

## Development of a Hybrid Test Method Complied With the Agile Methodology

Emin Borandag<sup>1</sup> And Fatih Yucalar<sup>1</sup>

<sup>1</sup>Department of Software Engineering, Manisa Celal Bayar University, Manisa, Turkey  
Corresponding Author: Emin Borandag1

---

**ABSTRACT:** Nowadays, software systems are essential to many lines of business. The need for advanced software systems is growing with every passing day as a result of the effects of increasing competition, improving technology and the rising capabilities of software organizations. Various models have been developed in the last 30 years to evaluate the quality systems and processes that are used in software development, to refine the processes and to determine capabilities. Software errors remain an important issue, although models having different capabilities and features have been developed. This paper studies the development of a hybrid test method in accordance with agile methodology, which is then applied to software projects using a relational database. It also covers the application of the method to software projects developed, considering the software processes within the context of test process management. The developed model is applicable to projects which undergo many changes, especially projects developed in small-scale software companies.

**KEYWORDS:** Agile Software Development Methodology, Extreme Programming, Software Development Methodologies, Software Testing.

---

Date of Submission: 09-02-2018

Date of acceptance: 24-02-2018

---

### I. INTRODUCTION

The main objective of the software test is to compare the current status of the software with the requirements of the project to find out the differences. Software features are evaluated and examined for this purpose. In order to carry out these processes, it is necessary to determine the test activities and to create a plan in accordance with these activities [1]. All the methods and approaches that are followed up within the period of time until the acceptance and application of the tests are examined as quality oriented under test process management. In this context, various software development techniques have been applied in software to provide quality certification, improve processes, determine competence/skill and meet increasing customer needs. These software development methods which have been applied may vary depending on the company's development and the structure of Software Company. Extreme Programming (XP) is the most commonly utilized and widely applied method among these methods, which is generally applied in small to medium sized software companies and called Agile [2], which changes the software life cycle from beginning to end. By applying XP, software errors can be detected earlier.

In this study, it has been dealt with how software testing, which is often less emphasized by small and medium-sized software organizations, will be implemented at all phases of the software life cycle. In this context, the aim of the article is to propound how a hybrid test method which can reduce the error that inherent in the software and which was developed by XP in software projects applied to developed software and what kinds of advantages have been obtained. In addition, a software tool has been developed for the use of the method. The second part provides basic information about software testing techniques and test methods. In the third part of the work, a software development method XP has been introduced. In the fourth part, the proposed hybrid test method has been explained. In the final part of the study, the results and recommendations have been given.

## II. SOFTWARE TESTING

Software testing is carried out to find out the errors of a program or system, to determine whether its features or capabilities match the desired outcomes [4, 5]. It is not possible to find all the errors in the software, but it is significant to reveal the detected software errors at the earliest possible phase [6]. Software errors may occur at any phase of the software developing, including misidentification of requirements, lack of communication, incomplete identification of processes, etc. [7]. There are two major issues in software testing: white box and black box tests.

**White Box Test:** White box testing can also be named as structural testing and logical testing. It is a significant method that applied in testing software. The aim of the White Box Test is to validate the defined conditions and the correctness of the algorithm by analyzing all the paths that the code can follow in the software codes. Functions not performed in the structural test cannot be determined [1]. It consists of various test techniques within itself.

- *Statement Coverage:* This technique is the most applied technique in the white box test. The aim of this test technique is to execute each piece of code in the software [8].
- *Branch Coverage:* It is based on testing each branch of codes received from the code blocks in the software. The difficulty of the branching technique is that a separate test program is required to obtain correct and incorrect results [6].
- *Condition Coverage:* A basic statement is that the branch coverage technique is applied to aggregated values. Complexity is increasing in direct proportion to the increase of conditional clauses and logical operators [9].
- *Path Coverage:* The most efficient technique among the above-mentioned test techniques is the technique of path coverage. The path coverage technique is executed by taking a part of the codes applied in the software [10].
- *Data Flow:* It includes the test according to the calculation results of a variable defined in the software. In this method, it is tried to determine whether the variable is assigned different values and is consistent with previously calculated values [6].
- *Loop Coverage:* It should be tested in a loop that is frequently applied in the software [11].

**Black Box Test:** Software testing technique for specification. It is executed to test whether the software has progressed according to the features in the specification [12]. The objective of the test case software is to detect errors in the software. A repeatable template is generated with the defined test cases [12]. At the beginning of the test, the software tries to find out if there is a difference between the requirements of the software, the processes of the software and the characteristics of each process. At the same time, it is tried not to determine the features of the software which is brought to the market by software testing [13].

## III. EXTREME PROGRAMMING

The problem with software production is increasing complexity in software production in parallel to increasing customer requirements in developed software. The changes that take place in the project cause the time and budget to be exceeded. As a result of the problems encountered, it appears that a significant part of the software is not effectively applied [14]. As a result of efforts to overcome these problems, methods have been developed called Agile which change the software life cycle from beginning to end. The most widely applied one of these methods is Extreme Programming-XP [3], developed by Kent Beck. A different perspective has been introduced to XP agile software production methods. The XP content has a structure that can be applied to dynamic and variable projects. The most significant advantage is communication with the customers. With XP, it aims to present a software product to the customer service in a short time without any errors, responding quickly to changing requests of the customer in accordance with the principles of software engineering. In addition, XP method is a software development method that emphasizes intra-group communication [14]. The emergence of XP as a discipline of software development is to provide a software development method that makes it easy to use, easier to use, provide more communication, and greater demand for customer feedback [15-17]. XP can provide tangible improvements in the early phases of the software. Software errors can be detected early on. Software errors can be compensated by feedbacks [16-20].

## IV. PRE-EMPTIVE PRIORITY-BASED TEST HYBRID SOFTWARE DEVELOPMENT METHOD

A hybrid test method that can be applied to software developed by using the XP methodology that can reduce the amount of error encountered during and after software development has been developed and a software tool for implementing this method has been developed.

**Beginning a Project with XP:** In the work to be carried out, the requirements of the software are derived by determining which information the user will enter into the system and which information will be reached. The

obtained requirements are converted to customer cards and technical cards. For this phase of testing, black-box testing has been applied during the analysis phase of the projects. By applying this test technique, it has been determined that the demands of the specifications have been satisfied within the program and the error costs have been tried to be reduced. The limits and constraints of the programs to be developed have been determined. It has been also determined how the technological infrastructure can be created. By applying XP's planning game application, it has been ensured that the project team and the client work together on the project to be developed. The information obtained from the customer cards has been transferred to the hybrid test-based program developed earlier. As a result of the analysis information obtained, the time-sheet record has been made for the project size.

**Project Cycle:**

The project begins after the life-cycle, user stories, system similarities and technological constraints have been identified. In this phase, the information that is formed as a result of the planning game has been examined in detail. The structure of database and main and side processes of the project to be developed have been defined. The information obtained after these definitions has been also recorded in the developed hybrid test-based program. With this information entered into the program, deadline of the project has been revised. This information is used to calculate the time in the light of this information. Risks and possible risks are evaluated within the planning game. Project related commitments are identified.

**Test-Driven Software Development:** In order to write a test program, firstly inputs and outputs that will be entered in the main program are determined. The control list for the test is prepared. With considering data of the test, it is tested how the program works well. With the occurred errors, the reasons behind the errors can be learned.

**Test Code and Main Software:** After the tests of the previous phase have been successfully completed, a new test is written and applied to the functional code. The functional code is rearranged to be successful in these tests. These steps are continued until all the tests specified by the programmer are successful.

**V. Developed Software Projects**

Three comprehensive software projects in which XP methodology and the recommended test method were applied have been carried out. Developed projects have been actively used. These software projects have been developed in a way that adheres to XP practices and working environment. In addition, appropriate hybrid test method and test tool developed for the test part of these software projects are applied. Descriptive information related to the projects is given in Table 1 below.

**Table 1. Basic information of projects**

	<b>Project-1</b>	<b>Project-2</b>	<b>Project-3</b>
Software Team Size	3 Software Developer 1 Customer	3 Software Developer	3 Software Developer, 1 Test Specialist
Educational Level	2 Master of Science, 2 Undergraduate	1 Master of Science, 1 Associate's Degree	1 Master of Science, 1 Undergraduate, 2 Associate's Degree
Software Experience Information	< 5 year: 2 people 5-10 year: 2 people	5-10 year: 1 people 0-5 year: 1 people	> 10year: 1 people 5-10 year: 3 people
Expertise Information	High	High	High
Programming Language Information	High	High	High
Number of Client Cards	62	2	89
Domain	Windows Application, C#, MS SQL	Windows Application, C#, MS SQL Server	Windows Application, C#, MS SQL

Measurements: A program which is called as Source Monitor program has been applied in order to measure the data of the three different projects conducted. All calculations displayed on the chart have been obtained by this program. Table 2 below shows the number of lines of the project, the number of commands,

Comment rate, the average complexity ratios, the number of files in the program, and the complexity ratios of the codes.

**Table 2. Code information of the projects**

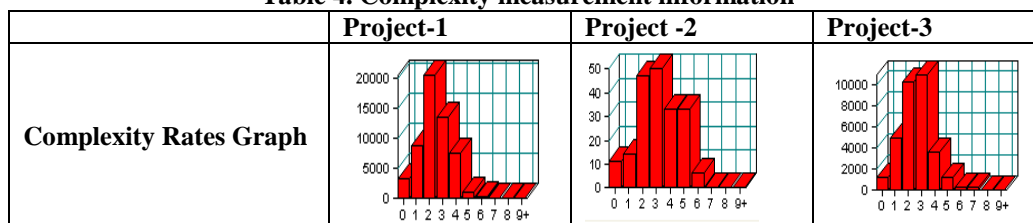
	Project-1	Project-2	Project-3
<b>Total Number of Code</b>	62.815	334	53.556
<b>Total Number of Commands</b>	54.307	194	32.236
<b>Command Ratio</b>	10.6%	9%	2.5%
<b>Average Complexity</b>	2.31	3.05	2.52
<b>Number of Files</b>	131	6	80

The average complexity ratios of the codes have been found as 2.31. It has seen that only 2.24% of the codes have complexity value of 5 and above. These obtained ratios have indicated that the software is in a good position in terms of manageability [21, 22]. Software complexity is a factor affecting software quality. Therefore, for the quality of software management and software quality it is significant that the complexity of 5 or more is low. In software, the complexity is usually calculated by the number of components that constitute the system, or the number of other subcomponents that the components call in themselves. The 2.31 ratio obtained as a result of the project is also an important indicator of the requirements of the software complexity, the technologies used and the efficient use of customer relations. White box and black box test methods, which are two different test methods, have been applied together with XP in the software development process to reduce the errors that have been tried during the software development with the hybrid test technique applied in the project. The main modules of the projects and their development processes have been mentioned below. In addition, information about the complexity of the projects and the information about the number of commands and the measurement information of the developed projects have been given in Table 3 and Table 4 below.

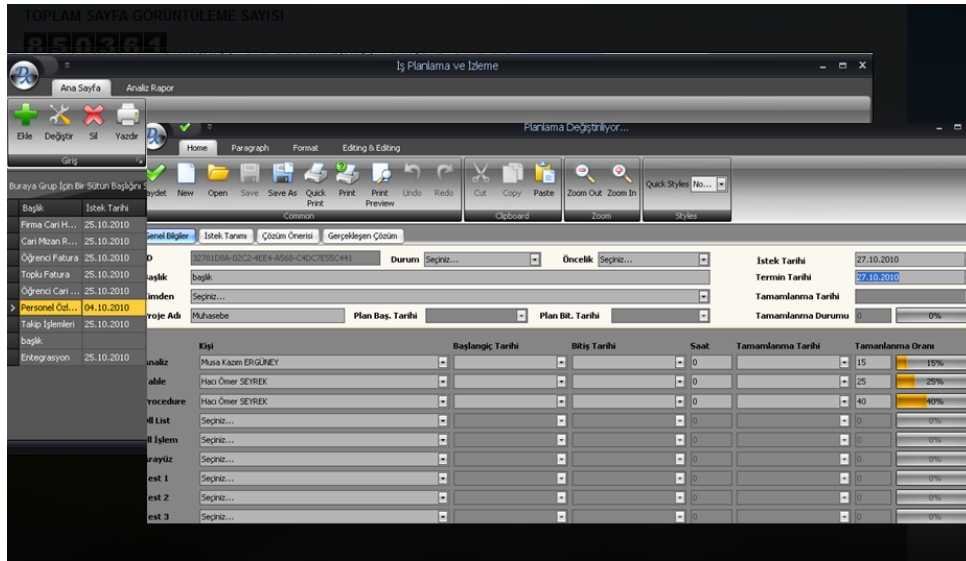
**Table 3. Project command information**

Complexity Rates	Project-1	Project-2	Project-3
0	3257	11	1206
1	8755	14	4851
2	20370	47	10124
3	13312	50	10799
4	7392	33	3530
5	871	33	1124
6	186	6	280
7	114	0	265
8	36	0	43
9+	14	0	14

**Table 4. Complexity measurement information**



The average complexity ratios of the codes have been found as 2.52. This is a significant indicator that the software is manageable and that there will be less error when the software starts to be applied. Also, the complexity of 5 or more in the rates is very important in terms of software management and software quality. It has seen that only 5.35% of the codes has a complexity value of 5 and above. Two different test methods, white box and black box test methods, have been applied together with XP in software development process in order to reduce the errors occurred during software development. XP's life cycle and applications have been tried to be used in accordance with the specifications and size of software projects. Determination of the main and side processes of the project with the method used has been determined as design, coding of the software, maintenance and new demands processes. Within each process, XP's applications and lifecycle features have been applied by giving priority to testing in each process. White box and black box test methods have also been effectively applied in the hybrid software development method. In Fig-1, there is a screenshot of the developed software tool for the application of the method.



**Fig.1. Hybrid software tool**

First, test cases (text-cases) have been defined before the coding phase in accordance with the test method. Hybrid test tool has provided to develop software projects regularly and efficiently with recording required information on relational database, demands from customers and required technical details, customer and software developer tests for developing software. Through this defined method and tool, three different software projects have been developed. A software team consisting of five people has participated in the development of the projects. Test-driven hybrid software development method and hybrid test tool have been effectively applied in developed software projects. In accordance with the software project size estimations estimated by XP's planning game application, the projects have been passed to the phase of completed maintenance and new demands within the determined time. Over 100,000 lines of code have been coded for the projects. These three different software projects, which have been developed as a test, have been applied for more than two years. Errors that indicated by customers have been solved by applying a test-driven hybrid software development method and a hybrid test tool in a XP-based software conversion. When we look at the complexity ratios that the most basic method for software quality, it is seen that only 1.86% of the complexity values of the projects generated have a complexity value of 5 or more.

Nowadays, desired success rates in software projects cannot be reached due to the usage of methods that are complicated and do not attach importance to testing. Software development methods developed for major software projects have great difficulties in terms of usage and management. However, the developed test hybrid software development method and the hybrid test tool have been effectively managed with software projects. Thanks to the method applied, errors have been detected at an earlier phase and tried to be corrected. With applying XP in the method, ease of use has been provided for the software companies. White box and black box test methods have been also applied in order to keep the testing prioritized. The recommended pre-emptive priority-based test software development method is applicable to small and medium sized different software projects. Thanks to the developed hybrid test tool, development levels of the projects and instantaneous error conditions can also be observed.

## VI. CONCLUSION

Among the most significant assets of the companies, it is significant that the software that applied to create the information works at full strength and errorless. The most important problem in this regard is the increase in complexity in parallel with the increase in customer requirements. One of the biggest causes of the errors is that the analysis and design process is not carried out in the prescribed proportion. The other is that the developed software products are not sufficiently tested. Software testing is ruled out, especially in small and medium-sized software organizations. Due to these reasons, there are significant financial losses and resource wastes in the software supplied to the customers. As a result of the problems encountered, it appears that most of the software has not been effectively used. In this study, a hybrid test method and tool based on Extreme Programming have been developed with using relational databases for small and medium sized software companies.

In this study, hybrid test model based on Extreme Programming, a total of 116,705 lines of code have been generated and a software team consisting of 5 people have participated in the development of the projects. Considering the complexity values of the projects carried out, it has seen that only 1.86% have a complexity

value of 5 and above. It is also significant for software management and software quality that the complexity of 5 and more is low in these ratios. These software projects have been developed in a way that is substantially dependent on XP's lifecycle and working environment. In addition, appropriate hybrid test method and test tool have been developed for the test part of these software projects.

The following findings have been obtained in this study:

- The method has been applied on 3 different software projects and the software has been developed on a test basis.
- Thanks to the applied method, software errors have been detected at an earlier phase and the required corrections have been allowed to be corrected before.
- Three comprehensive software projects have been carried out by applying the XP methodology and the recommended test method.

### References

- [1]. Gerald D. E., Raymond M. Jr. (2007). "Software Testing Across the Entire Software Development Life Cycle". Wiley-IEEE Computer Society.
- [2]. Schwaber K., Beedle M. (2001). "Agile Software Development with Scrum". 1<sup>st</sup> ed. Pearson.
- [3]. Beck, K. (2002). "eXtreme Programming". Addison-Wesley.
- [4]. IEEE Standard (1998). 829-1998, IEEE Standard for Software Test Documentation.
- [5]. Gray, L. (2008). "Standard for Information Technology. Software Life Cycle Processes Guidebook to IEEE/EIA 12207", Abella Corp.
- [6]. Burnstein I. (2003). "Practical Software Testing: A Process-Oriented Approach", 1st edition. Springer.
- [7]. McGregor J. D., Sykes D. A. (2001). "A Practical Guide to Testing Object-Oriented Software". Addison-Wesley Professional.
- [8]. Schach, S. R. (2007). "Object-Oriented & Classical Software Engineering", 7<sup>th</sup>ed., McGraw Hill.
- [9]. Elodie V. (2011). "White Box Coverage and Control Flow Graphs", Testing Techniques Course Note. <http://wwwhome.cs.utwente.nl/~timmer/teaching/WhiteBoxCoverageCFG.pdf>
- [10]. Torkar, R. (2006). "Towards Automated Software Testing: Techniques, Classifications and Frameworks". Doctoral thesis, Blekinge Institute of Technology, School of Engineering, Department of Systems and Software Engineering.
- [11]. Beizer, B. (1990). "Software Testing Techniques", 2nd edition, Itp – Media.
- [12]. Myers G. J., Sandler C., Badgett T. (2011). "The Art of Software Testing". 3<sup>rd</sup> ed., John Wiley & Sons, Inc.
- [13]. Kaner C., Bach J., Pettichord B. (2001). "Lessons Learned in Software Testing: A Context-Driven Approach". 1<sup>st</sup> ed., John Wiley & Sons.
- [14]. Highsmith, J. (2002). "Agile Software Development Ecosystems", 1<sup>st</sup> ed., Addison-Wesley Professional.
- [15]. da Silva A. F., Kon F., Torteli C. (2005). "XP south of the equator: An experience implementing XP in Brazil". International Conference on Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, vol 3556. Springer, Berlin, Heidelberg, pp. 10-18.
- [16]. Yael Dubinsky Y., Talby D., Hazzan O., and Keren A. (2005). "Agile Metrics at the Israeli Air Force", Proceedings of the Agile Development Conference (ADC'05), IEEE.
- [17]. Acar Ö. (2009). "Extreme Programming". Istanbul: PusulaYayincilik.
- [18]. Beck, K., Andres C., (2004). "Extreme Programming Explained: Embrace Change", 2<sup>nd</sup> ed., Addison-Wesley.
- [19]. Williams L., Krebs W., Layman L., Antón A. I., Abrahamsson P. (2004). "Toward a framework for evaluating eXtreme Programming". 8th International Conference on Empirical Assessment in Software Engineering (EASE 2004), pp. 11-20.
- [20]. Bossi P. (2003). "eXtreme Programming applied: a case in the private banking domain". In Proceedings of OOP, Munich.
- [21]. Shepperd M. (1988). "A Critique of Cyclomatic Complexity as a Software Metric", Software Engineering Journal, 3(2): 30-36.
- [22]. Jørgensen M. (2004). "A Review of Studies on Expert Estimation of Software Development Effort", Journal of Systems and Software, 70(1-2): 37-60.

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with  
Sl. No. 3822, Journal no. 43302.

Emin Borandag1. "Development of a Hybrid Test Method Complied With the Agile  
Methodology" International Journal of Engineering Science Invention (IJESI), vol. 07, no. 02,  
2018, pp. 43–48.