Computational Research Progress in Applied Science & Engineering

ISSN 2423-4591

CRPASE Vol. 05(04), 127-130, December 2019

Root Cause Analysis of Defects For Large Enterprise Software Projects

Metin Altınışık*, A. Behiye Tarımcı

Turkcell Technology, Istanbul, Turkey

Keywords

Software,
Root cause analysis.

Abstract

It is very important that large enterprise software projects are delivered to the production environment in a quality manner in a timely manner and at the desired budget. The fact that the projects produces too many problems after being deployed into production environment decreases the productivity of the company and leads to serious customer dissatisfaction. In this study, we made root defect analysis of two big projects that were recently taken into production in telecom sector. By grouping these root causes, we completed the missing parts of the software cycle and eradicated the roots of these defects. After this root cause analysis, we found significant reductions in the rate of arrival of the defects and we shared our promising results.

1. Introduction

Without software it is almost impossible for any company to survive. In recent years the largest technology companies ranks highest in the ranking of the largest companies in all countries [1]. Companies make all their workflows through software and digitalization [2,3,4], which is popular recently, almost eliminates manual workflow. In particular, companies have to make their software both cheap and high quality in order to compete quickly in the market and to respond to the increasing customer needs very quickly. Therefore, many companies abandon the old popular software methodologies [8,9] like waterfall and replace them with agile methodologies [5,6,7]. In this way, they have the opportunity to maintain both quality and speed balance better. The quality of the software [10,11] must not only respond accurately and consistently to the customer's demands, but also provide the same experience across all the company's channels (internet, dealer, call center etc.). At the same time, these software must meet the latest security standards and simultaneously perform thousands of transactions very quickly.

In this complex software world, the defects made in the software both reduce the efficiency of the companies and cause serious customer complaints [12]. Especially in telecom companies subject to regulation, invoicing errors made against the customer cause very serious financial sanctions on the income of the company and even serious problems such as the cancellation of licenses. In addition, in today's world, customers who are active in social media are very aware of the mistakes made and there is a risk of losing thousands of customers at a time. In order to avoid these problems, companies do a lot of work on software quality. The most important of these is to organize test teams and detect problems that may occur in the software beforehand and they aim to solve them before they come to production [13,14].

In this study, we conducted root cause analysis (RCA) of the defects after two large projects from the telecom sector recently deployed in the production environment. We went down to the depth of why these defects was formed and identified the root causes. We then introduced additional measures in the software cycle that needed to be taken for these root causes. For these two projects, we repeated these analyzes at 2-month intervals. We found significant reductions in the defect trend with the application of the results.

The remainder of this paper is organized as follows. In the following section, we provide background information on five whys. We summarize the related studies in Section 3. We present the approach in Section 5, which is illustrated in Section 4, in the context of the industrial case study. We discuss the results in Section 6. Finally, in Section 7, we conclude the paper.

2. Background

In this section, we provide background information on “five whys” which we use in out study to identify the root cause analysis of the defects.

* Corresponding Author:
E-mail address: metin.altinisik@turkcell.com.tr

Received: 18 November 2019; Accepted: 21 December 2019

2.1. Five Whys

Five Whys analysis is a questioning technique used to explore the cause-effect relationships underlying a particular problem in quality management [18]. The main purpose of the technique is to determine the source of a defect or problem.

Five Whys is used in cause-effect-judgment analysis in the field of quality management. The aim is to determine whether a cause is a disorder or a problem. The five here is symbolic, not necessarily five queries, but is usually applied with five questions. The important thing is to clearly explain why the error was caused.

If it is not known how to solve a problem or where to start to overcome the problem, five why analysis techniques that are central to the cause-effect relationship play a key role. If the answers given to the questions start to become meaningless, then the questioning process is interrupted and proceeds through the last logical answer to find the source and start to produce a solution. To illustrate the workings of cause analysis, this example may be appropriate to better understand; It's lower grade than a student expects from the exam result. Five whys of this problem are analyzed;

Why did he get a low grade? (First why)

Why couldn't he give the right answers? (Second why)

Why couldn't he understand the logic of the problem? (Third why)

Why did they have absences? (Fourth why)

Why didn't you study with a scheduled program until the exam date. (Fifth why, a root cause)

As it is seen in the analogy given above, the source of the problem faced by the students has been descended with the questions of why and this problem has been eliminated with the last answer given.

We applied these “five whys” analysis for each problem and tried to find the true root causes of the defect. In this way, we were able to identify the correct actions to eradicate these defects by addressing the real causes underlying the defect.

3. Related Work

When we look at the software world, we see that the problems experienced during the major project transitions are common [22]. Post Mortem and Retro meetings are defacto step in the world to improve the software process management. When we look at the studies on Root Cause Analysis (RCA), Leszak et al. [19] found that when they did Root Cause analysis, they divided the errors into 3 main classes and classified the root causes of the errors into three main groups. These three categories are Algorithm, Functionality and Other Causes. They analyzed 427 data and performed root cause analysis in 13 sub-categories. In our study, we examined 530 problems and grouped root causes into 3 main categories and 10 sub-categories. Kalote et al. [20] examined the errors in 3 different subcategories (Logical, Standard and Redundant Code). They did RCA with Brainstorming. In our study, we used The 5 Whys method and in the RCA team, in our study, in addition to the participant roles of analyst, developer and Kalote et al., Participants from the operation admin team participated in the root cause analysis evaluation. Similarly Kumaresh and Baskaran’s work [21], when we examine the number of errors and the correct size of the project size in our projects, we saw a correct correlation between the size of the code and the number of errors.

In recent years, intelligent methods have been widely applied in many industrial fileds and excellent results have been obtained. And also, the root cause analysis has been extended by using automation and machine learning [23-26]. It is wirth mentioning, in our study, automation and artificial intelligence methods were not used and evaluated.

4. Industrial Case Study

The analyzed projects are two major projects of Turkcell's customer relationship management (CRM) application. Turkcell is Turkey's largest GSM operator company which is operating since 1993. These projects are the 2 biggest projects of the CRM transformation. One of them is the Processes project that includes all core processes of CRM. The other is the Voice project, one of Turkcell's most important fixed products. The development of these projects is implemented by Turkcell Technology, which serves as the R & D company of Turkcell. The application serves approximately 10,000 users and 35 million customers. In the Process and Voice projects examined from this study, there is a data model suitable for TM FORUM's eTom model [15]. At the same time the application was developed using 3-tier architecture and the latest trend technologies in the software industry (such as Spring Framework, Devops etc.) [17]. The order fullfilment of the applications is designed in accordance with SOA architecture. There are 320 Oracle SOA suite [16] bpel projects with 763.037 lines of code. Likewise, there are a total of 7,788 java classes and interfaces with 693,574 lines of java code. A total of 200 developers, analysts, testers and project managers worked in the project. The project lasted about 1.5 years and was successfully taken into the production environment. Root defect analysis of the projects within this scope has been an important study for the success of the projects. Developers who will develop large projects like this can use the output from this work to input their projects and use these practices to develop better quality software.

5. Approach

For the root cause analysis of the problems, we have applied the five why analysis reasons rules mentioned in section 2.1. We examined a total of 280 defect in the voice project and a total of 250 defect in the processes project. We analyzed all defects using the techniques mentioned in the background section. Accordingly, we identified the true root cause of each defect. Here we also see that each defect actually addresses one of the 3 categories. These 3 categories are human, process and technology. Here we went up to 5 questions for some defect, while others were able to identify root cause in up to 3 questions. Figure 1 shows the distribution of the sound project and the distribution of the processes project according to these 3 parts.

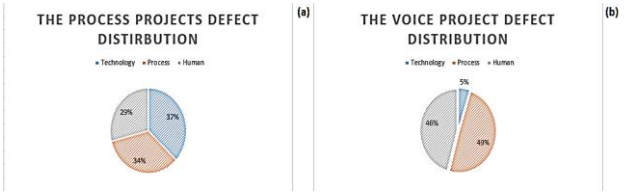

Figure 1. The process project defect distribution (a) and the voice project defect distribution (b)

One may notice that there is a difference in the distribution between the voice project and the process project. It can be seen that there is a significant difference in the technology part especially in the process project. The main reason for this is that the process project is a horizontal project that develops the general infrastructure of the system. In other words, it is an infrastructure project, so it is normal for the technology part to be the root cause of the defects in this project.

Table 1 shows distribution of the total number of the problems according to the code size for each project, in the below. Software Size (%/Total Size) means the percentage of this software size according to whole crm code size.

Table 1. The distribution of total number of problems according to code size

	The Voice Project	The Processes Project
# of Defects	280	250
Software Size (%/Total Size)	30.38%	28.47%

6. Results and Discussion

Figure 2 shows the distribution of root cause results of the voice project. Here we see that human errors are divided into 4 root causes, with 78 of them we found that manual data update error is the biggest root cause. In the case of process errors, we found that the biggest root cause was the missing use case, followed by the lack of analysis. Due to when the voice project is a product project, the technology part has not caused a big root cause. Here we have seen that dependence on 3. party firms are the biggest root cause.

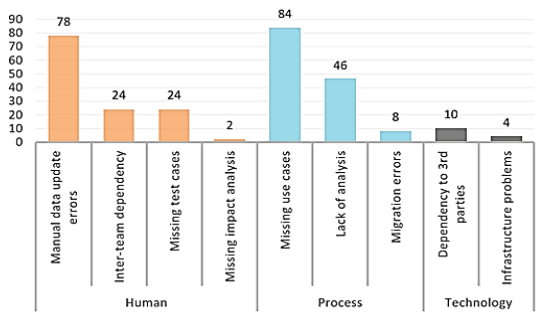

Figure. 2 Distribution of root cause results of the voice project

Figure 3 shows the distribution of root cause results of the processes project. Here we see that the error groups are equally distributed. As mentioned in section 5, since the processes project is an infrastructure project, it is seen that technology errors are much higher. We have identified technology errors as architectural errors, performance errors and infrastructure errors as root causes. We have identified root causes in this project too, such as missing cases, missing test cases and missing analysis.

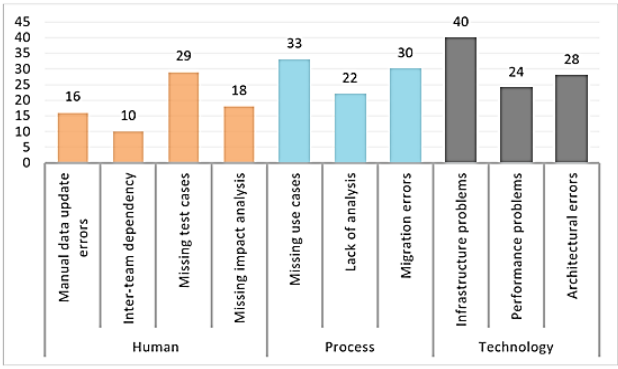

Figure 3. Distribution of root cause results of the processes project

After identifying these root causes, we made changes to our software cycle accordingly. For example, in order to improve the code quality, we have made the code review process compulsory. Likewise, we have taken additional actions such as internal trainings, user visits and process documentation to address the use case deficiencies. In the same way, we have redesigned infrastructures that produce many defect. We made significant improvements in performance improvement by working closely with the operational units. We were able to reduce the defects of nearly 80% by taking significant actions for each root cause.

7. Conclusions

In this study, we conducted root cause analysis of two newly developed projects of the telecom sector. We applied the five whys technique for each defect in the system. We have also categorized these root causes in human, process and technology breakdowns. For these root causes, we have added processes that are incomplete or inadequate in our software lifecycle. Thus, we have seen significant decreases in the rate of arrival of the defects. At the same time, we achieved serious positive results in software quality and customer satisfaction in live environment. Focusing on the root causes of the problems, we obtained 80% reduction in defect arrival rate with improvements in our software cycle, infrastructure and processes.

Acknowledgements We would like to thank software developers, tester, analyst and project managers at Turkcell Technology for sharing their project and code base with us and supporting our case study. In addition, we would like to thank all of our department managers and in particular the function director who led this study.

References

[1] The Largest Technology Companies In 2019, https://www.forbes.com/sites/jonathanponciano/2019/05/15/worlds-largest-tech-companies-2019/#79553884734f

[2] C. Loebbecke, P. Arnold, Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda, The Journal of Strategic Information Systems 24 (2015) 149–157.

[3] C. Legner, et al, Digitalization: opportunity and challenge for the business and information systems engineering

community, Business & information systems engineering 59 (2017) 301–308.

[4] P. Parviainen, et al, Tackling the digitalization challenge: How to benefit from digitalization in practice, International journal of information systems and project management 5 (2017) 63-77.

[5] F. Paetsch, E. Armin, F. Maurer, Requirements engineering and agile software development, WET ICE 2003. Proceedings, Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. IEEE, 2003.

[6] H. Ming, et al, Software quality and agile methods, Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004, IEEE, 2004.

[7] P. Abrahamsson, et al, New directions on agile methods: a comparative analysis, 25th International Conference on Software Engineering, 2003. Proceedings Ieee, 2003.

[8] C. Riemenschneider, K. Bill, C. Hardgrave, F.D. Davis, Explaining software developer acceptance of methodologies: a comparison of five theoretical models, IEEE transactions on Software Engineering 28 (2002) 1135-1145.

[9] M.A. Awad, A comparison between agile and traditional software development methodologies, University of Western Australia (2005) 30.

[10] J.P. Cavano, J.A. McCall, A framework for the measurement of software quality, ACM SIGMETRICS Performance Evaluation Review 7 (1978) 3-4.

[11] S. Kan, H. Metrics, models in software quality engineering, Addison-Wesley Longman Publishing Co. Inc., 2002.

[12] E.W. Anderson, C. Fornell, D.R. Lehmann, Customer satisfaction, market share, and profitability: Findings from Sweden, Journal of marketing 58 (1994) 53–66.

[13] R. G. Mays, et al, Experiences with defect prevention, IBM Systems Journal 29 (1990) 4–32.

[14] M. McDonald, R. Musson, R. Smith, The practical guide to defect prevention, Microsoft Press, 2007.

[15] Business Process Framework (eTOM), https://www.tmforum.org/business-process-framework/.

[16] Oracle SOA Suite, https://docs.oracle.com/cd/E28280_01/doc.1111/e10223/01_components.htm.

[17] Spring Framework, https://spring.io/.

[18] Determine The Root Cause: 5 Whys , https://www.isixsigma.com/tools-templates/cause-effect/determine-root-cause-5-whys/.

[19] A. Leszak, D.E. Perry, D. Stoll, A Case Study in Root Cause Defect Analysis, Proceedings of the 22nd international conference on Software engineering, 2000.

[20] J. Pankaj, N. Agrawal, Using defect analysis feedback for improving quality and productivity in iterative software development, 2005 International Conference on Information and Communication Technology, IEEE 2005.

[21] K. Sakthi, R. Baskaran, Defect analysis and prevention for software process quality improvement, International Journal of Computer Applications 8 (2010) 42–47.

[22] D. Sandeep, R. Singh Chhillar, Empirical study of root cause analysis of software failure, ACM SIGSOFT Software Engineering Notes 38 (2013) 1–7.

[23] I. Md Johirul, et al, A Comprehensive Study on Deep Learning Bug Characteristics, arXiv preprint arXiv:1906.01388(2019).

[24] I. Bargegol, M. Nikookar, R. Vatani Nezafat, E. Jafarpour Lashkami, A.M. Roshandeh, Timing Optimization of Signalized Intersections Using Shockwave Theory by Genetic Algorithm, Computational Research Progress in Applied Science & Engineering 1 (2015) 160–167.

[25] M.M.Naddaf-Sh , S.Hosseini , J. Zhang , Nicholas A. Brake, H. Zargarzadeh, Real-Time Road Crack Mapping Using an Optimized Convolutional Neural Network, Hindawi, Complexity (2019). https://doi.org/10.1155/2019/2470735

[26] A. Ebrahimi, J. Addeh, Classification of ECG Arrhythmias Using Adaptive Neuro-Fuzzy Inference System and Cuckoo Optimization Algorithm. Computational Research Progress in Applied Science & Engineering, Computational Research Progress in Applied Science & Engineering 1 (2015) 134–140.