

BAB

17

JAMINAN KUALITAS PERANGKAT LUNAK

A. Pendahuluan

Pengendalian dan jaminan mutu merupakan aktivitas penting bagi setiap bisnis yang memproduksi produk untuk digunakan oleh orang lain. Sebelum abad kedua puluh, pengendalian kualitas adalah hal yang paling penting tanggung jawab sepenuhnya dari pengrajin yang membuat suatu produk. Seiring berjalannya waktu dan massa teknik produksi menjadi hal yang lumrah, pengendalian kualitas menjadi suatu kegiatan dilakukan oleh orang lain selain orang yang membuat produk tersebut.

Fungsi jaminan dan pengendalian kualitas formal pertama diperkenalkan di Bell Labs pada tahun 1916 dan menyebar dengan cepat ke seluruh dunia manufaktur. Selama Pada tahun 1940-an, pendekatan pengendalian mutu yang lebih formal diusulkan. Ini bergantung pada pengukuran dan perbaikan proses berkelanjutan sebagai elemen kunci manajemen mutu.

Saat ini, setiap perusahaan memiliki mekanisme untuk menjamin kualitas produknya. Nyatanya, pernyataan eksplisit tentang kepedulian perusahaan terhadap kualitas telah menjadi suatu pemasaran taktik selama beberapa dekade terakhir. Sejarah penjaminan mutu dalam pengembangan perangkat lunak sejajar dengan sejarah kualitas dalam pembuatan perangkat keras. Selama hari-hari awal komputasi (1950an dan 1960an), kualitas adalah tanggung jawab programmer. Standar kualitas jaminan untuk perangkat lunak diperkenalkan dalam

pengembangan perangkat lunak kontrak militer selama tahun 1970an dan telah menyebar dengan cepat ke pengembangan perangkat lunak di dunia komersial. Memperluas definisi yang disajikan sebelumnya, kualitas perangkat lunak jaminan adalah “pola tindakan yang terencana dan sistematis” yang diperlukan untuk memastikan kualitas tinggi dalam perangkat lunak. Ruang lingkup tanggung jawab penjaminan mutu mungkin paling baik dicirikan dengan memparafrasekan iklan mobil yang pernah populer: “Kualitas Adalah Pekerjaan #1.” Implikasinya terhadap perangkat lunak adalah banyaknya konstituen yang berbeda memiliki tanggung jawab penjaminan kualitas perangkat lunak—insinyur perangkat lunak, manajer proyek, pelanggan, tenaga penjualan, dan individu yang bertugas dalam grup SQA. Grup SQA berfungsi sebagai perwakilan internal pelanggan. Itu adalah orang yang melakukan SQA harus melihat perangkat lunak dari sudut pandang pelanggan melihat. Apakah perangkat lunak cukup memenuhi faktor kualitas yang disebutkan. Memiliki pengembangan perangkat lunak telah dilakukan sesuai dengan standar yang telah ditetapkan? Memiliki disiplin teknis menjalankan perannya dengan baik sebagai bagian dari aktivitas SQA? Itu Kelompok SQA mencoba menjawab pertanyaan ini dan pertanyaan lainnya untuk memastikan perangkat lunak tersebut kualitas tetap terjaga.

B. Tugas SQA

Piagam grup SQA adalah untuk membantu tim perangkat lunak dalam mencapai produk akhir berkualitas tinggi. Institut Rekayasa Perangkat Lunak merekomendasikan seperangkat SQA tindakan yang membahas perencanaan penjaminan mutu, pengawasan, pencatatan, analisis, dan pelaporan. Tindakan ini dilakukan (atau difasilitasi) oleh SQA independen kelompokkanitu:

1. Mempersiapkan rencana SQA untuk sebuah proyek.

Rencana tersebut dikembangkan sebagai bagian dari perencanaan proyek dan ditinjau oleh seluruh pemangku kepentingan. Kualitas asuransi tindakan yang dilakukan oleh

tim rekayasa perangkat lunak dan grup SQA adalah diatur oleh rencana tersebut. Rencana tersebut mengidentifikasi evaluasi yang harus dilakukan, audit dan peninjauan yang akan dilakukan, standar yang berlaku untuk proyek, prosedur pelaporan dan pelacakan kesalahan, produk kerja itu dihasilkan oleh kelompok SQA, dan umpan balik yang akan diberikan kepada tim perangkat lunak.

2. Berpartisipasi dalam pengembangan proses perangkat lunak proyek keterangan.

Tim perangkat lunak memilih proses pekerjaan yang akan dilakukan dilakukan. Kelompok SQA meninjau deskripsi proses untuk kepatuhan terhadap kebijakan organisasi, standar perangkat lunak internal, dan eksternal standar yang diberlakukan (misalnya, ISO-9001), dan bagian lain dari perangkat lunak rencana proyek.

3. Meninjau aktivitas rekayasa perangkat lunak untuk memverifikasi kepatuhan terhadap proses perangkat lunak yang ditentukan.

Grup SQA mengidentifikasi, mendokumentasikan, dan melacak penyimpangan dari proses dan memverifikasi bahwa koreksi telah dilakukan.

4. Mengaudit produk kerja perangkat lunak yang ditunjuk untuk memverifikasi kepatuhannya yang didefinisikan sebagai bagian dari proses perangkat lunak.

Kelompok SQA mengulas produk kerja terpilih; mengidentifikasi, mendokumentasikan, dan melacak penyimpangan; memverifikasi bahwa koreksi telah dilakukan; dan secara berkala melaporkan hasilnya pekerjaan kepada manajer proyek.

5. Memastikan bahwa penyimpangan dalam pekerjaan perangkat lunak dan produk kerja didokumentasikan dan ditangani sesuai dengan prosedur yang terdokumentasi.

Penyimpangan mungkin ditemui dalam rencana proyek, deskripsi proses, standar yang berlaku, atau produk kerja rekayasa perangkat lunak.

6. Catat setiap ketidakpatuhan dan laporkan kepada manajemen senior.

Item ketidakpatuhan dilacak sampai diselesaikan. Selain tindakan ini, kelompok SQA mengoordinasikan pengendalian dan pengelolaan perubahan dan membantu mengumpulkan dan menganalisis metrik perangkat lunak.

C. Peran dan Pentingnya Jaminan Kualitas Perangkat Lunak

Jaminan kualitas perangkat lunak (*Software Quality Assurance* atau SQA) adalah proses sistematis untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi standar kualitas yang ditetapkan dan berfungsi sesuai dengan kebutuhan dan harapan pengguna. SQA mencakup seluruh siklus hidup pengembangan perangkat lunak, mulai dari perencanaan, desain, pengkodean, hingga pengujian dan pemeliharaan. Fokus utama SQA adalah untuk mencegah kesalahan dan cacat dalam perangkat lunak, serta memastikan bahwa produk akhir dapat diandalkan, aman, dan efisien.

Memastikan Kualitas Produk: SQA memastikan bahwa perangkat lunak memenuhi standar kualitas yang ditetapkan, sehingga produk akhir bebas dari cacat dan berfungsi dengan baik.



Gambar 7. 1 Jaminan Kualitas Perangkat Lunak

Berikut ini ketentuan dari Jaminan Kualitas Perangkat Lunak

1. Meningkatkan Keandalan dan Kinerja: Dengan menerapkan praktik SQA yang ketat, perangkat lunak menjadi lebih andal dan berkinerja tinggi, mengurangi risiko kegagalan sistem.
2. Meminimalkan Biaya Perbaikan: Deteksi dan perbaikan cacat pada tahap awal pengembangan lebih murah dibandingkan dengan memperbaiki masalah setelah perangkat lunak diluncurkan.
3. Meningkatkan Kepuasan Pengguna: Perangkat lunak yang berkualitas tinggi memberikan pengalaman pengguna yang baik, meningkatkan kepuasan dan kepercayaan pengguna.
4. Mematuhi Standar dan Regulasi: SQA membantu memastikan bahwa perangkat lunak mematuhi standar industri dan regulasi yang berlaku.

D. Proses Jaminan Kualitas Perangkat Lunak

Proses Jaminan Kualitas Perangkat Lunak (*Software Quality Assurance* atau SQA) adalah rangkaian kegiatan yang dirancang untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi standar kualitas yang telah ditetapkan. Proses ini mencakup berbagai langkah mulai dari perencanaan, desain, pengembangan, hingga pengujian dan pemeliharaan.

Berikut adalah tahapan utama dalam proses SQA:

1. Perencanaan Kualitas (*Quality Planning*)

Perencanaan kualitas adalah langkah awal dalam proses SQA yang mencakup identifikasi standar, metodologi, dan alat yang akan digunakan untuk menjamin kualitas perangkat lunak.

a. Penentuan Standar Kualitas

Menetapkan standar kualitas yang akan diikuti, seperti ISO/IEC 9126, ISO/IEC 25010, atau standar internal perusahaan.

b. Rencana Jaminan Kualitas

Membuat rencana jaminan kualitas yang mendetail, mencakup tujuan kualitas, strategi untuk mencapainya, sumber daya yang dibutuhkan, dan jadwal kegiatan SQA.

2. Definisi Kebutuhan (*Requirements Definition*)

Menetapkan kebutuhan perangkat lunak berdasarkan analisis kebutuhan pengguna dan pemangku kepentingan.

a. Pengumpulan Kebutuhan

Melakukan wawancara, survei, dan observasi untuk mengumpulkan kebutuhan pengguna.

b. Dokumentasi Kebutuhan

Menyusun dokumen spesifikasi kebutuhan yang jelas dan lengkap, termasuk fungsionalitas, kinerja, dan batasan.

3. Desain dan Pengembangan (*Design and Development*)

Desain perangkat lunak harus mempertimbangkan aspek kualitas sejak awal untuk memastikan bahwa produk akhir memenuhi standar yang ditetapkan.

a. Desain Arsitektur

Merancang arsitektur perangkat lunak yang mendukung modularitas, skalabilitas, dan pemeliharaan.

b. Desain Detil

Membuat desain rinci untuk setiap komponen perangkat lunak, termasuk diagram alur, struktur data, dan antarmuka pengguna.

c. Review Desain

Melakukan review dan inspeksi desain untuk mendeteksi kesalahan dan memastikan bahwa desain memenuhi kebutuhan yang telah ditentukan.

4. Implementasi (*Implementation*)

Proses pengkodean di mana desain perangkat lunak diterjemahkan menjadi kode program.

a. Pengembangan Kode

Menulis kode sumber sesuai dengan standar pengkodean yang telah ditetapkan.

b. Review dan Inspeksi Kode

Melakukan review dan inspeksi kode untuk memastikan bahwa kode bebas dari kesalahan dan mengikuti standar kualitas.

5. Penggunaan Alat Otomasi

Memanfaatkan alat otomatisasi seperti linters, static code analysis, dan continuous integration untuk mendeteksi dan memperbaiki kesalahan lebih awal.

6. Pengujian Perangkat Lunak (*Software Testing*)

Pengujian adalah tahap kritis dalam SQA yang bertujuan untuk mendeteksi cacat dan memastikan bahwa perangkat lunak berfungsi sesuai dengan kebutuhan. Pengujian Unit: Menguji setiap unit atau komponen perangkat lunak secara individual.

a. Pengujian Integrasi

Menguji interaksi antara berbagai unit atau komponen untuk memastikan bahwa mereka bekerja bersama dengan benar.

b. Pengujian Sistem

Menguji seluruh sistem perangkat lunak sebagai satu kesatuan untuk memastikan bahwa semua bagian berfungsi sesuai spesifikasi.

c. Pengujian Akseptansi

Pengujian akhir oleh pengguna atau pemangku kepentingan untuk memastikan bahwa perangkat lunak memenuhi kriteria penerimaan.

7. Pemantauan dan Kontrol Kualitas (*Quality Monitoring and Control*)

Tahap ini melibatkan pemantauan dan pengukuran proses dan produk untuk memastikan bahwa kualitas tetap terjaga sepanjang siklus hidup perangkat lunak.

a. Pemantauan Proses

Memantau proses pengembangan untuk memastikan bahwa semua aktivitas SQA dilakukan sesuai rencana.

b. Audit Kualitas

Melakukan audit internal atau eksternal untuk mengevaluasi kesesuaian proses dan hasil dengan standar kualitas.

c. Pengukuran Kinerja

Mengukur kinerja perangkat lunak menggunakan metrik seperti waktu respon, throughput, dan penggunaan sumber daya.

8. Manajemen Cacat (*Defect Management*)

Proses ini mencakup pelacakan, analisis, dan perbaikan cacat yang ditemukan selama pengujian atau setelah perangkat lunak diluncurkan.

a. Pelacakan Cacat

Menggunakan alat manajemen cacat untuk mencatat dan melacak cacat yang ditemukan.

b. Analisis Akar Penyebab

Melakukan analisis untuk menemukan akar penyebab cacat dan mencegah terulangnya masalah yang sama.

c. Perbaikan dan Validasi

Memperbaiki cacat dan melakukan pengujian ulang untuk memastikan bahwa perbaikan berhasil.

9. Dokumentasi dan Pelaporan (*Documentation and Reporting*)

Mendokumentasikan seluruh proses SQA dan menyusun laporan kualitas yang memberikan gambaran tentang status kualitas perangkat lunak.

a. Dokumentasi Proses

Mendokumentasikan semua kegiatan SQA, termasuk hasil pengujian, audit, dan pemantauan.

b. Laporan Kualitas

Menyusun laporan kualitas yang disajikan kepada manajemen dan pemangku kepentingan lainnya.

c. Pemeliharaan (*Maintenance*)

Setelah perangkat lunak diluncurkan, SQA terus berlanjut melalui pemeliharaan untuk memastikan bahwa perangkat lunak tetap berkualitas tinggi selama masa pakainya.

10. Pemantauan Pasca Peluncuran

Memantau kinerja perangkat lunak setelah diluncurkan untuk mendeteksi masalah yang mungkin muncul.

11. Perbaikan dan Pembaruan

Melakukan perbaikan dan pembaruan perangkat lunak berdasarkan umpan balik pengguna dan analisis kinerja.

Proses Jaminan Kualitas Perangkat Lunak adalah kerangka kerja yang komprehensif dan sistematis yang bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan berkualitas tinggi, dapat diandalkan, dan memenuhi kebutuhan pengguna. Melalui penerapan praktik SQA yang tepat, organisasi dapat meningkatkan efisiensi pengembangan, mengurangi biaya perbaikan, dan memberikan produk yang memuaskan pengguna. Dengan demikian, SQA tidak hanya berkontribusi pada kesuksesan proyek perangkat lunak, tetapi juga pada reputasi dan keberlanjutan organisasi dalam jangka panjang.