

Strategies to Reduce Rework in Software Development on IT Organization Case Study of: PT. XYZ

Nurul Kholis Kurniawan

Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia

Article Information

Received: 4 July 2018
Accepted: 5 August 2018
Published: 3 October 2018
DOI: 10.33555/ejaict.v5i2.58

Corresponding Author:

Nurul Kholis Kurniawan
Email:

ISSN 2355-1771

ABSTRACT

Today's world, developing an application that meets exactly the need of organizational business is getting tougher and more challenging, with the additional complexity of business flow and the increasing and diversity of the data that an organization has. In addition, the customers are more demanding than before in term of using an information system or software application and within an instant result which gives an extra pressure for those who work as an application developer. Rework of the software application tends to contain very negative impact to the company because it is impacting the cost for the organization. This paper is trying to resolve the problem that causes a lot of rework occurred in the case study provided. By finding out the root cause of the problem, the literature study is conducted in order to have some recommendations to be applied to the case study given. After having a discussion by conducting interview session with the expert, the result in form of a recommendation is then defined. As the final result, it is decided that the best and most appropriate recommendation to be applied is by implementing System Configuration Management (SCM) and the SMC tools chosen is Team Foundation Server or usually called as TFS.

Keywords: Software Development, System Configuration Management, Team Foundation Server

1. Introduction

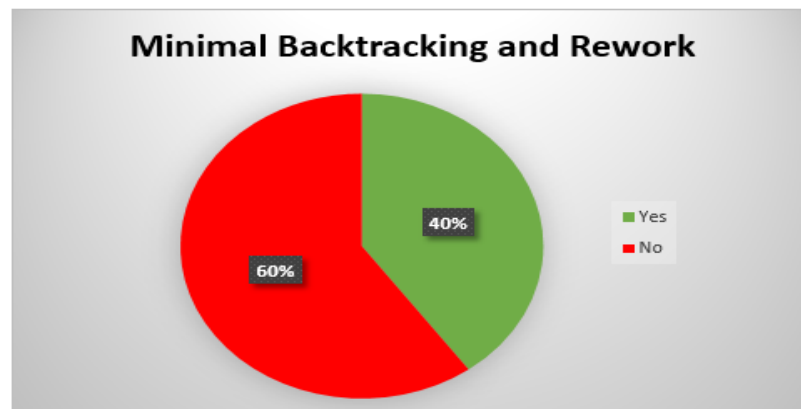
Today's world, developing an application that meets exactly the need of organizational business is getting tougher and more challenging, with the additional complexity of business flow and the increasing and diversity of the data that an organization has. In addition, the customers are more demanding than before in term of using an information system or software application and within an instant result which gives an extra pressure for those who work as an application developer. This leads the software development team cause a lot of rework in implementing software application before being delivered to the customer. Carnegie Mellon Software Engineering Institute stated that 60-80% of the cost in software development are spent in doing a rework (Nielsen P.D.). While according to Boehm's (1987), his research shows that the cost of the rework are about 40-50% of all software development expenditure (Ramdoo, V. and Huzoore, G., 2015). In software development activity, rework becomes an unavoidable thing to do. According to (Ramdoo, V. and Huzoore, G., 2015), they described a rework in software development as an additional effort of redoing a process or activity that was incorrectly implemented in the first instance or due to changes in requirement from clients. The rework normally comes out from some problems, such as an error or failure that occurred in particular feature of the application being developed, miscommunication between the client and the development team during requirement gathering, or poor coordination between one member to other members. Even though a rework has been becoming an unavoidable and becoming a culture in software development, it directly gives an impact towards the performance and the productivity of the team which leads to the profit margin that should benefit the company, and not only the profit margin, more importantly having a lot rework in software development will cause the delay of software deliverable to the customer.

PT. XYZ as a leading IT consultant company that was established in the year of 2001 has been delivering successful projects for a diverse range of customers such as banking, multi financial, insurance, pharmaceutical, as well as construction sector. They are a leading information technology company that has been experiencing to provide IT solutions using Microsoft technologies and specializing in Business Intelligent application, Business Process Management, Enterprise Project Management, enterprise application development and enterprise project management. As an IT organization that has been experiencing in many of software development project in many industries, the company has also been facing with the deliverable of the application project during the implementation because of the rework occurred.

Table 1. Close - Ended Interview Result

Software Project 01	Software Project 02	Software Project 03	Software Project 04	Software Project 05	Software Project 06	Software Project 07	Software Project 08	Software Project 09	Software Project 10
Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework	Minimal backtracking and rework
	The answer is yes								
	The answer is no								

As it can be seen on table 1, it shows the close ended interview session asked to the project manager in each of the software development projects which have been finished between the years of 2010 until early 2016. The figure displays total 10 projects being accomplished but did not meet the timeframe given to finish the project. Based on the result of the interview, one of the main problems was because there are some backtracking and rework occurred in developing software application project. It means that 60 % out of 100 % of the software development project exceeded the timeline given due to a lot of rework in the process of implementing the application for its clients.

**Figure 1.** Percentage of close ended interview result

Therefore, it is very important and mandatory to identify rework in each software development project. It is realized that the rework cannot be eliminated from software development project, but reducing rework will be the best solution so that the plan that has previously been set can be ran as planned. This paper provides the information based on the current situation in the case study given and the result of the paper will give the most appropriate solution to be applied to the case study according to the current situation occurred in the company.

2. Related Work

According to (Szymanski, F., 2016), in his article, he explained that one of the rework occurred in software development project comes from under requirement phase. He added, the key to eliminate the rework in software development project is by defining the right and sharp requirement up front. In his article, he provided four tips to help organization reducing rework with the help of agile method. These four tips are collaborate, be lean, iterate, and visualize. Through these four steps, it will be able to allow the development team to perform their tasks more efficiently, lower cost, and more importantly reducing the rework in software development project.

Based on (Charette, R., 2005), through his research study, he found that software specialist spend about 40 to 50 percent of their time on avoidable rework rather than on what they call value-added work, which is basically work that has done right the first. He added that a lot of rework done in implementing a software application gives negative influence in term of cost and benefits for the organization and it is mostly considered software development project fails. He also provides list of the most common factors why the software development fails. One of the most crucial factor he listed is inability to handle the project's complexity. This factor leads the software development project causing the rework. Through case study, he concluded that IT software development project fails usually because of the rework exceeds the work that has been budgeted for. As the solution in his research, he found that providing Capability Maturity Model (CMM) as one of the best solutions that is believed to be able to reduce rework for software development project in order to avoid a failure. Through CMM, the organization should be able to understand the variations in the process and practices it applies.

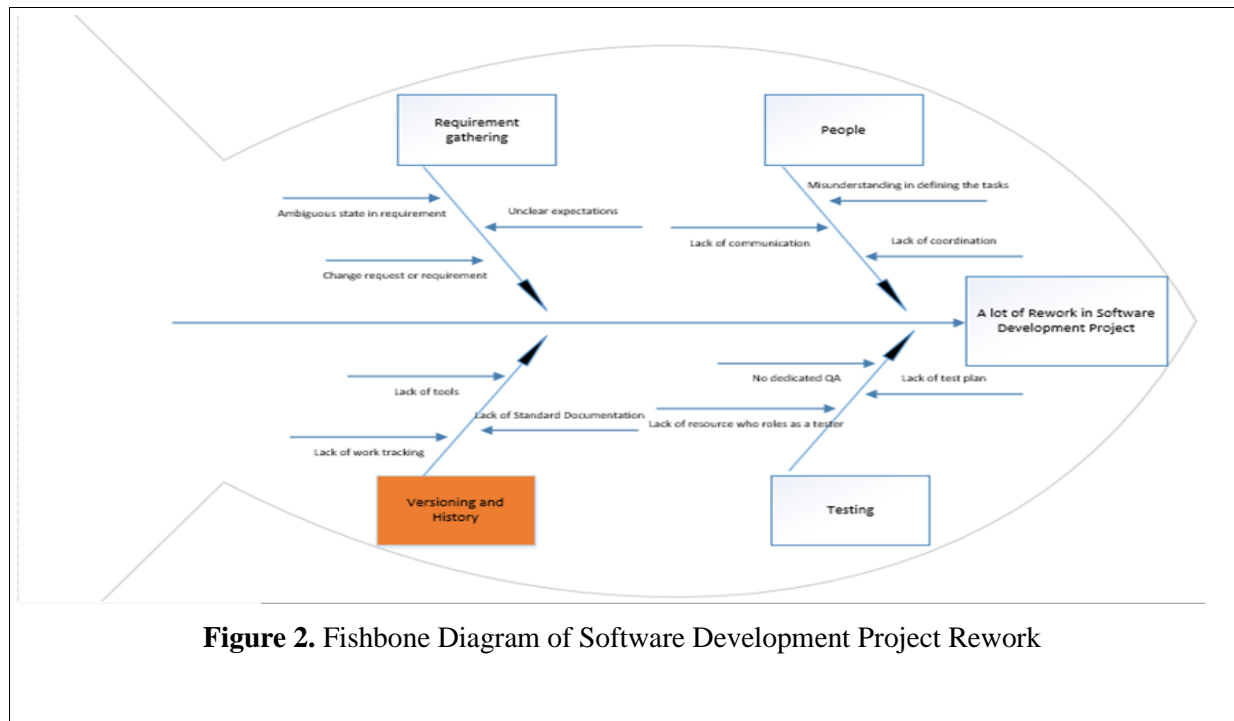
Based on the data in (Rational Software, 2009)., it said that 60-80% of the cost spent in software development project occurred in doing the rework rather than the actual work itself. The paper mentioned as well that, the cause of the rework is infrequently the result of individuals not doing their jobs very well. The most common factor that causes the rework in doing software implementation is due to improper planning, poor communication, and inadequate testing. Even though, the rework in software development cannot totally be eliminated, reducing the rework would be the best solution to remain the delivery of the project on time according to the timeframe. One way reducing the rework in software development project is through effective requirements management. According to this paper, it stated that effective requirements management will be able to help development team of the project dramatically reduce the rework, because effective requirements management provides very clear definition of what needs to be built and makes it visible across the development team, by providing the development team with an accurate and up to date view.

According to (Ramdoo, V. and Huzooree, G., 2015), in their research paper, they found that most of problem that causes a lot of rework under software development project coming out from poor communication and poor coordination between one and another in the development team. Having a low of rework in implementing software application gives significantly negative impact towards the organization such as wasting time, money, and effort of the work. They explained the most common factors that causes the rework are coming from four area of concerns. These areas are coming from the people, testing, ambiguous project requirement, and history and versioning. Through these four-key areas of concerns, they provide several solution or alternative to be applied to the case study given under their research paper. After performing the analysis, they recommend three alternatives. The first alternative is by having standards and procedures such as applying Capability Maturity Model (CMM) as well as International Standard Organization (ISO) that fits to the case study. The second alternative they stated by performing audit and review of the work such as performing as assessment that evaluates software development against the benchmark, or performing a technical review periodically to check for the correctness of their work. The last alternative they found is by applying Software Configuration Management. It is the process of tracking and controlling

changes to the software and hardware. As the best and most appropriate alternative to the case study given, they found that applying Software Configuration Management is believed to be the most appropriate solution to the company.

3. Research Methodology

After performing a study and review from the related work mentioned before, this research paper is trying to find out the main problem that causes the rework in software development project from PT. XYZ. In order to visually find out the root cause of the problem, the fish bone diagram or sometimes called as cause and effect diagram is provided.



Based on fish diagram in figure 2 above, there are four perspective that causes the rework in software development project taken from the case study given. These are testing, people, requirement gathering, versioning and history. Based on the perspective of testing, it is found that the company has lack of resource of the dedicated QA for testing the application being developed. It is important to have that kind of role because by having a QA, the testing is not only focussing on the tasks that refer to the requirement, but also it is focussing on the negative testing as well or well known as a non-functional testing. From people perspective, it is important that the team member should be able to collaborate very well between one to another in order to avoid misunderstanding between team members. This lack collaboration between team members is also considered as the main root caused faced by the company that causes the rework in doing software development project. The next root cause comes from requirement gathering. It is very common that a lot of rework in implementing software application because of the requirement collected does not meet the user expectations. This situation happens within the case study given as well. The last perspective is versioning and history. Lacking of traceability of the work can cause a lot of rework as well.

For the purpose of this research study, this paper will be focusing on the last perspective of the root cause which is versioning and history. After having analysed the particular root cause, the study of the literature is performed in order to find the best practice and recommendation to be applied to the case study given. After performing the literature, through the expert review, the best and most appropriate recommendation will be applied to the company as the solution.

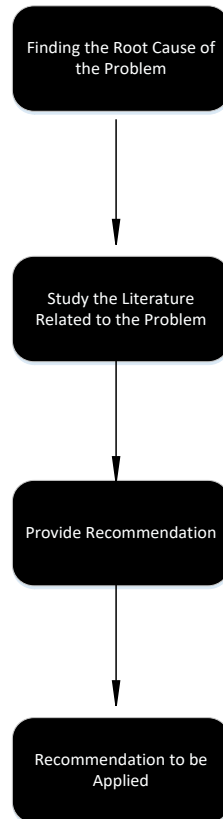


Figure 3. Research Method of the Research Paper

It can be seen based on figure 3. It shows the process of this research paper starts from finding the problem in the case study given until the finish. Each process is explained below.

3.1. Study the Literature Related to the Problem

This phase will be conducted after the phase of finding the problem under the case study given. The literature conducted will be based on the previous phase, where the problem or area of root cause is found. This literature review will also be conducted based on the research question and objective of this research study. The source information gathered for this literature should be in form of scholarly articles, books, dissertations, journals which are relevant to the problem.

3.2. Provide Recommendation

After finding the root cause and studying the literature related to the problem, the recommendation then will be provided so that the problem that occurred under the case study given can be solved. The recommendation provided will be a form of the best practice available. These best practices found during the study of literature then will be listed. The list of the best practice then is discussed with the expert who works in the company under the case study.

3.3. Recommendation to be Applied

The last phase of this research paper is to apply the most appropriate solution or recommendation to the company in order to solve the issue of having a lot of rework in software development project. By conducting the expert review, the most appropriate solution will be given to the case study and it is believed that the solution will help the company to reduce the rework in implementing software application for their client.

4. Analysis of the Finding

The objective of this phase is to evaluate some recommendations available after performing literature study related to the problem of the case study. The aim of it is nothing but to reduce a lot of rework occurred in software development project. The analysis of each recommendation found is performed in order to determine the feasibility and to be considered as the most and possible recommendation to be applied for the company so that the rework can be reduced. The perspective of each recommendation is defined below.

4.1. Applying Capability Maturity Model (CMM) as a Standard Procedure for the Company

According to (Rouse M., 2016), she defined CMM as a methodology that is designed to develop and refine the organization's software development process. The model describes five level of evolutionary path of increasingly organized and systematically more mature process. The Capability Maturity Model for software provides a guidance for software organization to gain fully control of their software processes starting from developing the application through maintaining the application that has been built and to find out how to evolve towards the culture of software engineering and management excellence. The main objective of CMM is to provide the design for the software within the organization in selecting process improvement strategies by determining current process maturity and identifying the few issues most critical to software quality and process improvement (Paulk, M., Curtis, B., Chrissis, M. and Weber, C., 1993). CMM provides five level of software processes. These processes are initial level, repeatable level, defined level, managed level, and optimizing level.

- **Initial level:** This process is characterized as a backup process. In this level, all processes are disorganized and more likely to be chaotic. The success criteria in initial level of CMM level will be depending on the individual efforts instead of team efforts.
- **Repeatable level:** In this level, the fundamental project management techniques are built. The success of it can be repeated for another software development project, because through this level, all prerequisite and process is starting to be documented.
- **Defined level:** Starting from this level, organization tries to develop their own standard for processing the activity for their own software application.
- **Managed level:** In this level, the organization starts to control and monitor their own process through data analysis and control
- **Optimizing level:** In this highest level of CMM, the process is constantly being improved through monitoring feedback from the current process and provide an innovative process to server a better solution for their software based on organizational needs.

4.2. Implementing Software Configuration Management

Software configuration management (SCM) is the process of tracking and controlling changes to software and hardware. General SCM systems provide tracing information within the development environment (Kim, D.Y. and Youn, C., 2010).. Software Configuration Management is about controlling the progress of the software being developed as well as be maintained. The main practice of using SCM includes controlling the source code being used

towards the particular application. It is added that, many feature can be found under SCM application, such as providing standard documentation about the particular software application, controlling and versioning the source code being used, and storing the history including the issues being faced during both development phase and maintenance phase alongside with the solution to solve these issues. According to (Kim, D.Y. and Youn, C., 2010)., “the necessity of SCM tools have increased in software development organizations since software is continuously growing in size and complexity, and standards such as ISO emphasizes traceability management as a core factor in determining software quality and customer satisfaction”. There are a lot of SCM tools available nowadays, starting from the open source tool until the licensed one, such as Redmine, Apache Subversion, and the licensed one which is Team Foundation Server (TFS) developed by Microsoft. Even though these tools are considered as an end to end application, it is much better to firstly define practices and policies in order to avoid the common configuration when implementing SCM application to the organization. Before implementing SCM tool, the organization should have some years of practical experience in order to develop a strong and fundamental standard for their software application such as create baseline at the project baseline, record and track requirement change request, maintain stable and consistent workspace.

5. Discussion and Recommendation

By performing the analysis in order to find out some solution to be recommended to the company of the case study, the discussion and recommendation is conducted. By having the interview session with the expert who has been involving in software development project for more than 10 years, some recommendations mentioned in section 4 will be provided. Through the expert review, the most possible and applicable of the recommendation will be applied to the company in order to solve the problem.

After having the review from the expert, it gives the result that according to the expert, he stated that it is very best to the company to apply the Capability Maturity Model (CMM) to the company. The objective is not only to make the company to become more mature towards their software being developed, but also by applying CMM, there will be a standard audit and review of the application. But in the process of applying the CMM as a standard procedure for the company, there are some issues. The main issue that is put to be the first consideration is because it could take longer time to apply the CMM into the background of the company. Secondly the expert stated that there is a need to hire an expert to train the employee and certification which could distract the cost of the company. When discussing the second solution which is applying the SCM to the company, the expert though to be the most appropriate solution. Because the company has already had some standards in implementing the application such as programming language used to develop an application and the framework and method to build the application for their client. He also added that between those three SCM tools mentioned in the previous section, the most appropriate and suitable tool to be applied to the company is Team Foundation Server (TFS). The reason behind the selection of the tool can be looked from the background of the company that has a partnership with Microsoft in selling and implementing the product. It is added that, other tool being used to implement the application by the developer are made from the Microsoft product as well, and by applying another tool that is made by the same developer will be much better with consideration that all these tools can be integrated between one to another very well.

TFS is SCM tool to provide a source code management, reporting, requirement management, project management, lab management, and release management capability. The primary purpose of Team Foundation is to enable collaboration on a team to make it easier to build a product, or complete a project. There are many types of projects ("Team Foundation Overview"). TFS is as well able to track the progress or milestone of the software project.

Another good feature of the TFS tool is the ability to store the history of the previous project as the purpose of reusability.

As the recommendation, the author realizes that this research paper still needs to be improved as the purpose of the future research. One the recommendation that can be applied for this paper is that to find out another alternative to reduce the rework in doing software application development. As the alternative applied for this research was focusing on applying System Configuration Management to the company, therefore it is expected that another and better alternative to be applied as a good and possible alternative for the company.

6. Conclusion

After having the discussion and finding the result in form of a recommendation to be applied to the case study, it can be concluded that even though the recommendation is believed to be able to reduce the rework in the company, it also makes people realize that the rework in software development project cannot be totally eliminated due to a lot of factors. It is added that, not all rework represents the weakness of the development team, as not all of it can be found as soon as possible. And even though the rework seems to give negative impact to the company, it is sometimes necessarily to have some rework towards the application or software being implemented in order to make the application much stronger and can be used in longer period of time and give a lot benefit to the company and its employee in the future.

References

- Charette, R., (2005). "Why Software Fails", *IEEE Spectrum: Technology, Engineering, and Science News*, [Online]. Available: <http://spectrum.ieee.org/computing/software/why-software-fails>. [Accessed: 30- Jul- 2016].
- Kim, D.Y., & Youn, C. (2010). Traceability enhancement technique through the integration of software configuration management and individual working environment. Secure Software Integration and Reliability Improvement (SSIRI), Fourth International Conference on IEEE.
- Nielsen P.D., "About Us: From Director and CEO Paul D. Nielsen," Carnegie Mellon Software Engineering Institute, <http://www.sei.cmu.edu/about/message/>.
- Paulk, M., Curtis, B., Chrissis, M., & Weber, C. (1993)., *Capability Maturity Model for Software*, 1st ed. Pennsylvania: SEI, 4-5.
- Ramdo, V.,& Huzoore, G. (2015). "Strategies to Reduce Rework in Software Development on An Organization in Mauritius", *IJSEA*, 6(5), 9 – 20.
- Rational Software, (2009). "Reducing Rework through Effective Requirements Management", *Requirement Management*.
- Rouse M., (2016). "What is Capability Maturity Model (CMM)? - Definition from WhatIs.com", *SearchSoftwareQuality*, [Online]. Available: <http://searchsoftwarequality.techtarget.com/definition/Capability-Maturity-Model>. [Accessed: 31- Jul- 2016].

"SCM Best Practice", *Pearson Highered*. [Online]. Available: <https://www.pearsonhighered.com/samplechapter/0321200195.pdf>. [Accessed: 31- Jul- 2016].

Szymanski, F., (2016). "AgileConnection | Four Agile Tips to Eliminate Rework in Application Development | Page 1", *AgileConnection*, [Online]. Available: <https://www.agileconnection.com/article/four-agile-tips-eliminate-rework-application-development?page=0%2C0>. [Accessed: 28- Jul- 2016].

"Team Foundation Overview", *Msdn.microsoft.com*. [Online]. Available: [https://msdn.microsoft.com/en-us/library/ms242904\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/ms242904(VS.80).aspx). [Accessed: 31- Jul- 2016].

Vanbrabant, B., & Joosen, W. (2013). A framework for integrated configuration management tools. *Integrated Network Management (IM 2013)*, IFIP/IEEE International Symposium on IEEE.

Sallé M., (2004.) IT Service Management and IT Governance: review, comparative analysis and their impact on utility computing, *Hewlett-Packard Co*. pp. 8–17.

Stoneburner, G., Goguen, A. Y., & Feringa, A. (2002). Sp 800-30. risk management guide for information technology systems.

Von Solms, B. (2005). Information Security governance: COBIT or ISO 17799 or both?, *Comput. Secur.*, 24(2), pp. 99–104.