

УДК 378.147

*Макарчук Александра Васильевна, кандидат технических наук,
Государственный университет морского и речного флота имени
адмирала С. О. Макарова, г. Санкт-Петербург*

*Макарчук Наталия Васильевна, кандидат технических наук
Государственный университет морского и речного флота имени
адмирала С. О. Макарова, г. Санкт-Петербург*

СРАВНИТЕЛЬНАЯ ЭФФЕКТИВНОСТЬ КАНАЛОВ ОБРАТНОЙ СВЯЗИ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ: ТЕКСТ, ГОЛОСОВОЕ СООБЩЕНИЕ И ВИДЕОКОММЕНТАРИЙ

Аннотация: В статье представлены результаты сравнительного педагогического эксперимента по оценке эффективности трех каналов обратной связи (текстовый комментарий, голосовое сообщение, короткий видеокomментарий) в процессе обучения программированию студентов вуза. На основе анализа ответов 48 студентов направления «Информационные системы и технологии» и «Прикладная информатика» выделены критерии эффективности применительно к специфике задач по программированию. Показано, что выбор оптимального канала зависит от типа ошибки в коде. Предложены практические рекомендации для преподавателей программирования.

Ключевые слова: обратная связь, обучение программированию, каналы коммуникации, текстовый комментарий, голосовое сообщение, видеокomментарий, формирующее оценивание, высшее образование.

Abstract. The article presents the results of a comparative pedagogical experiment on evaluating the effectiveness of three feedback channels (textual commentary, voice message, and short video commentary) in the process of teaching programming to university students. Based on the analysis of the responses of 48 students in the fields of Information Systems and Technologies and Applied Informatics, the criteria for effectiveness were identified in relation to the specific

tasks of programming. It was shown that the choice of the optimal channel depends on the type of error in the code. Practical recommendations for programming teachers were proposed.

Key words: *feedback, programming education, communication channels, text commentary, voice message, video commentary, formative assessment, higher education.*

1. Введение

Обучение программированию имеет ряд особенностей, которые делают проблему обратной связи особенно острой. Данная связь может быть очень важна для развития компетенций в программировании, особенно у студентов-новичков. При этом среди экспертов часто отсутствует общая точка зрения относительно того, когда и как давать обратную связь при проверке и оценивании работ студентов при выполнении ими задач по программированию [1,2].

В условиях цифровой образовательной среды преподаватель программирования имеет доступ к нескольким техническим каналам доставки обратной связи: в LMS Moodle (система управления обучением), электронной почте, мессенджере. Он может отправить текстовый комментарий, голосовое сообщение или короткий видеокomментарий (скринкаст с голосом). Использование мессенджеров в сочетании с LMS Moodle помогает организовать эффективную коммуникацию в обучении, но требует методического обоснования выбора канала.

Проблема исследования: при каких условиях каждый из каналов обратной связи (текст, голос, видео) демонстрирует максимальную педагогическую эффективность при обучении программированию?

Цель статьи - представить результаты сравнительного эксперимента с участием студентов-программистов и сформулировать практические рекомендации по выбору канала обратной связи в зависимости от типа ошибки в коде.

2. Организация и методы исследования

Исследование проводилось на базе Государственного университета морского и речного флота имени адм. С.О. Макарова в период сентябрь–декабрь 2025 года. В нем приняли участие 48 студентов 2–3 курсов направления «Информационные системы и технологии» и «Прикладная информатика» (12 женщин, 36 мужчин, средний возраст 20,1 лет) очного обучения. Все студенты на момент эксперимента завершили изучение основ языка Python и имели опыт выполнения лабораторных работ.

Студенты были разделены на три равные группы по 16 человек, сопоставимые по успеваемости (средний балл за предыдущий семестр — 4,1–4,3).

Платформа для коммуникации. Базовым каналом взаимодействия выступала LMS Moodle, где размещались материалы лабораторных работ, формулировки заданий, критерии оценивания и итоговые оценки. Студенты загружали в LMS Moodle выполненные лабораторные работы. Для доставки текстовых комментариев от преподавателя использовалась встроенная в LMS Moodle система обратной связи (поля для комментариев в работах студентов). Голосовые сообщения и видеокomментарии доставлялись через мессенджер мессенджер Max, который использовался как дополнительный канал для оперативной коммуникации.

Типы учебных задач (специфика программирования). На основе классификации типов ошибок в программировании [1] были сформулированы четыре типа задач для эксперимента, основанные на этих типах:

Таблица 1

Классификация типов ошибок в программировании

Тип ошибки	Описание	Пример
Синтаксическая ошибка	Нарушение правил языка программирования	Пропущенное двоеточие, непарная скобка
Логическая ошибка	Алгоритм работает, но дает неверный результат	Неправильное условие в цикле, ошибка в формуле

Стилистическая ошибка/рефакторинг	Код работает, но не соответствует стандартам	«Магические числа», неинформативные имена переменных
Алгоритмическая доработка	Требуется изменение подхода к решению	Неэффективный алгоритм, требующий оптимизации

Каждый студент в течение семестра получал обратную связь по каждому типу задач через разные каналы (сбалансированный дизайн). После получения обратной связи студенты заполняли анкету по четырем критериям (по 5-балльной шкале):

- 1. Скорость восприятия - насколько быстро понял(а) суть замечания.*
- 2. Точность понимания - насколько точно понял(а), что именно нужно исправить.*
- 3. Ясность инструкции - насколько понятно сформулировано замечание.*
- 4. Готовность к доработке - насколько повысилась мотивация исправить код.*

3. Результаты эксперимента

3.1. Общая эффективность каналов

Усредненные данные по всем типам задач показали следующие результаты (max = 5):

Таблица 2

Общая эффективность каналов

Канал	Скорость восприятия	Точность понимания	Ясность инструкции	Готовность к доработке
Текст	4,3	4,4	4,2	3,7
Голос	4,0	4,1	3,8	3,9
Видео	4,7	4,8	4,6	4,6

Предварительный вывод: текстовый комментарий показывает хорошие результаты для определенных типов задач. Видеокомментарий оказался лучшим по всем критериям. Голосовое сообщение проиграло по всем позициям.

3.2. Взаимодействие канала и типа задачи

Для синтаксических ошибок:

Текст показал наивысшую скорость восприятия (4,8) и точность понимания (4,9). Студенты отмечали: «для исправления одной строчки кода проще прочитать текстовый комментарий, чем слушать голосовое сообщение или смотреть видео». Голосовые сообщения и видео воспринимались как «избыточные»

Для логических ошибок:

Видеокомментарий (скринкаст с демонстрацией выполнения кода) оказался наиболее эффективным (точность понимания — 4,9 и ясность инструкции 4,8). Голос без визуализации показал наихудшие результаты — студенты жаловались, что «на слух сложно запомнить последовательность действий». Текст оставался приемлемым (4,2), но требовал более детального описания.

Для рефакторинга и стилистических замечаний:

Текстовый комментарий с конкретными примерами показал лучшие результаты (точность понимания — 4,7). Видео также эффективно, но избыточно для простых стилистических правок.

Для алгоритмической доработки (оптимизации):

Видеокомментарий с объяснением альтернативного подхода и разбором алгоритмов показал максимальную готовность к доработке (4,9). Студенты отмечали: «когда преподаватель запускает код и видно, что он тормозит на больших объемах данных, а потом показывает оптимизированный вариант, то понимаешь в чем разница. Без видео не понимаешь масштаба проблемы».

4. Обсуждение

Полученные результаты показывают, что информативная обратная связь при обучении программированию должна быть не просто указанием на ошибку, но содержать объяснение причин и путей исправления [2, 4].

Исследования показывают, что использование мессенджеров в сочетании с LMS Moodle помогает организовать непрерывную коммуникацию в процессе обучения программированию. В данном эксперименте

использовался мессенджер Max для доставки голосовых сообщений, видео и оперативного обмена уточнениями.

Важным результатом является то, что единого «лучшего» канала не существует. Это подтверждает выводы о том, что отсутствие консенсуса среди экспертов относительно формата обратной связи имеет объективные основания, связанные с типом задачи [1].

Отдельного внимания заслуживает методика взаимного рецензирования [3,4]. Она представляет собой процесс, при котором студенты проверяют и комментируют учебные работы друг друга. Реализуется данная методика следующим образом: студент А знакомится с кодом студента Б, находит синтаксические, логические или стилистические ошибки, оставляет комментарии с предложениями по улучшению, после чего авторы дорабатывают свои работы на основе полученных замечаний. Для автоматизации этого процесса могут использоваться специализированные платформы. GitHub и GitLab позволяют оставлять комментарии непосредственно к строчкам кода [4].

Педагогические преимущества взаимного рецензирования в обучении программированию:

- 1. Формирование навыка чтения чужого кода. В профессиональной деятельности программист тратит значительную часть времени на анализ и доработку кода, написанного другими разработчиками.*
- 2. Развитие критического мышления и аргументации. Студент учится не просто констатировать «это плохо», а объяснять причины ошибки и предлагать конкретные пути исправления, что закрепляет его собственные знания.*
- 3. Снижение нагрузки на преподавателя. В группе из 30–40 человек преподаватель физически не может предоставить развёрнутую обратную связь каждому студенту по каждой лабораторной работе. Методика взаимного рецензирования распределяет эту нагрузку между*

студентами, при этом преподаватель сохраняет контроль за качеством рецензий.

4. Психологическая адаптация к профессиональной среде. Умение спокойно воспринимать критику своего кода и конструктивно на неё реагировать - обязательный навык для работы в IT-команде.

5. Практические рекомендации для преподавателей программирования

На основе полученных данных предлагается следующее правило выбора канала обратной связи:

Таблица 3

Выбор канала связи

Тип ошибки	Рекомендуемый канал	Обоснование
Синтаксическая ошибка	Текст (короткий, с указанием строки кода)	Максимальная скорость восприятия, не требует просмотра видео
Логическая ошибка	Видео (скринкаст с выполнением кода)	Визуализация выполнения помогает понять, где алгоритм дает сбой
Рефакторинг / стиль	Текст (с примерами «плохо → хорошо»)	Легко копировать примеры, можно оформить в виде таблицы
Алгоритмическая доработка	Видео (объяснение альтернативного подхода)	Требуется визуализация сложности и структуры данных

Дополнительные рекомендации:

1. *Комбинируйте каналы:* для сложной лабораторной работы можно дать текстовый комментарий по синтаксическим ошибкам и видеокomментарий по алгоритмическим недочетам.
2. *Учитывайте уровень студента:* для начинающих программистов важнее пошаговые объяснения (лучше в видеоформате), тогда как более опытные студенты предпочитают текстовые комментарии.
3. *Используйте возможности LMS Moodle в сочетании с оперативной коммуникацией через мессенджер Max.*
4. *Внедряйте методику взаимного рецензирования студентами как дополнительный канал. Это снизит нагрузку на преподавателя и будет*

формировать профессиональные навыки работы с чужим кодом у студентов.

6. Заключение

Новизна исследования заключается в том, что впервые было проведено сравнительное экспериментальное изучение эффективности текстового, голосового и видеоканалов обратной связи при обучении программированию с дифференциацией по типам ошибок в коде. В отличие от работ [3,4], фокусирующихся на методике взаимного рецензирования и технической интеграции LMS, в данной статье предложены практические рекомендации по выбору канала обратной связи в зависимости от типа учебной задачи. Текстовые комментарии остаются эффективными для синтаксических ошибок и замечаний по стилю кода. Видеокомментарии (скринкасты) незаменимы при объяснении логических ошибок и алгоритмической оптимизации. Голосовые сообщения без визуализации показали наименьшую эффективность при выполнении работ по программированию, что отражает специфику инженерного мышления.

Перспективы дальнейших исследований: изучение эффективности методики взаимного рецензирования в программировании; сравнение автоматизированной и человеческой обратной связи [5]; разработка методики обучения преподавателей использованию видео-обратной связи при проверке кода.

Использованные источники:

1. Lohr D., Kiesler N., Keuning H., Jeurig J. 'Ignore These Errors for Now' - How Experts Provide Feedback on Steps Novices Take Towards Solving Programming Problems // CompEd 2025 - Proceedings of the ACM Global Computing Education Conference. - Association for Computing Machinery, 2025. - P. 149-155. DOI: 10.1145/3736181.3747150
2. Kazamia C., Petropoulou O., Retalis S. CODE & REFLECT: Student Engagement with Automated Feedback in Programming Courses //

ICERI2025 Proceedings. - IATED, 2025. - P. 5688-5696. DOI: 10.21125/iceri.2025.1572

3. Кузнецов Д.И., Смирнова О.Л. Методика организации взаимного рецензирования кода при обучении программированию в вузе // Информатика и образование. - 2024. - № 5. - С. 42-51. - ISSN 0234-0453.
4. Иванов А.А., Петрова Е.В. Интеграция систем контроля версий с LMS Moodle в высшем образовании // Открытое образование. - 2025. - Т. 29. - № 2. - С. 23-31. - DOI: 10.21686/1818-4243-2025-2-23-31
5. Jung Y., Lee Y., Bae J., Kim D., Choi H., Kang M., Lee U. Exploring the Role of Automated Feedback in Programming Education: A Systematic Literature Review // arXiv:2602.00089. - 2026. - Режим доступа: <https://arxiv.org/abs/2602.00089>