

ANALISA PENGEMBANGAN SOFTWARE QUALITY ASSURANCE PLAN (SQAP) UNTUK PENGEMBANG SOFTWARE KECIL DI INDONESIA DENGAN MENGACU PADA DOKUMEN IEEE STD 730TM-2002 DAN ISO/IEC 90003:2004

Oleh
I Ketut Purnamawan
Jurusan Manajemen Informatika, FTK, UNDIKSHA

ABSTRAK

Kualitas rata-rata projek software masih rendah, ini dapat dilihat dari hasil survey yang menyatakan bahwa 23 persen dari pojek sortware yang disurvei dinyatakan gagal, 49 persen tidak sesuai dengan requirement, dan hanya 28 persen yang dinyatakan sukses. Di lain pihak, Software Quality Management belum digunakan secara tepat pada sebagian besar pengembangan software. Paper ini berisi suatu analisa pengembangan suatu Software Quality Assurance Plan (SQAP) bagi pengembang-pengembang software kecil di Indonesia. Analisa dilakukan dengan mengenali karakteristik pengembang-pengembang kecil di Indonesia, lalu menyesuaikan standar-standar yang ada dengan karakteristik tersebut. Hasil analisa berupa suatu rekomendasi standar-standar yang harus dimasukkan, dan standar-standar yang dapat tidak dimasukkan ke dalam SQAP. Dokumen standar yang digunakan adalah IEEE Std 730TM-2002[ieee] dan ISO/IEC 90003:2004.

Kata kunci : SQAP, SQA, Pengembangan Software, Pengembang Software

ABSTRACT

The average of quality of software development projects is still low. A survey states that 23 percent of surveyed project has failed, 49 percent are not in accordance with the requirement, and only 28 percent declared a success. On the other hand, Software Quality Management have not been used appropriately in the majority of software development. This paper contains an analysis of the development of a Software Quality Assurance Plan (SQAP) for small software developers in Indonesia. Analysis carried out by identifying the characteristics of small developers in Indonesia, then adjust the existing standards with those characteristics. The results of the analysis is a recommendation of standards that must be included, and that can be not included in SQAP. Standard document used is the IEEE Std 730TM -2002 [IEEE] and ISO / IEC 90003:2004.

Kata kunci : SQAP, SQA, Software Development, Software Developers

Analisa Pengembangan Software Quality Assurance Plan.....(I Ketut Purnamawan)

1. PENDAHULUAN

Software Quality Management dipandang sebagai aktifitas paling menantang dalam *life cycle* sebuah proyek. Namun *quality management* belum digunakan secara tepat pada sebagian besar pengembangan *software* dan *IT service organisations*. Itu semua terjadi lebih dikarenakan oleh kurangnya pengetahuan tentang *standard quality management*, dan yang lebih penting, ketidakadaannya petunjuk dalam mendefinisikan patokan *feasibility* spesifik organisasi terhadap aktifitas *software quality management*[3].

Pada dua dekade terakhir, proyek software sering kali gagal memenuhi harapan pengguna, yaitu biasanya terlambat dalam perampungan, dan kebanyakan melebihi anggaran yang ditetapkan. Standish Group (2000) melakukan studi terhadap 13.522 proyek dalam sebuah survey yang dinamai EXTREME CHAOS. Studi ini menyatakan 23 persen dari proyek yang di survey gagal, 49 persen tidak sesuai dengan *requirement*, dan hanya 28 persen sukses[4].

Di Indonesia sendiri, semakin pesatnya perkembangan teknologi informasi (TI) mengakibatkan kebutuhan terhadap software meningkat. Hal ini mengakibatkan banyak bermunculan pengembang-pengembang software, baik pengembang besar, maupun pengembang kecil. Pengembang-pengembang kecil yang terbentuk banyak diantaranya tidak disertai oleh sistem manajemen dan sumber daya yang memadai, baik sumber daya manusia, maupun infrastruktur. Hal ini mengakibatkan kualitas software yang dihasilkan rendah.

Pada paper ini akan dilakukan analisa pengembangan *Software Quality Assurance Plan* (SQAP) untuk pengembang software kecil di Indonesia dengan mengacu kepada dokumen IEEE Std 730™-2002[1] dan ISO/IEC 90003:2004[2]. Hasil analisa ini diharapkan nantinya bisa dipergunakan sebagai acuan dalam pengembangan SQAP oleh pengembang software kecil di Indonesia.

2. PEMBAHASAN

2.1 IEEE Std 730™-2002

IEEE Std 730™-2002 adalah standar SQAP yang dikeluarkan oleh Institute of Electrical and Electronics Engineers (IEEE). Menurut IEEE Std 730™-2002, suatu SQAP harus mempunyai 16 bagian sebagai berikut.

1. *Purpose*: Bagian ini menggambarkan tujuan dan jangkauan wilayah SQAP, berisi daftar item-item software yang dicakup oleh SQAP dan maksud kegunaan software, dan mengkondisikan porsi *software life cycle* yang dicakup oleh SQAP untuk setiap item software tertentu.
2. *Reference documents*: Bagian ini berisi daftar dokumen-dokumen yang diacu oleh SQAP.
3. *Management*: Bagian ini mendeskripsikan struktur organisasi proyek, tugas-tugasnya, perannya dan tanggung jawab nya. Bagian ini juga berisi perkiraan sumberdaya dan biaya yang dihabiskan untuk dalam melaksanakan pekerjaan *quality assurance* dan *quality control*.
4. *Documentation*: Bagian ini salah satunya berisi minimum documentation requirement yang terdiri dari software requirements description (SRD), software design description (SDD), verification and validation plans, verification results report and validation results report, user documentation, dan software configuration management plan (SCMP).
5. *Standards, practices, conventions, and metrics*: Bagian ini berisi standar-standar yang digunakan, kebiasaan-kebiasaan, kesepakatan-kesepakatan, quality requirements, dan metrics yang.
6. *Software reviews*: Bagian ini salah satunya berisi standar *software review* minimal yang terdiri dari software specification review (SSR), architecture design review (ADR), detailed design review (DDR), verification and validation plan review, functional audit, physical audit, in-process audits, managerial reviews, software configuration management plan review (SCMPR), dan post-implementation review.
7. *Test*: Bagian ini menyatakan semua tes yan tidak termasuk dalam software verification and validation plan.

8. *Problem reporting and corrective action*: Menggambarkan praktek dan prosedur yang diikuti untuk pelaporan, pelacakan, dan pemecahan masalah-masalah yang diidentifikasi baik didalam item-item software, maupun pada proses pengembangan dan perawatan software.
9. *Tool, techniques, and methodologies*: Bagian ini menyatakan software tools, teknik, dan metode-metode yang digunakan untuk mendukung proses-proses SQA.
10. *Media control*.
11. *Supplier control*.
12. *Records collection, maintenance, and retention*.
13. *Training*.
14. *Risk management*.
15. *Glossary*.
16. *SQAP change procedure and history*.

Pada bagian ini, dokumen IEEE Std 730TM-2002 tidak dibahas secara lebih mendetail. Bagian-bagian yang diperlukan dalam proses analisa akan dibahas kembali secara lebih mendetail. Contoh dokumen yang merujuk dokumen ini adalah NASA-STD-8739.8[6].

2.2 ISO/IEC 90003:2004

Dokumen ISO/IEC 90003:2004 adalah suatu standar *quality management* untuk *software* komputer. Dokumen ini menggantikan dokumen ISO 90003:1997. ISO/IEC 90003:2004 menerangkan bagaimana ISO 9001:2000 bisa diterapkan pada software. ISO 9001:2000 sendiri merupakan standar *quality management system*. ISO/IEC 90003:2004 dibagi menjadi beberapa point. Point kesatu sampai point ketiga berisi penjelasan mengenai topik-topik yang berkaitan. Point keempat berisi standar tentang kebutuhan-kebutuhan dan pedoman-pedoman yang berhubungan dengan quality system. Point kelima berisi standar kebutuhan-kebutuhan dan pedoman-pedoman manajemen. Point keenam berisi standar kebutuhan-kebutuhan dan pedoman-pedoman terhadap sumberdaya. Point ketujuh berisi standar

kebutuhan-kebutuhan dan pedoman-pedoman dalam proses realisasi produk. Point ini merupakan point yang paling banyak berisi pedoman-pedoman. Point kedelapan berisi standar kebutuhan-kebutuhan dan pedoman-pedoman yang berhubungan dengan perbaikan.

Dokumen ISO/IEC 90003:2004 tidak dibahas secara detail pada bagian ini. Bagian-bagian yang diperlukan dalam analisa akan dibahas kembali secara lebih mendetail sesuai dengan keperluan.

2.3 Karakteristik Pengembang Software Kecil di Indonesia

Pengembang-pengembang software kecil di Indonesia dibentuk dengan sumber daya yang tidak besar, baik sumber daya pendanaannya maupun sumber daya manusianya. Dengan modal yang tidak besar, software yang diproduksi adalah software-software bersekala kecil. Pasar mereka adalah instansi-instansi pemerintah dan lembaga-lembaga swasta kecil. Para pekerjanya bisa masuk dan keluar organisasi dalam jangka waktu yang tidak lama. Hal ini menyebabkan struktur organisasinya tidak tetap untuk waktu yang lama. Bahkan banyak pengembang yang tidak jelas struktur organisasinya.

Dari segi *customer*, pengembang-pengembang kecil ini berhadapan dengan instansi-instansi pemerintah dan lembaga-lembaga swasta kecil. Banyak diantara instansi pemerintah dan lembaga swasta ini tidak punya divisi khusus yang menangani masalah teknologi informasi.

2.4 Analisa pengembangan SQAP

Karakteristik pengembang software kecil kalau dirangkum adalah sebagai berikut.

- Sumber daya pendanaan kecil.
- Sumber daya manusia kecil
- Struktur organisasi tidak konsisten.
- Pelanggannya adalah orang-orang yang pengetahuan tentang teknologi informasi nya sangat minim.

Model SQAP yang akan dikembangkan harus mempertimbangkan hal-hal ini. Untuk selanjutnya karakteristik ini akan dijadikan konstrain dalam pengembangan model SQAP.

Dengan sumber daya pendanaan yang kecil maka proses-proses SQA yang membutuhkan biaya besar namun kurang mendesak untuk pengembang kecil tidak bisa dilakukan. Dengan sumber daya manusia yang kecil, maka proses-proses SQA yang memerlukan sumber daya manusia besar dan memerlukan keahlian khusus tidak bisa dilakukan. Struktur organisasi yang tidak konsisten mengakibatkan model SQAP yang dikembangkan harus bisa fleksibel dalam urusan yang ada kaitannya dengan struktur organisasi

Yang perlu dijadikan perhatian utama adalah fakta bahwa pelanggan yang dihadapi pengembang-pengembang kecil ini adalah orang-orang yang pengetahuan tentang teknologi informasi nya sangat minim, dan fakta survey yang menyatakan bahwa 49 persen proyek software tidak sesuai dengan *requirement*. Fakta survey memang terbukti. Banyaknya proyek software yang berlarut-larut pengerejaanya karena harapan user belum tercapai. Hal ini terjadi sebagian besar karena tidak samanya persepsi *requirement* antara pengguna dan pengembang. Berangkat dari hal ini, maka proses pendefinisian requirement dalam model SQAP yang akan dibentuk harus diutamakan.

Struktur organisasi yang tidak jelas membuat pembagian tugas menjadi tidak tegas. Hal ini mengakibatkan garis tanggung jawab menjadi tidak jelas. Dan masalah yang lebih besar yang ditimbulkan oleh hal ini adalah, pekerjaan yang dihasilkan menjadi tidak bersih. Seorang programmer kadang kala harus merangkap sebagai seorang desainer, padahal dia tidak punya kemampuan yang memadai untuk itu. Seorang programmer bisa harus mengerjakan beberapa modul yang berbeda pada waktu bersamaan. Hal ini kurang bagus, karena dapat memecah konsentrasi pekerjaan. Masalah ini sebenarnya bukan hanya ditimbulkan oleh struktur organisasi yang tidak jelas. Faktor kurangnya sumber daya manusia juga berperan penting dalam menimbulkan masalah ini.

Permasalahan utama lain adalah kurangnya pengetahuan pengembang kecil terhadap pentingnya desain suatu sistem. Para pengembang kecil menganggap

desain sistem adalah sesuatu yang remeh, bahkan menganggap bahwa itu adalah sesuatu yang bisa tidak ada. Mereka hanya peduli terhadap kode program. Mereka tidak menyadari pentingnya suatu desain sistem dalam proses pengemabangan software. Mereka tidak menyadari bahwa desain sistem dapat membuat pembagian tugas menjadi lebih jelas. Desain sistem juga dapat membuat kerja programmer menjadi jauh lebih efisien. Tidak adanya suatu desain yang jelas juga dapat menyebabkan penyelesaian software menjadi lama danberlarut-larut, dan kemampuan untuk memenuhi *user requirement* menjadi rendah.

Masalah *system requirement* dan *design sistem* adalah masalah utama yang menyebabkan rendahnya kualitas software produksi pengembang kecil. Untuk itu masalah *system requirement* dan *design sistem* akan dijadikan masalah utama yang harus ditanggulangi dalam membangun model SQAP ini. Permasalahn utama ini dan konstrain-konstrain yang lain akan menjadi acuan dalam menentukan bagian mana dari standar-standar yang ada yang harus dimasukkan dalam model SQAP ini, dan bagian mana yang tidak perlu atau tidak mungkin untuk dimasukkan.

Untuk menanggulangi masalah *system requirement* dan *design sistem* maka sebagian dari *minimum documentation requirement* yang terdapat pada point 4 di dalam IEEE Std 730TM-2002 yaitu SRD dan SDD, dan sebagian dari *minimum software review* yang terdapat pada point 6 di dalam IEEE Std 730TM-2002 yaitu SSR, ADR dan DDR mutlak harus ada dalam model SQAP ini. Point 5.2, 6.1, 7.1, dan 7.2 dari ISO/IEC 90003:2004 yang berkaitan dengan *system requirement* serta point 7.3 dari ISO/IEC 90003:2004 yang berkaitan dengan desain sistem mutlak harus dilaksanakan, sehingga harus dimasukkan kedalam SQAP ini.

Karena sumber pendanaan yang kecil, maka, proses-proses lain yang seharusnya memerlukan pendanaan besar harus disesuaikan, atau bahkan memungkinkan untuk tidak dimasukkan kedalam SQAP ini. Proses-proses audit yang memerlukan biaya besar dapat tidak dimasukkan ke dalam SQAP. Proses software inspection yang juga dikenal memerlukan biaya yang cukup besar dapat tidak dimasukkan ke dalam SQAP, atau dicari alternatif dengan megggunakan metode yang sudah disimpelkan dan memungkinkan dilakukan dengan dana terbatas, seperti metode yang dikembangkan oleh Mishra [5], dibanding dengan

menggunakan metode yang dikembangkan Fagan[7] yang memerlukan biaya tinggi. Proses quality measurement, yang memerlukan biaya dan waktu yang cukup besar dapat diadakan dari SQAP ini. Untuk menanggulangi struktur organisasi yang tidak konsisten, maka point 3 dari IEEE Std 730™-2002 harus dibuat fleksibel dan bisa diubah kapanpun.

Selain point-point yang sudah disebutkan. Point point lain yang penting dalam peningkatan kualitas dan harus dimasukkan ke dalam SQAP ini adalah point 6.2 ISO/IEC 90003:2004 yang berkaitan dengan sumber daya manusia yang berkualitas, dan point 6.4 ISO/IEC 90003:2004 yang berkaitan dengan pembentukan lingkungan kerja yang berkualitas. Pembentukan lingkungan kerja yang berkualitas dapat diciptakan dengan tidak memerlukan biaya yang besar, namun mempunyai efek manfaat yang besar. Lingkungan kerja yang berkualitas dapat meningkatkan efektifitas kerja, dan dapat menarik sumberdaya-sumberdaya manusia yang berkualitas.

Point-point selain yang dibahas diatas dapat dimasukkan maupun tidak dimasukkan kedalam SQAP. Dimasukkannya point-point ini bisa disesuaikan dengan kondisi yang dihadapi.

3. PENUTUP

Dalam mengembangkan suatu SQAP yang diperuntukkan bagi pengembang-pengembang software kecil di Indonesia, Point-point standar yang berkaitan dengan *system requirement* dan *design system* mutlak harus dimasukkan. Kalau SQAP yang dikembangkan mengacu pada dokumen IEEE Std 730™-2002 dan ISO/IEC 90003:2004, maka sebagian dari *minimum documentation requirement* yang terdapat pada point 4 di dalam IEEE Std 730™-2002 yaitu SRD dan SDD, dan sebagian dari *minimum software review* yang terdapat pada point 6 di dalam IEEE Std 730™-2002 yaitu SSR, ADR dan DDR mutlak harus ada dalam model SQAP ini. Point 5.2, 6.1, 7.1, 7.2, dan 7.3 dari ISO/IEC 90003:2004 mutlak harus dilaksanakan, sehingga harus dimasukkan kedalam SQAP.

Proses-proses yang memerlukan biaya besar dan tidak terlalu urgent bagi pengembang kecil dapat tidak dimasukkan kedalam SQAP. Proses-proses tersebut

diantaranya adalah proses-proses audit, software inspection, dan quality measurement.

Point point lain yang penting dalam peningkatan kualitas dan harus dimasukkan ke dalam SQAP ini adalah point 6.2 ISO/IEC 90003:2004 yang berkaitan dengan sumber daya manusia yang berkualitas, dan point 6.4 ISO/IEC 90003:2004 yang berkaitan dengan pembentukan lingkungan kerja yang berkualitas.

Point-point selain yang dibahas diatas dapat dimasukkan maupun tidak dimasukkan kedalam SQAP. Dimasukkannya point-point ini bisa disesuaikan dengan kondisi yang dihadapi.

DAFTAR PUSTAKA

- [1] *IEEE Standard for Software Quality Assurance Plans*, IEEE Std 730™-2002, 2002.
- [2] *ISO IEC 90003 2004 Software Standard Translated Into Plain English*. [Online]. Bisa di akses di: <http://www.praxiom.com/iso-90003.htm>
- [3] K. M. M. Hussain, *Software Quality Management Planning Model (SQuaMaPMo)*, M&H Informatics (BD) Ltd. Bangladesh, Januari 2007.
- [4] R. C Nienaber dan A. Barnard, *Software Quality Management Supported by Software Agent Technology*, University of Sout Africa.
- [5] D. Mishra, A. Mishra, *Simplified Software Inspection Process in Compliance with International Standards*, Elsevier B.V., Nopember 2008.
- [6] *Software Assurance Standard, NASA Technical Standard*, NASA-STD-8739.8.
- [7] M. E. Fagan, *Design and Code Inspection to reduce errors in program development*, IBM System Journal 15 (3) (1976) 182-211.
