Новости ИИ в 2026: как строить процесс вокруг агента, а не надеяться на промпт

05.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Почему новости об ИИ всё чаще про процесс, а не про модель

В 2026 году анонсы моделей выходят почти ежедневно, но в рабочих чатах обсуждают другое: как заставить агента не ломать сборку, не терять контекст и не выдавать «готово, все тесты проходят» там, где тесты вообще не запускались. **ИИ-агент** — это программа, которая получает цель, планирует шаги и выполняет действия в среде: читает файлы, пишет код, запускает команды, обращается к API. Чем больше свободы, тем важнее обвязка вокруг.

Показательный кейс из новостей ноября 2026 года: продакт без разработчика в команде собрал расширение для Chrome силами двух агентов — Claude Code и Codex. В проекте накопилось 593 коммита и более 78 тысяч строк кода. Результат стал возможен не из-за «умной модели», а из-за правил, проверок, запретов и навыков (скиллов), которые автор выстроил вокруг агента. Это и есть главный сдвиг года: ценность смещается от промпта к инженерной обвязке.

Ниже — практический разбор аспектов, которые редко попадают в заголовки, но определяют, будет ли ИИ полезен команде.

Обвязка агента: что это и из чего состоит

Обвязка — совокупность правил, инструментов и проверок, которые ограничивают и направляют работу агента. Без неё агент оптимизирует ближайшую цель и легко создаёт скрытый долг. Минимальный набор в 2026 году выглядит так:

- **Правила и запреты.** Файл с инструкциями в репозитории: какие команды можно запускать, какие каталоги трогать нельзя, что делать при неоднозначности.
- **Спецификация до кода.** Короткое описание поведения, входов и выходов. Агент проверяет себя по ней, а не по своему же ответу.
- **Автоматические проверки.** Линтеры, типы, тесты, сборка. Если проверка не запускается одной командой, агент её пропустит.
- **Границы среды.** Отдельная ветка, контейнер или песочница, куда у агента есть доступ, и запрет на прямые изменения в основной ветке.
- **Журнал действий.** Лог команд и изменений, чтобы разбирать инциденты не по памяти, а по фактам.

  Правило, которое экономит часы: агент не считается «закончившим», пока не приложен вывод проверок. Формулировка «всё работает» без лога — не результат.

Спецификации вместо петли багов

Один из материалов ноября 2026 года описывает типичную ловушку: руководитель ждёт, что агент получит задачу одним промптом и сам выкатит код. На практике команда попадает в петлю — каждый баг на проде порождает переписывание кусков системы и новые баги.

Рабочая альтернатива — работа по спецификациям. Порядок шагов:

1. Сформулировать поведение: что должно происходить, какие ограничения, что считается ошибкой.
2. Разбить на небольшие проверяемые изменения, каждое — с критерием приёмки.
3. Поручить агенту реализацию одного изменения и прогон проверок.
4. Проверить diff человеком или отдельным агентом-ревьюером.
5. Зафиксировать результат и только потом переходить к следующему шагу.

Такой цикл медленнее «одного промпта», но предсказуемее. Он же снижает риск, что агент незаметно изменит несвязанный модуль.

Критик рядом с агентом

В новостях 2026 года появился термин **агент-критик** — отдельная роль, задача которой испортить удовольствие от почти согласованного решения вопросами: «Откуда эти цифры?», «Это точно разрешено?», «У похожего проекта ведь не получилось?». В СИБУРе такую роль добавили в систему оценки идей замены материалов, чтобы проверять, на чём держится убедительный ответ модели.

Практика применима и в разработке. Критик не пишет код, а проверяет допущения: соответствие спецификации, источники данных, побочные эффекты. Важно, чтобы у критика был доступ к тем же артефактам, что у исполнителя, иначе спор превращается в обмен мнениями.

Память ассистента и цена контекста

Ещё один кейс ноября 2026 года — голосовой ассистент GROM на хакатоне в «Большом Физтехе». Ассистент отвечал участникам и подводил итоги обсуждения в прямом эфире. После спора о внешности персонажа он начал возвращаться к его бороде даже в ответах на другие вопросы. Причина — устройство памяти: часть контекста выпала из обработки, а часть закрепилась слишком сильно.

Вывод для продуктовых команд: память агента — это проектное решение, а не побочный эффект. Нужно явно определить, что сохраняется между сессиями, что забывается, а что переспрашивается. Иначе ассистент будет уверенно повторять устаревшую деталь.

Что проверить в своём процессе на этой неделе

- Есть ли одна команда, которая запускает все проверки? Если нет — агент будет обходить их.
- Записан ли запрет на действия в основной ветке и в продакшн-среде?
- Сформулированы ли критерии приёмки для типовых задач или каждый раз импровизация?
- Кто играет роль критика и имеет ли он доступ к спецификации и логам?
- Где хранится память ассистента и как её очищать?

Новости ИИ в 2026 году всё меньше похожи на гонку моделей и всё больше — на инженерную дисциплину. Агент ускоряет работу только там, где есть спецификация, проверки и человек, готовый задать неудобный вопрос. Остальное — хайп, который разбивается о первый баг на проде.

Источник: https://cms.dolinais.ru/novosti-ii-v-2026-kak-stroit-process-vokrug-agenta-a-ne-nadeyatsya-na-prompt