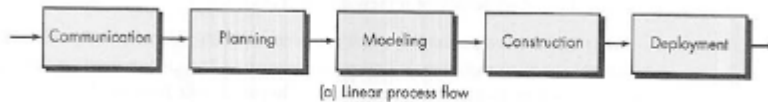


Software Process Model

1. Linear Process Flow = Waterfall



Linear Process Flow mengeksekusi satu persatu dari lima aktivitas framework secara berurutan. Model proses ini merupakan yang tertua diantara yang lain. Karena itulah saya posisikan di urutan pertama, karena pastinya sudah terbukti kinerjanya sehubungan dengan penggunaanya yang sudah lama digunakan. Selain alasan itu, karena proses ini dikerjakan berurutan dan terstruktur dengan baik, jadi lebih efisien.

✓ Kelebihan

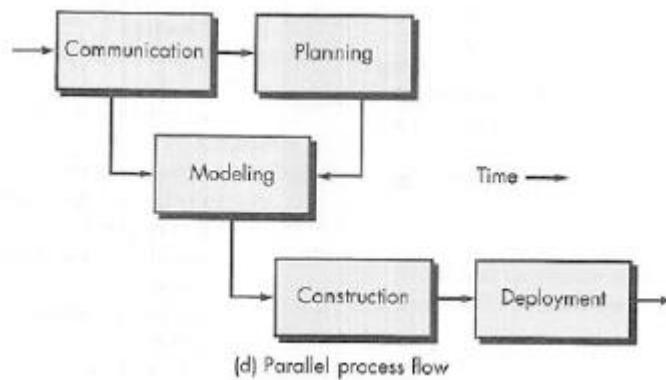
- 1) Mudah untuk diaplikasikan.
- 2) Cocok untuk system software berskala besar.
- 3) Cocok untuk system software yang bersifat generic.
- 4) Pengerjaan project system akan terjadwal dengan baik dan mudah dikontrol.
- 5) Cocok digunakan untuk produk software yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.
- 6) Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.
- 7) Document pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.
- 8) Software yang dikembangkan dengan metode ini biasanya menghasilkan kualitas yang baik.
- 9) Merupakan model pengembangan paling handal dan paling lama digunakan.
- 10) Model ini akan memudahkan para pengembang perangkat lunak untuk menetapkan tanggung jawab, memonitor penjadwalan proyek serta akuntabilitas peran personal dalam proyek perangkat lunak.
- 11) Memiliki tingkat visibilitas yang tinggi (setiap tahap mempunyai dokumen yang jelas).

✓ Kekurangan

- 1) Susah dalam merespon kebutuhan customer.
- 2) Model ini hanya cocok jika kebutuhan customer sudah sangat jelas dan perubahan kebutuhan dibatasi.
- 3) Customer harus sabar untuk menanti produk selesai, karena dikerjakan tahap per tahap, menyelesaikan tahap awal baru bisa ke tahap selanjutnya.
- 4) Perubahan ditengah-tengah pengerjaan produk akan membuat bingung team work yang sedang membuat produk.
- 5) Adanya waktu menganggur bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya.
- 6) Hasil software yang dikembangkan baru akan diketahui lama setelah proyek pengembangan dimulai.

- 7) Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses.

2. Parallel Process Flow = Rapid Application Development (RAD)



Parallel Process Flow mengeksekusi satu atau lebih dari aktifitas paralel secara bersamaan dengan aktivitas lain. Jadi aktivitas ini menghemat waktu. Meski memerlukan banyak pekerja, namun cara ini cukup efektif.

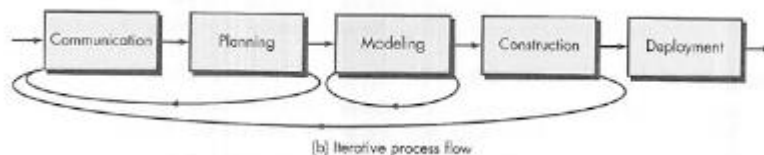
✓ Kelebihan

- 1) Lebih efektif dari pendekatan *waterfall/sequential linear* dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan.
- 2) Cocok untuk proyek yang memerlukan waktu yang singkat.

✓ Kekurangan

- 1) RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.
- 2) Membutuhkan orang yang banyak untuk menyelesaikan sebuah proyek berskala besar.
- 3) Pengembang dan customer harus punya komitmen yang kuat untuk menyelesaikan sebuah software.
- 4) Jika sistem tidak dibangun dengan benar maka RAD akan bermasalah.
- 5) Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan customer.

3. Iterative Process Flow = Incremental



Iterative Process Flow mengulangi satu atau lebih aktivitas sebelum memproses proses selanjutnya. Jadi memerlukan waktu yang cukup panjang, serta tenaga yang lebih. Selain itu, pelanggan akan mengira aktivitas yang pertama dibuat merupakan aktivitas yang sudah baku, padahal bisa jadi itu masih sebuah prototipe yang belum final. Karena itulah, metode ini saya tempatkan di urutan ketiga.

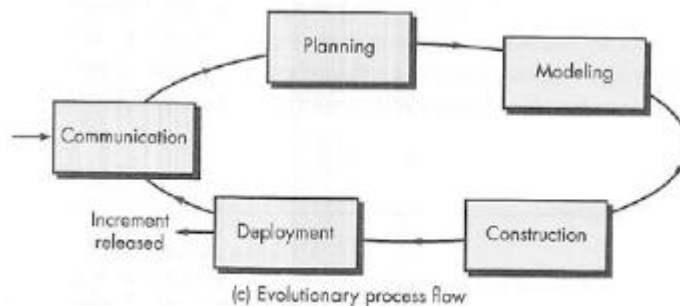
✓ Kelebihan

- 1) Mampu mengakomodasi perubahan kebutuhan customer.
- 2) Cocok digunakan bila pembuat software tidak banyak/kekurangan pembuat.
- 3) Merupakan model dengan manajemen yang sederhana.
- 4) Pihak konsumen dapat langsung menggunakan dahulu bagian-bagian yang telah selesai dibangun
- 5) Resiko untuk kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah dapat ditemukan pada beberapa inkremen, bias saja beberapa inkremen diserahkan dengan sukses kepada pelanggan.
- 6) Pelanggan akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada inkremen system yang paling kecil.
- 7) Memaksimalkan pengembalian modal investasi konsumen.

✓ Kekurangan

- 1) Penambahan staff dilakukan jika hasil incremental akan dikembangkan lebih lanjut.
- 2) Hanya cocok untuk proyek dengan skala kecil.
- 3) Kemungkinan tiap bagian tidak dapat diintegrasikan.
- 4) Hanya akan berhasil jika tidak ada staffing untuk penerapan secara menyeluruh.
- 5) Inkremen harus relative lebih kecil (tidak lebih dari 20.000 baris kode) dan setiap inkremen harus menyediakan sebagian dari fungsional system.
- 6) Adanya kesulitan untuk memetakan persyaratan pelanggan pada inkremen dengan ukuran yang benar.

4. Evolutionary Process Flow = Prototyping, Spiral



Evolutionary Process Flow mengeksekusi aktivitas dalam sebuah lingkaran. Merupakan metode yang terbaru. Jadi belum terbukti keefektifan kinerjanya. Maka dari itu, saya tempatkan di urutan keempat.

Prototyping :

✓ Kelebihan

- 1) Adanya komunikasi yang baik antara pengembang dan pelanggan.
- 2) Dapat menghemat waktu pengembangan.
- 3) User dapat berpartisipasi aktif dalam pengembangan sistem.
- 4) Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- 5) Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.
- 6) Dapat memberikan bukti konsep yang diperlukan untuk menarik dana.

7) Meningkatkan kecepatan pengembangan sistem.

✓ Kekurangan

- 1) Proses perancangan dan analisis terlalu singkat.
- 2) pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.
- 3) Biasanya kurang fleksibel terhadap perubahan.
- 4) Ketidaksadaran user bahwa ini hanya suatu model awal bukan model akhir,
- 5) Pengembang kadang-kadang membuat implementasi yang sembarangan.
- 6) Teknik dan tools yang tidak optimal pada prototipe yang akan tetap digunakan pada software sesungguhnya.

Spiral :

✓ Kelebihan

- 1) Pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses.
- 2) Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar.
- 3) Menggunakan prototype sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk.
- 4) Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi permasalahan yang serius.

✓ Kekurangan

- 1) Sulit meyakinkan pelanggan bahwa pendekatan evolusioner bisa dikontrol.
- 2) Memerlukan tenaga ahli untuk memperkirakan resiko, dan harus mengandalkannya supaya sukses.
- 3) Butuh waktu lama untuk menerapkan paradigma ini menuju kepastian yang absolut.
- 4) Model spiral ini merupakan model yang masih baru sehingga belum terbukti apakah model ini efisien atau tidak..