

Dalam penerapan proses perangkat lunak yang bermakna, pada dasarnya dipandu oleh prinsip – prinsip inti yang menjadi dasar untuk mengeksekusi metode – metode perangkat lunak yang efektif. Secara rinci prinsip – prinsip yang harus diikuti secara seksama untuk memandu proses-proses dan praktik-praktik rekayasa perangkat lunak modern saat ini adalah sebagai berikut.

Berikut adalah prinsip – prinsip inti yang dapat diterapkan pada kerangka kerja dan dengan perluasannya dapat diterapkan pada setiap proses yang ada di dalam rekayasa perangkat lunak.

Cepat, apapun model proses yang dipilih bersifat preskriptif atau cepat, kecepatan seharusnya merupakan pendekatan penting yang akan memandu. Berusahalah agar pendekatan teknis kita sesederhana mungkin.

Pada setiap langkah berfokuslah pada kualitas, setiap aktivitas proses, tindakan dan pekerjaan seharusnya berfokus pada kualitas produk – produk kerja yang akan dihasilkan.

Siap untuk beradaptasi, adaptasilah pendekatan pada batasan yang muncul pada permasalahan, orang dan pada proyek rekayasa perangkat lunak itu sendiri.

Bentuklah tim yang efektif, tim yang efektif adalah tim yang dapat mengorganisasi dirinya sendiri, yang saling mempercayai dan saling menghargai.

Menetapkan mekanisme untuk komunikasi dan koordinasi.

Mengelola perubahan, mekanisme tertentu harus ditetapkan untuk mengelola bagaimana perubahan – perubahan diminta, dinilai, diterima dan diimplementasikan.

Lakukan penilaian terhadap risiko, sesuatu yang buruk mungkin saja terjadi saat perangkat lunak dikembangkan. Dengan demikian merupakan hal yang esensial untuk menetapkan langkah – langkah untuk mengatasinya.

Buat produk – produk kerja yang bermanfaat untuk lainnya.

Praktik rekayasa perangkat lunak memiliki sasaran yaitu mengirimkan tepat waktu, memiliki kualitas yang tinggi, berupa perangkat lunak operasional yang memuat fungsi – fungsi dan fitur – fitur yang sesuai dengan kebutuhan semua penyandang dana. Prinsip inti yang menyadari suatu praktik rekayasa perangkat lunak adalah sebagai berikut.

Bagi dan pecahkan, sebuah permasalahan yang berukuran besar akan lebih mudah diselesaikan jika dibagi – bagi ke dalam sejumlah elemen – elemen yang merupakan bagiannya.

Memahami penggunaan abstraksi, abstraksi sesungguhnya adalah suatu penyederhanaan dari beberapa elemen kompleks suatu perangkat lunak yang digunakan untuk mengkomunikasikan suatu makna tertentu dalam bentuk yang relative lebih sederhana. Abstraksi pada dasarnya adalah untuk menghilangkan kebutuhan untuk komunikasi yang bersifat sangat rinci. Tanpa pemahaman yang baik tentang rincian, penyebab dari permasalahan tidak akan bisa dengan mudah didiagnosa.

Berusaha untuk konsisten, prinsip konsistensi pada dasarnya menyarankan sesuatu yang mudah dipahami akan membuat perangkat lunak mudah untuk digunakan.

Berfokus pada pengalihan informasi, dalam setiap kasus, informasi – informasi akan mengalir melintasi antarmuka dan sebagai akibatnya muncul kemungkinan – kemungkinan untuk terjadinya kesalahan – kesalahan atau pengabaian – pengabaian atau kemenduaan arti, Implikasi dari prinsip ini adalah bahwa kita harus memberikan perhatian khusus pada analisis, perancangan, konstruksi dan pengujian antarmuka – antarmuka.

Kembangkan perangkat lunak yang menekankan pada modularitas yang efektif. Setiap perangkat lunak yang kompleks pada dasarnya dapat dibagi menjadi modul – modul yang merupakan bagian-bagiannya.

Mencari pola. Formalisasi penyelesaian permasalahan – permasalahan dan hubungan – hubungannya akan sangat membantu untuk keberhasilan dalam mendapatkan isi pengetahuan yang mendefinisikan pemahaman tentang arsitektur perangkat lunak yang baik, yang akan memenuhi kebutuhan para pengguna.

Jika mungkin, lihatlah permasalahan dan penyelesaiannya dari berbagai sudut pandang yang berbeda.

Ingatlah bahwa seseorang akan melakukan pemeliharaan terhadap perangkat lunak yang dikembangkan. Aktivitas – aktivitas pemeliharaan perangkat lunak ini dapat difasilitasi dengan baik jika praktik rekayasa perangkat lunak yang baik diterakan di sepanjang proses pengembangan perangkat lunak.

Berikut adalah prinsip yang secara umum dapat diterapkan pada semua bentuk komunikasi yang terjadi di dalam proyek perangkat lunak.

Dengarkan. Berfokuslah pada kata-kata, alih-alih mencoba merumuskan tanggapan atas kata-kata tersebut. Lakukanlah pertanyaan-pertanyaan klarifikasi jika sesuatu tidak jelas tetapi berusaha untuk tidak melakukan interupsi.

Lakukan persiapan sebelum melakukan komunikasi. Luangkanlah waktu untuk memahami permasalahan tersebut sebelum bertemu dengan orang terkait. Jika mungkin lakukanlah penelitian sederhana untuk memahami jargon – jargon yang sesuai ranah bisnis lawan bicara.

Seseorang sebaiknya menjadi fasilitator aktivitas komunikasi. Setiap komunikasi sebaiknya memiliki pemimpin agar percakapan berada pada ranah yang benar, ada yang dapat menjadi penengah konflik dan untuk memastikan prinsip-prinsip komunikasi yang lain diikuti.

Komunikasi yang dilakukan secara tatap muka adalah yang terbaik.

Lakukan pencatatan dan dokumentasi.

Usahakan untuk bekerjasama.

Tetaplah terfokus, lakukan modularitas pembahasan.

Jika sesuatu tidak jelas, gambarlah. Sketsa atau gambar seringkali memperjelas saat kata – kata gagal menjelaskannya.

Pada prinsip ke sembilan ini ada tiga catatan yaitu: Sekali kita telah membuat kesepakatan tentang suatu pembahasan, bergeraklah ke pembahasan lainnya; Jika kita tidak bisa mencapai kesepakatan tentang suatu pokok pembahasan, bergeraklah dengan segera ke pembahasan lainnya; Jika suatu fitur atau fungsi tidak jelas dan tidak bisa diklarifikasi saat ini, bergeraklah pada pembahasan berikutnya.

Negoisasi bukanlah suatu kontes atau perlombaan. Merupakan hal yang terbaik saat kedua pihak menang.

Aktivitas perencanaan sesungguhnya merupakan sejumlah praktik manajemen dan teknis yang memungkinkan tim perangkat lunak untuk mendefinisikan suatu peta jalan yang pada gilirannya memungkinkan tim perangkat lunak mencapai tujuan – tujuan

yang bersifat strategik dan taktis. Berikut adalah prinsip – prinsip perencanaan yang sebaiknya diterapkan.

Memahami lingkup proyek. Lingkup proyek menyediakan bagi tim perangkat lunak suatu tujuan.

Melibatkan stakeholder pada aktivitas perencanaan. Para stakeholder harus mendefinisikan prioritas – prioritas dan menetapkan batasan proyek perangkat lunak.

Memahami bahwa perancangan bersifat iteratif. Suatu perencanaan proyek tidak pernah bersifat kaku. Saat pekerjaan dimulai, sangat mungkin bahwa segala sesuatunya akan mengalami perubahan – perubahan. Sebagai akibatnya, perencanaan harus diatur sedemikian rupa untuk bisa mengakomodasi perubahan – perubahan itu.

Melakukan perkiraan berdasarkan apa yang kita mengerti. Perhatian yang perlu dimasukkan dalam suatu perencanaan adalah indikasi usaha, biaya, lamanya pengerjaan pekerjaan – pekerjaan, berdasarkan pemahaman tim saat ini tentang bagaimana mengerjakan proyek perangkat lunak.

Pertimbangkan risiko saat kita mendefinisikan perencanaan. Perencanaan proyek perangkat lunak semestinya disesuaikan dengan risiko – risiko yang mungkin akan terjadi.

Cobalah bersikap realistis. Hampir tidak pernah ada rekayasawan perangkat lunak yang tidak pernah membuat kesalahan. Kenyataan ini perlu dipertimbangkan saat perencanaan proyek perangkat lunak.

Sesuaikan granularitas saat mendefinisikan perencanaan. Granularitas merujuk pada peringkat rincian yang dilakukan saat perencanaan proyek perangkat lunak dilakukan.

Definisikan bagaimana memikirkan hal – hal yang berkaitan dengan kualitas. Perencanaan proyek perangkat lunak seharusnya mengidentifikasi bagaimana tim perangkat lunak berfikir tentang bagaimana cara memastikan kualitas.

Deskripsikan bagaimana kita akan melakukan sesuatu yang berkaitan dengan cara kita mengakomodasi perubahan. Bagaimanapun bagus nya perencanaan kita, perencanaan itu dapat saja digagalkan oleh perubahan yang tidak terkendali. Kita

harus mengidentifikasi bagaimana perubahan – perubahan akan diakomodasi saat pekerjaan rekayasa perangkat lunak berlangsung.

Lakukan pelacakan terhadap perencanaan dan buatlah penyesuaian – penyesuaian jika dibutuhkan.

Dalam pekerjaan rekayasa perangkat lunak, ada dua jenis model yang dapat dibuat yaitu model-model yang berkaitan dengan spesifikasi kebutuhan dan model-model yang berkaitan dengan perancangan yang mengarah pada implementasi selanjutnya. Model-model spesifikasi kebutuhan memperlihatkan spesifikasi – spesifikasi kebutuhan pengguna dengan menggambarkan perangkat lunak dalam tiga ranah yang berbeda yaitu ranah informasi, ranah fungsional dan ranah perilaku. Sedangkan ranah perancangan menggambarkan karakteristik – karakteristik perangkat lunak yang akan sangat membantu para praktisi untuk mengembangkan perangkat lunak secara efektif yaitu arsitektur perangkat lunak, antarmuka pengguna dan rincian berperingkat komponen. Scott Ambler dan Rob Jeffries dalam mendefinisikan sejumlah prinsip pemodelan proses cepat sebagai berikut.

Tujuan utama tim perangkat lunak adalah untuk mengembangkan perangkat lunak, bukan model-model.

Jangan membuat model lebih banyak dari yang kita butuhkan.

Berusahalah membuat model sederhana yang menjelaskan permasalahan atau perangkat lunak.

Kembangkan model – model sedemikian rupa sehingga perubahan dimungkinkan.

Berusahalah untuk menetapkan suatu kegunaan eksplisit untuk masing – masing model yang dibuat.

Adaptasi model – model yang dikembangkan dengan perubahan yang terjadi pada sistem.

Jangan kaku dengan sintaks model. Jika model saat ini dapat mengkomunikasikan isi dengan baik, penampilan adalah nomor dua.

Jika secara intuisi kita merasa bahwa model tidak benar meski kelihatan diatas kertas tidak ada masalah apa – apa, kita mungkin memiliki alasan untuk mempertimbangkan ulang.

Dapatkan umpan balik sesegera mungkin.

Prinsip – prinsip Pemodelan Spesifikasi Kebutuhan.

Ranah informasi dari suatu permasalahan harus ditampilkan dan dipahami. Ranah informasi menggambarkan data yang mengalir ke dalam sistem, menggambarkan data yang akan keluar dari sistem, menggambarkan penyimpanan data yang mengumpulkan dan mengorganisasi objek – objek data yang bersifat persisten.

Fungsi – fungsi yang akan dilakukan perangkat lunak harus didefinisikan. Fungsi – fungsi perangkat lunak tentunya menyediakan manfaat langsung pada para pengguna akhir dan juga menyediakan dukungan internal untuk fitur – fitur yang tampak oleh pengguna.

Perilaku perangkat lunak akibat kejadian – kejadian yang bersifat eksternal harus direpresentasikan. Input diberikan oleh pengguna akhir, kendali data dilakukan sistem eksternal atau pemantauan data yang dikumpulkan melintas jaringan, semuanya menyebabkan perangkat lunak berperilaku dengan cara tertentu.

Model yang menjelaskan informasi, fungsi dan perilaku harus dipisahkan dalam bentuk yang tidak menyingkapkan rinciannya dan harus digambarkan dalam bentuk perlapisan atau hierarki. Spesifikasi – spesifikasi kebutuhan memungkinkan untuk memahami permasalahan dan menetapkan penyelesaiannya.

Pekerjaan analisis seharusnya bergerak dari informasi yang bersifat esensial menuju rincian implementasi.

Prinsip – prinsip Pemodelan Rancangan:

Rancangan harus bisa dilacak balik ke model spesifikasi kebutuhan;

Selalu pertimbangkan arsitektur perangkat lunak yang akan dikembangkan;

Perancangan data sama pentingnya dengan perancangan fungsi – fungsi pemrosesan;

Antarmuka – antarmuka harus dirancang secara hati – hati;

Perancangan antarmuka pengguna seharusnya disesuaikan dengan kebutuhan pengguna. Meski demikian, dalam setiap kasus, antarmuka harus dibuat dengan menekankan kemudahan penggunaannya;

Perancangan peringkat komponen sebaiknya mandiri secara fungsional;

Komponen – komponen seharusnya bersifat saling tidak bergantung satu sama lain dan juga tidak bergantung pada lingkungan eksternal;

Representasi rancangan atau model seharusnya dapat dipahami dengan mudah;

Perancangan seharusnya dikembangkan secara iteratif, dimana masing – masing iterasi, perancangan seharusnya menekankan kesederhanaan.

Saat prinsip – prinsip perancangan diatas diterapkan, maka akan menghasilkan suatu rancangan yang menampilkan baik kualitas internal maupun eksternal.

Faktor kualitas eksternal adalah properti perangkat lunak yang dapat dilihat oleh para pengguna.

Sedangkan faktor kualitas internal adalah kepentingannya untuk memandu pekerjaan rekayasawan perangkat lunak.

Aktivitas konstruksi mencakup di dalamnya sejumlah pekerjaan penulisan kode dan pengujian hingga perangkat lunak yang siap dikirimkan ke pelanggan dan ke para pengguna akhir. Berikut adalah prinsip – prinsip dan konsep – konsep yang mendasari penulisan kode program dan pengujiannya.

Prinsip – prinsip persiapan. Sebeum kita menulis baris pertama kode program komputer kita, pastikan bahwa:

Memahami permasalahan yang akan diselesaikan; Memahami prinsip – prinsip dan konsep – konsep dasar perancangan; Telah memilih bahasa pemograman yang sesuai dengan kebutuhan perangkat lunak yang akan dikembangkan dan lingkungan dimana ia akan bekerja; Telah memilih lingkungan pemograman yang memiliki perkakas – perkakas yang akan mempermudah pekerjaan; Membuat sejumlah pengujian unit yang akan diterapkan setelah kode program selesai.

Prinsip – prinsip pemograman. Saat mulai menulis kode program komputer, pastikan bahwa: Mengimplementasikan algoritma yang akan digunakan dalam program komputer yang ditulis dengan mengikuti praktik pemograman terstruktur.

Mempertimbangkan penggunaan pemograman parsial.

Memilih struktur data yang sesuai dengan apa yang telah ditetapkan oleh rancangan yang telah dibuat sebelumnya.

Memahami arsitektur program dan membuat antarmuka – antarmuka yang konsisten terhadap asitektur program.

Membuat logika kondisional sesederhana mungkin.

Membuat peulangan – perulangan bersarang dengan cara sedemikian rupa sehingga perulangan bersarang tersebut mudah diuji.

Memilih nama – nama perubah yang bermakna dan penamaan itu seharusnya mengikuti standar pengkodean yang dimiliki oleh bahasa pemograman yang dipilih.

Menulis kode yang dapat mendokumentasikan dirinya sendiri.

Membuat kode – kode dengan cara yang memudahkan pemahaman.

Prinsip – prinsip validasi. Setelah menyelesaikan penulisan kode pastikan bahwa:

Jika mungkin, melakukan penelusuran pada kode programan yang telah ditulis untuk melakukan pemeriksaan ulang kebenaran sintaks dan logikanya.

Melakukan pengujian – pengujian unit dan melakukan pembetulan kesalahan – kesalahan yang sebelumnya tidak tetingkap.

Melakukan refaktor terhadap kode yang dihasilkan.

Prinsip – prinsip pengujian menurut Glen Myers dalam (Roger S. Pressman, 2012):

Pengujian merupakan proses eksekusi suatu program dengan tujuan untuk menemukan kesalahan – kesalahan yang ada didalamnya.

Kasus pengujian yang baik adalah pengujian yang memiliki kemungkinan yang tinggi untuk menemukan kesalahan – kesalahan yang tidak ditemukan sebelumnya.

Pengujian yang berhasil adalah pengujian yang mampu menyingkapkan kesalahan yang tidak ditemukan sebelumnya.

Prinsip pengujian menurut Davis dalam (Roger S. Pressman, 2012) dapat diadaptasi sebagai berikut:

Semua pengujian seharusnya dapat dilacak balik ke spesifikasi kebutuhan pelanggan.



Pengujian – pengujian seharusnya direncanakan lama sebelum pengujian – pengujian itu dimulai.

Prinsip Pareto dapat diterapkan pada pengujian perangkat lunak. Prinsip Pareto mengatakan bahwa 80 persen kesalahan yang tidak terungkap selama pengujian dilaksanakan pada umumnya berada di 20 persen komponen – komponen program secara keseluruhan.

Pengujian seharusnya bergerak dari bagian yang lebih kecil dan menerus ke bagian yang semakin besar.

Pengujian yang lengkap merupakan hal yang tidak mungkin.

Aktivitas penyerahan perangkat lunak kepada pelanggan (deployment) memiliki tiga aksi penting yaitu pengiriman, dukungan dan umpan balik. Berikut adalah prinsip – prinsip yang seharusnya dipatuhi saat tim perangkat lunak mengirimkan suatu versi perangkat lunak.

Harapan – harapan pelanggan untuk perangkat lunak harus dikelola dengan baik.

Paket – paket pengiriman lengkap seharusnya dirakit dan sudah diuji.

Sistem – sistem pendukung harus ditetapkan sebelum perangkat lunak dikirimkan.

Material – material instruksional yang sesuai harus disediakan bagi para pengguna.

Perangkat lunak yang penuh dengan kesalahan seharusnya diperbaiki lebih dahulu, pengiriman bisa dilakukan di waktu – waktu selanjutnya.