Proses Pengembangan Perangkat Lunak (2)

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi

Politeknik Negeri Malang

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

Outline

- Mengatasi Perubahan
- Pembuatan prototipe perangkat lunak
- Model Prototyping
- Boehm’s spiral model

Tujuan

- Memahami aktivitas dalam proses pengembangan perangkat lunak menggunakan model prototyping dan spiral
- Memahami penggunaan model prototyping dan spiral

Mengatasi Perubahan

Perubahan tidak dapat dihindari di semua proyek perangkat lunak terutama perangkat lunak yang besar.

- Perubahan bisnis → perubahan persyaratan sistem
- Teknologi baru → meningkatkan implementasi
- Perubahan platform → perubahan aplikasi

Perubahan menyebabkan harus dilakukan pengerjaan ulang (*rework*)

Biaya (***cost***) perubahan melingkupi **biaya pengerjaan ulang** (misal. re-analysing requirements) serta **biaya penerapan fungsionalitas** yang baru

Mengurangi Biaya Pengerjaan Ulang

1. Menghindari perubahan
    - Mengantisipasi kemungkinan adanya perubahan sebelum pengerjaan ulang yang signifikan diperlukan.
    - Contoh :
        - Membuat *Prototype* untuk menunjukkan fitur utama ke *customer*.
        - Eksperimen pada prototype
        - Menyempurnakan kebutuhan sebelum berkomitmen pada biaya produksi perangkat lunak yang tinggi

Mengurangi Biaya Pengerjaan Ulang

2. Toleransi perubahan

- Proses dirancang sedemikian rupa sehingga perubahan dapat diakomodasi dengan biaya yang relatif rendah.
- Biasanya melibatkan beberapa bentuk pengembangan bertahap.
- Perubahan yang diusulkan dapat diimplementasikan dalam peningkatan yang belum dikembangkan.
- Jika hal ini tidak memungkinkan, maka hanya satu peningkatan (bagian kecil dari sistem) yang mungkin harus diubah untuk memasukkan perubahan tersebut)

Model Prototyping

- Pendekatan yang secara langsung **mendemonstrasikan bagaimana** sebuah **perangkat lunak** akan **bekerja dalam lingkungannya** sebelum tahapan konstruksi aktual dilakukan.

Software prototyping

Prototype : Versi awal dari system digunakan untuk menunjukkan konsep atau proses kerja dari sistem

Prototype : Bukan Produk final

Prototype dapat digunakan pada :

- *requirements engineering*: membantu elisitasi dan validasi kebutuhan sistem/.
- *design processes*: eksplorasi bagian tertentu dari software dan mendukung desain UI.

Manfaat pembuatan prototipe

- Peningkatan kegunaan sistem.
- Kecocokan yang lebih dekat dengan kebutuhan nyata pengguna.
- Peningkatan kualitas desain.
- Peningkatan pemeliharaan.
- Mengurangi upaya pengembangan.

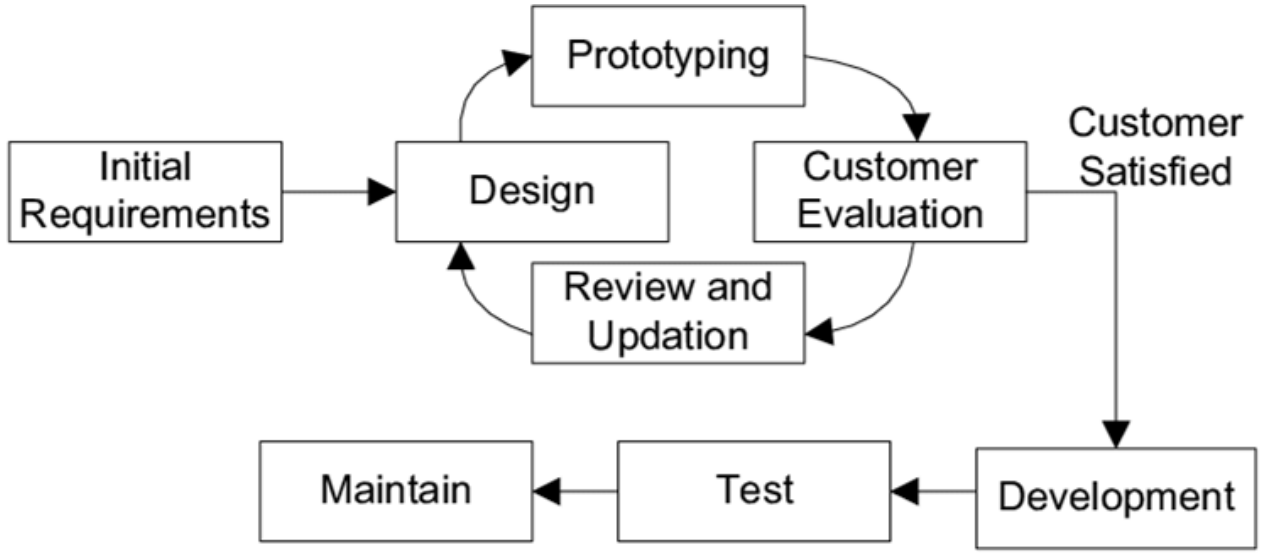
Tahapan - Model Prototyping

1.Initial Requirements

- Klien dan developer bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

2.Design

- Pada tahap ini dilakukan **penerjemahan dari keperluan atau data** yang telah dianalisis **ke dalam bentuk yang mudah dimengerti oleh user**.

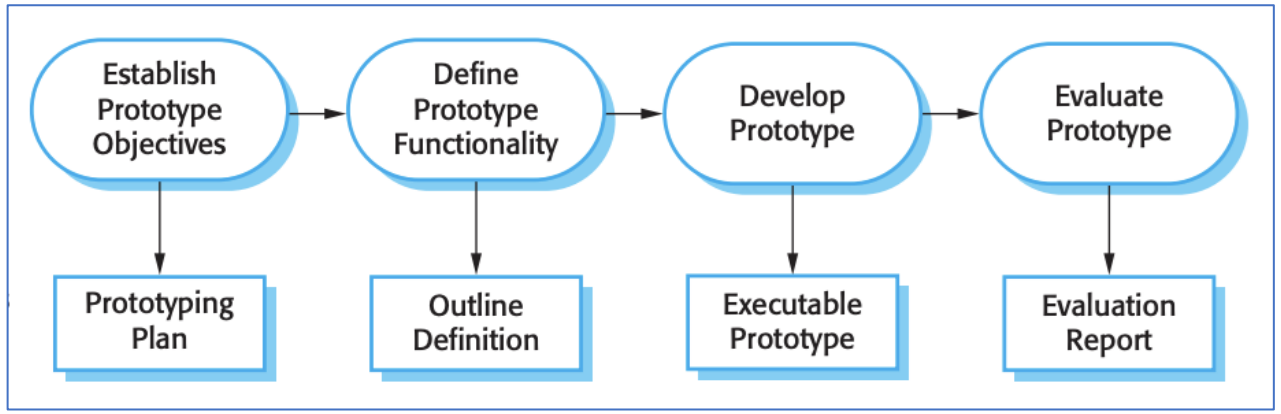


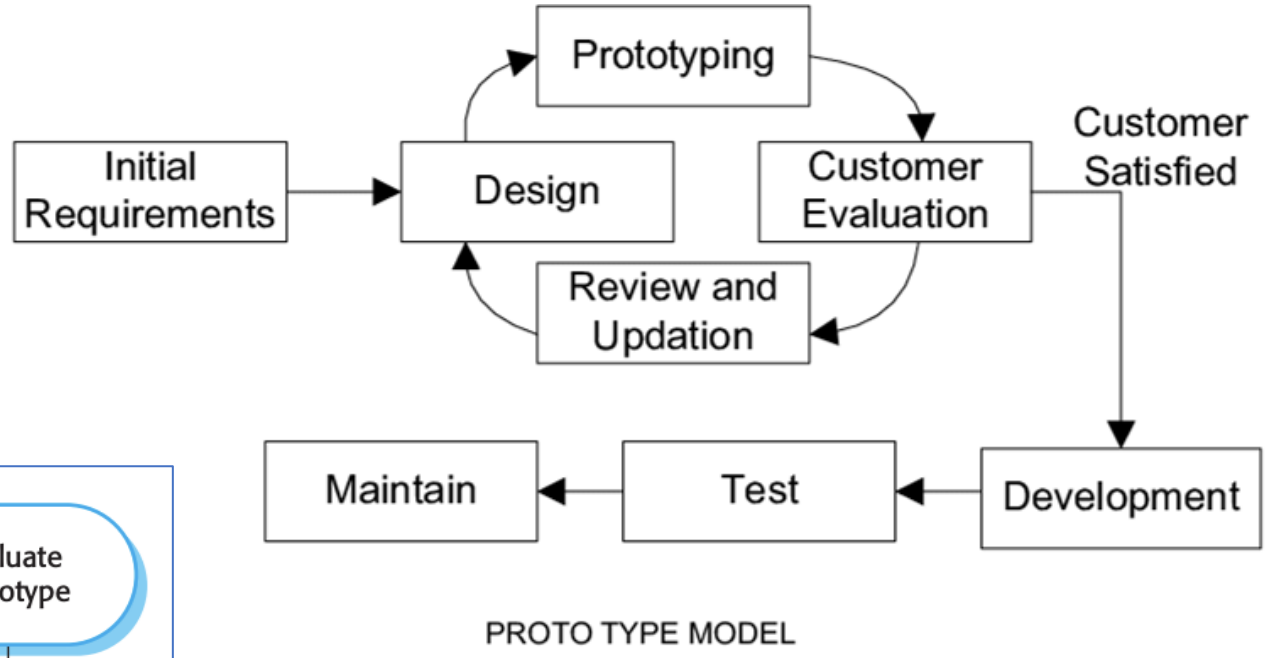
PROTO TYPE MODEL

Tahapan - Model Prototyping

3.Prototyping

- Membangun prototyping dengan **membuat perancangan sementara yang berfokus pada penyajian kepada klien**, misalnya dengan membuat input dan format output.



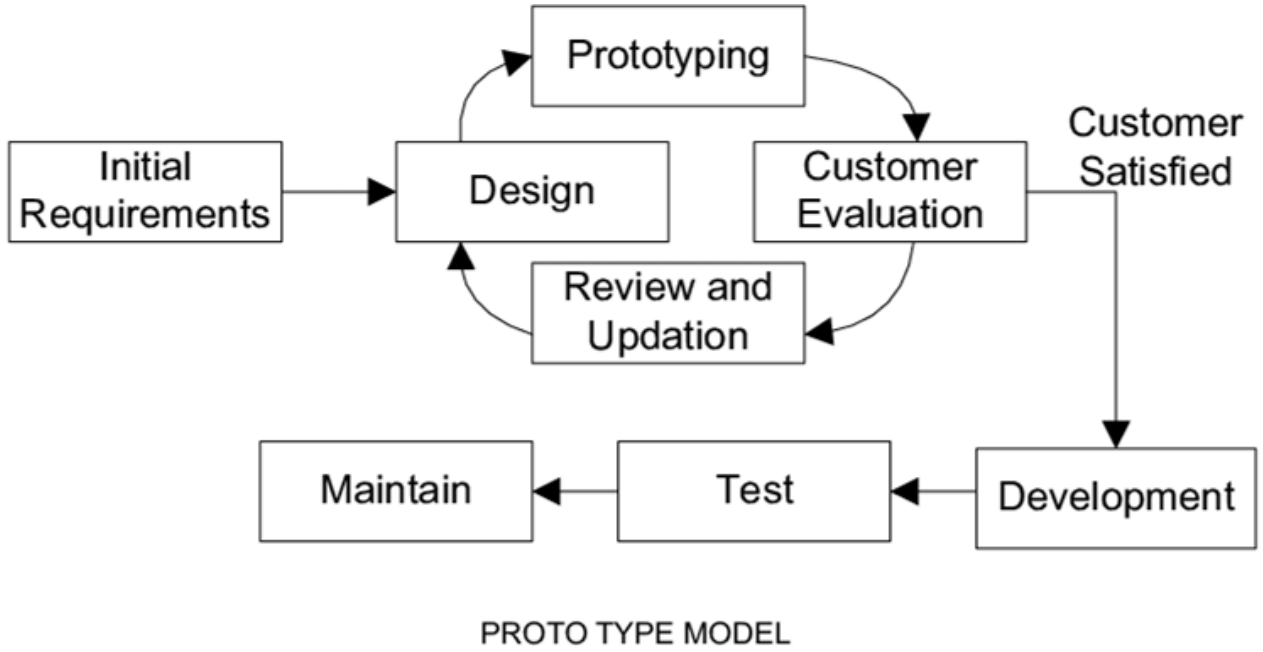


PROTO TYPE MODEL

Tahapan - Model Prototyping

3.Prototyping

- Fitur Terpisah: Jika sistem memiliki beberapa fitur berbeda, beberapa prototipe bisa dikembangkan untuk masing-masing fitur.
- Eksplorasi Alternatif: Kadang kala, lebih dari satu prototipe dibuat untuk mengeksplorasi solusi desain yang berbeda.
- Iterasi Berdampingan: Pengembang bisa menghasilkan banyak iterasi dari prototipe untuk mengevaluasi perubahan pada setiap iterasi berdasarkan umpan balik yang diterima dari pengguna.

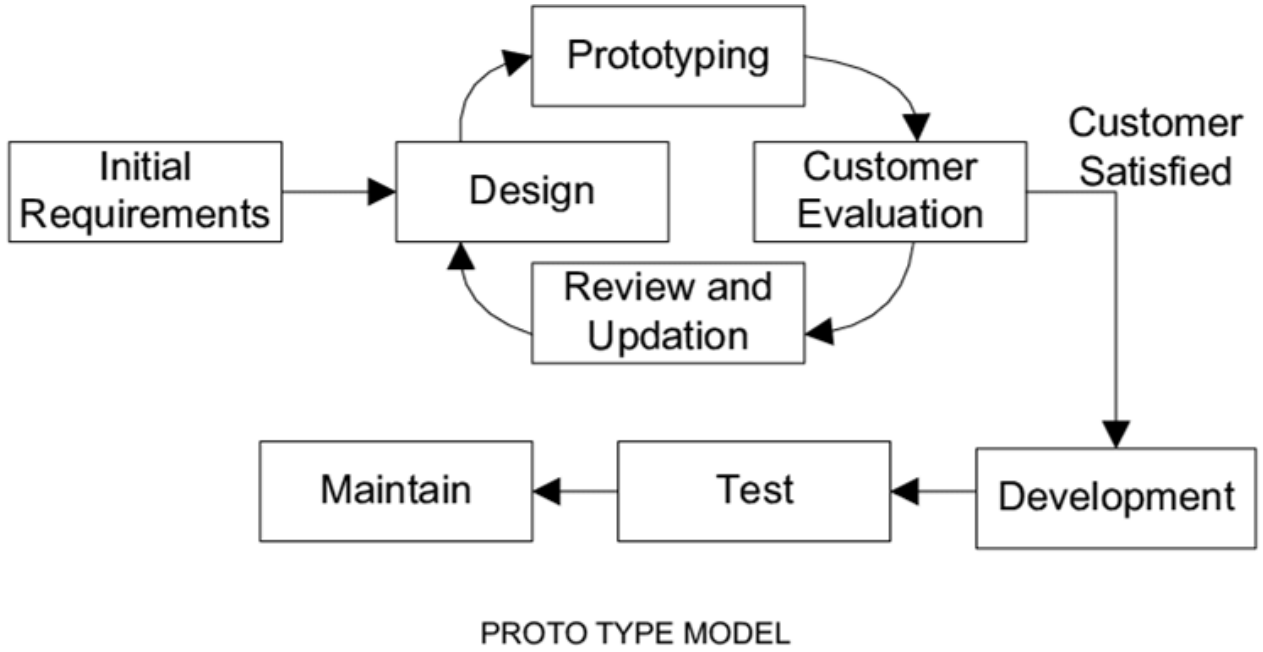


PROTO TYPE MODEL

Tahapan - Model Prototyping

4.Customer Evaluation

- Evaluasi ini dilakukan oleh klien, **apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan klien**.
- Jika sudah sesuai, maka proses dilanjutkan ke tahap selanjutnya.
- Namun jika tidak, *prototyping* direvisi dengan mengulang langkah-langkah sebelumnya.

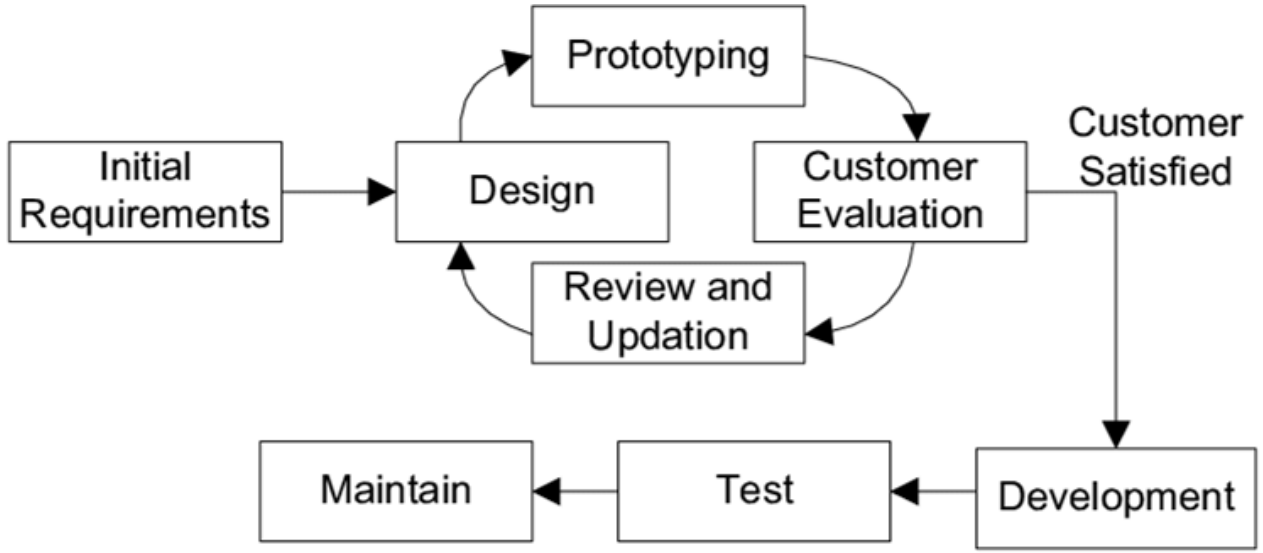


PROTO TYPE MODEL

Tahapan - Model Prototyping

5.Development

- Pada tahap ini, *prototyping* yang sudah disepakati **diterjemahkan ke dalam bahasa pemrograman** yang sesuai.

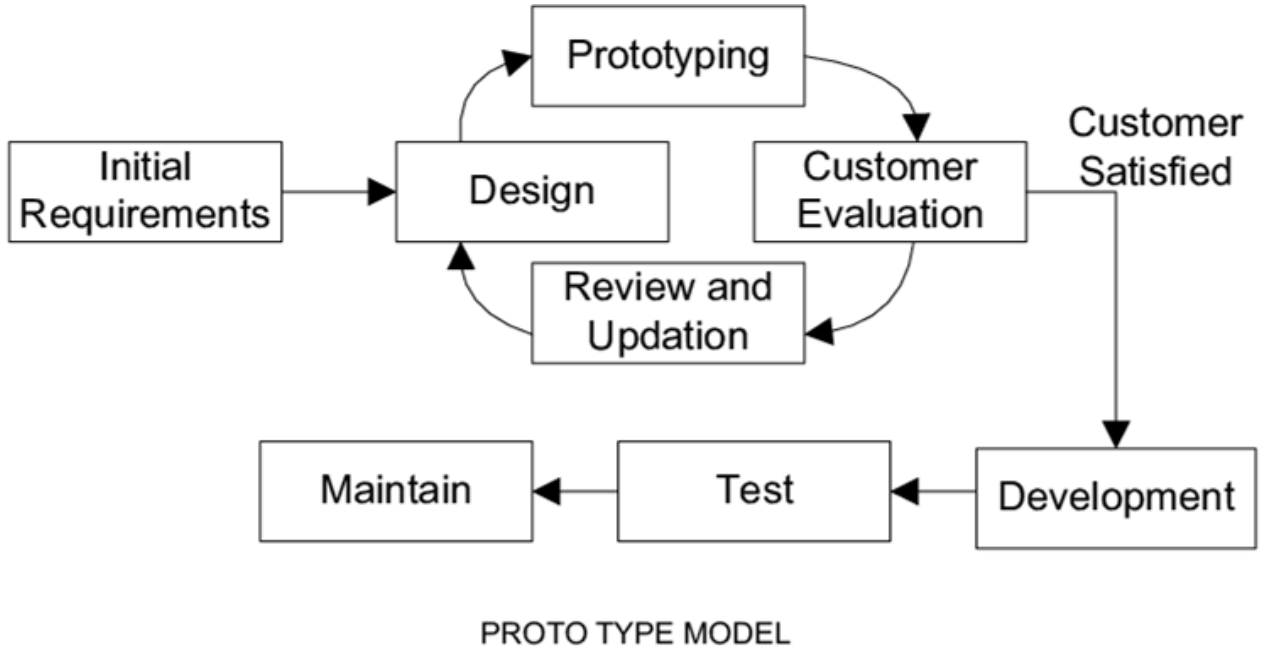


PROTO TYPE MODEL

Tahapan - Model Prototyping

6.Test

- Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, selanjutnya **dilakukan proses pengujian.**

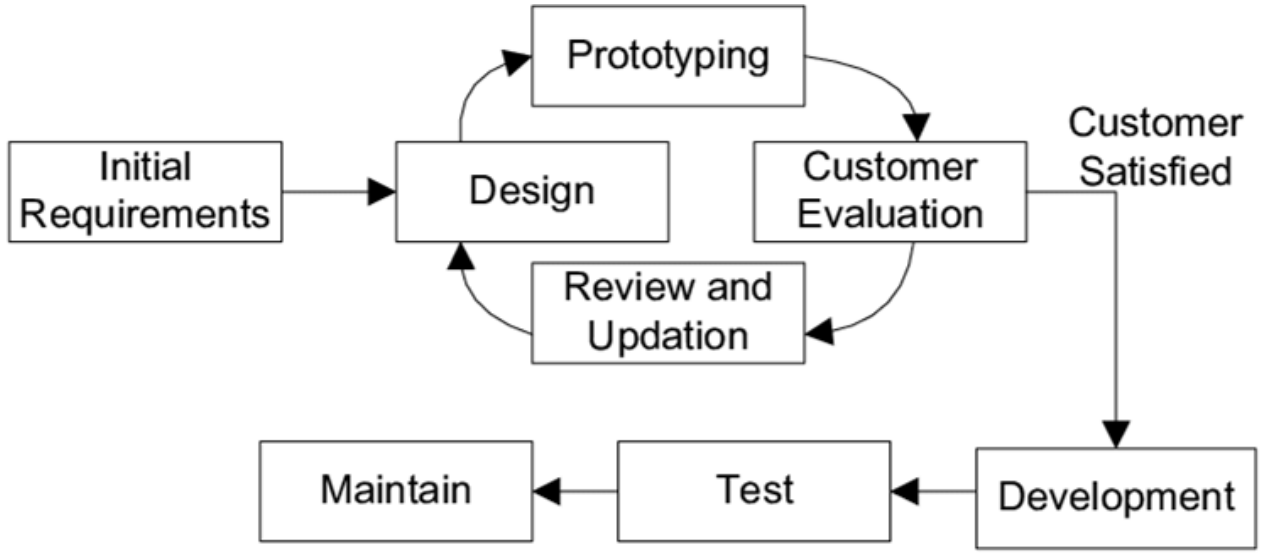


PROTO TYPE MODEL

Tahapan - Model Prototyping

7.Maintain

- Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan, selain itu juga **dilakukan pemeliharaan yang meliputi perbaikan kesalahan** yang tidak ditemukan pada langkah sebelumnya.



PROTO TYPE MODEL

Kelebihan & Kekurangan Model Prototyping

Kelebihan

- Meningkatkan keterlibatan *user*.
- Mengurangi waktu dan biaya.
- Kesalahan yang terjadi dalam *prototyping* dapat dideteksi lebih dini.
- Penerapan menjadi lebih mudah karena *user* mengetahui apa yang diharapkannya.

Kekurangan

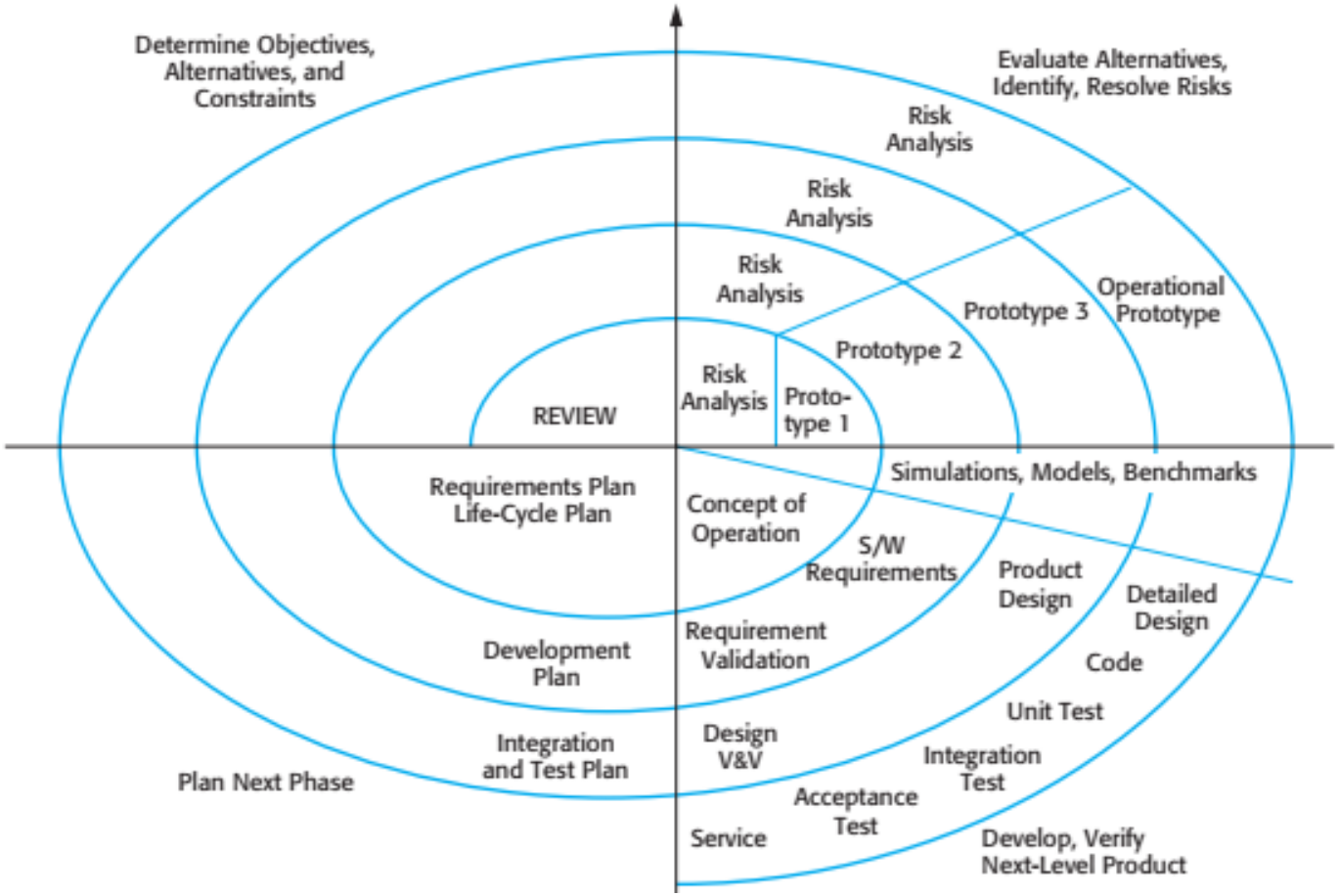
- Proses analisis dan perancangan terlalu singkat.
- Biaya untuk membuat *prototyping* cukup tinggi.
- Biasanya kurang fleksibel dalam menghadapi perubahan.

Situasi Penggunaan Model Prototyping

Metode *prototyping* cocok digunakan untuk:

- proyek yang membutuhkan waktu singkat dan
- *user* mengetahui bagaimana proses pembuatan proyek hingga cara menerapkan proyek tersebut karena antara *developer* dengan *user* terjalin komunikasi yang baik

Boehm's Spiral Model



- Boehm Spiral Model adalah model pengembangan perangkat lunak yang dikembangkan oleh Barry Boehm pada tahun 1986.
- Setiap loop dalam spiral mewakili fase proses perangkat lunak.
- Dengan demikian, loop terdalam berkaitan dengan kelayakan system (*feasibility*), loop berikutnya dengan definisi persyaratan *(requirement),* loop berikutnya dengan desain sistem, dan seterusnya.
- Model ini unik karena fokusnya yang kuat pada identifikasi dan mitigasi risiko. Di setiap putaran, risiko potensial dievaluasi dan ditangani sebelum melanjutkan ke tahap pengembangan berikutnya.

Boehm’s Spiral Model

Setiap loop dalam spiral dibagi menjadi empat sektor:

1. **Objective setting**

   Kebutuhan dan tujuan proyek diidentifikasi. Risiko potensial yang mungkin dihadapi selama pengembangan juga dianalisis.

2. **Risk assessment and reduction**

   Menganalisis Resiko baik **secara teknikal maupun secara manajerial** dan mengambil langkah untuk mengurangi atau menghilangkannya.

3. **Development and validation**

   Setelah risiko terkendali, produk dikembangkan sesuai kebutuhan yang sudah diidentifikasi. Prototipe atau versi awal produk sering kali dibuat di sini.

4. **Planning**

   Proyek ditinjau dan melakukan perencanaan fase berikutnya.

Kelebihan dan Kekurangan

Kelebihan:

- *User* dan developer bisa memahami dengan baik perangkat lunak yang dibangun.
- Estimasi (perkiraan) menjadi lebih realistik seiring berjalannya proyek karena masalah ditemukan sesegera mungkin.
- *Software engineer* bisa bekerja lebih cepat.

Kekurangan:

- Membutuhkan waktu yang lama dan biaya yang besar.
- Membutuhkan rencana jangka panjang yang baik.
- Mempunyai resiko yang harus dipertimbangkan ulang oleh klien dan developer.

Situasi Penggunaan Model Spiral

Model *spiral* cocok digunakan untuk mengembangkan **sistem perangkat lunak berskala besar** karena memiliki proses analisis resiko yang dapat sangat meminimalisir resiko yang mungkin terjadi dan dengan target waktu dan biaya yang tidak terlalu mengikat.

Model *spiral* memungkinkan *developer* untuk menggunakan *prototype* pada setiap tahap untuk mengurangi resiko.

Contoh Penggunaan Model Spiral

- **Pengembangan Sistem Perangkat Lunak Perbankan**: Sistem perbankan umumnya kompleks dan berisiko tinggi karena menyangkut keamanan data dan integritas transaksi. Spiral model memastikan setiap elemen dievaluasi secara rinci sebelum sistem dikembangkan sepenuhnya.
- **Proyek Sistem Militer atau Kedirgantaraan**: Sistem ini melibatkan banyak risiko karena harus memastikan keselamatan, keandalan, dan kinerja optimal. Spiral model cocok karena memungkinkan perencanaan, prototyping, dan pengujian berulang.
- **Sistem yang Memerlukan Pengembangan Jangka Panjang**: Untuk sistem yang membutuhkan banyak peningkatan atau iterasi, seperti pengembangan perangkat lunak untuk operasi skala besar, spiral model memberikan fleksibilitas untuk penyesuaian terus-menerus.

Any questions?