

LECTURE NOTES

Software Engineering

Minggu 2

Software Process Model

LEARNING OUTCOMES

Setelah menyelesaikan pembelajaran ini, mahasiswa akan mampu:

- ☒ LO1 – Menjelaskan konsep dari proses model piranti lunak

Outline Materi (Sub-Topic) :

1. *Software Process Structure*
2. *Process Models*
3. *Agile Development*
4. *Human Aspects of Software Engineering*
5. Studi Kasus

ISI MATERI

1. *Software Process Structure*

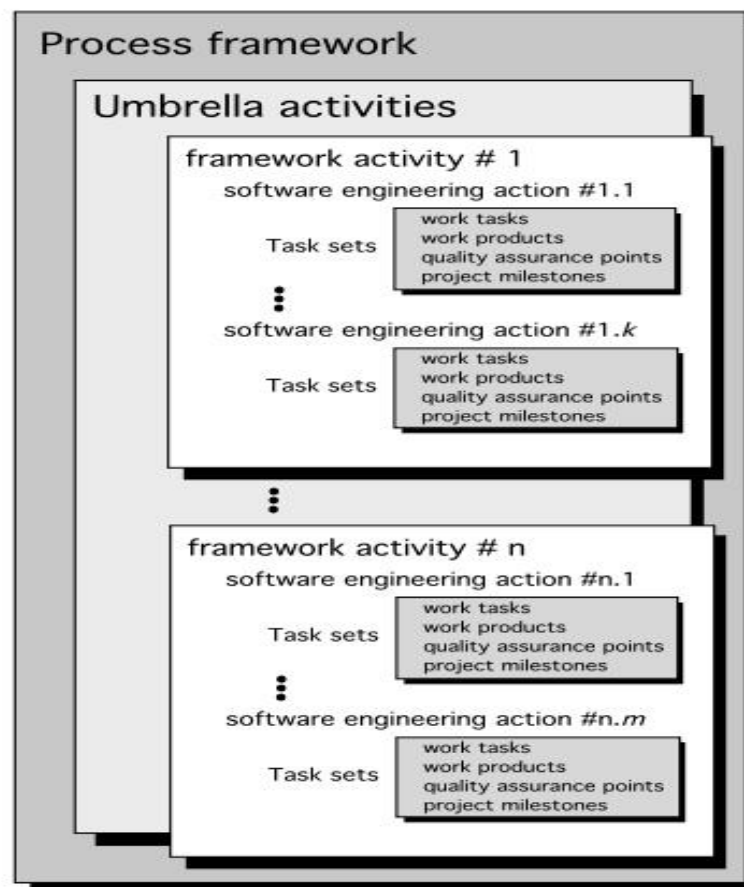
Software Process Structure berhubungan dengan struktur atau komponen yang dijadikan dasar di dalam pengembangan suatu *software*.

Misalnya:

Anda akan membuat suatu aplikasi kepegawaian pada perusahaan, maka Anda dapat mengikut salah satu proses pengembangan software, yang dapat terdiri dari kerangka kerja atau framework, metode dan kegiatan-kegiatan di dalamnya yang mendukung proses pembuatan software.

Berikut ini adalah kerangka proses dari pengembangan sebuah *software*:

Software process



Pada setiap kerangka aktivitas akan terdapat sekumpulan tugas yang harus dijalankan dalam satu proses pengembangan *software*. Sekumpulan tugas tersebut dapat didefinisikan sebagai kerja nyata yang dilakukan untuk mencapai tujuan dari software engineering. Sekumpulan tugas tersebut dapat berupa:

- Daftar tugas yang harus diselesaikan
- Daftar produk kerja yang akan dihasilkan
- Daftar filter penjaminan kualitas yang harus dipakai

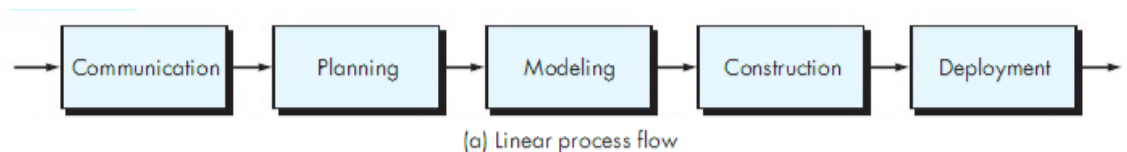
Contoh kegiatan di dalam proses pengembangan perangkat lunak misalnya:

- Mendesain basis data
- Mendesain user interface
- Mendesain suatu class
- Mendesain beberapa fungsi yang digunakan

Aliran proses dari struktur proses sebuah software dapat dibagi menjadi beberapa tipe:

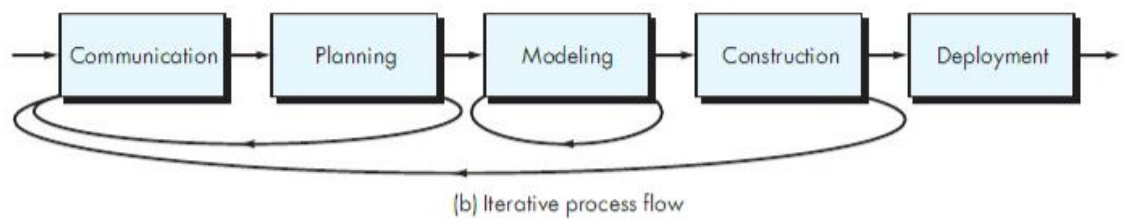
- Aliran proses linear.

Pada aliran proses linear, proses dilakukan dalam satu *flow* dari awal hingga akhir pada satu garis lurus. Pada gambar dibawah ini, dapat dilihat bahwa aliran proses dimulai dari proses komunikasi, dilanjutkan pada proses perencanaan, pemodelan, pembuatan *software* hingga deployment. Pada aliran proses linear, tidak memungkinkan untuk dilakukan aliran balik dari proses yang telah dilakukan ke proses sebelumnya.



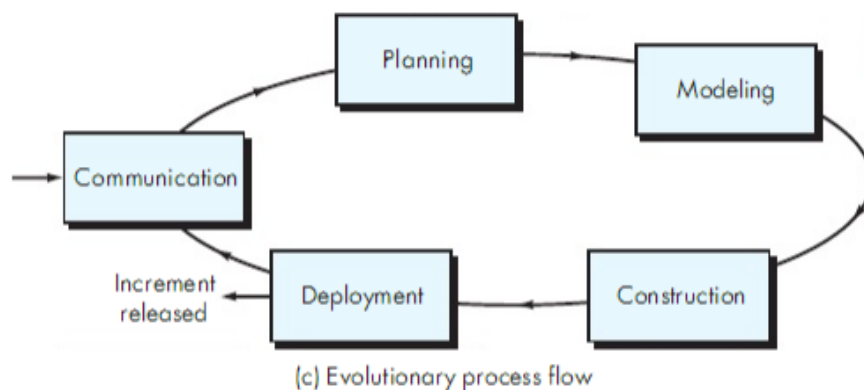
- Aliran proses iterative (berulang)

Aliran proses iterative merupakan aliran proses yang dapat dilakukan secara berulang. Pada contoh dibawah ini, setelah melewati proses komunikasi dan perencanaan, apabila dibutuhkan melakukan komunikasi ulang, proses dapat kembali ke tahapan sebelumnya. Begitu pula proses perulangan pada tahapan itu sendiri.



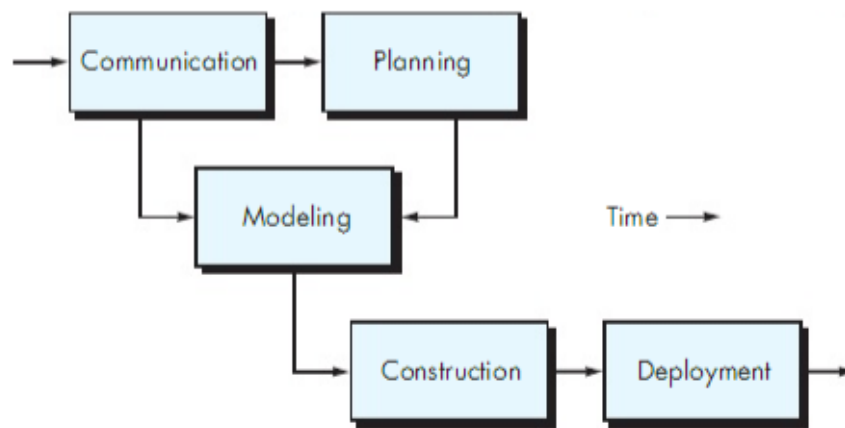
- Aliran proses evolusioner

Aliran proses evolusioner merupakan proses yang melakukan iterasi terhadap satu siklus proses yang ada. Seperti tampak pada gambar, sebuah software dapat dibagi menjadi beberapa modul, dimana satu modul akan melewati satu fase proses dari komunikasi hingga *deployment*, kemudian dimulai lagi untuk komunikasi jika dibutuhkan perbaikan atau lanjut ke modul selanjutnya. Berikut ini adalah ilustrasi dari aliran proses evolusioner:



- **Aliran proses parallel**

Aliran proses parallel menunjukkan proses yang dapat dijalankan secara bersamaan antara 2 atau lebih proses. Seperti pada contoh dibawah ini, planning dan modeling dilakukan secara parallel setelah proses komunikasi dilakukan.



(d) Parallel process flow

2. *Process Models*

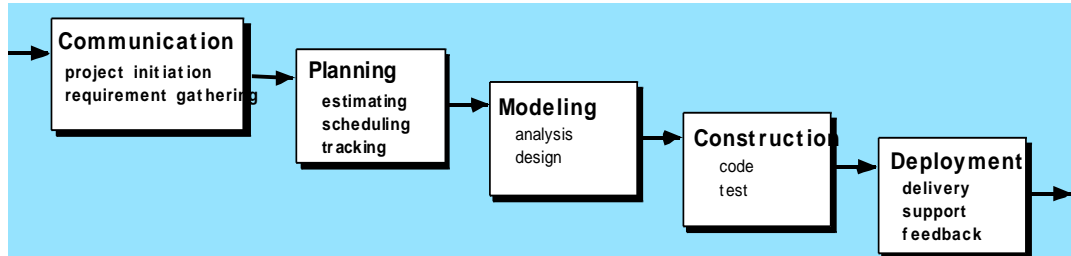
Proses model dibagi menjadi dua jenis: *prescriptive* dan *agile*. *Software* process model *prescriptive* menganjurkan pendekatan yang teratur dan cenderung kaku. Hal ini menimbulkan beberapa pertanyaan, seperti:

- Jika model *prescriptive* menuntut model yang terstruktur dan teratur, apakah model tersebut cocok digunakan untuk mengembangkan *software* yang membutuhkan perubahan?
- Sebaliknya, jika proses model *prescriptive* yang lebih tradisional ini diganti dengan model yang kurang terstruktur, apakah mungkin untuk melakukan koordinasi dan koherensi kerja yang baik dalam pengembangan sebuah *software*?

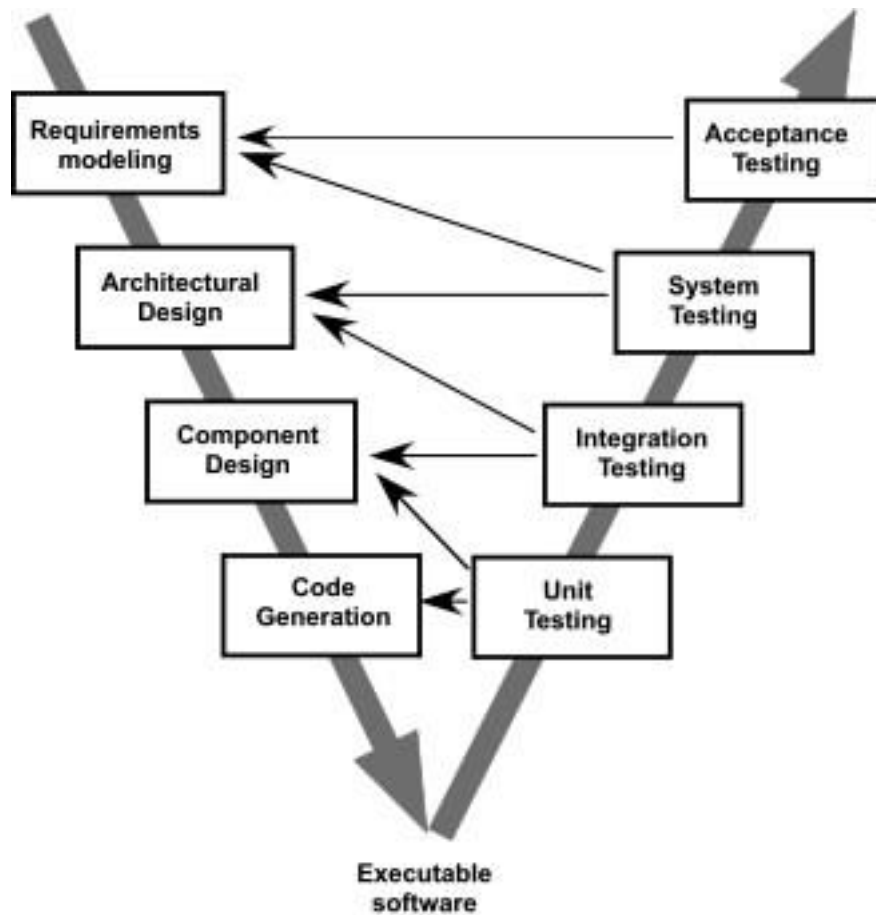
Silahkan dilakukan diskusi mengenai kedua pertanyaan diatas.

Berikut ini adalah contoh proses model *prescriptive*:

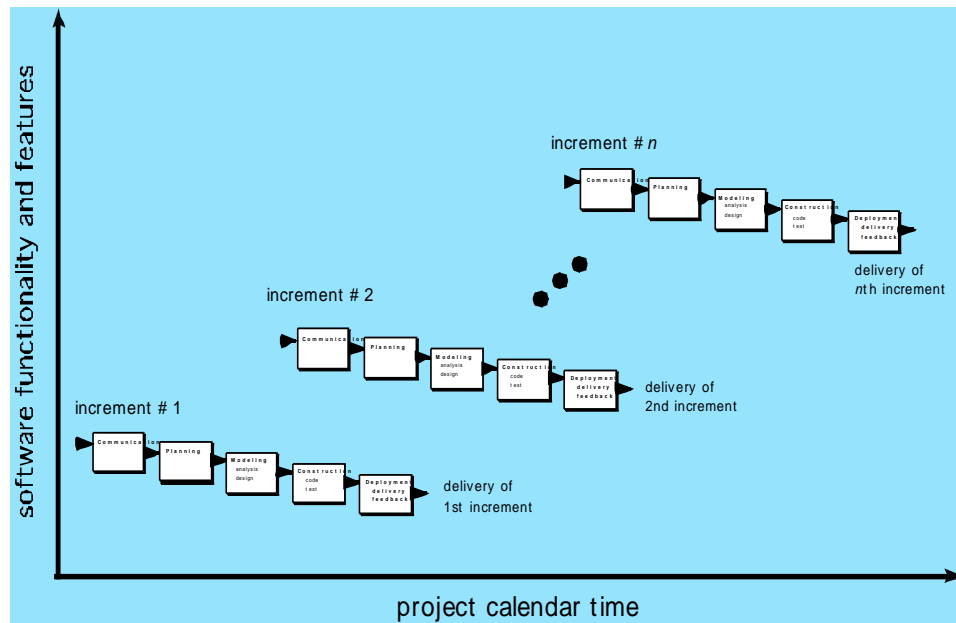
A. Waterfall Model



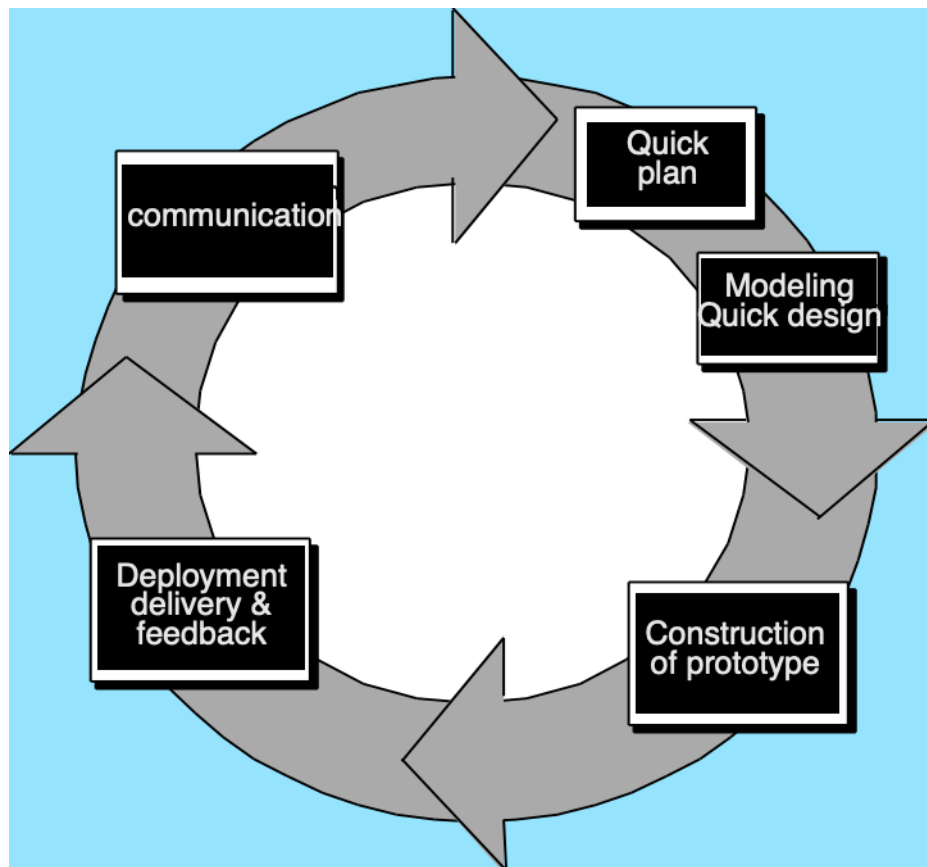
B. V- Model



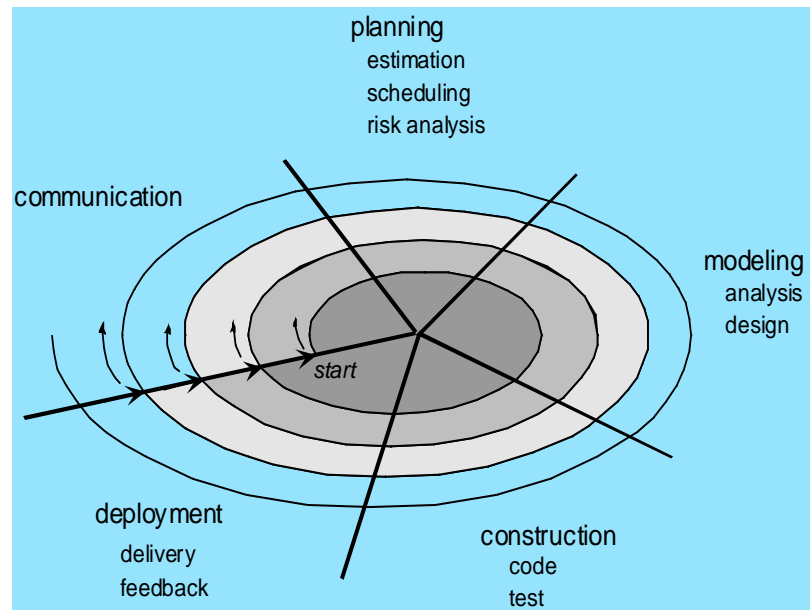
C. Incremental Model



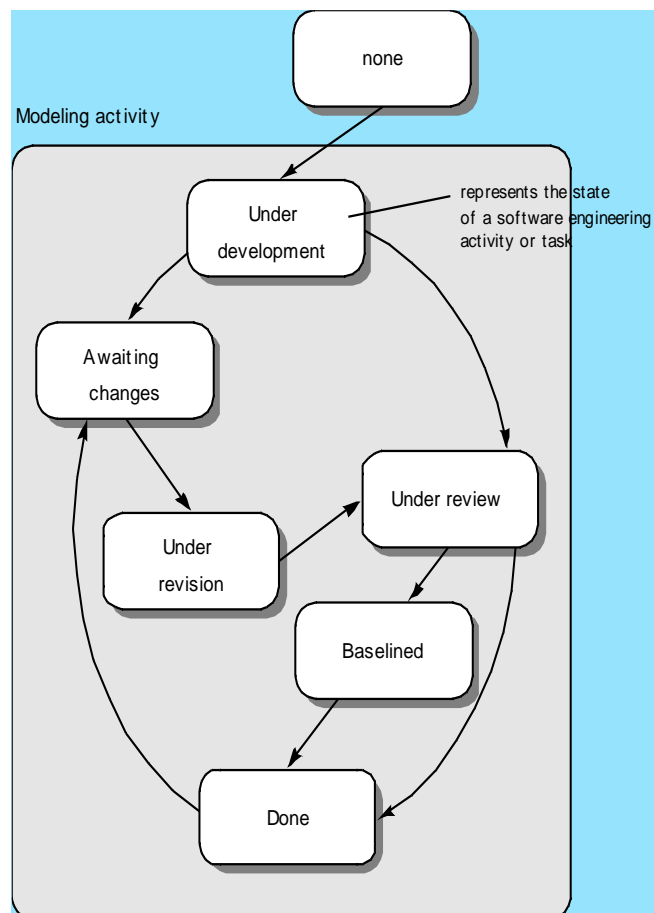
D. Model evolutioner: Prototyping



E. Model evolutioner: *Spiral*



F. Model evolutioner: *Concurrent*

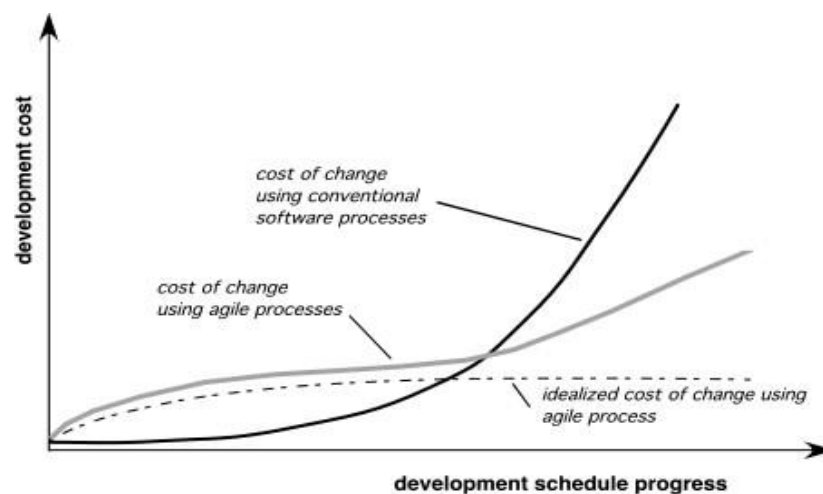


3. Agile Development

Agility merupakan:

- Efektif (cepat dan *adaptive*) dalam merespon perubahan yang ada
- Komunikasi yang efektif di antara *stakeholders*
- Melibatkan customer dalam tim pengembang
- Mengorganisasikan tim sehingga dapat mengontrol kinerja dari pekerjaan
- Cepat, dan secara *incremental software* disampaikan kepada *customer*

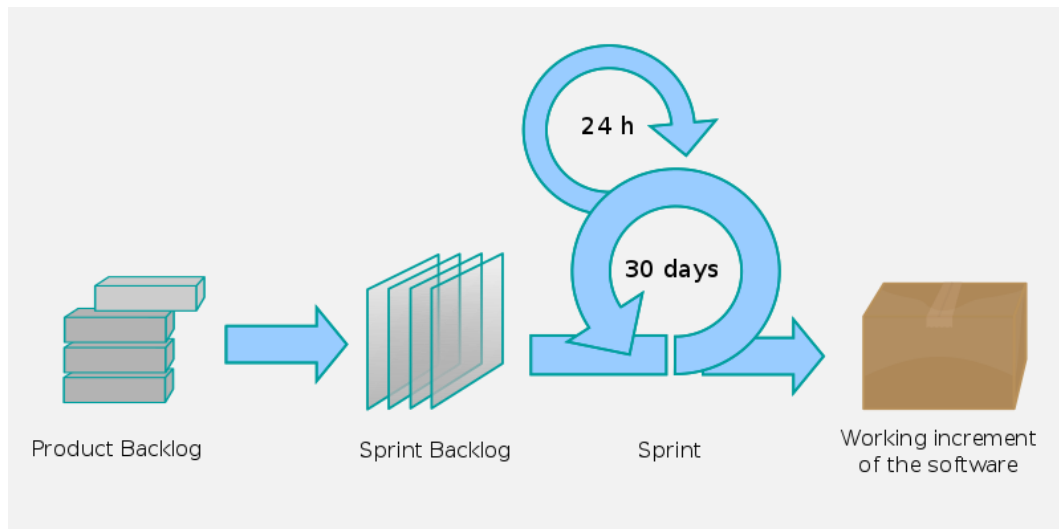
Berikut ini adalah diagram yang menggambarkan hubungan antara *agility* dan biaya yang dikeluarkan untuk mengakomodasi perubahan:



Ada beberapa tipe proses model *agile*:

- *Extreme programming*
- *Adaptive software development*
- *Dynamic system development method*
- *Scrum*
- *Crystal*
- *Feature Driven Development*

Masing-masing proses model di atas memiliki kelebihan dan kekurangan. Kita juga dapat melakukan kombinasi di dalam penggunaan metode tersebut, misalnya kombinasi antara implementasi Scrum dan Extreme Programming.



Contoh Proses Scrum

Di dalam menjalankan proyek pengembangan perangkat lunak berbasis agile, para software engineer harus mengikuti panduan pelaksanaan agile seperti yang tertuang pada agile manifestore berikut:

“Kami menemukan cara yang lebih baik untuk mengembangkan perangkat lunak dengan melakukan dan membantu sesama untuk menggunakannya.

Melalui usaha ini kami telah dapat menghargai:

Individu dan interaksi lebih dari proses dan sarana perangkat lunak

Perangkat lunak yang bekerja lebih dari dokumentasi yang menyeluruh

Kolaborasi dengan klien lebih dari negosiasi kontrak

Tanggap terhadap perubahan lebih dari mengikuti rencana

Demikian, walaupun kami menghargai hal di sisi kanan, kami lebih menghargai hal di sisi kiri.”

4. Human Aspects of Software Engineering

Karakteristik dari software engineer yang dikategorikan sebagai sifat yang efektif:

- Memiliki kesadaran akan tanggung jawab individual
- Memiliki kesadaran yang tajam
- Jujur
- Ketahanan dalam menghadapi tekanan
- Memiliki rasa keadilan yang tinggi
- Memberikan perhatian pada detail
- Pragmatis

Studi Kasus

Sebelum memulai mengerjakan proyek pengembangan perangkat lunak, Anda harus menentukan terlebih dahulu mengenai metode apa yang dipakai apakah menggunakan metode waterfall dan agile. Jika business requirement dapat ditentukan secara jelas pada saat awal pengembangan, maka waterfall cocok untuk digunakan, sedangkan jika business requirement tidak dapat ditentukan dengan jelas di awal dan terdapat ketidakpastian yang tinggi, maka pendekatan agile lebih baik untuk digunakan.

DAFTAR PUSTAKA

- Software engineering : a practitioners approach : Chapter 3/ Pages 30, Chapter 4/Pages 40, Chapter 5/Pages 66, Chapter 6/Pages 87
- Software Engineering Incremental Model,
<http://www.youtube.com/watch?v=9cBkihYP1rY>
- Video Water Fall, V model,, <http://www.youtube.com/watch?v=KaPC0gsEQ68>
- https://id.wikipedia.org/wiki/Berkas:Scrum_process.svg