

Requirement Engineering

Requirements engineering adalah fase terdepan dari proses rekayasa perangkat lunak (software engineering), dimana software requirements (kebutuhan) dari user (pengguna) dan customer (pelanggan) dikumpulkan, dipahami dan ditetapkan.

Requirements engineering adalah suatu pekerjaan yang sangat penting.

Kebanyakan kegagalan pengembangan software disebabkan karena adanya :

1. Ketidakkonsistenan (inconsistent),
2. Ketidaklengkapan (incomplete),
3. Ketidakbenaran (incorrect) dari requirement specification (spesifikasi kebutuhan).

Contoh Kasus :

1. Customer (pelanggan) yang datang kepada kita untuk mengerjakan sebuah proyek pengembangan software, adalah ibarat seseorang yang mengatakan kepada kita, “Tolong buatlah saya batu”.
2. Ketika kita memberikan kepadanya sebuah batu, dia akan melihatnya sebentar dan mengatakan kepada kita, “Ya terima kasih, tapi sebenarnya yang saya inginkan adalah sebuah batu kecil berwarna biru”.
3. Dan ketika kita bawakan untuknya batu kecil berwarna biru, dia mengatakan bahwa yang diinginkan adalah yang “bentuknya bulat”.

4. Demikian seterusnya proses iterasi (iteration) terjadi berulang kali sampai akhirnya kita dapatkan yang sebenarnya diinginkan customer kita adalah “batu pualam kecil berwarna biru”. Meskipun mungkin sebenarnya bukan “tepat yang diinginkan”, tapi paling tidak “paling dekat” dengan yang diinginkan customer.
5. Dan mungkin saja terjadi, customer kita mengubah pikiran tentang requirements pada saat proses interaksi dengan pengembang terjadi (dari iterasi pertama yang sekedar batu, sampai iterasi terakhir yang menghasilkan batu pualam kecil berwarna biru).

Akibatnya :

Frustrasi terjadi baik di pihak pengembang maupun customer.

Pengembang frustrasi karena merasakan perbedaan signifikan antara requirements pertama yang dijelaskan customer dengan yang sebenarnya diinginkan customer, belum lagi waktu dan biaya besar sudah dikeluarkan pengembang dalam tiap iterasi.

Di lain pihak, customer juga frustrasi karena merasa sudah menjelaskan dengan baik dan pengembang tidak bisa memahami yang diinginkan, sehingga memerlukan waktu yang sangat lama.

Fenomena senada dengan masalah batu diatas sering juga disebut dengan “Yes, But Syndrome” (That is what I meant, but not exactly what I meant) dan “Undiscovered Ruins Syndrome” (Now that I see it, I have another requirement to add)

Dari ilustrasi diatas, muncul keinginan untuk menerapkan pendekatan engineering (engineering approach) untuk memecahkan masalah tersebut. yang akhirnya membawa arus deras kemunculan cabang ilmu requirements engineering.

KESIMPULAN

Hasil dari fase requirements engineering terdokumentasi dalam requirements specification.

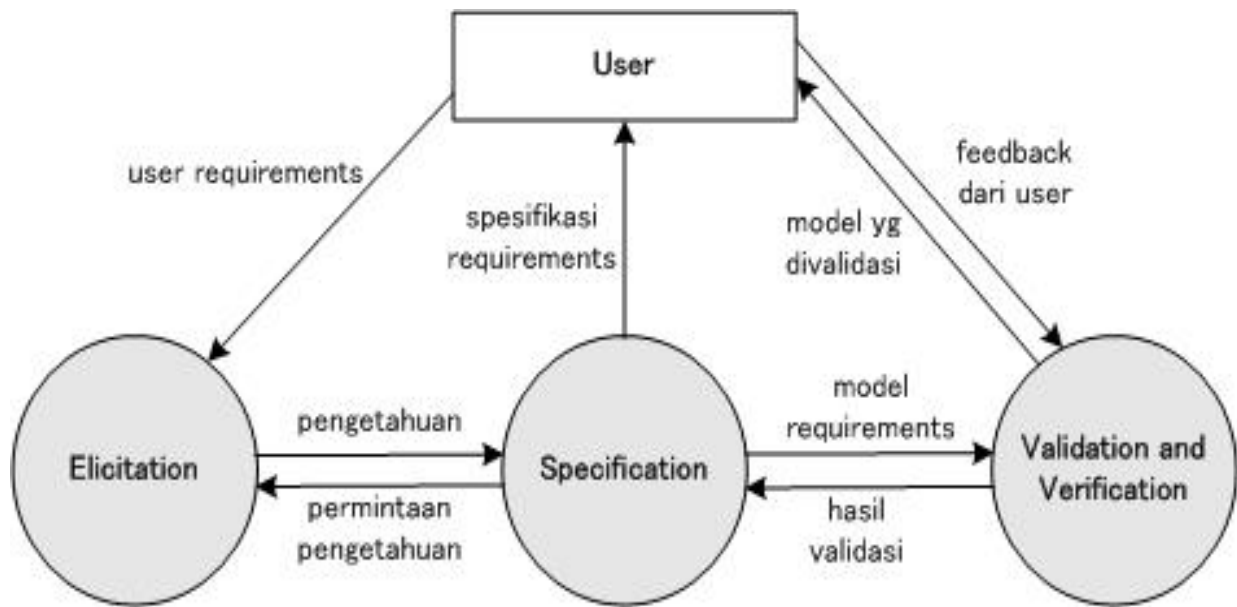
Requirements specification berisi kesepakatan bersama tentang permasalahan yang ingin dipecahkan antara pengembang dan customer, dan merupakan titik start menuju proses berikutnya yaitu software design.

Sistemisasi proses negosiasi pengembang dan customer dalam requirements engineering dibagi dalam 3 proses besar yaitu :

1. Elicitation,
2. Specification,
3. Validation and verification.

Formula ini kemudian juga dikenal dengan nama *The Three Dimensions of Requirements Engineering*.

Proses requirements engineering ini dilakukan secara iterasi dengan mengakomodasi adanya feedback dari customer (user).



Requirements Elicitation

Adalah proses mengumpulkan dan memahami requirements dari user. Kadang masalah yang muncul berakar dari gap masalah knowledge domain (perbedaan disiplin ilmu yang dimiliki). Customer adalah expert pada domain yang softwarenya ingin dikembangkan (domain specialist), dilain pihak sang pengembang (requirements analyst) adakalanya sama sekali buta terhadap knowledge domain tersebut, meskipun tentu memahami dengan benar bagaimana sebuah software harus dikembangkan. Gap knowledge domain tersebut yang diharapkan bisa diatasi dengan adanya interaksi terus menerus dan berulang (iterasi) antara pengembang dan customer. Proses interaksi tersebut kemudian dimodelkan menjadi beberapa teknik dan metodologi diantaranya adalah interviewing, brainstorming, prototyping, use case, dsb.

Requirements Specification

Setelah masalah berhasil dipahami, pengembang mendeskripsikannya dalam bentuk dokumen spesifikasi dokumen. Spesifikasi ini berisi tentang fitur dan fungsi yang diinginkan oleh customer, dan sama sekali tidak membahas bagaimana metode pengembangannya. IEEE mengeluarkan standard untuk dokumen spesifikasi requirements yang terkenal dengan nama IEEE Recommended Practice for Software Requirements Specifications [IEEE-830]. Dokumen spesifikasi requirements bisa berisi functional requirements, performance requirements, external interface requirements, design constraints, maupun quality requirements.

Requirements Validation and Verification

Setelah spesifikasi requirements berhasil dibuat, perlu dilakukan dua usaha: Validation (validasi), yaitu proses untuk memastikan bahwa requirements yang benar sudah ditulis. Verification (verifikasi), yaitu proses untuk memastikan bahwa requirements sudah ditulis dengan benar. Proses validasi dan verifikasi ini melibatkan customer (user) sebagai pihak yang menilai dan memberi feedback berhubungan dengan requirements.

REFERENSI

- [Leffingwell-00] Dean Leffingwell and Don Widrig, Managing Software Requirements: A Unified Approach, Addison Wesley, 2000.
- [Romi-02] Romi Satria Wahono and Jingde Cheng, Extensible Requirements Patterns of Web Application, IEEE International Symposium on Cyber Worlds (CW 2002), Japan, 2002.
- [Romi-03] Romi Satria Wahono, Analyzing Requirements Engineering Problems, IECI Japan Workshop 2003 (IJW-2003), Japan, 2003.
- [Standish-94] The Standish Group, Charting the Seas of Information Technology Chaos, The Standish Group International, 1994.
- [Zave-97] Pamela Zave, Classification of Research Efforts in Requirements Engineering, ACM Computing Surveys, 29(4), pp. 315-321, 1997.