

Metode rekayasa ulang perangkat lunak untuk meningkatkan kualitas dengan mengadopsi Software Craftmanship : studi kasus tim pengembang berskala kecil = Software reengineering methods for quality improvement by Software Craftmanship adoption : a case study of Small-Scale development teams

Ahmadi, author

Deskripsi Lengkap: <https://lib.ui.ac.id/detail?id=20524773&lokasi=lokal>

Abstrak

Banyak perangkat lunak yang berkembang cepat karena tuntutan bisnis tanpa melalui proses yang matang sehingga berkualitas buruk dan menjadi lebih sulit untuk dikelola. Untuk kasus seperti ini bisa diperbaiki dengan melakukan rekayasa ulang dengan memperhatikan praktik yang baik dalam proses pengembangan perangkat lunak tersebut. PT RAKITEK sebagai perusahaan dengan tim berskala kecil yang mengembangkan produk perangkat lunak CRM untuk multi tenant juga mengalami masalah evolusi perangkat lunak, sehingga berkualitas buruk. Permasalahan dari perangkat lunak tersebut adalah sulitnya pengelolaan karena belum mendukung multi-tenancy dan menghasilkan defect yang susah diperbaiki. Software craftsmanship sebagai ideologi yang menekankan penggunaan praktik keunggulan teknis untuk menghasilkan perangkat lunak yang berkualitas dan mudah dikelola bisa diadopsi untuk mengatasi masalah ini. Namun, untuk mengadopsi software craftsmanship dengan baik, perlu merancang metode rekayasa ulang perangkat lunak yang tepat sesuai kondisi perusahaan. Penelitian ini bertujuan merancang metode rekayasa ulang perangkat lunak yang mengadopsi software craftsmanship dan menerapkan praktik agile dengan menggunakan metodologi extreme programming yang menekankan keunggulan kualitas perangkat lunak (source code). Faktor kualitas perangkat lunak yang perlu dicapai dari rancangan metode rekayasa ulang adalah maintainability, reliability, testability, dan reusability. Penelitian ini menghasilkan RUPL-SC, sebuah metode Rekayasa Ulang Perangkat Lunak yang mengadopsi Software Craftmanship, dan hasil pengujian kualitas perangkat lunak yang diperbaiki dengan metode tersebut. Pengujian efektifitas metode RUPL-SC dilakukan dengan studi kasus rekayasa ulang perangkat lunak CRM yang dikembangkan PT RAKITEK. Tim pengembang menggunakan alat bantu SonarQube untuk menilai source code yang ditulis, sehingga kualitas perangkat lunak dapat terus dipantau seiring dengan perkembangan perangkat lunak. Hasil pengujian metode berhasil memperbaiki permasalahan multi tenancy dan meningkatkan kualitas perangkat lunak.

.....Much software evolves fast, sacrifices the quality, and becomes more challenging to maintain.

Reengineering by considering the good practice in the software development process is a solution for cases like this. PT RAKITEK, a company with a small-scale team that develops CRM software products for multi-tenant, also experiences this software evolution problem, resulting in poor quality. The main problem is that it is difficult to manage because it does not support multi-tenancy and produces complex defects to avoid. Software craftsmanship as an ideology that emphasizes technical excellence practices to deliver quality and easy-to-manage software can be adopted to overcome this problem. However, it is necessary to design software reengineering methods that consider company conditions to adopt software craftsmanship smoothly. This study aims to develop a software reengineering method that adopts software craftsmanship and applies agile practices using an extreme programming methodology that emphasizes the power of

software quality (source code). The reengineering method design should achieve software quality factors such as maintainability, reliability, testability, and reusability. This research produces RUPL–SC, a Software reengineering method that adopts software craftsmanship, and the results of software quality testing improved by that method. The RUPL–SC testing method's effectiveness is shown using a case study of CRM software reengineering developed by PT RAKITEK. The development team uses the SonarQube tool to assess the source code to monitor the quality of the software in the software development. The method test results show that it succeeded in fixing the multi-tenancy problem and improving the quality of the software.