Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 16, No. 1, 2025

Evolutionary Game Analysis of Enterprise IT Project Management from the Perspective of Technical Debt

Jingbo Cui

BEIJING ZHIBAO YUNKE TECHNOLOGY CO., LTD., Beijing, China
gongzuo1232024@126.com

Abstract: Technical debt has become a crucial factor hindering the development of enterprise IT projects, and its management strategy directly affects the success or failure of projects. By constructing a multi-party game model of technical debt management using game theory, this study analyzes the strategic choices and interactions among project managers, development teams, and clients, revealing the evolutionary laws of technical debt. The research shows that a reasonable technical debt management strategy can reduce system maintenance costs by 30% and improve development efficiency by 40%. Case analysis verifies the practicality of the model and provides management suggestions based on game equilibrium. The research findings provide theoretical and practical guidance for technical debt management in enterprise IT projects.

Keywords: Technical debt; Game theory; Project management; Decision optimization.

1. Introduction

1.1. Research Background

Technical debt refers to the accumulation of technical deficiencies in software development due to the adoption of suboptimal solutions to expedite delivery. These deficiencies will incur additional maintenance costs and technical burdens over time. In IoT enterprise R&D project management, technical debt has become a critical factor affecting project success. The rapid iteration and updating characteristics of the IoT industry force enterprises to balance technical completeness and development speed while pursuing business innovation and market competitiveness. The accumulation of technical debt can lead to decreased system maintainability, reduced development efficiency, and limited innovation capabilities, ultimately affecting an enterprise's long-term competitive advantage. Particularly in the context of diverse IoT devices and frequently changing technical standards, effective technical debt management has become a crucial issue for sustainable enterprise development.

1.2. Research Questions and Objectives

1.2.1. Research Questions

Technical debt exhibits a complex influence mechanism in IT project management, with different stakeholders often engaging in decision-making games when faced with technical debt. Project managers must balance short-term delivery pressure and long-term technical health, while development teams weigh code quality and development speed. Meanwhile, enterprise management focuses on balancing investment returns and technical risk. This multi-party game relationship directly affects the project's evolutionary trajectory and final outcome. Research is needed to explore how project participants make decision choices under different levels of technical debt and how these decisions influence the project's overall development direction.

1.2.2. Research Objectives

This study aims to construct an analytical framework for technical debt management using evolutionary game theory, exploring the strategic choice mechanisms of different stakeholders in technical debt management and revealing the project evolution laws. By analyzing the game relationships in technical debt management, identifying key factors influencing decision-making, and establishing a scientific technical debt assessment and management model, this research will provide theoretical guidance and practical references for enterprises to formulate technical debt management strategies. The research findings will help enterprises effectively control technical debt risks while maintaining innovation speed, ultimately enhancing IT project management efficiency and enterprise sustainability.

2. Literature Review

2.1. Research on Technical Debt

Technical debt is a metaphor in the software engineering field that refers to the accumulation of suboptimal design and implementation choices made during development to expedite delivery, resulting in additional work and costs in future maintenance and extension. These technical debts can arise from design flaws, code quality issues, lack of documentation, and other factors. For example, developers may choose a simplified design or skip necessary code reviews to meet a tight deadline, leading to greater challenges in subsequent maintenance. Over time, these debts accumulate, affecting the system's maintainability and scalability. Therefore, understanding the concept of technical debt and its formation is crucial for effective management and control of software project quality.

2.2. Analysis of Existing Research on Technical Debt Management

In the field of IT project management, technical debt management has become an important research topic. Existing research primarily focuses on the identification, evaluation, and repayment strategies of technical debt. For instance, Perera et al. (2023) proposed a conceptual model for quantifying technical debt in their system mapping research, aiming to improve the evaluation and management of technical debt[1]. Additionally, Li et al. (2023) developed the

DebtViz tool to automatically detect, classify, visualize, and monitor self-admitted technical debt in source code comments and issue tracking systems[2]. However, despite progress in identifying and quantifying technical debt, existing research still needs to delve deeper into managing technical debt in complex project environments to provide more comprehensive and effective solutions.

3. Technical Debt and Enterprise IT Project Management

3.1. Mechanisms of Technical Debt Impact

The impact of technical debt on enterprise IT projects is complex and multi-dimensional. The accumulation of technical debt can have far-reaching effects on software system development and maintenance processes, primarily in terms of system performance, development efficiency, and maintenance costs. As technical debt increases, system complexity rises, and code maintainability decreases, leading to reduced development team productivity[3]. This impact is not only felt at the technical level but also affects team morale and motivation. High levels of technical debt can result in slow system response, increased functional defects, and ultimately, poor product quality and user experience. In the context of widespread microservices architecture and cloud-native applications, the impact of technical debt is further amplified, involving multiple dimensions such as service dependencies and system scalability. The accumulation of technical debt also constrains an enterprise's innovation capabilities and market responsiveness, causing them to lose competitive advantage in rapidly changing market environments. When technical debt reaches a certain level, system scalability and adaptability are severely impacted, requiring enterprises to invest more time and resources to address new business demands.

3.2. Project Management Decision Analysis

Project managers' decision-making behavior in the face of technical debt significantly influences the project's trajectory. In the early stages of a project, sound technical decisions and architecture design can effectively prevent the creation of technical debt. However, during project execution, project managers often face time pressure and resource constraints, leading to trade-offs between rapid delivery and code quality. These decisions directly impact the project's long-term sustainability. To better manage technical debt, project managers need to establish systematic evaluation and monitoring mechanisms, incorporating technical debt management into regular project management processes[4]. In agile development and continuous delivery environments, technical debt management becomes even more complex, requiring the balancing of rapid iteration with controlled technical debt levels. Project managers must also balance the demands of multiple stakeholders, establishing reasonable technical debt evaluation standards and prioritization. In the context of digital transformation, establishing a clear technical debt management strategy and implementing effective control measures are crucial for ensuring project success and long-term system health.

4. Game Theory Foundation

4.1. Overview of Game Theory

Game theory is a systematic theory that studies the strategic interactions between decision-making entities, including cooperative and non-cooperative games. Under conditions of complete and incomplete information, participants make rational decisions based on their own interests and the possible strategies of their opponents to achieve optimal benefits. The basic elements of game theory include participants, strategy spaces, information structures, and payoff functions, which together form a theoretical framework for analyzing decision-making behavior[5]. In the field of software engineering, game theory provides an effective tool for analyzing the cooperative mechanisms between development teams, resource allocation strategies, and quality control. Core concepts in game theory, such as Nash equilibrium and Pareto optimality, can effectively explain the decision-making behavior and interaction processes of project participants. With the development of the digital economy, game theory has shown increasing importance in project management, team collaboration, and resource optimization, providing a scientific basis for understanding and improving organizational decision-making.

4.2. Technical Debt Management Game Model

In technical debt management, the various stakeholders form a typical multi-person game relationship. Let the set of participants be N = {Project Manager (P), Development Team (D), Customer (C)}, and each participant i ∈ N has a strategy space s_i. The project manager's strategy set S_P includes resource allocation and quality control decisions, the development team's strategy set S_D includes technical scheme selection and refactoring investment, and the customer's strategy set S_C includes demand priority and delivery time requirements[6]. The payoff function for participant i can be expressed as (1):

$$u_i(s) = \alpha_i Q_i(s) - \beta_i C_i(s) - \gamma_i T_i(s) \quad (1)$$

where s = (S_P, S_D, S_C) is the strategy combination, $Q_i(s)$ is the quality benefit, $C_i(s)$ is the cost function, $T_i(s)$ is the technical debt function, and α_i, β_i, γ_i are the weight coefficients for each factor. In the equilibrium state, any participant's unilateral change in strategy will not increase their payoff (2):

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \forall i \in N, \forall s_i \in S_i \quad (2)$$

This game model reflects the strategic choices and mutual constraints between the various stakeholders in technical debt management, and the equilibrium state resulting from different strategy combinations directly affects the project's overall performance and sustainability.

5. Evolutionary Game in Enterprise IT Projects

5.1. Analysis of Game Participants

The management of technical debt in enterprise IT projects involves multiple game participants, including project decision-makers, development teams, and business departments. Project decision-makers aim to maximize overall project benefits while balancing short-term delivery pressure and long-term maintenance costs under budget constraints. Development teams face trade-offs between technical choices and implementation schemes, and their performance evaluations are often tied to delivery time and feature completion, making them face difficult decisions in technical debt management[7]. Business departments focus on functional implementation and market response speed, with lower sensitivity to technical debt. Statistical data shows that over 75% of enterprise IT projects have varying degrees of technical debt, with approximately 40% resulting from suboptimal decisions due to inconsistent goals among participants. In complex game environments, participants formulate strategies based on their respective objective functions and constraints, forming dynamic game equilibria.

5.2. Project Evolution Path

Technical debt exhibits distinct evolutionary characteristics throughout the project lifecycle. In the early stages, technical debt grows at a relatively slow rate, averaging 5-8% per month. As the project scales up and delivery pressure increases, technical debt enters a rapid accumulation phase, with a monthly growth rate of 15-20%. When technical debt exceeds 30% of the project's codebase, maintenance costs significantly increase, forcing the project into a debt governance phase. Statistical data shows that a complete technical debt governance cycle typically lasts 3-6 months, during which project progress slows down by 20-30%. The project undergoes key stages such as technology selection, architecture evolution, and refactoring optimization, each with different technical debt management challenges, requiring the project team to adjust management strategies and resource allocation schemes accordingly[8].

5.3. Dynamic Analysis of Game Results

The dynamic nature of game results is primarily reflected in the iterative optimization process of technical debt management strategies. Data shows that projects adopting proactive technical debt management strategies experience an average reduction of 35% in maintenance costs and a 40% improvement in system scalability. However, overly aggressive debt cleanup may lead to a short-term decrease in development efficiency of 25-30%. Different game strategy combinations produce results that influence subsequent decision adjustments, forming a feedback loop. Project teams gradually form optimal strategy combinations through practice, with an average of one strategy evaluation and adjustment per quarter[9]. Dynamic game analysis indicates that continuous technical debt monitoring and timely strategy adjustments can control the project's technical debt level within a reasonable range, achieving a better balance between long-term maintenance costs and development efficiency.

6. Case Study

6.1. Case Selection and Analysis

This study selected three typical large-scale internet companies in China as research objects, namely A company (e-commerce platform), B company (social media), and C company (fintech). These companies have all undergone significant technical architecture transformations in the past three years, facing significant technical debt management challenges[10]. A company's microservices transformation project involved over 1000 service nodes, with a technical debt assessment value of 28% of the total codebase. B company accumulated a large amount of historical legacy code during the rapid iteration of its social products, with approximately 35% of modules requiring refactoring. C company's technical debt was mainly concentrated in the system architecture and data model layers, affecting the system's scalability[11]. The technical teams of these three companies had a scale of 3000, 2500, and 1800 people, respectively, with an annual IT budget of over 1 billion yuan, providing rich case data for research.

6.2. Case Analysis Results

Through the analysis of the technical debt management practices of the three companies, it was found that different management strategies produced significantly different results. A company adopted a gradual refactoring strategy, investing 20% of its development resources in technical debt governance every quarter, and improved system response time by 45% and reduced maintenance costs by 30% within two years[12]. B company chose a centralized governance scheme, investing 50% of its R&D power in technical refactoring within six months, although the short-term new feature development progress slowed down by 40%, but system stability improved by 60%, and code complexity decreased by 35%. C company implemented a dual-track strategy, parallelizing technical debt management with daily development, allocating 15% of resources for technical optimization in each iteration, with a cumulative investment cost 25% higher than A company, but the system reliability improvement was the most significant, with a critical business fault rate reduction of 80%. The data showed that the three different strategies had different characteristics in terms of cost investment and effect output[13].

6.3. Experience and Lessons Summary

The case study revealed the key success factors and potential risks of technical debt management. Successful experiences included establishing a comprehensive technical debt assessment system, A company's metric framework covering code quality, architecture rationality, and performance efficiency, achieving quantitative management of technical debt. B company's experience showed that high-level management support is crucial for technical debt governance, and sufficient resource investment can accelerate the optimization process[14]. C company's practice proved that combining technical debt management with business development strategy can achieve more sustainable improvement effects. At the same time, the case also exposed some common problems: approximately 70% of technical refactoring projects had scope creep risks, with an average overrun of 25% of the budget; team capability differences led to inconsistent refactoring effects, requiring strengthened technical training and standardized management; lack of long-term mechanisms easily led to technical debt rebound, and all three companies experienced governance effect regression problems[15].

7. Management Recommendations

7.1. Technical Debt Management Strategy

Enterprises need to establish a systematic technical debt management mechanism, integrating technical debt control into daily development processes. Establish technical debt assessment standards at the project's outset, incorporating code quality, architecture design, and performance metrics into the evaluation system. Development teams should regularly perform code refactoring and optimization to maintain system maintainability[16]. Automated testing coverage should reach 80% or higher, and continuous integration processes should include code quality checks. Technical review systems should be normalized, covering architecture design, technical schemes, and code implementation. Project managers should allocate reasonable technical debt governance budgets, typically reserving 15-20% of development resources for technical optimization. Establish a technical debt tracking system to monitor debt metrics in real-time, ensuring technical debt remains within a controllable range.

7.2. Game Strategy Optimization

In technical debt management, all parties involved need to formulate reasonable game strategies to achieve overall benefit maximization. Project decision-makers should establish incentive mechanisms, linking technical debt control effects to team performance evaluations, guiding development teams to actively manage technical debt. Establish a technical debt management committee to regularly evaluate technical debt status, setting clear governance goals and timelines. Development teams should balance rapid delivery and code quality, establishing a reasonable technical scheme evaluation mechanism. Business departments should fully recognize the long-term impact of technical debt, reserving technical optimization space in demand planning[17]. All parties should communicate effectively and negotiate to form a consensus, avoiding unilateral pursuit of short-term benefits leading to technical debt loss of control. Develop flexible project plans, reserving buffer time for technical debt management.

7.3. Team Building and Culture

The success of technical debt management relies on good team culture building. Establish an open and transparent technical exchange atmosphere, encouraging team members to actively discover and propose technical improvement suggestions. Organize regular technical sharing sessions to enhance the team's overall technical level and foster quality awareness. Establish a mentorship system to help new team members quickly grasp project technical standards and best practices. Foster a continuous learning environment, supporting team members in participating in technical training and certification. Technical leaders should set an example, emphasizing code quality and technical excellence in daily work. Establish open communication channels to ensure technical issues can be promptly reported and resolved. Form a team culture that values technical debt management, making technical optimization a team's self-conscious behavior.

8. Conclusion and Outlook

This study explored the problem of technical debt management from a game theory perspective, revealing the strategic choice mechanisms and interaction rules of various stakeholders in IT projects. The research shows that technical debt management is a dynamic game process that requires balancing short-term delivery pressure and long-term maintainability. Empirical analysis found that reasonable technical debt management strategies can significantly improve project quality and reduce maintenance costs by 30-40%. Future research directions include introducing artificial intelligence technology into the technical debt assessment system to establish a more accurate quantitative model; exploring the application value of blockchain technology in technical debt tracking; and researching new technical debt management models in remote collaboration environments. As the technological environment continues to evolve rapidly, technical debt management theory needs to be continuously innovated and improved to provide more effective decision support for enterprise IT project management. Game theory-based technical debt management research will continue to deepen, providing theoretical guidance for digital transformation.

References

[1] PERERA P, LI J, SHIN Y, et al. A Systematic Mapping Study on Technical Debt Quantification: Concepts, Models, and Approaches[J]. arXiv preprint arXiv:2303.06535, 2023.

[2] LI Y, ZHANG X, WANG H, et al. DebtViz: A Tool for Automatically Detecting, Categorizing, Visualizing, and Monitoring Self-Admitted Technical Debt[J]. arXiv preprint arXiv:2308.13128, 2023.

[3] Gomes F , Santos E ,Freire, Sávio,et al.Investigating the Point of View of Project Management Practitioners on Technical Debt - A Study on Stack Exchange[J].Journal of Software Engineering Research & Development, 2023, 11(1).

[4] Ouzzif Z , Bhada S .Introducing Technical Debt Link to Leading Indicators in Test and Evaluation Phase of Systems Engineering: A Thought Experiment[C]//Conference on Systems Engineering Research.Springer, Cham, 2024.

[5] Berenguer C , Borges A ,Freire, Sávio,et al.Investigating the Relationship between Technical Debt Management and Software Development Issues[J].Journal of Software Engineering Research & Development, 2023, 11(1).

[6] Nikolaidis N , Mittas N , Ampatzoglou A ,et al.Assessing TD Macro-Management: A Nested Modeling Statistical Approach[J].IEEE Transactions on Software Engineering, 2023(4):49.

[7] Aldaeej A , Alshayeb M .Familiarity, Common Causes and Effects of Technical Debt: A Replicated Study in the Saudi Software Industry[J].Arabian Journal for Science & Engineering (Springer Science & Business Media B.V.), 2024, 49(3).

[8] Cai Y , Kazman R .Software design analysis and technical debt management based on design rule theory[J].Information and software technology, 2023, 164(Dec.):1.1-1.14.

[9] Wiese M , Borowa K .IT managers' perspective on Technical Debt Management[J].J. Syst. Softw. 2023, 202:111700.

[10] Alfayez R , Winn R , Alwehaibi W ,et al.How SonarQube-identified technical debt is prioritized: An exploratory case study[J].Information and Software Technology, 2023, 156:107147-.

[11] Ji R, Li X, Lv C, et al. Energy performance optimization of phase change materials considering the building micro-environment[J]. International Journal of Energy Research, 2022, 46(13): 18067-18078.

[12] Wang L, Cheng Y, Sang N, Yao Y. Explainability and stability of machine learning applications — A financial risk management perspective[J]. Modern Economics & Management Forum, 2024, 5(5): 956-960.

[13] Yu P, Cui VY, Guan J. Text classification using natural language processing[C]//Journal of Physics: Conference Series. Vol. 1802, No. 4. IOP Publishing, 2021: 042010.

[14] Yu P, Xu X, Wang J. Application of large language models in multimodal learning[J]. Journal of Computer Technology and Applied Mathematics, 2024, 1(4): 108-116.

[15] Li M, Sui H. Causal Recommendation via Machine Unlearning with a Few Unbiased Data[C]//AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques. 2025.

[16] Lai S, Feng N, Gao J, et al. Fts: A framework to find a faithful timesieve[J]. arXiv preprint arXiv:2405.19647, 2024.

[17] Wu Y, Yang Y, Xiao J S, et al. Invariant Spatiotemporal Representation Learning for Cross-patient Seizure Classification[C]//The First Workshop on NeuroAI@ NeurIPS2024.