

План внедрения AI на проекте

Цель: Повысить продуктивность команды разработки программного обеспечения за счет использования AI как инструмента поддержки, улучшения качества кода и ускорения выполнения задач.

Основная идея: AI должен быть интегрирован в рабочие процессы как продвинутый инструмент, который помогает разработчикам, но не заменяет их. Для успешного внедрения необходимо сосредоточиться на улучшении текущих процессов, использовании лучших инструментов на рынке и создании системного подхода к обучению и рефакторингу.

План внедрения AI на проекте

Этап 0: Подготовка команды и инфраструктуры

1. Оценка текущего состояния:

- **Аудит навыков разработчиков:** Провести оценку уровня знаний команды по ключевым фреймворкам, лучшим практикам разработки и навыкам работы с AI. Выявить сильные стороны и области, требующие улучшения.
- **Анализ кодовой базы:** Оценить состояние кодовой базы на соответствие современным стандартам, выявить области с техническим долгом и определить задачи для рефакторинга.
- **Выявление гэпов:** Составить карту навыков команды, чтобы определить, какие знания и навыки необходимо развивать для эффективного использования AI.

2. Обучение промт-инжинирингу:

- Выявить потребность и организовать дополнительные тренинги по основам работы с AI, включая создание эффективных промтов.
- Провести индивидуальные практические занятия, где разработчики учатся ставить задачи AI, проверять результаты и корректировать запросы.

3. Обновление кодовой базы:

- Привести кодовую базу к действующим стандартам, чтобы AI мог работать с ней более эффективно.
 - Устранить технический долг и внедрить единые правила кодирования.
-

Этап 1: Внедрение AI в рабочие процессы

1. Использование AI для рефакторинга и устранения технического долга:

- Использовать AI для автоматизации рефакторинга, включая оптимизацию алгоритмов, улучшение читаемости кода и приведение его к современным стандартам.
- Разделить задачи рефакторинга на этапы, чтобы минимизировать риски и обеспечить постепенное улучшение.

2. Интеграция AI в инструменты разработки:

- Убедиться, что все разработчики используют GitHub Copilot.
- Настроить AI для работы с текущими фреймворками и библиотеками, чтобы повысить его эффективность.

3. Фокус на улучшении текущих процессов:

- Вместо создания новых AI-решений, сосредоточиться на применении лучших инструментов на рынке, которые уже доказали свою эффективность.
 - Постепенно внедрять AI во все этапы жизненного цикла программного обеспечения, включая обновление процесса разработки целиком.
-

Этап 2: Развитие культуры обмена знаниями

1. Создание базы знаний:

- Организовать внутреннюю платформу для обмена промтами, успешными кейсами и сложными ситуациями.
- Включить раздел с примерами, где AI не смог дать релевантный ответ, и предложить альтернативные подходы.

2. Регулярные встречи и обсуждения:

- Проводить регулярные митапы и воркшопы для обмена опытом использования AI.
- Делать обзоры новых возможностей AI и лучших практик.

3. Менторство и поддержка:

- Назначить ответственных за обучение и поддержку коллег в работе с AI.
 - Внедрить систему наставничества для новичков.
-

Этап 3: Постоянное улучшение

1. Мониторинг эффективности:

- Регулярно собирать обратную связь от команды о работе AI.
- Оценивать влияние AI на производительность, качество кода и скорость выполнения задач.

2. Обновление навыков:

- Проводить периодические тренинги по новым возможностям AI и актуальным практикам разработки.
- Поддерживать актуальность базы знаний.

3. Эксперименты и исследования:

- Постоянно тестировать новые подходы к использованию AI.
 - Внедрять улучшения на основе результатов экспериментов.
-

1. Улучшение текущих процессов:

- Приведение кодовой базы в порядок, устранение технического долга и рефакторинг с использованием AI.
- Автоматизация рутинных задач и повышение качества разработки.

2. Использование лучших инструментов:

- Сосредоточиться на применении проверенных AI-решений, таких как GitHub Copilot, вместо разработки собственных инструментов с высоким риском невостребованности.

3. Обучение и развитие команды:

- Повысить уровень знаний команды по ключевым фреймворкам, лучшим практикам разработки и навыкам работы с AI.
- Развивать навыки промт-инжиниринга у разработчиков.
- Создавать культуру обмена знаниями и постоянного обучения.

Прогнозируемые результаты

1. Краткосрочные:

- Повышение уровня знаний разработчиков по ключевым фреймворкам, лучшим практикам разработки и навыкам работы с AI.
- Улучшение качества кода за счет рефакторинга и устранения технического долга.
- Ускорение выполнения задач за счет эффективного использования AI.

2. Среднесрочные:

- Снижение количества ошибок в коде и улучшение тестового покрытия.
- Ускорение выполнения задач благодаря эффективному использованию AI.

3. Долгосрочные:

- Формирование культуры обмена знаниями и постоянного обучения.
- Повышение общей продуктивности команды и конкурентоспособности проекта.
- Повышение качества разработки за счет использования AI.

Заключение

AI уже интегрирован в рабочие процессы команды через инструменты, такие как GitHub Copilot, и прошли общие тренинги. Следующим шагом является системное улучшение текущих процессов, включая рефакторинг, устранение технического долга и развитие навыков разработчиков. Основной акцент делается на использовании лучших инструментов на рынке, чтобы минимизировать риски и максимизировать эффективность. Это заложит фундамент для устойчивого роста и повышения качества разработки в долгосрочной перспективе.

Внедрение AI открывает возможность не только повысить качество разработки, но и сделать команду более клиенто-центрированной — мы сможем проще и быстрее проходить «дополнительную милю» для клиента на любом этапе работы. Также важно выявить сотрудников, которые скептически относятся

к внедрению AI, и организовать индивидуальную и групповую работу с ними для снижения сопротивления и успешной интеграции новых инструментов.