p–ISSN: 2723 - 6609 e-ISSN: 2745-5254
Vol. 5, No. 8 August 2024 http://jist.publikasiindonesia.id/

Software Defect Elimination Information System in Software House Company

Johannes Simatupang
Universitas Bina Nusantara, Indonesia
Email: johannes.simatupang@binus.ac.id

*Correspondence

Keywords: software testing, software defects, zero defect application software.

ABSTRACT

In software that is free from defects or errors, a fixed method and technique of software testing is required. For this step, an Information System is needed to help eliminate software defects, so that testing or testing work is not a burden on software costs but is a perfection of software development work so that Zero Defect Application Software can be realized. From several software testing methods and techniques, a strategy suitable for the size of a software development project is required. So even though the software development project is small, it still requires a suitable test. And for software development testing to have reliable quality, it is necessary to measure and criteria for each software testing practice, so that a software development project can be determined in quality. To support software testing work, an Information System is needed that supports the need for processors to assess a quality software development test. With the Software Defect Elimination Information System, an objective quality of software development testing can be produced.

Introduction

Software or Software as a form of work of software development is an important right. The quality of a Software or Software reflects the quality of the Software Engineer team itself (Rifa'i, Ananda, & Fadhli, 2018). Customer satisfaction determines how successfully a software or software is implemented in an organization's environment. Software that is categorized as office automation steps the management in making decisions, accelerates the filing of a file and steps up personnel in collaborating with other colleagues using the internet as an infrastructure (Jauhari, Anamisa, & Mufarroha, 2022). Companies that process software are not spared from defects or defects in production results. Software called service packs, patches, and percentage updates are common actions to fix defects in their work (Hutabarat, 2022).

A study conducted by the International Business Machine (IBM) (James W. Cortada, 1995) shows that every software application development at IBM results in 60 (60) software defects. A study conducted by IBM also showed that 50% (fifty per cent)

of programming time is used for error detection. And 80% (eighty per cent) is used for error detection after the programming stage (Sudipa et al., 2023). At least 10 (10) times the cost is spent on correcting errors at the programming stage and 100 (one hundred) times the cost is spent on correcting production errors (Reivaldi Kesuma Kagi, Muchammad Ficky Duskarnaen, & Hamidillah Ajie, 2020).

William E Perry explained, "Too little testing of software is a crime, too much is a sin". Most of the problems that arise concerning software testing are failure to define software testing goals, software testing being carried out at the wrong stage in the software development cycle, and ineffective testing techniques.

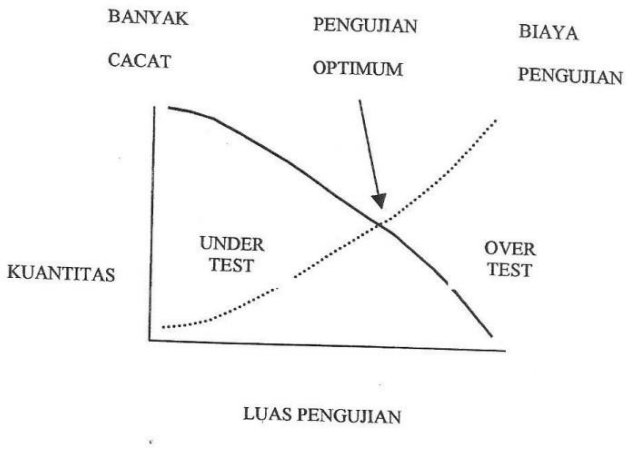


Figure 1. Software Testing Cost Curve (William E. Perry, 2006)

Based on Figure 1 If the Cost of Software Testing increases, then the number of undetected software defects decreases. However, cost-effectiveness can be achieved if software testing is optimal, and not over-tested. What is meant by Over Test is testing excessive software so that the cost of testing is more expensive (Prastyo, Suhartono, Faisal, Yaqin, & Firdaus, 2024).

Software that is categorized as Application Software, which is produced by a Software House or Consulting Company is used to support the business process (Business Process) of a client company. Application Software is made in a tailor-made way, tailored to the needs of the end-user (Pamuji, 2018). User Expectation of an Application Software is a benchmark of customer satisfaction. Before the Application Software is released and implemented, the software developer conducts software testing or Software Testing. Internally, at this time the Software House Company is focusing on providing a service in the form of Zero-Defect Application Software, Definition of Zero-Defect, trying to minimize software defects produced by the Software Engineer Team. If Zero-Detect Application Software is achieved, it can be concluded that customer satisfaction is met. Then customer relationship management can be easily formed so that the status of the customer will change from a satisfied customer to a loyal customer towards the Software House.

Software developed in a tailor-made manner, at the implementation stage based on daily experience, there are deviations from the end-user and third-party software, resulting in an Application Software that is developed tends to have defects or defects. This is experienced by the Software House Software Engineer Team. Of course, based on the bad experience at the implementation stage, the management wants a solution strategy for "Software Defect Elimination". The solution strategy that the management expects is of course by paying attention to the best cost, time, and human resources.

"V-Concept" Software Testing

The steps of the software test process are, first, Assess the development plan and status: the testers create a test plan that will be used for the evaluation of the software to be implemented. Second, Develop the test plan, and form a test plan with the same pattern for several software development testing processes. Third, test software requirements, the inability to get the right needs during the request collection stage can increase implementation costs and must determine end-user requests accurately and completely and do not cause conflicts with others. Fourth, the software design test, externally and internally, the software design has been tested, the testers pay attention to the design which is the objectivity of the end-user request. Fifth, test software construction, the chosen approach to develop software from the internal design will determine the type and activity for testing, if the software is developed using the "Waterfall" life cycle, then verification and subject to error are needed. Sixth, Execute tests, involve testing coding in a dynamic state, approach methods and tools specifically in the test plan used to validate the executable code according to the end-user requirements and structural specifications of the design. Seventh, Acceptance test, acceptance testing allows users of software to evaluate the applicability of software that will be used daily. Eighth Report test result, report testing is a continuous process, defects will appear in the test, and then corrections can be made by considering low costs. Ninth, test software installation, the ability to execute software that is the result of production, so it is necessary to test the environment of the software. Tenth Test software changes, after the software is successfully implemented, and then the performance of the software is maintained. Eleventh Evaluate test effectiveness, improving the quality of testing can be achieved by conducting an effective evaluation of the test at the end of each software development.

Research Methods

For the development of an information system for the imitation of software defects, the following steps are needed to research the problem: formulating the research problem and limiting it, after which data collection through interviews, observations on several software development projects and literature studies by referring to books and the internet. Analyze relevant data filtering case studies, group and classify data as needed, and select strategies and information system design models to support the proposed strategy. Solution Design for case studies, namely, System design analysis in general through the formation of data models and process models, Information System prototype

design to support software defect elimination strategies as well as design testing and analysis.

Information System development products are presented in the form of software. Software is a form of work of one or several programmers. Software consists of programs that are organized to achieve the goals of the software as a representation of the information system being developed. Models of the software development cycle have evolved since the history of software development and the programming language generation was developed. Another software development model is Prototyping, which is the process of a "Software Engineer" creating a software model to be developed. The difference in the properties of the desired product is called a defect, a state in which software is free from defects is called a Zero-Defect. Software defect prediction is one of the important activities in the testing phase or stage of the software development life cycle (De, Malik, Ghosh, Saha, & Saha, 2015) proven in 2002, according to NIST (National Institute of Standards and Technology).

Information is one of the important resources available to managers (Mcleod Jr, 1995). The output of information generated by the computer is used by both managers and other people in an organization. The first important computer application is processing data for the accounting section (Accounting Information System - AIS). Strategy management is defined as a set of decisions and actions that result in the formulation and implementation of plans designed to achieve the Company's objectives. Customer satisfaction is a function of performance and expectations. If the performance is below expectations, the customer is said to be dissatisfied. If the performance is by expectations (expectations). Software testing is a key component and quality assurance of software and represents the subject matter of specifications, design and coding (Herdiyanto & Lukman, 2022). In software development, software engineers try to develop software from abstract concepts to visible implementations, and then test the software. Software Engineer creates a list of test cases intended to disassemble software that has already been built. For any software development project, there is an inherent conflict of interest that occurs at the time testing begins. Several companies engaged in software development technology services provide services for software testing in addition to software development services themselves.

Black-box testing is an alternative to the white-box method, which is a complementary approach. By applying the black-box method, we attract a series of test care, the first of which is a test care that reduces the price of more than one, and a test care that tells us something about the presence or absence of an error class. Then using the white-box testing method, the software engineer can provide assurance that all paths are independent on a capital, use all logical decisions on the true and false sides, and use structure and internals to guarantee validity. Both methods use testing techniques for their application.

Table 1
Testing Technique

No.	Technique	Description	Method

1.	STRESS	Determine System Performs With Expected Volumes	**WHITE-BOX**
2.	EXECUTION	System Achieves Desired Level Of Proficiency	
3.	OPERATIONS	System Can Be Returned To An Operational Status After A Failure	
4.	COMPLIANCE	System Can Be Executed In A Normal Operational Status	
5.	SECURITY	System Is Protected By Standards And Procedure	
6.	REQUIREMENTS	System Is Protected By Importance To Organization	
7.	REGRESSION	System Performs As Specified	BLACK-BOX
8	REGRESSION	Verifies That Anything Unchanged Still Performs Correctly	
9	ERROR HANDLING	Errors Can Be Prevented Or Detected	
10	MANUAL SUPPORT	The People-Computer Interaction Works	
11	INTERSYSTEMS	Data Is Correctly Passed From System To System	
12	CONTROL	Controls Reduce System Risk to An Acceptable Level	
13	PARALLEL	Old System and New System Are Run And The Results Compared To Detect	

Software quality measurement is a quantification of the results of evaluation. The quantification of the results was obtained using scoring, as a measurement tool to describe the state of software testing. Each rating obtained is multiplied by a factor, and then a score evaluation worksheet for measuring the quality of soft warfare testing is prepared. so by measuring the quality of software testing, the quality of the software developed can be determined. By using a measurement model with 3 levels, namely factor, criteria and metrics.

Results and Discussion

Based on the results of software testing, it is not only the responsibility of the Software Engineer Team. It is the joint responsibility of a Company to demonstrate software that has been tested or is free of defects. Therefore, testing of software requires good and proper management. When planning or planning a software development

project, it is necessary to consider the strategic plans of a software development project to be implemented. When organizing or organizing a team of software testers, it can be determined who the staff is involved, there is already a clear job description to be implemented and the goals of the job description. At the time of actuating or implementing software testing, the software tester has the expected goal of a device being tested. At the time of controlling or controlling the work, each person who is responsible for the formation of a software tester team can already account for the results of testing software developed for the project leader. The controlling started at the planning time and continued at the time of organizing and more presentations (%) supervision at the time of actuating. To avoid the impact that will arise, it is necessary to evaluate the software development work and develop an appropriate formula for designing application software to support the work process of eliminating software defects.

The developed Software Defect Elimination Information System can be described using the following Context Diagram:

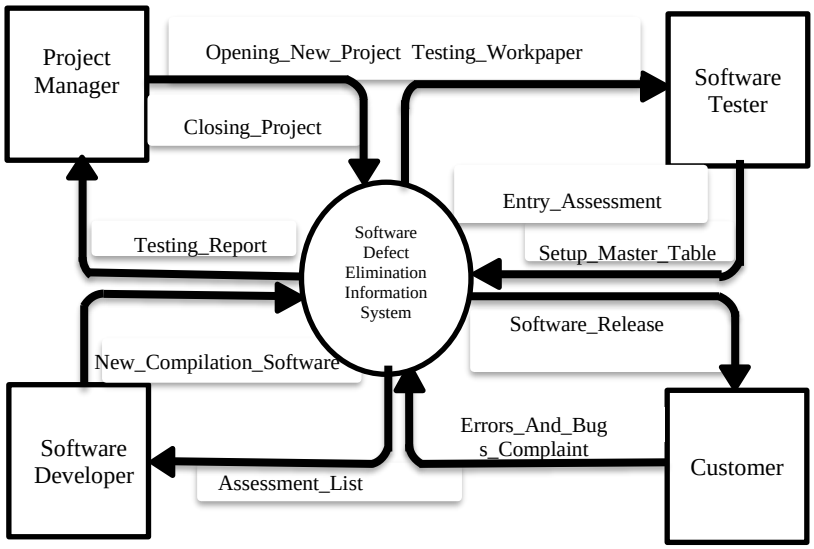


Figure 1 Context Diagram of Elimination Information

The context of the diagram describes the data flows between system entities. The context diagram of the Software Defect Elimination Information System in Software House describes the data that is related to software testing.

The context of the sister diagram above illustrates the relationship between the entities involved in the software testing that is carried out, namely: first, the project manager who receives an assignment from management to lead and manage a software development project including testing as an activity to realize customer satisfaction. The project manager is responsible for managing projects that are being started and will be completed. Second, software developers receive assignments from project managers who also serve as software architects, to determine the software architecture developed with standardization. Third, a software tester is a software test that is being developed, working together in an integrated manner. The fourth customer, namely the user of the software

developed, is coordinated by a counterpart from the customer side and works together with the project manager in completing the software project.

For Software Submission Monitoring, in the Information System, there are two monitoring tools, namely the Testing Score Evaluation Graph and the Testing Cost Evaluation Graph. Figure 2 Software Test Score Evaluation shows how many Test Iterations are performed so that the best Score can be achieved.

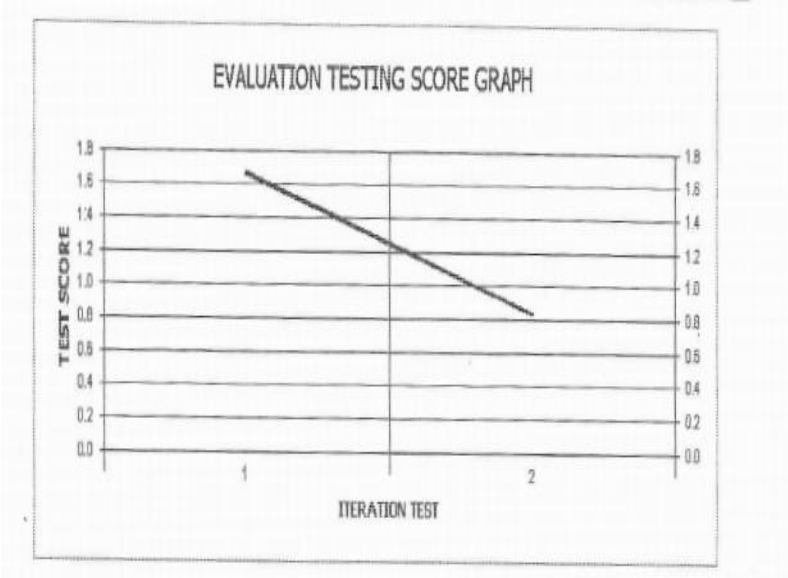
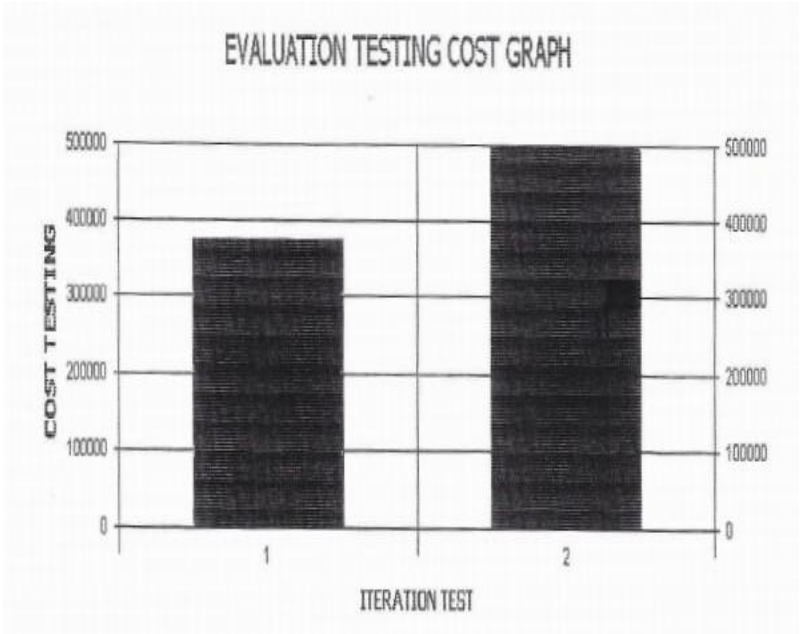


Figure 2 Evaluation of Software Testing Score

In addition to paying attention to the Software Testing Score Evaluation Graph, other oversight is needed regarding the cost of submission. In Graph 7. Describe the costs required for each Software Testing Iteration.



Graph 3 Software Testing Cost Evaluation

Benchmarking the quality of software testing requires standardization of assessment. The best quality assessment today uses *scoring techniques* because the technique is assumed to have high objectivity. After several tests, an *Assessment score* will be obtained. Assessment scores are used as an assessment for each activity, the goal is to avoid test failures so that other testing work is not wasted. After the assessment is passed with the conditions that meet the requirements, the test assessment is continued using weighting for each *testing factor* so that an index factor is obtained. Then the index factor is obtained, then the conclusion of software testing evaluation based on the evaluation criteria is a benchmark for the success rate of software development so that the goal of making *Zero Defect Application Software* is realized.

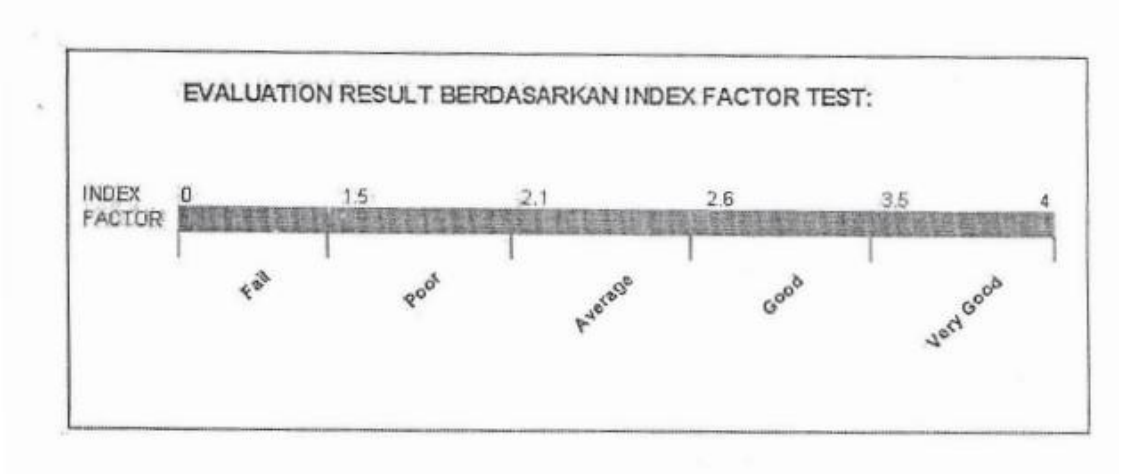


Figure 4 Evaluation Result based on Index Factor Test

Conclusion

The Software Defect Elimination Information System is designed to assist software developer stakeholders in taking action and planning for software testing. The system assists in evaluating tests, displaying graphs of test scores, as well as costs for considerations, and recording software that has been released and fixed for bugs or errors. By conducting software testing, it is proven that software defects can be eliminated. The magnitude of the defects eliminated depends on the number of test iterations performed and how many test techniques are empowered. The more testing techniques used, the higher the weight of the test.

Software testing is not just an individual activity, but a teamwork, so a proper and efficient organization is needed for the implementation of software testing. Stakeholders involved in this test include daily implementers or users, software engineers, project managers, and Quality Control staff or software testers. Software testing can be done by referring to the Matrix of strategies, methods, and techniques of software testing so that in practice testing on a software development project is adjusted to the cost and size of the project. The Software Defect Elimination Information System has a clear size and criteria to determine the quality of software development tests based on Index Factors and Evaluation Results so that Zero-Defect Application Software can be realized, which has an impact on customer satisfaction with the company.

With a strategy for software development testing, the need for human resources can be planned, and the testing stages can be carried out according to the available techniques. This system helps the tester team assess the software testing work objectively because the testing work paper data is stored in a database. Software testing iterations can be carried out as efficiently as possible with the help of the Software Defect Elimination Information System. Defects or bugs that arise after the software is released will be input into the database for re-analyzing software development or as a knowledge base for new software development testing work.

Bibliography

De, Sourav, Malik, Susanta, Ghosh, Aniruddha, Saha, Rumpa, & Saha, Bidyut. (2015). A review on natural surfactants. *RSC Advances*, *5*(81), 65757–65767.

Herdiyanto, Henri, & Lukman, Lukman. (2022). Sistem Informasi Pramuka Berbasis Website Menggunakan Rational Unified Process. *Building of Informatics, Technology and Science (BITS)*, *4*(2), 819–828.

Hutabarat, Julianus. (2022). *Pengantar Teknik Industri*. Media Nusa Creative (MNC Publishing).

Jauhari, Achmad, Anamisa, Devie Rosa, & Mufarroha, Fifin Ayu. (2022). *Rekayasa perangkat lunak*. Media Nusa Creative (MNC Publishing).

Mcleod Jr, Raymond. (1995). Systems theory and information resources management: integrating key concepts. *Information Resources Management Journal (IRMJ)*, *8*(2), 5–15.

Pamuji, Agus. (2018). Analisa Studi Empirik Kerangka Kerja Pengukuran Kualitas Perangkat Lunak Bebas Cacat. *Jurnal Informatika: Jurnal Pengembangan IT*, *3*(1), 130–135.

Prastyo, Edwin Hari Agus, Suhartono, Suhartono, Faisal, Muhammad, Yaqin, Muhammad Ainul, & Firdaus, Reza Augusta Jannatul. (2024). Naive Bayes Classification Untuk Prediksi Cacat Perangkat Lunak. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *9*(2), 782–791.

Reivaldi Kesuma Kagi, Muchammad Ficky Duskarnaen, & Hamidillah Ajie. (2020). Desain Dan Implementasi Pada Wifi Pustikom Free Access Di Pusat Teknologi Informasi Dan Komunikasi Universitas Negeri Jakarta Menggunakan Mikrotik Dan Wireshark Untuk Analisis Terhadap Serangan Packet Sniffing Dan Netcut. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, *4*(2), 37–40. https://doi.org/10.21009/pinter.4.2.7

Rifa'i, Muhammad, Ananda, Rusydi, & Fadhli, Muhammad. (2018). *Manajemen peserta didik (Pengelolaan peserta didik untuk efektivitas pembelajaran)*.

Sudipa, I. Gede Iwan, Ariantini, Made Suci, Pomalingo, Suwito, Ridwan, Achmad, Primasari, Dewi, Ariana, Anak Agung Gede Bagus, Ibrahim, Rohmat Nur, Ilham, Rozali, Arsana, I. Nyoman Alit, & Irmawati, Irmawati. (2023). *Buku Ajar Rekayasa Perangkat Lunak*. PT. Sonpedia Publishing Indonesia.