MODUL REKAYASA PERANGKAT LUNAK (RPL)

(CCC-110)

MODUL 03

Iterative Waterfall Model, Prototyping Model, Evolutionary Model , Spiral Model

DISUSUN OLEH

MALABAY,S.KOM,M.KOM

UNIVERSITAS ESA UNGGUL

2020

Iterative Waterfall Model, Prototyping Model, Evolutionary Model , Spiral Model

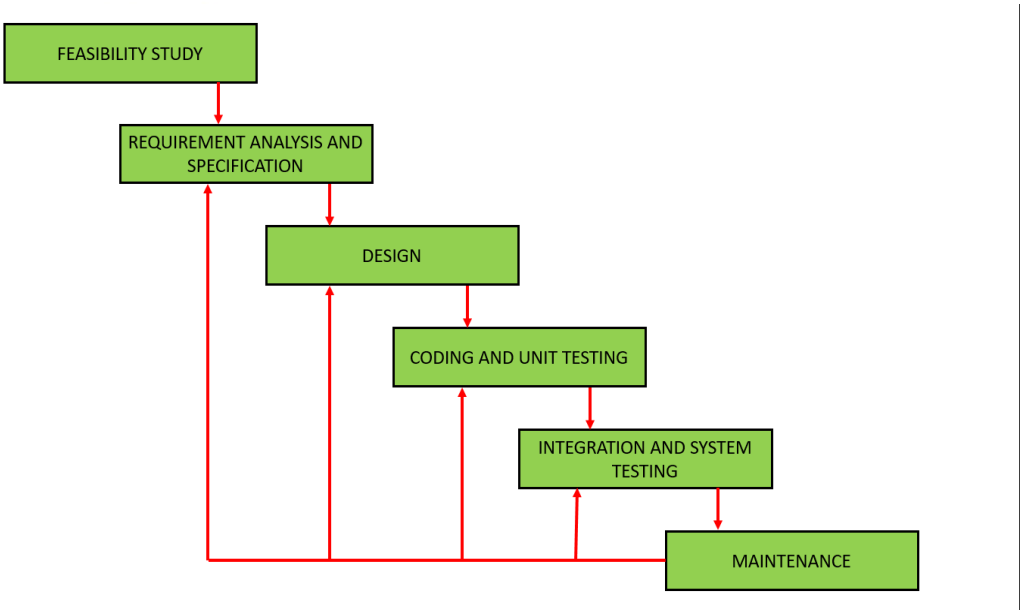
A. Kemampuan Akhir Yang Diharapkan

Setelah mempelajari modul ini, diharapkan mahasiswa mampu : Mahasiswa mampu memahami pengertian *Iterative Waterfall Model, Prototyping Model, Evolutionary Model , Spiral Model*

B. Uraian dan Contoh

Dalam proyek pengembangan perangkat lunak praktis, model air terjun klasik sulit digunakan. Jadi, model air terjun berulang dapat dianggap menggabungkan perubahan yang diperlukan pada model air terjun klasik agar dapat digunakan dalam proyek pengembangan perangkat lunak praktis. Ini hampir sama dengan model air terjun klasik kecuali beberapa perubahan dilakukan untuk meningkatkan efisiensi pengembangan perangkat lunak.

Model air terjun iteratif memberikan jalur umpan balik dari setiap fase ke fase sebelumnya, yang merupakan perbedaan utama dari model waterfall klasik.



Gambar 1. *Iterative Waterfall Model*

Ketika kesalahan terdeteksi di beberapa fase selanjutnya, jalur umpan balik ini memungkinkan koreksi kesalahan yang dilakukan oleh programmer selama beberapa fase. Jalur umpan balik memungkinkan fase untuk dikerjakan ulang di mana kesalahan dilakukan dan perubahan ini tercermin dalam fase selanjutnya. Namun, tidak ada jalur umpan balik ke tahap - studi kelayakan, karena sekali proyek telah diambil, proyek tidak mudah menyerah.
Mendeteksi kesalahan dalam fase yang sama saat terjadi adalah hal yang baik. Ini mengurangi usaha dan waktu yang dibutuhkan untuk memperbaiki kesalahan.

Fase Penahanan Kesalahan: Prinsip mendeteksi kesalahan sedekat mungkin dengan titik komitmen mereka dikenal sebagai Fase penahanan kesalahan.

Keunggulan Model Air Terjun Iteratif
Jalur Umpan Balik: Dalam model air terjun klasik, tidak ada jalur umpan balik, jadi tidak ada mekanisme untuk koreksi kesalahan. Namun dalam jalur umpan balik model air terjun iteratif dari satu fase ke fase sebelumnya memungkinkan koreksi kesalahan yang dilakukan dan perubahan ini tercermin di fase selanjutnya.
Sederhana: Model air terjun berulang sangat mudah dipahami dan digunakan. Itulah mengapa ini adalah salah satu model pengembangan perangkat lunak yang paling banyak digunakan.

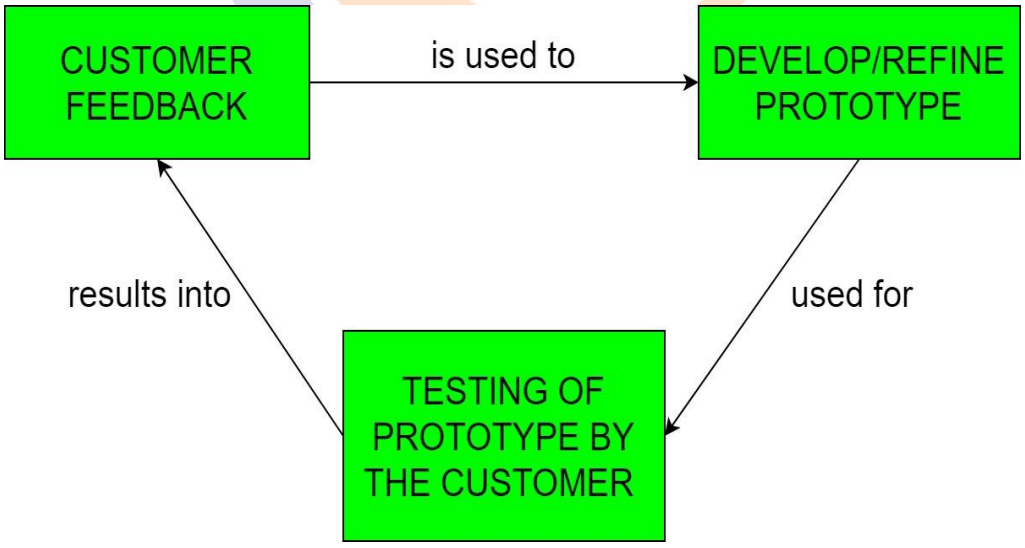
Kekurangan Model Air Terjun Iteratif
Sulit untuk memasukkan permintaan perubahan: Kelemahan utama dari model air terjun iteratif adalah bahwa semua persyaratan harus dinyatakan dengan jelas sebelum memulai fase pengembangan. Pelanggan dapat mengubah persyaratan setelah beberapa waktu, tetapi model waterfall berulang tidak meninggalkan ruang lingkup apa pun untuk memasukkan permintaan perubahan yang dibuat setelah fase pengembangan dimulai.
Pengiriman inkremental tidak didukung: Dalam model air terjun iteratif, perangkat lunak lengkap dikembangkan dan diuji sepenuhnya sebelum dikirimkan ke pelanggan. Tidak ada ruang untuk pengiriman perantara. Sehingga pelanggan harus menunggu lama untuk mendapatkan software tersebut.

Tumpang tindih fase tidak didukung: Model air terjun berulang mengasumsikan bahwa satu fase dapat dimulai setelah menyelesaikan fase sebelumnya. Namun dalam proyek nyata, fase mungkin tumpang tindih untuk mengurangi tenaga dan waktu yang dibutuhkan untuk menyelesaikan proyek.

Penanganan risiko tidak didukung: Proyek mungkin mengalami berbagai jenis risiko. Namun, model air terjun berulang tidak memiliki mekanisme penanganan risiko.

Interaksi pelanggan terbatas: Interaksi pelanggan terjadi pada awal proyek pada saat pengumpulan persyaratan dan pada penyelesaian proyek pada saat pengiriman perangkat lunak. Interaksi yang lebih sedikit dengan pelanggan ini dapat menyebabkan banyak masalah karena perangkat lunak yang akhirnya dikembangkan mungkin berbeda dari kebutuhan pelanggan yang sebenarnya.

Prototyping didefinisikan sebagai proses mengembangkan replikasi kerja dari suatu produk atau sistem yang harus direkayasa. Ini menawarkan faksimili skala kecil dari produk akhir dan digunakan untuk mendapatkan umpan balik pelanggan.



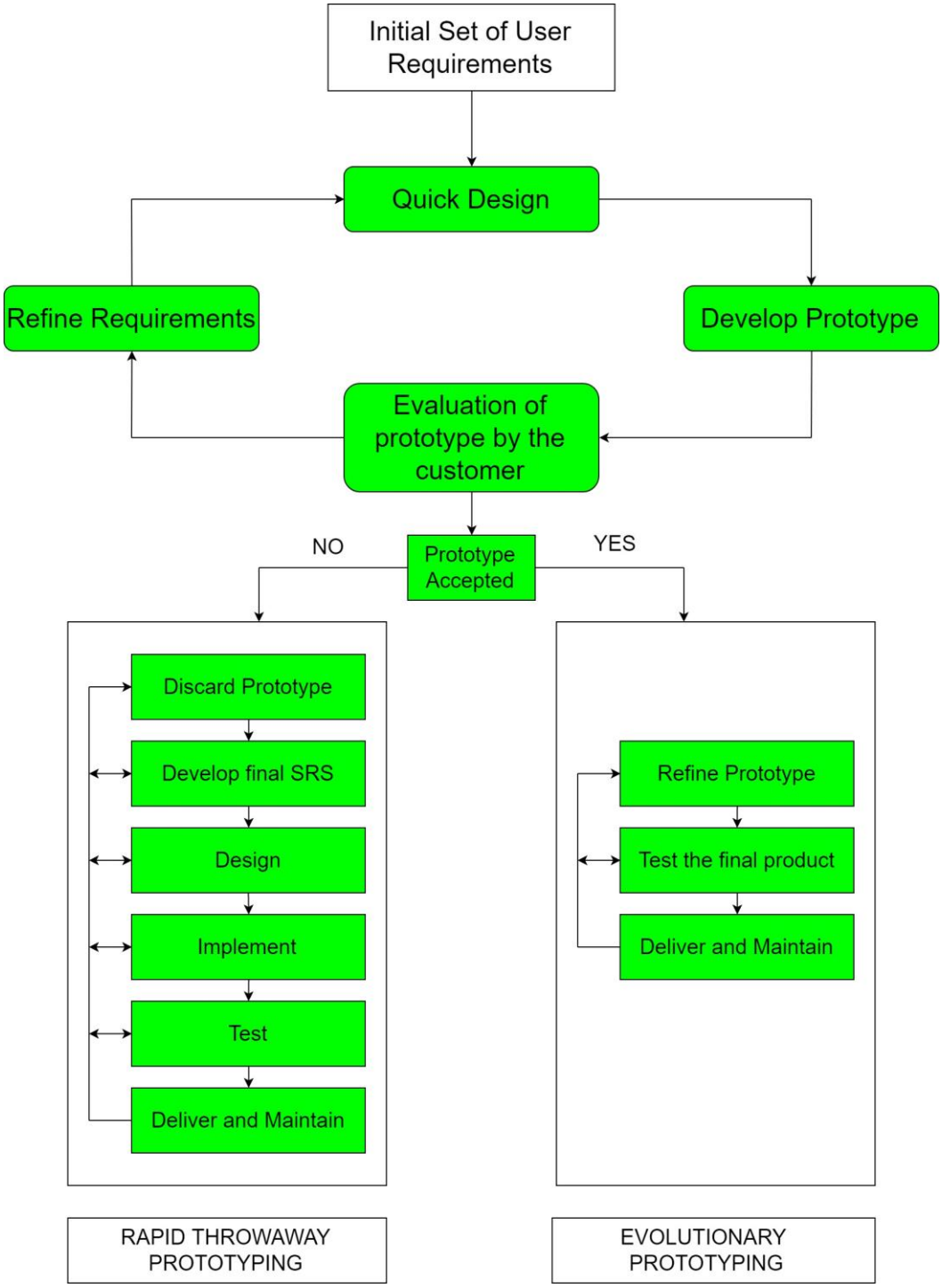
Gambar 2. *Prototyping Model*

Model Prototyping adalah salah satu Model Siklus Hidup Pengembangan Perangkat Lunak (model SDLC) yang paling populer digunakan. Model ini digunakan ketika pelanggan tidak mengetahui persyaratan proyek yang tepat sebelumnya. Dalam model ini, prototipe produk akhir pertama kali dikembangkan, diuji, dan disempurnakan sesuai umpan balik pelanggan berulang kali hingga prototipe akhir yang dapat diterima tercapai yang menjadi dasar untuk mengembangkan produk akhir.

Dalam model proses ini, sistem diimplementasikan secara parsial sebelum atau selama tahap analisis sehingga memberikan kesempatan kepada pelanggan untuk melihat produk di awal siklus hidup. Prosesnya dimulai dengan mewawancarai pelanggan dan mengembangkan model kertas tingkat tinggi yang tidak lengkap. Dokumen ini digunakan untuk membangun prototipe awal yang hanya mendukung fungsionalitas dasar yang diinginkan oleh pelanggan. Setelah pelanggan mengetahui masalahnya, prototipe tersebut selanjutnya disempurnakan untuk menghilangkannya. Proses berlanjut sampai pengguna menyetujui prototipe dan menemukan model kerja yang memuaskan.

Ada 2 pendekatan untuk model ini:
Pembuatan Prototipe Pembuangan Cepat -
Teknik ini menawarkan metode yang berguna untuk mengeksplorasi ide dan mendapatkan umpan balik pelanggan untuk masing-masing ide. Dalam metode ini, prototipe yang dikembangkan tidak perlu menjadi bagian dari prototipe yang akhirnya diterima. Umpan balik pelanggan membantu dalam mencegah kesalahan desain yang tidak perlu dan karenanya, prototipe akhir yang dikembangkan memiliki kualitas yang lebih baik.
Pembuatan Prototipe Evolusioner -
Dalam metode ini, prototipe yang awalnya dikembangkan secara bertahap disempurnakan berdasarkan umpan balik pelanggan hingga akhirnya diterima. Dibandingkan dengan Rapid Throwaway Prototyping, ini menawarkan pendekatan yang lebih baik yang menghemat waktu dan tenaga. Ini karena

mengembangkan prototipe dari awal untuk setiap iterasi proses terkadang bisa sangat membuat frustasi bagi pengembang.



Gambar 3. Pendekatan untuk Model Prototyping

Model pembuatan prototipe juga merupakan model siklus hidup pengembangan perangkat lunak yang populer. Model prototyping dapat dianggap sebagai perpanjangan dari model Air Terjun Iteratif. Model ini menyarankan untuk membangun Prototipe sistem yang berfungsi, sebelum pengembangan perangkat lunak yang sebenarnya.

Prototipe adalah mainan dan implementasi kasar dari suatu sistem. Ini memiliki kemampuan fungsional yang terbatas, keandalan yang rendah, atau kinerja yang tidak efisien dibandingkan dengan perangkat lunak yang sebenarnya. Prototipe dapat dibangun dengan sangat cepat dengan menggunakan beberapa pintasan dengan mengembangkan fungsi yang tidak efisien, tidak akurat, atau dummy.

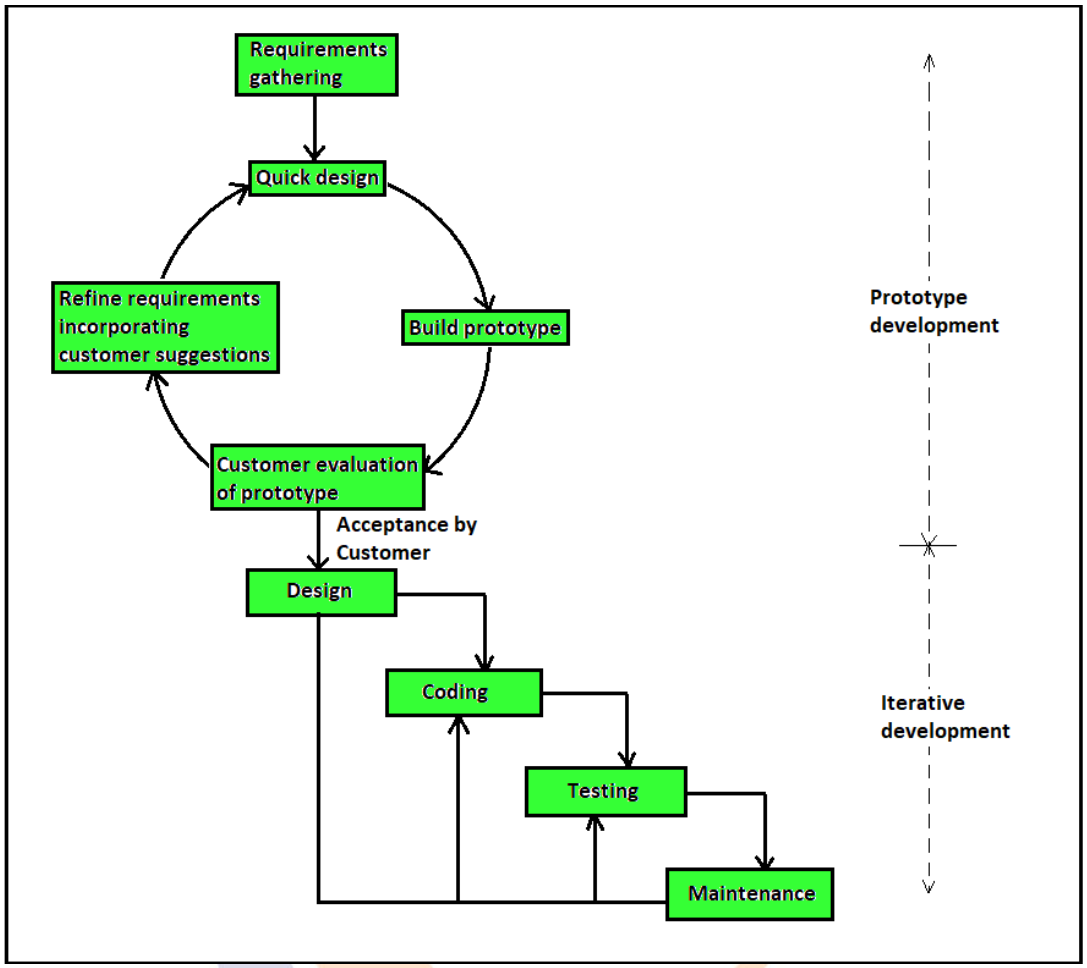
Perlunya Model Prototyping -

Ini menguntungkan untuk mengembangkan bagian Graphical User Interface (GUI) dari perangkat lunak menggunakan Model Prototyping. Melalui prototipe, pengguna dapat bereksperimen dengan antarmuka pengguna yang berfungsi dan mereka dapat menyarankan perubahan apa pun jika diperlukan.

Model pembuatan prototipe sangat berguna ketika solusi teknis yang tepat tidak jelas bagi tim pengembangan. Prototipe dapat membantu mereka untuk memeriksa secara kritis masalah teknis yang terkait dengan pengembangan produk. Kurangnya keakraban dengan teknologi pengembangan yang dibutuhkan adalah risiko teknis. Ini dapat diselesaikan dengan mengembangkan prototipe untuk memahami masalah dan mengakomodasi perubahan pada iterasi berikutnya.

Tahapan Model Prototyping -

Model Pembuatan Prototipe pengembangan perangkat lunak secara grafis ditunjukkan pada gambar di bawah ini. Perangkat lunak ini dikembangkan melalui dua kegiatan utama - satu adalah pembuatan prototipe dan yang lainnya adalah pengembangan perangkat lunak berbasis air terjun berulang.



Gambar 4. Tahapan Mode Pembuatan Prototipe

Pengembangan Prototipe - Pengembangan prototipe dimulai dengan fase pengumpulan persyaratan awal. Sebuah desain cepat dilakukan dan prototipe dibangun. Prototipe yang dikembangkan diserahkan ke pelanggan untuk evaluasi. Berdasarkan umpan balik pelanggan, persyaratan disempurnakan dan prototipe dimodifikasi dengan sesuai. Siklus untuk mendapatkan umpan balik pelanggan dan memodifikasi prototipe berlanjut sampai pelanggan menyetujui prototipe tersebut.

Pengembangan Iteratif - Setelah pelanggan menyetujui prototipe, perangkat lunak sebenarnya dikembangkan menggunakan pendekatan air terjun berulang. Terlepas dari ketersediaan prototipe yang berfungsi, dokumen SRS biasanya perlu dikembangkan karena Dokumen SRS sangat berharga untuk melakukan analisis traktabilitas, verifikasi, dan desain kasus uji selama fase selanjutnya.

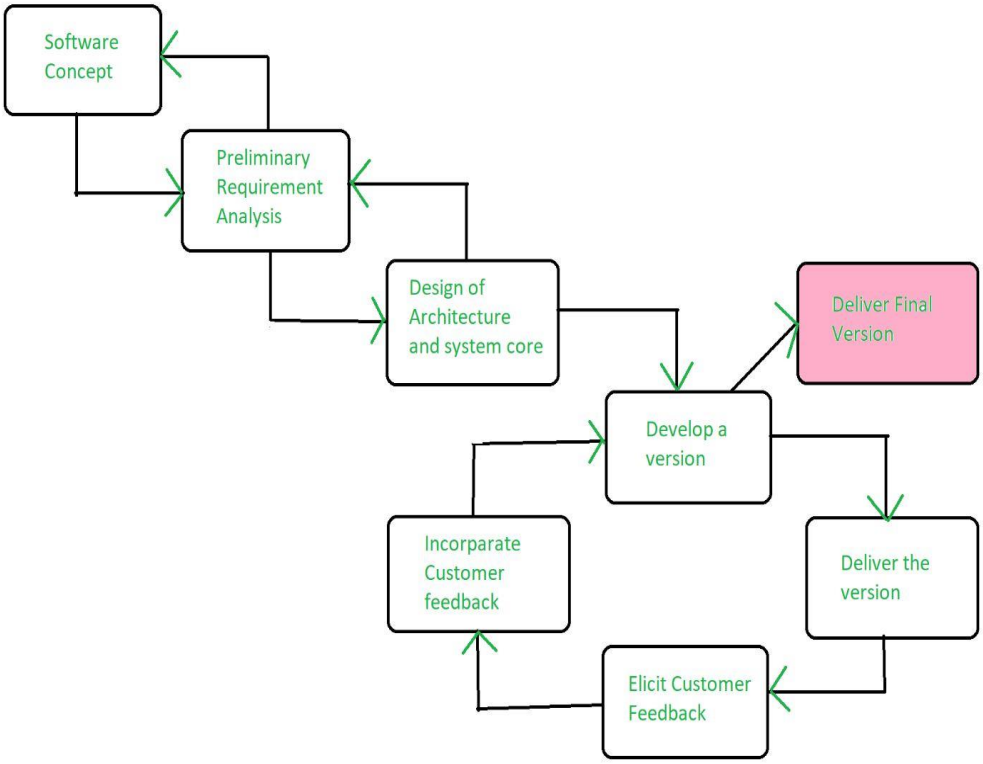
Kode untuk prototipe biasanya dibuang. Namun, pengalaman yang dikumpulkan dari pengembangan prototipe sangat membantu dalam mengembangkan perangkat lunak yang sebenarnya. Dengan membuat prototipe dan mengirimkannya untuk evaluasi pengguna, banyak persyaratan pelanggan ditentukan dengan benar dan masalah teknis diselesaikan dengan bereksperimen dengan prototipe. Ini meminimalkan permintaan perubahan kemudian dari pelanggan dan biaya desain ulang terkait.

Keuntungan Model Prototyping - Model ini paling sesuai untuk proyek yang mengalami risiko teknis dan persyaratan. Prototipe yang dibangun membantu mengatasi risiko ini.

Model evolusioner merupakan kombinasi dari Iterative dan Incremental model dari siklus hidup pengembangan perangkat lunak. Menyampaikan sistem Anda dalam rilis besar, mengirimkannya dalam proses bertahap dari waktu ke waktu adalah tindakan yang dilakukan dalam model ini. Beberapa persyaratan awal dan visi arsitektur perlu dilakukan.

Lebih baik untuk produk perangkat lunak yang rangkaian fiturnya didefinisikan ulang selama pengembangan karena umpan balik pengguna dan faktor lainnya. Model pengembangan Evolusioner membagi siklus pengembangan menjadi model air terjun inkremental yang lebih kecil di mana pengguna bisa mendapatkan akses ke produk di akhir setiap siklus.

Umpan balik diberikan oleh pengguna pada produk untuk tahap perencanaan siklus berikutnya dan tim pengembangan merespons, seringkali dengan mengubah produk, rencana atau proses. Oleh karena itu, produk perangkat lunak berkembang seiring waktu.
Semua model memiliki kelemahan yaitu durasi waktu dari mulai proyek hingga waktu pengiriman solusi sangat tinggi. Model evolusi memecahkan masalah ini dengan pendekatan yang berbeda.



Gambar 5. Model evolusioner

Model evolusioner menyarankan pemecahan pekerjaan menjadi bagian-bagian yang lebih kecil, memprioritaskannya dan kemudian mengirimkan potongan-potongan itu kepada pelanggan satu per satu. Jumlah potongan sangat besar dan merupakan jumlah pengiriman yang dilakukan ke pelanggan. Keuntungan utamanya adalah kepercayaan pelanggan meningkat karena dia terus-menerus mendapatkan barang atau jasa yang dapat diukur sejak awal proyek untuk memverifikasi dan memvalidasi persyaratannya. Model ini memungkinkan untuk mengubah persyaratan serta semua pekerjaan diuraikan menjadi potongan-potongan pekerjaan yang dapat dipelihara.

Penerapan Model Evolusi:

Ini digunakan dalam proyek besar di mana Anda dapat dengan mudah menemukan modul untuk implementasi tambahan. Model evolusioner biasanya digunakan ketika pelanggan ingin mulai menggunakan fitur inti daripada menunggu perangkat lunak lengkap.

Model evolusioner juga digunakan dalam pengembangan perangkat lunak berorientasi objek karena sistem dapat dengan mudah dibagi menjadi beberapa unit dalam hal objek.

Keuntungan:
Dalam model evolusioner, pengguna mendapat kesempatan untuk bereksperimen dengan sistem yang dikembangkan sebagian.
Ini mengurangi kesalahan karena modul inti diuji secara menyeluruh.

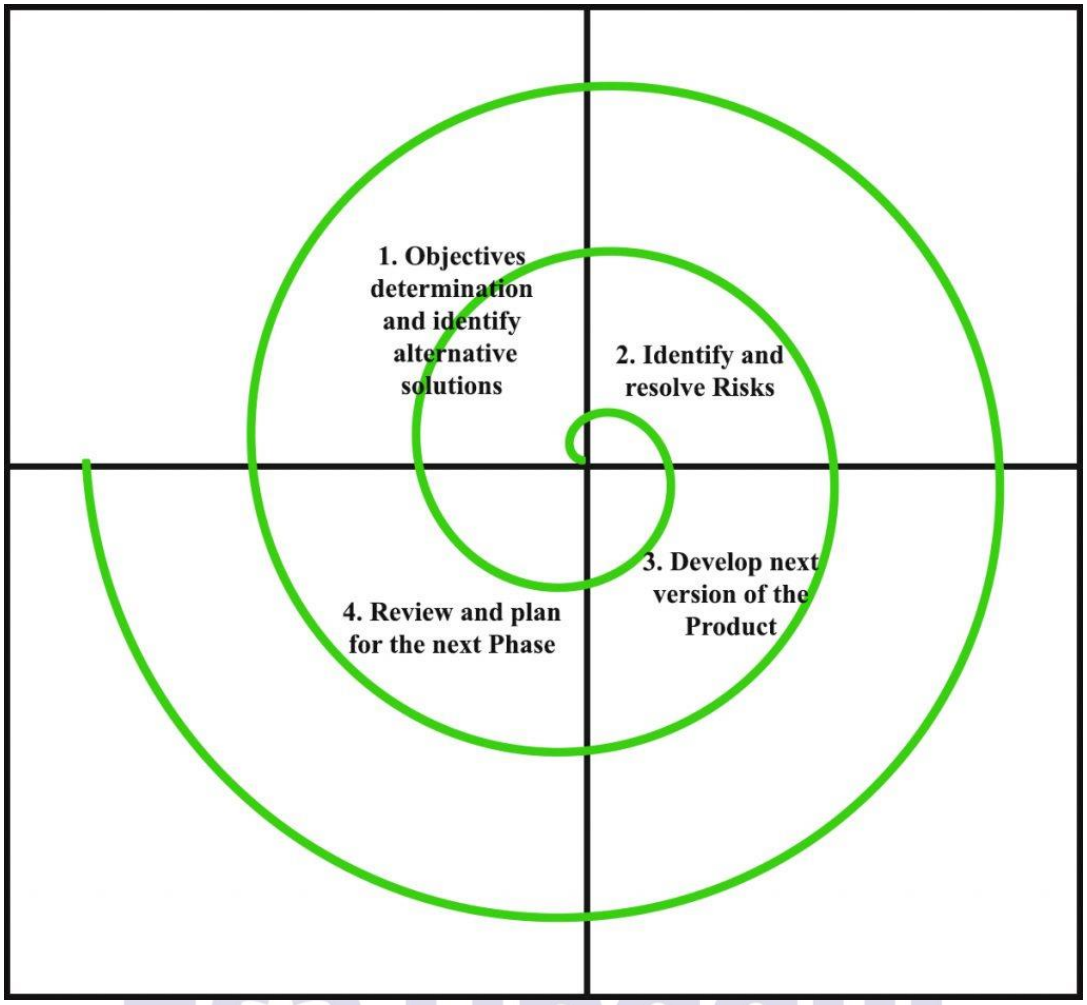
Kekurangan:
Terkadang sulit untuk membagi masalah menjadi beberapa versi yang dapat diterima pelanggan yang dapat diterapkan dan disampaikan secara bertahap

Model evolusioner juga disebut sebagai model versi berurutan dan terkadang sebagai model inkremental. Dalam model Evolutionary, kebutuhan perangkat lunak pertama-tama dipecah menjadi beberapa modul (atau unit fungsional) yang dapat dibangun dan dikirim secara bertahap (lihat Gambar 5).

Pengembangan pertama kali mengembangkan modul inti dari sistem. Modul inti adalah modul yang tidak membutuhkan layanan dari modul lain. Kerangka produk awal disempurnakan menjadi tingkat kemampuan yang meningkat dengan menambahkan fungsi baru dalam versi yang berurutan. Setiap model evolusi dapat dikembangkan dengan menggunakan model pengembangan air terjun iteratif.

Model spiral adalah salah satu model Siklus Hidup Pengembangan Perangkat Lunak yang paling penting, yang memberikan dukungan untuk Penanganan Risiko. Dalam representasi diagramnya, terlihat seperti spiral dengan banyak loop. Jumlah pasti loop spiral tidak diketahui dan dapat bervariasi dari satu proyek ke proyek lainnya. Setiap lingkaran spiral disebut Fase proses pengembangan perangkat lunak. Jumlah fase yang dibutuhkan untuk mengembangkan produk dapat bervariasi oleh manajer proyek tergantung pada risiko proyek. Karena manajer proyek secara dinamis menentukan jumlah fase, maka manajer proyek

memiliki peran penting untuk mengembangkan produk dengan menggunakan model spiral. Radius spiral di setiap titik mewakili biaya (biaya) proyek sejauh ini, dan dimensi sudut mewakili kemajuan yang dibuat sejauh ini dalam fase saat ini.



Gambar 6. Fase Model Spiral:

Fungsi dari keempat kuadran ini dibahas di bawah ini-

Penentuan tujuan dan mengidentifikasi solusi alternatif: Persyaratan dikumpulkan dari pelanggan dan tujuan diidentifikasi, diuraikan dan dianalisis pada awal setiap fase. Kemudian solusi alternatif yang memungkinkan untuk fase tersebut diusulkan di kuadran ini.

Identifikasi dan selesaikan Risiko: Selama kuadran kedua, semua solusi yang mungkin dievaluasi untuk memilih solusi terbaik. Kemudian risiko yang terkait

dengan solusi tersebut diidentifikasi dan risiko tersebut diselesaikan dengan menggunakan strategi terbaik. Di ujung kuadran ini, Prototipe dibangun untuk solusi terbaik.

Kembangkan versi Produk berikutnya: Selama kuadran ketiga, fitur yang diidentifikasi dikembangkan dan diverifikasi melalui pengujian. Di akhir kuadran ketiga, versi perangkat lunak berikutnya tersedia.

Tinjau dan rencanakan untuk Fase berikutnya: Di kuadran keempat, Pelanggan mengevaluasi versi perangkat lunak yang dikembangkan sejauh ini. Pada akhirnya, perencanaan untuk tahap selanjutnya dimulai.

Model Spiral disebut sebagai Model Meta karena model ini menggabungkan semua model SDLC lainnya. Misalnya, satu lingkaran spiral sebenarnya mewakili Model Air Terjun Iteratif. Model spiral menggabungkan pendekatan bertahap dari Model Air Terjun Klasik. Model spiral menggunakan pendekatan Prototyping Model dengan membangun prototipe pada awal setiap tahapan sebagai teknik penanganan risiko. Selain itu, model spiral dapat dianggap mendukung model evolusi - iterasi di sepanjang spiral dapat dianggap sebagai tingkat evolusi yang melaluinya sistem yang lengkap dibangun.

Keunggulan Model Spiral: Di bawah ini adalah beberapa keunggulan Model Spiral.

- Penanganan Risiko: Proyek-proyek dengan banyak risiko yang tidak diketahui yang terjadi saat pengembangan berlangsung, dalam hal ini, Model Spiral adalah model pengembangan terbaik untuk diikuti karena analisis risiko dan penanganan risiko di setiap tahap.
- Baik untuk proyek besar: Direkomendasikan untuk menggunakan Model Spiral dalam proyek besar dan kompleks.
- Fleksibilitas dalam Persyaratan: Permintaan perubahan dalam Persyaratan di tahap selanjutnya dapat digabungkan secara akurat dengan menggunakan model ini.

- Kepuasan Pelanggan: Pelanggan dapat melihat perkembangan produk pada tahap awal pengembangan perangkat lunak dan dengan demikian, mereka terbiasa dengan sistem dengan menggunakannya sebelum penyelesaian total produk.

Kerugian Model Spiral: Di bawah ini adalah beberapa kelemahan utama model spiral.

- Kompleks: Model Spiral jauh lebih kompleks daripada model SDLC lainnya.
- Mahal: Model Spiral tidak cocok untuk proyek kecil karena mahal.
- Terlalu banyak bergantung pada Analisis Risiko: Keberhasilan penyelesaian proyek sangat bergantung pada Analisis Risiko. Tanpa keahlian yang sangat berpengalaman, pengembangan proyek menggunakan model ini akan gagal.
- Kesulitan dalam manajemen waktu: Karena jumlah fase tidak diketahui pada awal proyek, maka estimasi waktu menjadi sangat sulit.

Beberapa pernyataan lain pada keuntungan dan kerugian Model Spiral adalah:
Keuntungan Model Spiral:

- Perangkat lunak diproduksi di awal siklus hidup perangkat lunak.
- Penanganan risiko adalah salah satu keuntungan penting dari model Spiral, model pengembangan terbaik untuk diikuti karena analisis risiko dan penanganan risiko di setiap tahap.
- Fleksibilitas dalam persyaratan. Dalam model ini, kami dapat dengan mudah mengubah persyaratan di tahap selanjutnya dan dapat digabungkan secara akurat. Selain itu, Fungsi tambahan dapat ditambahkan di kemudian hari.
- Ini bagus untuk proyek besar dan kompleks.
- Itu bagus untuk kepuasan pelanggan. Kami dapat melibatkan pelanggan dalam pengembangan produk pada tahap awal pengembangan perangkat lunak. Selain itu, perangkat lunak diproduksi di awal siklus hidup perangkat lunak.

- Persetujuan yang kuat dan kontrol dokumentasi.
- Sangat cocok untuk proyek berisiko tinggi, di mana kebutuhan bisnis mungkin tidak stabil. Produk yang sangat disesuaikan dapat dikembangkan menggunakan ini.

Kekurangan Model Spiral:

- Tidak cocok untuk proyek kecil karena mahal.
- Ini jauh lebih kompleks daripada model SDLC lainnya. Proses itu rumit.
- Terlalu banyak bergantung pada Analisis Risiko dan membutuhkan keahlian yang sangat spesifik.
- Kesulitan dalam manajemen waktu. Karena jumlah fase tidak diketahui pada awal proyek, maka estimasi waktu menjadi sangat sulit.
- Spiral mungkin terus berlanjut tanpa batas.
- Akhir proyek mungkin tidak diketahui lebih awal.
- Ini tidak cocok untuk proyek berisiko rendah.
- Mungkin sulit untuk menentukan pencapaian yang objektif dan dapat diverifikasi. Sejumlah besar tahap menengah membutuhkan dokumentasi yang berlebihan.

Yang membedakan Model Spiral dengan Model Waterfall Adalah:

1. Model spiral bekerja dengan metode evolusioner.
2. Dalam model spiral, kesalahan atau risiko diidentifikasi dan diperbaiki sebelumnya.
3. Model spiral diadopsi oleh pengembang.
4. Model Spiral digunakan untuk proyek besar.
5. Pada kebutuhan model spiral dan perencanaan tahap awal diperlukan jika diperlukan
6. Fleksibilitas untuk mengubah model spiral tidaklah sulit.
7. Ada risiko jumlah yang rendah.
8. Biaya model spiral sangat mahal.

C. Latihan

1. Proses mengembangkan replikasi kerja dari suatu produk atau sistem yang harus direkayasa,disebut ?

2. Model Spiral disebut sebagai ……. karena model ini menggabungkan semua model SDLC lainnya

D. Kunci Jawaban

1. Prototyping
2. Model Meta

E. Daftar Pustaka

1. Roger S. Pressman, Software Engineering A Practioner's Apporach, 2014
2. Ian Sommerville, Software Engineering (10th Edition), 2015
3. https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/?ref=rp
4. https://www.geeksforgeeks.org/software-engineering-prototyping-model/?ref=rp
5. https://www.geeksforgeeks.org/software-engineering-evolutionary-model/
6. https://www.geeksforgeeks.org/software-engineering-spiral-model/
7. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-using-spiral-model/?ref=rp