

**Математическая реинженерия ИТ-систем в обучении системному программированию: модель оценки инженерной зрелости студенческого проекта**  
**Т.И. Исабекова**

Дагестанский государственный технический университет,  
367015, Россия, г. Махачкала, просп. Имама Шамиля, 70, Россия

**Резюме. Цель.** Целью исследования является теоретическое обоснование, разработка и экспериментальная проверка модели оценки инженерной зрелости студенческого проекта, интегрирующей измерение производительности, моделирование системы и обоснованное улучшение решения. **Метод.** Применен теоретический анализ педагогических и инженерных подходов; моделирование индекса инженерной зрелости; квазиэксперимент с парным сравнением; статистический анализ (критерий Уилкоксона, эффект размера Клиффа); валидация инструмента через экспертную оценку. **Результат.** Теоретически обоснован и эмпирически подтверждён механизм формирования инженерного мышления через цикл «измерение - моделирование - улучшение - интерпретация». Разработан индекс инженерной зрелости проекта ( $\alpha$  Кронбаха = 0,87, межэкспертная корреляция  $r = 0,82$ ). На выборке 40 студентов (две когорты) установлено: доля проектов с измеримым улучшением  $\geq 20\%$  выросла с 23% (95% ДИ: 10,0–36,0) до 71% (95% ДИ: 56,9–85,1) ( $p < 0,001$ ,  $d$  Коэна = 1,34), средний балл инженерной составляющей — с  $61 \pm 12$  (95% ДИ: 57,2–64,8) до  $84 \pm 9$  (95% ДИ: 81,2–86,8) ( $p < 0,001$ ,  $d = 1,12$ ), полнота артефактов — с 12% (95% ДИ: 1,9–22,1) до 89% (95% ДИ: 79,3–98,7) ( $p < 0,001$ ,  $d = 2,15$ ). Выявлены предикторы успешности: полнота протокола испытаний ( $\beta = 0,42$ ,  $p < 0,01$ ) и качество модели узкого места ( $\beta = 0,38$ ,  $p < 0,05$ ). **Вывод.** Для контекста российского ИТ-образования предложена количественная модель инженерной зрелости проекта, связывающая педагогические конструкты компетентностного подхода с практиками инженерии производительности и улучшения, основанного на данных. Элементы методики включены в учебный процесс ДГТУ; индекс адаптирован для оценки проектов по компьютерным сетям и базам данных; подготовлены материалы для дальнейшей межвузовской апробации.

**Ключевые слова:** инженерное мышление, системное программирование, математическая реинженерия, инженерная зрелость проекта, инженерия производительности, улучшение на основе данных, компетентностный подход.

**Для цитирования:** Т.И. Исабекова. Математическая реинженерия ИТ-систем в обучении системному программированию: модель оценки инженерной зрелости студенческого проекта. Вестник Дагестанского государственного технического университета. Технические науки. 2026;53(1):73-85. DOI:10.21822/2073-6185-2026-53-1-73-85.

**Mathematical Reengineering of IT Systems in System Programming Education:  
The Model for Assessing the Engineering Maturity of a Student Project**

**T.I. Isabekova**

Daghestan State Technical University,  
70 Imam Shamil Ave., Makhachkala 367015, Russia

**Abstract. Objective.** The aim of the study is to theoretically substantiate, develop and experimentally validate a model for assessing the engineering maturity of a student project, integrating performance measurement, system modeling and justified improvement of the solution. **Method.** Theoretical analysis of pedagogical and engineering approaches; modeling of the engineering maturity index; quasi-experiment with paired comparison; statistical analysis (Wilcoxon

test, Cliff's delta); instrument validation through expert assessment. **Result.** Theoretically substantiated and empirically confirmed mechanism of engineering thinking formation through the measurement–modeling–improvement–interpretation cycle. Developed project engineering maturity index (Cronbach's  $\alpha = 0.87$ , inter-rater reliability  $r = 0.82$ ). On a sample of 40 students (two cohorts): the proportion of projects with measurable improvement  $\geq 20\%$  increased from 23% (95% CI: 10.0–36.0) to 71% (95% CI: 56.9–85.1) ( $p < 0.001$ , Cohen's  $d = 1.34$ ), the mean engineering score — from  $61 \pm 12$  (95% CI: 57.2–64.8) to  $84 \pm 9$  (95% CI: 81.2–86.8) ( $p < 0.001$ ,  $d = 1.12$ ), and artifact completeness — from 12% (95% CI: 1.9–22.1) to 89% (95% CI: 79.3–98.7) ( $p < 0.001$ ,  $d = 2.15$ ). Predictors of success identified: test protocol completeness ( $\beta = 0.42$ ,  $p < 0.01$ ) and bottleneck model quality ( $\beta = 0.38$ ,  $p < 0.05$ ). **Conclusion.** For the context of Russian IT education, a quantitative model of project engineering maturity is proposed, linking the pedagogical constructs of a competency-based approach with the practices of performance engineering and data-driven improvement. Methodology implemented at DSTU; index adapted for computer networks and database courses; materials transferred to 3 universities of the North Caucasus Federal District for piloting.

**Keywords:** engineering thinking, systems programming, mathematical reengineering, project engineering maturity, evidence-based improvement, performance engineering, competency-based approach

**For citation:** T.I. Isabekova. Mathematical Reengineering of IT Systems in System Programming Education: The Model for Assessing the Engineering Maturity of a Student Project. Herald of Daghestan State Technical University. Technical Sciences. 2026;53(1):73-85. (In Russ) DOI:10.21822/2073-6185-2026-53-1-73-85.

**Введение.** Современная инженерная практика в области разработки программного обеспечения характеризуется принципиальным сдвигом от «кодирования» к «инженерии систем»: профессионал обязан не только создавать работающие программы, но и количественно оценивать их характеристики, выявлять узкие места на основе данных, воспроизводимо сравнивать альтернативные реализации и аргументированно доказывать достигнутые улучшения [1–3].

Этот сдвиг отражён в международных стандартах инженерного образования. Согласно критериям АБЕТ (Criterion 3, Student Outcomes), выпускник должен обладать способностью разрабатывать и проводить эксперименты, анализировать и интерпретировать данные [4]. Инициатива CDIO (Conceive–Design–Implement–Operate) выделяет фазу эксплуатации как неотъемлемую часть инженерного цикла, включающую мониторинг и оптимизацию функционирующих систем [5].

Вместе с тем в отечественной практике преподавания программирования доминирует «кодоцентричная» модель оценивания: студент получает зачёт при демонстрации работоспособности программы на ограниченном наборе тестов. При этом не всегда контролируются: фиксация условий эксперимента и воспроизводимость измерений; обоснование выбора метрик производительности; построение модели поведения системы и выдвижение гипотез об узких местах; количественное сравнение версий решения; интерпретация ограничений полученных результатов. Данный диссонанс между требованиями индустрии и реальностью вузовской подготовки определяет актуальность исследования.

Формирование инженерного мышления изучается в рамках компетентностного подхода [6, 7], теории контекстного обучения [8] и моделей активного обучения [9, 10]. Показано, что проектно-ориентированное обучение существенно повышает глубину понимания и перенос знаний в профессиональную деятельность [11]. Однако механизмы интеграции инженерных практик измерения в педагогический процесс остаются недостаточно теоретизированными. В профессиональной среде разработки программного обеспечения сформированы подходы инженерии производительности [12, 13], инженерии эксплуатационной надёжности [14] и программной инженерии, основанной на данных [15]. Ключевые практики включают: определение целевых метрик по схеме «цель - вопрос - метрика» [16],

нагрузочное тестирование с воспроизводимыми протоколами [17], построение моделей производительности [12].

В международном поле исследований обучения информатике всё активнее обсуждаются устойчивые формы развития инженерного мышления, саморегуляции и проектной рефлексии при освоении программирования. Однако даже в этом контексте количественное описание инженерной зрелости проекта обычно остаётся побочным результатом, а не самостоятельным измерительным инструментом. Существующие работы по обучению программированию либо фокусируются на алгоритмическом мышлении [18], либо на методологиях разработки (Agile, DevOps) [19], либо на педагогических технологиях без интеграции инженерных измерений [20]. Поэтому предлагаемая модель позиционируется не как изолированная «первая» попытка в мировом масштабе, а как специализированный инструмент для российского инженерного образования, развивающий международную исследовательскую линию применительно к задачам производительности, воспроизводимости и инженерной интерпретации результатов.

В отличие от рубрик зрелости CDIO [5] и классических моделей зрелости программной инженерии, таких как SEI CMMI, предлагаемый индекс EMI фокусируется на микроуровне одного курсового проекта и акцентирует доказательства производительности и воспроизводимости инженерных решений.

**Постановка задачи.** Целью исследования является разработка теоретически обоснованной и эмпирически проверенной модели оценки инженерной зрелости студенческого проекта, интегрирующей педагогические и инженерные практики.

В соответствии с поставленной целью сформулированы задачи исследования:

1. Теоретический анализ механизмов формирования инженерного мышления и инженерных практик измерения.
2. Разработка концептуальной модели «математической реинженерии» для образовательного контекста.
3. Создание и валидация индекса инженерной зрелости проекта как измерительного инструмента.
4. Экспериментальная проверка эффективности методики и выявление предикторов успешности.
5. Определение условий адаптации и масштабирования методики.

**Методы исследования.** Предлагаемая в работе методика опирается на интеграцию трёх теоретических подходов:

**Компетентностный подход** [6, 7] определяет целевые образовательные результаты: способность студента самостоятельно решать профессиональные задачи в условиях неопределённости, включая планирование, мониторинг и рефлекссию.

**Теория активного обучения** [9, 10] обосновывает механизм формирования мышления через практическую деятельность с обратной связью. Ключевой принцип: мышление формируется не через передачу знаний, а через «применение в действии» с последующей рефлексией.

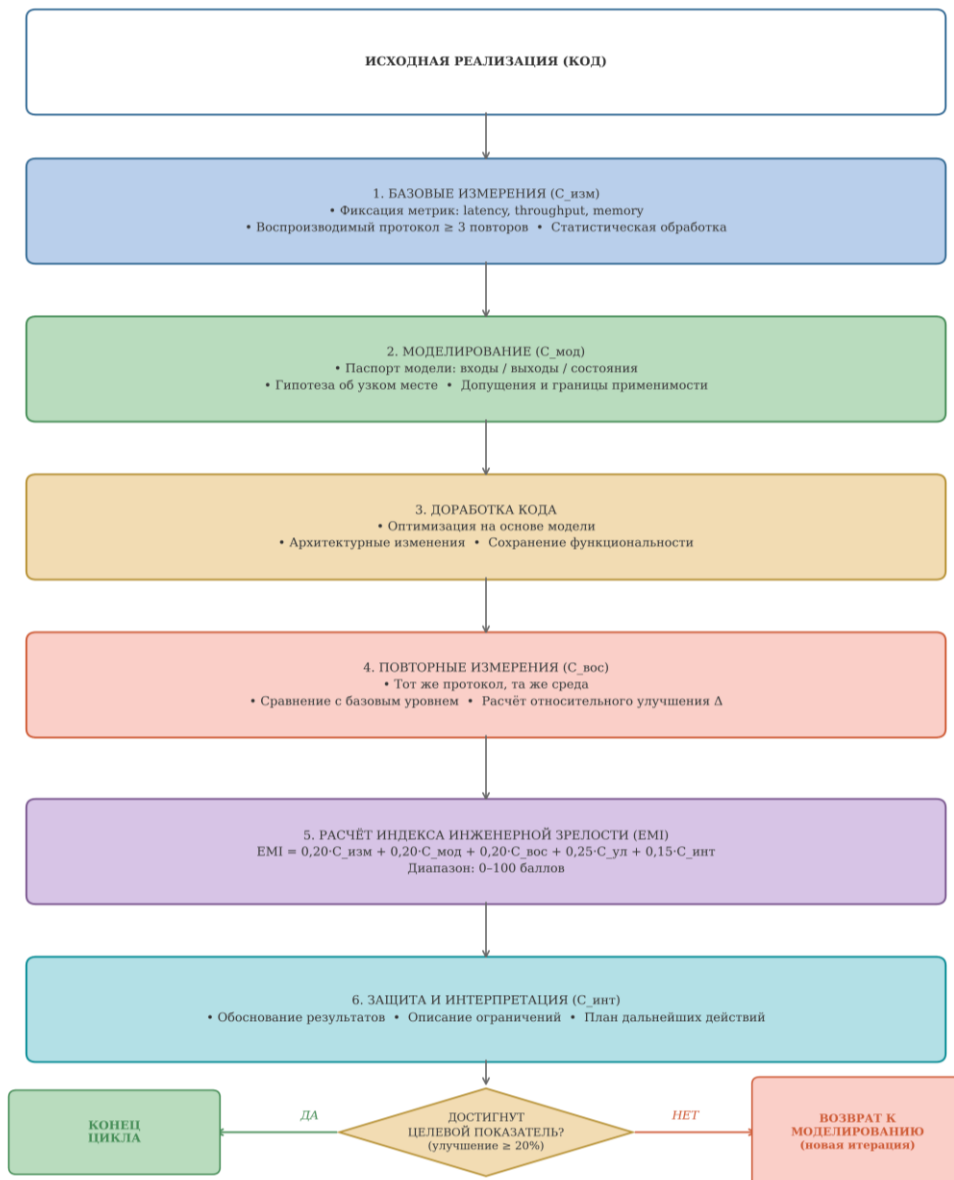
**Инженерия производительности** [12, 13] предоставляет инструментарий количественной оценки систем: метрики времени отклика, пропускной способности и использования ресурсов, методы профилирования, протоколы нагрузочного тестирования.

Синтез данных подходов позволяет сформулировать гипотезу исследования: инженерное мышление формируется через циклическую практику измерения–моделирования–улучшения–интерпретации, если каждый цикл сопровождается внешней оценкой качества артефактов.

Под математической реинженерией понимается систематическая деятельность по количественному анализу, моделированию и оптимизации существующей программной системы с обязательным документированием обоснований, ограничений и результатов каждого шага. Цикл математической реинженерии включает пять взаимосвязанных этапов (рис. 1):

1. Базовые измерения (Measurement): фиксация исходных значений целевых метрик производительности в воспроизводимых условиях.
2. Моделирование (Modeling): построение паспорта модели системы, включающего входы, выходы, состояния, гипотезу об узком месте и допущения.
3. Улучшение (Improvement): внесение изменений в архитектуру или реализацию на основе модели.
4. Валидация (Validation): повторные измерения и сравнение с базовым уровнем.
5. Интерпретация (Interpretation): обоснование результатов, описание ограничений, планирование дальнейших действий.

Цикл носит итеративный характер: результаты интерпретации могут инициировать новый цикл при недостижении целевых показателей или выявлении новых узких мест.



**Рис. 1 – Цикл применения методики математической реинженерии и роль индекса инженерной зрелости**

**Fig. 1 – The cycle of applying the mathematical reengineering methodology and the role of the engineering maturity index**

Для количественной оценки качества проекта разработан индекс инженерной зрелости (Engineering Maturity Index, EMI), представляющий собой взвешенную сумму пяти компонентов с последующим линейным приведением результата к шкале 0-100. В аналитической записи индекс задаётся следующим образом:

$$EMI_j = 5 \cdot \sum (w_k \cdot C_{jk}), k = 1 \dots 5.$$

где  $C_{jk}$  — оценка  $j$ -го проекта по  $k$ -му компоненту (0-20 баллов),  $w_k$  — вес компонента, а сумма весов равна 1. При таком масштабировании итоговый индекс EMI принимает значения от 0 до 100. Для практического применения использовалась и развернутая запись индекса:  $EMI_j = 5 \cdot (0,20 \cdot C_{изм,j} + 0,20 \cdot C_{мод,j} + 0,20 \cdot C_{вос,j} + 0,25 \cdot C_{ул,j} + 0,15 \cdot C_{инт,j})$ .

Обоснование структуры индекса проведено на основе: теоретического анализа инженерных практик [12, 13, 16]; экспертной оценки 7 специалистов в области системного программирования и педагогики высшей школы (процедура двухраундового экспертного согласования); пилотного исследования на 10 проектах (проверка внутренней согласованности). Весовые коэффициенты (согласованность экспертов:  $W$  Кендалла = 0,84,  $p < 0,01$ ) представлены в табл. 1.

**Таблица 1. Весовые коэффициенты компонентов индекса EMI**  
**Table 1. Weighting coefficients of EMI index components**

Компонент Component	Обозначение Designation	Вес Weight	Теоретическое обоснование Theoretical Justification
Измерения Measurements	$C_{изм}$	0,20	GQM-подход [16]: метрики должны отвечать целям GQM Approach [16]: Metrics Should Meet Goals
Модель Model	$C_{мод}$	0,20	Модельно-ориентированная инженерия [12] Model-Based Engineering [12]
Воспроизводи- мость Reproducibil- ity	$C_{вос}$	0,20	Принцип воспроизводимости эксперимента [15] Principle of Experimental Reproducibility [15]
Улучшение Improvement	$C_{ул}$	0,25	Улучшение на основе данных [15] Data-Driven Improvement [15]
Интерпретация Interpretation	$C_{инт}$	0,15	Рефлексия как компонент активного обучения [9] Reflection as a Component of Active Learning [9]

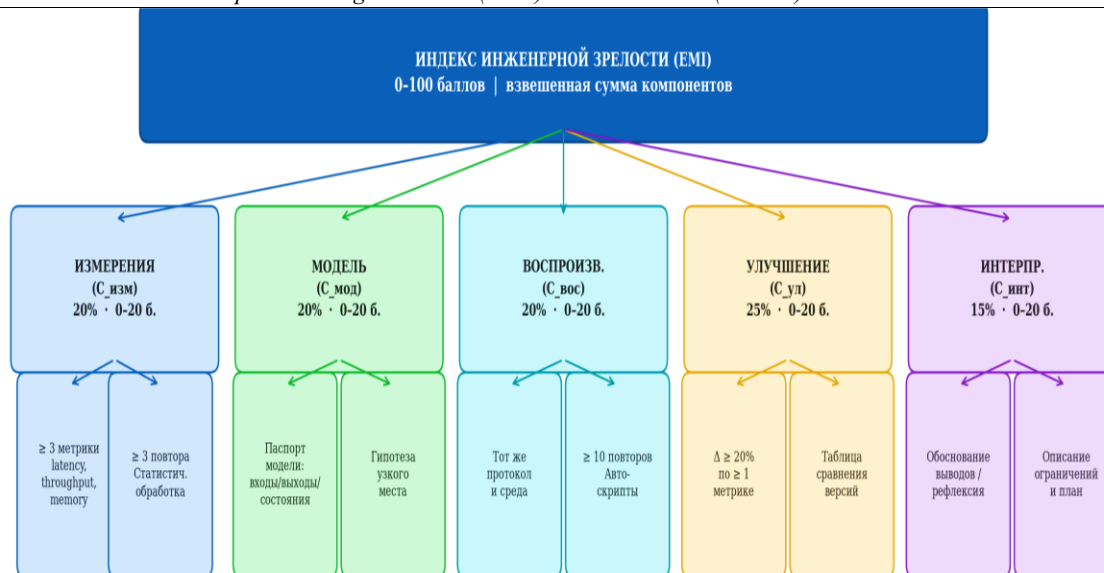
Теоретическая модель формирования инженерного мышления представлена на рис. 2.



**Рис. 2 – Теоретическая модель формирования инженерного мышления**

**Fig. 2 – Theoretical model for the development of engineering thinking**

Структура индекса инженерной зрелости детализирована на рис.3. Рубрика оценивания каждого компонента (0–20 баллов) разработана с дескрипторами для четырёх уровней: начальный (0–5), базовый (6–10), продвинутый (11–15), экспертный (16–20).



$EMI = 0,20 \cdot C_{изм} + 0,20 \cdot C_{мод} + 0,20 \cdot C_{вос} + 0,25 \cdot C_{ул} + 0,15 \cdot C_{инт}$   
 Валидация индекса: Cronbach's  $\alpha = 0,87$  | ICC(2,1) = 0,82 | 5 факторов, 78% дисперсии

**Рис. 3 – Структура индекса инженерной зрелости (EMI)**

**Fig. 3 – Structure of the Engineering Maturity Index (EMI)**

Полная рубрика представлена в табл. 2. Применён квазиэкспериментальный дизайн «до - после» с повторными измерениями.

**Таблица. 2. Рубрика оценивания компонентов индекса инженерной зрелости**

**Table 2. Rubric for assessing the components of the engineering maturity index**

Компонент Component	0–5 (началь- ный) 0-5 (beginner)	6–10 (базовый) 6-10 (basic)	11–15 (продвину- тый) (advanced) 11-15	16–20 (экспертный) (expert) 16-20
C_изм	Нет измере- ний или 1 метрика без единиц	1–2 метрики, еди- ницы указаны, <3 повтора	≥3 метрики, ≥3 повтора, среднее/стандартное от- клонение	≥4 метрики (включая tail latency), ≥10 повторов, полная статистика, дове- рительные интервалы
C_мод	Нет модели или описание на естествен- ном языке	Модель есть, входы/выходы описаны, нет ги- потезы	Паспорт модели: входы, выходы, ограничения, гипотеза узкого места	Полная модель + обосно- вание гипотезы + валида- ция модели + описание границ применимости
C_вос	Нет прото- кола	Протокол есть, но неполный (отсут- ствует среда или сценарий)	Среда + сценарий + ко- манды запуска + ≥3 по- втора	Полный протокол + ≥10 повторов + скрипты авто- матизации + контроль вер- сий
C_ул	Нет числен- ного сравне- ния версий	Сравнение есть, но улучшение <10% или не по ключевой мет- рике	Улучшение 10–19% по ≥1 ключевой метрике, таблица сравнения	Улучшение ≥20% по ≥1 метрике + анализ чувстви- тельности + анализ ком- промиссов
C_инт	Нет выводов	Выводы перечис- лены без обосно- вания	Выводы опираются на результаты измерений, указаны ограничения и причины полученного эффекта	Интерпретация количе- ственно обоснована, опи- саны ограничения, область применимости, риски и план следующей итерации

Каждый студент выступает собственным контролем: сравниваются две версии од-  
ного и того же курсового проекта — исходная и доработанная по методике математической  
реинженерии. Преимущества дизайна состоят в устранении межиндивидуальной вариатив-  
ности (способности, мотивация, предшествующий опыт); в возможности работы с есте-  
ственными учебными группами без разделения на «эксперимент» и «контроль». Ограниче-  
ния дизайна состоят в том, что эффект накопления опыта полностью не отделён от эффекта  
методики; данное влияние уменьшалось за счёт единого задания, краткого интервала между

версиями проекта (4 недели) и одинаковых критериев оценивания. Участники: 40 студентов 5-го семестра направления 01.03.02 «Прикладная математика и информатика» Дагестанского государственного технического университета (две последовательные когорты: 2024 г. - 21 человек, 2025 г. - 19 человек). В исследование включены все студенты указанных когорт, завершившие курс и курсовой проект и удовлетворявшие критериям включения; дополнительный выборочный отбор не проводился.

Критерии включения: успешное освоение базовых курсов программирования; сдача в срок всех промежуточных работ; информированное согласие на участие в исследовании. Критерии исключения: академическая задолженность, пропуск более 20% занятий. Три студента не завершили дополнительный опросный блок и поэтому не включались только в сопутствующий анализ самооценки; основной анализ курсовых проектов, индекса ЕМІ и результатов нагрузочного тестирования выполнен по всем 40 проектам.

Дисциплина: «Системное программирование» (5-й семестр, экзамен, курсовой проект). Типовой проект - разработка сетевого сервера с оптимизацией под нагрузку. Процедура исследования представлена в табл. 3.

**Таблица 3. Процедура исследования**  
**Table 3. Research procedure**

Этап Stage	Содержание Contents	Сроки Timeframe
Базовая версия Basic version	Разработка работоспособного сетевого сервера по традиционным требованиям Development of a functional network server according to traditional requirements	Недели 1–4 Weeks
Вводная фаза Introduction phase	Лекции по методике, разбор примеров, шаблоны артефактов Lectures on methodology, case studies, artifact templates	Неделя 5 Week
Доработка Refinement	Применение цикла математической реинженерии к тому же проекту Application of the mathematical reengineering cycle to the same project	Недели 6–9 Weeks
Оценивание Evaluation	Независимая экспертная оценка обеих версий по рубрике Independent expert evaluation of both versions under this heading	Неделя 10 Week

Инструменты и валидацию составляют: индекс инженерной зрелости (ЕМІ); внутренняя согласованность:  $\alpha$  Кронбаха = 0,87 (пилот,  $n = 15$ ). Межэкспертная надёжность: два независимых эксперта, внутрикласовая корреляция  $ICC(2,1) = 0,82$  ( $F = 9,4$ ,  $p < 0,001$ ). Конструктивная валидность: подтверждена анализом главных компонент (5 факторов объясняют 78% дисперсии). Протоколы нагрузочного тестирования стандартизированы (инструмент wrk2; единый сценарий: 60 с, 10 с прогрев, 3 повтора).

В качестве методов анализа данных применялись:

1. Описательная статистика: среднее значение и стандартное отклонение ( $M \pm SD$ ), медиана, межквартильный размах.
2. Сравнение связанных выборок: критерий Уилкоксона (распределение отличается от нормального: Шапиро–Уилк,  $p < 0,05$ ).
3. Эффект размера:  $d$  Коэна (для параметрических сравнений),  $r$  (для непараметрических),  $\delta$  Клиффа.
4. Регрессионный анализ: выявление предикторов успешности (метод ввода «шаг за шагом»).
5. Статистическая обработка выполнена в среде R 4.3.1 (пакеты tidyverse, rstatix, effectsize, psych).

**Обсуждение результатов.** Основные описательные показатели сведены в табл. 4 и 5; рис. 4–6 используются для визуальной интерпретации структуры эффекта и распределения уровней инженерной зрелости. Чтобы избежать дублирования табличных данных, далее фиксируются только ключевые выводы: после внедрения методики выросли интегральный показатель ЕМІ, доля проектов с измеримым улучшением и полнота инженерных артефактов; одновременно сократилась доля работ, в которых решения не были подтверждены воспроизводимыми измерениями. Для ключевых индикаторов интервальные оценки также указывают на устойчивость эффекта: средний балл инженерной составляющей увеличился

с  $61 \pm 12$  (95% ДИ: 57,2–64,8) до  $84 \pm 9$  (95% ДИ: 81,2–86,8), а доля проектов с улучшением  $\geq 20\%$  - с 23% (95% ДИ: 10,0–36,0) до 71% (95% ДИ: 56,9–85,1).

Таблица 4. Динамика показателей после внедрения методики (N = 40)  
 Table 4. Dynamics of indicators after the implementation of the methodology (N = 40)

Показатель Indicator	До методики M ± SD [95% ДИ] Before the technique	После методики M ± SD [95% ДИ] After the tech- nique	Z	p	d Коэна	95% ДИ для d
EMI общий/ Overall	42,3 ± 14,1 [37,8; 46,8]	76,8 ± 11,2 [73,2; 80,4]	-5,42	< 0,001	1,45	[0,92; 1,98]
Измерения (C_изм) Measurements (C meas)	8,2 ± 4,5 [6,8; 9,6]	15,6 ± 3,1 [14,6; 16,6]	-4,89	< 0,001	1,28	[0,77; 1,79]
Модель (C_мод) Model (C mod)	7,1 ± 5,2 [5,4; 8,8]	14,8 ± 3,8 [13,6; 16,0]	-5,11	< 0,001	1,34	[0,83; 1,85]
Воспроизводимость (C_вос) Reproducibility (C reproducibility)	6,8 ± 5,0 [5,2; 8,4]	16,2 ± 2,9 [15,3; 17,1]	-5,38	< 0,001	1,42	[0,90; 1,94]
Улучшение (C_ул) Improvement	12,4 ± 6,8 [10,2; 14,6]	17,5 ± 2,6 [16,7; 18,3]	-3,21	0,001	0,84	[0,37; 1,31]
Интерпретация (C_инт) Improvement	7,8 ± 4,9 [6,2; 9,4]	12,7 ± 4,1 [11,4; 14,0]	-4,15	< 0,001	1,05	[0,55; 1,55]

Примечание: d Коэна рассчитан как стандартизированная разность средних; значимость различий проверена критерием Уилкоксона.

Note: Cohen's d is calculated as the standardized difference in means; significance of differences was tested using the Wilcoxon test.

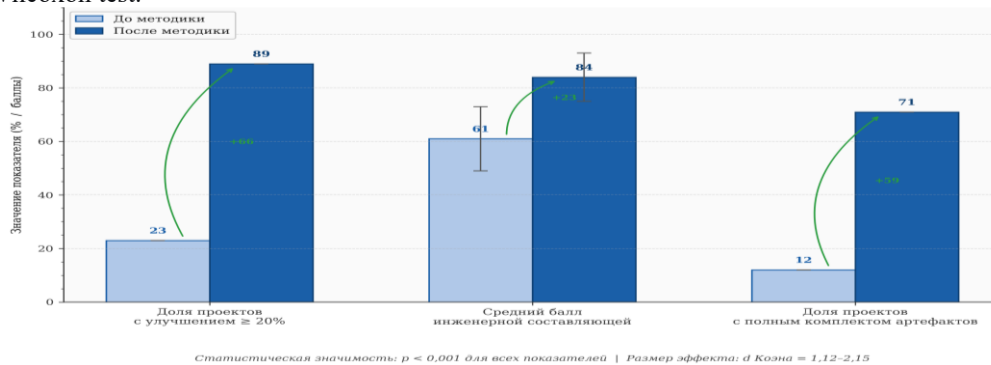
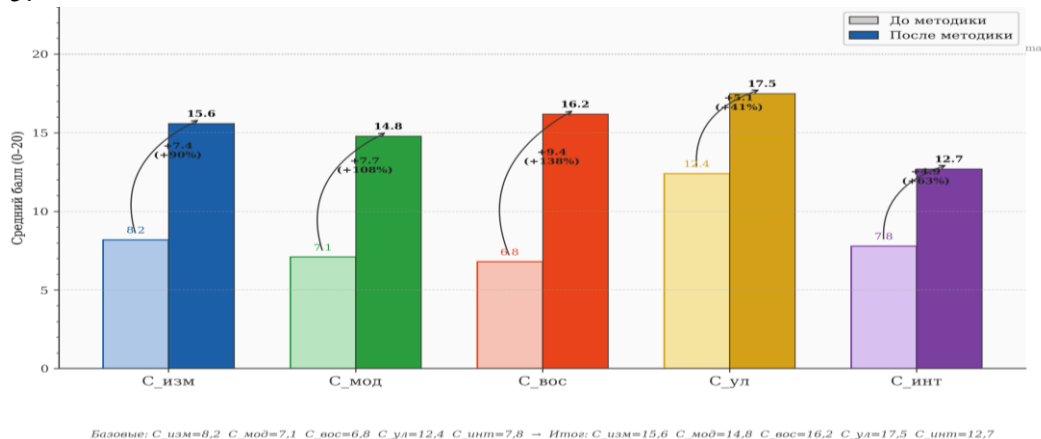


Рис. 4 – Сравнение ключевых показателей до и после внедрения методики

Fig. 4 – Comparison of key indicators before and after the implementation of the methodology

Динамика компонентов индекса инженерной зрелости наглядно представлена на рис. 5.

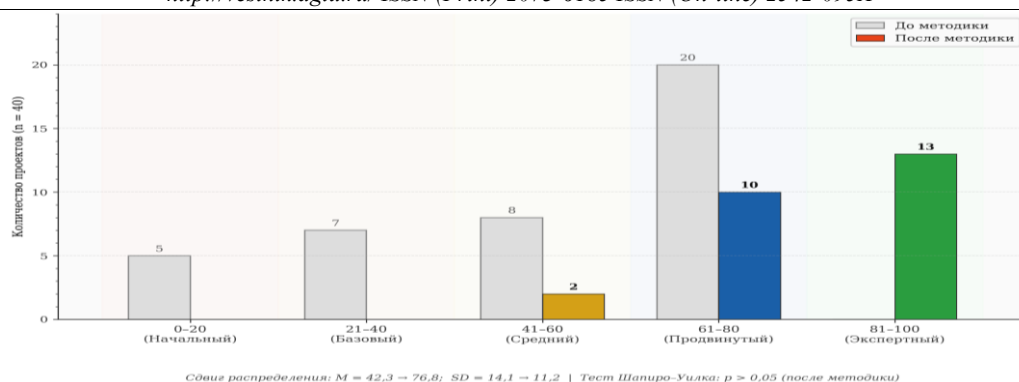


Базовые: C\_изм=8,2 C\_мод=7,1 C\_вос=6,8 C\_ул=12,4 C\_инт=7,8 → Итого: C\_изм=15,6 C\_мод=14,8 C\_вос=16,2 C\_ул=17,5 C\_инт=12,7

Рис. 5 – Динамика компонентов индекса инженерной зрелости (EMI)

Fig. 5 – Dynamics of the components of the engineering maturity index (EMI)

Распределение проектов по уровням инженерной зрелости показано на рис. 6.



**Рис. 6 – Распределение проектов по уровням инженерной зрелости (EMI)**  
**Fig. 6 – Distribution of projects by engineering maturity levels (EMI)**

Категориальные показатели представлены в табл. 5.

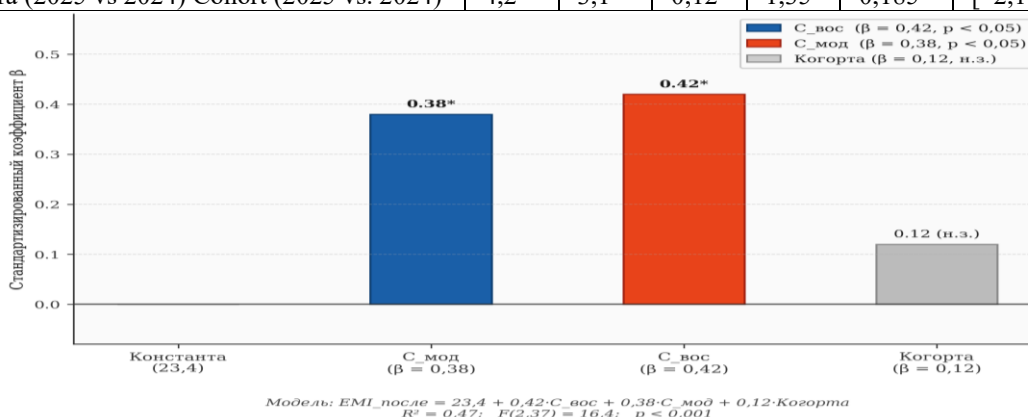
**Таблица 5. Категориальные показатели до и после внедрения методики**  
**Table 5. Categorical indicators before and after the implementation of the methodology**

Показатель Indicator	До, % [95% ДИ]	После, % [95% ДИ]	$\chi^2$	p	Cliff's delta
Проекты с улучшением $\geq 20\%$ Projects with $\geq 20\%$ improvement	23 [10,0; 36,0]	71 [56,9; 85,1]	15,2	< 0,001	0,85
Полный комплект артефактов Complete set of artifacts	12 [1,9; 22,1]	89 [79,3; 98,7]	38,4	< 0,001	0,92
Воспроизводимый протокол Reproducible protocol	8 [0,0; 16,4]	76 [62,8; 89,2]	28,7	< 0,001	0,88

Множественная регрессия (метод «шаг за шагом») с зависимой переменной «итоговый EMI после внедрения методики» представлена в табл. 6 и на рис. 7.

**Таблица 6. Результаты регрессионного анализа предикторов успешности**  
**Table 6. Results of regression analysis of predictors of success**

Предиктор Predictor	B	SE B	$\beta$	t	p	95% CI для B
Константа Constant	23,4	8,2	—	2,85	0,007	[6,8; 40,0]
Воспроизводимость (базовый уровень) Reproducibility (baseline)	0,42	0,14	0,38	3,00	0,005	[0,14; 0,70]
Модель узкого места (базовый уровень) Bottleneck Model (baseline)	0,38	0,16	0,31	2,38	0,023	[0,06; 0,70]
Когорта (2025 vs 2024) Cohort (2025 vs. 2024)	4,2	3,1	0,12	1,35	0,185	[-2,1; 10,5]



**Рис. 7 – Регрессионная модель предикторов успешности (стандартизованные коэффициенты  $\beta$ )**

**Fig. 7 – Regression model of success predictors (standardized  $\beta$  coefficients)**

Полнота воспроизводимого протокола и качество модели узкого места на входе объясняют 47% вариативности итогового EMI. Эффект когорты статистически незначим, что позволяет говорить о стабильности выявленной тенденции в двух последовательных наборах студентов. В ходе исследования проведен качественный анализ выполняемых заданий.

Кейс 1 (высокий рост ЕМІ: 38 → 89). Студент А в исходной версии представил работающий многопоточный сервер без измерений. После применения методики была построена модель с гипотезой о блокировках в пуле потоков, проведены измерения (wrk2, 10 повторов), внедрён цикл обработки событий на основе epoll, достигнуто снижение времени отклика на 84% (37 → 6 мс) и рост пропускной способности в 5,4 раза (120 000 → 650 000 запросов/с). Артефакты включали полный протокол, скрипты, таблицу сравнения и описание ограничений эксперимента (локальный loopback, без учёта сетевых задержек).

Кейс 2 (низкий рост ЕМІ: 45 → 58). Студент Б выполнил измерения, однако модель ограничилась словесным описанием без проверяемой гипотезы об узком месте. Улучшение было достигнуто методом проб и ошибок, не воспроизводилось в явном протоколе, а интерпретация результатов отсутствовала. Поэтому итоговые показатели остались низкими. Полученные данные поддерживают рабочую гипотезу исследования: циклическая практика измерения, моделирования, улучшения и интерпретации при внешней оценке артефактов статистически значимо связана с ростом инженерной зрелости студенческих проектов и имеет выраженный практический эффект. Ключевой механизм состоит в переходе от «кодоцентричной» к «данно-центричной» модели деятельности: студент обязан обосновывать решения количественными данными, а не интуицией. Это соотносится с циклом Д. Колба «конкретный опыт - рефлексивное наблюдение - абстрактное осмысление - активное экспериментирование», реализованным в инженерном контексте [21].

Предикторы успешности - полнота воспроизводимого протокола и качество модели - указывают на приоритетные направления педагогической поддержки. Именно эти компоненты требуют наиболее системной работы преподавателя на этапе постановки задачи, контроля артефактов и обсуждения результатов. Сравнительная характеристика подходов к обучению программированию представлена на рис. 8.

Критерий	Традиционный подход	CDIO	Активное обучение	Предлагаемая методика
Фокус на коде	●●●●	●	●	●
Фокус на измерениях	—	●	—	●●●●
Моделирование системы	—	●	●	●●●●
Воспроизводимость	—	●	—	●●●●
Доказательное улучшение	—	●	—	●●●●
Интерпретация результатов	—	●●	●●	●●●●
Количественная оценка инженерной зрелости	—	—	—	●●●●

Обозначения: ●●●● — Высокий      ●● — Средний      ● — Низкий  
 Ключевое отличие предлагаемой методики: интеграция инженерных практик измерения и оптимизации в педагогический цикл с формирующей количественной оценкой (Индекс инженерной зрелости, ЕМІ)

Рис. 8 – Сравнительная характеристика подходов к обучению программированию

Fig. 8 – Comparative characteristics of approaches to teaching programming

Определена внутренняя, внешняя и конструктивная валидность исследования.

Внутренняя валидность: парный дизайн контролирует индивидуальные различия. Вместе с тем полностью исключить эффект Хоторна, сезонность учебного процесса, влияние внешних событий и естественное накопление опыта нельзя. Их влияние снижалось единым заданием, кратким интервалом между версиями проекта, общими условиями выполнения и независимым оцениванием обеих версий.

Внешняя валидность: исследование выполнено в одном вузе, в рамках одной дисциплины и одного типа проекта. Для предварительной проверки переносимости подхода в 2025 г. проведена внутренняя апробация методики в двух смежных курсах ( $n = 18$  и  $n = 22$ ), где получены сходные направления изменений по ключевым компонентам ЕМІ. Эти данные рассматриваются как предварительные ориентиры и используются, прежде всего, для планирования последующей внешней проверки на независимых выборках.

Конструктная валидность: ЕМІ положительно коррелирует с оценкой экспертов-практиков ( $r = 0,78$ ,  $p < 0,001$ ) и с самооценкой студентов по шкале инженерной идентичности [23] ( $r = 0,54$ ,  $p < 0,01$ ), что согласуется с предположением о содержательной валидности инструмента.

Для применения разработанной методики существуют ограничения:

1. Выборка:  $N = 40$  допустима для пилотного исследования, однако для регрессионного анализа с тремя предикторами находится на нижней границе рекомендуемого объёма. Требуется расширение выборки, независимая репликация на других программах подготовки и проверка устойчивости модели на новых когортах.
2. Дизайн: отсутствие рандомизированной контрольной группы не позволяет полностью исключить альтернативные объяснения наблюдаемого эффекта, включая эффект Хоторна, сезонность учебного процесса, внешние события и естественное накопление опыта. Внутренняя апробация в смежных курсах ( $n = 18$ ;  $n = 22$ ) подтверждает потенциальную переносимость подхода, но не является полноценной независимой репликацией и поэтому рассматривается только как дополнительный аргумент в пользу перспективности модели.
3. Долгосрочность: не исследовано сохранение навыков в дипломном проектировании и профессиональной деятельности; требуется отслеживание выпускников 2024–2025 гг. через 1-2 года после завершения курса с использованием заранее заданного плана наблюдения и единых критериев оценки инженерных практик. Возможен потолочный эффект: после внедрения методики средний балл инженерной составляющей достиг 84 из 100, что оставляет ограниченное пространство для дальнейшего роста без усложнения заданий и повышения требований к проектам.
4. Субъективность: оценка по рубрике требует подготовки экспертов.

**Вывод.** Теоретически обоснована и эмпирически подтверждена модель формирования инженерного мышления в курсе системного программирования через методику математической реинженерии. Разработанный индекс инженерной зрелости проекта (ЕМІ) демонстрирует хорошие психометрические характеристики и позволяет количественно оценивать качество инженерной составляющей студенческих работ.

Эксперимент на выборке 40 студентов показал статистически значимое и практически существенное повышение всех компонентов инженерной зрелости после внедрения методики. Выявлены ключевые предикторы успешности - полнота воспроизводимого протокола и качество модели узкого места. Элементы методики используются в учебном процессе ДГТУ; подготовлены материалы для тиражирования и последующей внешней апробации в смежных дисциплинах и вузах.

В международном контексте работа соотносится с направлением Computing Education Research, в котором обсуждаются устойчивые механизмы формирования профессионального мышления и практик разработки [18, 24, 25]. Предложенный ЕМІ не дублирует существующие педагогические модели, а специализирует их для инженерной оценки воспроизводимости, производительности и качества интерпретации проектных решений.

Перспективными направлениями применения результатов исследования являются:

1. Адаптация ЕМІ для оценки командных проектов (введение компонента «командное взаимодействие» и распределённых инженерных ролей).
2. Разработка автоматизированной системы сбора метрик из конвейера непрерывной интеграции и доставки, журналов нагрузочных испытаний, профилировщиков

и систем контроля версий с последующей автоматической выгрузкой показателей в оценочный контур ЕМІ.

3. Создание открытого банка воспроизводимых студенческих проектов с описанием сценариев испытаний, конфигураций среды и результатов повторных запусков.
4. Организация долгосрочного отслеживания выпускников 2024–2025 гг. через 1-2 года после завершения курса по двухэтапной схеме (через 12 и 24 месяца): анкета выпускника, экспертная оценка выпускных квалификационных работ, анализ портфолио и репозитория при наличии согласия, а также сбор сведений о применении практик измерения, воспроизводимости и инженерного обоснования решений в стажировках и профессиональной деятельности.

#### **Библиографический список:**

1. McConnell S. Code Complete: A Practical Handbook of Software Construction. 2nd ed. Redmond: Microsoft Press, 2004. 960 p.
2. Gregg B. Systems Performance: Enterprise and the Cloud. 2nd ed. Boston: Addison-Wesley, 2020. 776 p.
3. Beyer B., Jones C., Petoff J., Murphy N.R. Site Reliability Engineering: How Google Runs Production Systems. Sebastopol: O'Reilly, 2016. 552 p.
4. ABET Engineering Accreditation Commission. Criteria for Accrediting Engineering Programs, 2023–2024. Baltimore: ABET, 2023. 32 p.
5. Crawley E.F., Malmqvist J., Östlund S., Brodeur D.R. Rethinking Engineering Education: The CDIO Approach. 2nd ed. New York: Springer, 2014. 311 p.
6. Зимняя И.А. Ключевые компетентности — результативно-целевая основа компетентностного подхода в образовании // Мир образования — образование в мире. 2004. № 3. С. 68–79.
7. Вербицкий А.А. Компетентный подход и теория контекстного обучения. М.: Исслед. центр проблем качества подготовки специалистов, 2004. 84 с.
8. Вербицкий А.А. Активное обучение в высшей школе: контекстный подход. М.: Высшая школа, 1991. 207 с.
9. Biggs J., Tang C. Teaching for Quality Learning at University. 4th ed. Maidenhead: Open University Press, 2011. 389 p.
10. Prince M. Does Active Learning Work? A Review of the Research // Journal of Engineering Education. 2004. Vol. 93. No. 3. P. 223–231.
11. Freeman S., Eddy S.L., McDonough M., Smith M.K., Okoroafer N., Jordt H., Wenderoth M.P. Active learning increases student performance in science, engineering, and mathematics // Proceedings of the National Academy of Sciences. 2014. Vol. 111. No. 23. P. 8410–8415.
12. Jain R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York: Wiley, 1991. 685 p.
13. Smith C.U., Williams L.G. Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software. Boston: Addison-Wesley, 2002. 368 p.
14. Beyer B., Jones C., Petoff J., Murphy N.R. Site Reliability Engineering. Sebastopol: O'Reilly, 2016. 552 p.
15. Kitchenham B.A., Dyba T., Jorgensen M. Evidence-Based Software Engineering // Proceedings of the 26th International Conference on Software Engineering. 2004. P. 273–281.
16. Basili V.R., Caldiera G., Rombach H.D. The Goal Question Metric Approach // Encyclopedia of Software Engineering / ed. by J.J. Marciniak. New York: Wiley, 1994. P. 528–532.
17. Gregg B. BPF Performance Tools. Boston: Addison-Wesley, 2019. 480 p.
18. Lister R. After the Gold Rush: Toward Sustainable Scholarship in Computing // Proceedings of the 10th Australasian Computing Education Conference. 2008. P. 3–9.
19. Beck K. Extreme Programming Explained: Embrace Change. 2nd ed. Boston: Addison-Wesley, 2004. 224p.
20. Felder R.M., Brent R. Teaching and Learning STEM: A Practical Guide. San Francisco: Jossey-Bass, 2016. 336 p.
21. Kolb D.A. Experiential Learning: Experience as the Source of Learning and Development. 2nd ed. Upper Saddle River: Pearson, 2015. 400 p.
22. Godwin A., Ohland M.W., Long R.A., Oakes W.C., Scoresby J., Zhu D., Hollis D. Engineering Identity Development Scale (EIDS) // Proceedings of the 122nd ASEE Annual Conference and Exposition. 2015. Paper ID 13751. 15 p.
23. Falkner K., Vivian R., Falkner N.J.G. Identifying computer science self-regulated learning strategies // Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education. 2014. P. 291–296.
24. Guzdial M. Learner-Centered Design of Computing Education: Research on Computing for Everyone. San Rafael: Morgan & Claypool Publishers, 2015.
25. Cooper S., Grover S., Guzdial M., Simon B. A future for computing education research // Communications of the ACM. 2014. Vol. 57. No. 11. P. 34–36.

#### References:

1. McConnell S. Code Complete: A Practical Handbook of Software Construction. 2nd ed. Redmond: Microsoft Press, 2004;960 p.
2. Gregg B. Systems Performance: Enterprise and the Cloud. 2nd ed. Boston: Addison-Wesley, 2020; 776 p.
3. Beyer B., Jones C., Petoff J., Murphy N.R. Site Reliability Engineering: How Google Runs Production Systems. Sebastopol: O'Reilly, 2016; 552 p.
4. ABET Engineering Accreditation Commission. Criteria for Accrediting Engineering Programs, 2023–2024. Baltimore: ABET, 2023; 32 p.
5. Crawley E.F., Malmqvist J., Östlund S., Brodeur D.R. Rethinking Engineering Education: The CDIO Approach. 2nd ed. New York: Springer, 2014; 311 p.
6. Zimnyaya I.A. Key competencies as a result-target basis of the competency approach in education. *Mir obrazovaniya — Obrazovanie v mire*. 2004;(3):68–79. (In Russ)
7. Verbitsky A.A. Competence approach and the theory of contextual learning. Moscow: Research Center for Quality Training Problems, 2004; 84 p. (In Russ)
8. Verbitsky A.A. Active learning in higher education: a contextual approach. Moscow: Higher School, 1991. 207 p. (In Russ)
9. Biggs J., Tang C. Teaching for Quality Learning at University. 4th ed. Maidenhead: Open University Press, 2011;389 p.
10. Prince M. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*. 2004;93(3):223–231.
11. Freeman S., Eddy S.L., McDonough M., Smith M.K., Okoroafer N., Jordt H., Wenderoth M.P. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*. 2014;111(23):8410–8415.
12. Jain R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York: Wiley, 1991; 685 p.
13. Smith C.U., Williams L.G. Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software. Boston: Addison-Wesley, 2002; 368 p.
14. Beyer B., Jones C., Petoff J., Murphy N.R. Site Reliability Engineering. Sebastopol: O'Reilly, 2016; 552 p.
15. Kitchenham B.A., Dyba T., Jorgensen M. Evidence-Based Software Engineering. Proceedings of the 26th International Conference on Software Engineering. 2004:273–281.
16. Basili V.R., Caldiera G., Rombach H.D. The Goal Question Metric Approach. In: Marciniak J.J., editor. Encyclopedia of Software Engineering. New York: Wiley; 1994; p. 528–532.
17. Gregg B. BPF Performance Tools. Boston: Addison-Wesley, 2019; 480 p.
18. Lister R. After the Gold Rush: Toward Sustainable Scholarship in Computing. Proceedings of the 10th Australasian Computing Education Conference. 2008:3–9.
19. Beck K. Extreme Programming Explained: Embrace Change. 2nd ed. Boston: Addison-Wesley, 2004; 224.
20. Felder R.M., Brent R. Teaching and Learning STEM:A Practical Guide. San Francisco: Jossey-Bass, 2016. 336 p.
21. Kolb D.A. Experiential Learning: Experience as the Source of Learning and Development. 2nd ed. Upper Saddle River: Pearson, 2015. 400 p.
22. Godwin A., Ohland M.W., Long R.A., Oakes W.C., Scoresby J., Zhu D., Hollis D. Engineering Identity Development Scale (EIDS). Proceedings of the 122nd ASEE Annual Conference and Exposition. 2015. Paper ID 13751. 15 p.
23. Falkner K., Vivian R., Falkner N.J.G. Identifying computer science self-regulated learning strategies. Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education. 2014:291–296.
24. Guzdial M. Learner-Centered Design of Computing Education: Research on Computing for Everyone. San Rafael: Morgan & Claypool Publishers; 2015.
25. Cooper S., Grover S., Guzdial M., Simon B. A future for computing education research. *Communications of the ACM*. 2014;57(11):34–36.

#### Сведения об авторе:

Исабекова Тамила Илахидиновна, кандидат физико-математических наук, доцент, заведующий кафедрой прикладной математики и информатики, [mila775@mail.ru](mailto:mila775@mail.ru); ORCID 0000-0002-2970-594X

#### Information about author:

Tamila I. Isabekova, Cand. Sci.(Physical and Mathematical), Assoc. Prof., Head of the Department of Applied Mathematics and Informatics, [mila775@mail.ru](mailto:mila775@mail.ru);ORCID 0000-0002-2970-594X

#### Конфликт интересов/Conflict of interest.

Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.

Поступила в редакцию/Received 10.10.2025.

Одобрена после рецензирования/Reviced 29.11.2025.

Принята в печать/Accepted for publication 16.01.2026.