Playing the Project: Elevating Software Development Team Performance through Gamification

Nurfauza Jali
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak, Malaysia
jnurfauza@unimas.my

Cheah Wai Shiang
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak, Malaysia
wscheah@unimas.my

Suriati Khartini Jali
Institute For Tourism Research And Innovation (ITRI)
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak, Malaysia
jskhartini@unimas.my

Syahrul Nizam Junaini
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak, Malaysia
syahruln@unimas.my

Tan Ping Ping
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Kota Samarahan, Sarawak, Malaysia
pptan@unimas.my

Abstract

Gamification has emerged as a promising approach to improving collaboration, accountability, and engagement in team-based software development education. This paper presents the Team Performance Gamification Measurement (TPGM) framework, designed to address persistent challenges such as free-riding, unequal task distribution, and limited visibility of individual contributions in academic software projects. The framework integrates role-specific performance metrics, self- and peer-evaluation mechanisms, gamification elements (points, badges, and leaderboards), and real-time performance dashboards within an Agile-oriented project workflow. An empirical evaluation was conducted with 201 undergraduate students enrolled in a Software Engineering Laboratory course. Baseline findings indicate that 54.23% of participants experienced free-riding at least sometimes prior to intervention. Post-intervention results show strong perceived impact, with 81.69% of participants reporting that gamification enhanced accountability and reduced free-riding behaviour. System usage verification and stakeholder feedback further indicate improved transparency, engagement, and fairness in assessment. The study demonstrates that embedding gamification directly into project governance and evaluation processes, rather than applying it as a superficial engagement layer, can effectively mitigate free-riding as a process-level issue. The TPGM framework is supported by validated system modules and functional test cases, providing a replicable and deployment-ready model for software engineering education and related collaborative learning contexts.

ICSCA 2026, Langkawi, Malaysia

ACM ISBN 979-8-4007-2291-2/2026/02
https://doi.org/10.1145/3801619.3801667

CCS Concepts

• **Software and its engineering**; • **software creation and management**; • **Collaboration in software development**; • **gamification, team performance, free-riding, peer assessment, educational technology**;

ACM Reference Format:
Nurfauza Jali, Cheah Wai Shiang, Suriati Khartini Jali, Syahrul Nizam Junaini, and Tan Ping Ping. 2026. Playing the Project: Elevating Software Development Team Performance through Gamification. In *2026 15th International Conference on Software and Computer Applications (ICSCA 2026), February 03–05, 2026, Langkawi, Malaysia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3801619.3801667

1 Introduction

Software development is a complex process that requires strong collaboration, effective communication, and creative problem-solving among team members. In Software Engineering (SE) education, these competencies are commonly developed through team-based laboratory and project-oriented courses that adopt Agile and Scrum practices. Despite the collaborative nature of such learning environments, challenges such as inefficiencies, low engagement, and unequal participation frequently arise. One of the most persistent issues is free-riding, where some team members contribute significantly less than others while benefiting equally from group outcomes. This behaviour not only affects the quality of the final project deliverables but also diminishes team morale and leads to inequitable assessment outcomes.

To mitigate these challenges, prior studies have explored various strategies, including task delegation, structured team management, and peer assessment. While these approaches are conceptually effective, their practical impact often depends on consistent implementation, transparency of contribution tracking, and alignment with course assessment workflows. In many SE laboratory courses, project management, assessment, and collaboration activities remain fragmented across multiple tools, limiting instructors' ability to systematically monitor individual contributions and intervene when imbalances occur.

Gamification, defined as the incorporation of game design elements into non-game contexts, has gained increasing attention as a means of improving motivation, accountability, and engagement. In the context of software engineering education, gamification has been applied to enhance student participation, learning experiences, and collaborative teamwork. Existing studies have reported positive motivational effects through the use of points, badges, and leaderboards. However, despite these benefits, the systematic integration of gamification into team performance monitoring and assessment workflows within software development projects remains relatively underexplored.

To address this gap, this study introduces the Team Performance Gamification Measurement (TPGM) framework, a gamification-enabled approach that supports transparency and accountability in team-based SE projects. The framework integrates self-assessment and peer evaluation with gamified components, namely points, badges, and leaderboards, within a performance dashboard that monitors individual and team contributions. The system is aligned with Agile development practices, enabling continuous feedback, sprint-level monitoring, and timely instructional intervention throughout the project lifecycle.

Although free-riding in team-based learning has been widely examined in prior research, this study does not seek to introduce a new theoretical explanation of the phenomenon. Instead, it contributes a practically grounded, system-level operationalisation for mitigating free-riding within Software Engineering education. The proposed TPGM framework translates established pedagogical and gamification principles into an integrated course management system that aligns assessment structures, team workflows, and performance visibility with Agile-based project practices. By emphasising deployment realism and instructional applicability, this work demonstrates how free-riding can be addressed through process design and assessment integration rather than theoretical abstraction.

The TPGM framework was implemented and evaluated within the TME3413 Software Engineering Laboratory course at Universiti Malaysia Sarawak to assess its impact on student engagement, team accountability, and the reduction of free-riding behaviour. As reported by Jali et al. [1], the framework forms part of a broader immersive learning approach that has shown positive outcomes in enhancing practical learning experiences, team cohesion, and stakeholder collaboration in real-world project settings. The findings of this study aim to support educators and practitioners in fostering more equitable collaboration and improving team performance in software development education.

2 Background and Related Work

2.1 Definitions and Context

Gamification refers to the use of game-design elements, such as points, badges, and leaderboards, in non-game settings to influence user engagement, motivation, and behaviour [2]. In educational contexts, gamification has been employed to make learning experiences more interactive, encouraging participation and persistence. In parallel, free-riding (or social loafing) refers to the tendency of individuals to contribute less effort in group settings, relying on others to complete the task. This phenomenon is particularly prevalent in student group projects and negatively affects both individual and group performance [3], [4].

Team performance in a collaborative learning environment is often hindered by unequal effort distribution. According to Hall and Buzwell [3], free-riding can seriously undermine learning outcomes, while Gedamu and Shewangezaw [4] observed that students with lower levels of interest are more likely to disengage. Ramdeo et al. [5] further argued that successful teamwork can be fostered through peer evaluations, transparency of workload, and effective team leadership.

2.2 Gamification in Software Engineering and Education

Agile software development emphasises collaboration, self-organisation, and iterative progress, making it a fertile ground for gamification techniques that increase engagement and accountability. In both industry and education, gamification has been explored as a means to motivate software developers and students alike [6][7][8]. Monteiro et al. [9] proposed a framework to evaluate gamified software engineering training programs, highlighting commonly used criteria such as engagement, satisfaction, and productivity.

Several empirical studies have supported the use of gamified elements, such as leaderboards, badges, and point systems, in software engineering education. Garcia-Iruela et al. [10] and Souza et al. [11] demonstrated that gamification improved student motivation and participation in project-based learning. Similarly, Portela [12] and Castro et al. [13] found that interactive teaching with game elements enhanced classroom engagement and learning outcomes. Stol et al. [14] confirmed that gamification fosters job satisfaction and productivity in professional software teams.

In the educational domain, Lesley [15] found that transparent gamification strategies helped address free-riding by increasing visibility and accountability among students. John and Fertig [16] reinforced this by showing that badges and points provided students with a clear sense of achievement, thereby improving their sustained motivation throughout software projects.

2.3 Gaps in Current Frameworks

Although promising, many existing gamification implementations treat game elements as supplementary rather than integrated mechanisms for assessment and project tracking. Most do not combine peer, self, and client feedback with real-time dashboards or performance analytics. Moreover, few frameworks actively monitor free-riding behaviour or integrate gamification into Agile-based workflows tailored to educational contexts.

Galeano-Ospino et al. [17] and Barbosa Monteiro et al. [9] emphasised the need for more objective and holistic evaluation models that combine both qualitative and quantitative data. Despite the growing literature on the benefits of gamification, a lack of scalable, integrated frameworks remains which address behavioural issues such as free-riding while enhancing learning transparency and motivation.

To address these gaps, this study proposes the TPGM (Team Performance Gamification Measurement) framework, a structured,

gamified system that embeds evaluation into the project development cycle. It incorporates visual feedback mechanisms, multi-source assessment, and role-specific tracking to promote equity and performance in team-based software development courses.

3 Methodology

This study adopts a system-level, design-based mixed-methods methodology to operationalise and evaluate the Team Performance Gamification Measurement (TPGM) framework as an integrated instructional system for mitigating free-riding in team-based software engineering education. Rather than proposing a new behavioural theory, the methodology focuses on translating established pedagogical, assessment, and gamification principles into a deployable, course-embedded system that restructures how team performance is monitored, evaluated, and reinforced.

The methodology comprises four tightly coupled phases: (i) system-oriented needs assessment, (ii) framework operational design, (iii) system implementation, and (iv) empirical evaluation under authentic course conditions. This approach ensures that the contribution lies in process design and system integration, rather than isolated gamification elements.

3.1 Research Design and Educational Context

A design-based, quasi-experimental research design was employed to evaluate the effectiveness of a course-level operational system in addressing free-riding behaviour. The study was conducted within the TME3413 Software Engineering Laboratory course at Universiti Malaysia Sarawak, a semester-long, project-centric course structured around Agile development, sprint execution, and iterative assessment.

The TPGM framework was not introduced as an external tool or optional intervention. Instead, it was operationalised as a web-based course management system that replaced fragmented tools (e.g., spreadsheets, ad-hoc peer assessments) with a single integrated platform combining:

- task and sprint management,
- individual and team performance measurement,
- multi-source evaluation,
- and gamified performance feedback.

By embedding the framework directly into the course workflow, the study evaluates how system design and assessment architecture influence behaviour, rather than relying solely on self-reported motivation.

3.2 Participants, Team Structure, Roles, and System Needs Assessment

The participants comprised 201 undergraduate software engineering students enrolled in the Software Engineering Laboratory course, organised into small Agile development teams. Each team was responsible for designing, developing, and delivering a complete software system across multiple sprint cycles, following iterative Agile practices. This course structure provided an authentic environment for examining collaboration dynamics, contribution distribution, and accountability within team-based software development.

Prior to system design, a system-oriented needs assessment was conducted to identify structural limitations in existing team-based learning and assessment practices. Survey findings indicated that team project development was perceived as the most challenging learning component, with 58% of respondents identifying it as such. More importantly, the prevalence of free-riding behaviour emerged as a significant and recurring issue. As illustrated in Figure 1, 65 respondents (32.34%) reported sometimes encountering free-riding, 35 respondents (16.92%) often encountered it, and 10 respondents (4.98%) consistently encountered it. In contrast, 58 respondents (28.86%) stated that free-riding occurred seldom, while 34 respondents (16.92%) reported never encountering such behaviour.

Collectively, these findings demonstrate that a substantial proportion of students experienced uneven contribution to varying degrees, confirming that free-riding represents a systemic challenge rather than an isolated incident within team-based software projects. From a system needs perspective, this prevalence highlights limitations in existing assessment and monitoring mechanisms, particularly in visibility into contributions, early identification of underperforming team members, and sustained accountability throughout sprint execution. Furthermore, over 81% of respondents expressed that a structured, gamification-supported system could enhance accountability and improve assessment fairness, providing strong empirical support for a system-level intervention grounded in process and infrastructure design.

To support scalable system deployment, three formal roles were operationally defined and implemented within the platform:

I. Lecturer: responsible for system configuration, assessment orchestration, performance monitoring, and grading oversight;
II. Student: responsible for task execution, sprint participation, self- and peer-evaluation, and progress tracking;
III. Client: responsible for providing external, practice-oriented evaluation of project deliverables.

These roles were not merely conceptual distinctions but functional system actors, each interacting with dedicated modules and workflows within the platform. This role-based operationalisation ensured that accountability, evaluation, and feedback processes were embedded directly into the system architecture, rather than relying on ad-hoc coordination or instructor-mediated enforcement.

3.2.1 Needs Analysis and Design Mapping. To ground the proposed framework in actual learning and collaboration challenges, a needs analysis was conducted prior to system design using a pre-survey administered to students enrolled in the Software Engineering Laboratory course. The survey examined key aspects of team-based project work, including free-riding, perceptions of assessment fairness, accountability during project execution, and the effectiveness of existing evaluation practices.

The survey results indicated that free-riding was a recurring issue, with many students reporting uneven contribution levels within teams. While overall assessment fairness was viewed positively, participants expressed concerns regarding the limited visibility of individual contributions and the delayed identification of underperforming team members. Accountability gaps were also observed during sprint execution, particularly in task ownership and progress monitoring.

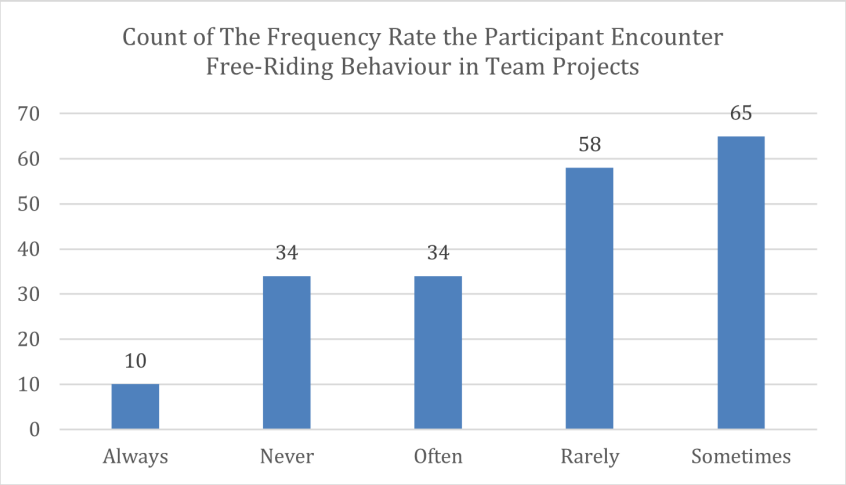

Figure 1: Frequency of Encountering Free-Riding Behaviour in Team Projects

Table 1: Survey Findings and Corresponding TPGM Design Features

Survey Finding	Design Implication	TPGM Feature
Frequent free-riding in team projects	Individual contributions must be traceable	Task completion logs and contribution tracking
Limited visibility of individual effort	Performance transparency is required	Real-time individual performance dashboard
Accountability gaps during sprints	Monitoring must align with Agile workflows	Sprint-based task assignment and tracking
Concerns about assessment fairness	Individual performance should inform grading	Integrated self- and peer-evaluation
Limited external evaluation perspective	Multi-stakeholder feedback is needed	Client evaluation component
Fluctuating motivation over project phases	Continuous engagement reinforcement required	Points, badges, and leaderboards
Delayed performance feedback	Feedback should be timely and continuous	Automated dashboard feedback

These findings directly informed the design of the Team Performance Gamification Measurement (TPGM) framework. Individual-level performance indicators, such as task completion records and contribution tracking, were incorporated to improve transparency. Multi-perspective evaluation mechanisms, including self-assessment, peer evaluation, and client feedback, were integrated to strengthen accountability and fairness in assessment. To sustain motivation and engagement, gamification elements such as points, badges, and leaderboards were aligned with meaningful project milestones rather than isolated activities.

Table 1 presents the mapping between key survey findings and the corresponding design features of the TPGM framework. This alignment ensures that each system component addresses empirically identified needs, thereby strengthening the methodological rigour and practical relevance of the proposed solution.

The insights obtained from the needs assessment were subsequently mapped to specific design decisions in the development of the TPGM framework, as described in the following subsection.

3.3 TPGM Framework and Data Collection

Guided by the issues identified during the needs assessment phase, the Team Performance Gamification Measurement (TPGM) framework was structured around five key components that integrate performance measurement, gamification, evaluation, and feedback within a unified web-based system. The integration of these components was intended to address key concerns related to assessment fairness, contribution transparency, and accountability, as identified by students as critical (Figure 2). Table 2 provides an overview of the TPGM framework's core components and their respective roles.

Data collection was conducted using multiple complementary sources to support triangulation. First, survey questionnaires were used to capture student perceptions of collaboration effectiveness, assessment usefulness, and free-riding behaviour. Second, system-generated data were automatically recorded, including task completion status, peer- and self-evaluation scores, badge acquisition, and leaderboard rankings. Finally, assessment records from lecturers and clients provided formal performance indicators aligned with project deliverables.

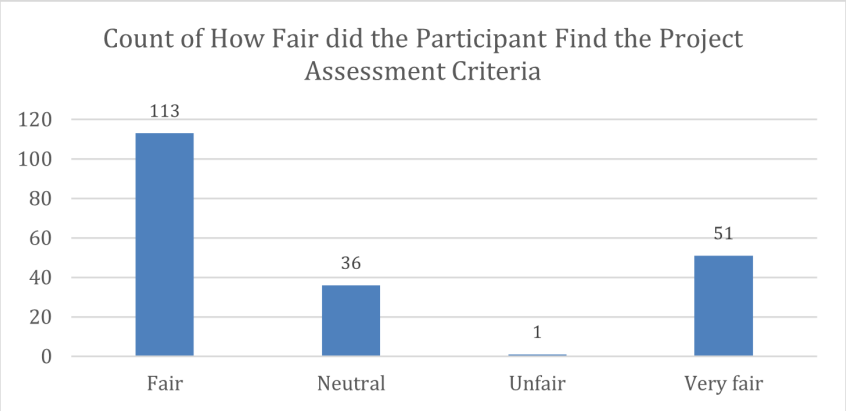

Figure 2: Participants' Perception of the Fairness of Project Assessment Criteria

Table 2: Key components of the TPGM framework

Component	Description
Performance Metrics	Role-specific KPIs were established for frontend and backend developers, testers, and project leads to guide and measure individual contributions.
Gamification Elements	Game-based elements such as points, badges, leaderboards, and performance tiers were embedded to incentivise engagement and align with milestone achievements.
Evaluation Mechanisms	Peer and self-evaluations were conducted at each sprint cycle. These assessments were weighted to represent each member's contribution fairly.
Visualisation Tools	A real-time dashboard was developed to present performance data, individual task activity, and badge achievements in an accessible format.
Feedback Loop	Feedback was collected after every sprint and visually reported, enabling ongoing reflection and performance adjustment for both students and lecturer.

Together, these data sources enabled both perceptual and behavioural analysis of team performance and supported a comprehensive evaluation of the TPGM framework.

3.4 Empirical Evaluation

The effectiveness of the TPGM framework was evaluated using a multi-method empirical approach combining system-generated metrics, survey data, and qualitative observations. This approach enabled a comprehensive examination of student engagement, accountability, and perceptions of fairness during the team-based software development process.

3.4.1 Experimental Procedure. The empirical evaluation was conducted over one academic semester and followed a structured five-stage procedure to ensure consistent application of the TPGM framework across all project teams. First, lecturers configured the system by creating project batches, forming student teams, defining deliverables, and setting evaluation and gamification parameters. Second, students executed their software projects using the integrated task and project management modules, submitting deliverables at predefined sprint milestones.

Third, self- and peer-assessments were conducted at each sprint cycle (sprint milestone) using the evaluation module, generating individual contribution scores. Fourth, gamification feedback, including points, badges, and leaderboard rankings, was automatically updated based on task completion and evaluation outcomes. Finally, post-intervention analysis was conducted using survey responses and system-generated logs captured by the evaluation, task, and gamification modules to assess engagement levels, accountability, and the prevalence of free-riding behaviour.

This structured procedure ensured that all teams experienced the same evaluation and feedback mechanisms throughout the project lifecycle.

3.4.2 Evaluation Methods and Measures. To obtain a comprehensive understanding of the framework's impact, the evaluation employed multiple complementary methods, as summarised in Table 3.

All collected data were anonymised and handled in accordance with institutional ethics protocols. Overall, the evaluation results indicate that the use of gamification within the TPGM framework meaningfully reduced free-riding behaviour, enhanced individual accountability, and improved overall student satisfaction in team-based software engineering projects.

Table 3: Evaluation Methods and Measures

Method	Description
Quantitative Metrics	System logs and badge acquisition data were analysed to assess changes in engagement and task completion across sprints.
Survey Feedback	Post-implementation surveys and student reflections were used to evaluate usability, perceived fairness, and motivation.
Qualitative Observations	Lecturer observations and informal student interviews provided contextual understanding of behavioural changes during the project.

4 System Design and Implementation

The system follows a client-server architecture accessible through web browsers. It integrates various gamification elements and project management functionalities explicitly tailored for software engineering education.

4.1 System Design and Architecture

The Software Engineering Lab Project Management system follows a role-based architecture comprising three primary user roles: Lecturer, Student, and Client, each with distinct access privileges and functional responsibilities. All users interact with the system through a responsive web interface, while backend services manage authentication, evaluation logic, gamification processing, and centralised data storage to support consistent and secure operation.

I. Lecturer: Oversees batch creation, user management (students and clients), project deliverables, evaluations, and project marking. Lecturers also configure and manage gamification components, including badges and visibility of the leaderboard.
II. Student: Engages with the system by managing personal profiles, forming or joining project groups, executing tasks, submitting deliverables, and participating in self- and peer-evaluations. Students monitor their progress through a real-time, gamified performance dashboard.
III. Client: Functions as an external evaluator who assesses project deliverables and provides professional feedback through simplified access using secure evaluation links.

Figure 3 presents the refined use case model of the system, organised around role-centric operational responsibilities. Lecturer use cases focus on course orchestration, assessment configuration, and performance monitoring; student use cases emphasise sprint execution, contribution accountability, and continuous self- and peer-evaluation; while client use cases provide external validation of deliverables. System support services, such as authentication, are abstracted to reduce visual complexity and highlight core instructional workflows. This structure reflects the system-level operationalisation of accountability embedded within the TPGM framework.

All interactions occur through a responsive and intuitive web interface, ensuring smooth integration with the project-based learning workflow. The backend architecture supports user authentication, data processing, and real-time tracking of tasks, evaluations,

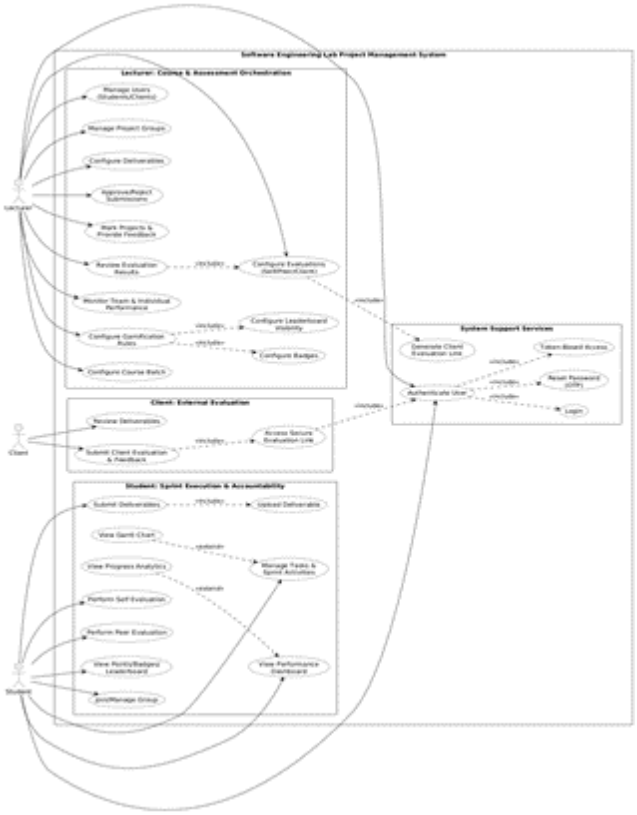

Figure 3: Software Engineering Lab Project Management System Use Case Diagram

and gamification metrics, with centralised data management enabling seamless information sharing among lecturers, students, and clients.

The framework was deployed on a custom-built web platform, accessible via https://seproject.net. Interface snapshots illustrating authentication, navigation, batch management, and badge configuration are shown in Figure 4, Figure 5, Figure 6 and Figure 7. Key modules, including user authentication, batch creation, project management, badge assignment, and leaderboard display, were validated through comprehensive functional test cases. These test cases

Figure 4: Login page

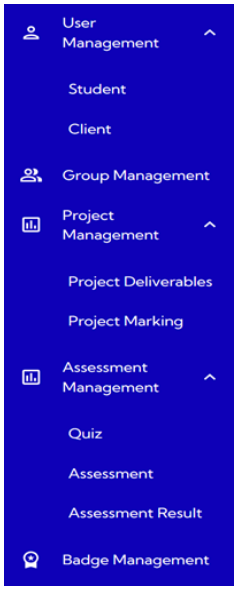

Figure 5: Application menu

confirmed the system's ability to automate essential operations, including badge generation, performance scoring, and evaluation form management, demonstrating its robustness and deployment readiness.

4.2 System Implementation

The system features role-specific interfaces tailored to the needs of three primary user types: lecturer, students, and clients. As shown in Table 4, each interface grants access to a defined set of modules aligned with its responsibilities within the project workflow.

4.2.1 Integration with Project-Based Learning Workflow. The system integrates seamlessly into the project-based learning environment. From initial group formation and task delegation through to final project submission, the system provides comprehensive support. Students engage actively with gamification elements that foster motivation, accountability, and healthy competition. Lecturer monitor team dynamics and project progression, utilising dashboards to intervene or encourage teams as necessary. Clients enhance the learning experience through external feedback on deliverables, directly affecting student learning outcomes and project refinement.

4.2.2 Key Functional Test Cases. The system was validated using a set of functional test cases to ensure reliability, robustness, and compliance with both technical and instructional requirements. These test cases were derived from the system's functional specifications and user roles, focusing on correct module behaviour, data integrity, and appropriate error handling.

Table 5 summarises the main system modules and their corresponding tested functions. For the Authentication Module, test cases verified credential validation, secure login workflows, OTP-based password recovery, and password reset procedures. Error scenarios such as invalid credentials and expired OTPs were also tested to ensure proper system responses.

The User Management Module was tested for batch creation, bulk student imports, and individual profile management for students and clients. These tests ensured correct role assignment, data consistency, and handling of incomplete or duplicate records during bulk operations.

For the Gamification Module, test cases validated badge creation, modification, removal, and ordering, as well as leaderboard visibility and ranking accuracy. These tests ensured that gamification features reflected verified performance data and were updated correctly.

Testing of Group and Project Management covered group creation, project allocation, deliverable submission, approval workflows, and project marking. Feedback visibility and role-based access control were also verified to ensure alignment with assessment requirements.

The Evaluation Management Module was tested for the creation and administration of self-, peer-, and client-evaluations. This included distributing evaluation links, validating submissions, and reviewing completed evaluations to ensure accurate data capture and weighting.

Finally, the Task Management Module was tested for sprint creation, task and todo management, editing and deletion functions, and progress visualisation using Gantt charts. These tests confirmed that task updates were synchronised with performance metrics and dashboard displays.

Overall, the functional test cases confirmed that all system modules operated as intended, handled common error scenarios effectively, and supported transparency, accountability, and fair assessment in team-based Software Engineering projects.

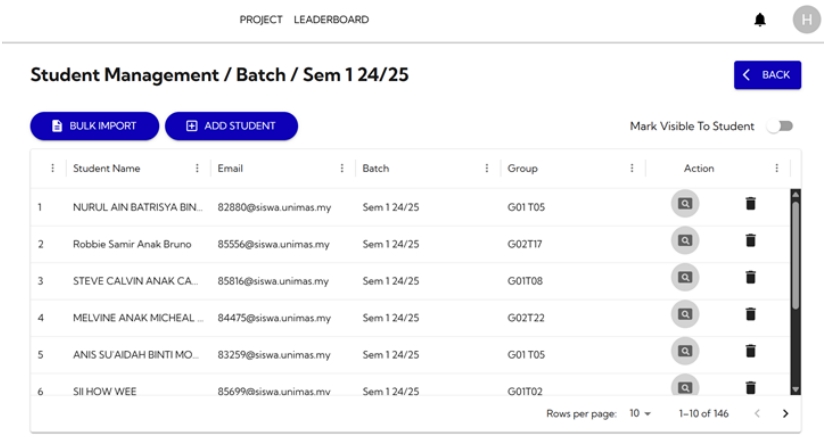

Figure 6: Students and batch management

Table 4: Role-specific functional modules available in the system interface

Module	Lecturer Interface	Student Interface	Client Interface
Authentication	Login, Forgot Password, Reset Password	Login, Forgot Password, Reset Password	Email-based access
User Management	Create & Manage Students, Batches, Clients	Manage own profile, Join/Create Groups	–
Gamification	Create & Manage Badges, Leaderboards	View Points, Badges, Leaderboards	–
Group Management	Oversee Groups, Add/Remove Members	Join, Leave, Edit Groups	–
Project Management	Manage Deliverables, Approvals, Markings	Create/Submit Projects & Deliverables	Review and Evaluate Submissions
Task Management	Oversee Task Progress (To-do, Sprints)	Create/Manage Tasks, To-dos, Gantt Charts	–
Evaluation	Create Assessments (Self, Peer, Client), Review Results	Perform Self & Peer Assessments	Provide Feedback

Table 5: Function and Features of each Module

Module	Functions and Features
Authentication Module	Validating login credentials, password recovery through OTP (One-Time Password), and password resets.
User Management	Batch creation, bulk student imports, individual student and client profile management.
Gamification Module	Badge creation, updating, removal, badge ordering verification, and leaderboard visibility checks.
Group and Project Management	Creation, viewing, and management of student groups, approval workflows for submitted deliverables, and project marking.
Evaluation Management	Creation and management of quizzes and assessments for self, peer, and client evaluations; sending evaluation links to clients; reviewing student and client evaluation attempts.
Task Management	Creation of sprints, standalone todos, and tasks; editing and deletion functionalities; visualization via Gantt charts.

Figure 7: Badge management

5 Results and Analysis

This section presents the findings from a mixed-method empirical evaluation comprising baseline (pre-intervention) survey data, post-intervention perception surveys, system usage analytics, and stakeholder feedback. Data were collected from 201 undergraduate students enrolled in a Software Engineering Laboratory course, and were complemented by observations from lecturers and feedback from client evaluators following system deployment.

5.1 Baseline Survey Results (Pre-Intervention)

The baseline survey revealed critical issues affecting collaborative performance in team-based software development projects prior to the introduction of the TPGM framework. As shown in Figure 8, nearly half of the respondents (49.26%) reported experiencing or observing free-riding within their project teams. In addition, 65% expressed dissatisfaction with the fairness of traditional assessment methods in accurately capturing individual contributions.

These findings establish a clear pre-intervention context, highlighting persistent challenges related to accountability, contribution visibility, and assessment equity. The prevalence of free-riding and dissatisfaction with existing evaluation practices provides a necessary baseline against which post-intervention outcomes can be interpreted.

5.2 Post-Intervention Survey Findings and Effectiveness of Gamification Elements

Post-intervention survey results indicate a substantial positive shift in student perceptions following the implementation of the TPGM framework. As illustrated in Figure 9, 81.69% of participants reported that the introduction of gamification enhanced team accountability and reduced free-riding behaviour. This contrasts with the baseline findings, suggesting a strong perceived impact of the framework after deployment.

The effectiveness of individual gamification elements was evaluated using post-intervention perception surveys, supported by verification of consistent interaction patterns in system usage logs across sprint cycles. As shown in Figure 10, students rated badges and real-time points as the most effective motivational tools, followed by dashboards and leaderboards. These features were perceived to support engagement by providing continuous recognition and real-time feedback on progress.

Although some students expressed concerns regarding leaderboard-induced anxiety, a substantial majority (73%) viewed leaderboards as promoting healthy competition and timely task completion. The performance dashboard was also positively received for transparently displaying contribution scores and enabling self-reflection. As illustrated in Figure 11, visible achievement tracking played a key role in supporting both motivation and self-regulation within teams. Collectively, these findings indicate that the effectiveness of gamification arises from the complementary roles of its components rather than any single element in isolation.

5.3 Feedback from Stakeholders

Qualitative feedback from key stakeholders further supports the observed trends. Lecturer observed improved participation levels and increased punctuality in deliverable submissions. They noted that the system made previously hidden group dynamics and individual efforts more visible, enabling more evidence-based and objective assessment.

Client evaluators, invited through secure evaluation links, successfully reviewed project deliverables and provided structured feedback. Their input was integrated into the evaluation module, enriching the assessment process with external professional perspectives.

Student reflections indicated that gamification features enhanced engagement and reduced complacency through progression indicators and recognition mechanisms. Some students suggested additional refinements, such as customisable badges and private peer feedback, to further support personalised learning experiences.

Taken together, the convergence of feedback from students, lecturers, and clients indicates improved transparency, accountability, and engagement, strengthening the practical validity of the TPGM framework.

5.4 Summary of Analytical Trends

Across baseline survey data, post-intervention perceptions, system usage verification, and stakeholder feedback, a consistent trend emerges: increased visibility of individual contributions through dashboards, evaluations, and gamification feedback is closely associated with improved accountability and engagement. Students who reported clearer insight into their performance also expressed greater satisfaction with collaboration and with the fairness of assessments.

While the findings are based on descriptive statistics and self-reported perceptions, and thus do not support causal inference, the observed patterns suggest that integrating gamification with continuous performance tracking can meaningfully address free-riding behaviours in team-based software engineering education.

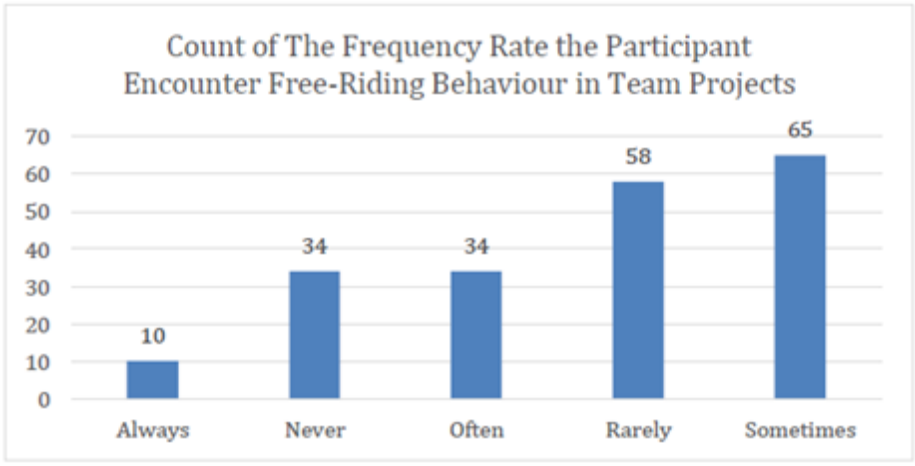

Figure 8: Frequency of Encountering Free-Riding Behaviour in Team Projects

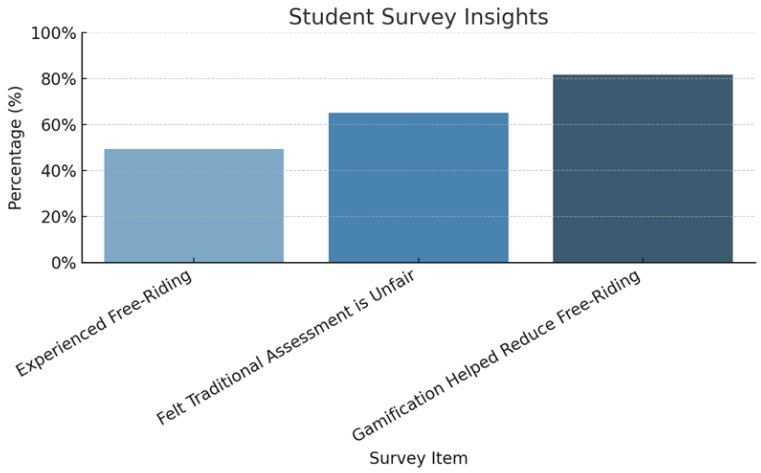

Figure 9: Students' perceptions of free-riding and the impact of gamification

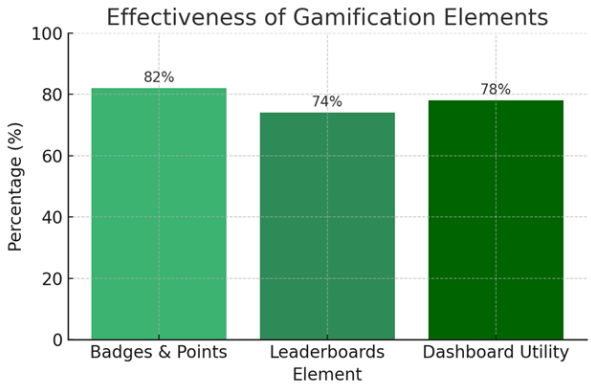

Figure 10: Effectiveness ratings of gamification elements by students.

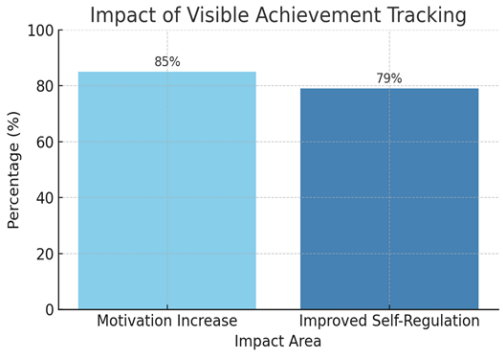

Figure 11: Impact of visible achievement tracking

6 Discussion

This study demonstrates that the gamification-enhanced TPGM framework positively influenced student engagement, accountability, and assessment transparency in team-based software engineering projects. The findings suggest that free-riding is not solely an individual behavioural issue but a process and visibility problem commonly embedded in collaborative project environments. By integrating self- and peer-evaluation, performance dashboards, and gamification feedback directly into the project workflow, the framework transformed accountability into a continuous, observable process rather than a retrospective grading activity.

A substantial proportion of students (81.69%) reported that gamification reduced free-riding and enhanced participation. Elements such as points, badges, and leaderboards increased awareness of progress and responsibility for task completion, while the inclusion of peer and client evaluations provided multidimensional feedback that strengthened perceptions of fairness. Although some students expressed discomfort with public leaderboard visibility, the overall response reflected improved motivation and more timely task completion.

These results are consistent with prior studies highlighting the motivational and regulatory potential of gamification in educational contexts. Pedreira et al. [8] emphasised the role of gamified architectures in improving performance tracking and transparency, while Lesley [15] reported increased participation and accountability through well-integrated game elements. The key contribution of the TPGM framework lies in its holistic integration of role-specific evaluations, gamification mechanisms, and real-time dashboards within Agile-aligned workflows. Rather than functioning as isolated incentives, gamification operates as an ongoing performance measurement and feedback mechanism that enables early identification of contribution imbalances.

Several limitations should be noted. The study was conducted within a single course and institution, limiting generalisability, and did not examine long-term behavioural or learning retention effects. Additionally, individual emotional responses to leaderboard visibility and the depth of qualitative client feedback were not explored extensively. Future research should consider multi-institutional deployments, longitudinal evaluations, and adaptive gamification strategies to better accommodate diverse learner preferences.

7 Conclusion and Future Work

This study introduced and evaluated the Team Performance Gamification Measurement (TPGM) framework as a structured intervention to address free-riding, enhance engagement, and improve transparency in team-based software development education. By integrating gamified elements, such as badges, points, leaderboards, and real-time dashboards, with self, peer, and client evaluations, the framework provided a multifaceted approach to measuring and motivating student performance. Empirical findings from surveys and feedback indicated increased accountability, motivation, and student satisfaction, confirming the framework's potential to address persistent issues in collaborative learning environments.

For educators and course designers, the TPGM framework offers practical benefits by making student contributions more visible, encouraging balanced participation, and enabling data-driven assessments. The modular system design also allows for integration into existing project-based learning workflows, providing real-time feedback for both students and instructors. Educational technology developers can leverage this framework to enhance Learning Management System (LMS) platforms or build standalone, gamified environments that support scalable, team-based projects.

Future research should explore adaptive gamification strategies that personalise motivational elements based on individual learner profiles and performance trends. Longitudinal studies across semesters and institutions could offer deeper insights into behavioural change, learning retention, and scalability. Additionally, enhancements such as customisable avatars, private peer feedback, and more nuanced analytics may further support inclusive and compelling learning experiences. By continuing to refine and contextualise the framework, this line of inquiry can make meaningful contributions to both software engineering education and the broader field of technology-enhanced learning.

Acknowledgments

The authors appreciate the reviewers' thoughtful and rigorous comments, which enhanced the manuscript. This research is fully supported by the Universiti Malaysia Sarawak (UNIMAS), the Scholarship of Teaching and Learning (SoTL) Grant, UNI/F08/SoTL-RG/85053/2023. The authors fully acknowledged UNIMAS for the approved funding, which enabled this important research to be viable and effective.

References

[1] Jali, N. M. Hamdan, C. W. Shiang, and S. K. Jali, "Developing an evaluation framework for immersive learning experiences for software engineering project course," in Proc. Knowledge Management International Conf. (KMICe), Virtual, Feb. 2021.
[2] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," Proc. 15th International Academic MindTrek Conference, pp. 9–15, 2011.
[3] N. Hall and S. Buzwell, "The problem of free-riding in group projects," Australian Journal of Education, vol. 57, no. 1, pp. 25–41, 2013.
[4] A. Gedamu and A. Shewangezaw, "Free riding and student disengagement in group work," Journal of Education and Practice, vol. 13, no. 5, pp. 55–63, 2022.
[5] R. Ramdeo, K. V. Patel, and S. Perumal, "Managing free-riding in student group work: Technology and peer review strategies," Education and Information Technologies, vol. 27, no. 6, pp. 8345–8360, 2022.
[6] A. Mishra and P. Gupta, "Gamification in education: A review," Education and Information Technologies, vol. 26, no. 2, pp. 1233–1247, 2021.
[7] B. Priyadi, D. Wibowo, and T. Iswahyudi, "Gamification implementation for software engineering education," Journal of Physics: Conf. Series, vol. 1933, 2023.
[8] Ó. Pedreira, F. García, M. Piattini, A. Cortiñas, and A. Cerdeira-Pena, "An Architecture for Software Engineering Gamification," arXiv preprint arXiv:2402.00233, 2024.
[9] B. Monteiro, T. Barbosa, and R. Queirós, "A framework for assessing gamification in software engineering education," Computer Applications in Engineering Education, vol. 29, no. 3, pp. 617–628, 2021.
[10] A. Garcia-Iruela, J. M. Garcia, and A. Villalobos, "Gamification and engagement in software engineering courses," IEEE Global Engineering Education Conference, pp. 540–544, 2020.
[11] M. Souza, P. Barros, and C. Fonseca, "Using gamification to enhance the learning experience in software engineering education," IEEE Frontiers in Education, pp. 1–7, 2020.
[12] M. Portela, "Gamified learning for computer science classrooms: A case study," International Journal of Emerging Technologies in Learning, vol. 15, no. 10, pp. 175–183, 2020.
[13] M. Castro, L. Brito, and J. Silva, "Games for software engineering education: A mapping study," IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 15, no. 1, pp. 33–40, 2020.

[14] K. Stol, F. Fitzgerald, and M. Stol, "Gamification, developer engagement, and team performance," Empirical Software Engineering, vol. 27, no. 5, pp. 98–117, 2022.
[15] L. Lesley, Exploring the Application of Gamification in the Software Development Process, M.Sc. thesis, 2022.
[16] T. John and M. Fertig, "Gamification and student engagement in computer science," International Journal of Engineering Pedagogy, vol. 12, no. 4, pp. 47–60, 2022.
[17] H. Galeano-Ospino, M. Barbosa Monteiro, and B. Priyadi, "Knowledge transfer and collaboration through gamification in software development teams: A review," Journal of Systems and Software, vol. 190, 2022.