

CHAPTER 1

Tantangan Mutu Perangkat Lunak

Dua pertanyaan dasar yang diajukan sebelum kita melakukan pendaftaran jenis dari subjek dan rincian dari buku :

- 1) Apakah membuat buku yang di khususkan untuk penjaminan perangkat lunak dapat dibenarkan atau, dengan kata lain, kenapa tidak menggunakan buku tentang penjaminan mutu pada umum nya yang tersedia dan dapat diterapkan di berbagai macam industri ?
- 2) Jika telah memutuskan untuk mengembangkan buku yang dikhususkan untuk penjaminan mutu perangkat lunak, dari bagian mana kah dari penjaminan perangkat lunak, apakah mulai dari hobi amatir sampai professional ?

Tujuan dari bab ini ialah untuk menjawab pertanyaan diatas dengan cara menelusuri berbagai masalah terkait.

Setelah menyelesaikan bab ini , anda akan mampu :

- Mengidentifikasi keunikan karakteristik dari perangkat lunak sebagai produk dan sebagai proses produksi dengan penanganan masalah mutu yang berbeda pada umum nya.
- Mengenali karakteristik dari lingkungan pengembangan software professional dan perawatan software.
- Menjelaskan masalah utama yang dihadapi pada pengembangan perangkat lunak dan tim perawatan.

1.1 Keunikan dari Penjaminan Mutu Perangkat Lunak

“Lihat ini,” seru teman saya sembari memberikan ku selebaran garansi terbatas Dagal Feature. “ Bahkan Dagal Feature tidak bisa mengatasi bugs pada software.”

Dia menunjukkan paragraf pendek di halaman 3 pada selebaran yang menyatakan kondisi garansi dari AMGAL, leading software master product yang terjual di seluruh dunia. Selebaran tersebut menyatakan :

GARANSI TERBATAS

Dagal Features tidak menyediakan jaminan, baik secara tersurat maupun tersirat, sehubungan dengan kinerja, keandalan, atau kesesuaian AMGAL untuk tujuan tertentu apa pun. Dagal Features tidak menjamin bahwasanya software atau dokumentasi yang ada akan memenuhi keperluan anda. Meskipun Dagal Features telah melakukan berbagai macam test terhadap software dan telah mengkaji dokumentasi nya, Dagal Features tidak menjamin software dan dokumentasi terbebas dari kecacatan. Dagal Features tidak akan menjadi bertanggung jawab atas segala kerusakan, insidental, langsung, tidak langsung atau konsekuensial, timbul sebagai akibat dari data yang rusak, biaya pemulihan, kerugian laba, dan ketiga klaim pihak. perangkat lunak dilisensikan "sebagaimana adanya". Pembeli mengasumsikan risiko lengkap yang berasal dari penerapan program AMGAL, kualitas dan kinerja.

Jika kerusakan fisik ditemukan pada dokumentasi atau CD yang di distribusikan oleh AMGAL, maka Dagal Features akan menggantinya, tanpa biaya, selama 180 hari masa pembelian dengan bukti pembelian terlampir

“Apakah software buatan AMGAL sangat spesial sampai sampai para developer tidak mampu menjamin produk mereka bebas dari bug ? “ lanjut teman saya.

“Apakah software yang lain juga membatasi garansi sebagaimana yang dilakukan AMGAL ?.”

Meskipun Dagal Features dan AMGAL adalah fiktif, pemeriksaan terhadap jaminan yang ditawarkan oleh pengembang perangkat lunak lain memiliki pola yang sama. Tidak ada developer perangkat lunak yang menyatakan produk mereka terbebas dari kecacatan atau bug. Pernyataan ini sebenarnya menunjukkan

perbedaan unsur penting antara perangkat lunak dengan industri lainnya seperti mobil, mesin cuci atau radio. Perbedaan nya dapat di kategorikan sebagai berikut :

- 1) **Kompleksitas Produk.** Kompleksitas produk dapat diukur menggunakan angka dari mode operasional yang diizinkan produk. Produk industri, bahkan mesin sekalipun tidak mengizinkan lebih dari beberapa ribu mode operasional, yang dibuat dari berbagai macam pengaturan mesin. Dilihat dari software package pada umum nya, kita bisa melihat jutaan lebih kemungkinan operasi software yang dapat dilakukan. Memastikan bahwa orang banyak kemungkinan operasional didefinisikan dengan benar dan dikembangkan adalah tantangan bagi industri perangkat lunak.
- 2) **Visibilitas Produk.** Dimana pada umum nya produk industri terlihat, akan tetapi produk software tidak terlihat. Kebanyakan kecacatan pada produk industri dapat terdeteksi pada saat proses manufaktur. Lebih-lebih lagi jika suatu bagian hilang pada produk industry maka akan sangat terlihat (bayangkan mobil baru tanpa pintu). Akan tetapi, kecacatan pada produk software sangat susah terlihat, bahkan mungkin saja ada beberapa bagian software yang tidak ada.
- 3) **Pengembangan Produk dan Proses Produksi.** Mari kita ulas fase-fase dimana kemungkinan kecacatan produk dapat dideteksi:
 - **Pengembangan Produk.** Pada fase ini, desainer dan staff penjamin mutu memeriksa dan mencoba prototype produk, dengan tujuan untuk mengetahui dimana letak kecacatan nya.
 - **Perencanaan Produksi Produk.** Pada fase ini, proses produksi dan alat telah dirancang dan disediakan. Sebagian produk memerlukan rancangan produksi khusus. Pada fase ini dilakukan peninjauan Kembali dengan tujuan untuk mendeteksi kecacatan yang “lolos” pada fase sebelum nya.
 - **Manufaktur.** Pada fase ini, prosedur QA dilakukan dengan tujuan mendeteksi kegagalan produk . Biasanya kecacatan yang dideteksi pada tahap awal manufaktur dapat di koreksi dengan cara

mengganti desain atau material agar menghilangkan kecacatan kedepan nya.

Dibanding dengan produk industri, produk software tidak melakukannya pada tiga tahap produksi. Tahap ketika kecacatan produk dapat di deteksi adalah pada tahap development(pengembangan). Mari kita ulas fase atau tahap untuk mendeteksi kecacatan produk pada produksi software:

- **Pengembangan Produk.** Pada tahap ini, tim pengembang dan staff penjaminan mutu diarahkan untuk mendeteksi cacat bawaan produk. Setelah itu prototype yang telah di approve dan siap diproduksi telah tersedia.
- **Perencanaan Produksi Produk.** Tahap ini tidak diperlukan untuk proses produksi perangkat lunak, karena pembuatan salinan perangkat lunak dan pencetakan manual perangkat lunak dilakukan secara otomatis. Ini berlaku untuk produk perangkat lunak apa pun, baik jumlah salinannya kecil, seperti dalam perangkat lunak yang dibuat khusus, atau besar, seperti dalam paket perangkat lunak yang akan dijual kepada masyarakat umum.
- **Manufaktur** Seperti yang dikatakan sebelum nya, proses manufaktur software hanya sebatas meng-copy produk dan mencetak software manual. Akibatnya, sangat jarang kecacatan produk terdeteksi pada tahap ini.

1.2 Lingkungan di mana metode SQA berada dikembangkan

Perangkat lunak yang dikembangkan oleh banyak individu dan dalam situasi yang berbeda memenuhi berbagai kebutuhan:

- Murid dan siswa mengembangkan perangkat lunak sebagai bagian dari pendidikan mereka.
- Amatir perangkat lunak mengembangkan perangkat lunak sebagai hobi.
- Profesional di bidang teknik, ekonomi, manajemen, dan bidang lainnya mengembangkan perangkat lunak untuk membantu mereka dalam pekerjaan mereka, untuk melakukan perhitungan, merangkum kegiatan penelitian dan survei, dan sebagainya.

- Profesional pengembangan perangkat lunak (analisis sistem dan pemrograman) mengembangkan produk perangkat lunak atau firmware sebagai tujuan karir profesional saat bekerja di rumah perangkat lunak atau dengan pengembangan perangkat lunak dan unit pemeliharaan (tim, departemen, dll.) besar dan kecil industri, keuangan dan organisasi lainnya.

Semua orang yang berpartisipasi dalam kegiatan ini diharuskan untuk berurusan dengan perangkat lunak masalah kualitas ("bug"). Namun, masalah kualitas di sebagian besar mereka bentuk parah mengatur pengembangan perangkat lunak profesional.

Oleh karena itu, buku ini dikhususkan untuk mendefinisikan dan memecahkan banyak perangkat lunak masalah jaminan kualitas (SQA) yang dihadapi oleh pengembangan perangkat lunak dan profesional pemeliharaan. Namun, semua jenis pengembang perangkat lunak lainnya dapat menemukan bagian-bagian dari buku yang berlaku dan direkomendasikan untuk mereka upaya pengembangan perangkat lunak sendiri.

Mari kita mulai dengan pemeriksaan lingkungan perangkat lunak profesional pengembangan dan pemeliharaan (selanjutnya disebut "lingkungan SQA"), karena adalah pertimbangan utama dalam pengembangan metodologi SQA dan penerapan. Ciri-ciri utama dari lingkungan ini adalah sebagai berikut:

(1) Kondisi kontrak. Sebagai hasil dari komitmen dan kondisi didefinisikan dalam kontrak antara pengembang perangkat lunak dan pelanggan, kegiatan pengembangan dan pemeliharaan perangkat lunak perlu mengatasi:

- Daftar persyaratan fungsional yang ditentukan oleh perangkat lunak yang dikembangkan dan pemeliharaannya harus dipenuhi.

- Anggaran proyek.
- Jadwal proyek.

Manajer proyek pengembangan dan pemeliharaan perangkat lunak membutuhkan untuk menginvestasikan sejumlah besar upaya dalam pengawasan kegiatan di untuk memenuhi persyaratan kontrak.

(2) Tunduk pada hubungan pelanggan-pemasok. Sepanjang proses pengembangan dan pemeliharaan perangkat lunak, kegiatan berada di bawah pengawasan dari pelanggan. Tim proyek harus bekerja sama terus menerus dengan pelanggan: untuk mempertimbangkan permintaannya untuk perubahan, untuk mendiskusikan kritiknya tentang berbagai aspek proyek, dan untuk mendapatkan persetujuannya untuk perubahan yang diprakarsai oleh tim pengembangan. Hubungan seperti itu tidak biasanya ada ketika perangkat lunak dikembangkan oleh profesional non-perangkat lunak.

(3) Diperlukan kerja sama tim. Tiga faktor biasanya memotivasi pembentukan: tim proyek daripada menugaskan proyek ke satu profesional:

- Persyaratan jadwal. Dengan kata lain, beban kerja yang dilakukan selama periode proyek membutuhkan partisipasi lebih dari satu orang jika proyek harus diselesaikan tepat waktu.
- Kebutuhan akan berbagai spesialisasi untuk melaksanakan proyek.
- Keinginan untuk mendapatkan manfaat dari saling mendukung dan meninjau secara profesional untuk peningkatan kualitas proyek.

(4) Kerjasama dan koordinasi dengan tim perangkat lunak lain. Pelaksanaannya proyek, terutama proyek skala besar, oleh lebih dari satu tim adalah peristiwa yang sangat umum di industri perangkat lunak. Dalam hal ini, kerjasama mungkin diperlukan dengan:

- Tim pengembangan perangkat lunak lain dalam organisasi yang sama.
- Tim pengembangan perangkat keras dalam organisasi yang sama.
- Tim pengembangan perangkat lunak dan perangkat keras dari pemasok lain.
- Tim pengembangan perangkat lunak dan perangkat keras pelanggan yang ambil bagian dalam pengembangan proyek.

Garis besar kebutuhan kerjasama, dilihat dari sudut pandang tim pengembang, ditunjukkan pada Gambar 1.1.

(5) Antarmuka dengan sistem perangkat lunak lain. Saat ini, sebagian besar sistem perangkat lunak menyertakan antarmuka dengan paket perangkat lunak lain. Antarmuka ini memungkinkan data dalam bentuk elektronik mengalir di antara sistem perangkat lunak, gratis dari memasukkan data yang diproses oleh sistem perangkat lunak lain. Satu bias mengidentifikasi jenis antarmuka utama berikut:

- Antarmuka input, di mana sistem perangkat lunak lain mengirimkan data ke sistem perangkat lunak.
- Antarmuka keluaran, tempat sistem perangkat lunak Anda mentransmisikan proses data ke sistem perangkat lunak lain.
- Antarmuka input dan output ke papan kontrol mesin, seperti dalam medis dan sistem kontrol laboratorium, peralatan pemrosesan logam, dll.

Paket perangkat lunak pengolah gaji memberikan contoh yang baik dari tipikal antarmuka input dan output ke paket perangkat lunak lain – lihat Gambar 1.2. Pertama mari kita lihat antarmuka input. Untuk menghitung gaji, satu membutuhkan informasi kehadiran karyawan, seperti yang ditangkap oleh waktu

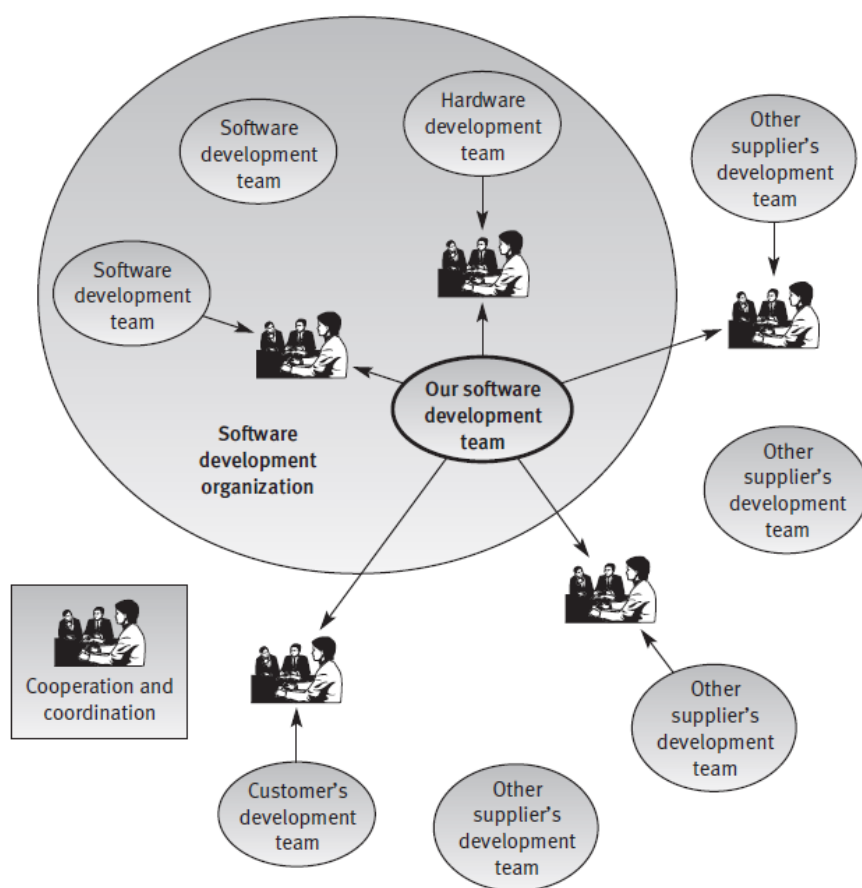
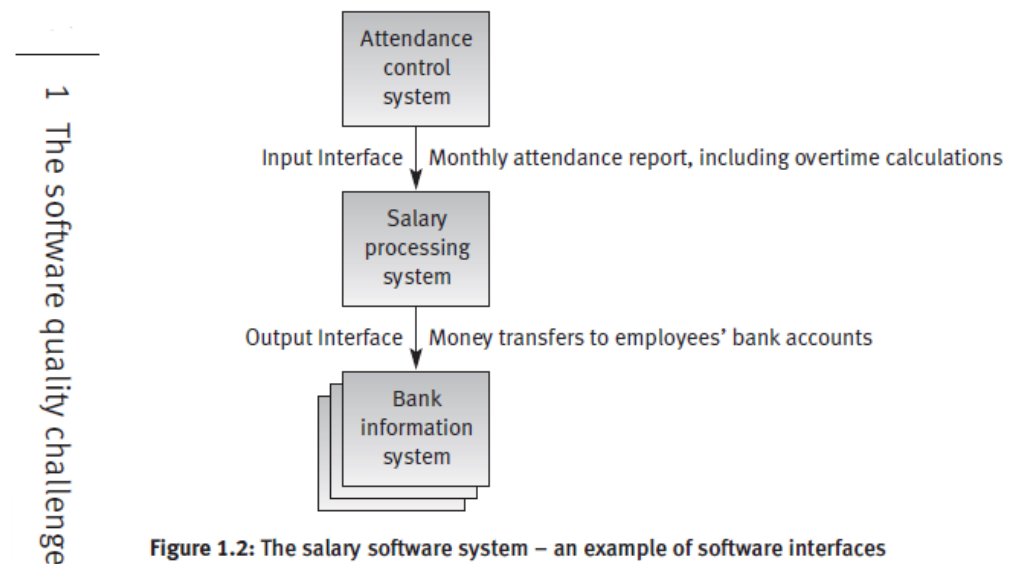


Figure 1.1: A cooperation and coordination scheme for a software development team of a large-scale project

jam ditempatkan di pintu masuk gedung kantor dan diproses nanti oleh sistem perangkat lunak kontrol kehadiran. Sebulan sekali, informasi ini (daftar kehadiran termasuk data lembur)

ditransmisikan secara elektronik dari sistem kontrol kehadiran hingga pemrosesan gaji sistem. Transmisi informasi ini mewakili antarmuka input untuk sistem perangkat lunak pengolah gaji; pada saat yang sama itu mewakili antarmuka output ke sistem kontrol kehadiran. Sekarang, mari kita periksa antarmuka keluaran sistem kami. Salah satu output dari gaji sistem pemrosesan adalah daftar gaji "bersih", setelah dikurangi pajak penghasilan dan hal-hal lain yang harus dibayarkan kepada karyawan. Daftar ini, termasuk rincian rekening bank karyawan, harus dikirim ke bank. Itu transmisi daftar pembayaran gaji dilakukan secara elektronik, yang mewakili antarmuka output untuk sistem pemrosesan gaji dan input antarmuka untuk sistem rekening bank.



(6) Kebutuhan untuk terus melaksanakan proyek meskipun anggota tim perubahan. Sangat umum bagi anggota tim untuk meninggalkan tim selama periode pengembangan proyek, baik karena promosi ke yang lebih tinggi pekerjaan tingkat, beralih

majikan, transfer ke kota lain, dan sebagainya. Pemimpin tim kemudian harus mengganti anggota tim yang pergi juga oleh karyawan lain atau oleh karyawan yang baru direkrut. Bagaimanapun caranya banyak upaya diinvestasikan dalam melatih anggota tim baru, "pertunjukan" harus dilanjutkan", yang berarti jadwal kontrak proyek asli tidak akan berubah.

(7) Kebutuhan untuk terus melakukan pemeliharaan perangkat lunak untuk waktu yang lama Titik. Pelanggan yang mengembangkan atau membeli sistem perangkat lunak mengharapkan untuk terus menggunakannya untuk waktu yang lama, biasanya selama 5-10 tahun. Selama masa layanan, kebutuhan akan pemeliharaan pada akhirnya akan muncul. Di kebanyakan kasus, pengembang diharuskan untuk menyediakan layanan ini secara langsung. "Pelanggan" internal, dalam kasus di mana perangkat lunak telah dikembangkan in-house, berbagi harapan yang sama mengenai pemeliharaan perangkat lunak selama periode layanan sistem perangkat lunak.

Karakteristik lingkungan menciptakan kebutuhan yang intensif dan berkesinambungan upaya manajerial sejajar dengan upaya profesional yang harus diinvestasikan untuk menjamin kualitas proyek, dengan kata lain untuk menjamin keberhasilan proyek.

Ringkasan karakteristik utama lingkungan SQA ditampilkan di Bingkai 1.2.

Sejumlah besar perangkat lunak serta pengembangan firmware tidak dilakukan dengan tunduk pada kontrak formal

atau hubungan formal pelanggan-pemasok, seperti yang disebutkan dalam dua karakteristik lingkungan SQA pertama. Ini jenis aktivitas biasanya menyangkut perangkat lunak yang dikembangkan sendiri untuk penggunaan internal

BINGKAI 1.2 || Ringkasan karakteristik utama lingkungan SQA

1. Dikontrak
2. Ketundukan pada hubungan pelanggan-pemasok
3. Persyaratan untuk kerja tim
4. Perlu kerjasama dan koordinasi dengan tim pengembangan lainnya
5. Kebutuhan akan antarmuka dengan sistem perangkat lunak lain
6. Perlu terus menjalankan proyek saat tim berubah
7. Perlu terus memelihara sistem perangkat lunak selama bertahun-tahun

6. Perlu terus menjalankan proyek saat tim berubah

7. Perlu terus memelihara sistem perangkat lunak selama bertahun-tahun

atau untuk pemasaran sebagai paket perangkat lunak dan pengembangan firmware internal. Hubungan antara departemen pemasaran yang memulai dan mendefinisikan kualifikasi produk baru dan perangkat lunak internal masing-masing departemen pengembangan sering menyerupai kontrak dan pelanggan-hubungan pemasok. Hal yang sama berlaku untuk permintaan internal untuk perangkat lunak baru sistem atau untuk meningkatkan perangkat lunak atau firmware saat ini menjadi diimplementasikan oleh departemen perangkat lunak organisasi.

Hubungan yang sebenarnya antara "pelanggan" internal dan departemen pengembangan ditemukan sangat bervariasi ketika diukur dengan skala formal-informal. Beberapa manajer mengklaim bahwa semakin dekat hubungan dengan bentuk formal, semakin besar prospek keberhasilan proyek.

