+7 (800) 101-45-71
edu-tritec.ru



Курс: "Инженерный подход в разработке ПО с использованием ИИ"

© АНО «Учебный центр «Трайтек», 2026. Все права защищены.

Направление обучения	Искусственный интеллект и нейросети
Длительность	16 акакдем. ч.
Документы об окончании	Сертификат Трайтек, Удостоверение о повышении квалификации
Цена	32 000 руб. (актуально на 21.11.2026)
Планируемая дата	15.02.2027. Идет набор в группу

Целевая аудитория

- Разработчики (middle/senior)
- Тимлиды
- QA-инженеры
- Архитекторы, уже эпизодически использующие ИИ-ассистенты

Требования к слушателям

- Опыт коммерческой разработки от 1 года, базовое использование Git

© АНО «Учебный центр «Трайтек», 2026

О курсе

Цель курса:

Заменить неструктурированный «вайб-кодинг» (генерация кода по наитию без анализа и проверки) на осознанный инженерный процесс.

Цели обучения

- отличать вайб-кодинг от инженерной AI-разработки и аргументировать риски (безопасность, техдолг, «чёрный ящик»);
- составлять спецификации требований (PRD, EARS-формулировки, acceptance criteria) и прогонять их через SDD-конвейер;
- выбирать SDD-фреймворк под тип проекта (greenfield/brownfield) и стандартизировать его применение в команде: OpenSpec, GitHub Spec Kit, Kiro, BMAD-METHOD, Task Master, Agent OS и аналоги.
- проектировать архитектуру с участием ИИ и фиксировать решения в ADR;
- собирать «контекстный пакет» проекта: AGENTS.md/CLAUDE.md/rules-файлы, глоссарий, примеры эталонного кода;
- декомпозировать план на задачи с критериями готовности, пригодные для делегирования агенту;
- настраивать ограничения агента: права доступа, hooks, линт-гейты, скоуп файлов;
- вести TDD-цикл «красный-зелёный-рефакторинг» с генерацией тестов из acceptance criteria;
- работать минимум с двумя типами ИИ-агентов и оркестровать их (plan mode, субагенты, параллельная работа в git worktrees);
- проводить код-ревью ИИ-сгенерированного кода по чек-листу и находить типовые галлюцинации

Программа курса

Тема 1. AI-first мышление и AI-SDLC

- • Что такое вайб-кодинг: происхождение термина, типовой сценарий («прими всё, разберёмся потом») и его последствия: скрытые дефекты, уязвимости, деградация архитектуры, утрата понимания кодовой базы.
- • Антипаттерны: бесконечный цикл «поправь вот это», промпты без контекста, отсутствие критериев готовности, слепой merge.

© АНО «Учебный центр «Трайтек», 2026

- • Обзор AI-SDLC. Роль человека на каждом этапе.
- • Таксономия ИИ-инструментов: автодополнение, чат, агент, оркестратор агентов. Где проходит граница ответственности.
- • Модель зрелости AI-разработки команды (5 уровней) – самодиагностика.
- Практическая работа: Поиск дефектов в сгенерированной фиче, сравнение с дисциплинированным проходом.

Тема 2. Спецификация: Spec Driven Development

- • Уровни формализации намерения.
- • SDD-конвейер. Обзор классов SDD-инструментов (GitHub Spec Kit, OpenSpec, Agent OS и аналоги).
- • Требования: PRD, user stories, EARS-нотация («When , the system shall…»), acceptance criteria в формате Given/When/Then.
- • Техника уточнений: как задавать агенту вопросы до кода (этап clarify), работа с неоднозначностями.
- • Спецификация как контракт между человеком и агентом; спецификация как материал для тестов.
- Практическая работа: Полный конвейер в выбранном фреймворке.

Тема 3. Архитектура, план и контекст-инжиниринг

- • Роль ИИ в проектировании: генерация вариантов архитектуры, trade-off анализ, ADR (Architecture Decision Records) при участии агента.
- • Декомпозиция: план, граф задач, порядок с зависимостями.
- • Контекст-инжиниринг как новая дисциплина: AGENTS.md (открытый стандарт), CLAUDE.md, .cursor/rules, specs/, changes/, .agent-os/, copilot-instructions, глоссарий, примеры «золотого кода», структура каталогов как подсказка.
- • Управление бюджетом контекста.
- Практическая работа: Сборка контекстного пакета.

Тема 4. Ограничения и дисциплина ИИ-агентов

- • Модель угроз при работе с агентом: выход за скоуп, порча соседнего кода, коммит секретов, установка небезопасных зависимостей, галлюцинации.
- • Уровни ограничений:
  - **Правила** (rules/instructions) – что агент должен и не должен делать;
  - **Права и песочница** – allow/deny списки команд и путей, режим подтверждения;

© АНО «Учебный центр «Трайтек», 2026

    - **Hooks/гейты** – автоматические проверки до/после действий (запрет разрушительных команд без подтверждения, блокировка коммита без линтера/тестов);
    - **Процессные** – запрет трогать файлы вне скоупа задачи.
- • Изоляция.
- Практическая работа: Настройка deny-списков, прав, скоупа файлов.

Тема 5. TDD с ИИ и верификация

- • TDD – естественный способ управления агентом: тесты как исполняемая спецификация и автоматический критик.
- • Генерация тестов из acceptance criteria.
- • Цикл red-green-refactor в AI-варианте.
- • Тесты для легаси перед ИИ-рефакторингом.
- • Антипаттерны: тесты, написанные после кода; тесты-галлюцинации; мутационная устойчивость.
- Практическая работа: Генерация тестов.

Тема 6. Оркестрация агентов, качество

- • Ландшафт агентов (обзорно, с учётом разрешённых в компании); критерии выбора (контекст, инструменты, политика данных, стоимость).
- • Протоколы расширения: MCP (подключение корпоративных источников контекста), субагенты, skills/кастомные команды.
- • Надстройки управления работами: слой планирования над агентами.
- • Параллельная разработка: несколько агентов в worktrees; паттерны «архитектор + рабочие пчёлы», «реализатор + ревьюер».
- • Качество: чек-лист ревью ИИ-кода (логика, зависимости, секреты, лицензионные риски, галлюцинированные API); ИИ-ревью как первый контур.
- • Итоговая модель процесса команды: политика AI-разработки в одном документе.
- Практическая работа: Параллельные задачи в worktrees.

Итоговая практическая работа: Реализация фич по полному циклу AI-SDLC.

© АНО «Учебный центр «Трайтек», 2026