

Rekayasa Perangkat Lunak (RPL) adalah sebuah disiplin ilmu yang mempelajari cara-cara pengembangan, pemeliharaan, dan pembangunan kembali perangkat lunak dengan menerapkan prinsip-prinsip rekayasa. Tujuan utamanya adalah untuk menghasilkan perangkat lunak yang bekerja dengan efisien dan efektif bagi pengguna.

Ciri-ciri Perangkat Lunak yang Baik

- **Efisiensi dan Efektivitas:** Perangkat lunak yang baik harus bekerja dengan efisien dan efektif, memenuhi kebutuhan pengguna dengan baik.
- **Kualitas Tinggi:** Perangkat lunak yang berkualitas tinggi harus bebas dari kesalahan dan memenuhi standar kualitas yang tinggi.
- **Pembaruan Berkala:** Perangkat lunak harus dapat diperbarui dan ditingkatkan secara berkala untuk memastikan kinerja yang optimal.
- **Kompatibilitas:** Perangkat lunak harus dapat berjalan dengan baik pada berbagai platform dan sistem operasi.
- **Kemampuan Adaptasi:** Perangkat lunak harus dapat beradaptasi dengan perubahan teknologi dan tren industri yang sedang berkembang.
- **Kemampuan Kolaborasi:** Perangkat lunak harus dapat digunakan oleh berbagai tim dan individu dengan efektif.
- **Keamanan:** Perangkat lunak harus memiliki tingkat keamanan yang tinggi untuk melindungi data dan informasi pengguna.

Jenis-jenis model pengembangan perangkat lunak yang umum digunakan adalah:

- Model Waterfall:

Model ini menggunakan pendekatan berurutan yang dimulai dari persyaratan, desain, penerapan, pengujian, penyebaran, dan pemeliharaan. Setiap fase harus diselesaikan sebelum fase berikutnya dimulai. Model ini cocok untuk proyek dengan persyaratan yang jelas dan tidak mungkin berubah.

- Model V:

Model ini adalah pengembangan dari model Waterfall. Semua persyaratan dikumpulkan di awal dan tidak dapat diubah. Setiap fase memiliki aktivitas pengujian yang sesuai. Model ini mudah dipahami dan memudahkan manajemen proyek, tetapi tidak cocok untuk proyek kompleks atau dengan persyaratan yang tidak jelas.

- Model Incremental:

Model ini membagi fungsionalitas sistem menjadi peningkatan kecil yang disampaikan satu demi satu secara berurutan. Fungsi yang paling penting diimplementasikan pada tahap awal. Model ini cocok untuk proyek yang memiliki persyaratan yang dapat diubah secara bertahap.

- Model Spiral:

Model ini adalah model proses perangkat lunak iteratif yang digerakkan oleh risiko. Perangkat lunak dikembangkan dalam suatu seri incremental release. Model ini cocok untuk proyek yang memiliki risiko yang besar dan perubahan yang sering.

- Model Agile:

Model ini mendorong iterasi pengembangan dan pengujian yang berkelanjutan. Setiap bagian inkremental dikembangkan selama satu iterasi. Model ini cocok untuk proyek dengan persyaratan yang berubah dan lebih sesuai diterapkan untuk proyek yang kompleks.

- Model RAD (Rapid Application Development):

Model ini menggunakan pendekatan yang cepat dan fleksibel untuk mengembangkan perangkat lunak. Model ini cocok untuk proyek yang memiliki waktu pengembangan yang singkat dan persyaratan yang tidak jelas.

- Model Prototipe:

Model ini membangun prototipe awal dari sistem yang akan dikembangkan. Model ini cocok untuk proyek yang memiliki persyaratan yang tidak jelas dan perlu pengujian awal.

- Model Iterative:

Model ini mengulang beberapa tahap dari siklus pengembangan perangkat lunak. Model ini cocok untuk proyek yang memiliki persyaratan yang berubah secara terus-menerus.

System Analysis adalah proses yang dilakukan untuk mempelajari dan memahami sistem yang ada, termasuk struktur, fungsi, dan perilaku sistem tersebut. Tujuannya adalah untuk mengidentifikasi tujuan dan tujuan sistem, serta menciptakan sistem dan prosedur yang akan mencapainya dengan efisien.

Peran System Analyst dalam Rekayasa Perangkat Lunak

- Identifikasi Masalah: System analyst bertanggung jawab untuk mengidentifikasi masalah yang dihadapi oleh sistem yang ada dan menentukan apa yang harus dilakukan untuk memperbaiki atau mengembangkan sistem tersebut.
- Kumpulan Persyaratan: Mereka mengumpulkan persyaratan dari pengguna sistem dan mengidentifikasi apa yang dibutuhkan oleh sistem untuk mencapai tujuan yang diinginkan.
- Pembuatan Model: System analyst menggunakan berbagai model, seperti Data Flow Diagrams (DFD), Use Cases, dan Entity-Relationship (ER) diagrams, untuk memvisualisasikan sistem dan interaksi yang terjadi di dalamnya.
- Pemilihan Model Pengembangan: Mereka memilih model pengembangan yang sesuai, seperti Waterfall, Agile, atau RAD, berdasarkan jenis proyek dan persyaratan yang ada.
- Pengembangan Rencana: System analyst membantu dalam pengembangan rencana yang detail untuk mengimplementasikan sistem yang diinginkan, termasuk menentukan arsitektur sistem, komponen perangkat lunak dan perangkat keras, serta prosedur operasional.

- Pengujian dan Pemeliharaan: Mereka terlibat dalam pengujian sistem dan memelihara sistem setelah diimplementasikan, termasuk mengumpulkan feedback dari pengguna dan membuat perubahan yang diperlukan untuk meningkatkan kinerja sistem.