

## Предложение и план внедрения AI в разработку ПО

### Вводные и гипотезы

- **AI — это инструмент, а не самостоятельный интеллект.**

Он работает как продвинутый поисковый и генеративный механизм, выдающий наиболее релевантные и популярные решения на основе анализа большого объема данных. Для максимальной эффективности AI требует четкой постановки задач, промежуточного контроля и валидации результатов (вручную или через тесты).

- **Знания и навыки — основа продуктивности с AI.**

Для эффективной работы с AI необходимы:

- Глубокое знание используемых фреймворков и лучших практик разработки.
- Навыки prompt-инжиниринга (умение формулировать запросы к AI).
- Понимание ограничений и принципов работы AI (AI — это языковая модель, а не эксперт).

- **Золотая середина.**

Необходимо выработать баланс между самостоятельной работой и использованием AI: AI — это помощник, которому нужно ставить цели, формулировать задачи и проверять каждый шаг.

- **Чем сложнее задача, тем выше риск некорректного результата.**

Для крупных и комплексных задач AI должен использоваться как вспомогательный инструмент, а не как основной исполнитель.

- **Фундаментальная подготовка — ключ к успеху.**

Перед внедрением AI важно:

- Привести кодовую базу к современным стандартам.
- Выявить и закрыть пробелы в знаниях команды по фреймворкам и best practices.
- Научить всех разработчиков основам prompt-инжиниринга.

- **Обмен знаниями — фундамент долгосрочного успеха.**

Необходимо создать культуру обмена промтами, кейсами, сложными ситуациями и примерами неудачного применения AI. Это не даст мгновенного эффекта, но заложит прочную основу для будущего роста.

---

### Что уже реализовано

- **Лицензии на GitHub Copilot** выданы всем разработчикам, инструмент интегрирован в IDE.
  - **Общие тренинги по работе с AI** уже проведены для всей команды.
  - **Планируется индивидуальная работа** с каждым разработчиком, включая практические занятия по prompt-инжинирингу и эффективному использованию AI.
- 

### План внедрения AI

#### Этап 0: Фундаментальная подготовка

- **Аудит навыков и кодовой базы:**  
Оценить уровень знаний по фреймворкам, best practices, выявить пробелы и технический долг.
- **Обучение prompt-инжинирингу:**  
Провести тренинги и практические занятия для всех разработчиков.
- **Приведение кода к стандартам и рефакторинг:**  
Провести рефакторинг, не только устраняя технический долг, но и улучшая архитектуру, читаемость и поддерживаемость кода. С помощью AI эти задачи теперь решаются значительно проще и быстрее.

### Этап 1: Внедрение AI в процессы

- **Пилотная волна:**  
В первую очередь обучить и вовлечь ключевых и ведущих разработчиков, чтобы они стали инфлюенсерами и задали тренд для остальных.
- **Интеграция AI-инструментов:**  
Внедрить лучшие решения (например, GitHub Copilot) и настроить их под используемые технологии.
- **Постепенное расширение:**  
После успешного пилота — масштабировать на всю команду.

### Этап 2: Развитие культуры обмена знаниями

- **Создание внутренней базы знаний:**  
Платформа для обмена промтами, кейсами, сложными ситуациями и примерами неудач.
- **Регулярные обсуждения:**  
Митапы, воркшопы, обзоры новых возможностей AI.
- **Менторство:**  
Назначить ответственных за поддержку и обучение коллег.

### Этап 3: Постоянное совершенствование

- **Мониторинг и обратная связь:**  
Регулярно собирать фидбек, оценивать влияние AI на производительность и качество.
- **Обновление навыков:**  
Проводить периодические тренинги и поддерживать актуальность базы знаний.
- **Эксперименты:**  
Постоянно тестировать новые подходы и делиться результатами.

---

### Основные акценты

- **AI — это инструмент, эффективность которого зависит от уровня разработчика и культуры команды.**
- **Фундаментальная подготовка и обмен знаниями — залог долгосрочного успеха.**
- **Внедрение AI — это не разовая акция, а системный процесс, требующий вовлечения лидеров, обучения и постоянного совершенствования.**
- **Благодаря интеграции AI и лучших инструментов на рынке, таких как GitHub Copilot, задачи, которые ранее были сложными и трудоемкими (рефакторинг, устранение технического долга, обучение и проверка знаний), теперь решаются значительно проще и**

**быстрее. Это позволяет быстро привести кодовую базу и процессы в порядок, заложив фундамент для дальнейшего обновления и совершенствования всего процесса разработки.**

- **Основной акцент делается на улучшении существующего и использовании проверенных решений, а не на создании собственных AI-инструментов с высоким риском невостребованности.**
- **AI востребован на всех этапах работы: от выявления гэпов в знаниях и их устранения, до высвобождения капасити у ключевых разработчиков и рефакторинга кодовой базы.**
- **Вместо создания очередной собственной "крутой тулы" важно сосредоточиться на подготовке к внедрению AI в масштабе всей команды.**
- **Да, эффект может быть замечен не сразу, но это фундамент для "многоэтажного дома", который появится на следующих этапах развития команды и процессов.**

---

Если нужно, могу оформить это в виде markdown-файла или интегрировать в ваш текущий документ.