Scholar Publishing Group
International Journal of Business Management and Economics and Trade
https://doi.org/10.38007/IJBMET.2025.060117
ISSN 2790-0959 Vol. 6, Issue 1:169-178

Research on Building and Optimizing Software Testing Asset Library for Financial Scenarios

Ziyi Song

1995 Turk St, Apt 4, San Francisco, 94115, California, US

Keywords: Bank software testing, Process optimization, Test asset library, TOPSIS evaluation

Abstract: With the continuous development of technologies such as 5G communication, blockchain and artificial intelligence, the business structure and operation mode of traditional commercial banks are undergoing profound changes. Technological progress has brought about efficient and convenient service experiences, but it has also exposed the banking industry to a new competitive landscape and management pressure. The banking industry is strictly regulated, with a large number of systems that are closely interrelated. Any minor malfunction may affect the overall operation and threaten the safety of customer funds. In such a high-risk environment, banks undertake the dual tasks of improving quality and accelerating delivery in software development and testing. Due to the complex organizational structure, independent departmental division of labor, and low communication efficiency, testing work is difficult to be consistent with development and business. This paper, based on the software testing practices of financial institutions, proposes a testing improvement plan for financial scenarios. With process optimization and asset accumulation as the core concepts, it combines the TMMi model and DevOps ideas, integrates testing activities throughout the entire project cycle, and enhances the stability and reliability of testing through means such as automated script risk control and use case reuse. The research, based on the analysis of the financial industry and the comprehensive evaluation using the TOPSIS method, shows that the optimized testing system has significant improvements in defect detection efficiency, execution speed, and system stability, verifying the application value of this scheme in enhancing the quality of financial software testing and supporting digital transformation.

1. Introduction

With the continuous development of fintech, the business models, service processes and operational systems of commercial banks are undergoing profound transformation. The application of technologies such as mobile communication, cloud computing, artificial intelligence and biometric recognition has accelerated the iteration speed of financial products, while also driving the rapid growth of the number of customers and capital flows. This places higher demands on the stability, security and operational efficiency of the bank's information system. As a strictly regulated industry, any problem in the bank's large and highly interrelated business system at any

stage may trigger systemic risks and affect a large number of customers. Therefore, banks must strictly control quality and safety in software development and operation management. At the same time, it is difficult to directly adopt the rapid iteration model of other industries.

The banking industry still faces problems such as an incomplete process system, limited accumulation of testing resources, and low cross-departmental collaboration efficiency in software testing practices. To some extent, these issues restrict the role of testing activities in supporting business innovation and risk control. Although mature international testing management standards provide theoretical references, banks still need to optimize and adjust them in light of their own business characteristics. To ensure that regulatory requirements are met while also enhancing test coverage and execution efficiency. Based on the analysis of the existing processes and organizational structure, this paper proposes a test resource management and process optimization method for financial scenarios. By systematically integrating test cases, test tools and risk control measures, test activities are carried out throughout all stages of software operation, achieving standardized, continuous and sustainable development of test management.

After the method design was completed, this paper took the financial industry as the analysis object and adopted a comprehensive evaluation approach to verify the optimization scheme in terms of defect discovery efficiency, execution efficiency and system stability. The results show that improving the test resource library and optimizing the management process can significantly enhance the software development and operation efficiency of banks, while ensuring business security and continuity. This research provides strong support for the rapid iteration and digital transformation of financial products. It not only offers operational references for domestic banks in their testing management practices but also provides theoretical basis and practical guidance for financial institutions in introducing standardized management for testing optimization.

2. Review of relevant literature

In the research of financial software testing, the characteristics of financial software are different from those of ordinary software. Testing not only needs to verify functions but also pay attention to profitability, system stability and performance optimization. Oyemade has proposed a multi-dimensional testing framework. Integrating broker testing, currency pair testing, spread testing, weekday and weekend testing, backtesting and live trading testing, and time and space cost testing into one process to evaluate each link of the software, the empirical results show that the framework can improve the software's profitability under different market conditions [1]. To address the issue of insufficient robustness of large language model financial software when handling natural language tasks, Xiao proposed the ABFS automated testing method, which uniformly processes input prompts and examples, and generates highly natural test cases through combined optimization and best-first search. Experiments show that this method can increase the success rate of tests, enhance the transferability of test cases, and reduce testing costs. Provide reliable evaluation before the deployment of financial software [2].

Xie has built a financial investment data management system by integrating big data and cloud computing. Through system design, development and testing, the system's functions, stability, response speed and data access efficiency have been verified, providing a reference for financial software testing and investment data management [3]. Byrne conducted research on the behavior of software testing engineers in Irish financial institutions. By using interview and self-ethnography methods, it analyzed the testing execution and use case development of engineers in different projects, experiences, and organizational environments, providing practical experience for optimizing team performance and return on investment And enrich the application of organizational behavior in financial software testing [4].

Jureczko analyzed the agile testing process of large-scale financial projects, compared the actual operation with agile principles, revealed the efficiency improvement and limitations of agile methods in financial software testing, and provided a reference for test management in the future agile environment[5]. Huda research has found that fintech can directly improve the financial behavior of small, medium and micro enterprises and play a role through financial inclusion, providing data support for the design of financial software, functional testing and the promotion of inclusive finance [6]. Through empirical research on large bank teams, Chatterjee found that AI programming tools such as GitHub Copilot can improve development efficiency, optimize code quality and enhance team satisfaction, providing a reference for financial software development and testing optimization [7].

Kuncova analyzed the bank software testing process through discrete event simulation, evaluated the required manpower and proposed improvement plans, showing that the reasonable allocation of personnel and adjustment of processes can improve testing efficiency and perfect the management of the testing asset library [8]. Overall, existing research explores financial software testing from multiple perspectives such as profit assessment, system robustness, data management, team behavior, agile processes, fintech applications, and AI-assisted development, providing theoretical and practical foundations for building a scientific, systematic, and optimizable financial software testing asset library. It also points out the research potential in the future in intelligent testing, multi-scenario adaptation and multi-dimensional performance evaluation.

3. Current Situation of Financial Software Testing and Application of TMMi Model

The software systems of commercial banks undertake the functions of supporting business processes, providing financial services and ensuring customer operations. Software testing identifies problems and assesses whether the software meets business and technical requirements by examining its performance under different conditions. With the increase in the volume of banking business and the complexity of financial product functions, it is no longer sufficient to ensure software quality merely through debugging during the development stage. Professional software testing has become an essential step, covering functional verification, logic checking, and boundary condition testing. The goal is to ensure the stable operation of the software and meet various business requirements. Testers need to prepare detailed plans, set clear goals, and design complete test cases and data to ensure that each stage of testing can proceed smoothly.

During the test execution phase, the software is operated in accordance with business processes and development logic. By comparing the actual results with the expected results, defects are identified and fed back to the development and operation and maintenance departments to support system optimization and business improvement. With the accumulation of testing experience, the bank has established a testing asset library, which organizes and preserves test cases, data, automation scripts, simulation tools and historical defects, enabling the reuse of tools, methods and experience. The testing asset library has improved testing efficiency, expanded coverage, optimized processes and accumulated experience. It ensures software quality and system stability. At the same time, it provides support for the update of financial products, digital transformation and business development, and also helps banks maintain their competitiveness in technology and services.

Financial institutions rely highly on the stability and reliability of software systems in their daily operations. Although the development stage is often emphasized, the software testing stage often lacks systematic management. The Test Maturity Model (TMMi) provides an improvement method, dividing testing activities into different stages and standardizing their management. Covering the entire process from requirement analysis, design verification, code review to functional testing, non-functional testing and organizational management, financial software is guided by the process

under the requirements of high complexity, high security and high reliability. The five maturity levels clearly define the improvement goals for each stage From the initial stage of scattered testing, the formation of testing strategies in the management stage, the coverage of the entire life cycle and non-functional reviews in the definition stage, the optimization of the process with data analysis in the measurement stage to the realization of defect prevention and continuous improvement in the optimization stage, it provides practical and theoretical support for financial institutions to establish a stable, efficient and reusable software testing system.

In the financial business environment, the software testing asset library is built by the testing management department. By integrating various data such as test cases, historical defect records, automated scripts, and environment configurations, it forms a complete, traceable, and reusable resource, providing support for test design, execution, and evaluation. Asset library managers conduct quantitative analysis of test effects based on indicators and evaluation methods Use the results to guide the continuous optimization of the asset library, enabling it to support software quality management, business continuity and system stability in high-risk and high-pressure financial scenarios. At the same time, it improves testing efficiency, shortens the defect discovery cycle, reduces repair costs, and provides a foundation for automated testing, intelligent analysis and the application of innovative methods. Enable financial institutions to maintain the simultaneous improvement of software quality and business stability in a rapidly changing market environment.

4. Analysis of Challenges and Problems Faced by Financial Software Testing

4.1 Complexity of Financial Business and Test Coverage Requirements

As the financial industry moves towards digitalization and intelligence, traditional banks are confronted with high demands from customers for convenient operation, service efficiency and methods of obtaining funds. While ensuring the stable operation of core business, bank information systems also need to support new business functions and innovative services, and guarantee the normal operation of transactions, account management, payment settlement and self-service in complex environments. Software testing is responsible for verifying the integrity of business functions, maintaining system reliability, and managing cross-system data interaction during this process. The testing team assesses the risks that system changes may bring by covering functional modules and business processes, and tracks the impact of historical business changes. At the same time, a reusable and traceable test asset library is established by using test scripts, historical cases, defect records and business rules. The asset library supports the verification work of different business modules and the optimization of test resources, providing standard references for system updates and iterations. It also needs to meet regulatory requirements such as identity authentication, fund security, anti-money laundering, anti-fraud, digital currency promotion and the application of domestic technologies. Ensure the system is safe and reliable for a vast number of customers and diverse usage scenarios across the country, providing a foundation for the efficient operation of banking business and the implementation of intelligent testing.

The software products of large commercial banks, from proposal to development, testing and operation, are handled by different departments respectively. Each department manages business requirements, process optimization, function realization, system testing and maintenance. This division of labor leads to obstacles in cross-departmental communication, affecting the response speed of software development and testing. Banking business seems simple However, each transaction involves multi-level interactions among the front-end system, core accounting system, clearing system, reporting system, payment system and electronic voucher system. Changes in a single link may affect the entire business chain. Testing needs to cover the entire process from customer operation, account accounting, transaction execution, report generation to electronic

voucher storage to ensure data accuracy and business consistency. The current testing process is carried out in a phased sequence. After the business department completes the requirements document, the development department implements the functions. The testing department designs and executes test cases and feeds back defects. After the business and development departments fix the problems, the testing department conducts regression verification and confirms the system reliability. Eventually, the software is put into production and enters the operation monitoring stage. A software testing asset library for financial scenarios should integrate historical test cases, business rules, system dependencies, defect records, and customer operation scenarios to form a reusable and scalable asset system, thereby enhancing testing efficiency, reducing risks, and ensuring the safe operation of complex financial businesses across different customers and multiple systems.

4.2 Bank Testing Process Efficiency and Asset Library Optimization

In large commercial banks, the efficiency of software testing directly affects the time it takes for financial products to go from development to use. Currently, the process is carried out in the order of requirement writing, function development, and test execution. The business requirements department, development department, and testing department are independent of each other and use the waterfall V model. On average, each test project takes about fifty days. It usually takes three months from the proposal of requirements to the production of the product. The time-consuming aspects of testing mainly focus on handling critical process defects, preparing test data, and integrating with third-party systems. Critical process defects can lead to test suspensions, and issues with requirement documents or business processes require rework and re-testing. The preparation for complex business tests consumes a significant amount of time. For instance, opening a bank card requires completing identity verification, information entry, image collection, and system writing. Every day, a large number of repetitive tasks need to be handled. The banking business also relies on the collaboration of external institutions. The fund, wealth management, digital RMB and social security business must be integrated with external systems for debugging. The debugging time is uncontrollable and the testing cycle is prolonged. Building a software testing asset library for financial scenarios can integrate historical cases, business rules, system dependencies and defect information to form reusable resources. Improve testing efficiency and ensure business stability in a multi-system and multi-customer environment.

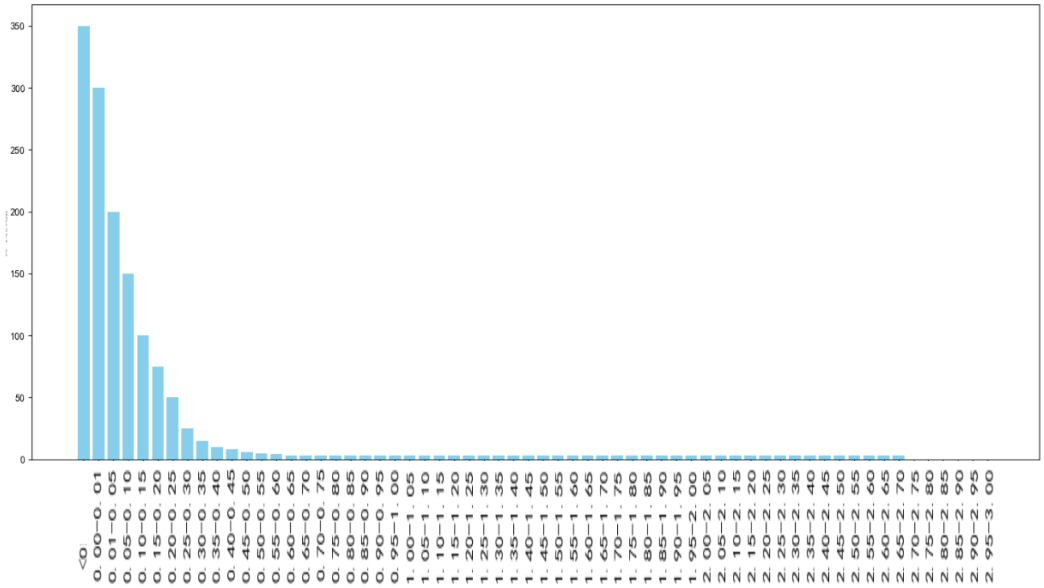

Figure 1 Comparison of the average daily number of defects

In the software testing process of large commercial banks, the efficiency and quality of testing directly affect the stability of the system and the speed of launching new products. The bank assigns the tasks of requirement writing, function development, test execution, defect repair and system launch to different departments, each of which is independently responsible. Although this division of labor is clear, it prolongs the cycle from the completion of development to the production of the product. Take a certain bank as an example. Among the numerous testing projects it undertakes throughout the year, it usually takes several weeks or even months for a single project to go from development and delivery to formal use. The number of defects found in daily tests is relatively small and scattered. Only a few defects are identified in every hundred test cases. This trend is reflected in Figures 1 and 2. The main reasons for the long testing cycle lie in the handling of key process defects, the preparation of testing materials, and the integration and debugging with third-party systems. When defects occur in key processes, whether it is code issues or business process problems, the entire test may be forced to be suspended and restarted, and the preparation of test materials also consumes a great deal of manpower and time. For instance, opening a bank card involves multiple steps such as customer identity information entry, image collection, online verification, information supplementation and card writing. Complex businesses such as anti-fraud model testing also require the deployment of a large amount of transaction data according to trigger conditions.

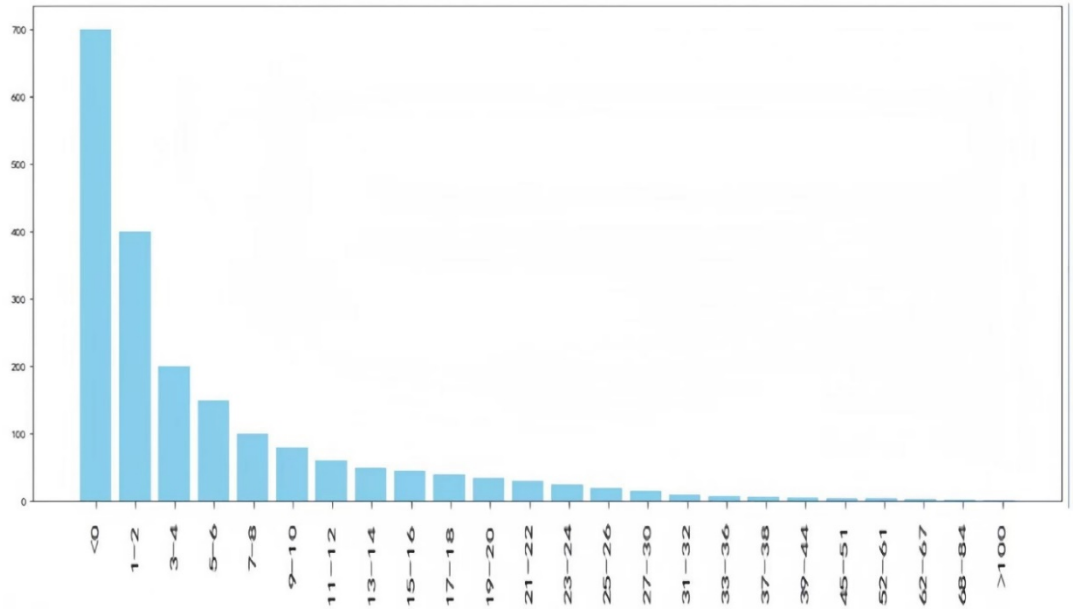

Figure 2 Number of defects in cases

Banking services must also undergo joint debugging with external institutions. The timing of joint debugging is affected by external arrangements, making the testing cycle unpredictable. In terms of test quality, since the test scenarios cannot cover all the actual usage situations, some defects will flow into the production environment. However, regression testing usually only covers the current modification or historical high-risk modules and cannot guarantee complete coverage. The differences in the capabilities of testers can also affect the efficiency of defect discovery. Experienced personnel are more likely to identify key issues, while those with less experience tend to overlook them. To address these issues, establishing a software testing asset library for financial scenarios can integrate historical defect records, business rules, test cases, and system dependencies, providing references and guidance for testing. This can optimize processes, shorten cycles, and enhance the efficiency of defect discovery, ensuring business stability and security in multi-system,

multi-business, and cross-institutional environments.

In the IT departments of large commercial banks, software development and testing are mostly undertaken by external professional companies. The frequent turnover of these personnel can easily affect the continuity and efficiency of testing projects. As the most fundamental asset, test cases can retain the achievements of previous work, enabling newly recruited or replaced testers to quickly start working during the iteration of business systems. This ensures the stable progress of the testing activities and the planned launch of the software products. With the development of the financial market and the evolution of fintech, traditional testing methods are difficult to cope with complex business scenarios. A single testing case cannot meet the actual needs of bank software testing. It is necessary to supplement and improve the existing testing assets and gradually establish a comprehensive testing asset library for financial scenarios to enhance the level of testing management.

In financial software testing, data initialization, bank card opening, and various interface calls are repetitive processes and fixed operations. Through automated scripts, data preparation and functional verification can be quickly completed, significantly enhancing efficiency and freeing up testing human resources. Banking business involves cross-system operations and high-frequency trading. By simulating system programs, functions can be verified and abnormal situations can be covered without relying on external institutions. At the same time, historical operation and maintenance issues can be formed into a risk database for testing reference to avoid the recurrence of defects. Visualizing and organizing business processes and associating them with the testing platform can help new testers quickly master the operation steps and branch nodes, improve testing efficiency and optimize the performance of testing projects. The comprehensive testing asset library can provide continuous support for bank software testing, ensuring that the launch of new functions does not affect existing business, and enhancing the overall business quality and testing management level.

5. Design and evaluation methods for test improvement plans

In the testing work of financial systems, centralized testing often fails to ensure the stability and delivery quality of the system. Problems tend to accumulate during the design and development phases, which will increase the cost of later fixes and prolong the project schedule. If testing can be intervened at an early stage, potential risks can be identified during the requirements and design phases, and quality can be controlled through interface checks and code verification during the development phase, allowing problems to be resolved in the early stage. The testing work also needs to be extended to after the system goes online. The operation logs and abnormal data can provide a basis for optimization. By analyzing these data, testers can expand the scope of their tests and make the test content more in line with the actual business situation. To enhance the effectiveness of testing, financial institutions need to establish a unified testing asset system, centrally manage and continuously update scripts, data and cases, thereby covering the entire process of system design, development and operation.

The stability and business continuity of bank software systems in financial scenarios highly rely on scientific testing management systems and optimized process design. This study aims to quantify the effect of bank testing process improvement by establishing a systematic comprehensive evaluation method, thereby providing an accurate basis for the construction and optimization of the software testing asset library (Formulas 1 and 2). This study selects the TOPSIS method as the assessment tool. By mapping the bank's test data over many years to the ideal solution and the negative ideal solution and calculating the relative closeness, a comprehensive score is generated. This method can comprehensively reflect the performance changes of the test process in terms of

efficiency and quality (Formula 3).

$$b_{ij} = a_j^{max} - a_j^{min} a_{ij} - a_j^{min} \quad (1)$$

$$b_{ij} = a_j^{max} - a_j^{min} a_j^{max} - a_{ij} \quad (2)$$

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^{m} p_{ij} \ln p_{ij}, p_{ij} = \frac{b_{ij}}{\sum_{i=1}^{m} b_{ij}} \quad (3)$$

During the evaluation process, this study selected several core indicators to reflect test efficiency, workload and quality level, including the number of test items, the input of test personnel, the average duration of individual tests, the average number of functional points, the average number of cases, the average number of defects and the defect situation in the production environment. These indicators can present the implementation of testing activities and defect discovery capabilities from multiple perspectives, providing a data basis for overall assessment. Due to the differences in measurement units, numerical ranges and data properties among various indicators, the research has unified their processing. For positive indicators such as the average number of defects, mapping methods have been adopted to adjust them to a unified range, while for negative indicators such as test duration and personnel input, reverse mapping has been carried out. Thus, different indicators can be directly compared and analyzed on the same scale.

The research further combines the entropy weight method to calculate the weights of each index. Through the entropy value of the index, the degree of variation of the index is obtained, and the contribution ratio of each index in the comprehensive score is determined by the degree of variation, ensuring that the comprehensive score can reflect the actual impact of different indicators on the test improvement effect. The calculation of the comprehensive score is obtained by calculating the Euclidean distance between the weighted standardized index matrix and the positive and negative ideal solutions. The analysis results in Figure 3 show that the comprehensive score remained relatively stable from the two years before the improvement to the year of improvement, rose significantly in the first year after the improvement and continued to increase in the second year. In the initial stage of test improvement, additional human resources were needed to build mechanisms, platforms and tools, which led to a slight increase in test costs that year. However, as the improvement measures were gradually implemented, both test efficiency and quality significantly improved. These data provide a reliable basis for the optimization and construction of the bank's software testing asset library.

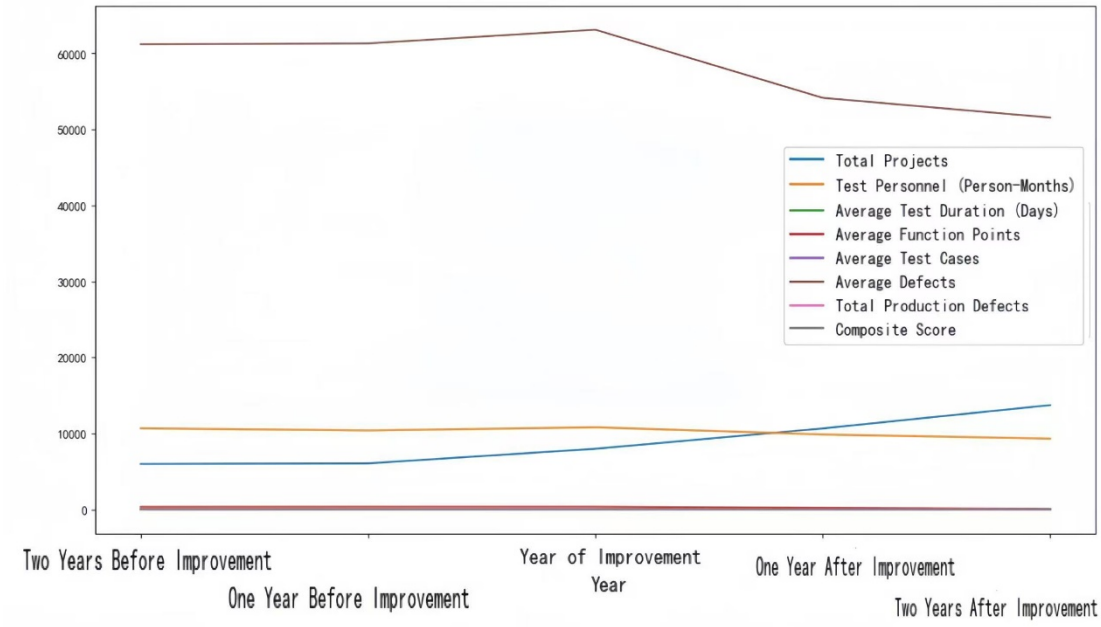

Figure 3 shows the main test indicators and the changes in the comprehensive score

6. Conclusions and Prospects

In the current rapidly changing financial market environment, the business models and service methods of banks are constantly being adjusted. New technologies are gradually being integrated into financial services. Software testing plays an important role in ensuring business stability, improving efficiency and optimizing customer experience. This paper proposes a systematic method for optimizing the software testing process. Through a comprehensive evaluation model, it quantitatively analyzes the improvement effects, providing a scientific basis for building a usable testing asset library. The research takes the financial industry as the object, combines automated scripts and testing tools to optimize the testing process. The evaluation results show that the comprehensive scores of each year after improvement have shown a continuous upward trend, and the testing efficiency and defect discovery ability have been significantly enhanced. The comprehensive evaluation model determines the weights of each indicator by calculating data variability, thereby achieving an objective evaluation of multi-dimensional test data and providing a quantitative reference for the management of the test asset library. This method has a limited sample size, but it also provides other financial institutions with a referenceable optimization solution. Future research can increase horizontal comparison data, introduce more automated means, expand the functions of the asset library, continuously improve the software testing process and the construction of the asset library, and provide long-term support for the stable operation and service optimization of the financial system.

References

[1] *Oyemade D A. Software Testing Framework for the Financial Market[J]. American Journal of Software Engineering and Applications, 2024, 12(1): 36-43.*

[2] *Huda N, Risman A. The Behavioral Finance of MSMEs: Financial Inclusion and Financial Technology (Case Study on MSMEs in West Jakarta)[J]. 2024.*

[3] *Liu Y. Blockchain Future in Cloud Computing: The Challenges to Implement Blockchain Technology in Cloud Computing[J]. Journal of Computer, Signal, and System Research, 2025, 2(5): 15-23.*

[4] *Chen M. Research on the Application of Privacy-enhancing Technologies in AI-driven Automated Risk Detection Systems[J]. Advances in Computer and Communication, 2025, 6(4).*

[5] *Chen X. Research on AI-Based Multilingual Natural Language Processing Technology and Intelligent Voice Interaction System[J]. European Journal of AI, Computing & Informatics, 2025, 1(3): 47-53.*

[6] *Zhou Y. Cost Control and Stability Improvement in Enterprise Level Infrastructure Optimization[J]. European Journal of Business, Economics & Management, 2025, 1(4): 70-76.*

[7] *Tang X, Wu X, Bao W. Intelligent Prediction-Inventory-Scheduling Closed-Loop Nearshore Supply Chain Decision System[J]. Advances in Management and Intelligent Technologies, 2025, 1(4).*

[8] *Li, W. (2025). Research on Optimization of M&A Financial Due Diligence Process Based on Data Analysis. Journal of Computer, Signal, and System Research, 2(5), 115-121.*

[9] *Huang, J. (2025). Balance Model of Resource Management and Customer Service Availability in Cloud Computing Platform. Economics and Management Innovation, 2(4), 39-45.*

[10] *Liu, Y. (2025). The Importance of Cross-Departmental Collaboration Driven by Technology in the Compliance of Financial Institutions. Economics and Management Innovation, 2(5), 15-21.*

[11] *Chang, Chen-Wei. "AI-Driven Privacy Audit Automation and Data Provenance Tracking in Large-Scale Systems." (2025).*

[12] *Zhang K. Research on the Application of Homomorphic Encryption-Based Machine Learning Privacy Protection Technology in Precision Marketing[C]//2025 3rd International Conference on Data Science and Network Security (ICDSNS). IEEE, 2025: 1-6.*

[13] *Zhang, Xuanrui. "Automobile Finance Credit Fraud Risk Early Warning System based on Louvain Algorithm and XGBoost Model." In 2025 3rd International Conference on Data Science and Information System (ICDSIS), pp. 1-7. IEEE, 2025.*

[14] *Wu X, Bao W. Research on the Design of a Blockchain Logistics Information Platform Based on Reputation Proof Consensus Algorithm[J]. Procedia Computer Science, 2025, 262: 973-981.*

[15] *Su H, Luo W, Mehdad Y, et al. Llm-friendly knowledge representation for customer support[C]//Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025: 496-504.*

[16] *Z Zhong. AI-Assisted Workflow Optimization and Automation in the Compliance Technology Field [J]. International Journal of Advanced Computer Science and Applications (IJACSA), 2025, 16(10): 1-5.*

[17] *Wu Y. Software Engineering Practice of Microservice Architecture in Full Stack Development: From Architecture Design to Performance Optimization[J]. 2025.*

[18] *Sun Q. Research on Accuracy Improvement of Text Generation Algorithms in Intelligent Transcription Systems[J]. Advances in Computer and Communication, 2025, 6(4).*

[19] *Liu X. Emotional Analysis and Strategy Optimization of Live Streaming E-Commerce Users Under the Framework of Causal Inference[J]. Economics and Management Innovation, 2025, 2(6): 1-8.*

[20] *Yuan S. Application of Network Security Vulnerability Detection and Repair Process Optimization in Software Development[J]. European Journal of AI, Computing & Informatics, 2025, 1(3): 93-101.*