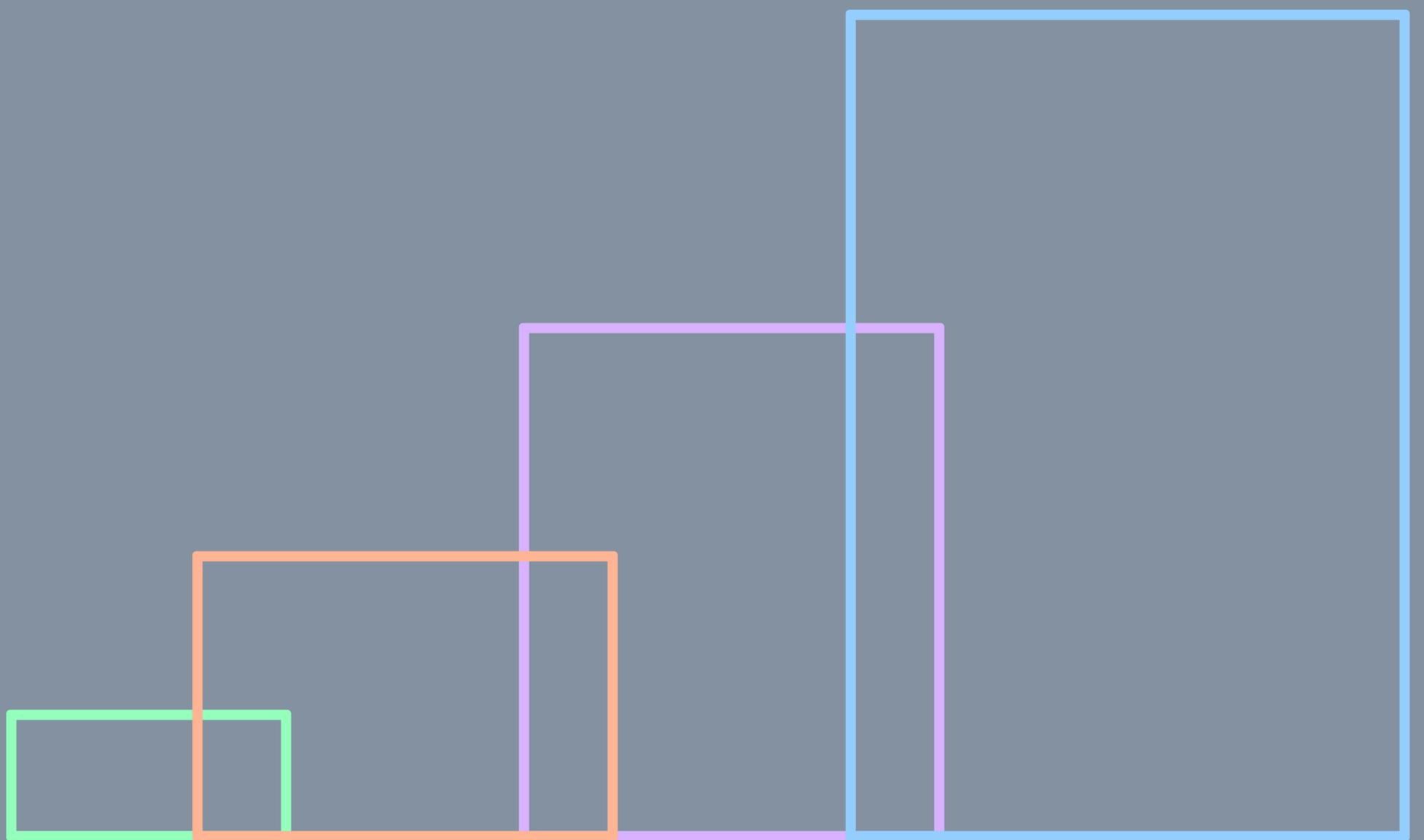


Искусственный интеллект в разработке: тренды и реальное применение



Содержание

- 3** Почему это исследование важно сейчас
- 7** О терминах и понятиях
- 8** Портрет аудитории
- 10** Эффективность разработчиков
- 19** Обзор инструментов для разработчиков
- 33** Взгляд в будущее: разработчики завтрашнего дня
- 37** Выводы

Почему это исследование важно сейчас

2021 год

появился первый массовый ИИ-ассистент, GitHub Copilot

GPT-4

модели уровня GPT-4 сделали генерацию кода доступной большинству разработчиков

Рынок ИИ-инструментов для разработки переживает взрывной рост, меняя наши представления об эффективности. Всё началось в 2021 году с GitHub Copilot, первого массового ИИ-ассистента. Вслед за ним появились Amazon CodeWhisperer, Google Gemini Code Assist и десятки других решений, а модели уровня GPT-4, Claude Sonnet и Qwen-Coder сделали генерацию кода доступной большинству разработчиков.

Следующий шаг в повышении эффективности — это уже не просто генерация кода, а полная интеграция ИИ в рабочую среду. Cursor и Windsurf встраивают его прямо в ядро IDE, превращая из дополнения в основу рабочего процесса. Claude Code идёт дальше — выступает уже как агент, который работает в терминале, понимает весь проект и способен выполнять задачи почти автономно.

Параллельно развиваются решения с открытым кодом — StarCoder, Codegen, DeepSeek Coder делают ИИ-технологии прозрачными и доступными для компаний любого размера. Low-code — и no-code-платформы — инструменты для создания приложений без глубокого знания программирования — интегрируют ИИ для автоматической генерации интерфейсов и бизнес-логики.

Почему это исследование важно сейчас

154 млрд \$

объём инвестиций
в ИИ-технологии
по всему миру
([IDC, 2023](#))

1,3 трлн \$

прогноз роста трат
на ИИ-технологии
к 2029 компаниями в мире
([IDC, 2025](#))

Масштаб трансформации подтверждают цифры. В 2023 году компании по всему миру потратили на ИИ-технологии более 154 млрд долларов ([IDC, 2023](#)). Согласно прогнозу IDC, к 2029 году эти расходы, во многом за счёт развития ИИ-агентов, достигнут 1,3 трлн долларов ([IDC, 2025](#)). Значительная часть инвестиций приходится на инструменты для разработки — от кодовых ассистентов до платформ полного цикла с интегрированным ИИ.

Бурный рост ИИ-инструментов заставляет переосмыслить само понятие продуктивности. В исследовании мы выяснили, как эти изменения влияют на эффективность и ежедневную работу разработчиков, какие инструменты действительно приживаются в командах и каким видят будущее профессии сами программисты.

«Когда мы говорим об ИИ в разработке, правильно разделять долгосрочное видение и практическое применение — выгоду от ИИ прямо сейчас. В перспективе ИИ — полноценный цифровой коллега-разработчик, который может выполнить задание по короткому описанию, возвращаясь к вам несколько раз за уточнениями или для выполнения ручных действий. Но сегодня он скорее выступает в роли „экзоскелета“, который не заменяет человека, а многократно усиливает его возможности. Мы рассматриваем ИИ как инструмент для сокращения времени в цикле создания, развёртывания и сопровождения ПО и для упрощения рутины. В Яндексе наша цель — на 30% ускорить выполнение задач, связанных с написанием и отладкой кода, а задачи, связанные с поиском информации, — в 2 раза. Есть и другие части цикла, такие, как работа SRE и QA, где уже есть первые эксперименты и успехи. Для удобства командной работы наших клиентов мы предоставляем SourceCraft со встроенным ИИ-помощником. Это позволяет компаниям сократить затраты на разработку и ускорить вывод продуктов на рынок».



Дмитрий Иванов

Руководитель платформы
для разработки SourceCraft

«Исследования реальных практик использования ИИ-ассистентов и платформ в разных областях важны для понимания эффективных сценариев, а также ограничений и перспектив развития.

Мы целенаправленно анализировали не только опыт индустриальных программистов, но и работников науки и образования: университеты и исследовательские институты уже интегрируют ИИ-технологии в учебные программы, чтобы студенты получали практический опыт на современных инструментах и быстрее включались в реальные проекты — как корпоративные, так и опенсорс-проекты.

Результаты могут быть полезны и пользователям ИИ-решений, и их разработчикам и исследователям».



Николай Никитин

Кандидат технических наук, лидер сообщества ITMO Open Source, руководитель лаборатории автоматического машинного обучения Института искусственного интеллекта Университета ИТМО

О терминах и понятиях

Прежде чем погрузиться в результаты, определимся с ключевыми понятиями, которые встретятся в исследовании.

Платформа разработки

Программная среда с набором инструментов для создания, тестирования и развёртывания приложений.

Кодовый ассистент

Программа на основе ИИ, которая помогает писать код. Она предлагает продолжения фраз, исправляет ошибки, генерирует целые функции по описанию на естественном языке.

ИИ-агенты

Автономное программное обеспечение на базе искусственного интеллекта, способное самостоятельно анализировать запросы, принимать решения и выполнять действия.

Модель искусственного интеллекта

Компьютерная программа, обученная распознавать закономерности в данных и решать конкретные задачи. Языковые модели — частный случай, они работают с текстом как на естественных языках, так и на языках программирования.

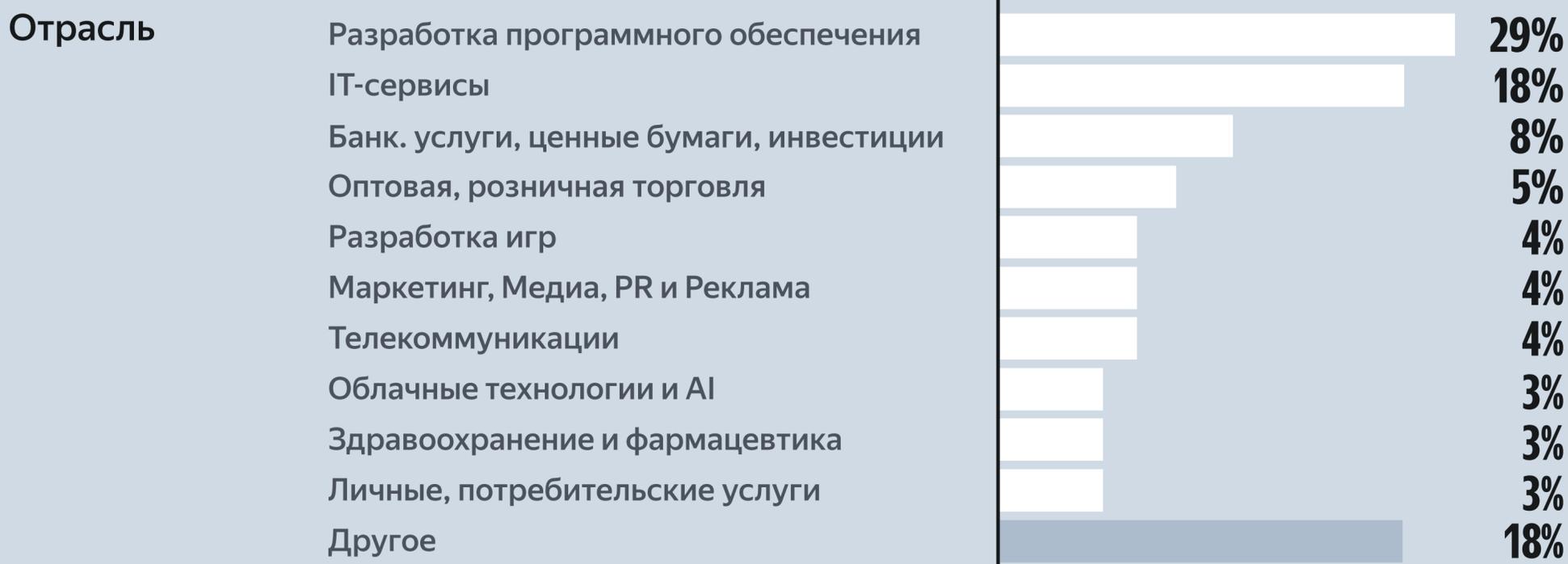
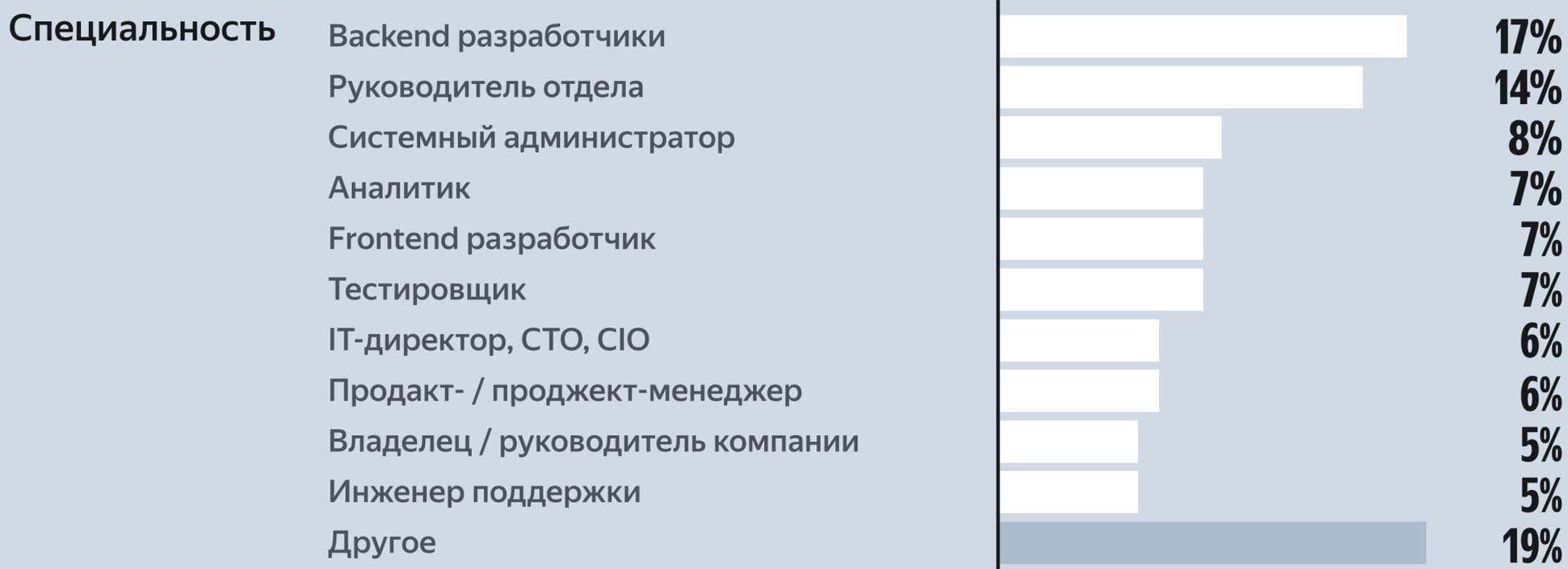
Опенсорс

Программное обеспечение с открытым исходным кодом. Любой может изучить, как оно работает, предложить улучшения или создать собственную версию.

Портрет аудитории

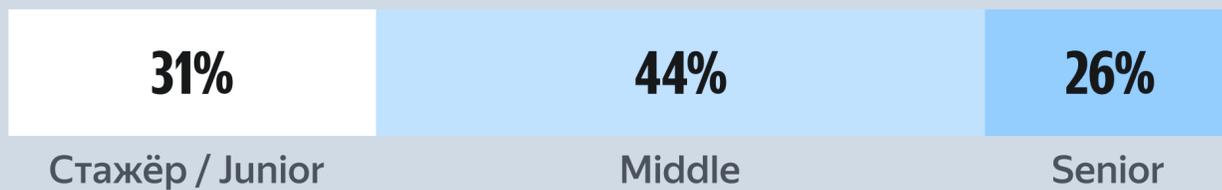
Чтобы получить полную картину, мы опросили три группы: практикующих IT-специалистов, преподавателей технических вузов и студентов, которые изучают программирование. Всего мы опросили **более 600 респондентов.**

Сегмент IT-специалисты: 524 респондента



Портрет аудитории

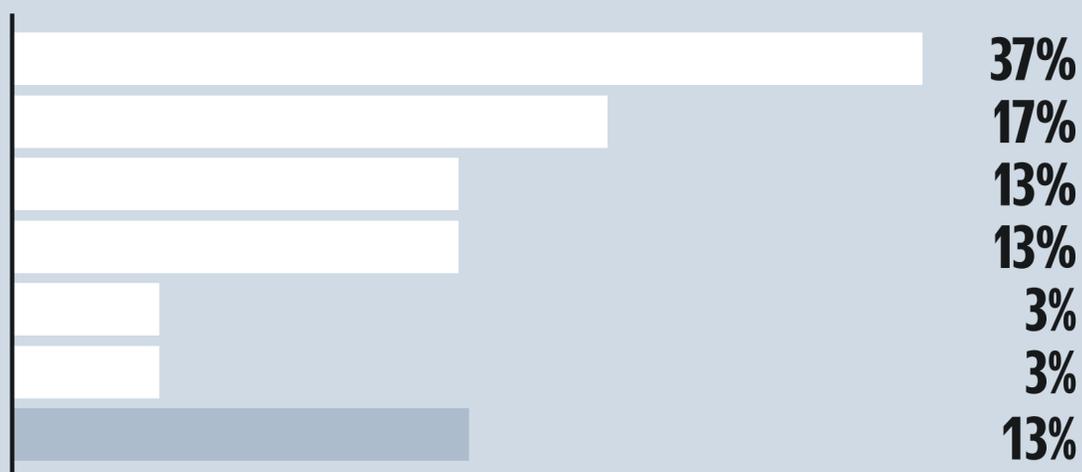
Грейд IT-специалиста



Сегмент сотрудники вузов, 30 респондентов

Должность

Ассистент, инженер
Доцент
Преподаватель
Научный сотрудник
Ректор/ проректор
Профессор
Другое



Студенты, 94 респондента

Отбирали по двум критериям: обучение в вузе и использование платформы разработки в учебном процессе.

Эффективность разработчиков

Мы спросили разработчиков, что больше всего влияет на их личную продуктивность.



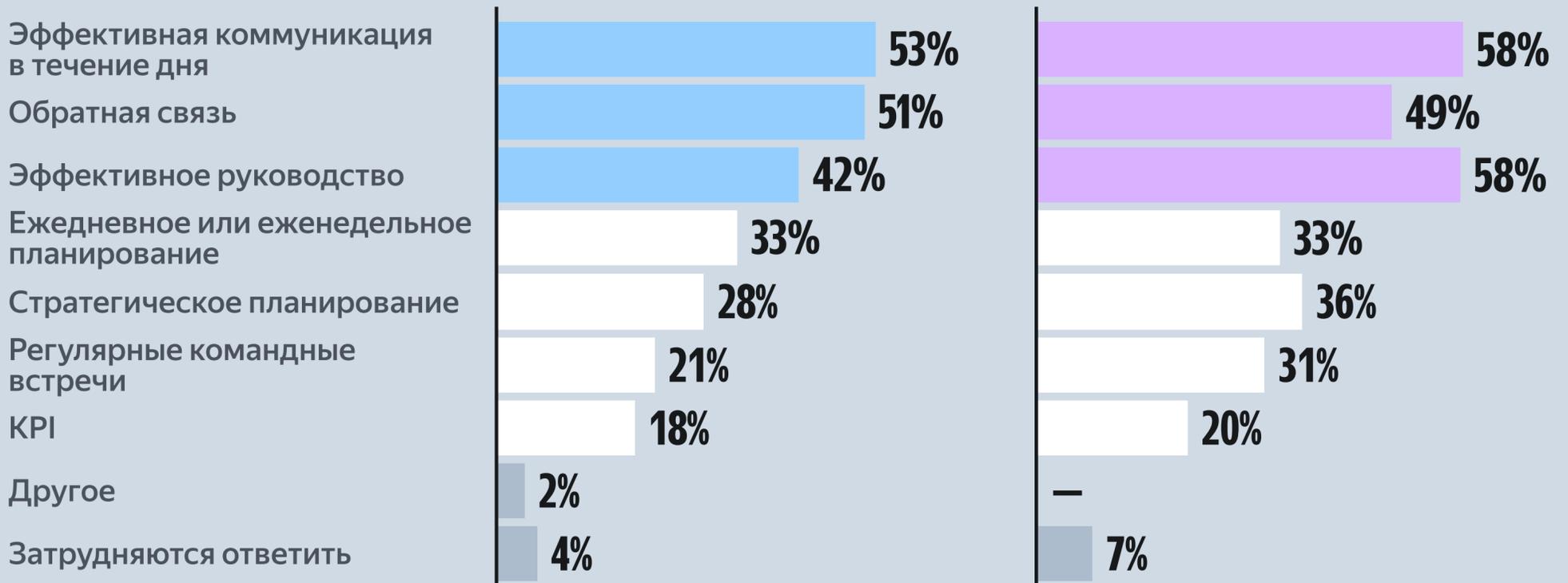
Топ-3 фактора: эффективная коммуникация в течение дня, обратная связь, грамотное руководство. Эти результаты подтверждают международные исследования. Согласно McKinsey, продуктивность сотрудников почти в пять раз выше, когда руководство детально и честно общается с ними по важным рабочим вопросам.

Опытные специалисты и руководители придают большое значение качеству управления — для них грамотный менеджмент является ключевым фактором продуктивности команды и влияет на конечный результат сильнее, чем KPI. Причина понятна: разработка ПО — это интеллектуальная и творческая деятельность, результаты которой трудно оценить простыми количественными метриками, и именно здесь на первый план выходит опыт и гибкость грамотного руководителя.

Что влияет на личную продуктивность

Все разработчики

Опытные специалисты и руководители



«Личная продуктивность складывается из самочувствия разработчика, интересности задач и качества среды. Нельзя выделить что-то одно — высокая зарплата и крутая команда не помогут, если задачи неинтересные. Качественный менеджмент, который и задачи подберёт, и команду соберёт, — точно поможет».

Алексей Смирнов

Основатель
CodeScoring

«Продуктивность зависит в первую очередь от грамотно поставленной задачи, затем — от качества инструментов разработки и dev-окружения. Платформы помогают удобно решить широкий спектр задач, а кодовые ассистенты берут на себя рутину».

Фёдор Сазонов

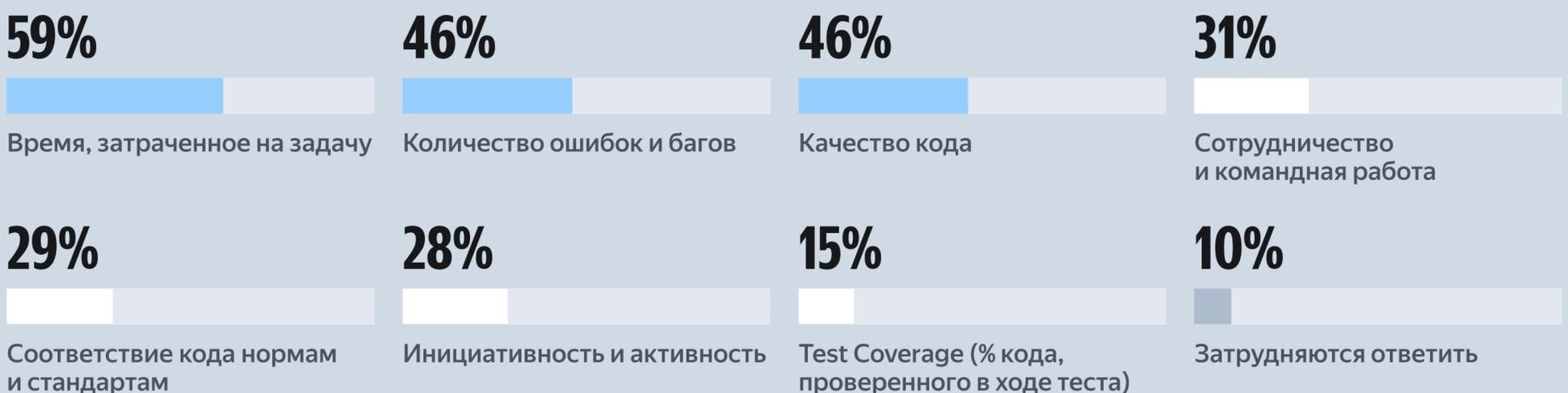
CEO OpenIDE

Важно, что **80% разработчиков** часто или постоянно работают в проектной команде. Современные инструменты разработки — от кодовых ассистентов до платформ — становятся средой, которая может сделать коммуникацию в команде прозрачнее и ускорить обратную связь. В частности, платформы разработки создают единое пространство для асинхронной коммуникации, упрощая поиск информации и координацию внутри команды.

Как измеряют эффективность команд

Главный критерий — время выполнения задач. За ним следуют качество кода и количество багов. Эти метрики понятны бизнесу и относительно легко измеримы.

Критерии эффективности работы



Эффективность разработчиков

На эти показатели напрямую влияют платформы разработки. Правильно подобранные инструменты не только помогают уменьшить число багов (68%) и повысить качество кода (64%), но и, по мнению 52% респондентов, улучшают сотрудничество внутри команды.

Критерии эффективности работы



■ Значительно улучшают

■ Создают некоторые трудности

■ Я не замечаю изменений

■ Вносят некоторые улучшения

■ Больше мешают, чем помогают

■ Затрудняюсь оценить

«Платформы для разработки помогают удобно решать широкий спектр задач за счёт инструментов, входящих в них. Кодовые ассистенты берут на себя рутинные операции, что повышает эффективность разработчиков».

Фёдор Сазонов

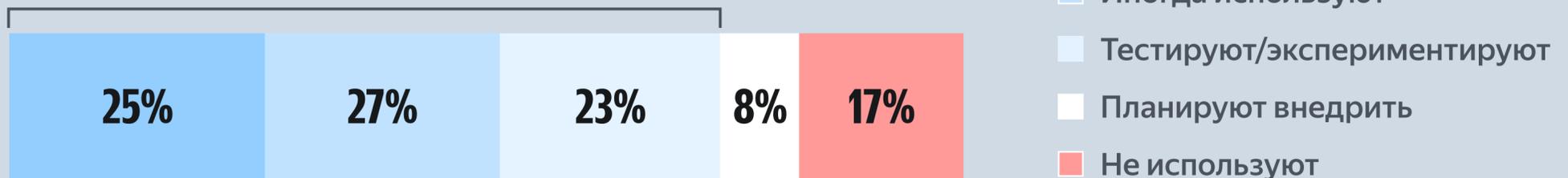
CEO OpenIDE

Роль ИИ в продуктивности

По данным нашего исследования, 75% опрошенных IT-специалистов уже используют кодовые ИИ-ассистенты. Это не просто эксперименты — инструменты стали частью ежедневной работы.

Использование кодовых ИИ-ассистентов в работе

75% используют или тестируют в работе



«По данным исследования Axiom JDK 2024 года, половина Java-разработчиков использует ИИ. Технология повышает продуктивность начинающих специалистов: их запросы, как правило, просты, и ИИ легко с ними справляется.

Опытным инженерам она помогает совмещать поиск информации с интерактивным диалогом. Но существуют ограничения. Передавать код сторонней нейросети небезопасно. Код, сгенерированный ИИ для новичка, требует тщательной проверки ведущими специалистами, что отнимает у команды дополнительное время. ИИ остаётся инструментом для профессионала, который понимает и дорабатывает результат».

«Если пользоваться ИИ с умом — влияние положительное. Например, в ситуациях, когда мысль „застряла“, он может подсказать возможные направления. Главное — не ограничиваться этими подсказками и не полагаться бездумно на интерфейсы ИИ. При правильном использовании ИИ приносит много пользы. Например, ИИ-ассистенты подходят при ревью и проверке безопасности кода — они упрощают рабочий процесс».

Илья Сазонов

Директор по продуктам Axiom JDK, направления Spring и OpenIDE

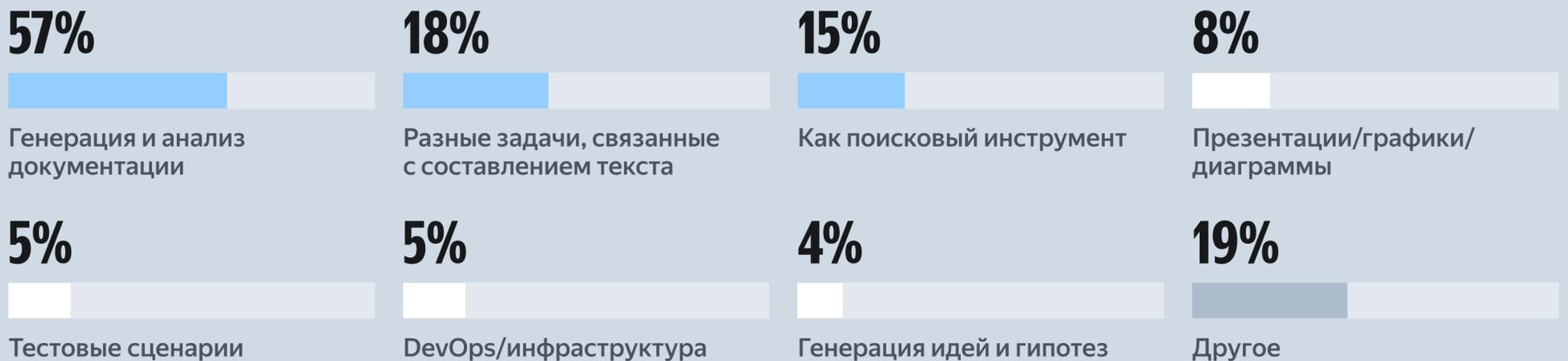
Алексей Смирнов

Основатель CodeScoring

Эффективность разработчиков: роль ИИ в продуктивности

По нашим данным мы видим важный тренд — массовое внедрение ИИ-ассистентов для помощи в написании кода в России. Каждый четвёртый разработчик (27%) применяет ИИ не только для написания кода. Часто — для генерации и анализа документации и работы с техническими текстами.

Области применения ИИ



«Эффективность IT-команд — это соотношение „польза/затраты“ в единицу времени, причём пользу можно измерять в деньгах, которые заработал бизнес, или в числе людей, которым результат облегчает жизнь. Личная эффективность разработчика — произведение „талант × мотивация × эффективность инструментов“. Платформы и ИИ-ассистенты кратно усиливают последний множитель, но не заменяют первые два. Главное — не делегировать мышление машине полностью: в критически важном ПО это фатально. ИИ повышает эффективность на порядки, как современные языки программирования превосходят ассемблер, но творческий замысел остаётся в руках архитектора — ИИ лишь усиливает возможности человека».

Владимир Рубанов

Член правления
Руссофт

«Мы находимся в процессе изменения подходов к разработке на всех этапах SDLC — от написания требований и документации до создания кода и вывода продуктов в продакшн. Все фазы так или иначе затронуты развитием ИИ и LLM. Вопрос “Использовать или не использовать ИИ?” сейчас не стоит — на него уже получен исключительно положительный ответ. Теперь задача заключается в максимальном расширении области применения и перестройки процессов на новые рельсы. Чем быстрее будет происходить адаптация людей, обнаруживаться новые и новые сценарии использования, тем скорее мы будем видеть результаты на уровне отдельных организаций и индустрии в целом».



Евгений Колесников

Менеджер продукта
SourceCraft Code Assistant

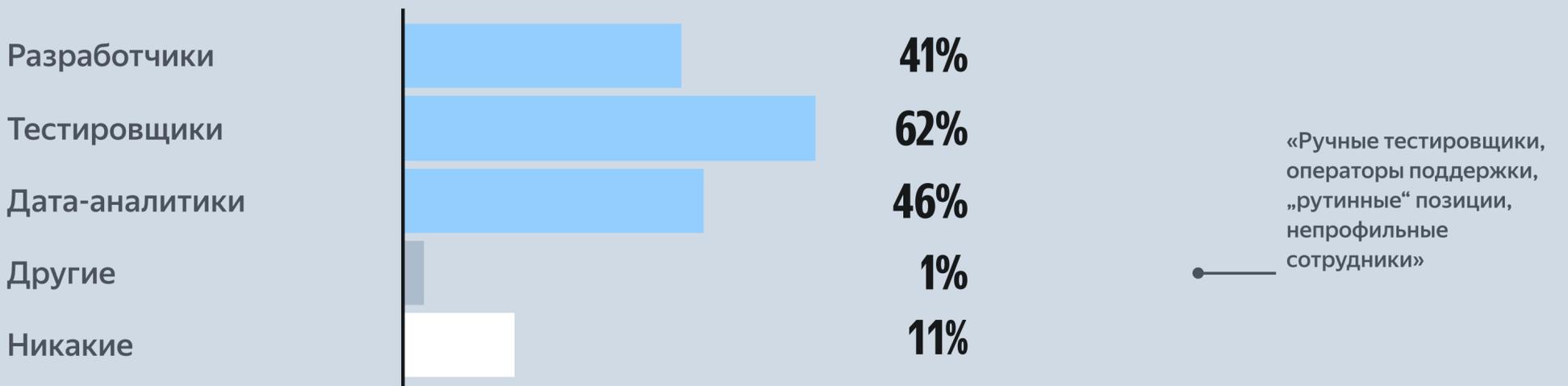
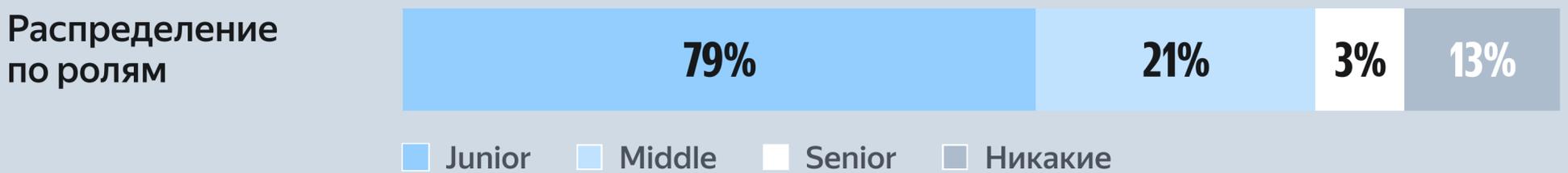
Эффективность разработчиков: влияние ИИ на рынок труда

Влияние ИИ на рынок труда

Только 6% IT-специалистов верят, что ИИ может автоматизировать конкретные задачи разработки. Большинство (61%) ожидают частичной автоматизации рутины.



Потенциал для автоматизации различается по ролям и уровням. 62% респондентов думают, что легче всего автоматизировать задачи тестировщиков, 46% — задачи дата-аналитиков, 41% — задачи разработчиков. Если говорить об уровнях специалистов, то больше всего изменения затрагивают джунов — 79% опрошенных видят влияние ИИ на работу начинающих специалистов.



Мнения о будущем рынка разделились. Часть специалистов видит сокращение рабочих мест, другие уверены — ИИ изменит именно характер работы. «Рынок адаптивен. ИИ не оставит человека без работы, просто пройдет переквалификация персонала», — отмечает один из респондентов.

Однако изменение характера работы — это не только новые возможности, но и новые вызовы. Например, возрастает вероятность «вайб-хакинга» — ситуации, когда младший специалист с помощью ассистента пишет формально рабочий, но стилистически или архитектурно чуждый проекту код. В конечном счёте, как подчёркивает другой эксперт, «человек без навыков программирования никогда не сделает с помощью ИИ что-то действительно классное».

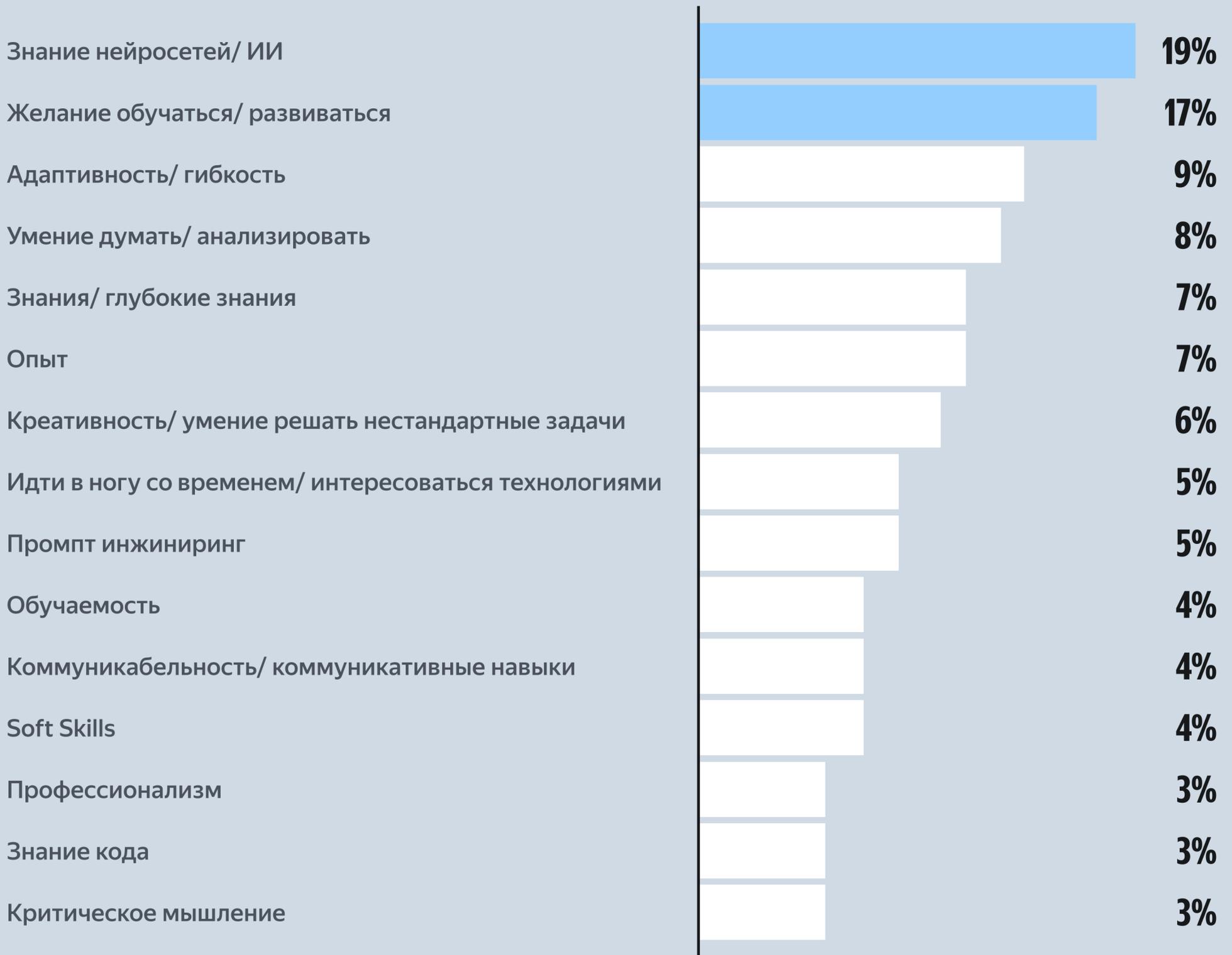
Эффективность разработчиков: влияние ИИ на рынок труда

Опрошенные единодушны в главном — порог входа в профессию растёт:

«То, что вчера хотели от мидла, сегодня хотят от джуна.
Низкого порога входа уже давно нет».

Чтобы оставаться востребованным, IT-специалисту нужно изучать ИИ и интегрировать его в работу, постоянно осваивать новое, развивать адаптивность, углублять экспертизу и расширять кругозор в смежных областях.

Необходимые навыки (открытый вопрос)



«Умение использовать ИИ в работе становится базовым навыком программиста. В ближайшее время мы увидим адаптацию процессов собеседований на рынке — компании будут проверять уровень владения новыми инструментами. Мы внутри Яндекса уже экспериментируем с устройством таких этапов отбора».



Сергей Бережной

Директор по взаимодействию с разработчиками, Яндекс

Обзор инструментов для разработчиков: платформы разработки

Теперь рассмотрим конкретные инструменты, которые помогают командам работать эффективнее.

Потенциал рынка

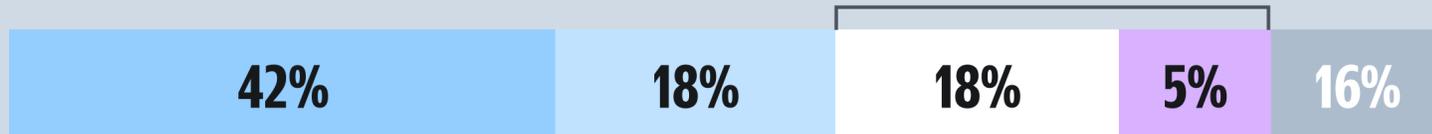
23% компаний, которые пока не используют платформы разработки, планируют внедрить их в течение года. Это показывает, что рынок продолжает расти.

Среди тех, кто уже использует платформы, каждая вторая компания рассматривает переход на новое решение в ближайший год. При этом 35% удовлетворены текущими инструментами и не планируют миграцию. Остальные либо не определились, либо имеют внешние причины для перехода — от смены вендора до изменения корпоративных требований.

Рассматриваете ли возможность внедрения/миграции на платформу разработки?

23% рассматривают возможность использования платформ

Не используют платформы разработки



- Нет, не планируют
- Нет, но рассматривают альт. решения
- Возможно, зависит от бюджета/ других проектов/ стейкхолдеров
- Да, изучают рынок и варианты
- Затрудняются ответить

50% рассматривают возможность тестирования/перехода в ближайший год

Уже используют платформы разработки



- Нет, устраивает текущее решение
- Возможно, зависит от бюджета/ других проектов/ стейкхолдеров
- Да, рассматривают российскую
- Да, рассматривают международную
- Да, рассматривают международную и российскую
- Затрудняются ответить

Обзор инструментов для разработчиков: платформы разработки

Главные причины для смены платформы — **новая функциональность, удобство использования и встроенный ИИ**. Снижение затрат тоже учитывают, но реже — только 14% компаний считают это приоритетом.

Для каких возможностей готовы перейти на новую платформу? (открытый вопрос)

Функциональность и тех. характеристики	41%	Удобство использования	19%
Функционал	10%	Удобно/ простота/ удобство использования	12%
ИИ возможности	7%	Удобный интерфейс	3%
Надёжность, стабильность	6%	Хорошая поддержка и удобная документация	4%
Широкий спектр возможностей	5%		
Скорость работы	5%		
CI/CD	3%		
Автоматизация процессов	3%		
Масштабируемость	1%		
Качество	1%		
Финансовые причины	15%	Российское ПО и безопасность	13%
Цена	12%	Российская разработка	4%
Хорошие условия/ бесплатный пробный период	2%	Безопасность/ хранение данных в России	3%
Удобство оплаты	1%	Санкции/ отсутствие санкций	3%
		Интеграция с российскими системами	3%

Обзор инструментов для разработчиков: платформы разработки

Интерес к платформе и её узнаваемость — ключевые факторы при выборе. Разработчики предпочитают проверенные решения с активным сообществом.

Что мотивирует к рассмотрению платформ ? (открытый вопрос)



Обзор инструментов для разработчиков: платформы разработки

Самые популярные платформы

GitHub и GitLab лидируют с большим отрывом — в сумме их используют 75% опрошенных. Локальные платформы выбирают 17% компаний. Такой выбор часто продиктован не только соображениями безопасности, но и необходимостью соблюдать отраслевые стандарты, требования регуляторов в таких сферах, как финансовый сектор и медицина.

GitHub показывает один из самых высоких уровней удовлетворённости среди пользователей. На том же уровне оценивают Atlassian Bitbucket и GitLab.

Удовлетворённость платформами, топ-3 платформы



Обзор инструментов для разработчиков: платформы разработки

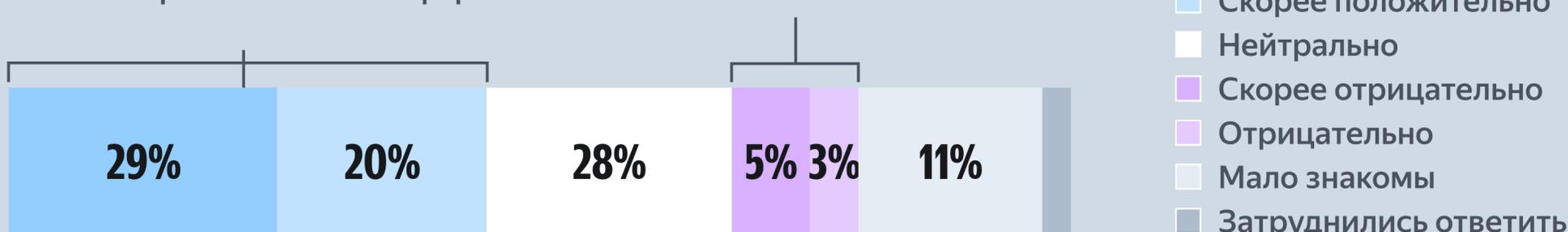
Отношение к российским платформам

Половина респондентов положительно относится к отечественным платформам разработки, 28% — нейтрально, только 8% настроены негативно.

Отношение к российским платформам

49% Положительно относятся к российским платформам

8% выразили негативное отношение



«На рынке существует более пяти российских платформ для разработки. Спрос есть, но российские компании предпочитают создавать инструменты самостоятельно на основе открытого исходного кода и с недоверием относятся к отечественным продуктам. В России, особенно при работе с кодом, преобладает модель локального развёртывания. Если отечественные платформы обеспечат необходимую гибкость и возможность интеграции с привычными инструментами — их ждёт успех».

**Алексей
Смирнов**

Основатель
CodeScoring

«Рынок находится в самом начале своего пути. Скорее всего, на нём останется лишь несколько крупных конкурирующих решений, а остальные компании либо объединятся, либо уйдут с рынка».

**Фёдор
Сазонов**

CEO OpenIDE

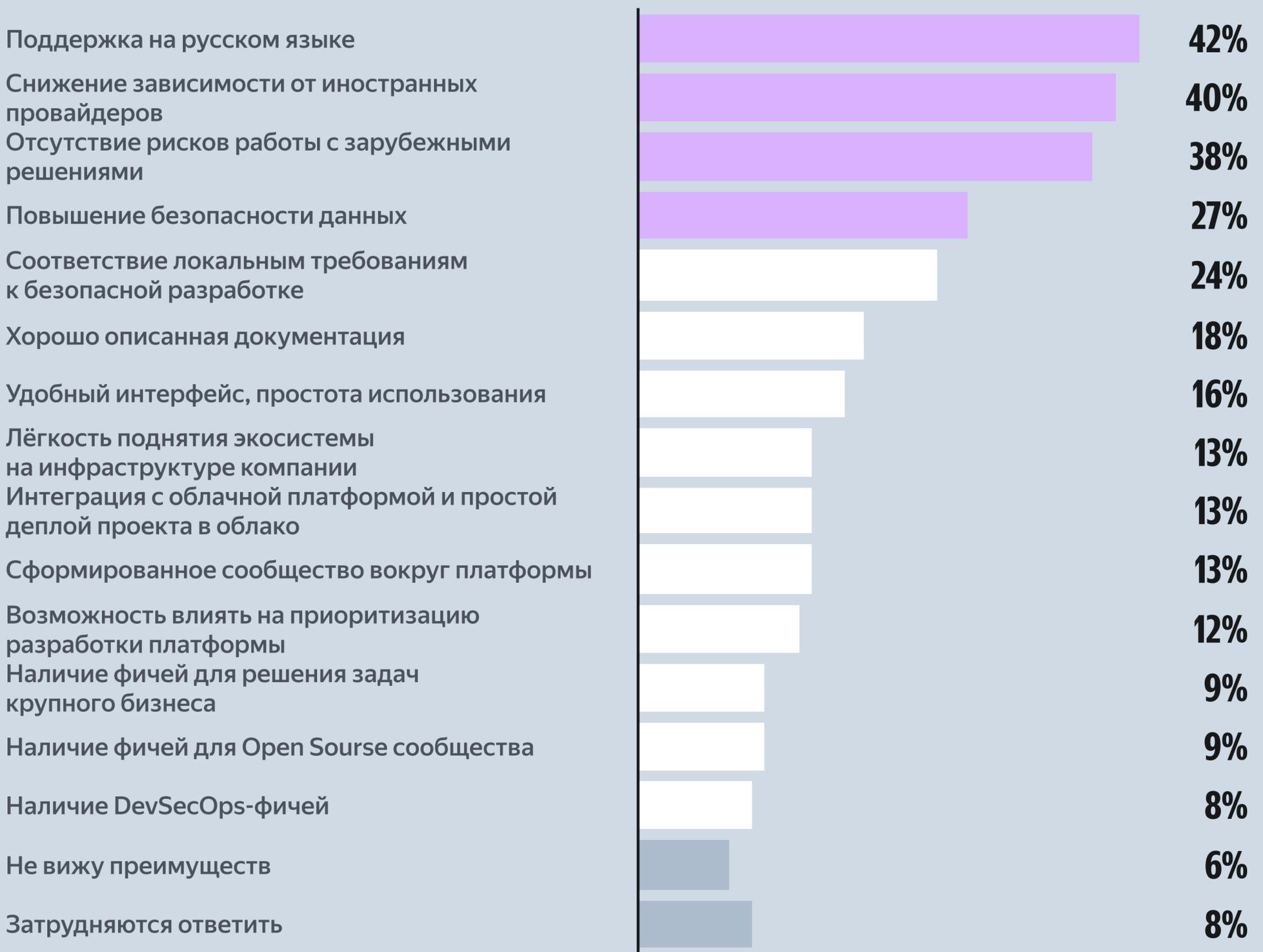
Отношение к российским платформам формируется на стыке двух мнений. С одной стороны, скептики указывают на недостаточную зрелость решений по сравнению с мировыми лидерами. С другой — для половины компаний преимущества отечественных продуктов уже перевешивают недостатки.

Обзор инструментов для разработчиков: платформы разработки

Согласно исследованию, ключевыми критериями, по которым компании готовы переходить на российские платформы, являются:

- **Геополитическая надёжность.** Независимость от внешних вендоров и снижение санкционных рисков становятся для многих решающим аргументом.
- **Информационная безопасность.** Соответствие локальным требованиям регуляторов и возможность полного контроля над данными внутри страны.
- **Удобство использования и поддержки.** Наличие документации и квалифицированной технической поддержки на русском языке значительно упрощает работу команд.

Преимущества использования российских платформ разработки



Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

Инструменты для работы с кодом делятся по способу интеграции: плагины для IDE и AI-first редакторы. Параллельно развиваются автономные ИИ-агенты, которые берут на себя целевые задачи разработки.

1 Кодовые ассистенты

Действует в роли «штурмана»: он работает в интегрированной среде разработки (IDE), предлагает варианты завершения кода и отвечает на вопросы, при этом разработчик всегда остаётся «за штурвалом», полностью контролируя процесс.

2 ИИ-агент

Напоминает «младшего разработчика», которому можно делегировать задачу целиком и переключиться на другие дела. После постановки высокоуровневой цели — например, «реализовать аутентификацию через Google» — агент автономно анализирует код, создаёт решение, тестирует его и представляет на проверку уже готовый результат.

Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

Кодовые ассистенты

Они работают бок о бок с разработчиком прямо в редакторе кода, выступая в роли «штурмана» или напарника в парном программировании. Их главная задача — предлагать варианты кода, от автодополнения строк до написания целых функций, при этом оставляя финальное решение за человеком.

Сегодня на рынке доминируют два подхода к реализации таких ассистентов:

Плагины для IDE

Самый распространённый формат. ИИ-сервис встраивается в популярные редакторы кода, такие как VS Code или JetBrains.

GitHub Copilot

GitLab Duo

Google Gemini Code Assist

Amazon Q Developer

JetBrains AI Assistant

SourceCraft Code Assistant (плагин — часть платформы SourceCraft)

СберТех GigaCode

AI-first-редакторы

В этом случае ИИ — не дополнение, а ядро продукта. Это полноценные среды разработки, созданные с нуля вокруг возможностей искусственного интеллекта.

Cursor

Яркий пример такого подхода. Это форк (модификация) VS Code, который предлагает более глубокую интеграцию ИИ для написания, редактирования и анализа кода.

Windsurf

Ещё один форк VS Code, который позиционируется как прямой конкурент Cursor. Он предлагает схожий набор ИИ-функций, встроенных в ядро редактора, включая расширенный поиск и агентные возможности.

Replit

Популярная браузерная IDE, которая глубоко интегрировала ИИ в свой основной продукт. Его Replit AI Agent может создавать приложения по текстовому описанию, настраивать окружение и помогать с развёртыванием. Это яркий пример среды, где ИИ — центральный элемент, а не дополнение.

Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

ИИ-агенты

Наряду с ИИ-ассистентами набирает популярность и другой класс инструментов — ИИ-агенты. Их ключевое различие заключается в уровне автономности и способе взаимодействия с разработчиком.

Claude Code (Anthropic)

Агент, работающий в терминале. Он способен понимать весь контекст проекта, составлять план и выполнять сложные задачи: от рефакторинга до исправления багов.

Codex (OpenAI)

Официальный ИИ-агент от OpenAI, позиционируемый как «напарник по программированию». Он способен автономно перемещаться по кодовой базе, редактировать файлы, запускать команды и выполнять тесты для решения комплексных задач. Доступен через IDE, CLI и облачный интерфейс.

SWE-agent

Опенсорс-агент для автоматического исправления багов.

AI Scientist (Sakana AI)

Прототип для проведения научных исследований.

Devin (Cognition Labs)

первый ИИ-агент, способный полностью выполнять проекты по разработке.

Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

Как разработчики используют агентов сегодня

Глобальное исследование [Stack Overflow 2025](#) показывает, что, хотя ИИ-инструменты получили широчайшее распространение (их используют 84% респондентов), применение именно автономных ИИ-агентов пока нельзя назвать массовым.

Ключевые выводы исследования относительно ИИ-агентов:

Большинство разработчиков (52%) либо не используют ИИ-агентов, либо придерживаются более простых ИИ-инструментов. Исследование однозначно констатирует, что эта технология ещё не стала мейнстримом в профессиональном сообществе.

Высокая оценка эффективности. Несмотря на низкий уровень внедрения, те, кто уже использует агентов, отмечают их значительную пользу: 69% пользователей ИИ-агентов согласны с тем, что эти инструменты повысили их продуктивность.

Таким образом, хотя ИИ-агенты ещё не вошли в повседневную практику большинства разработчиков, опыт их текущего применения указывает на высокий потенциал технологии для оптимизации рабочих процессов.

Что используют разработчики

Среди ИИ-инструментов, помогающих в написании кода, лидируют общие LLM, среди специализированных:

GitHub Copilot

Cursor AI

Gemini Code Assist

Вне зависимости от базовой технологии, главные критерии оценки всех этих инструментов — качество генерируемого кода и экономия времени. Это подтверждает тезис о том, что ИИ-ассистент в первую очередь призван повысить продуктивность разработчика.

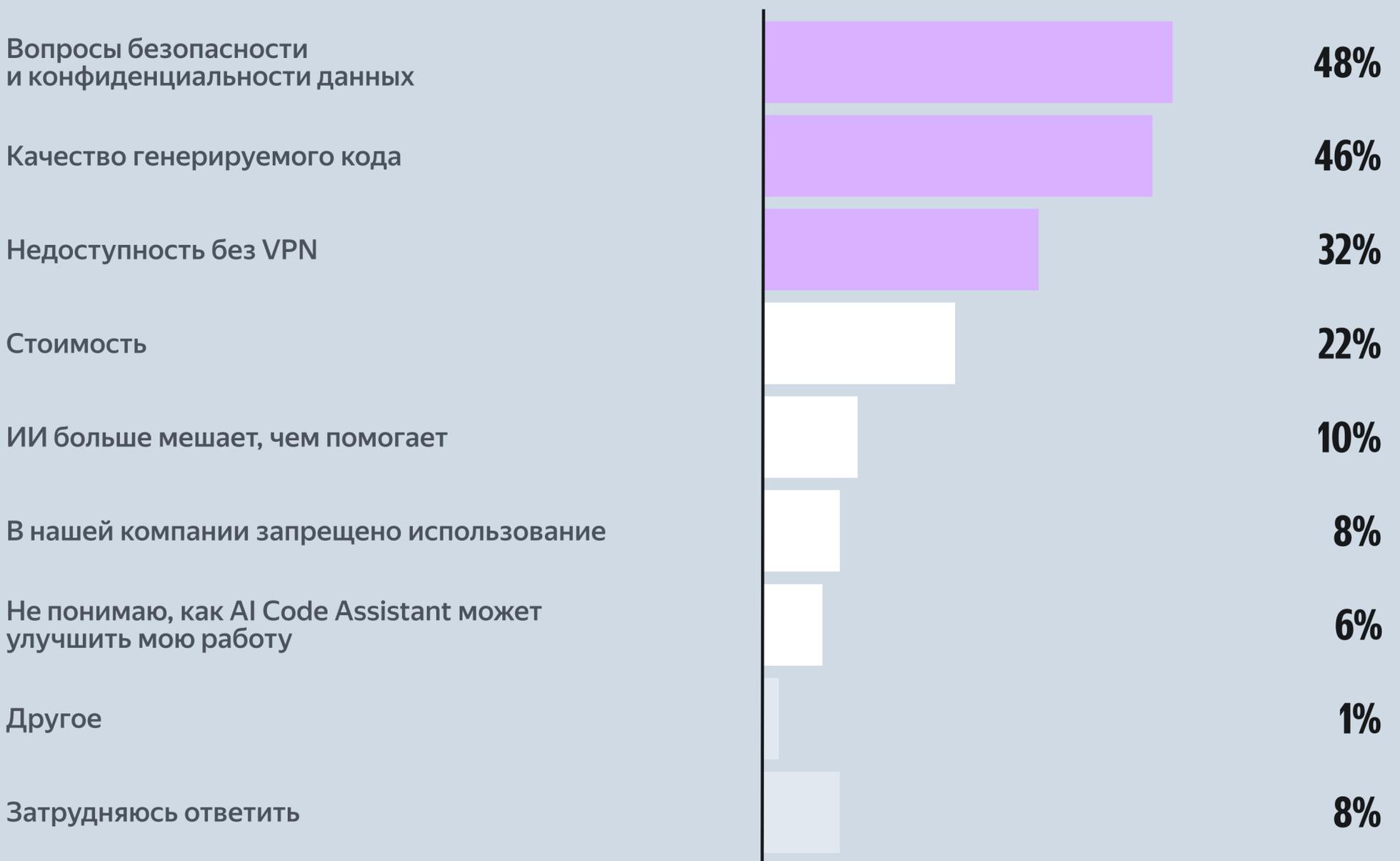
Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

Критерии оценки ассистентов



Основные барьеры для внедрения — **вопросы конфиденциальности данных и недостаточное качество генерируемого кода для сложных задач.**

Ограничения и риски при внедрении



«Универсальные языковые модели пока лидируют благодаря низкому порогу входа и удобному интерфейсу. Но специализированные инструменты, такие как GitHub Copilot, перспективнее — они решают комплексные задачи на уровне всего проекта. Главные барьеры для внедрения — не только конфиденциальность, но и ограниченная применимость в сложных задачах. Для непрофессиональных разработчиков, например учёных, важна способность ассистентов следовать лучшим практикам опенсорс-сообщества. Будущее за мультиагентными системами, которые декомпозируют сложные задачи на простые и используют специализированные инструменты».



Николай Никитин

Кандидат технических наук, лидер сообщества ITMO Open Source, руководитель лаборатории автоматического машинного обучения Института искусственного интеллекта Университета ИТМО

Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

«Сама по себе возможность генерации кода интригует и завораживает. Возможность создавать простые проекты или отдельные фрагменты кода, минуя ручной труд на написание юнит-тестов и наполнение базы знаний однотипными сигнатурами, — это огромный прорыв, экономящий силы и время. Но, на мой взгляд, создавать полноценные проекты и продукты с помощью ИИ-ассистентов пока невозможно. Поэтому я бы с осторожностью относился к инвестициям в этом направлении: для этого потребуется ещё один-два „квантовых скачка“ в развитии генеративных сетей. Впрочем, судя по темпам их развития, года через два ситуация может измениться. В то же время в более простых задачах — помощи с написанием кода, генерации простого кода и прототипировании — ИИ-ассистенты уже доказывают свою эффективность».

Денис Кораблёв

Управляющий директор, директор по продуктам
Positive Technologies

«Разработчики активно используют IDE с ИИ-ассистентами — Cursor AI, Windsurf, Claude Code. Ускорение заметно на всех этапах разработки: проще собрать прототип, чем писать 50-страничное ТЗ, и быстрее найти ошибку в коде. Stack Overflow уступил место ChatGPT. Многие ещё скептически относятся к кодовым моделям, но с каждым релизом игнорировать ИИ-инструменты становится сложнее».

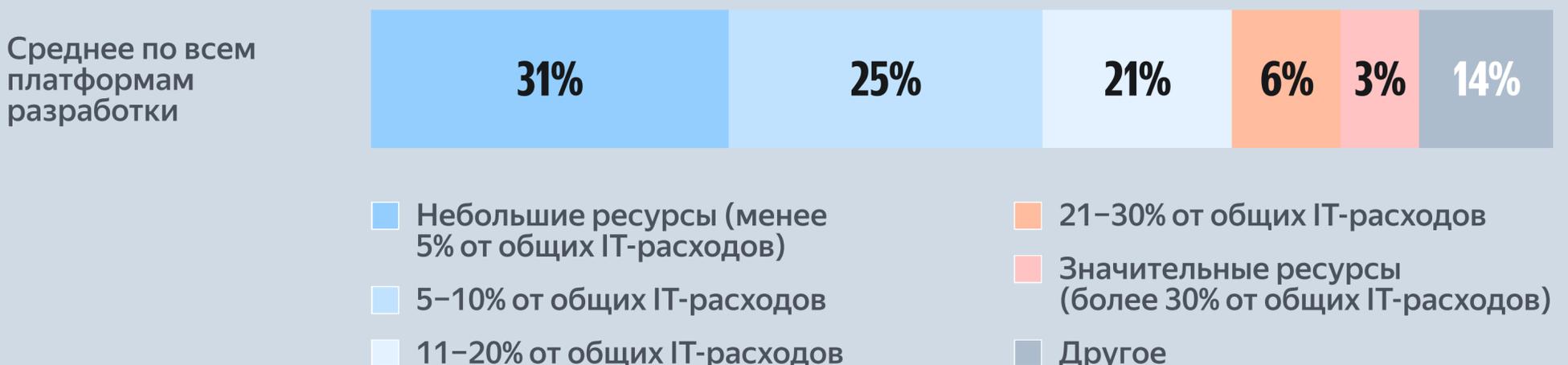
Евгений Кокуйкин

CEO HiveTrace.ai,
руководитель лаборатории AI Security, ИТМО

Инвестиции в инструменты разработки

Рассмотрим, сколько компании готовы тратить на современные инструменты.

В 56% случаев затраты на внедрение платформы составили до 10% от общих ИТ-расходов. Это показывает, что компании видят в инструментах разработки важную, но не доминирующую статью бюджета.



Обзор инструментов для разработчиков: кодовые ассистенты и ИИ-агенты

Для большинства компаний (около 75%) расходы на лицензии и модернизацию инструментов не превышают 10% IT-бюджета. Ответы респондентов говорят о том, что этой суммы достаточно для решения поставленных задач.

Только 12% компаний не планируют инвестировать в модернизацию инструментов и интеграцию ИИ. Это говорит о том, что большая часть рынка видит в этом направлении перспективу и выгоду для бизнеса. Две трети готовы выделить до 10% IT-бюджета на каждое направление.



Взгляд в будущее: разработчики завтрашнего дня

Чтобы понять, куда движется индустрия, мы изучили, как готовят новое поколение разработчиков в вузах.

Платформы разработки в образовании

88% технологических вузов с программами подготовки разработчиков используют платформы в учебном процессе. GitHub — безусловный лидер, его знают и используют студенты всех опрошенных вузов.

Половина преподавателей считает использование платформ эффективным для обучения. Низкие оценки объясняют ограниченным временем на освоение инструментов в учебном процессе и техническими ограничениями вузов.

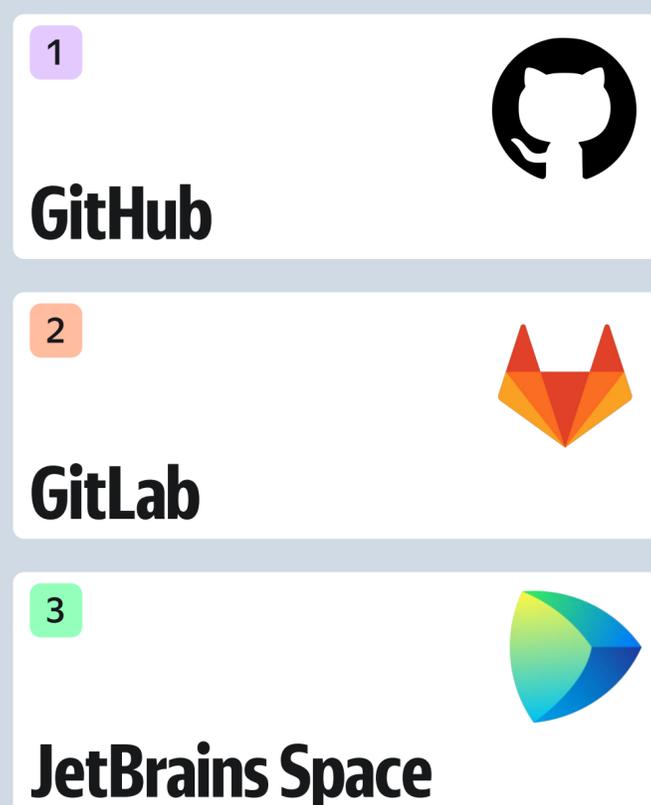
Студенты настроены оптимистичнее — 77% считают платформы эффективными для обучения и профессиональной адаптации.

Интересно, что студенты оценивают эффективность платформ разработки выше, чем преподаватели. Возможно, молодое поколение быстрее адаптируется к новым инструментам и видит в них больше возможностей.

Эффективны ли платформы разработки в учебном процессе и научной деятельности?

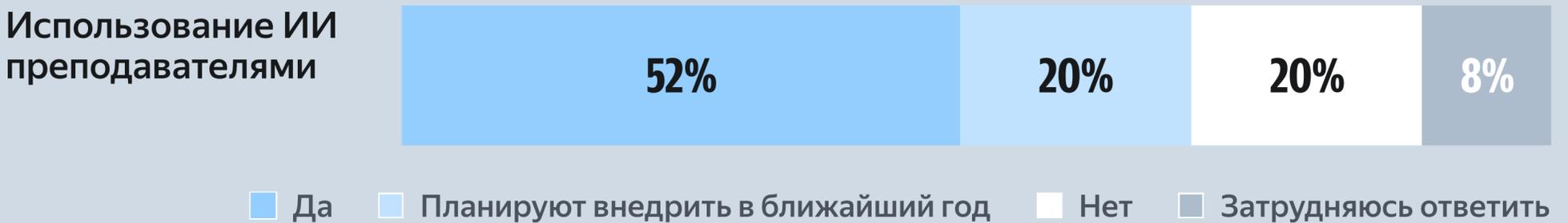


Топ платформ разработки



Кодовые ассистенты в вузах

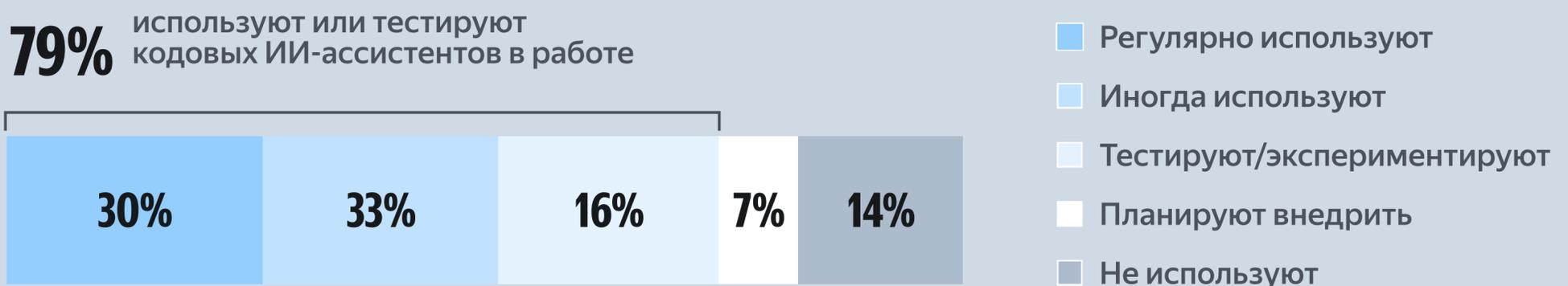
Половина преподавателей уже использует ИИ-ассистенты, ещё 20% планируют внедрить их в течение года. Это показывает, что образование не отстаёт от индустрии.



79% студентов активно используют кодовые ИИ-ассистенты — такая же доля, как среди работающих специалистов. Новое поколение разработчиков учится программировать сразу с ИИ-инструментами.

Студенты выбирают те же инструменты, что и профессионалы: универсальные языковые модели, GitHub Copilot, Gemini Code Assist и Cursor AI.

Использование кодовых ИИ-ассистентов в работе среди студентов

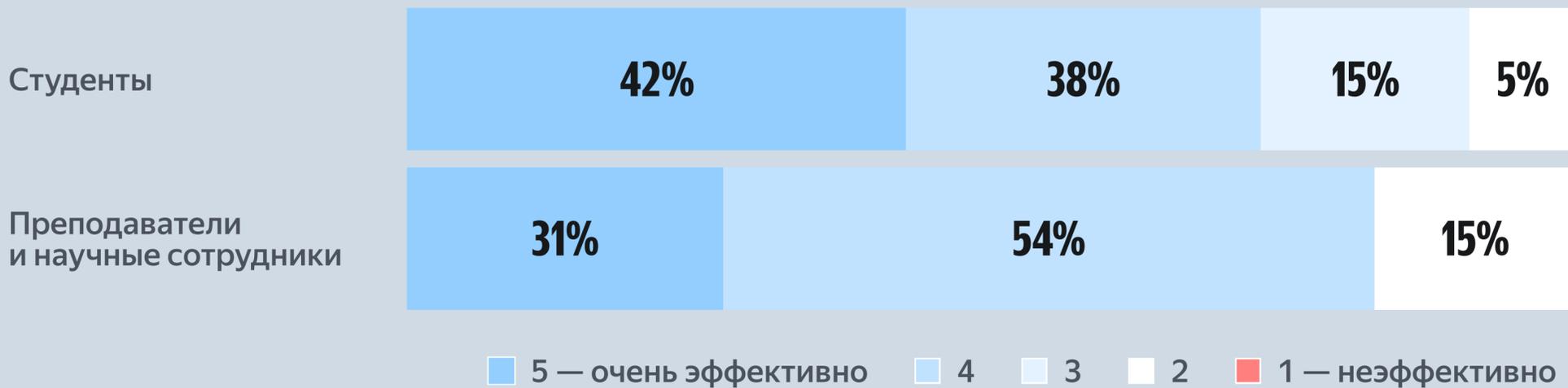


85% представителей вузов считают ИИ-ассистентов эффективными, хотя пока внедрили их только 52%. Разрыв между желаемым и действительным постепенно сокращается.

80% студентов отмечают эффективность ИИ-ассистентов. Они ценят уменьшение рутины, ускорение работы и возможность учиться на сгенерированном коде. Негативные оценки связаны с тем, что ИИ недостаточно понимает сложный код и допускает ошибки в генерации.

Взгляд в будущее: разработчики завтрашнего дня

Эффективны ли ИИ-ассистенты в учебном процессе и научной деятельности?



В учебных программах появляются сценарии, которые тренируют не только написание, но и проверку кода — с акцентом на анализ ошибок и объяснение решений.

Такой подход формирует привычку перепроверять ответы модели и разбирать причины ошибок — это задача первого уровня для безопасного и эффективного использования ассистентов.

«Неожиданный приём использования ИИ-ассистентов в образовании базируется на потенциальной способности LLM к галлюцинированию и созданию неоптимального кода, но — в учебных целях. За счёт дополненного промптинга возможно создавать версии одного и того же кода с разными недостатками (от мёртвых переменных — до неоптимальных алгоритмов), которые должен затем выявить и скорректировать обучаемый. При этом сам ассистент может выступать как ментор первого уровня, автоматизированно оценивая действия обучаемого и выдавая необходимые подсказки. Таким образом, это качественно меняет процессы изучения технологий программирования, изначально прививая навыки проверки сгенерированного кода, что является основой эффективного применения вайб-кодинга».



Александр Бухановский

д.т.н., руководитель Института искусственного интеллекта ИТМО, директор мегафакультета трансляционных информационных технологий, Университет ИТМО

Выводы

75%

разработчиков
уже используют
кодовые ассистенты

3 из 4

компаний планируют
вкладывать в модернизацию
платформ и внедрение ИИ

Исследование показывает: индустрия разработки проходит через фундаментальные изменения. ИИ-инструменты из экспериментальных технологий превратились в неотъемлемую часть рабочего процесса. 75% разработчиков уже используют кодовые ассистенты — такая же доля наблюдается и среди студентов.

Рынок труда не столько сокращается, сколько трансформируется. Меняются требования к специалистам — теперь недостаточно просто уметь писать код. Нужно понимать, как эффективно работать с ИИ, постоянно учиться новому и развивать экспертизу в смежных областях.

Большинство компаний готовы инвестировать в новые инструменты — три четверти планируют вкладывать в модернизацию платформ и внедрение ИИ. При этом российские решения набирают популярность благодаря локальной поддержке и соответствию требованиям безопасности.

Образование не отстает от индустрии — вузы активно внедряют современные инструменты, готовят специалистов, которые с первого дня смогут эффективно работать в реальных проектах.

Будущее разработки — это симбиоз человека и ИИ, где машина берёт на себя рутину, а человек фокусируется на архитектуре, бизнес-логике и творческих задачах. Эти изменения уже происходят.