

Proses Pengembangan Perangkat Lunak (2)

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi
Politeknik Negeri Malang

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

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

Mengurangi biaya pengerjaan ulang

- Menghindari perubahan : Mengantisipasi kemungkinan adanya perubahan sebelum pengerjaan ulang yang signifikan diperlukan.
 - Contoh : Membuat *Prototype* untuk menunjukkan fitur utama ke *customer*.

Software prototyping

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

Proptotype : Bukan Produk final

Protorype dapat digunakan pada :

- *requirements engineering*
- *design processes*
- *design processes*

Manfaat pembuatan prototipe



Improved system usability.



A closer match to users' real needs.



Improved design quality.



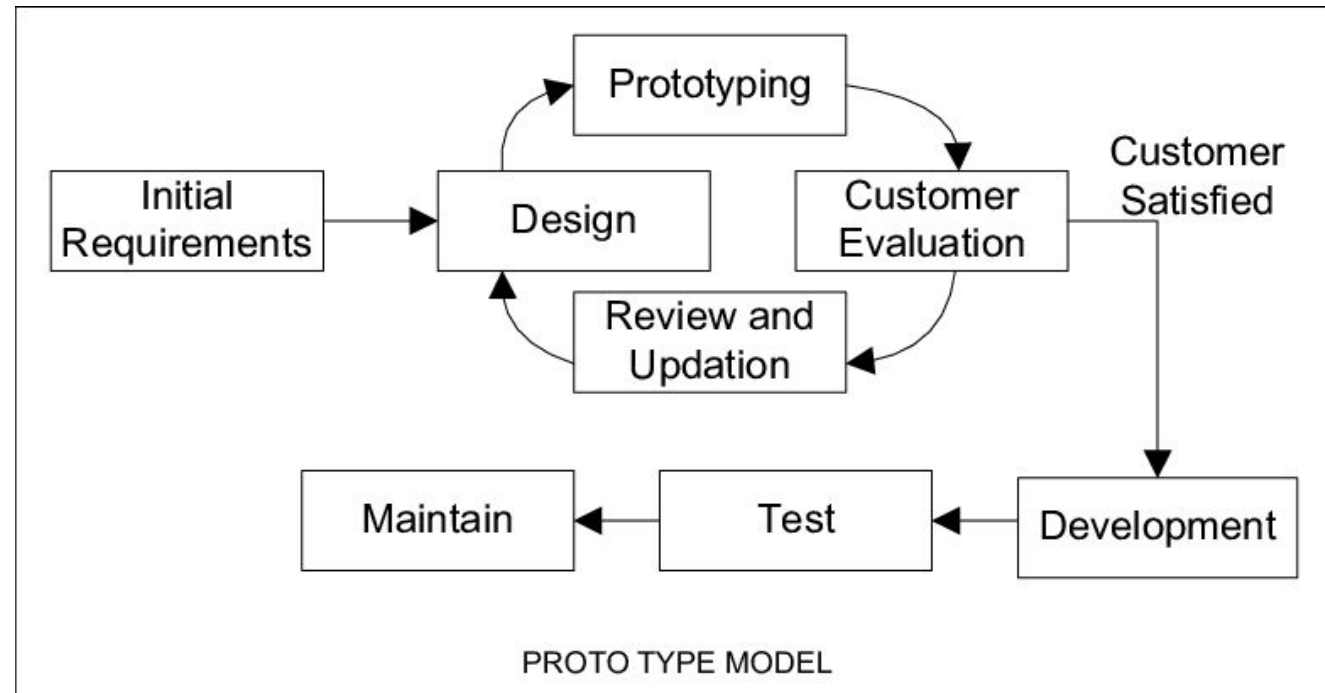
Improved maintainability.



Reduced development effort.

Model Prototyping

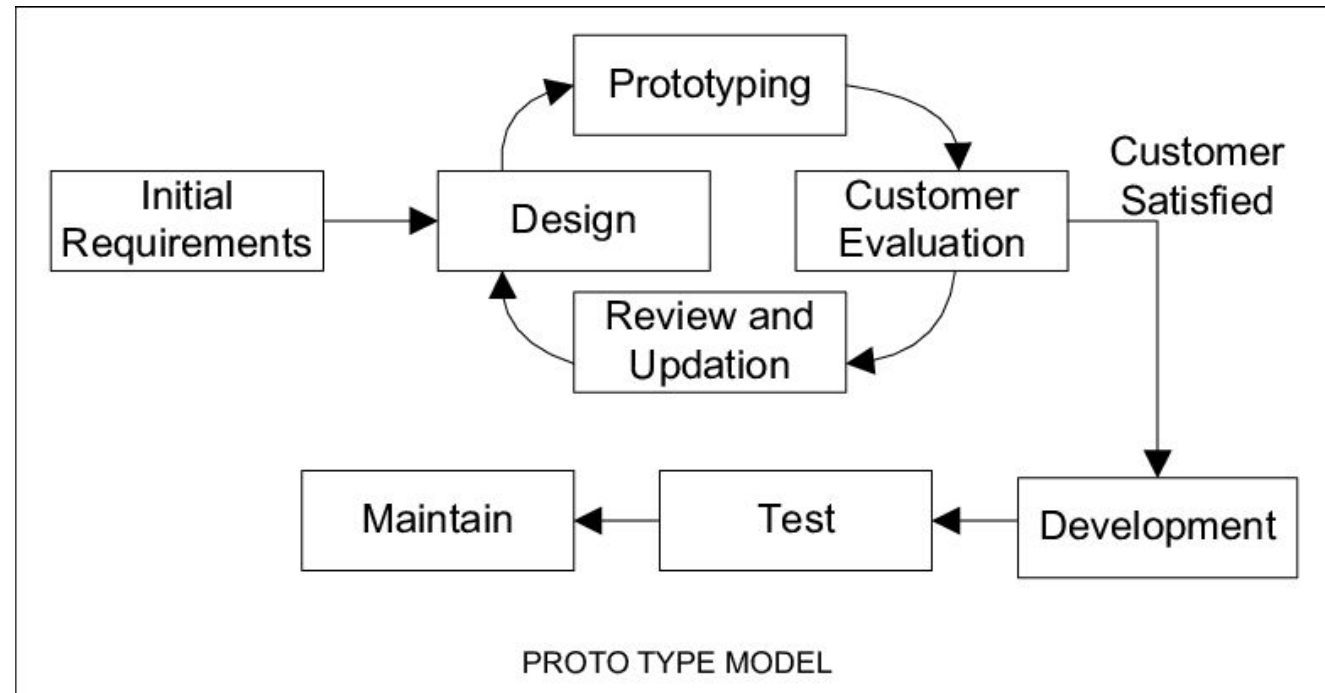
- Pendekatan yang secara langsung **mendemonstrasikan bagaimana sebuah perangkat lunak akan bekerja dalam lingkungannya** sebelum tahapan konstruksi aktual dilakukan.



Tahapan - Model Prototyping

1. *Initial Requirements*

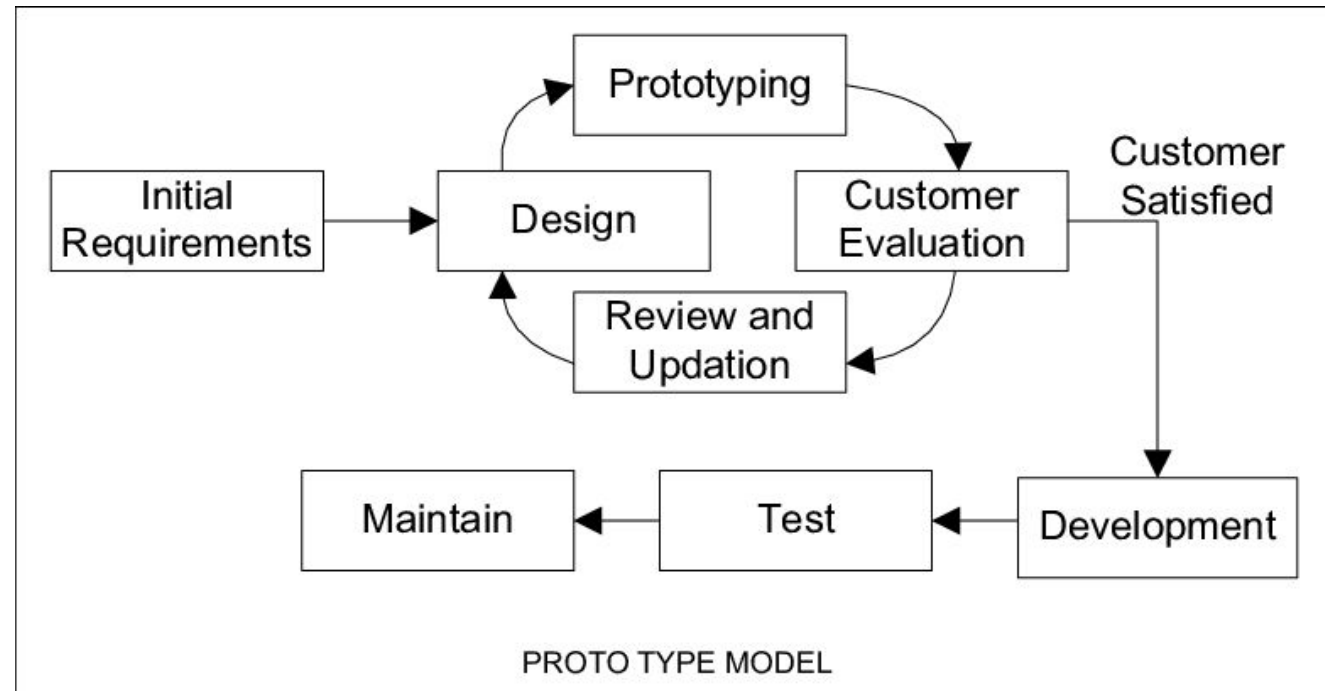
- **Klien dan *developer* bersama-sama mendefinisikan format seluruh perangkat lunak**, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.



Tahapan - Model Prototyping

2. Design

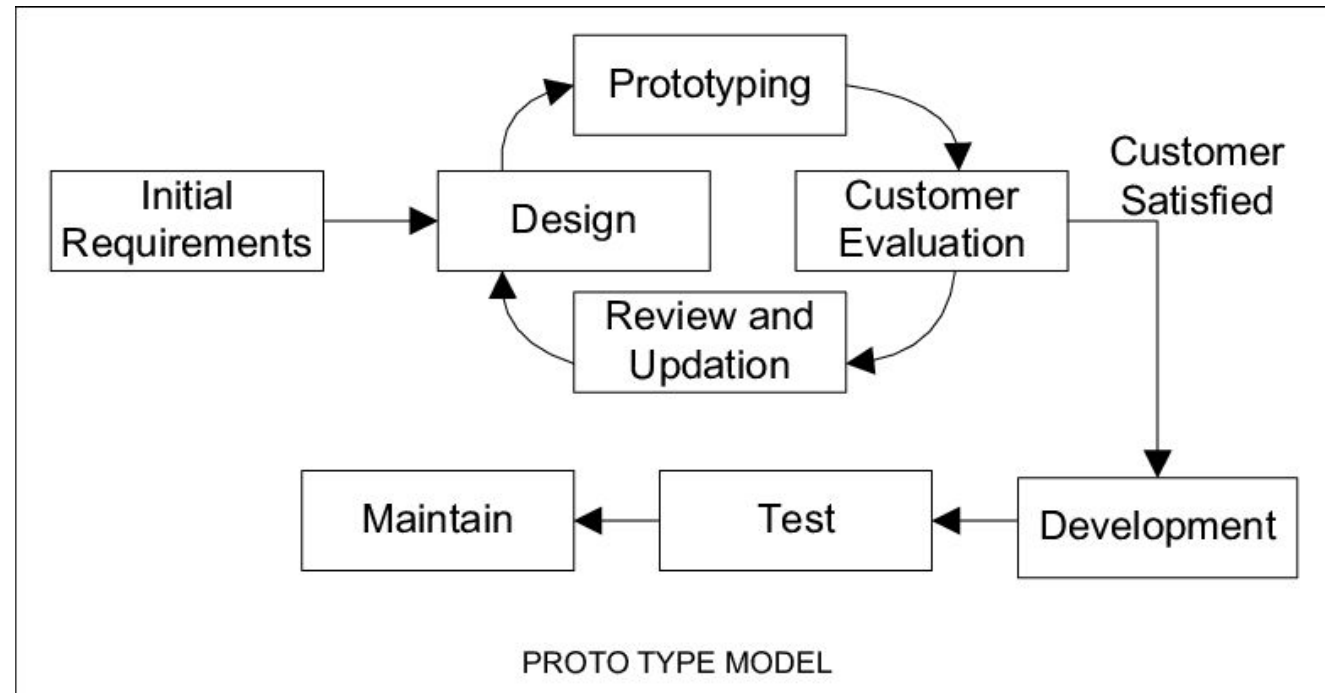
- Pada tahap ini dilakukan **penerjemahan dari keperluan atau data** yang telah dianalisis **ke dalam bentuk yang mudah dimengerti oleh user.**



Tahapan - Model Prototyping

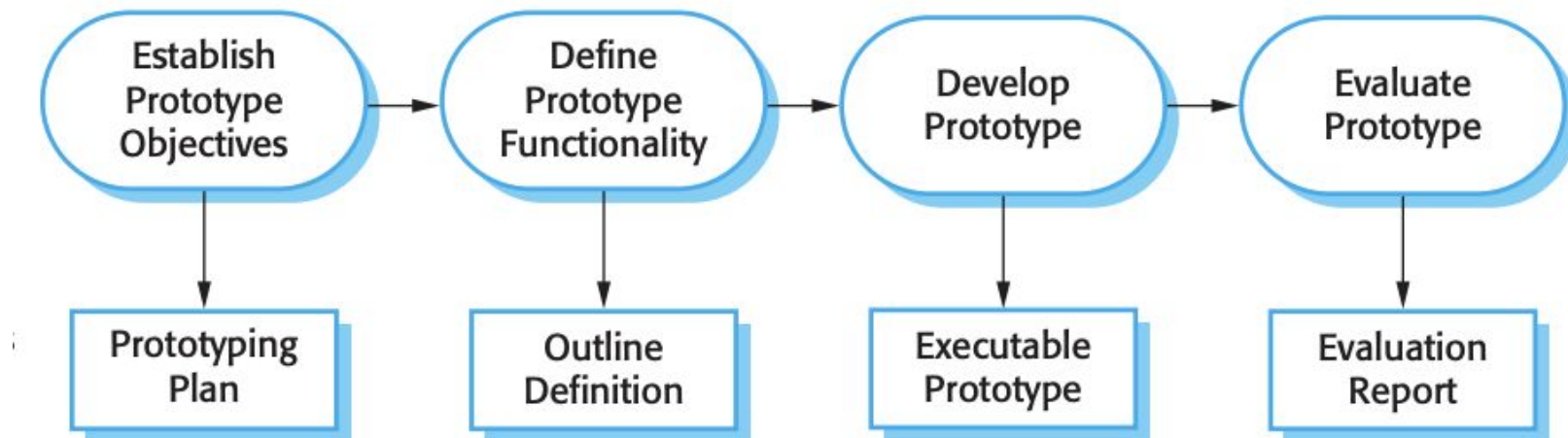
3. Prototyping

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



Tahapan - Model Prototyping

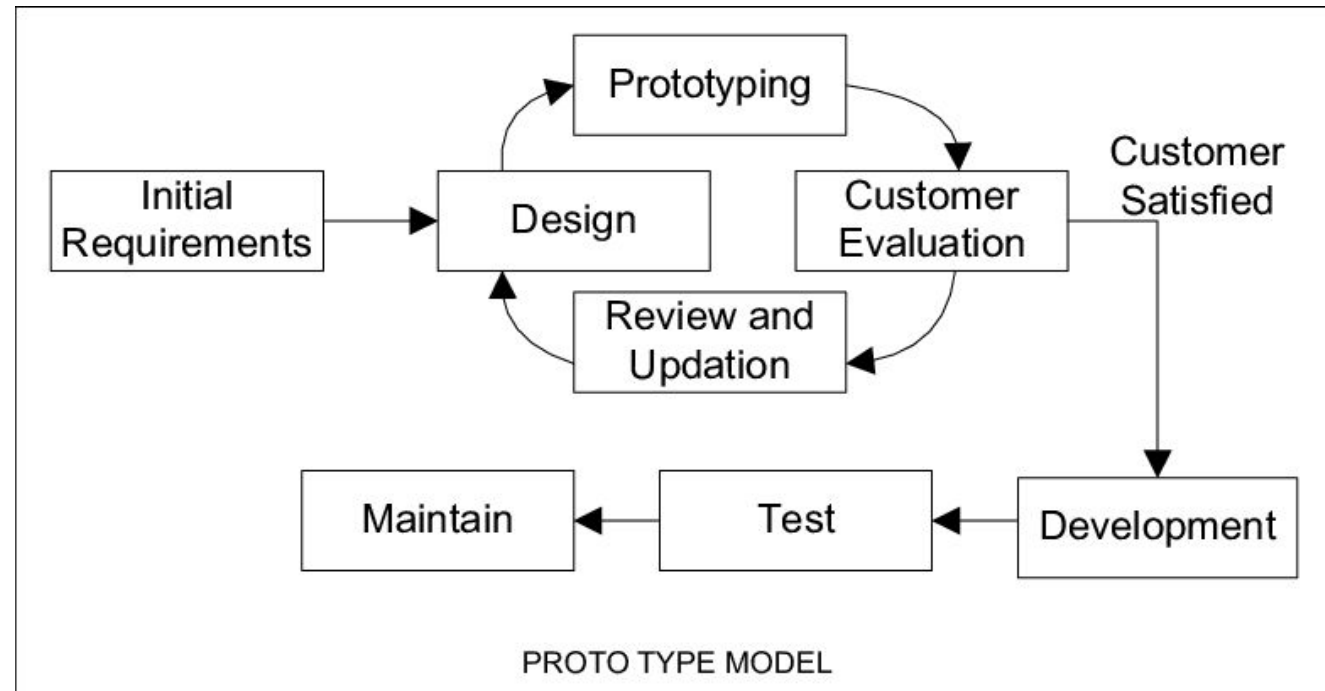
3. *Proses pengembangan Prototype*



Tahapan - Model Prototyping

4. Customer Evaluation

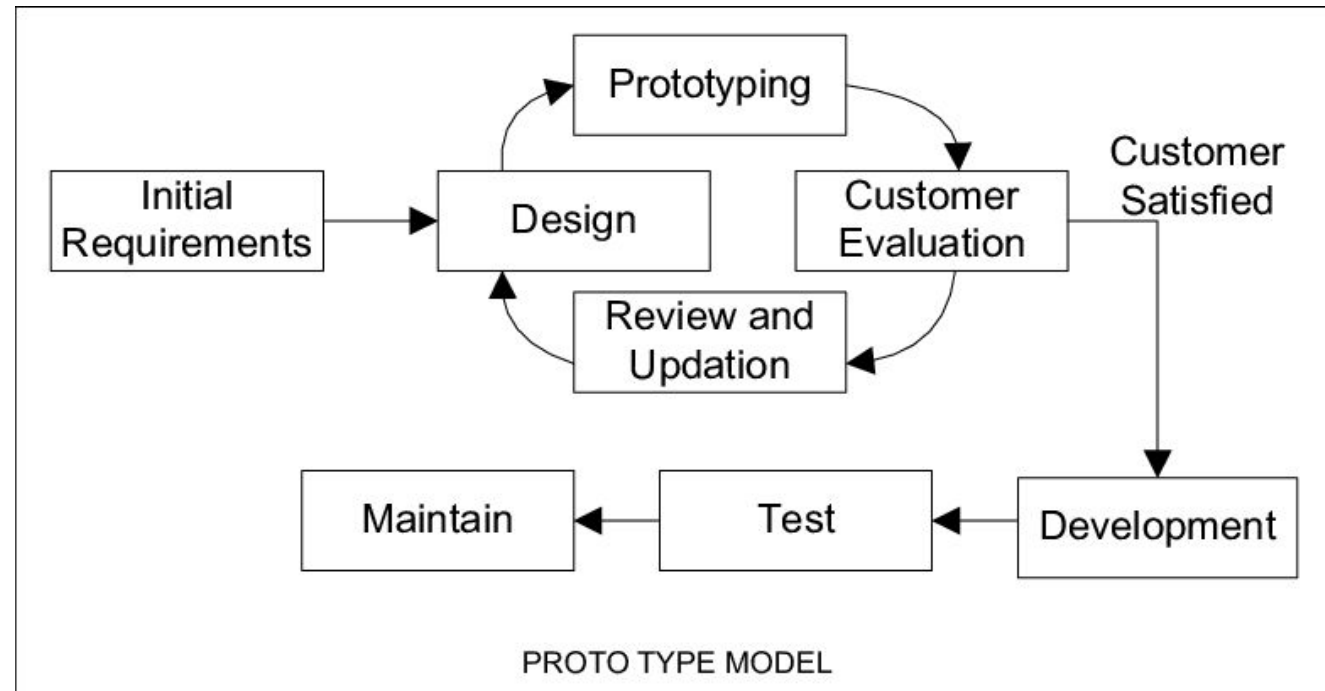
- Evaluasi ini dilakukan oleh klien, **apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan klien**. Jika sudah sesuai, maka proses dilanjutkan ke tahap selanjutnya. Namun jika tidak, *prototyping* direvisi dengan mengulang langkah-langkah sebelumnya.



Tahapan - Model Prototyping

5. Development

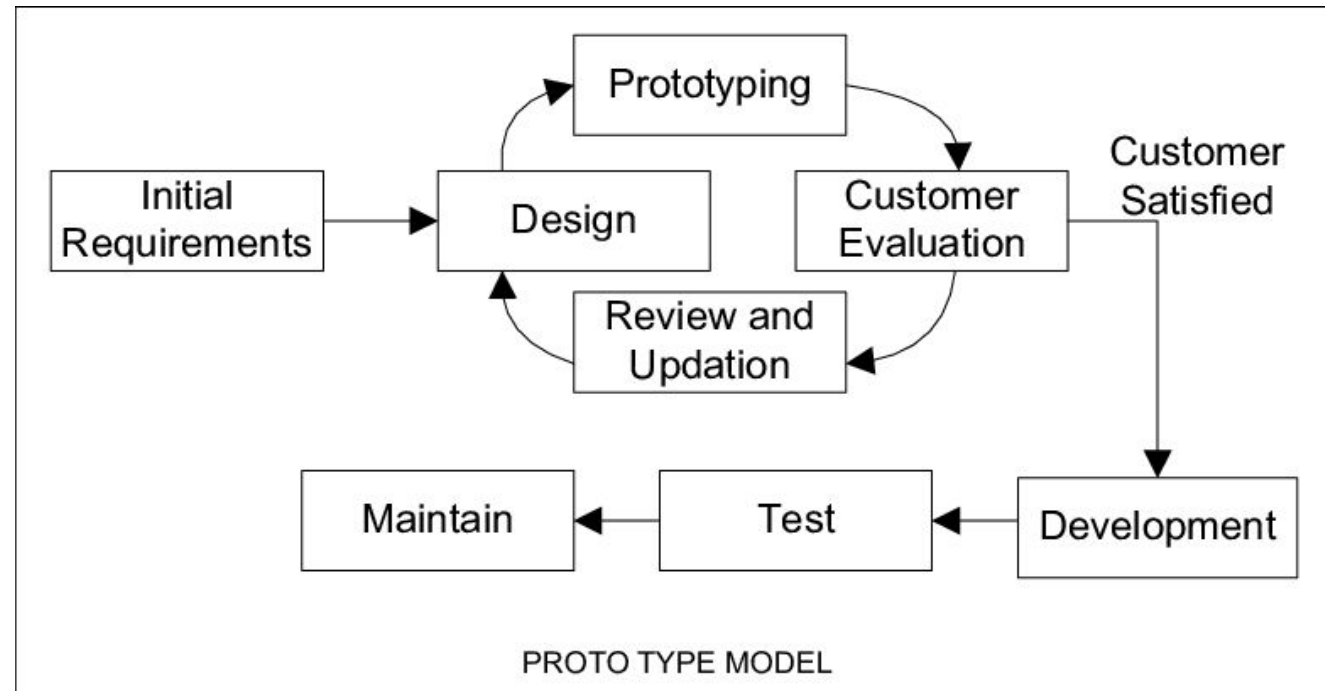
- Pada tahap ini, *prototyping* yang sudah disepakati **diterjemahkan ke dalam bahasa pemrograman** yang sesuai.



Tahapan - Model Prototyping

6. Test

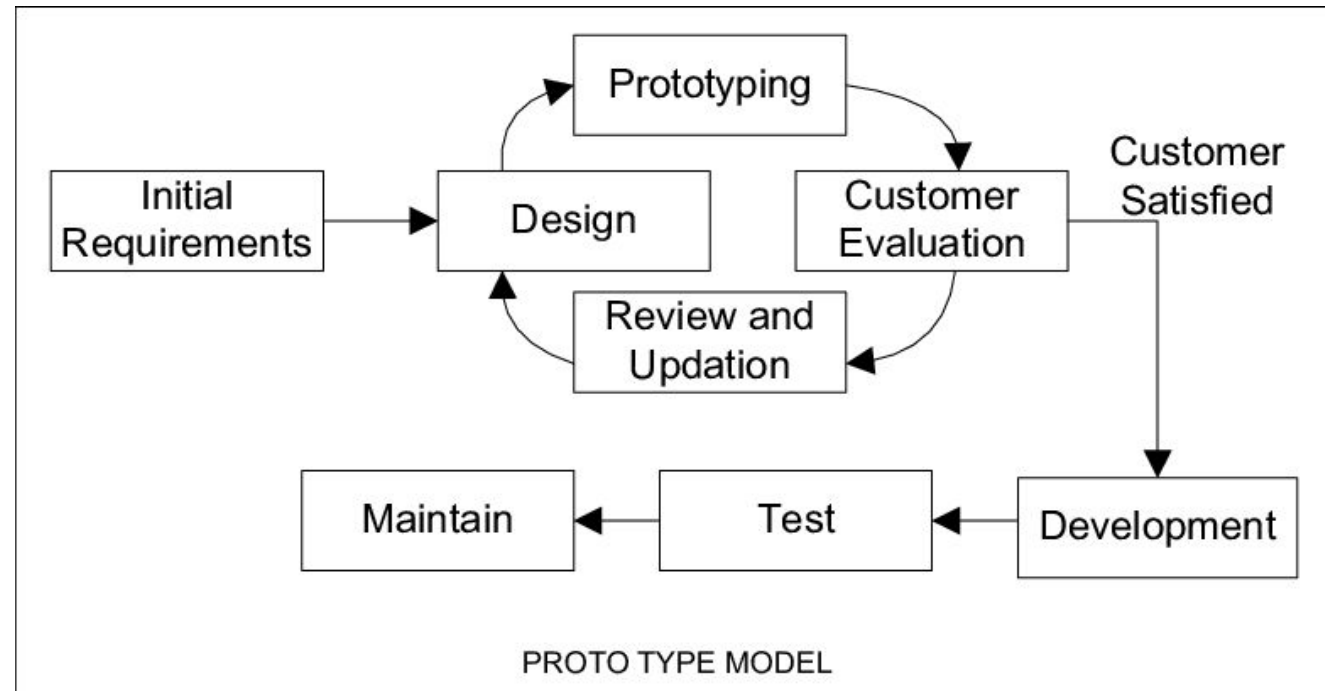
- Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, selanjutnya **dilakukan proses pengujian.**



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.



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

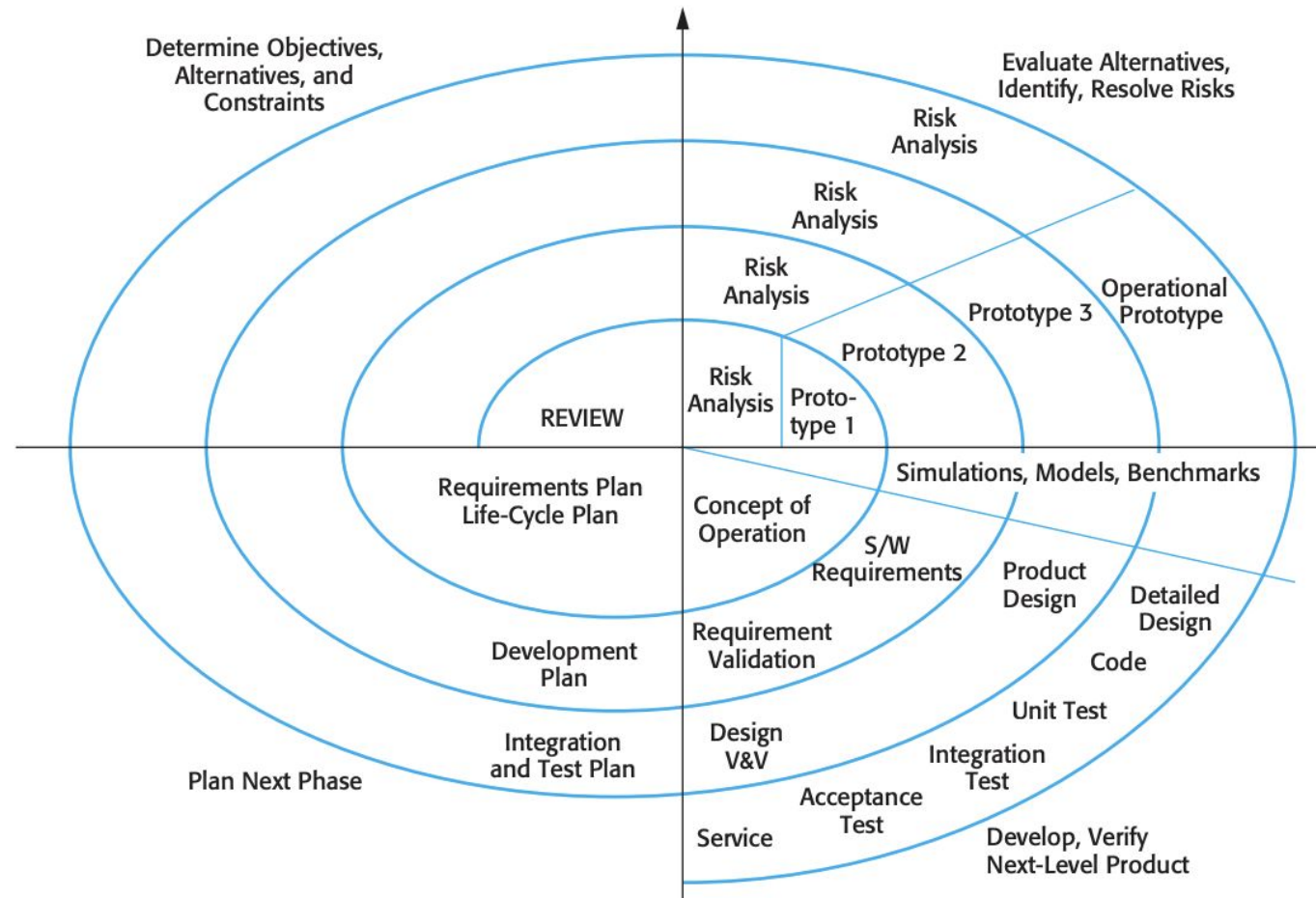
Boehm's spiral model

- Proses direpresentasikan sebagai ***spiral*** bukan sebagai urutan kegiatan
- Setiap ***loop*** dalam spiral mewakili fase dalam proses.
- **Tidak ada fase tetap** seperti spesifikasi atau desain - loop dalam spiral dipilih tergantung pada apa yang dibutuhkan.
- Risiko secara eksplisit dinilai dan diselesaikan selama proses berlangsung.

Boehm's spiral model

- Setiap loop dalam spiral dibagi menjadi empat sector
 - *Objective setting* : Tujuan khusus untuk fase tersebut diidentifikasi.
 - *Risk assessment and reduction* : Menganalisis Resiko baik **secara teknis maupun secara manajerial**
 - *Development and validation* : Pemilihan **model pengembangan** untuk system, misal : jika resiko antar muka lebih dominan ☐ menggunakan model prototyping, jika resiko integrasi sub system lebih dominan ☐ gunakan model waterfall
 - *Planning* : Proyek ditinjau dan melakukan perencanaan fase berikutnya.

Boehm's spiral model



Kelebihan dan Kekurangan

Kelebihan :

User dan developer bisa memahami dengan baik perangkat lunak yang dibangun.

Estimasi (perkiraan) menjadi lebih realistis 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.



Any questions?