🤖 Claude Code: от одного агента к тысячам

🤖 Эволюция работы с Claude Code: от одного агента к тысячам

Ключевые тезисы:

- 🔄 **Автоматизация вместо инструкций**: Когда Claude ошибается, его учат не через промпты, а через создание навыков или документации.
- ✅ **Верификация — ключевой вызов**: Это не просто юнит-тесты, а способность агента *запустить* и проверить работоспособность системы.
- 🤝 **Слияние ролей**: С Claude Code продукт-менеджеры, дизайнеры и финансисты пишут код, а инженеры — управляют продуктом.
- 🚀 **Автономные рутины**: Агенты могут проактивно мониторить баги, создавать и мержить фиксы без участия человека.
- 🛡️ **Безопасность через авторежим**: Эмпирически доказано, что авторежим с классификатором безопаснее, чем ручное подтверждение каждого действия.
- 📱 **Новая парадигма работы**: Кодинг с телефона, удалённое управление и работа с десятками агентов одновременно становятся нормой.
- 🧠 **Минимализм контекста**: Лучше дать модели минимум информации и свободу действий, чем её "микроменеджить".

🎯 От Claude Code к армиям агентов

- Год назад Claude Code решал простые инженерные задачи.
- Сегодня автор управляет **деревьями из тысяч агентов**, где один агент может управлять другими.

- **Главный принцип**: Если Claude ошибается, его не просят сделать иначе. Ему поручают **записать правильный способ в файл (CLAUDE.md) или создать навык**. Это позволяет системе работать автономно.

🔍 Верификация: настоящий вызов для агентов

- Верификация для агентов — это **не просто автоматические тесты** (юнит-тесты, линтеры).
- Это проверка, **может ли агент запустить и выполнить задачу**, что часто требует нестандартных решений.
- **Пример**: Claude Opus 4 однажды протестировал собственный функционал, запустив CLI и проверив себя. Сейчас это обычная практика с симуляторами iOS/Android.

🛠️ Практика: навыки и автономное исправление

- Команда создаёт **навыки** (skills), которые учат Claude работать с конкретными системами (например, десктопным приложением).
- Когда агент сталкивается с проблемой (например, баг в staging-среде), он:
  1. Анализирует контекст (читает Slack, проверяет статус).
  2. Дебажит проблему.
  3. **Автоматически обновляет соответствующий навык**, чтобы в будущем избежать этой ошибки.
- Навык для десктоп-приложения позволяет Claude с помощью **computer use** кликать по интерфейсу, тестировать новый UX и крайние случаи.

🧩 Слияние ролей в эпоху ИИ

- **Все в команде пишут код**: PM, дизайнеры, финансисты, data scientists.
- Claude Code становится **универсальным инструментом** для разных ролей:
  - Дизайнеры правят код кнопок напрямую.
  - Финансовый отдел строит прогнозы.
  - Продуктологи вносят изменения в приложение.

- **Что теперь важно**: Не столько навык написания кода, сколько **идеи, продуктовый и бизнес-контекст, понимание пользователя**.

⚙️ Автономные рутины (Routines)

- **Рутины** — прорывное применение Agent SDK.
- **Пример**: Инженер настроил рутину, которая:
  1. Слушает все тикеты и баг-репорты, связанные с голосовым режимом.
  2. **Проактивно создаёт фикс**.
  3. Отправляет PR на мерж (простые проверяются и мержатся автоматически).
- Другая рутина отслеживает баг-репорты, на которые **нет ответа более 5 часов**, и автоматически создаёт фиксы.
- Результат: **"Кто-то уже пофиксил"** — обычная ситуация, так как чужие Claude тоже работают над проблемами.

🎮 Изменение workflow: Plan Mode → Auto Mode → Loop

- **Plan Mode** устарел для современных моделей (начиная с Claude 4.6/4.7).
- **Auto Mode** — новый стандарт:
  - Агент начинает работать сам, пользователь может переключиться на другую задачу.
  - **Безопаснее ручного подтверждения**: Классификатор модели отклоняет подозрительные запросы. Человек видит только важные исключения, а не 99% очевидных разрешений.
- **Loop** — следующий скачок: пользователь общается не с агентом, а с Loop/рутиной, которая сама промптит Claude.

🔒 Безопасность Auto Mode: эмпирический подход

- Чтобы доверять Auto Mode, команда провела масштабную работу:
  1. Собрали **тысячи транскриптов** полных траекторий агентов с permission-промптами.
  2. Научили Auto Mode-классификатор определять безопасность.
  3. **Редактировали (red teaming)**: Привлекли специалистов для взлома и prompt-инъекций, чтобы улучшить защиту.
  4. Внутренние команды также пытались "взломать" систему для создания эвалов.
- **Вывод**: Безопасность в ИИ часто выглядит не так, как ожидаешь. Нужно постоянно тестировать и пересматривать подходы.

💡 Claude в центре всех бизнес-процессов

- **Историческая параллель**: Компьютеры дали рост производительности, только когда стали **центром всех процессов**, а не просто дополнением к бумажным архивам.
- **В Anthropic Claude — центр всего**:
  - Новые сотрудники задают вопросы Claude, а не людям.
  - Написание кода, код-ревью, security-ревью, заполнение форм — всё делает Claude.
  - Компании, которые преуспевают, ставят ИИ в центр, а не используют его на периферии.
- **Для инженера**: Исчезает рутина и список задач (todo-list). Остаётся **придумывание идей** и общение с клиентами. Claude всё строит.

📱 Новая парадигма работы: сотни агентов

- **Старый способ**: 6 вкладок терминала с разными git checkout.

- **Новый способ**:
  - **Agent View** в одном окне.
  - **Десктопное приложение**, которое само управляет work trees.
  - **Половина работы с телефона**: Удалённое управление (remote control) позволяет запускать и проверять агентов из любого места.
- **Пример**: Обсуждаешь идею с коллегой → запускаешь агента с телефона через голосовой режим → он начинает строить, не требуя возврата к компьютеру.

🧠 Контекст-инжиниринг: философия минимализма

- **Раньше** (Sonnet 3.5, Opus 4) нужны были промпт- и контекст-инжиниринг.
- **Сейчас**: Дайте модели **минимум необходимого контекста** и **свободу действий**.
- **Философия**: Не микроменеджьте модель. Дайте ей возможность найти лучший путь к цели.
- Платформа становится "тоньше", оставляя больше места для пользовательских промптов.

Выводы:

1. **Будущее — за автономией**: Агенты учатся на ошибках, создают навыки и работают в рутинах, минимизируя человеческое вмешательство.
2. **Роли сливаются**: Продуктовое мышление и умение генерировать идеи становятся важнее узкой специализации. Все становятся и инженерами, и продуктологами.
3. **Доверие через безопасность**: Авторежим и тщательная работа над безопасностью позволяют доверять агентам и масштабироваться до сотен параллельных процессов.
4. **ИИ в центре**: Максимальная производительность достигается, когда ИИ (Claude) становится центральным элементом каждого бизнес-процесса, а не инструментом "на подхвате".
5. **Форматы меняются**: Работа с десятками агентов, кодинг с телефона и удалённое управление — это новая реальность, которая будет развиваться дальше.