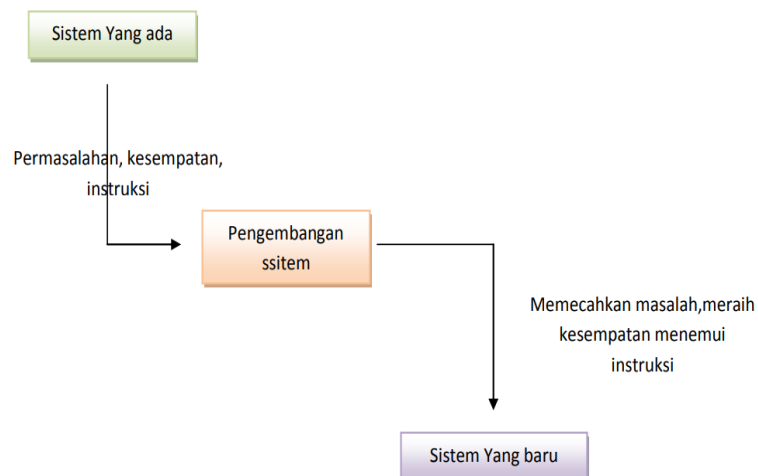


BAB IV

PENDEKATAN-PENDEKATAN PENGEMBANGAN SISTEM

Disamping metode tradisional SDLC, ada beberapa metode yang dikembangkan untuk melengkapi kelemahan-kelemahan yang ada dalam metode SDLC. Metode metode itu antara lain:

- o Structured analysis and structured design
- o Object oriented analysis and design
- o Prototyping
- o Joint Application Design (JAD)
- o Participatory design

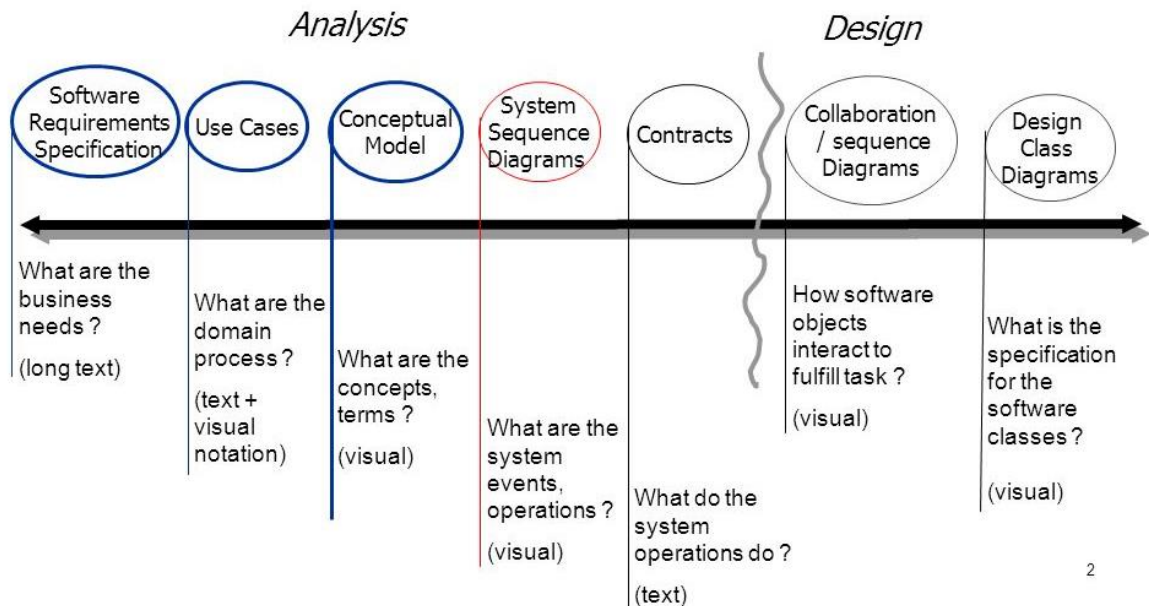


Gambar 4.1 Proses Pengembangan Sistem

4.1 Structured Analysis dan Structured Design

Pendekatan ini lebih berfokus pada bagaimana mereduksi waktu dan maintenance dalam pengembangan sistem. Pendekatan ini juga langsumng memngintegrasikan perubahan jika diperlukan.

4.2 Object Oriented Analysis and Design (OOAD)

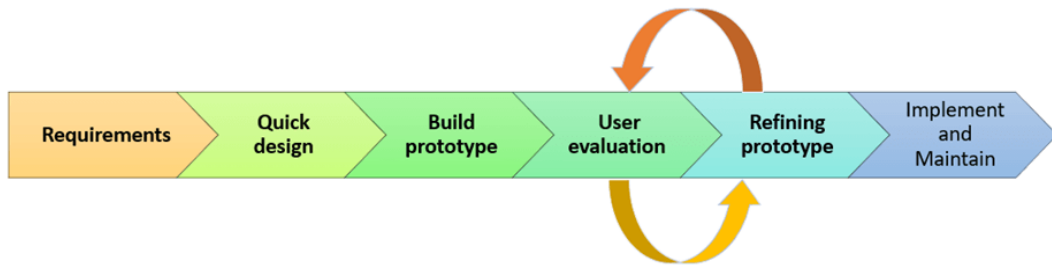


Gambar 4.2 Object Oriented Analysis and Design

Pendekatan baru untuk pengembangan sistem, sering disebut sebagai pendekatan ketiga setelah pendekatan yang berorientasi data dan berorientasi proses. OOAD adalah metode pengembangan sistem yang lebih menekankan pada objek dibandingkan dengan data atau proses. Ada beberapa ciri khas dari pendekatan ini yaitu object, Inheritance dan object class

- Object adalah struktur yang mengenkapsulasi atribut dan metode yang beroperasi berdasarkan atribut-atribut tadi.
- Objek adalah abstraksi dari benda nyata dimana data dan proses diletakkan bersama untuk memodelkan struktur dan perilaku dari objek dunia nyata.
- Object class adalah sekumpulan objek yang berbagi struktur yang sama dan perilaku yang sama. Inheritance. Properti yang muncul ketika tipe entitas atau object class disusun secara hirarki dan setiap tipe entitas atau object class menerima atau mewarisi atribut dan metode dari pendahulunya.

4.3. Prototyping



Gambar 4.3 Prototyping Methodology

Prototyping adalah proses iterative dalam pengembangan sistem dimana requirement diubah ke dalam sistem yang bekerja (working system) yang secara terus menerus diperbaiki melalui kerjasama antara user dan analis. Prototype juga bisa dibangun melalui beberapa tool pengembangan untuk menyederhanakan proses. Prototyping merupakan bentuk dari Rapid Application Development (RAD).

Beberapa kerugian RAD:

- RAD mungkin mengesampingkan prinsip-prinsip rekayasa perangkat lunak
- Menghasilkan inkonsistensi pada modul-modul sistem
- Tidak cocok dengan standar
- Kekurangan prinsip reusability komponen

Prototype methodology

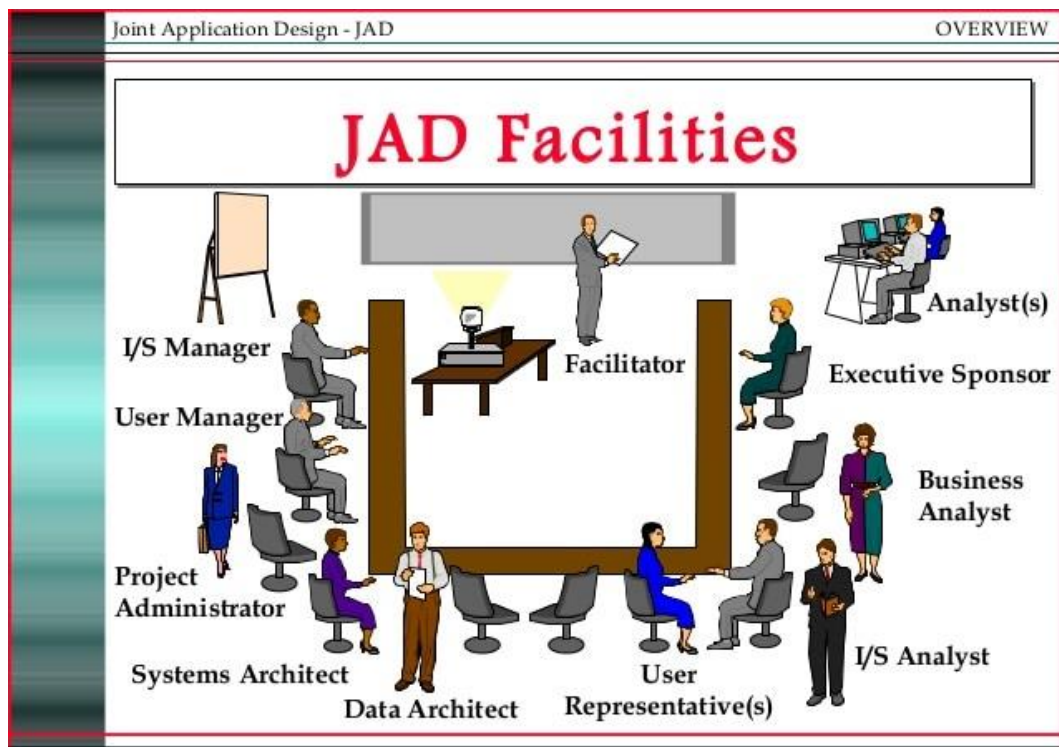
- Analis bekerja dengan tim untuk mengidentifikasi requirement awal untuk sistem
- Analis kemudian membangun prototype. Ketika sebuah prototype telah selesai. User bekerja dengan prototype itu dan menyampaikan pada analis apa yang mereka sukai dan yang tidak mereka sukai.
- Analis kemudian menggunakan feedback ini untuk memperbaiki prototype
- Versi baru diberikan kembali ke user
- Ulangi langkah-langkah tersebut sampai user merasa puas

Keuntungan prototype

- Prototype melibatkan user dalam analisa dan desain
- Punya kemampuan menangkap requirement secara konkret daripada secara abstrak
- Untuk digunakan secara standalone
- Digunakan untuk memperluas SDLC

4.4 Joint Application Design (JAD)

Pada akhir 1970 an personil pengembangan sistem di IBM mengembangkan proses baru untuk mengumpulkan requiremen SI dan mereview desain dengan nama JAD. JAD adalah proses terstruktur dimana user, manager dan analis bekerja bersamasama selama beberapa hari dalam 1 pertemuan bersama untuk mengumpulkan requiremen sistem yang akan dibangun.



4.5 Participatory Design

End user dilibatkan dalam pengembangan sistem dalam satu meja untuk persetujuan tentang sistem requirement dan system desain.

Pada perkembangannya desain sistem banyak disupport oleh penggunaan software dan teknologi baru. Analisis mengandalkan tool dengan tujuan :

- Meningkatkan produktifitas
- Berkomunikasi lebih efektif dengan user
- Mengintegrasikan pekerjaan yang telah dilaksanakan dari awal pengembangan sampai akhir.

Contoh-contoh tool yang digunakan adalah :

- Computer-Aided Systems Engineering (CASE -tools)
- Application Development Environments (ADE -tools)
- Process and Project Managers

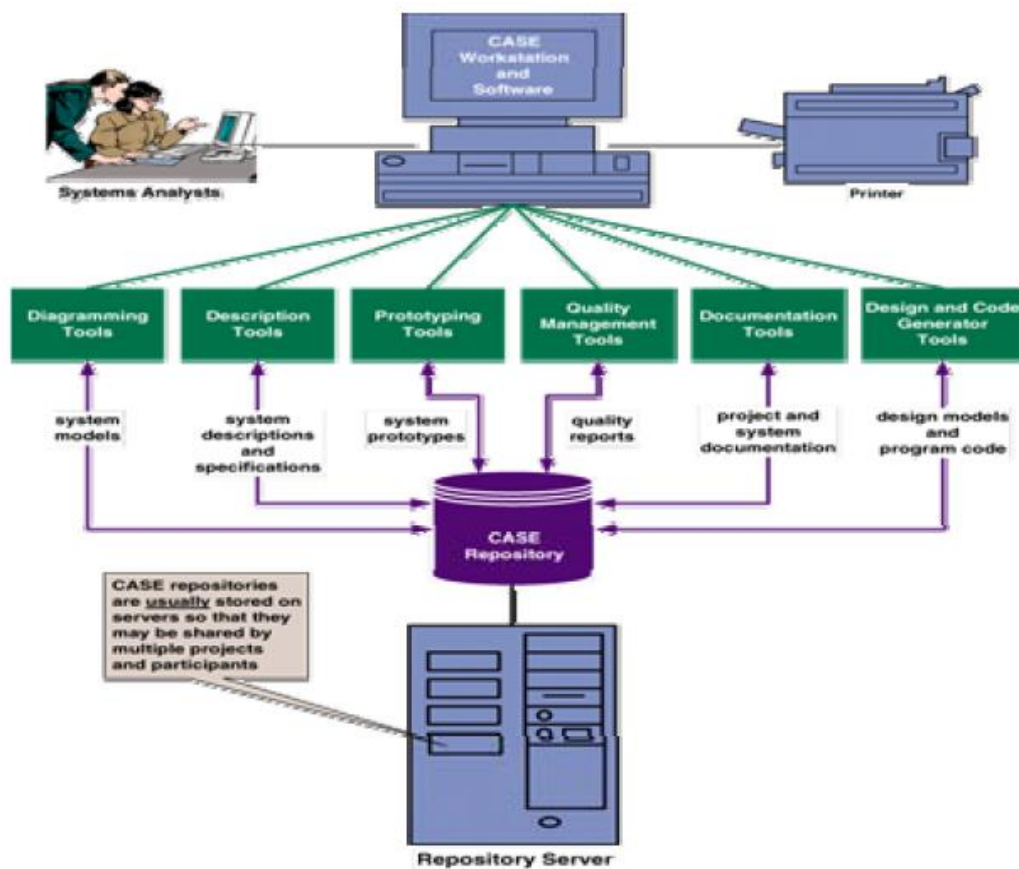
4.6 Meningkatkan Produktifitas Pengembangan Sistem Informasi

Teknologi komputer dapat digunakan untuk meningkatkan produktifitas. CASE tool sebagai contoh, menyediakan lebih banyak produktifitas untuk meningkatkan kemampuan seperti pengembangan kode, tool diagramming dan pendesainan layer dan laporan.

Computer-Aided Systems Engineering: CASE tools

Case adalah software yang digunakan untuk mengotomasi atau mendukung penggambaran dan analisa dari model sistem dan menyediakan translasi dari model sisystem ke sistem aplikasi.

Berikut ini gambaran arsitektur CASE



Gambar 4.1 Arsitektur CASE Tool