情報処理学会論文誌 Vol.0 No.0 1–9 (??? 1959)

LLM活用を前提とした個人自由制作型PBLにおける成果物規模と品質の年度間比較

片岡 葉奈[1,a)] 李 家隆[2,b)] 依田 みなみ[1,c)] 松野 裕[1,d)]

受付日 xxxx年0月xx日, 採録日 xxxx年0月xx日

概要：大規模言語モデル（LLM）の普及により，学生が生成 AI を用いながら課題設定，設計，実装，テストを進める状況が一般化しつつある．本研究では，大学 3 年次のソフトウェア工学科目における個人自由制作型 PBL を対象に，2024 年度と 2025 年度の成果物を比較し，成果物の機能規模と品質関連指標にどのような年度間差が観察されるかを分析した．機能規模分析では，研究利用に同意が得られた 2024 年度 48 名，2025 年度 46 名の成果物を対象とし，COSMIC 法に基づく LLM 補助測定を行った．また，2025 年度については，CFRP に基づく基礎的プログラミング習熟度と，共通プロンプトにより作成したテストケースの通過率との関連を探索的に分析した．分析の結果，2025 年度の平均機能規模は 2024 年度の約 3.2 倍であり，LLM 利用環境の改善と矛盾しない規模差が認められた．一方で，CFRP には天井効果が認められ，習熟度と成果物規模の相関は弱かった．これに対し，習熟度とテスト通過率の間には中程度の正の相関が見られた．本稿では，LLM を機能数測定およびテスト作成にも用いている点を踏まえ，LLM 活用を前提とした個人自由制作型 PBL における成果物評価の課題を議論する．

キーワード：ソフトウェア工学教育，大規模言語モデル，個人自由制作型 PBL，成果物評価，実証研究

Year-over-Year Comparison of Artifact Scale and Quality in LLM-Augmented Open-Ended Individual Project-Based Learning

Hana Kataoka[1,a)] Jialong Li[2,b)] Minami Yoda[1,c)] Yutaka Matsuno[1,d)]

Received: xx xx, xxxx, Accepted: xx xx, xxxx

Abstract: The widespread availability of Large Language Models (LLMs) is changing how students conduct software development exercises. This paper compares artifacts produced in an open-ended individual project-based learning (PBL) activity in a third-year undergraduate software engineering course between 2024 and 2025. For functional size analysis, we examined artifacts from 48 consenting students in 2024 and 46 in 2025, using LLM-assisted measurement based on the COSMIC method. For the 2025 cohort, we further explored the relationships among basic programming proficiency measured by CFRP, functional size, and test-case pass rates based on a common prompt scaffold. The average functional size in 2025 was approximately 3.2 times that in 2024, which is consistent with the improved LLM environment. However, CFRP scores showed a ceiling effect, and their correlation with functional size was weak, while their correlation with test-case pass rates was moderate. Because this study also uses LLMs for functional size measurement and test generation, the findings should be interpreted with these measurement limitations in mind. We discuss the remaining challenges of artifact assessment in open-ended individual PBL settings where LLM use is assumed.

Keywords: Software Engineering Education, Large Language Models, Open-Ended Individual Project-Based Learning, Artifact Assessment, Empirical Study

[1] 日本大学理工学部
[2] 早稲田大学高等研究所
[a)] cshn24004@g.nihon-u.ac.jp
[b)] Corresponding_Author:lijialong@fuji.waseda.jp
[c)] yoda.minami@nihon-u.ac.jp
[d)] matsuno.yutaka@nihon-u.ac.jp

1. はじめに

近年，大規模言語モデル（LLM）の性能向上により，ソフトウェア開発における支援ツールの利用形態は変化している [1]．コード生成やデバッグ支援，アーキテクチャ設計の補助など，LLM は開発実務で広く利用されつつあり，教育現場においてもこれらを統合的に活用する能力（LLM リテラシー）の育成が求められている [2], [3]．

こうした技術的変化の中で，ソフトウェア工学（SE）教育における PBL の設計と評価も再検討が必要となっている．本稿では，学生が個人単位で自ら課題を設定し，要求定義，設計，実装，テストまでを一通り経験する形式を「個人自由制作型 PBL」と呼ぶ．このような PBL は，内発的動機づけや自律性の促進に有効である一方 [4]，成果物の規模，品質，開発過程が学生の裁量に強く依存する．LLM 活用が許容または推奨される状況では，学生自身の基礎スキル，LLM への依存度，要求の分解能力，生成物の検証能力が複合的に関係するため，成果物から解釈可能な範囲を慎重に整理する必要がある．従来の初学者向けプログラミング教育 [5], [6], [7] や，特定のタスクに限定した LLM 教育活用 [8], [9], [10] は進展している．一方で，学生が個別にテーマを設定する個人自由制作型 PBL において，成果物の規模と品質を年度間で比較し，さらに評価指標そのものの制約を明示した研究は十分ではない．さらに，LLM 技術は更新の周期が短く，学生が無料または学生向け特典で利用できるモデルも年度によって変化する．そのため，単一年度の実践報告だけでなく，同一科目内での年度間差を把握し，その解釈範囲を明確にすることが教育設計上の課題となる．

本研究では，日本大学の 3 年次 SE 科目における個人自由制作型 PBL を対象として，2024 年度から 2025 年度にわたる 2 年間のデータを分析した．本稿は，LLM 利用環境の変化だけを因果要因として分離するものではない．授業内容の一部変更，学生集団差，研究同意によるサンプル選択の影響を完全には排除できないため，観察された年度間差を，LLM 利用環境の改善と同時期に生じた現象として慎重に解釈する．具体的には，以下 3 つの RQ を設定する．

- **RQ1：成果物規模の年度間差**個人自由制作型 PBL において，2024 年度と 2025 年度の成果物の機能規模にはどのような差が観察されるか．
- **RQ2：基礎的習熟度と成果物指標の関係** 2025 年度において，CFRP で測定した基礎的プログラミング習熟度は，成果物の機能規模およびテスト通過率とどのような関連を持つか．
- **RQ3：LLM 補助評価の妥当性上の制約** LLM を用いて機能規模測定およびテストケース作成を行う場合，成果物評価の解釈にはどのような制約があるか．

本研究の主な貢献は以下の 2 点である．

- 同一科目内で実施された個人自由制作型 PBL について，2 年間の成果物規模を比較し，LLM 利用環境が変化する時期に観察された年度間差を定量的に整理した．
- CFRP，機能規模，テスト通過率を組み合わせた探索的分析を行い，LLM 活用を前提とする成果物評価において，どの指標が何を示し，どの点に限界があるかを明示した．

本研究の初期成果は，[11] にて報告している．本論文ではこれを以下の点において拡張した．第一に，年度ごとの授業構成の相違や演習プロセスなど，実験設計と実施の詳細を追加し，研究の透明性を高めた．第二に，CFRP で測定した基礎的習熟度と成果物指標の関係を探索的に分析した．第三に，LLM 補助による機能規模測定およびテスト作成を用いる際の妥当性上の制約を整理し，結果の解釈範囲を明確化した．

本論文の構成は以下の通りである．まず第 2 章では，情報科学教育における LLM 活用の最新動向と課題，およびソフトウェア開発を伴う PBL における評価指標について概観する．第 3 章では，2 年間にわたる観察研究の枠組みを述べ，3 つの評価指標を用いた実験設計について詳述する．第 4 章では，収集データの統計的分析を行い，3 つの RQ について議論する．最後に第 5 章で本研究を総括し，今後の展望を述べる．

2. 関連研究

本章では，情報科学教育における LLM 活用に関する研究と，SE 分野の PBL における評価指標を紹介する．

2.1 情報科学教育における LLM 活用の潮流と課題

LLM の普及に伴い，情報科学教育における指導方法や評価方法の見直しが進んでいる．近年の包括的なレビュー [3] によれば，この分野の研究は急速に増加しているものの，その多くは入門プログラミング（CS1/CS2）におけるコード生成やデバッグ支援，あるいは要求生成といった特定タスクにおける取り組みに集中している．

初期の研究では，LLM が高い確率で基礎的なプログラミング課題を解く能力を持つことが示されてきた [5], [8]．しかし，実務的な開発環境に近づくほど，LLM の出力には「動作はするが設計原則に反する」という課題も報告されている．例えば，オブジェクト指向の設計課題において，LLM は抽象化を欠いたコードや不適切な継承を生成する傾向があり [9]，これは LLM 活用環境下においても，学生自身の設計スキルや検証能力が依然として重要であることを示唆している．

また，LLM が学習成果に与える影響については，相反する知見が報告されている．開発効率の向上や学習時の心理的負担の低下が確認される一方で [7], [10]，学生が公式資料

を参照しなくなる「LLM 依存」のリスクや，適切な誘導ができないことによる学習の逸脱も報告されている [6], [12]．このように，LLM は学習支援として有用である一方，利用方法や教育設計によっては，学生が解決過程を十分に経験しないまま成果物を得る可能性がある．

さらに，LLM を使いこなすための能力は「プロンプトエンジニアリング」を超え，出力の妥当性を評価する「批判的推論」を含む新たなスキルとして概念化されつつある [13]．これを体系的に教育しなければ，元々の習熟度が高い学生とそうでない学生の間で「LLM 活用の格差」が生じ，新たな学習上の差となる懸念も指摘されている [13], [14], [15]．

近年は，LLM をプログラミング教育に組み込む具体的な授業設計も提案されている．例えば，CS50 における AI 支援の導入 [16]，AI 生成コードの誤り訂正を含む学生の認識の分析 [17]，ChatGPT 利用時の問題解決行動の分析 [18]，LLM をデバッグの teachable agent として用いる教育設計 [19]，および教室配備型の LLM 支援ツールの評価 [20] などである．また，LLM が初学者と熟練者に異なる影響を与えることや，学習成果・成果物品質に与える影響についても報告されている [15], [21], [22]．

これらの研究は，LLM がプログラミング教育に与える効果とリスクを明らかにしている．これに対し，本研究は，学生が個別にテーマを設定する個人自由制作型 PBL を対象とし，年度間で成果物規模がどのように異なるか，また CFRP，機能規模，テスト通過率という複数の指標がどの範囲で成果物評価に利用できるかを検討する点に焦点を置く．本研究はチーム PBL 一般の効果を論じるものではなく，個人自由制作型 PBL における成果物評価の観察的研究として位置付ける．

2.2 ソフトウェア開発を伴う PBL における評価指標

学生の成果や能力を定量化する試みは，評価の対象に応じて大きく「プロダクト」「プロセス」「学生の能力」の 3 つの観点に分類できる．

第一に，成果物（プロダクト）の規模や品質を測る指標である．古くは LOC（行数）が用いられてきたが，言語依存性を排除するため，データ移動量に着目した COSMIC 法 [23] のような機能規模測定手法が国際標準として活用されている．また，コードの論理的複雑さを示す循環的複雑度 [24] といった静的解析指標は，単なる完成度だけでなく，ソフトウェア工学的に適切な構造で実装されているかを評価する客観的な尺度となる．

第二に，開発の過程（プロセス）や行動に着目した指標である．ラーニングアナリティクスの発展により，エラーの頻度や提出回数，コーディング過程のスナップショット分析などを通じて，学生の試行錯誤を可視化する研究が進んでいる [25]．これにより，最終成果物だけでは見えない学生の躓きや，共同作業における貢献度を動的に把握することが可能となっている．

第三に，学生自身のスキルや心理的変容を測る指標である．プログラミング能力の直接的な評価には，国内の CFRP[26] のように，言語に依存しない標準的なルーブリックが提案されている．これに加え，プログラミング自己効力感や学習動機といった情意面の評価も，教育介入の効果を多角的に検証する上で重要な役割を果たしている．

3. 実験設計

3.1 授業概要

本研究は，日本大学理工学部応用情報工学科の 3 年次春学期選択科目「ソフトウェア工学」（全 15 週）を対象として実施した．第 1-7 週では，要件定義，設計，実装，テストからなるソフトウェア工学の基礎を講義形式で教授し，第 8 週に中間試験，第 15 週に期末試験を行った．本演習の設計，実施，および成果物の評価は，ソフトウェア工学教育に 15 年以上の経験を持つ同一の教員が担当しており，2 年間を通じて一貫した水準を維持している．

第 9 週以降の演習プロセスにおいては，年度間で構成および実施順序の最適化を図った（表 1）．2024 年度は第 9-10 週にテスト手法演習を集中的に行った後，第 11-14 週の 4 週間で個人自由制作型 PBL を実施した．対して 2025 年度は，開発とテストのサイクルをより密接に関連付けるため，第 10 週から個人自由制作型 PBL を開始し，テスト手法演習（第 9 週・第 11 週）を並行して進める構成とした．また，講義内容の重点も技術動向に合わせて調整している．2024 年度は UML モデリングと LLM による設計評価に重点を置いたのに対し，2025 年度は Git/GitHub の活用および LLM を用いたより実践的な開発体験に重点を置いた教育設計となっている．

表 1 年度間の授業シラバスの比較

週	2024 年度	2025 年度
6	状態遷移図	オブジェクト指向
7	オブジェクト指向／生成 AI を用いたプログラミング試行	Git の使い方／生成 AI を用いた Web ページ作成
9	テスト手法演習	テスト手法演習
10	テスト手法演習	個人自由制作型 PBL 1 回目
11	個人自由制作型 PBL 1 回目	テスト手法演習
12-14	個人自由制作型 PBL 2-4 回目	個人自由制作型 PBL 2-4 回目

個人自由制作型 PBL は 4 週間にわたって実施され，各年度とも，学生自身が日常生活の課題からテーマを設定し，最終成果物として要件定義書，各種設計図，ソースコード，およびテスト結果報告書の提出を義務付けた．本演習は個人単位で実施されており，本稿ではチーム開発やチーム内

コミュニケーションの効果は分析対象としない．本演習ではテーマ選定から最終的な品質検証に至る全プロセスにおいて，LLM をアイデア検討や設計議論を支援するツールとして積極的に活用することを推奨した．ただし，LLM 利用そのものを成績評価上の必須条件とはしなかったため，学生ごとの利用頻度や利用局面には差がある可能性がある．この点は，本研究が LLM 利用の因果効果ではなく，LLM 活用が許容・推奨された授業環境下での成果物指標を分析する観察研究であることを意味する．各週の具体的な活動として，第 1 週（要件定義）では，抽出した課題を解決するアプリケーションの構想を Miro を用いて視覚化した．第 2 週（モデル作成）では，Jacobson のオブジェクト指向開発論に基づき，ユースケース図，クラス図，シーケンス図，状態遷移図を作成した．その際，LLM にアプリケーションの概要を入力し，Mermaid 記法で各種ダイアグラムを生成させる手法を導入した．第 3 週（開発とテスト）では，作成したモデルを基に LLM で初期コードを生成し，動作確認とデバッグを反復した．LLM の提案のみで解決しない不具合については，学生自身による手動のコード修正を求めた．最終週となる第 4 週（最終調整と提出）では，開発したソースコード一式（GitHub リポジトリ），動作デモを記録した YouTube 動画，および任意で LLM へのプロンプト記録を提出させた．

3.2 年度間における LLM 利用環境の差異

本演習では特定の LLM を有料ライセンスとして配布する形式は取らなかったが，教育上の公平性を担保するため，学生には Gemini の学生向け無料プランや Google AI Studio など，無料で利用可能な比較的高性能なモデルの活用を推奨した．2024 年度は利用モデルの個別調査を行わなかったが，2025 年度の調査では，ChatGPT（57 名），Gemini（55 名），Claude（13 名）などが併用されていた（複数回答可）．なお，この調査は利用したサービス名を尋ねたものであり，個々の学生が実際にどのモデルバージョンをどの開発局面でどの程度利用したかまでは測定していない．

2024 年度から 2025 年度にかけては，LLM の性能向上と利用可能性の改善が進んだ．2024 年 7 月時点では，GPT-4o や Claude 3.5 Sonnet といったモデルがリリースされていたものの，無料枠での利用は限定的であり，短時間の試行で旧世代モデルへと制限される状況があった．対して 2025 年 7 月時点では，Gemini 2.5 Pro[27]，OpenAI o3[28]，Claude 4[29] といった推論能力を強化したモデルが登場し，かつ Google AI Studio や Gemini の学生向けプラン等を通じて，比較的高性能なモデルを学生が利用しやすい環境が整っていた．ただし，本研究では各学生の実利用ログを取得していないため，以下の分析では「高性能モデルの利用可能性向上による効果」と断定せず，LLM 利用環境の改善と同時期に観察された年度間差として扱う．

3.3 評価指標

表 2 年度間の条件比較

項目	2024 年度	2025 年度
履修者数	109 名	106 名
分析対象（機能数）	48 名	46 名
機能数測定	○	○
習熟度評価	−	○
品質評価	−	○

表 3 本研究における分析単位と有効サンプル数

分析	年度	有効 n	備考
機能規模	2024	48	研究同意が得られた成果物
機能規模	2025	46	研究同意が得られた成果物
CFRP 前後変化	2025	59	前後 2 回の対応が取れた学生
テスト通過率	2025	54	共通プロンプトによるテスト結果報告あり

成果物の分析に「機能数（規模）」「プログラミング習熟度」「品質関連指標」の 3 つの指標を採用した．表 2 に示す通り，2024 年度は機能数のみを測定したが，2025 年度からは学生の基礎スキルや成果物の動作確認結果との関係を探索的に検証するため，指標を拡充した．各分析の有効サンプル数は表 3 に示す．本研究において，一般的な評価指標の中からこれら 3 点を選択した根拠と制約は以下の通りである．

第一に，「機能数（規模）」の測定に COSMIC 法 [23] を採用した．理由としては，個人自由制作型 PBL における言語の多様性に対応し，実装言語や LOC に依存しない規模指標を得るためである．従来の LOC（行数）はプログラミング言語や LLM の生成スタイル（冗長なコメントやコード記述等）に大きく左右されるため，年度間比較には適さない．COSMIC 法は，実装技術から独立し，利用者とソフトウェア間のデータ移動に基づいて機能規模を測定するため，本研究では成果物規模の比較指標として採用した．ただし，COSMIC 法は実装された機能規模を測る方法であり，その機能を学生自身が設計したのか，LLM が提案したのかを区別する方法ではない．したがって，本研究における機能数は「学生と LLM を含む開発環境のもとで最終成果物として実現された規模」を表す指標であり，学生単独の設計能力や実装能力を直接測定するものではない．機能数の測定は，学生が提出したソースコード一式を LLM に入力し，COSMIC 法の測定ルールを記述した共通プロンプトに基づいて算出した．この手順により多数の多

様な成果物を同一基準で処理できる一方，LLM による測定誤差や再現性の問題は残る．そのため，本稿では機能数を完全に客観的な値ではなく，LLM 補助測定による近似的な成果物規模指標として扱う．

第二に，基礎的プログラミング習熟度の指標として CFRP[26] を用いた．CFRP は，変数，条件分岐，配列，関数，オブジェクト指向などの言語非依存な基礎概念を測る指標である．ただし，本演習は基礎プログラミング訓練ではなく，既に基礎科目を履修した学生を対象に，要求定義からテストまでを一巡する SE 開発演習として設計されている．したがって，CFRP の前後変化は本演習の主たる教育効果を直接測るものではなく，成果物指標を解釈するための背景変数として位置付ける．特に，対象学生には高得点層が多く，天井効果が生じ得るため，相関分析の解釈には注意が必要である．

表 4 CFRP レベル判定基準

レベル	スコア範囲	習得内容
Lv1	0.0–2.0 点	簡単な演算，変数，条件分岐
Lv2	2.1–4.0 点	文字列操作，配列，繰り返し
Lv3	4.1–6.0 点	データ型，乱数，関数
Lv4	6.1–8.0 点	定数，複雑な制御構造，配列操作
Lv5	8.1–10.0 点	例外処理，多次元配列，ソート
Lv6	10.1–12.0 点	並行処理，オブジェクト指向

第三に，「開発規模と動作確認結果の乖離」を把握するため，品質関連指標としてテストケース通過率を採用した．先行研究で指摘されている通り，LLM は構文的に正しく動作しても，不適切な設計や例外処理の欠如を含むコードを生成しやすい．そのため，単なる機能の実装数（規模）だけでなく，正常系，境界値，異常系の観点から作成されたテストケースに対する通過率を導入した．ただし，本研究のテスト通過率は，最終成果物の品質を完全に代表するものではなく，共通プロンプトに基づく限定的な動作確認指標である．

品質関連指標のばらつきを抑えるため，本演習では全学生に対し，共通の「テスト設計プロンプト・スキャフォールディング」を提供した．図 1 に示す通り，このプロンプトには SHIFT 社の「良質テスト 3 視点」[30] を組み込んでおり，正常系・境界値・異常系の各観点からテストケース（計 24 件）が生成されるよう設計されている．これにより，少なくともテスト観点と件数を揃えたうえで，生成されたテストに対する実行結果の通過率を比較指標とした．一方で，LLM が生成した各テストケースの妥当性や期待値の正確性については，人手による完全な再評価を実施していない．そのため，RQ3 ではこの評価方法自体の限界を明示的に議論する．

```
あなたは大学 3 年生向け実習のテスト設計アシスタントです．私はこの後”CODE”タグで囲ったプログラムを提示します．（複数プログラムがある場合は”CODE”〜”/CODE”を続けて複数挿入して構いません）

【SHIFT 良質テスト 3 視点：参考】
1. 正常系：代表的・想定どおりの入力で正常に完了するか
2. 境界入力：0，最大値，空，NULL など端や例外的な入力でも壊れないか
3. 異常系：想定外の障害や例外時に安全に失敗できるか

【タスク】
1. プログラムの単体テストケースを合計 24 件作成してください
   ・目安配分：正常系 8 件，境界入力 8 件，異常系 8 件
2. Markdown 表のみを出力してください．表以外の説明・コードは含めないでください
3. 表の列は次にしてください：
   — テスト対象 — テスト観点 — テスト条件 — テスト手順 (1 行) — 期待値 (1 行) — 結果 (〇/× ) —
```

図 1 均質なテストケース生成を目的として学生に提供したプロンプト・スキャフォールディング

3.4 分析対象

本研究の分析対象は，2024 年度（109 名）および 2025 年度（106 名）の履修登録者のうち，研究へのデータ利用に同意が得られた学生である．本演習では，成績評価への影響を排除し，公平な学習環境を確保する観点から，本来の講義内容と関係性が薄いテストの受験や研究協力はすべて学生の任意としている．

具体的には，機能規模分析では，成績評価のために学生全員に成果物を提出してもらったが，本研究ではそのうちデータ利用の同意が得られた計 94 名（2024 年度 48 名，2025 年度 46 名）を分析対象とした．プログラミング習熟度分析では，2025 年度の前後 2 回の CFRP テストをいずれも受験した 59 名を対象とした．テスト通過率の評価には，共通プロンプトによるテスト結果を報告した 54 名のデータを用いた．履修登録者数とサンプル数の乖離は，主に研究同意の有無や任意受験に起因するものであり，講義履修の完了率および成果物の提出率は両年度とも良好であった．

4. 実験結果および考察

本章では，2 年間にわたる観察データの分析結果を報告する．まず，両年度における成果物のジャンル分布を図 2 に示す．いずれの年度も「生活便利ツール」が最多（各 44%）を占めた一方で，特定のジャンル構成には変化が認められ

た．具体的には，2024 年度に 30%を占めていた「キャンパス生活関連」は，2025 年度には 8%へと減少した．対照的に，「学習サポート」（9%→14%）や「クリエイティブ・エンタメ」（3%→10%）は増加傾向を示した．ただし，個人自由制作型 PBL では年度ごとの学生の関心や流行によってテーマが変化し得るため，本稿ではジャンル分布の変化を LLM 利用環境の直接的効果とは解釈しない．以下では，成果物規模，基礎的習熟度との関係，評価方法の制約という 3 つの観点から分析結果を示す．

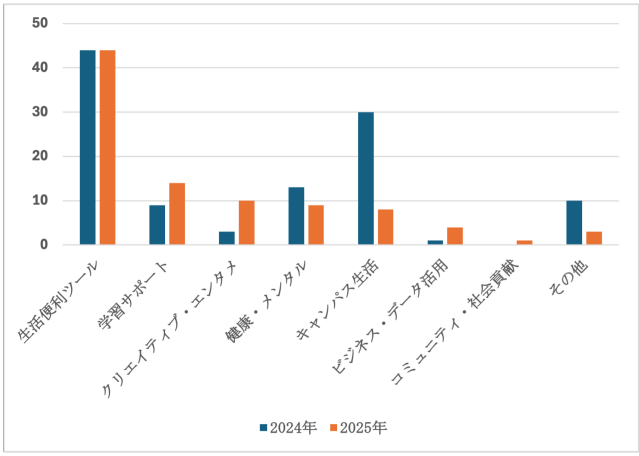

図 2　成果物のジャンル分布の年度間比較

4.1　RQ1：成果物規模の年度間差

本節では，2024 年度から 2025 年度にかけて，学生の成果物の機能規模にどのような年度間差が観察されたかを分析する．両年度の機能数（CFP）に関する記述統計量を表 5 に示す．

表 5　機能数の記述統計量

統計量	2024 年度	2025 年度	変化率
標本数	48	46	—
平均値	9.56	30.13	3.2 倍
中央値	8.00	29.50	3.7 倍
第 1 四分位	3.75	17.75	4.7 倍
第 3 四分位	13.00	41.50	3.2 倍
標準偏差	7.06	17.16	2.4 倍
変動係数	0.74	0.57	0.77 倍
最小値	2	3	1.5 倍
最大値	38	68	1.8 倍

2025 年度の平均値は 2024 年度と比較して 3.2 倍（30.13 CFP）に増加し，中央値も 3.7 倍に増加した．両年度では標準偏差が大きく異なるため，等分散を仮定しない Welch の t 検定を用いたところ，$t(59.3) = 7.54, p < 0.001$ であり，効果量も $d = 1.58$ と大きかった．この結果から，2025 年度の成果物は 2024 年度よりも大きな機能規模を持つ傾向が観察された．

分布の詳細を見ると，第 1 四分位（Q1）は 3.75 から 17.75 へと 4.7 倍に増加し，第 3 四分位（Q3）は 13.00 から 41.50 へと 3.2 倍に増加した．また，変動係数は 0.74 から 0.57 に低下しており，相対的なばらつきは小さくなった．一方で，標準偏差と最大値は増加しており，絶対的な規模差は拡大している．

以上から，RQ1 について，2025 年度の成果物は 2024 年度よりも機能規模が大きい傾向を示したと解釈できる．ただし，この年度間差は LLM 利用環境の改善と矛盾しないものの，授業構成の変更，学生集団差，個々の LLM 利用頻度の差を分離した因果効果としては解釈できない．

4.2　RQ2：習熟度と開発成果の関係

本節では，2025 年度のデータを用い，CFRP で測定した基礎的プログラミング習熟度と成果物指標との関連を探索的に分析する．まず，演習前後の CFRP レベルの遷移を表 6 に示す．

表 6　演習前後の CFRP レベル遷移（2025 年度，$n = 59$）

前＼後	Lv1	Lv2	Lv3	Lv4	Lv5	Lv6
Lv1	2	0	0	0	0	0
Lv2	1	1	0	0	0	1
Lv3	0	1	2	1	0	1
Lv4	0	2	1	3	1	2
Lv5	0	1	1	3	1	8
Lv6	0	0	0	3	2	21

前後対応の取れた 59 名について分析した結果，スコアが向上した学生は 14 名（23.7%），変化なしは 30 名（50.8%），低下した学生は 15 名（25.4%）であった．最高ランクである Lv6 の 26 名中 21 名（80.8%）が演習後も同レベルを維持しており，天井効果が認められた．また，対象学生の約 70%が演習前から Lv5 以上の高習熟度層に属していた．したがって，CFRP は本演習における基礎的背景能力の指標としては有用である一方，演習による習熟度変化を検出する指標としては感度に限界がある．

次に，CFRP と成果物の機能規模およびテスト通過率との相関を表 7 に示す．

表 7　CFRP と成果物指標の相関係数（2025 年度）

変数間	Pearson r	Spearman ρ
CFRP vs 機能数	0.16	0.16
CFRP vs テスト通過率	0.39**	0.33**
機能数 vs テスト通過率	0.09	0.04

**$p < 0.01$

分析の結果，CFRP と機能数の間には統計的に有意な相関が認められなかった（$r = 0.16, p = 0.29$）．この結果は，成果物の機能規模が，CFRP で測定される基礎的プログラミング概念の理解だけでは説明できないことを示して

いる．個人自由制作型 PBL では，テーマ選定，要求分解，LLM への指示，生成結果の統合など，CFRP が直接測定しない要因も成果物規模に影響すると考えられる．

一方で，CFRP とテスト通過率の間には中程度の正の相関が認められた（$r = 0.39, p < 0.01$）．ただし，本研究ではテストケース作成にも LLM を用いており，学生がどの程度コードを自力で読解し，どの程度 LLM に修正を依頼したかを示すプロセスログは取得していない．したがって，この相関を「基礎力が品質保証を直接可能にした」と強く解釈することはできない．より慎重には，CFRP で測定される基礎的習熟度が高い学生ほど，共通プロンプトに基づくテスト通過率が高い傾向を示した，と解釈する．

なお，機能数とテスト通過率の相関は小さかった（Pearson $r = 0.09$, Spearman $\rho = 0.04$）．この結果は，成果物の規模と，共通テストに対する通過率が異なる側面を表す指標であることを示唆する．

4.3 RQ3：LLM 補助評価の妥当性上の制約

本研究では，機能規模の測定とテストケース作成の双方に LLM を利用している．これは，多様な言語・フレームワークで作成された個人自由制作型 PBL の成果物を同一の手続きで分析するための現実的な方法である一方，結果の解釈には複数の制約がある．

第一に，LLM 補助による COSMIC 測定は，成果物の機能規模を効率的に推定するための手段であり，学生自身の設計意図や LLM の寄与を分離するものではない．したがって，RQ1 で示した機能数の年度間差は，学生単独の開発能力の差ではなく，授業環境，利用可能な LLM，学生のテーマ選定，実装支援ツールを含む開発環境全体の結果として解釈すべきである．

第二に，共通プロンプトによるテストケース作成は，正常系・境界値・異常系という観点を揃えるためには有効であるが，生成されたテストケースの妥当性を保証するものではない．特に，個人自由制作型 PBL の成果物では仕様が学生ごとに異なるため，LLM が生成した期待値が仕様と整合しているか，人間が確認する必要がある．本稿では，テスト通過率を品質そのものではなく，限定的な動作確認指標として扱う．

第三に，本研究では学生のプロンプト履歴や LLM との対話ログを体系的に取得していない．そのため，CFRP とテスト通過率の相関が，コード読解・デバッグ能力によるものなのか，LLM への質問の仕方によるものなのか，あるいはテーマの難易度差によるものなのかを分離できない．

以上から，RQ3 について，LLM 補助評価は個人自由制作型 PBL の成果物を比較するための実践的な方法である一方，機能数やテスト通過率を学生個人の能力指標として直接解釈するには限界があると整理できる．今後は，機能数測定の一部を人手で再測定して一致度を確認すること，テストケースの妥当性を人手で分類すること，さらにプロンプト履歴を用いて LLM 活用プロセスを分析することが必要である．

4.4 妥当性への脅威

本研究の妥当性に対する脅威として，内的妥当性，外的妥当性，構成概念妥当性について議論する．

第一に，2024 年度と 2025 年度の学生集団の完全な同質性は保証されていない．しかし，両年度とも同一の前提科目（プログラミング基礎，データ構造，アルゴリズム）の単位修得を必須としており，集団としての基礎能力のベースラインに極端な乖離はないと推測される．第二に，表 1 に示した授業構成の微調整（Git 導入の強化等）が，LLM 利用環境の変化以外の要因として成果物に寄与した可能性は否定できない．したがって，本研究は年度間差を LLM 性能向上の因果効果として主張するものではない．第三に，LLM 利用は推奨であり，利用ログを体系的に取得していないため，学生がどのフェーズでどのモデルをどの程度利用したかは十分に把握できていない．この点は，LLM 利用環境の改善と成果物規模の増加を結び付けて解釈する際の重要な制約である．

外的妥当性については，本研究は単一大学の情報系学科における 3 年次科目を対象としており，結果の一般化には限界がある．特に，本演習は「個人自由制作型 PBL」という高い裁量を伴う形式であり，初学者向けの指定課題型演習や長期にわたるチームプロジェクトでは，LLM の効果が異なる傾向を示す可能性がある．

構成概念妥当性については，本研究で採用した 3 指標が学生の多面的な能力を完全に捉えきれていない可能性がある．CFRP は基礎的なプログラミング概念の理解を測定するが，LLM 活用能力（プロンプト戦略）やシステム設計能力は測定対象外である．また，COSMIC 法による機能規模は「量」の指標であり，コードの保守性や再利用性といった「質」の側面を必ずしも反映しない．さらに，COSMIC 法は学生自身の寄与と LLM の寄与を分離できない．品質関連指標に用いた「24 件のテストケース」の数や生成プロセスについても検討の余地がある．1 つの成果物に対し 24 件のケースを課す設計は現実的だが，大規模な開発では網羅性が不十分な可能性がある．また，LLM 生成コードを LLM でテストする際の「自己検証の妥当性」という課題も残る．本研究では，プロンプトに SHIFT 社基準を明示することで検証観点のばらつきを抑制したが，今後は規模に応じた動的設定や人間による期待値の検証を組み込むなど，評価手法の改善が必要である．

5. おわりに

本研究では，LLM 活用が推奨された個人自由制作型 PBL を対象に，2024 年度と 2025 年度の成果物規模および 2025

年度の成果物指標を分析した．分析の結果，2025 年度の成果物は 2024 年度と比較して平均機能規模が約 3.2 倍大きく，LLM 利用環境の改善と同時期に規模差が観察された．また，2025 年度の CFRP には天井効果があり，CFRP と機能数の相関は弱かった一方で，CFRP と共通テスト通過率には中程度の正の相関が見られた．

本研究の意義は，LLM 活用を前提とした個人自由制作型 PBL において，成果物の「規模」と「動作確認結果」を分けて観察し，さらに LLM 補助評価の制約を明示した点にある．ただし，本研究は LLM 利用ログを取得しておらず，授業構成の年度差も存在するため，観察された年度間差を LLM 性能向上の因果効果として断定することはできない．

今後の課題は 3 点である．第一に，機能数測定の一部を人手で再測定し，LLM 補助測定との一致度を検証することである．第二に，LLM 生成テストケースの妥当性，実行可能性，期待値の正確性を人手で分類し，テスト通過率の信頼性を評価することである．第三に，プロンプト履歴や利用モデル，利用フェーズを収集し，学生が LLM をどのように用いて要求分析，設計，実装，デバッグを進めたかをプロセスとして分析することである．これらにより，LLM 活用環境下の個人自由制作型 PBL における成果物評価と教育支援の設計をより具体化できると考えられる．

謝辞 本研究の実施にあたり，研究の趣旨を理解し，調査に協力いただいた日本大学理工学部の受講学生諸氏に深く感謝いたします．受講学生に対し，研究協力は任意で成績に影響しないこと，外部公表の可能性があることを文書で説明し，書面同意を得た．前後対応のために教員が学生 ID で識別子を付与し，対応表は分離管理の上で廃棄した．分析担当者には識別子を除去した分析用データのみ提供し，個人が特定されない集計結果のみを本論文に用いている．

参考文献

[1] Li, J., Zhang, M., Li, N., Weyns, D., Jin, Z. and Tei, K.: Generative AI for Self-Adaptive Systems: State of the Art and Research Roadmap, *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 19, No. 3, pp. 13:1–13:60 (2024).

[2] Haque, M. A.: LLMs: A game-changer for software engineers?, *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, Vol. 5, No. 1, p. 100204 (2025).

[3] Sengul, C., Neykova, R. and Destefanis, G.: Software engineering education in the era of conversational AI: current trends and future directions, *Frontiers in Artificial Intelligence*, Vol. 7, p. 1436350 (2024).

[4] Ahmed, S., Bukhari, S. A. H., Ahmad, A., Rehman, O., Ahmad, F., Ahsan, K. and Liew, T. W.: Inquiry based learning in software engineering education: exploring students' multiple inquiry levels in a programming course, *Frontiers in Education*, Vol. 10, p. 1503996 (2025).

[5] Finnie-Ansley, J., Denny, P., Luxton-Reilly, A., Santos, E. A., Prather, J. and Becker, B. A.: My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises, *Proceedings of the 25th Australasian Computing Education Conference (ACE '23)*, pp. 97–104 (2023).

[6] Xue, Y., Chen, H., Bai, G. R., Tairas, R. and Huang, Y.: Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1, *Proc. 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24)*, pp. 331–341 (2024).

[7] Vadaparty, A., Zingaro, D., Smith IV, D. H., Padala, M., Alvarado, C., Benario, J. G. and Porter, L.: CS1-LLM: Integrating LLMs into CS1 Instruction, *Proc. 2024 Innovation and Technology in Computer Science Education (ITiCSE 2024)*, pp. 297–303 (2024).

[8] Wermelinger, M.: Using GitHub Copilot to Solve Simple Programming Problems, *Proc. 54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023)*, pp. 172–178 (2023).

[9] Cipriano, B. P. and Alves, P.: LLMs Still Can't Avoid Instanceof: An Investigation Into GPT-3.5, GPT-4 and Bard's Capacity to Handle Object-Oriented Programming Assignments, *Proc. 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24)*, pp. 162–169 (2024).

[10] Ziegler, A., Kalliamvakou, E., Li, X. A., Rice, A., Rifkin, D., Simister, S., Sittampalam, G. and Aftandilian, E.: Measuring GitHub Copilot's Impact on Productivity, *Communications of the ACM*, Vol. 67, No. 3, pp. 54–61 (2024).

[11] Kataoka, H., Li, J. and Matsuno, Y.: Amplifiers or Equalizers? A Longitudinal Study of LLM Evolution in Software Engineering Project-Based Learning, *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '26)* (2026).

[12] Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J. and Santos, E. A.: "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers, *ACM Transactions on Computer-Human Interaction*, Vol. 31, No. 1, pp. 4:1–4:31 (2024).

[13] Federiakin, D., Molerov, D., Zlatkin-Troitschanskaia, O. and Maur, A.: Prompt engineering as a new 21st century skill, *Frontiers in Education*, Vol. 9, p. 1366434 (2024).

[14] 佐藤美唯，野口結衣，伊東和香，梶浦照乃，高野志歩，倉光君郎：生成 AI を活用したプログラミングに重要なスキルの予備調査，2024 年度人工知能学会全国大会（第 38 回）(2024).

[15] Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., Savelka, J., Sheard, J., Simon and Wermelinger, M.: The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers, *Proceedings of the 2024 ACM Conference on International Computing Education Research (ICER 2024)*, pp. 469–486 (2024).

[16] Liu, R., Zenke, C., Liu, C., Holmes, A., Thornton, P. and Malan, D. J.: Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education, *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE 2024)*, pp. 750–756 (2024).

[17] Kohen-Vacs, D., Ronen, M. and Bar-Ness, V.: Integrating Generative AI into Programming Education: Student Perceptions and the Challenge of Correcting AI Errors, *International Journal of Artificial Intelligence in Education*, Vol. 35, pp. 3166–3184 (2025).
[18] Majumdar, R., Akçapinar, G., Flanagan, B. and Ogata, H.: Mining Epistemic Actions of Programming Problem Solving with ChatGPT, *Proceedings of the 17th International Conference on Educational Data Mining (EDM 2024)*, pp. 1–6 (2024).
[19] Ma, Q., Käser, T., Murphy, C. and Zilles, C.: How to Teach Programming in the AI Era? Using LLMs as a Teachable Agent for Debugging, *Proceedings of the 25th International Conference on Artificial Intelligence in Education (AIED 2024)*, pp. 265–279 (2024).
[20] Kazemitabaar, M., Chow, J., Ma, W., Ericson, B. J., Weintrop, D. and Grossman, T.: CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs, *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI 2024)*, pp. 1–20 (2024).
[21] Jošt, G., Leinonen, J. and Hellas, A.: The Impact of Large Language Models on Programming Education and Student Learning Outcomes, *Applied Sciences*, Vol. 14, No. 10, pp. 1–15 (2024).
[22] Chen, J., Fiannaca, A. J., Cai, C. J. and Terry, M.: Learning Agent-based Modeling with LLM Companions: Experiences of Novices and Experts Using ChatGPT and NetLogo Chat, *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI 2024)*, pp. 1–18 (2024).
[23] 日本産業規格: JIS X 0143:2013 ソフトウェア工学—COSMIC 機能規模測定手法 (2013).
[24] McCabe, T. J.: A Complexity Measure, *IEEE Transactions on Software Engineering*, Vol. SE-2, No. 4, pp. 308–320 (1976).
[25] Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C. and Toll, D.: Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies, *Proceedings of the 2015 ITiCSE on Working Group Reports (ITiCSE-WGR '15)*, pp. 41–63 (2015).
[26] プログラミング能力検定協会: CFRP とは，プログラミング能力検定（プロ検）（オンライン），入手先 ⟨https://programming-sc.com/cfrp/⟩（参照 2024-12-01）.
[27] Google: Gemini 2.5: Updates to our family of thinking models, Google Developers Blog (online), available from ⟨https://developers.googleblog.com/gemini-2-5-thinking-model-updates/⟩ (accessed 2026-05-16).
[28] OpenAI: Introducing OpenAI o3 and o4-mini, OpenAI (online), available from ⟨https://openai.com/index/introducing-o3-and-o4-mini/⟩ (accessed 2026-05-16).
[29] Anthropic: Introducing Claude 4, Anthropic (online), available from ⟨https://www.anthropic.com/news/claude-4⟩ (accessed 2026-05-16).
[30] SHIFT 株式会社: SHIFT 良質テスト 3 視点: 参考，SHIFT（オンライン），入手先 ⟨https://service.shiftinc.jp/column/8223/⟩（参照 2024-12-01）.

片岡 葉奈

2026 年 3 月，日本大学大学院理工学研究科情報科学専攻博士前期課程修了．同年 4 月より株式会社ベネッセコーポレーションに勤務．

李 家隆 （正会員）

2025 年早稲田大学基幹理工学研究科情報理工・情報通信専攻博士課程修了．2025 年より早稲田大学講師，東京科学大学特別研究員 (兼任)，現在に至る．自己適応システム，要求工学，大規模言語モデルの研究に従事．

依田 みなみ （正会員）

日本大学理工学部応用情報工学科助手．2017 年電気通信大学大学院情報システム学研究科博士前期課程修了．2017 年から 2019 年まで，トヨタ自動車株式会社において車載 OS 向けセキュリティ機能の開発に従事．2024 年，電気通信大学大学院情報理工学研究科情報学専攻後期博士課程修了．博士（工学）．IoT 機器の脆弱性の検出に関する研究に従事．

松野 裕 （正会員）

2001 年 3 月東京大学工学部電子工学科卒業．2006 年 3 月，東京大学大学院新領域創成科学研究科博士課程修了．博士（科学）．東京大学情報基盤センター特任講師，名古屋大学情報連携統括本部特任講師，電気通信大学大学院情報システム学研究科助教などを経て，2015 年 4 月より日本大学理工学部応用情報工学科准教授，2023 年 4 月同学科教授．プログラミング言語，システム保証，ディペンダビリティ，安全性などに興味を持つ．情報処理学会，日本ソフトウエア科学会，電子情報通信学会会員．