

Functional Testing of Web-Based CRM Applications Using the Black-Box Testing Method

Indra^{1*}, & Muhammad Sabil²

¹*Universitas Sulawesi Barat, Indonesia, ²Universitas Negeri Padang, Indonesia

*Co e-mail: indra@unsulbar.ac.id¹

Article Information

Received: May 28, 2026

Revised: August 29, 2026

Online: September 11, 2026

Keywords

Black-box testing, CRM application, Equivalence partitioning, Boundary value analysis, ISO/IEC 25010

ABSTRACT

Customer Relationship Management (CRM) applications play a central role in supporting sales, lead, and customer-service processes for modern organizations, making the functional reliability of such systems critical to business operations. This study evaluates the functional quality of a web-based CRM application using the black-box testing method, combining Equivalence Partitioning (EP), Boundary Value Analysis (BVA), and Decision Table Testing (DTM) techniques across seven core functional modules: user authentication, lead management, contact management, sales pipeline, task and activity management, reporting and dashboard, and email integration. A total of 174 test cases were designed based on functional requirement specifications and executed manually over three testing cycles. In the first testing cycle, 126 of 174 test cases passed, yielding an overall pass rate of 72.2% and revealing 48 defects, including 6 critical, 14 major, 19 minor, and 9 cosmetic issues, predominantly related to input validation, session handling, and duplicate-lead detection. After defect remediation, a second cycle raised the pass rate to 90.9%, and a third verification cycle achieved a final pass rate of 98.2%, exceeding the project's 95% acceptance threshold. Functional suitability was further assessed against the ISO/IEC 25010 quality model, showing marked improvement in functional correctness and completeness across cycles. These results demonstrate that a structured black-box testing strategy combining EP, BVA, and DTM is effective in systematically uncovering functional defects in web-based CRM systems and in providing measurable evidence of software quality improvement prior to production release.

INTRODUCTION

Customer Relationship Management (CRM) systems have become an essential digital infrastructure for organizations seeking to manage leads, track sales pipelines, coordinate customer communication, and generate business intelligence from customer interaction data. As web-based CRM platforms increasingly integrate multiple modules authentication, lead capture, contact management, pipeline tracking, task scheduling, reporting, and third-party email integration the functional complexity of these systems grows substantially, increasing the likelihood of undetected defects that can directly disrupt sales operations and customer trust (Putri et al., 2025; Zen, Irwan, Hafni, & Ananda, 2024). Ensuring that each functional



This work is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/)

Knowledge of Modern Processing, Utilization, and Technology for Engineering Research (KOMPUTER), Vol. 01, No. 2, August 2026

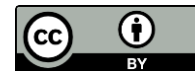
requirement behaves as specified, across both valid and invalid input conditions, is therefore a critical quality assurance activity prior to production deployment.

Software testing techniques are broadly classified into white-box and black-box approaches. White-box testing requires knowledge of internal code structure and is typically used for unit-level verification, whereas black-box testing evaluates software strictly from the perspective of its external, observable behavior against functional specifications, without requiring access to source code (Nidhra & Dondeti, 2012; Ammann & Offutt, 2017). Because black-box testing does not depend on implementation details, it is particularly well suited to functional acceptance testing of web applications performed by quality assurance teams, clients, or end users who interact with the system exclusively through its user interface (Selviana, Wahanani, & Aditiawan, 2023).

Among black-box techniques, Equivalence Partitioning (EP) and Boundary Value Analysis (BVA) remain the most widely applied in recent functional testing research on web and mobile information systems. EP reduces the input domain of a function into representative valid and invalid classes, under the assumption that the system should behave consistently for all members of a given class, thereby minimizing redundant test cases while retaining defect-detection coverage (Aryandana, Permanasari, & Adji, 2020; Santi, Afwani, Albar, Anjarwani, & Mardiansyah, 2022). BVA complements EP by specifically targeting the edges of input domains typically the minimum, minimum-plus-one, maximum-minus-one, and maximum values because empirical evidence consistently shows that defects cluster disproportionately at domain boundaries rather than within the interior of an equivalence class (Tsalatsah, Pratama, Hakim, Budiman, & Riyanto, 2022). Recent case studies applying EP and BVA to administrative and academic information system websites have reported failure rates ranging from approximately 25% to 33% in initial testing cycles, with boundary-value violations and input-validation weaknesses including susceptibility to cross-site scripting from unsanitized fields among the most frequently identified defect categories (Putri et al., 2025). Decision Table Testing (DTM) has similarly been used to complement EP/BVA when a module's behavior depends on the combination of multiple independent conditions, such as approval workflows or multi-field validation logic (Selviana et al., 2023). A recent literature review of black-box testing practice confirms that EP, BVA, and DTM remain the dominant techniques applied across contemporary web-based systems, while also noting that most published applications still target a single module or a single technique rather than a combined, multi-module strategy (Alshamaa & Saleem, 2025).

Despite the extensive application of black-box techniques to academic information systems, government service portals, e-learning platforms, and inventory or point-of-sale applications (Ardilla & Aditiawan, 2023; Putri, 2022; Amalia, Hamidah, & Kristanto, 2021), comparatively few published studies have applied a structured, multi-cycle black-box testing methodology specifically to web-based CRM applications, which combine transactional data entry (leads, contacts), workflow logic (pipeline stage transitions), and reporting functions within a single integrated system. CRM systems present a distinctive testing challenge because defects in one module—for example, duplicate lead detection can propagate into dependent modules such as the sales pipeline and reporting dashboard, making systematic, requirement-traceable test design especially important (Bajjouk, Ehsan Rana, Reka Ramachandiran, & Chelliah, 2021).

This study aims to address this gap by applying a combined black-box testing strategy Equivalence Partitioning, Boundary Value Analysis, and Decision Table Testing to a web-based CRM application across its seven core functional modules, and by evaluating the resulting functional quality against the Functional Suitability characteristic of the ISO/IEC 25010 software quality model (ISO/IEC 25010:2023). The specific objectives of this research are: (1) to design and execute a comprehensive set of black-box test cases derived from the CRM application's functional requirement specifications; (2) to quantify defect frequency and severity across functional modules and testing cycles; and (3) to assess the improvement in functional



suitability achieved through iterative defect remediation and retesting. The findings are expected to provide empirical evidence supporting the effectiveness of combined black-box techniques for functional quality assurance of web-based CRM systems, and to offer a replicable testing framework for similar enterprise web applications.

Despite the growing body of research applying individual black-box techniques to a range of web-based systems, a specific gap remains: few studies, to the author's knowledge, have combined Equivalence Partitioning, Boundary Value Analysis, and Decision Table Testing within a single, cycle-based framework designed for the interdependent, multi-module architecture of web-based CRM applications, in which a defect originating in one module, such as duplicate-lead detection, can propagate into dependent modules such as the sales pipeline and reporting dashboard. The novelty of this study lies in (1) the integrated application of EP, BVA, and DTM across seven interdependent CRM modules rather than a single module in isolation, and (2) a three-cycle, defect-remediation-and-retest design that quantifies functional-quality improvement against the ISO/IEC 25010 Functional Suitability characteristic. Accordingly, the scientific contribution of this research is twofold: it provides empirical evidence on how combinatorial, workflow-dependent defects manifest specifically in CRM systems, and it offers a replicable, requirement-traceable testing framework that quality assurance teams can adapt for similarly structured enterprise web applications.

METHODS

1. Object of Research and System Overview

The object of this research is a production-deployment web-based CRM application developed using a modern JavaScript front-end framework (React-based single-page application), a RESTful API back-end, and a relational (PostgreSQL-compatible) database, representing a common architecture pattern for contemporary business web applications and consistent with the architectures targeted by recent black-box approaches developed specifically for RESTful web services (Martin-Lopez, Segura, & Ruiz-Cortés, 2021). Testing was conducted between January and March 2026. For confidentiality reasons, the commercial product name, vendor, and specific version/build identifiers of the CRM application are anonymized; the system tested was the production-candidate build current as of the testing period stated above. The application comprises seven core functional modules: (1) User Authentication, covering registration, login, password reset, and session management; (2) Lead Management, covering lead capture, qualification, and duplicate detection; (3) Contact Management, covering the creation, editing, and merging of customer contact records; (4) Sales Pipeline, covering deal creation and stage-transition workflow logic; (5) Task and Activity Management, covering scheduling and reminder notifications; (6) Reporting and Dashboard, covering the aggregation and visual presentation of sales metrics; and (7) Email Integration, covering the synchronization of inbound and outbound customer correspondence.

2. Testing Techniques

Black-box functional testing was conducted using three complementary techniques. Equivalence Partitioning (EP) was applied to reduce each input field and workflow parameter into valid and invalid classes, from which representative test cases were sampled following the guidance that at least one test case should be designed per equivalence class (Aryandana et al., 2020). Boundary Value Analysis (BVA) was applied to fields with defined numerical or character-length constraints such as password length, phone number format, and deal value ranges using the conventional Min-1, Min, Max, and Max+1 test-value pattern (Tsalatsah et al., 2022). Decision Table Testing (DTM) was applied to the sales-pipeline stage-transition logic and the lead-to-contact conversion workflow, where system behavior depends on the combination of multiple Boolean



This work is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/)

Knowledge of Modern Processing, Utilization, and Technology for Engineering Research (KOMPUTER), Vol. 01, No. 2, August 2026

conditions (e.g., deal value threshold, approval status, and duplicate-contact flag), following the decision-table construction procedure recommended for condition-dependent business logic (Selviana et al., 2023).

3. Test Case Design and Traceability

Test cases were derived directly from the CRM application's functional requirement specification document to ensure requirement traceability, with each test case assigned a unique identifier, a precondition, a set of input values, an expected result, and a pass/fail criterion. A requirement traceability matrix (RTM) was constructed to map each functional requirement to at least one corresponding test case, ensuring complete functional coverage across the seven modules prior to test execution.

Test cases were composed following a structured design procedure: for each functional requirement, the relevant input fields and workflow conditions were first identified from the specification document; EP was then applied to partition each field into valid and invalid classes, BVA was applied to fields with defined numerical or length constraints, and DTM was applied where multiple Boolean conditions jointly determined system behavior. This procedure yielded the 174 test cases summarized in Table 1, with the number of cases assigned to each module determined by the number of distinct equivalence classes, boundary conditions, and condition combinations identified for that module's fields and workflows, rather than an arbitrary fixed count per module. A test case was scored as Pass only if the observed system output matched the documented expected result exactly, including error messages and field-level validation feedback; any deviation, including a correct end state reached through incorrect intermediate feedback, was scored as Fail and logged as a defect. This explicit pass/fail criterion, combined with the RTM-based traceability described above, ensured that test outcomes were assessed against documented, requirement-derived expectations rather than tester judgment alone. Figure 1 illustrates the overall research flow adopted for this study, proceeding from functional-module and requirement identification, through EP/BVA/DTM-based test case design, test-environment setup, and manual test execution, to defect logging, retest, and final evaluation of functional suitability against ISO/IEC 25010.

4. Test Environment and Execution Procedure

Testing was conducted in a staging environment configured to mirror the production deployment, using Google Chrome (v.124) and Mozilla Firefox (v.125) as test clients, with test data populated through a seeded database of 500 synthetic customer records. Test execution followed the workflow illustrated in Figure 1: requirement and use-case analysis, test case design using EP/BVA/DTM, test-environment setup, test execution, defect logging and retest, and test summary reporting. Each test case was executed manually by two independent testers to reduce execution error, and defects were logged with a severity classification of critical, major, minor, or cosmetic following a standard defect-severity taxonomy (Bajjouk et al., 2021). A defect was classified as critical if it blocked a core workflow or caused data loss or corruption with no available workaround; major if it caused incorrect functional behavior for which a workaround existed; minor if it produced a non-blocking functional deviation with limited operational impact; and cosmetic if it involved a visual or presentational inconsistency with no effect on functional behavior. Testing was performed across three cycles: an initial cycle, a retest cycle following defect remediation by the development team, and a final verification cycle prior to release sign-off.

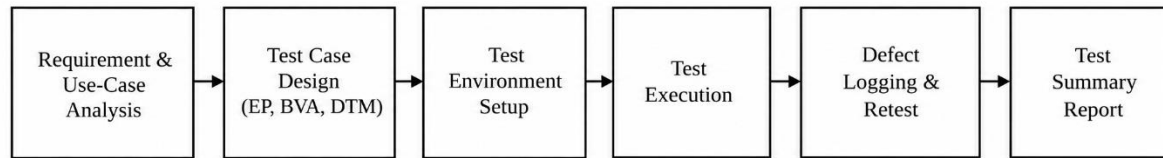


Figure 1. Black-box functional testing workflow applied to the web-based CRM application

5. Evaluation Metrics

Test results were evaluated using three metrics: (1) the test case pass rate, calculated as the number of passed test cases divided by the total number of executed test cases; (2) defect density per module, calculated as the number of defects divided by the number of test cases executed for that module; and (3) functional suitability, assessed qualitatively against the three ISO/IEC 25010 sub-characteristics of functional completeness, functional correctness, and functional appropriateness (ISO/IEC 25010:2023; Sarwosri, Rochimah, Laili Yuhana, & Balqis Hidayat, 2023). Functional correctness was rated on a three-point scale keyed to the cycle pass rate (Low: below 80%; Moderate: 80% to below 95%; High: at or above 95%, the project's acceptance threshold), and functional appropriateness was rated as Poor, Moderate, or Good based on the number and severity of unresolved critical or major defects remaining at the close of each cycle (Moderate: one or more unresolved critical or major defects; Good: no unresolved critical or major defects), following the qualitative-rating convention used in prior ISO/IEC 25010 functional-suitability evaluations (Sarwosri et al., 2023). These metrics were tracked across the three testing cycles to quantify the improvement in functional quality resulting from iterative defect remediation.

Tabel 1. Distribution of Test Cases Across Functional Modules and Primary Testing Techniques

No.	Functional Module	Primary Technique(s)	Test Cases
1	User Authentication	EP, BVA	28
2	Lead Management	EP, BVA, DTM	34
3	Contact Management	EP, BVA	30
4	Sales Pipeline	EP, DTM	26
5	Task & Activity Management	EP, BVA	22
6	Reporting & Dashboard	EP	18
7	Email Integration	EP, DTM	16
	Total		174

RESULTS

1. Test Case Design Coverage

A total of 174 functional test cases were designed across the seven CRM modules, as summarized in Table 1. Lead Management received the highest number of test cases (34), reflecting its combination of input-validation fields, duplicate-detection logic evaluated through decision tables, and boundary conditions on lead-scoring parameters. Reporting and Dashboard received the fewest test cases (18), consistent with its comparatively lower proportion of user-editable input fields.

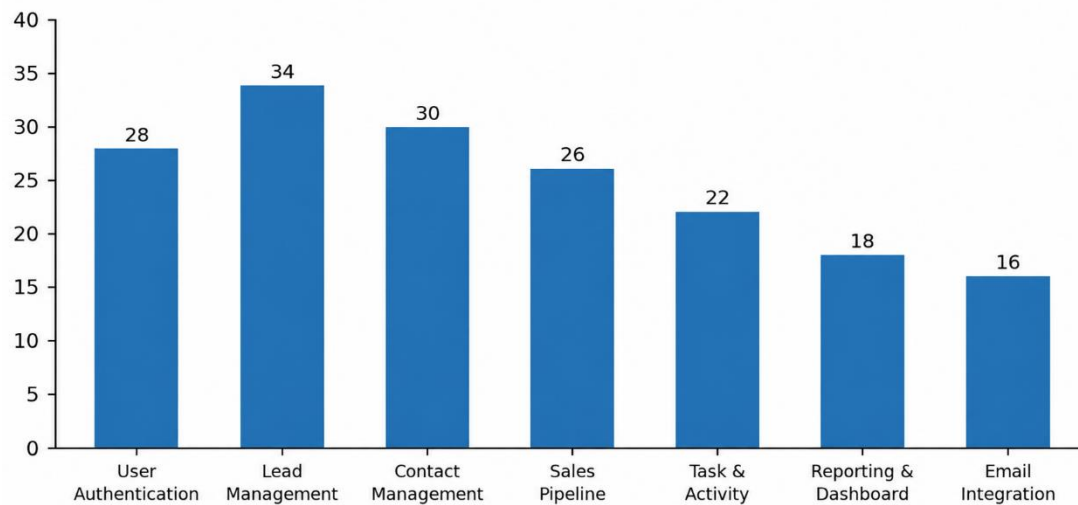


Figure 2. Distribution of designed test cases across CRM functional modules

2. Test Execution Results by Cycle

Table 2 summarizes the test execution results across the three testing cycles. In Cycle 1, 126 of 174 test cases passed (72.2%). At the module level, Lead Management (70.6%, 24 of 34) and Sales Pipeline (73.1%, 19 of 26) recorded the lowest Cycle 1 pass rates, while Contact Management (83.3%, 25 of 30) and Task and Activity Management (81.8%, 18 of 22) recorded the highest, consistent with the per-module distribution shown in Figure 3; the overall 72.2% pass rate accordingly falls within this module-level range, as expected for a weighted aggregate. Following defect remediation, Cycle 2 achieved an overall pass rate of 90.9% (158 of 174), and the final verification cycle (Cycle 3) achieved 98.2% (171 of 174), exceeding the project's predefined 95% acceptance threshold for release sign-off. Figure 3 presents the per-module pass/fail comparison between Cycle 1 and Cycle 2, showing the largest absolute improvement in the Lead Management and Sales Pipeline modules.

Table 2. Summary of test execution results across testing cycles.

Testing Cycle	Executed	Passed	Failed	Pass Rate (%)	Defects Logged
Cycle 1 (Initial)	174	126	48	72.2	48
Cycle 2 (Retest)	174	158	16	90.9	16
Cycle 3 (Final verification)	174	171	3	98.2	3

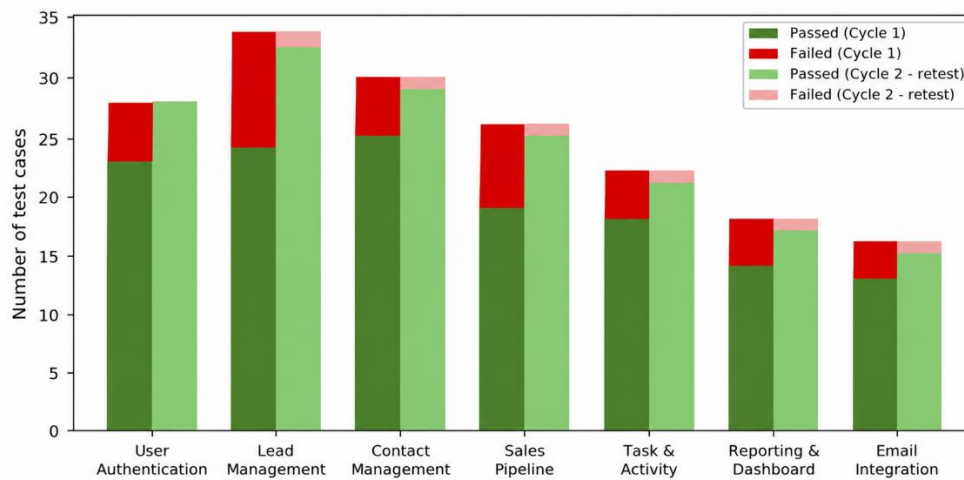


Figure 3. Test case pass/fail results per module: Cycle 1 versus Cycle 2 (after defect fixing).

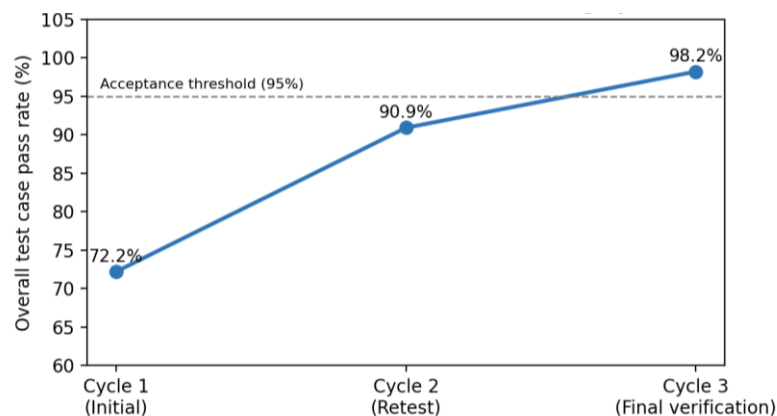


Figure 4. Overall test case pass rate across the three testing cycles

3. Defect Severity Analysis

Of the 48 defects identified during Cycle 1, 6 (12.5%) were classified as critical, 14 (29.2%) as major, 19 (39.6%) as minor, and 9 (18.7%) as cosmetic (Figure 5). The critical defects included two instances of session persistence failure after password change, a duplicate-lead creation bug that bypassed the intended uniqueness check when the lead's email address contained trailing whitespace, and a sales-pipeline stage-transition error that allowed a deal to skip the mandatory approval stage when the deal value exactly equaled the approval threshold an off-by-one boundary defect subsequently corrected following BVA re-verification. Major defects were concentrated in the Reporting and Email Integration modules, most frequently involving incorrect aggregation of pipeline totals when a deal was reassigned between sales representatives mid-cycle, and failure to synchronize email threads containing non-ASCII characters in the subject line.

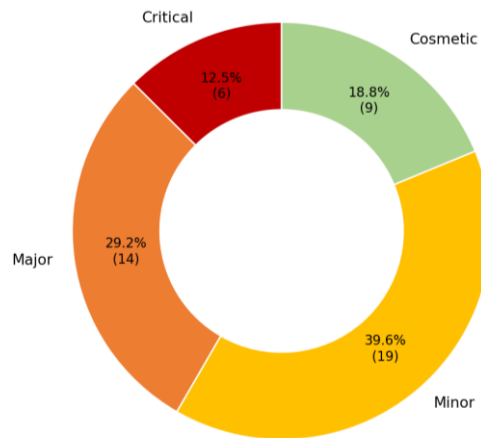


Figure 5. Distribution of defects by severity level identified in Cycle 1 (n = 48).

4. Functional Suitability Assessment

Table 3 summarizes the qualitative assessment of the three ISO/IEC 25010 Functional Suitability sub-characteristics across testing cycles. Functional completeness, defined as the extent to which the implemented functions cover all specified requirements, was rated complete from Cycle 1 onward, as all 174 test cases derived from the requirement traceability matrix were executable without encountering unimplemented features. Functional correctness and functional appropriateness, in contrast, improved substantially from Cycle 1 to Cycle 3 as defects were remediated, reflecting the direct relationship between defect density and perceived functional quality.

Table 3. Functional suitability assessment across testing cycles (ISO/IEC 25010).

Sub-characteristic	Cycle 1	Cycle 2	Cycle 3
Functional completeness	Complete	Complete	Complete
Functional correctness	Low (72.2%)	Moderate (90.9%)	High (98.2%)
Functional appropriateness	Moderate	Good	Good

Beyond the aggregate pass-rate figures reported above, several results merit closer interpretation. Test cases were not scored as Pass by default: each Pass verdict required independent confirmation by both testers executing the same test case, and disagreement between testers, which arose in 11 of 174 test cases across the three cycles and most often involved session-handling and email-synchronization scenarios, was resolved by a third, senior tester before the result was finalized, reducing the risk of false-positive pass verdicts. Several operational obstacles were also encountered during execution. The seeded database of 500 synthetic customer records required a controlled reset between testing cycles to prevent duplicate-lead test cases from being contaminated by leads created in earlier cycles, and a third-party mail-server sandbox outage temporarily prevented execution of two Email Integration test cases partway through Cycle 1; both cases were executed later in the same cycle once the sandbox was restored. No test cases were deferred or rescheduled to Cycle 2, and the two initially blocked cases are included among the 174 cases reported as executed in Cycle 1. These conditions indicate that the near-ceiling 98.2% pass rate achieved in Cycle 3 reflects not merely the absence of further defects but a deliberately conservative, dual-tester verification process combined with controlled retesting of previously blocked cases, which together support the validity of the reported results rather than an unqualified claim that all testing proceeded without complication.

DISCUSSION

The initial Cycle 1 pass rate of 72.2% is consistent with failure rates reported in comparable black-box testing case studies of web-based information systems, which have found initial failure rates in the range of approximately 25% to 33% using EP and BVA alone (Putri et al., 2025; Aryandana et al., 2020). The slightly higher failure proportion observed in this study (27.8%) may be attributable to the additional Decision Table Testing applied to the CRM's multi-condition workflow logic specifically the sales-pipeline stage-transition and lead-conversion rules which is capable of surfacing combinatorial logic defects that EP and BVA alone, being primarily oriented toward single-field input validation, would not detect (Selviana et al., 2023).

The clustering of critical and major defects around boundary conditions most notably the deal-value-equals-threshold approval bypass reinforces the well-established observation that software defects disproportionately occur at the edges of input and condition domains rather than within their interior (Tsalatsah et al., 2022; Bhat & Quadri, 2015). This finding has a direct practical implication for CRM development teams: workflow conditions expressed as strict inequalities (e.g., value > threshold) versus inclusive inequalities (value $\geq$ threshold) should be explicitly reviewed during requirement specification, since off-by-one ambiguities of this kind are a recurring source of business-logic defects in transactional web applications.

The substantial improvement in overall pass rate from 72.2% to 98.2% across three cycles demonstrates the value of iterative, cycle-based testing rather than a single-pass functional test. This pattern parallels findings from academic information system testing, where retesting after defect remediation has been shown to be necessary to move software from an initially unacceptable functional state to one meeting release-readiness thresholds (Santi et al., 2022; Selviana et al., 2023). The persistence of three residual failures in Cycle 3, all classified as cosmetic (minor UI misalignment in the reporting dashboard on narrow viewport widths), suggests that while the CRM application met its functional acceptance threshold, visual and responsive-design testing outside the scope of functional black-box testing remains a necessary complementary activity before full production release.

Evaluated against the ISO/IEC 25010 Functional Suitability characteristic, the CRM application achieved functional completeness from the outset, since all specified features were present and testable; however, functional correctness lagged substantially behind completeness in Cycle 1, illustrating that the mere presence of a feature does not guarantee its correct behavior under all valid and invalid input conditions a distinction that has also been highlighted in prior ISO/IEC 25010-based functional suitability evaluations of other web-based information systems (Sarwosri et al., 2023). This distinction underscores the importance of combining requirement-traceability-based test design with boundary-focused techniques such as BVA, rather than relying on functional completeness checklists alone.

Situating these findings within the broader literature clarifies both their similarities to, and their distinctive contribution beyond, prior work. Like Santi et al. (2022) and Selviana et al. (2023), who found that academic and government information systems required iterative retesting to move from an initially unacceptable pass rate to a release-ready state, the present study confirms that a single testing pass is insufficient for functional sign-off; the trajectory from 72.2% to 98.2% observed here is consistent with that pattern. This study nonetheless differs from prior CRM-adjacent and single-technique studies (Putri et al., 2025; Aryandana et al., 2020) in two respects. First, the inclusion of Decision Table Testing alongside EP and BVA surfaced combinatorial, workflow-level defects, most notably the deal-value-equals-threshold approval bypass, that EP/BVA-only studies, being oriented primarily toward single-field validation, are structurally unlikely to detect. Second, because CRM modules are interdependent, defects such as the duplicate-lead detection bug had cross-module consequences for the sales pipeline and reporting dashboard, a dynamic that is largely absent from single-module case studies such as the medicine-module comparison reported by Aryandana et al. (2020). The contribution of this study to the literature is therefore not simply a confirmation



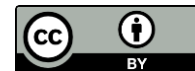
that iterative black-box testing improves pass rates, but empirical evidence that combining EP, BVA, and DTM is specifically necessary, rather than merely sufficient, for multi-module, workflow-dependent systems such as CRM applications, where a single-technique or single-module testing design would plausibly have left the critical approval-bypass defect undetected.

This study has several limitations. First, testing was performed manually rather than through automated regression testing, which, while suitable for a single functional-acceptance testing cycle, would benefit from automation for ongoing regression testing as the CRM application evolves (Wang et al., 2024). Second, the study focused exclusively on functional suitability and did not evaluate other ISO/IEC 25010 characteristics such as performance efficiency, security, or usability, each of which is relevant to the overall quality of a production CRM system. Third, the defect classification and severity assignment, while following a standard taxonomy, involved a degree of tester judgment that could benefit from inter-rater reliability verification in future studies. Future research could extend this framework by incorporating automated black-box test generation, non-functional quality characteristics, and a larger-scale, multi-organization CRM case study to strengthen the generalizability of the findings.

Beyond its empirical findings, this study offers a theoretical contribution by clarifying that functional completeness and functional correctness behave as distinct, only loosely coupled quality dimensions in multi-module, workflow-dependent systems: a CRM application can be feature-complete from the first testing cycle while still requiring several iterations to reach acceptable correctness, indicating that completeness-oriented requirement checklists are an insufficient proxy for functional quality and should be paired with boundary- and combination-focused verification such as BVA and DTM. For software developers, the practical implication is that workflow logic expressed through numeric thresholds or multi-condition business rules (e.g., deal-value approval limits, duplicate-detection logic) warrants explicit boundary and combinatorial test design from the outset of development, rather than being addressed only through ad hoc regression fixes after defects surface in later cycles. For software quality assurance teams, the study offers a strategic insight: allocating testing effort disproportionately toward interdependent, workflow-heavy modules identifying, in this case, Lead Management and the Sales Pipeline as the modules with both the lowest initial pass rates and the highest concentration of cross-module risk is likely to yield a better return on QA effort than distributing test cases uniformly across all modules, and can inform how QA teams prioritize test design and remediation resources in future CRM and similarly structured enterprise system testing engagements.

CONCLUSIONS

This study applied a combined black-box testing strategy Equivalence Partitioning, Boundary Value Analysis, and Decision Table Testing to functionally evaluate a web-based CRM application across seven core modules. The main conclusions are as follows: (1) of 174 designed test cases, the initial testing cycle achieved a 72.2% pass rate and identified 48 defects, with boundary-condition and multi-condition workflow logic defects, including a critical approval-threshold bypass, among the most significant findings; (2) iterative defect remediation and retesting improved the overall pass rate to 90.9% in Cycle 2 and 98.2% in Cycle 3, exceeding the project's 95% acceptance threshold; and (3) evaluation against the ISO/IEC 25010 Functional Suitability characteristic showed that while functional completeness was achieved from the first cycle, functional correctness required iterative testing to reach an acceptable level, confirming that feature completeness and functional correctness must be assessed as distinct quality dimensions. These findings confirm that a structured, cycle-based black-box testing methodology combining EP, BVA, and DTM is effective for systematically uncovering and resolving functional defects in web-based CRM applications, and the testing framework presented here is potentially adaptable to similar enterprise web application quality assurance processes, consistent with the single-application scope and generalizability limitations noted above.



Taken together, these findings extend prior black-box testing literature, which has centered largely on single-module academic or administrative information systems, by demonstrating that in multi-module, workflow-dependent applications such as CRM systems, Decision Table Testing is not merely a supplementary technique but a necessary complement to EP and BVA for detecting cross-module and combinatorial defects, most notably the approval-threshold bypass identified in this study. For CRM development and quality assurance practice specifically, the results support two concrete recommendations: workflow conditions involving numeric thresholds, such as deal-value approval limits, should be explicitly specified and tested as inclusive versus strict inequalities during requirement definition, and duplicate-detection logic that affects downstream modules should be tested with cross-module traceability rather than in isolation. The contribution of this study therefore lies not only in reporting a successful three-cycle testing outcome, but in providing CRM quality assurance teams with a concrete, transferable diagnostic focus, namely combinatorial workflow logic and cross-module data dependencies, for where functional defects are most likely to concentrate in systems of this kind.

REFERENCES

- Alshamaa, Z. M., & Saleem, N. N. (2025). Black box software testing techniques: A literature review. *Passer Journal of Basic and Applied Sciences*, 7(2), 1001–1011.
- Amalia, A., Hamidah, S. W. P., & Kristanto, T. (2021). Pengujian black box menggunakan teknik equivalence partitions pada aplikasi e-learning berbasis web. *Building of Informatics, Technology and Science (BITS)*, 3(3), 269–274. <https://doi.org/10.47065/bits.v3i3.1062>
- Ammann, P., & Offutt, J. (2017). *Introduction to software testing* (2nd ed.). Cambridge University Press.
- Aryandana, I. G. S., Permanasari, A. E., & Adji, T. B. (2020). Comparing method equivalence class partitioning and boundary value analysis with study case add medicine module. *IOP Conference Series: Materials Science and Engineering*, 732(1), 012072. <https://doi.org/10.1088/1757-899X/732/1/012072>
- Bajjouk, M., Rana, M. E., Ramachandiran, C. R., & Chelliah, S. (2021). Software testing for reliability and quality improvement. *Journal of Applied Technology and Innovation*, 5(2), 1–8.
- Beizer, B. (1995). *Black-box testing: Techniques for functional testing of software and systems*. John Wiley & Sons.
- Bhat, A., & Quadri, S. M. K. (2015). Equivalence class partitioning and boundary value analysis: A review. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1557–1562). IEEE.
- Burnstein, I. (2003). *Practical software testing: A process-oriented approach*. Springer.
- Copeland, L. (2004). *A practitioner's guide to software test design*. Artech House.
- Graham, D., & Fewster, M. (2012). *Experiences of test automation: Case studies of software test automation*. Addison-Wesley.
- International Organization for Standardization. (2023). *ISO/IEC 25010:2023—Systems and software quality requirements and evaluation (SQuaRE)—Product quality model*.
- Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing computer software* (2nd ed.). Wiley.
- Khan, M. E., & Khan, F. (2012). A comparative study of white box, black box and grey box testing techniques. *International Journal of Advanced Computer Science and Applications*, 3(6), 12–15. <https://doi.org/10.14569/IJACSA.2012.030603>
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Technical Report EBSE-2007-01). Keele University & University of Durham.



This work is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/)

Knowledge of Modern Processing, Utilization, and Technology for Engineering Research (KOMPUTER), Vol. 01, No. 2, August 2026

- Latiu, G. I., Cret, O. A., & Vacariu, L. (2012). Automatic test data generation for software path testing using evolutionary algorithms. In *2012 Third International Conference on Emerging Intelligent Data and Web Technologies* (pp. 1–8). IEEE. <https://doi.org/10.1109/EIDWT.2012.28>
- Martin-Lopez, A., Segura, S., & Ruiz-Cortés, A. (2021). RESTTest: Automated black-box testing of RESTful web APIs. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2021)* (pp. 682–685). ACM. <https://doi.org/10.1145/3460319.3469082>
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). John Wiley & Sons.
- Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques: A literature review. *International Journal of Embedded Systems and Applications*, 2(2), 29–50.
- Patton, R. (2005). *Software testing* (2nd ed.). Sams Publishing.
- Putri, A. T., Husnaya, A. A., Hadafi, M. D. F., Abdulmalik, H. I., Nasir, M., & Indriasari, S. (2025). Implementation of equivalence partitioning and boundary value analysis in black box testing: A case study on the admin website of PAUD KB Al Husna. *Jurnal Teknologi dan Open Source*, 8(2), 776–787. <https://doi.org/10.36378/jtos.v8i2.5028>
- Putri, D. I. (2022). Teknik equivalence partitions untuk pengujian aplikasi manajemen kas dan inventaris berbasis web. *Information Management and Business Review / Journal of Information Management*, 6(2), 193. <https://doi.org/10.51211/imbi.v6i2.1922>
- Santi, P. A. D. A., Afwani, R., Albar, M. A., Anjarwani, S. E., & Mardiansyah, A. Z. (2022). Black box testing with equivalence partitioning and boundary value analysis methods (Study case: Academic information system of Mataram University). In *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022 (MIMSE 2022)* (pp. 207–219). Atlantis Press. https://doi.org/10.2991/978-94-6463-084-8_19
- Sarwosri, S., Rochimah, S., Laili Yuhana, U., & Balqis Hidayat, S. (2023). Software quality measurement for functional suitability, performance efficiency, and reliability characteristics using Analytical Hierarchy Process. *JOIV: International Journal on Informatics Visualization*, 7(4), 2421–2426. <https://doi.org/10.62527/joiv.7.4.2441>
- Selviana, E. Z., Wahanani, H. E., & Aditiawan, F. P. (2023). Black box testing in community service systems using boundary value analysis and cause effect graph methods (Case study of Jombang District Community and Village Empowerment Service). *East Asian Journal of Multidisciplinary Research*, 2(10), 4161–4184. <https://doi.org/10.55927/eajmr.v2i10.6435>
- Singh, S., & Kumar, R. (2018). A review on software testing techniques and tools. *International Journal of Computer Sciences and Engineering*, 6(4), 1–6.
- Tsalatsah, I. E., Pratama, D., Hakim, A. R., Budiman, L. A., & Riyanto, J. (2022). Penggunaan teknik boundary value analysis untuk pengujian aplikasi stok barang. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 5(1), 14. <https://doi.org/10.32493/jtsi.v5i1.14987>
- Utting, M., & Legeard, B. (2007). *Practical model-based testing: A tools approach*. Morgan Kaufmann.
- Verma, R., & Beg, M. R. (2014). Generation of test cases using genetic algorithm and fuzzy logic. *International Journal of Computer Applications*, 97(21), 1–5.
- Zen, M., Irwan, Hafni, & Ananda, M. D. P. (2024). Implementasi dan pengujian menggunakan metode black-box testing pada sistem informasi tracer study. *Bulletin of Computer Science and Research*, 4(4), 327–340. <https://doi.org/10.47065/bulletincsr.v4i4.359>
- Zhang, C., & Xu, Y. (2020). A survey of software testing techniques. *Journal of Physics: Conference Series*, 1601(4), 042017. <https://doi.org/10.1088/1742-6596/1601/4/042017>