

# **Bahan Ajar Rekayasa Perangkat Lunak**

## **Agile Software Development**

Disiapkan oleh Umi Proboyekti

### **Pengantar**

Kata **Agile** berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep Agile Software Development dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa Agile Software Development adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus.

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. Namun demikian, sama seperti model proses yang lain, Agile Software Development memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. Agile Software Development memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi. Namun di sisi lain menyebabkan produktifitas menurun.

### **Prinsip Agile Software Development**

Salah satu ciri dari Agile Software Development adalah tim yang tanggap terhadap perubahan. Mengapa? Karena perubahan adalah hal yang utama dalam pembangunan software: perubahan kebutuhan software, perubahan anggota tim, perubahan teknologi dll. Selain itu Agile Software Development juga melihat pentingnya komunikasi antara anggota tim, antara orang-orang teknis dan businessmen, antara developer dan managernya. Ciri lain adalah klien menjadi bagian dari tim pembangun software. Ciri-ciri ini didukung oleh 12 prinsip yang ditetapkan oleh Agile Alliance. Menurut Agile Alliance, 12 prinsip ini adalah bagi mereka yang ingin berhasil dalam penerapan Agile Software Development:

1. kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus
2. menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan.
4. Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung
5. Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam

lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.

6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien
7. Software yang berfungsi adalah ukuran utama dari kemajuan proyek
8. dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan
9. perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile
10. kesederhanaan penting
11. arsitektur, kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri
12. secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

Kedua belas prinsip tersebut menjadi suatu dasar bagi model-model proses yang punya sifat agile.

Dengan prinsip-prinsip tersebut Agile Process Model berusaha untuk meniasati 3 asumsi penting tentang proyek software pada umumnya:

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

### **Faktor Manusia pada Agile Process Model**

Kunci faktor manusia pada model ini adalah proses didasari pada kebutuhan orang dan tim bukan sebaliknya, Untuk dapat sukses menerapkan model proses ini, pada faktor manusia ada beberapa kunci penting:

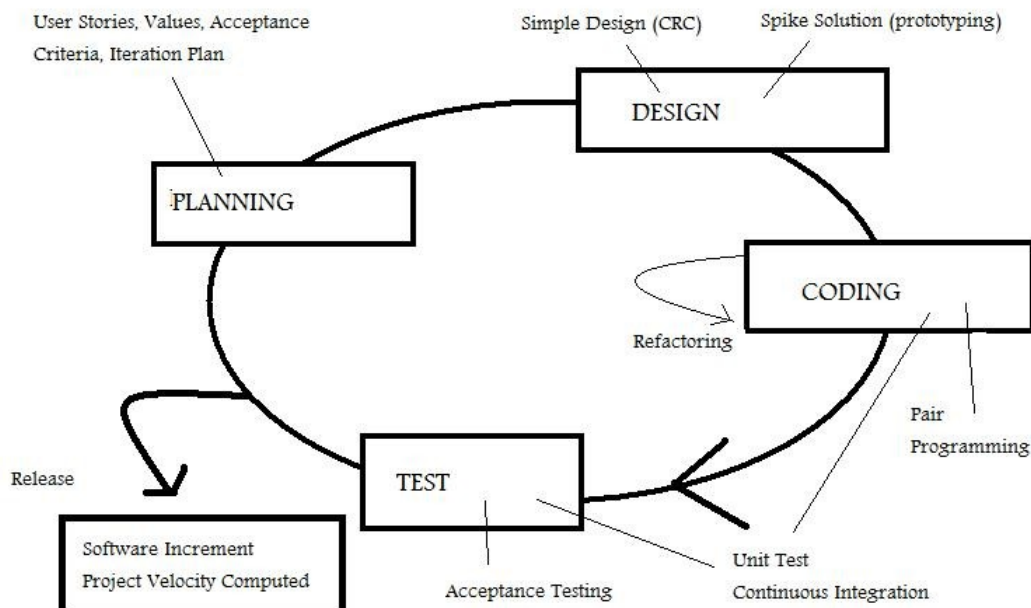
1. kompetensi: ketrampilan dalam membangun dan pengetahuan tentang proses membangun
2. Fokus: memiliki fokus yang sama sekalipun peran dalam tim berbeda
3. Kolaborasi : kerja sama dengan klien, anggota tim dan manajer.
4. Kemampuan ambil keputusan : tim pembangun memiliki otonomi dalam pengambilan keputusan terkait teknis dan proyek
5. Kemampuan fuzzy problem-solving: mampu menyelesaikan memilah masalah yang penting untuk dipecahkan segera atau nanti.
6. Saling percaya dan hormat: kekompakan tim yang didukung oleh rasa percaya dan saling menghargai satu sama lain.
7. Manajemen diri: tim mengatur diri untuk selesaikan proyek, mengatur proses untuk disesuaikan dengan lingkungannya, tim menjadwal dirinya untuk menyerahkan hasil.

## Agile Process Model

Berikut ini adalah model-model proses yang termasuk agile process model :

### 1. XP(Extreme Programming)

- a) dipublikasikan oleh Kent Beck tahun 1999
- b) Menggunakan pendekatan berorientasi objek
- c) Aktifitas Perencanaan : pengumpulan user stories dari klien yang klien tetapkan prioritasnya. Setiap story ditetapkan harga dan lama pembangunan, jika terlalu besar, story dapat dipecah menjadi beberapa story yang lebih kecil. Periksa dan pertimbangkan resiko
- d) Aktifitas Desain: berprinsip: sederhana.Memanfaatkan kartu CRC (Class-Responsibility-Collaborator) untuk identifikasi dan mengatur class-class di konsep OO. Jika temui kesulitan, prototype dibangun [ini namanya spike solution]. Lakukan refactoring, yaitu mengembangkan desain dari program setelah ditulis
- e) Aktifitas Pengkodean: siapkan unit test sebelum pengkodean dipakai sebagai fokus pemrogram untuk membuat program. Pair programming dilakukan untuk real time program solving dan real time quality assurance
- f) Aktifitas Pengujian: menggunakan unit test yang dipersiapkan sebelum pengkodean.

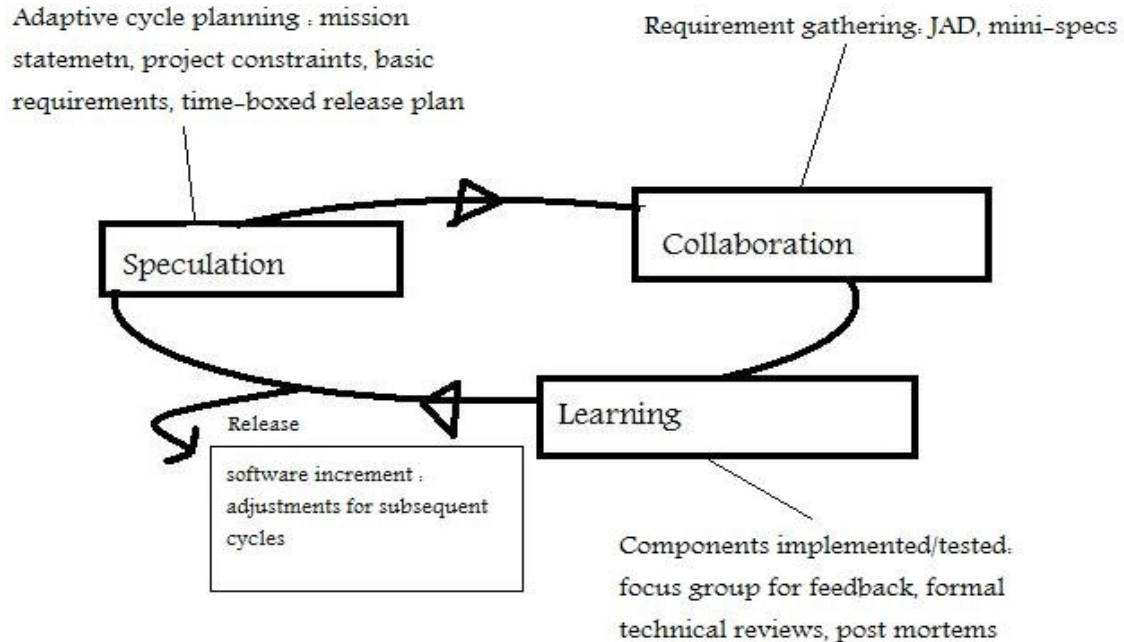


### 2. ADAPTIVE SOFTWARE DEVELOPMENT (ASD)

- a) Diajukan oleh Jim Highsmith sebagai teknik untuk membangun software dan sistem yang

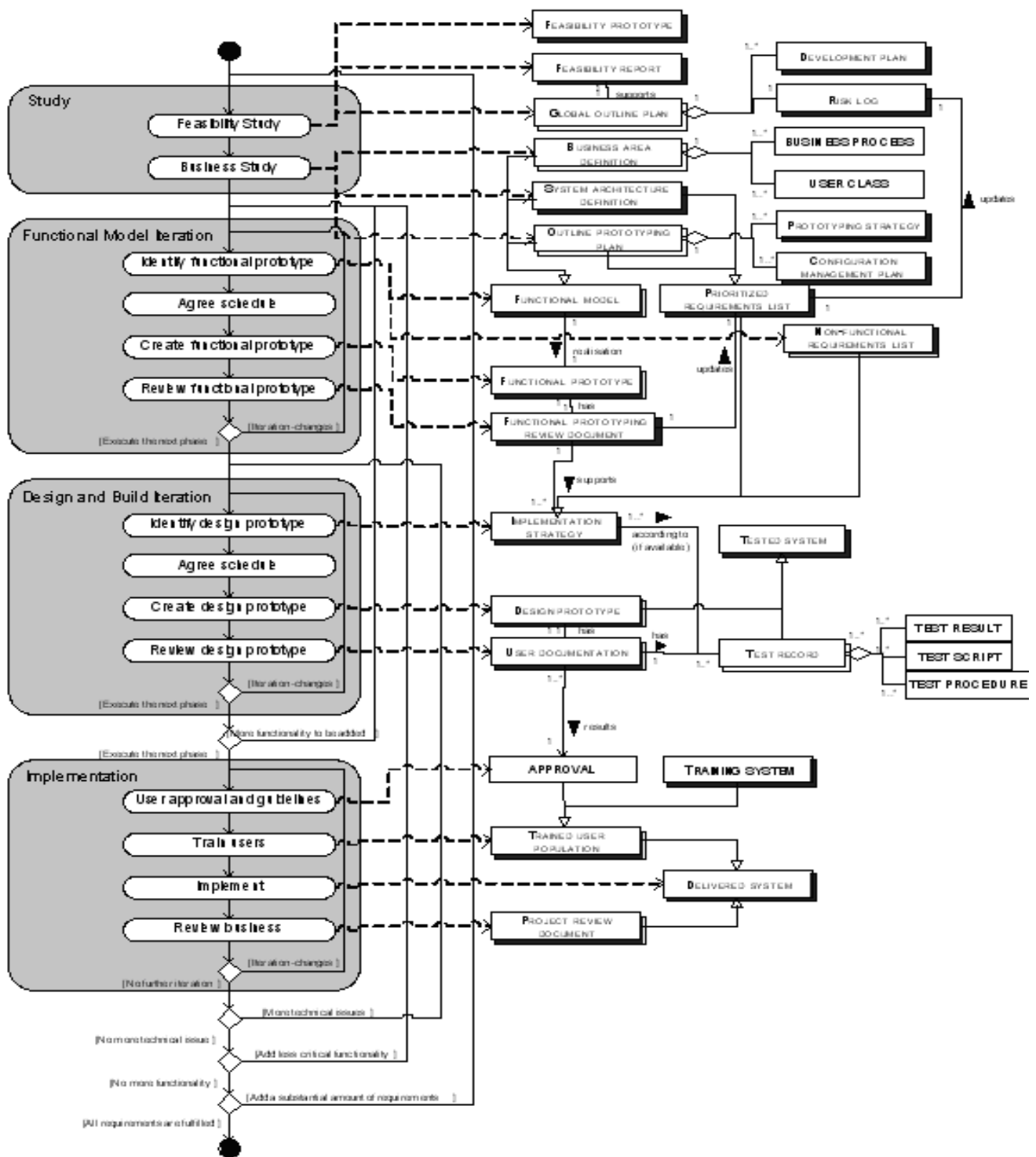
kompleks.

- b) Filosofi yang mendasari adalah kolaborasi manusia dan tim yang mengatur diri sendiri.
- c) Aktifitas: Speculation, Collaboration dan Learning
- d) Aktifitas Speculation : 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)
- e) Aktifitas Collaboration : orang-orang yang bermotivasi tinggi bekerja sama: saling melengkapi, rela membantu, kerja keras, trampil di bidangnya, dan berkomunikasi masalah untuk hasilkan penyelesaian yang efektif.
- f) Aktifitas Learning: tim pembangun sering merasa sudah tahu semua hal tentang proyek, padahal tidak selamanya begitu. Karena itu proses ini membuat mereka belajar lebih tentang proyek melalui 3 cara:
  - Focus group: klien dan pengguna memberi masukan terhadap software
  - Formal Technique Reviews: Tim ASD lengkap melakukan review
  - Postmortems: Tim ASD lakukan introspeksi pada kinerja dan proses.



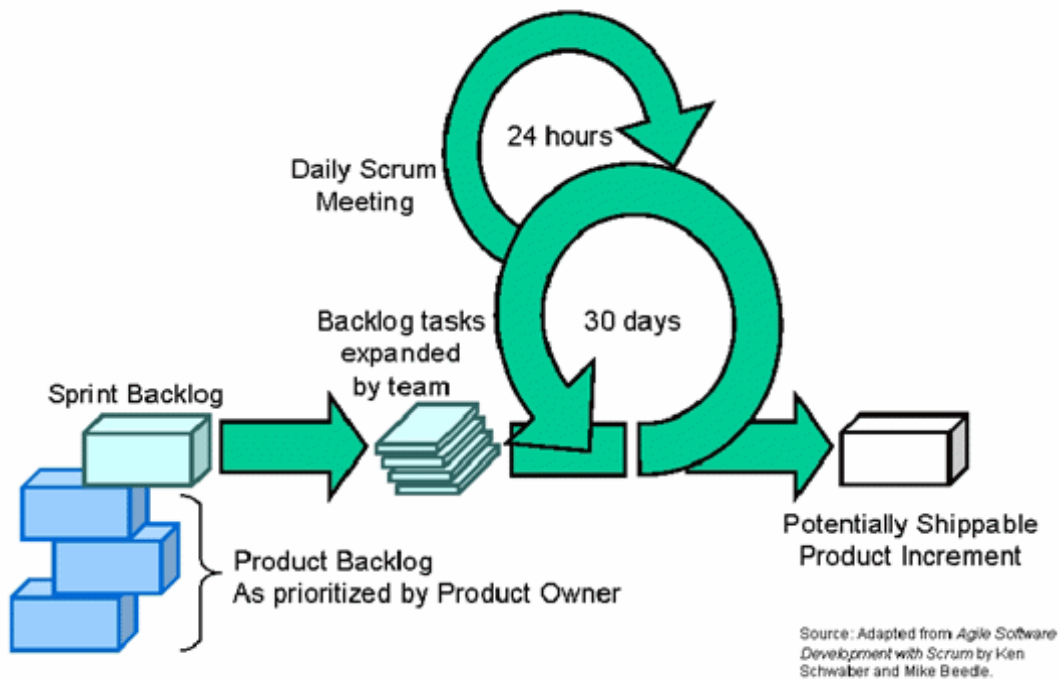
### 3. DYNAMIC SYSTEMS DEVELOPMENT METHOD

- a) menyajikan kerangka kerja (framework) untuk membangun dan memelihara sistem dalam waktu yang terbatas melalui penggunaan prototyping yang incremental dalam lingkungan yang terkondisikan.
- b) Membangun software dengan cepat: 80% dari proyek diserahkan dalam 20% dari waktu total untuk menyerahkan proyek secara utuh.
- c) Aktifitas: feasibility study : siapkan requirement, dan batasan, lalu uji apakah sesuai gunakan proses DSDM
- d) Aktifitas: Business Study: susun kebutuhan fungsional dan informasi, tentukan arsitektur aplikasi dan identifikasi kebutuhan pemeliharaan untuk aplikasi
- e) Aktifitas: Functional model iteration : hasilkan incremental prototype yang perlihatkan fungsi software ke klien untuk dapatkan kebutuhan lebih jelas dan konfirmasi
- f) Aktifitas: Design and Build Iteration : cek ulang prototype yang dibangun untuk pastikan bahwa prototype dibangun dengan cara yang memungkinkan fungsi tersebut benar-benar bekerja
- g) Aktifitas: Implementation: menempatkan software pada lingkungan sebenar sekalipun belum lengkap, atau masih ada perubahan.
- h) DSDM dapat dikombinasikan dengan XP menghasilkan kombinasi model proses yang mengikuti DSDM dan praktek yang sejalan dengan XP.



#### 4. SCRUM

- a) Diperkenalkan oleh Jeff Sutherland tahun awal tahun 1990an
- b) Pengembangan berikutnya dilakukan oleh Schwaber dan Beedle
- c) Scrum memiliki prinsip:
  - 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
  - proses scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan
- d) Aktifitas Scrum : Backlog, Sprints, Scrum Meetings, Demo



- e) Aktifitas Backlog : Backlog adalah daftar kebutuhan yang jadi prioritas klien. Daftar dapat bertambah.
- f) Aktifitas Sprints: unit pekerjaan yang diperlukan untuk memenuhi kebutuhan yang ditetapkan dalam backlog sesuai dengan waktu yang ditetapkan dalam time-box (biasanya 30hari). Selama proses ini berlangsung backlog tidak ada penambahan.

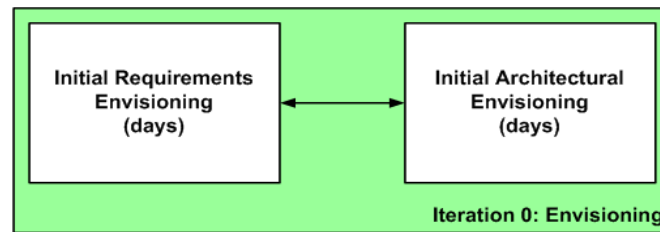
- g) Aktifitas Scrum Meeting: pertemuan 15 menit perhari untuk evaluasi apa yang dikerjakan, hambatan yang ada, dan target penyelesaian untuk bahan meeting selanjutnya.
- h) Aktifitas Demo: penyerahan software increment ke klien didemonstrasikan dan dievaluasi oleh klien.

## 5. AGILE MODELING

- a) banyak situasi pembangun software harus membangun sistem bisnis yang besar dan penting. Jangkauan dan kompleksitas sistem harus dimodelkan sehingga dapat dimengerti, masalah dapat dibagi menjadi lebih kecil dan kualitas dapat dijaga pada tiap langkah pembangunan software.
- b) AM adalah suatu metodologi yang praktis untuk dokumentasi dan pemodelan sistem software.
- c) AM adalah kumpulan nilai-nilai, prinsip dan praktek-praktek untuk memodelkan software agar dapat diaplikasikan pada software development proyek secara efektif.
- d) Prinsip dalam AM;
  - membuat model dengan tujuan: tentukan tujuan sebelum membuat model
  - menggunakan multiple models: tiap model mewakili aspek yang berbeda dari model lain.
  - travel light: simpan model-model yang bersifat jangka panjang saja
  - isi lebih penting dari pada penampilan: modeling menyajikan informasi kepada audiens yang tepat.
  - memahami model dan alat yang yang digunakan untuk membuat software
  - adaptasi secara lokal



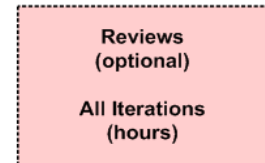
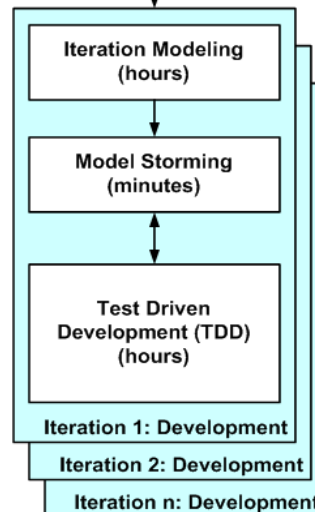
- Identify the high-level scope
- Identify initial "requirements stack"
- Identify an architectural vision



- Modeling is part of iteration planning effort
- Need to model enough to give good estimates
- Need to plan the work for the iteration

- Work through specific issues on a JIT manner
- Stakeholders actively participate
- Requirements evolve throughout project
- Model just enough for now, you can always come back later

- Develop working software via a test-first approach
- Details captured in the form of executable specifications



Copyright 2003-2007  
Scott W. Ambler

## Penutup

Dari model-model proses di atas dapat diambil beberapa poin penting:

1. komunikasi mempunyai peran penting dalam pembanguna software
2. kebutuhan software tidak mudah untuk diidentifikasi secara lengkap
3. kerja sama dalam tim menentukan kelancaran pembangunan software

Aktifitas yang terjadi di dalam Agile Model Process tetap mengandung aktifitas-aktifitas yang ada pada model proses generasi sebelumnya seperti waterfall, incremental, spiral dan RAD. Selain itu, model-model proses di atas tetaplah bukan model proses yang cocok untuk setiap jenis software.

Referensi:

Ambler, Scott W. "An Introduction to Agile Modeling." **Agile Modeling Home Page**. 2001-2008.

Diakses di <http://www.agilemodeling.com/essays/introductionToAM.htm>. 07/09/08.

Pressman, Roger S. **Software Engineering : A Practitioner's Approach**. 6<sup>th</sup> Ed. McGraw-Hall. NY. 2005.