

1. Feasibility Study

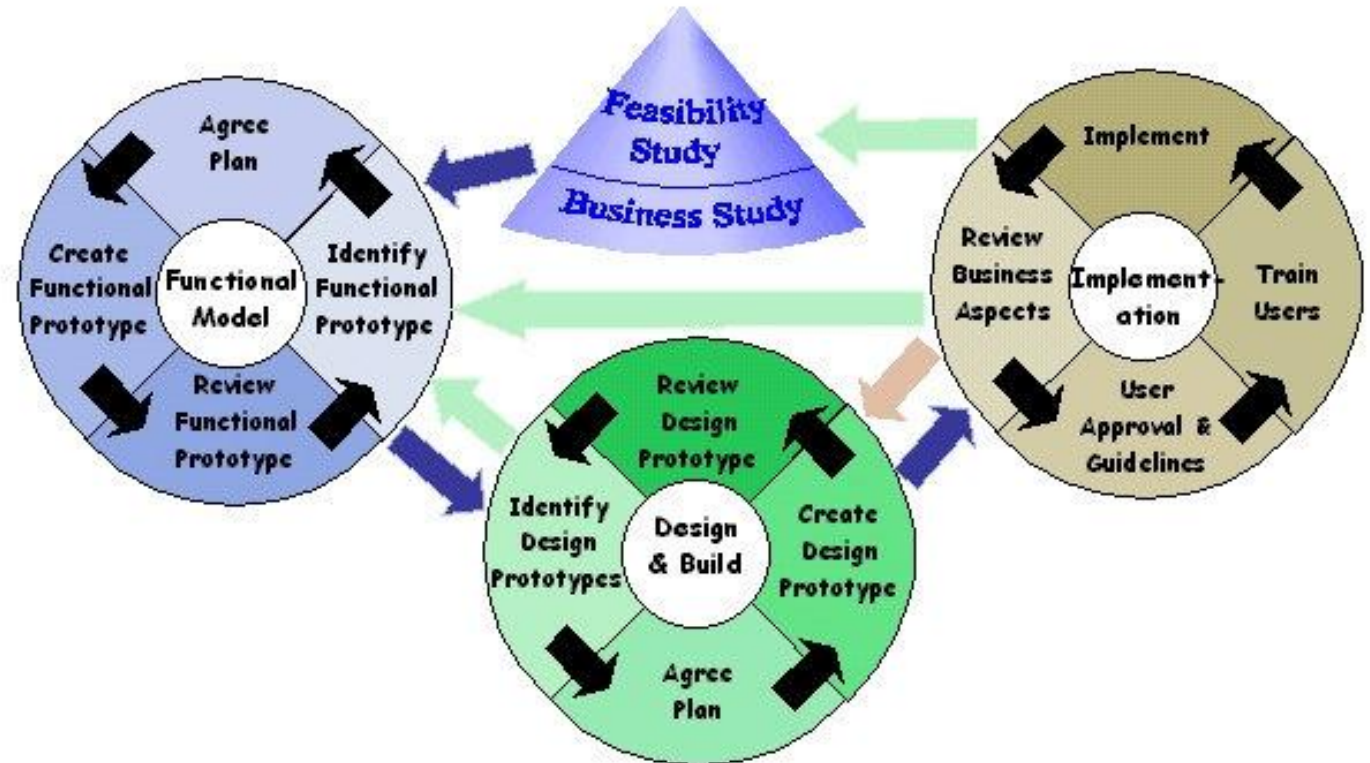
- Adalah laporan kelayakan tingkat tinggi yang memungkinkan komite pengarah proyek untuk memutuskan masa depan proyek, dan studi kelayakan lebih lanjut



Model Agile DSDM (Dynamic Systems Development Method)

2. Business Study

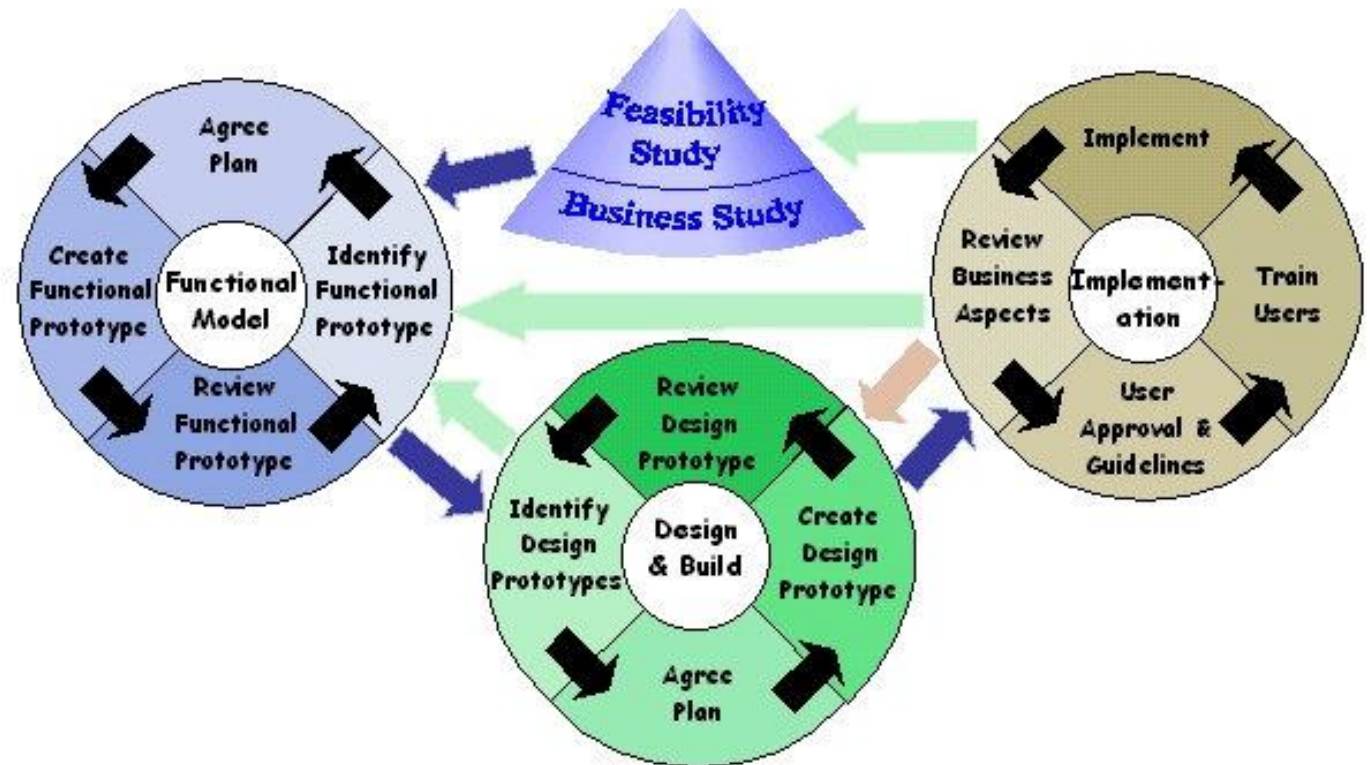
- Studi Bisnis memberikan dasar untuk semua pengembangan produk berikutnya
- Fase ini mengarah pada proses bisnis yang terkena dampak secara rinci dan informasi-informasi yang mereka butuhkan



Model Agile DSDM (Dynamic Systems Development Method)

3. Functional Model

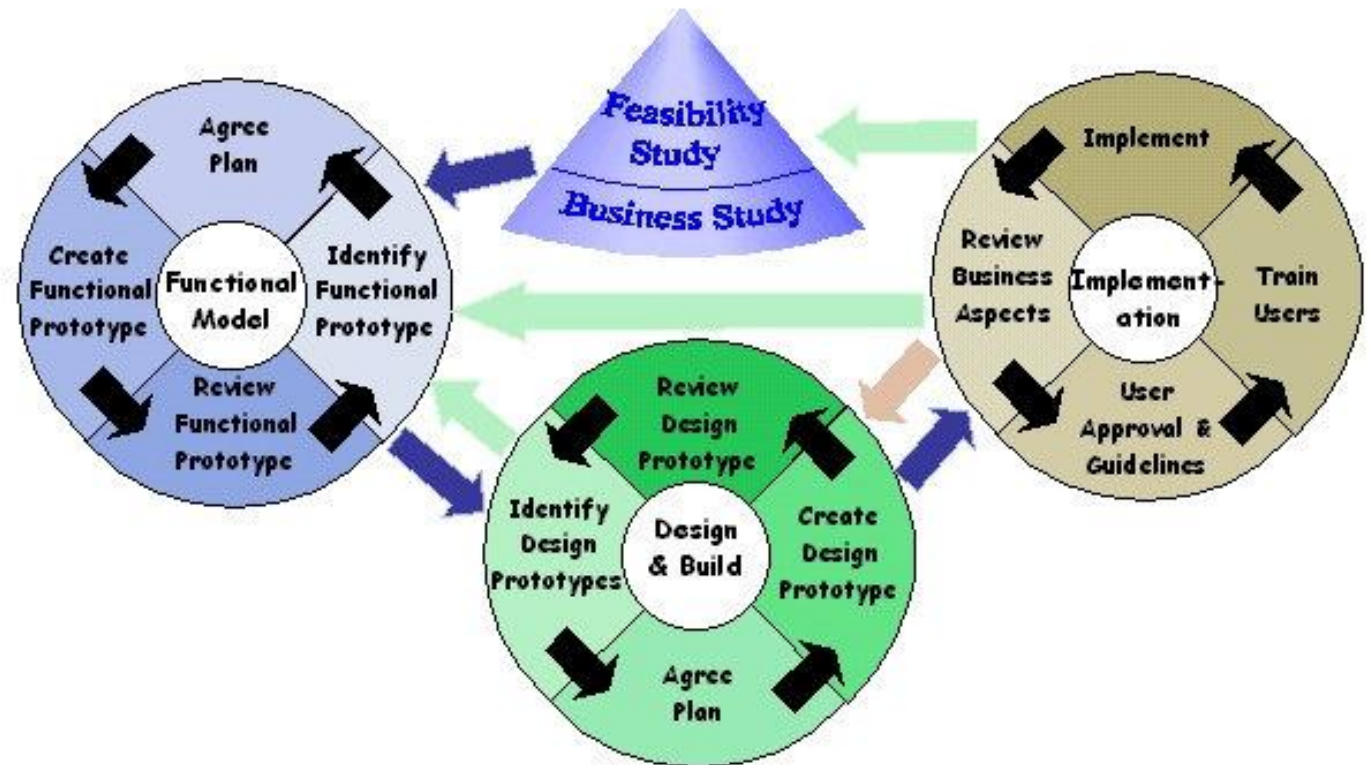
- Model fungsional yang terdiri dari prototipe perangkat lunak yang bekerja dan model statis.
- Fase ini menghasilkan pengolahan informasi yang diperoleh dalam business study.



Model Agile DSDM (Dynamic Systems Development Method)

4. Design and Build Iteration

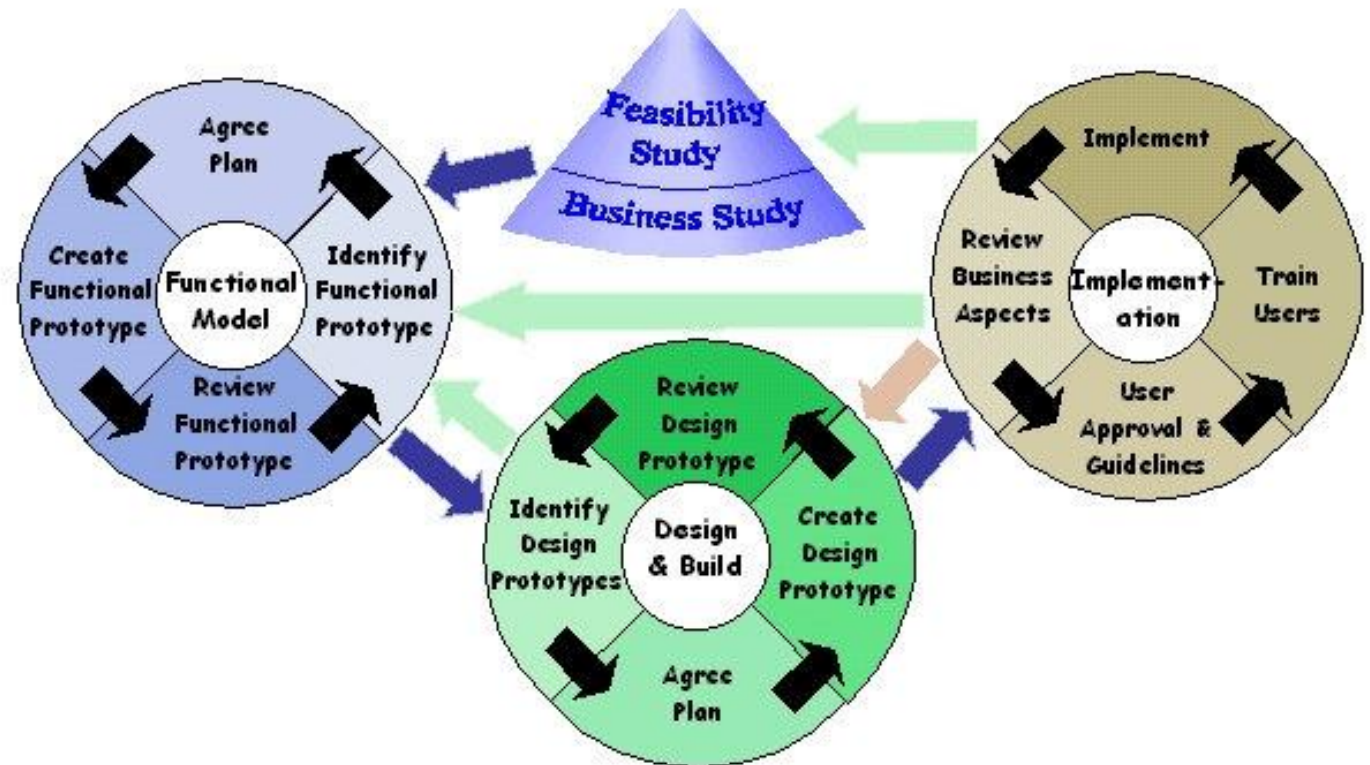
- Fase ini menyempurnakan prototype fungsional yang dikembangkan untuk memenuhi kebutuhan fungsional (Modul) yaitu: Mengidentifikasi, Merencanakan, mengembangkan, validasi fungsi



Model Agile DSDM (Dynamic Systems Development Method)

4. Implementation Phase

- Fase ini meliputi transisi dari lingkungan pengembangan untuk lingkungan operasional.
- Tujuan utama dari tahap ini adalah untuk menempatkan sistem yang diuji ke dalam lingkungan pengguna dan melatih individu – individu untuk menggunakannya.



Kelebihan **DSDM** (Dynamic Systems Development Method)



- Fleksibel terhadap perubahan – Dapat menyesuaikan dengan kebutuhan bisnis yang berubah.
- Pengendalian waktu dan biaya – Menggunakan pendekatan dengan waktu dan biaya tetap.
- Kolaborasi erat dengan bisnis – Pemangku kepentingan terlibat aktif dalam pengembangan.
- Iterasi dan umpan balik cepat – Umpan balik reguler membantu mengurangi risiko kegagalan.
- Pengujian berkelanjutan – Pengujian dilakukan sepanjang proyek untuk menjaga kualitas.
- Manajemen risiko yang baik – Prioritas fitur penting dan fleksibilitas terhadap fitur kurang kritis.

Kekurangan **DSDM** (Dynamic Systems Development Method)



- Butuh komitmen tinggi dari pemangku kepentingan – Keterlibatan aktif sangat penting.
- Tidak cocok untuk proyek dengan persyaratan kaku – Kurang ideal jika perubahan tidak memungkinkan.
- Penyusunan prioritas bisa sulit – Kesulitan dalam menetapkan prioritas yang tepat.

Kesimpulan



- Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak. Sehingga saat membuat perangkat lunak dengan menggunakan agile development methods diperlukan inovasi dan responsibility yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang
- Agile Software Development:
 - Berinteraksi dengan 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
 - Sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana

Agile Development Methods dikembangkan karena pada metodologi tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai tuntutan user.

Metodologi ini sudah cukup banyak berkembang, di antaranya adalah Extreme Programming (XP), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum Methodology, Crystal, Feature Driven Development (FDD), Agile Modeling (AM)

Any questions?