Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2026, 20(8), 172–184
https://doi.org/10.12913/22998624/221018
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.03.30
Accepted: 2026.06.01
Published: 2026.07.01

Model-based systems engineering in student engineering projects: A case study of Phoenix III Rover design

Andrzej Jałowiecki[1*], Jakub Gurgul[2]

[1] Department of Fundamentals of Machinery Design, Faculty of Mechanical Engineering, Silesian University of Technology, Gliwice, Poland
[2] Silesian University of Technology, Gliwice, Poland
* Corresponding author e-mail: andrzej.jalowiecki@polsl.pl

ABSTRACT

Model-based systems engineering (MBSE) shifts systems engineering from document centric to model-centric approaches. Its application in student engineering projects remains understudied in European educational contexts despite widespread industrial adoption. This paper investigates MBSE implementation in group-based student engineering projects, examining how systematic requirements management, architectural design, and traceability practices enhance project outcomes and team collaboration. A case study analysed the implementation of MBSE in Silesian Phoenix student association autonomous rover design for Rover Challenge competitions. Data collection included project documentation and technical artefacts (SysML models, requirements matrices, design specifications) across a 24-month project cycle. Structured MBSE implementation demonstrated significant improvements: requirements clarity increased, design traceability achieved much higher coverage, and interface incompatibilities were detected during the design review rather than the integration phase. Critical success factors included early methodology training, dedicated configuration management, and formal stakeholder validation. MBSE provides an effective pedagogical framework for engineering education, developing systems thinking capabilities while improving project execution. The study identifies implementation barriers and provides practical recommendations for educators integrating MBSE into project-based curricula.

Keywords: model-based systems engineering, MBSE, SysML, student projects, requirements, case study.

INTRODUCTION

Modern engineering challenges increasingly demand systematic approaches to managing requirements, architectural decisions, and system complexity [1]. The traditional document-centric systems engineering (SE) approach struggles to maintain consistency, traceability, and integration across large, heterogeneous teams [2]. Model-based systems engineering (MBSE) addresses these limitations by shifting focus from document generation to a formal system model serving as the single source of truth [3].

Student engineering competition projects — such as the Canadian International Rover Challenge (CIRC), the European Rover Challenge, or Formula Student — combine real-world constraints with educational objectives, providing unique environments for exploring MBSE. The Silesian Phoenix student association at Silesian University of Technology (SUT) exemplifies this setting: competing in the CIRC with a multidisciplinary team coordinating mechanical, electronics, control, and software work, it offers an authentic context for examining how MBSE manages project complexity [4].

While MBSE frameworks are increasingly adopted in industry, their application in student engineering education remains under-explored. Existing literature focuses principally on North American settings or isolated course modules [5, 6, 7], leaving the European academic context and sustained competition-based environments uncharted. Published studies moreover tend to report either technical or pedagogical outcomes in isolation — a gap reflected in engineering graduates' limited exposure to formal SE methodologies increasingly expected in professional practice.

Four questions motivate the study: How can MBSE be adapted for student-driven projects while maintaining pedagogical value? What benefits does structured MBSE provide to team coordination, requirements management, and design quality? What are the primary adoption barriers and how can they be addressed? How does MBSE practice shape the development of systems thinking and professional competencies?

This paper presents an exploratory case study — in the tradition of Yin [8] and Eisenhardt [9] of MBSE implementation within the Silesian Phoenix rover project. Rather than testing universal hypotheses, the study documents one implementation trajectory to yield transferable insights and theoretically generalisable propositions.

Five objectives guide the work: first, to characterise MBSE adoption in student engineering projects through a literature review; second, to document the implementation strategy within the Silesian Phoenix project; third, to analyse quantitative process indicators and qualitative participant perceptions; fourth, to identify critical success factors and implementation barriers; and fifth, to provide evidence-based recommendations for educators and project leaders integrating MBSE into comparable contexts.

BACKGROUND

System engineering and MBSE

Systems engineering (SE), as defined by INCOSE, constitutes a transdisciplinary and integrative approach enabling the successful realisation, use, and retirement of engineered systems [3]. The foundational document-centric paradigm, described by Kossiakoff et al. [1], relies on textual specifications as the primary vehicle of design intent. While effective for relatively stable, low-complexity systems, document-centric SE has been widely criticised for its inability to maintain consistency, traceability, and integration in large heterogeneous projects [2]. MBSE emerged as a response, formalising the application of modelling to support requirements, analysis, design, and verification across the full system lifecycle [3].

Madni and Purohit [10] provide an empirical economic case for MBSE, demonstrating measurable reductions in integration failures and rework costs across aerospace and defence programmes. Cole et al. [6] corroborate this through the TALOS technology maturation project, where a unified SysML model enabled traceability from performance requirements to hardware integration across a distributed team. Siddique [11] further documents MBSE adoption in small satellite development, arguing that even resource-constrained projects benefit from the architectural clarity and early risk identification afforded by formal models.

The literature does not, however, present a uniformly positive picture. Vogelsang et al. [5], drawing on twenty semi-structured interviews across ten embedded-systems companies, identify a persistent tension between MBSE's theoretical benefits and its practical adoption: key hindering forces include tool immaturity, unclear ROI for small-scale projects, and organisational resistance to process formalisation. Ma et al. [7] similarly highlight tool-chain fragmentation and the lack of standardised integration pathways as barriers disproportionately affecting teams without dedicated SE support. This tension has received limited empirical investigation in student project contexts, where resource constraints and transient team composition compound the challenges identified in industry.

SysML and requirement management

SysML, standardised by the Object Management Group [12], provides the primary formal notation underpinning MBSE practice, extending UML to encompass requirements, behaviour, structure, and parametric modelling within a single framework. SysML v2 introduces a unified metamodel, textual notation, and improved interoperability; however, its educational adoption remains nascent — most published instructional studies still employ v1 toolchains [13], and empirical evidence on its pedagogical effectiveness in student environments is largely absent.

Requirements management is one of the highest-value MBSE applications. ISO/IEC/IEEE 29148:2018 [14] establishes structured requirement statement templates, and Wheaton and Herber [15] demonstrate how an INCOSE-derived SysML meta-model can implement these programmatically, enabling automated traceability links between requirements, architectural elements, and verification artefacts — making requirements engineering an executable, traceable process rather than a documentation activity.

Bajczi et al. [13], reporting on an MBSE course with more than eighty students annually, show that version-controlled model repositories and automated feedback significantly reduce coordination overhead in large student teams — an insight directly relevant to the present case. Existing studies on SysML in educational settings nonetheless predominantly address tool selection and diagram usage rather than how students internalise requirements traceability as a disciplined practice; evidence specific to the European academic context, where MBSE is less embedded in curricula than in North American or aerospace-dominated environments, remains sparse.

Project-based learning and engineering competition projects

Project-based learning (PBL) has accumulated substantial empirical support for developing higher-order engineering competencies. Hmelo-Silver [16] established the foundational framework linking authentic problem contexts, student agency, and iterative design cycles to deep conceptual learning. Picard et al. [17], in a study of 306 engineering students, demonstrate that communication and cross-functional coordination are most reliably developed through team-based projects, while planning and risk management require more explicit scaffolding. Ghannam and Chan [18], reporting a hybrid PBL and Team-Based Learning approach over five years with more than 300 undergraduates, show that systems-level thinking — difficult to achieve through conventional instruction — can be cultivated through sustained project work.

Engineering competition projects are a particularly rich PBL context. Lemu [19] documents how Formula Student, European Rover Challenge, and comparable competitions provide external validation and impose authentic constraints — time pressure, budget limits, multidisciplinary coordination — absent from purely academic exercises. Talmi et al. [20] add the motivational dimension, demonstrating that Formula Student participants develop intrinsic motivation and twenty-first-century competencies — systems thinking, iterative problem-solving, professional communication — that persist beyond the project, findings directly applicable to rover challenge programmes sharing the same structural characteristics.

The integration of formal SE methodology into PBL-based competition projects remains poorly documented. Halvorson et al. [21] represent one of the few exceptions, reporting a SysML-based MBSE model deployed across approximately eighty-five undergraduates in a CubeSat mission and highlighting MBSE's value for managing knowledge transfer during the high turnover characteristic of student teams. Li et al. [22] propose a structured MBSE-driven PBL framework, demonstrating that lifecycle-oriented process modelling improves both systems thinking and project execution quality. Both studies originate from North American contexts; comparable empirical evidence from European competition-based rover or spacecraft projects is absent from the literature.

Research gap and positioning of the present study

The f oregoing review reveals three converging gaps. First, while MBSE's industrial benefits are well documented [10, 6], its adoption dynamics in student engineering projects remain empirically under-examined — existing studies either target professional engineers [5] or focus on isolated course modules rather than sustained competition projects [20, 21]. Second, the few studies addressing MBSE in student contexts [22, 21] originate almost exclusively from North American institutions; given that Vogelsang et al. [5] note adoption barriers vary substantially with context, the European academic setting has not been systematically studied. Third, despite strong evidence that competition-based PBL develops professional competencies [17, 20], the deliberate integration of MBSE into such projects — and its effect on both technical outcomes and student learning — has not been documented with longitudinal depth.

The present study addresses these gaps through an in-depth, longitudinal account of MBSE implementation within the Silesian Phoenix student association at the Silesian University of Technology — a Polish institution within the European engineering education system. Examining a 24-month project cycle with access to quantitative project metrics and qualitative participant perspectives, the study contributes empirical evidence on MBSE adoption dynamics, pedagogical outcomes, and implementation barriers in a context largely invisible in the existing literature.

METHODOLOGY

This research employs a single in-depth case study design, as outlined by Yin [8]. Case study methodology is particularly appropriate for addressing "how" and "why" questions in real-world contexts where the researcher has limited control and where contextual factors are important [9]. The choice of case study design reflects the research aim: to understand the nuanced process of MBSE implementation in authentic project contexts rather than to evaluate universal hypotheses.

Case study and context

The empirical investigation centres on the Silesian Phoenix student association Canadian International Rover Challenge project cycle at SUT. The team comprises approximately thirty engineering students from mechanical engineering, automation and robotics, electrical engineering, and computer science backgrounds. The project spans 24 months (September 2023 – August 2025), providing longitudinal depth for observing methodology adoption and maturation. The case was selected on five criteria: sufficient technical complexity to justify MBSE; team size and disciplinary composition requiring systematic coordination; extended duration enabling methodology maturation; feasible research access to artefacts and team members; and an explicit organisational decision by project leadership to implement MBSE, creating conditions favourable for embedded observation.

Data collection

Data collection proceeded through two complementary channels. The primary source consisted of project artefacts: the SysML v2 system model developed in Magic Systems of Systems Architect Community Edition, serving as the principal repository of requirements, architectural decisions, and interface specifications; requirements traceability matrices linking mission-level objectives to component-level specifications and verification evidence; and design review records documenting identified issues and change requests.

The secondary source comprised informal conversational interviews with student team members at two project milestones: the early implementation phase (months 4–6) and the closing design phase (months 20–22). These exchanges were unstructured, functioning as reflective discussions rather than standardised interviews, oriented towards capturing participant perceptions of the methodology's utility, challenges encountered, and observed changes in practice. Approximately fifteen students participated across both rounds, drawn from mechanical, software, and systems integration roles. Given their informal nature, these exchanges are treated as qualitative contextual evidence rather than primary interview data, with no claim of representational completeness.

Analysis framework

Quantitative indicators

Given the exploratory single-case design, the quantitative component does not employ inferential statistical methods. A set of process-oriented indicators was defined to enable structured comparison between the MBSE-supported Phoenix III cycle (2023–2025) and the immediately preceding design cycle for the same science module subsystem, executed by a different team without a formalised MBSE approach. These indicators serve as illustrative proxies rather than statistically validated outcome measures, and were defined as follows: (a) Requirements ambiguity rate — the proportion of requirements at a given milestone containing vague qualifiers (e.g., "should be robust") or lacking verifiable acceptance criteria, assessed through structured requirements review sessions. (b) Traceability coverage — the percentage of L2/L3 requirements with a complete upward link to mission objectives and a downward link to at least one verification test case within the SysML model at the time of the final design review. (c) Interface incompatibility detection rate — the proportion of interface-related issues identified during design review rather than during physical integration. (d) Rework iteration count — documented design change requests requiring rework of previously approved elements, recorded in the change management log during the integration phase.

Indicator values were estimated through systematic review of SysML artefacts, design review minutes, and change management records; comparison values for the preceding cycle were reconstructed from available documentation. This comparison is not a controlled experiment: the two cycles differed in team composition, accumulated experience, supplier relationships, and competition requirements. The indicators are therefore presented as contextual illustration, not as evidence of causal attribution to MBSE alone.

Qualitative analysis

Qualitative analysis draws on three evidence streams: project artefacts and documentation, informal participant conversations, and the researcher's embedded observation in the role of systems engineering advisor. Analytical procedures included document analysis tracing the evolution of requirements, model structure, and design decisions; pattern identification within participant accounts, focusing on recurring themes related to adoption, perceived benefits, and obstacles; and reflective interpretation of the researcher's observations, made explicit to address potential observer bias from the dual practitioner-investigator role. Data source triangulation [23] was employed to strengthen credibility: claims were cross-validated across at least two evidence streams before being reported as findings. Where convergence was absent, findings are presented as tentative observations, consistent with interpretive case study practice [24].

Limitation of the methodology

This investigation is subject to several explicit limitations. The single-case design constrains comparative and cross-case analysis. The bounded institutional context limits contextual diversity. The quantitative indicators derive from comparative estimation rather than controlled measurement, as the reference baseline involved a different team under partially different conditions. The dual role of the principal researcher as systems engineering advisor and investigator introduces potential observer bias, partially mitigated through triangulation.

A further consideration concerns the inherent difficulty of comparative interpretation in student engineering project research. Unlike industrial MBSE deployments executed by stable professional teams, student projects are characterised by structural variability that limits cross-case comparability: team composition changes substantially between cycles due to graduation and recruitment; members' prior experience varies across cohorts; and the competitive context evolves as organisers revise scoring criteria annually. The present study is therefore best understood as a documented academic experiment — an attempt to systematically implement MBSE in an authentic student engineering context and observe its effects — rather than a controlled evaluation of MBSE's causal impact. Following Yin [8] and Eisenhardt [9], its contribution aspires to analytic rather than statistical generalisation.

USE CASE: MBSE IMPLEMENTATION FOR THE SCIENCE MODULE

Motivation for MBSE implementation

Technical drivers

Increasing system complexity constituted the primary technical motivation. The progressive evolution of the rover architecture resulted in a dense network of interdependencies among mechanical, electrical, and software subsystems, intensifying integration challenges - particularly during system-level verification, where requirement inconsistencies frequently remained undetected until late project stages. The science module additionally required compliance with approximately eighty technical requirements and competition evaluation criteria, necessitating rigorous traceability across all design and verification activities. Supply chain constraints further strengthened the need for early and precise requirement formulation to mitigate late-stage design risks.

Organisational drivers

Transient team composition typical of student projects created a continuous need for systematic knowledge retention and transparent documentation. Collaboration across mechanical, software, and scientific subsystem teams — together with faculty advisors and external mentors - introduced coordination challenges that hindered efficient communication. Adopting MBSE also aligned with institutional expectations for professional engineering practices and strengthened the connection between student experience and INCOSE-aligned industrial systems engineering.

MBSE framework for the science module

Requirements management approach

Requirements engineering follows ISO/IEC/IEEE 29148 [14], ensuring a consistent language and hierarchical structure across abstraction levels. Each requirement adheres to the formalised template <subject> shall <action> <object> <condition>, facilitating clear interpretation and digital traceability. An illustrative functional requirement reads: "The system shall acquire regolith samples with a maximum extraction depth of 30 cm and a ±2 cm tolerance when operating on simulated Martian terrain." Requirements are structured into four levels: L0 — Mission-Level Objectives; L1 — System-Level Requirements; L2 — Subsystem-Level Requirements; and L3 — Component-Level Requirements. Bidirectional traceability was maintained across all layers: L0 objectives were decomposed into L1 system requirements, allocated to L2 subsystems, and expanded into L3 component specifications, with verification test cases explicitly mapped to the corresponding requirements.

SysML profile customisation

Given the scope and interdisciplinary nature of the rover system, a tailored SysML v2 modelling profile was developed rather than employing the complete language specification. This pragmatic strategy aimed to balance model precision and usability within an academic, student-driven environment, ensuring that essential system aspects were captured without excessive methodological overhead.

A tailored SysML v2 modelling profile was developed — rather than adopting the full language specification — to balance model precision and usability within a student-driven environment. The profile focused on four model view types: requirement and traceability views, capturing mission, system, and interface requirements with typed relationships (satisfy, verify, refine, derive) (Figure 1); structure views for hierarchical system decomposition and interface specification; use case and scenario views bridging stakeholder needs and behavioural models (Figure 2); and behaviour and state views representing operational modes, transition logic, and event-driven state changes.

In the context of this case study, most observed benefits are attributable to the shift towards a model-centric approach rather than to SysML v2-specific features per se. The unified metamodel and textual notation primarily lowered the entry barrier for students and improved model consistency; the underlying MBSE process gains could in principle also be achieved with a suitably disciplined SysML v1 toolchain.

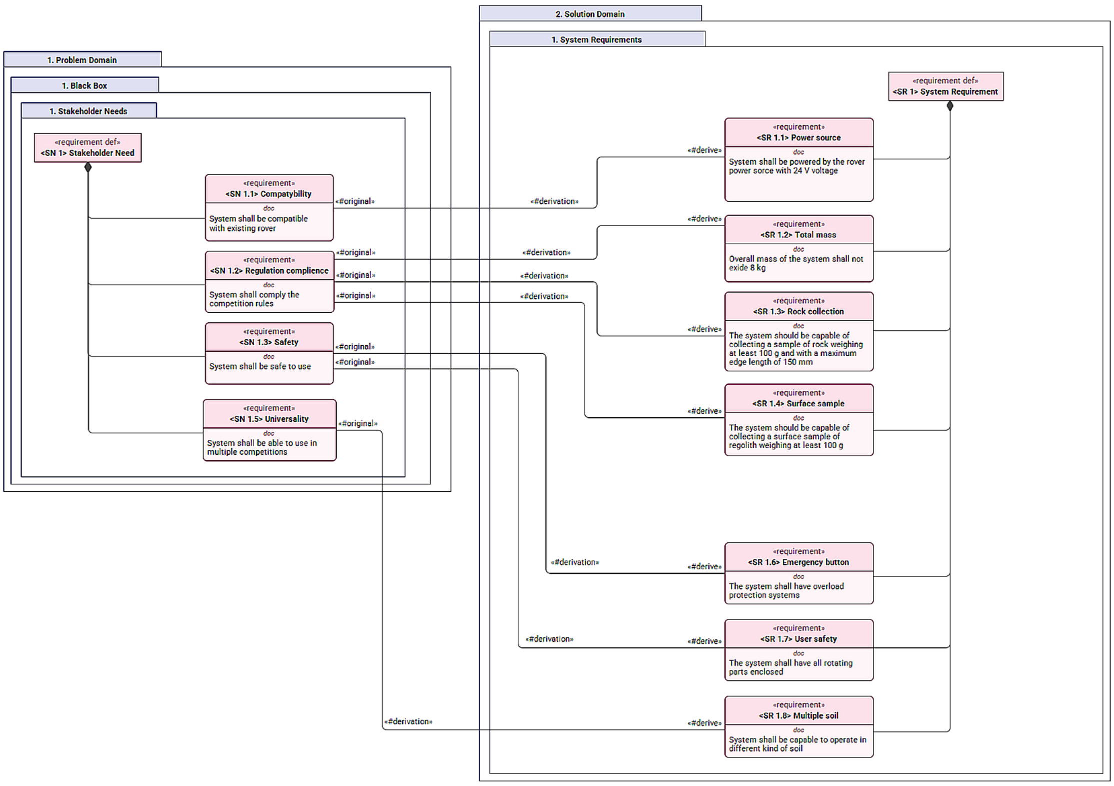

Figure 1. Requirements derivation view

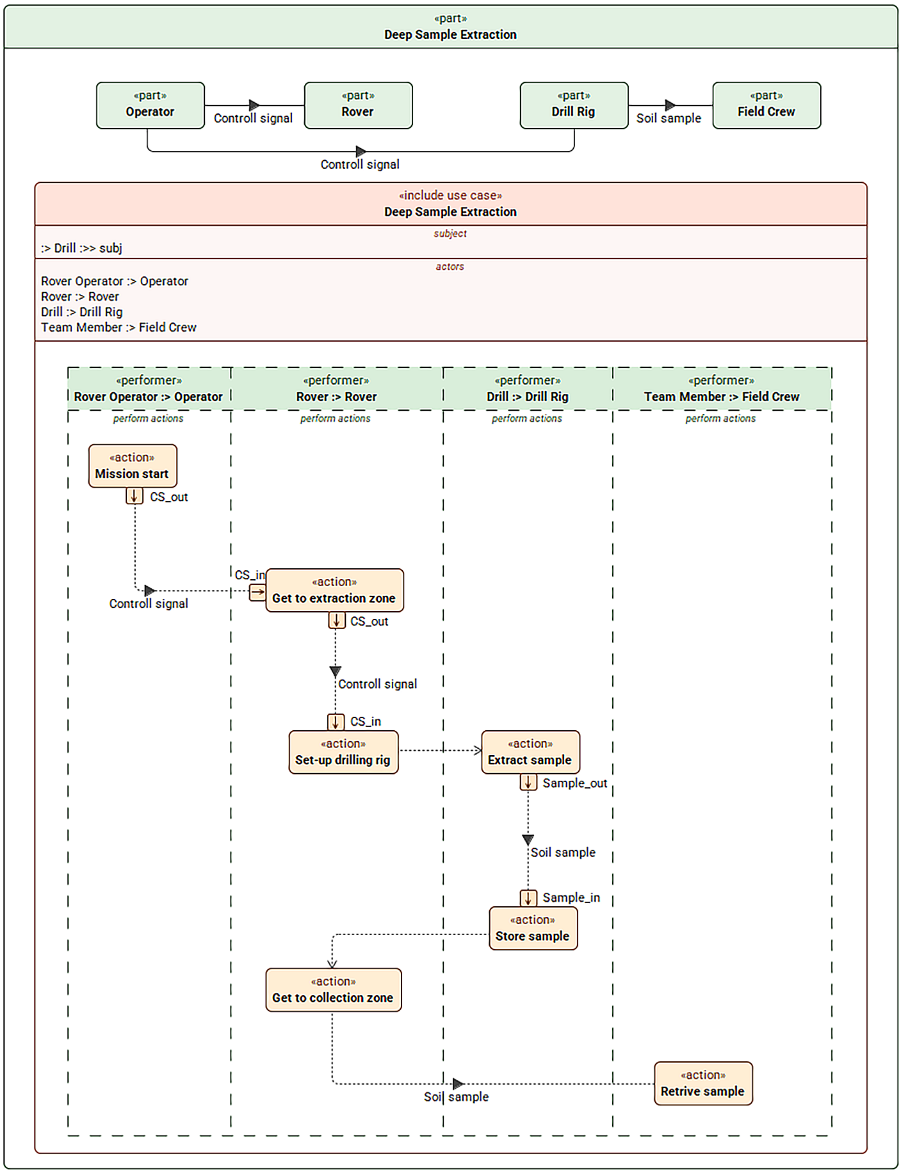

Figure 2. Use case view for deep sample collection

DISCUSSION

Table 1 summarises the key quantitative indicators observed before and after MBSE adoption in the Silesian Phoenix project.

Nature of contribution

The Silesian Phoenix case study constitutes an application-oriented contribution: its primary value lies not in advancing MBSE theory, but in providing a documented, contextualised account of how MBSE principles can be adapted and implemented within a student engineering project operating under authentic competitive constraints. The study does not seek to establish universal causal laws; rather, it offers an empirical description of one implementation trajectory - the choices made, the outcomes observed, and the conditions that shaped them — from which transferable insights for educators and project leaders may be drawn. The quantitative indicators discussed below function as illustrative process markers rather than statistically validated outcome measures.

MBSE attributable effects and confounding factors

Interpreting the observed improvements requires distinguishing between effects mechanistically linked to MBSE adoption and those reflecting broader organisational learning across

Table 1. Comparison of selected metrics before and after MBSE adoption

Metric	Before MBSE	After MBSE
Share of ambiguous requirements	~22%	~5%
Traceability coverage	~58%	~91%
Interface incompatibilities detected during design review (vs. integration)	~20%	~85%
Rework iterations in integration phase	4	2
Team members using the SysML v2 model	3	15

Note: Values are illustrative estimates based on systematic review of project artefacts and comparison with the preceding design cycle. They do not constitute statistically validated measurements.

competition cycles. The most directly MBSE-attributable outcomes are those depending on the presence of a coherent model-centric representation. The increase in requirements traceability coverage is the clearest example: bidirectional links between mission objectives, subsystem requirements, and verification test cases cannot exist without a structured model to host them. Similarly, the shift of interface incompatibility detection from integration to earlier design reviews is a direct consequence of formal interface specification, since typed SysML connections permit analytical consistency evaluation before physical hardware is assembled.

By contrast, improvements such as the reduction in integration-phase rework iterations are more appropriately attributed to an interaction between MBSE adoption and accumulated team maturity — returning members brought established supplier relationships and refined manufacturing knowledge that cannot be isolated from the methodological effect. Vogelsang et al. [5] note that the perceived value of MBSE varies substantially with team compositional stability and modelling expertise, while Cole et al. [6] observe that a unified system model's benefits become most tangible where shared understanding cannot be maintained through informal communication alone — a condition that characterises large multidisciplinary student associations.

Requirements engineering

The most striking qualitative observation concerns the transformation in how team members related to requirements. In the early implementation phase, participant accounts consistently described requirements as an administrative obligation — a checklist imposed by the competition or by project leadership. By the closing design phase, the same participants articulated requirements as a navigational instrument: a means of evaluating design trade-offs, justifying decisions to external reviewers, and detecting inconsistencies before they propagated downstream. This shift in epistemic stance represents the internalisation of a professional engineering discipline rather than mere procedural compliance, and is more significant than any individual metric improvement.

The structured requirement statement template derived from ISO/IEC/IEEE 29148 [14] – demanding that each requirement specify a subject, an action, an object, and a condition – played a central role in this transformation. Participants reported that writing verifiable requirements changed how they approached design problems: rather than proceeding from intuitive solutions toward post hoc justification, they increasingly began from explicit requirement statements. Wheaton and Herber [23] describe precisely this cognitive shift as a primary mechanism through which SysML-based requirements engineering improves design quality, noting that formalising requirements exposes ambiguities that informal communication consistently conceals. Requirements engineering represents a well-documented competency gap in engineering graduates [23]; several participants noted that the experience altered their approach to problem specification in other contexts, suggesting durable competency development rather than project-specific skill acquisition.

Interface management and architectural thinking

Prior to MBSE adoption, interface agreements between subsystems were typically negotiated bilaterally and informally, with no formal record binding subsequent design decisions. Under the MBSE framework, interface specifications were encoded as typed connections within the SysML structural model, making them visible to the entire team and subject to formal review. This shift produced two distinct benefits: systematic identification of inconsistencies – voltage mismatches, mechanical clearance conflicts, and communication protocol incompatibilities – before physical integration; and a shared reference point for cross-subsystem communication that reduced dependency on individual relationships and informal knowledge. The latter is particularly critical in student teams where personnel turnover is structural rather than exceptional. Halvorson et al. [21] identify knowledge transfer management during turnover as one of the primary justifications for a comprehensive integrated system model, and the Silesian Phoenix experience corroborates this directly.

Pedagogical outcomes: System thinking and professional competency

The MBSE implementation produced learning effects aligned with established frameworks for engineering competency development. Participants consistently identified systems thinking — the capacity to reason above the component level, trace how interface choices propagate through the architecture, and anticipate downstream implications of requirement changes — as a qualitative shift in their engineering perspective. Ghannam and Chan [18] report that students engaging with SE methodologies in authentic project contexts develop more sophisticated mental models than peers taught through conventional instruction, and Picard et al. [17] demonstrate that cross-functional coordination, structured communication, and systematic planning — precisely the competencies MBSE practice demands — are most robustly developed through extended team-based projects. The competition context amplified these effects, consistent with Talmi et al.'s [20] analysis of Formula Student: scoring criteria and expert evaluation gave requirements concrete, consequential meaning — an ambiguous requirement was a practical competitive risk — motivating sustained engagement with formal methodology that is difficult to replicate in purely academic settings.

Barriers to MBSE adoption and mitigation strategies

The most consistently reported barrier was the initial learning investment: approximately sixty hours of training in SE concepts and SysML modelling were required before productive independent work became possible, generating visible frustration among team members whose prior experience emphasised hands-on fabrication. Vogelsang et al. [5] identify the inability to demonstrate short-term ROI as a primary driver of resistance, and the student project context amplifies this dynamic. The mitigation strategy — explicit leadership commitment, communication that the upfront investment would prevent costlier downstream rework, and early celebration of instances where formal modelling prevented concrete errors — proved effective over the first four to six months.

Tool selection presented a secondary barrier. The initial trial of Papyrus generated significant friction due to interface complexity and limited SysML v2 documentation; the transition to Magic Systems of Systems Architect Community Edition reduced this friction, though it introduced dependency on a proprietary platform with licensing costs that may limit replicability — consistent with Ma et al.'s [7] findings on tool-chain fragmentation. An experienced systems engineering advisor emerged as a critical enabler: David et al. [24] and Li et al. [25] independently report that MBSE training effectiveness depends on expert facilitation during the early phases when students are constructing mental models of how modelling activity connects to design decisions, a finding with direct implications for institutions considering MBSE integration without prior investment in faculty SE capability.

Positioning with priori research and unique contributions

The findings are broadly consistent with prior MBSE case studies and educational research. Requirements management benefits align with Madni and Purohit's [10] economic analysis and Cole et al.'s [6] TALOS lessons; adoption barriers match those reported by Vogelsang et al. [5] in industry and by David et al. [24] and Bajczi et al. [13] in education; and the pedagogical benefits are consistent with competency development literature on competition-based PBL [17, 20].

Three distinctive contributions distinguish the present study. First, it provides longitudinal evidence spanning a full 24-month cycle from a European student engineering context largely absent from the literature — in contrast to the closest comparable studies, Halvorson et al. [21] and Li et al. [22], which originate from North

American institutions. Second, by integrating technical outcome analysis with pedagogical observation within a single case, it offers a holistic account of MBSE's dual role as engineering methodology and pedagogical framework. Third, by explicitly separating MBSE-attributable effects from those arising from team maturation, it contributes a more nuanced causal account than prior single-cycle case studies have offered.

Limitations and generalisability

The structural variability inherent in student engineering projects — fluctuating team composition, variable individual experience, evolving competition requirements, and episodic project cycles — means that observed outcomes cannot be straightforwardly attributed to MBSE adoption alone, nor assumed to replicate without modification in different environments. The reference baseline was conducted by a partially different team without the accumulated organisational learning the present team had developed, introducing confounding factors that cannot be disentangled through available data. Student engineering projects occupy a distinctive position between educational exercise and professional practice, requiring any applied methodology to accommodate team turnover, partially restarting learning curves, and the dual objective of producing a functional artefact while developing student competencies.

The study's contribution is therefore not a demonstration of specific, reproducible quantitative MBSE outcomes in student projects generally. Rather, it provides an in-depth account of how MBSE principles were adapted to this distinctive context, what challenges arose, and what forms of value emerged — constituting what Yin [8] terms an analytic generalisation that can inform subsequent empirical investigations across different institutions, competition formats, and team configurations.

CONCLUSIONS

This study reports the outcomes of a deliberate academic experiment: the structured introduction of MBSE methodology into a student engineering project operating under authentic competitive conditions. The results presented — improvements in requirements clarity, earlier detection of interface incompatibilities, reduced integration-phase rework, and the development of systems thinking competencies — are observed effects within a specific, bounded context, not universal outcomes replicable across all student project environments, nor statistically validated evidence of MBSE's causal superiority. The student project setting introduces variability — team composition turnover and cumulative organisational learning across cycles — that prevents straightforward causal attribution, and differences in team maturity and external conditions between the two compared cycles represent confounding factors acknowledged to ensure findings are interpreted at the appropriate level of evidential strength.

With respect to requirements management, the adoption of structured requirement statement templates aligned with ISO/IEC/IEEE 29148 [14] produced a measurable reduction in requirement ambiguity. The mechanism driving this improvement — the forced articulation of verifiable acceptance criteria — is independent of team-specific factors and represents a structural MBSE benefit theoretically expected to transfer across comparable contexts. Bidirectional traceability linking mission objectives through subsystem and component requirements to verification evidence was achieved at a substantially higher coverage level than in the reference cycle, enabling systematic change impact analysis throughout the design phase.

With respect to design integration, explicit interface specification within the SysML architectural model resulted in a substantial proportion of interface incompatibilities being identified during design reviews rather than during physical integration — one of the most directly MBSE-attributable findings of the study. The practical consequence of reduced late-stage design corrections was observed, though it also reflects team coordination improvements and accumulated manufacturing experience that cannot be fully isolated from the methodological effect.

With respect to pedagogical outcomes, participant accounts and the researcher's embedded observations indicate that sustained MBSE engagement developed systems thinking capabilities and requirements engineering competency among team members — competencies increasingly expected in professional engineering roles [5, 23] and difficult to develop through conventional coursework alone. This qualitative evidence is corroborated by established findings in the PBL literature [17, 20] and is theoretically well-grounded.

Regarding adaptation (RQ1): a pragmatic, phased approach using a tailored SysML v2 profile and a community-edition modelling tool proved sufficient for meaningful MBSE implementation. The key decisions — simplified modelling profile, version control integration, and phased introduction — are specific to the student context and may not transfer to industrial settings requiring higher process rigour.

Regarding benefits (RQ2): quantifiable improvements in requirements clarity and interface integration quality were observed, broadly consistent with industrial MBSE adoption studies [10, 6]. Their magnitude will vary with team maturity, institutional support, and system complexity; the findings should be understood as evidence of feasibility and directional consistency, not precise effect-size estimates.

Regarding barriers (RQ3): the primary challenges — a training investment exceeding forty hours, tool usability difficulties, and resistance from team members who perceived formal processes as overhead — are consistent with barriers reported in both industrial [5] and educational [13, 24] MBSE adoption studies. This cross-context convergence suggests these barriers reflect structural properties of MBSE adoption rather than artefacts of the specific case.

Regarding competency development (RQ4): structured MBSE practice within an authentic competition project provides effective conditions for developing professional SE competencies. The combination of genuine engineering stakes, extended project duration, and formal methodology appears to produce deeper competency development than either element alone, consistent with theory on authentic learning environments [18, 20].

For educators and curriculum designers, MBSE integration into competition-based student projects is both feasible and educationally valuable, provided three enabling conditions are met: faculty or mentor SE capability; sufficient project duration to recoup the initial learning investment; and explicit institutional legitimation of the formal process overhead MBSE imposes in early phases. Without these conditions, adoption is likely to remain superficial or be abandoned under schedule pressure. For student project leaders, the key lesson is that MBSE value is realised cumulatively and non-linearly: early months will feel slower than informal approaches, but the benefits — in design coherence, integration quality, and team communication — compound as the model matures. This requires leadership patience and willingness to invest in process discipline at the expense of short-term output velocity, a cultural shift that may be the single most important determinant of successful adoption.

This study is subject to several limitations. The single-case design precludes cross-case comparative analysis and limits statistical generalisability. The case is embedded in a specific institutional context — a Polish technical university with SE mentoring access and a competitive student association culture — that may not represent student project environments more broadly. Quantitative indicators are estimative rather than precisely measured, the comparison baseline differs in multiple confounding dimensions, and the dual researcher role as SE advisor and investigator introduces observer bias partially mitigated through triangulation.

Generalisation is therefore analytic rather than statistical [23], refining theoretical propositions about MBSE adoption dynamics and pedagogical effects in student engineering contexts. The findings are most likely to transfer to contexts sharing these structural characteristics: multidisciplinary student teams, moderate-to-high system complexity, project duration exceeding twelve months, SE mentoring availability, and an authentic external evaluation framework. Three future research directions would substantially advance understanding: comparative multi-case studies across European institutions and competition formats; longitudinal tracking of MBSE-engaged students to evaluate competency persistence into professional roles; and controlled pedagogical experiments comparing MBSE-integrated and conventional PBL approaches within the same institutional setting.

The Silesian Phoenix case (2023–2025) demonstrates that while MBSE requires deliberate upfront investment, the returns — in project quality and professional competency development — justify this investment in student projects of comparable scope. More importantly, it demonstrates that the student project context provides a distinctive environment in which MBSE's dual technical and pedagogical value can be simultaneously realised. As engineering education increasingly recognises systems thinking as a core graduate competency, the deliberate integration of MBSE into authentic project-based learning represents a promising mainstream pedagogical investment.

Acknowledgements

The authors gratefully acknowledge the Silesian Phoenix student scientific association and SUT for enabling this research and providing access to project artefacts and team members. Special thanks to the external systems engineering consultant who provided mentoring and training. Faculty advisors and team leadership demonstrated exceptional commitment to MBSE adoption despite competing priorities. Finally, we thank the more than forty student team members who engaged seriously with systematic methodologies while balancing the demands of a complex engineering project.

REFERENCES

1. Kossiakoff A, Sweet WN, Seymour SJ, Biemer SM. Systems Engineering: Principles and Practice. 2nd ed. Hoboken, NJ: John Wiley & Sons; 2011.
2. Ramos AL, Ferreira JV, Barceló J. Model-based systems engineering: An emerging approach for modern systems. IEEE Trans Syst Man Cybern C Appl Rev. 2011;42(1):101–111. https://doi:10.1109/TSMCC.2011.2106495
3. International Council on Systems Engineering (INCOSE). INCOSE Systems Engineering Handbook. Hoboken, NJ: John Wiley & Sons; 2023.
4. Uckun S, Kurtoglu T, Bunus P, Tumer I, Hoyle C, Musliner D. Model-based systems engineering for the design and development of complex aerospace systems. SAE Technical Paper; 2011. https://doi:10.4271/2011-01-2664
5. Vogelsang A, Amorim T, Pudlitz F, Gersing P, Philipps J. Should I stay or should I go? On forces that drive and prevent MBSE adoption in the embedded systems industry. arXiv preprint arXiv:1709.00266; 2017. https://doi:10.48550/arXiv.1709.00266
6. Cole BF, Mittal V, Gillespie SE, La N, Wise R, MacCalman AD. Model-based systems engineering: application and lessons from a technology maturation project. Procedia Comput Sci. 2019;153:202–209. https://doi:10.1016/j.procs.2019.05.071
7. Ma J, Wang G, Lu J, Vangheluwe H, Kiritsis D, Yan Y. Systematic literature review of MBSE tool-chains. Appl Sci. 2022;12(7):3431. https://doi:10.3390/app12073431
8. Yin RK. Case Study Research and Applications: Design and Methods. Thousand Oaks, CA: SAGE Publications; 2018.
9. Eisenhardt KM. Building theories from case study research. Acad Manage Rev. 1989;14(4):532–550. https://doi:10.2307/258557
10. Madni AM, Purohit S. Economic analysis of model-based systems engineering. Systems. 2019;7(1):12. https://doi:10.3390/systems7010012.
11. Siddique A. MBSE implementation in small satellite systems: rationale for adoption over traditional document-based systems engineering. Int J Geomatics Sci Technol. 2025;1(1). https://doi:10.46610/ijgst.2025.v01i01.001
12. Object Management Group (OMG). OMG Systems Modeling Language (SysML). Available from: https://www.omg.org/spec/SysML Accessed 2026 Feb 12.
13. Bajczi L, Szekeres D, Siegl D, Molnár V. Enhancing MBSE education with version control and automated feedback. arXiv preprint arXiv:2409.15294; 2024. https://doi:10.48550/arXiv.2409.15294
14. International Organization for Standardization. Systems and software engineering—Life cycle processes—Requirements engineering (ISO/IEC/IEEE 29148:2018(E)); 2018.
15. Wheaton JS, Herber DR. Digital requirements engineering with an INCOSE-derived SysML meta-model. arXiv preprint arXiv:2410.21288; 2024. https://doi:10.48550/arXiv.2410.21288
16. Hmelo-Silver CE. Problem-based learning: What and how do students learn? Educ Psychol Rev. 2004;16(3):235–266. https://doi:10.1023/B:EDPR.0000034022.16470.f3
17. Picard C, Hardebolle C, Tormey R, Schiffmann J. Which professional skills do students learn in engineering team-based projects? Eur J Eng Educ. 2021;47(2):258–278. https://doi:10.1080/03043797.2021.1920890
18. Ghannam R, Chan C. Teaching undergraduate students to think like real-world systems engineers: a technology-based hybrid learning approach. Syst Eng. 2021;25(1):14–27. https://doi:10.1002/sys.21683
19. Lemu HG. On competition-driven teaching of multidisciplinary engineering education: Implementation cases at University of Stavanger. Nordic J STEM Educ. 2017;1(1):278–286. https://doi:10.5324/njse.v1i1.2340
20. Talmi I, Hazzan O, Katz R. Intrinsic motivation and 21st-century skills in an undergraduate engineering project: the Formula Student project. High Educ Stud. 2018;8(4):46–58. https://doi:10.5539/hes.v8n4p46
21. Halvorson M, Fuchs J, Kung P, Thomas DL. A model-based systems engineering approach to space mission education of a geographically disperse student workforce. 4th Symposium on Space Educational Activities; 2022. https://doi:10.5821/conference-9788419184405.042

22. Li H, Yu M, Yuan S, et al. Exploring a project-based training model for engineering undergraduates driven by model-based systems engineering. Adv Educ Technol Psychol. 2025;9(6). https://doi:10.23977/aetp.2025.090619
23. Easterbrook S, Singer J, Storey M-A, Damian D. Selecting empirical methods for software engineering research. In: Shull F, Singer J, Sjøberg DIK, editors. Guide to Advanced Empirical Software Engineering. London: Springer; 2008. p. 285–311.
24. David P, Blanco E, Revol S, Noyrit F, Coatrine M. Model based systems engineering introduction within industrial engineering curriculum. Proc 21st Int Conf Eng Prod Des Educ (E&PDE); 2019. https://doi:10.35199/epde2019.70
25. Li J, Douglas K, Le H, Raz A, Huang W, Fentiman A. Applying social constructivism in model-based systems engineering online instructional module for engineering professionals. Proc IEEE Front Educ Conf (FIE); 2020. https://doi:10.1109/FIE44824.2020.9274251