

Управляемая разработка с ИИ-агентами

Как делегировать разработку
агентам и не потерять контроль

Code is cheap. Show me the spec.

Алекс Гусев

Управляемая разработка с AI-агентами

Как делегировать разработку агентам и не
потерять контроль

Alex Gusev

Эта книга предназначена для продажи на <https://leanpub.com/adsm-ru>

Эта версия была опубликована 2026-04-21



Это книга с [Leanpub](#) book. Leanpub позволяет авторам и издателям участвовать в так называемом [Lean Publishing](#) - процессе, при котором электронная книга становится доступна читателям ещё до её завершения. Это помогает собрать отзывы и пожелания для скорейшего улучшения книги. Мы призываем авторов публиковать свои работы как можно раньше и чаще, постепенно улучшая качество и объём материала. Тем более, что с нашими удобными инструментами этот процесс превращается в удовольствие.

© 2026 Alex Gusev

Оглавление

Введение	1
Потеря управляемости	5
Человеческая модель агента	5
Почему агент кажется человеком	5
Чем агент не похож на человека	5
Источник потери	5
СИМПТОМЫ	5
Связка «человек + агент»	5
Что меняется в работе	6
Аксиомы ADSM	7
Исходная проблема	7
Роль аксиом	7
Три аксиомы	7
Практический смысл	7
Эффект принятия	7
Цена игнорирования	7
Основание ADSM	8
Документация как опора разработки	9
Почему документация становится опорой	9
Что именно документация удерживает	9
Документация, код и тесты: разные роли	9
Собственная память проекта	9
Области документов и их назначение	9
Документация как шаблон воспроизведения кода	9
Что агент должен возвращать в документы	10
Что выигрывает разработчик	10
Чем проект платит за этот выигрыш	10
Место AGENTS.md в корпусе материалов	10

Рабочий контекст	11
Промпт	11
Два контекста	11
Старт	11
Соответствие	11
Слои	11
AGENTS.md	11
Иерархия	12
Сборка	12
Управление	12
Агент-исполнитель	13
Инструмент	13
Полезная зона	13
Распределение ролей	13
Фокус документации	13
Конфигурация агента	13
Структура контекста	13
Настройка делегирования	14
Цена антропоморфизма	14
Спираль разработки	15
Обратная связь	15
Виток спирали	15
Цикл обратной связи	15
Уточнение документации	15
Формирование ожиданий	15
Локальная реализация	15
Обратная связь и самостоятельность	16
Цена витков	16
Где ADMS даёт наибольший эффект	17
Цена будущих изменений	17
Где подход особенно полезен	17
Какие свойства проекта усиливают эффект	17
Что подход требует от проекта	17
Риск и безопасное делегирование	17
Где эффект снижается	17
Где подход становится избыточным	18

Как меняется роль разработчика	18
Практический итог	18
Практический пример	19
Проект	19
Базовые генерации	20
Инвариант результата	20
Устойчивая ошибка	21
Изменение контекста	22
Повторная генерация	23
Наблюдаемое изменение	24
Связь с ADSM	25
Итог	25

Введение

Большие языковые модели (LLM) уже вошли в разработку и быстро меняют её практику. Они помогают писать код, ускоряют реализацию и позволяют делегировать всё большую часть инженерной работы агентам. Вместе с этим в разработке проявляется новый вызов: при прямом использовании ИИ-агентов результат нередко оказывается нестабильным, слабо воспроизводимым и трудным для удержания в управляемом состоянии. Агент может заметно ускорять работу, однако сама по себе скорость исполнения ещё не задаёт способа разработки, устойчивого на длинной дистанции.

На мой взгляд, эта особенность связана с тем, что современная разработка по-прежнему в значительной степени опирается на `code-first` модель. Код воспринимается как главный источник истины и привычный центр управления развитием системы. Однако человек-специалист и языковая модель читают код по-разному. Для разработчика код кристаллизует профессиональный опыт, архитектурное понимание и знание ограничений системы. Для модели он тоже остаётся значимым материалом, но не активирует тот же инженерный фон автоматически. Напротив, модель соотносит код с гораздо более широким полем возможных интерпретаций, чем человек, чьё чтение обычно опирается на более конкретный и практически сформированный опыт. Поэтому при делегировании реализации агенту необходимо явно задавать тот проектный и смысловой контекст, который человек часто восстанавливает из кода сам, опираясь на собственную профессиональную специализацию. Отсюда и возрастает значение документации, формулировок и структуры контекста как средств управления поведением агента и воспроизводимостью результата.

Из этого вытекает и центральный практический вопрос: как организовать разработку с ИИ-агентами так, чтобы результат оставался предсказуемым, контролируемым и пригодным для дальнейшего развития? По мере распространения агентных подходов всё более важным становится умение не только получать от модели

отдельные решения, но и выстраивать устойчивый процесс работы с ней. В этой книге рассматривается один из возможных способов такой организации: через явное задание смысла, ограничений и условий реализации в текстовом контексте, с которым работает агент.

Этот подход я назвал ADSM (Agent Driven Software Management). Его основой стал мой практический опыт работы с Codex-агентом при разработке пакетов платформы Tequila Framework. В рамках этой работы код систематически создаётся, уточняется и развивается с участием агента, а сами принципы взаимодействия постепенно приобретают более явную и связную форму. Поэтому ADSM опирается на уже накопленный практический опыт, а результаты этой совместной работы открыты и публично доступны в репозиториях на GitHub.

В основе ADSM лежит простой, но важный сдвиг. Основным объектом управления при разработке продукта становится документация. В ней фиксируются знания о приложении, его назначение, поведение, ограничения, архитектурные решения и технологические условия. Документация при таком подходе служит системой управления: она задаёт границы, структуру и правила, в которых агент получает задачу и строит реализацию. Код при этом выступает как *производный и изменяемый* результат исполнения.

Однако для управляемой разработки недостаточно иметь документацию как таковую. Важно ещё и то, какая часть этого описания оказывается доступна агенту в момент выполнения конкретной задачи. Агент работает не со всем знанием о системе сразу, а с тем набором документов и их фрагментов, который он самостоятельно собирает для текущего шага. Поэтому качество разработки определяется не только полнотой и связностью документации, но и тем, как именно она используется в работе агента. Отбор, сборка и подача нужного материала становятся частью общего механизма управления.

Такой подход соответствует природе языковой модели. Модель не обладает инженерным пониманием в человеческом смысле и не действует из намерения. Она преобразует текстовый контекст в текстовый результат, который затем становится кодом, тестами, конфигурацией или другой частью приложения. Поэтому качество продукта на выходе в такой схеме определяется качеством описания,

с которым модель работает на входе. Чем точнее, связнее и полнее документация и чем точнее задан контекст её применения, тем надёжнее и воспроизводимее результат.

Разработка в ADSM представляет собой процесс последовательного уточнения документации и управляемого исполнения агентом. Она складывается из взаимосвязанных текстовых компонентов, описывающих приложение на разных уровнях — от назначения продукта и поведения его частей до языка программирования, платформы и принятых архитектурных решений. Совокупность этих уровней определяет итоговую реализацию. Изменение одного из них меняет результирующий код, иногда полностью, сохраняя при этом те свойства приложения, которые уже закреплены в остальном описании.

Такой подход даёт разработчику иной способ контроля над приложением. Его целостность удерживается через изменение описания и через уточнение условий, в которых агент выполняет работу. Код может быть пересобран, уточнён или переработан, сохраняя опору на уже зафиксированное знание о приложении.

Важным элементом этой схемы остаётся тестирование. Тесты дают агенту и разработчику обратную связь о том, соответствует ли исполнение требованиям, зафиксированным в документации. В ADSM тестирование входит в общий механизм контроля. Документация задаёт модель приложения, запрос задаёт направление очередного шага, агент обращается к доступному описанию и исходному коду, формирует на этой основе рабочий контекст и строит реализацию, а тесты подтверждают или опровергают её корректность. Благодаря этому итерационный процесс становится устойчивее, а качество — удерживаемым по мере развития приложения.

Хотя современные модели умеют работать с разными типами данных — аудио, изображениями, видео, — в разработке управляющим представлением для них всё равно остаётся текст. Любой медиаконтент начинает участвовать в создании приложения после того, как его смысл оказывается переведён в текстовый контекст, пригодный для работы модели. Поэтому именно текст становится рабочей средой управляемой разработки, а документация — главным носителем знания, доступного агенту.

ADSM описывает модель разработки, в которой человек работает на уровне смысла, структуры, ограничений и решений, а агент берёт на себя генерацию кода. Для разработчика это означает возможность быстрее создавать и развивать приложения, увереннее вносить изменения, делегировать часть реализации и сохранять контроль над поведением агента и над результатом его работы.

Практическая ценность ADSM состоит в том, что он придаёт агентной разработке форму, пригодную для длительного и последовательного развития приложения. Документация, отбор контекста, исполнение агентом и тестирование соединяются в связный механизм, который удерживает реализацию в пределах зафиксированных требований и позволяет расширять делегирование по мере роста приложения. ADSM предлагает одну из возможных форм такой организации разработки, уже подтверждённую практикой.

Дальнейшее изложение посвящено тому, как эта схема проявляется в инженерной практике и какие выводы из неё следуют для организации разработки. ADSM интересует меня прежде всего как способ сделать делегирование устойчивым по мере накопления изменений, усложнения продукта и продолжения работы над ним. Основанием для книги стал мой собственный опыт, однако её задача шире описания частной практики. Область разработки с ИИ-агентами ещё только складывается, и поэтому особенно важно делать явными те принципы и рабочие формы, которые уже проявляются в практике. Для читателя эта книга может стать ещё одной точкой зрения на быстро меняющуюся область, позволяющей по-новому увидеть знакомые проблемы, яснее определить опоры в работе с агентами и точнее выстроить собственный способ разработки.

Потеря управляемости

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Человеческая модель агента

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Почему агент кажется человеком

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Чем агент не похож на человека

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Источник потери

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Симптомы

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Связка «человек + агент»

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Что меняется в работе

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Аксиомы ADSM

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Исходная проблема

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Роль аксиом

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Три аксиомы

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Практический смысл

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Эффект принятия

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Цена игнорирования

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Основание ADSM

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Документация как опора разработки

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Почему документация становится опорой

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Что именно документация удерживает

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Документация, код и тесты: разные роли

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Собственная память проекта

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Области документов и их назначение

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Документация как шаблон воспроизведения кода

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Что агент должен возвращать в документы

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Что выигрывает разработчик

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Чем проект платит за этот выигрыш

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Место AGENTS.md в корпусе материалов

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Рабочий контекст

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Промпт

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Два контекста

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Старт

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Соответствие

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Слои

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

AGENTS.md

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Иерархия

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Сборка

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Управление

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Агент-исполнитель

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Инструмент

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Полезная зона

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Распределение ролей

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Фокус документации

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Конфигурация агента

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Структура контекста

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Настройка делегирования

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Цена антропоморфизма

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Спираль разработки

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Обратная связь

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Виток спирали

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Цикл обратной связи

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Уточнение документации

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Формирование ожиданий

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Локальная реализация

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Обратная связь и самостоятельность

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Цена витков

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Где ADSM даёт наибольший эффект

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Цена будущих изменений

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Где подход особенно полезен

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Какие свойства проекта усиливают эффект

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Что подход требует от проекта

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Риск и безопасное делегирование

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Где эффект снижается

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Где подход становится избыточным

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Как меняется роль разработчика

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Практический итог

Этот контент недоступен в ознакомительной версии книги. Книгу можно приобрести на Leanpub по адресу <https://leanpub.com/adsm-ru>.

Практический пример

Проект

К этому месту книга уже ввела основную рамку ADSM: проблему потери управляемости, роль документации, устройство рабочего контекста, характер делегирования и спираль разработки. Теперь полезно перейти от общей модели к одному небольшому проекту, на котором та же логика видна в наблюдаемой последовательности генераций.

Для этого использовался проект [@flancer32/spotify-playlist](#). Это небольшой CLI-инструмент на Node.js, который читает текстовый список треков, находит их через Spotify Web API и создаёт плейлист в аккаунте пользователя. Проект изначально готовился для публикации на Nabr, и его основная задача состояла не только в создании полезного инструмента, но и в демонстрации управляемости генерации кода через контекст проекта. В данном случае важен не сам домен, а свойства примера. Проект достаточно мал, чтобы сохранить обзорность, и одновременно достаточно реален, чтобы на нём проявились ограничения агентной генерации и влияние когнитивного контекста на результат.

С практической точки зрения здесь существенна не функция приложения сама по себе, а способ его организации. Проект удерживался как связка продукта и когнитивного контекста. В репозитории были зафиксированы документация, ограничения, структура контекста, правила исполнения и история изменений. За счёт этого можно было наблюдать не только очередную реализацию, но и зависимость результата от того текста, который агенту был доступен в работе.

Код всех четырёх версий доступен через prм-реестр на странице [@flancer32/spotify-playlist](#).

Полный доступ к примеру

Покупатели книги могут запросить у автора полный репозиторий проекта через контакты на странице `Leanpub`.

В него входят:

- вся история коммитов;
- весь когнитивный контекст (ctx/);
- все промежуточные состояния проекта между генерациями.

Это позволяет не просто прочитать пример, а воспроизвести его: проследить, как менялся контекст и как это влияло на поведение агента.

Базовые генерации

Экспериментальная последовательность включала четыре версии: v1.0.0, v2.0.0, v3.0.0 и v4.0.0. Первые три версии были независимыми регенерациями продукта из одного и того же когнитивного контекста. Контекст в ctx/ между этими тремя проходами не менялся. Менялись только тактические условия запуска: использовалась одна и та же модель GPT-5.4, но с разным уровнем рассуждения. Для версий v1.0.0, v2.0.0 и v3.0.0 это были соответственно high, medium и low.

Во всей последовательности использовалась агентная линия Codex, а во второй, третьей и четвертой версиях README прямо фиксирует Codex GPT-5.4. Для данного примера различие уровней рассуждения важно не как сравнение режимов модели, а как наблюдение предела их влияния. Изменение уровня рассуждения влияло на форму реализации, декомпозицию модулей и стиль кода, но не меняло повторяющийся выбор устаревшего API-паттерна.

С точки зрения функции приложения эти регенерации оставались близкими. Агент каждый раз восстанавливал один и тот же продуктовый контур: чтение файла, поиск треков через Spotify, создание плейлиста, добавление найденных треков, поддержку `--dry-run`, работу с OAuth callback и сопутствующие тесты. Реализация менялась, но продукт оставался тем же. Для разработки это важное наблюдение: при неизменном контексте проект воспроизводил не произвольный код, а ограниченный диапазон решений, совместимый с уже зафиксированным знанием о приложении.

Инвариант результата

На первых трёх версиях хорошо видно, что управляемость не означает буквального совпадения исходников. ADSM не предполагает, что агент будет каждый раз строить один и тот же файл с точностью до строк. Для разработки значим не такой уровень тождества, а устойчивость продуктовой формы и допустимого диапазона реализации.

Во всех трёх регенерациях проект снова получал тот же тип приложения, ту же базовую структуру, те же режимы запуска и ту же общую функциональность. Агент не переходил к другой архитектурной идее, не превращал CLI в иной продукт и не терял основные ограничения задачи. Это означает, что неизменный когнитивный контекст действительно удерживал генерацию в рабочем коридоре.

Практический смысл этого различия становится заметнее, если сопоставить его не с one-shot как таковым, а с разницей между инженерной организацией контекста и импровизационной работой. One-shot запуск тоже может дать близкий результат, если в стартовый запрос поместить весь необходимый когнитивный контекст. Но такой подход требует большого стартового промпта, может упереться в пределы рабочего контекста модели и, что важнее, плохо удерживает повторяемость на уровне самого человека. В вайб-кодинге человек не способен воспроизводить собственный предыдущий проход с той же точностью, с какой агент обрабатывает поданный ему текст.

ADSM решает эту проблему иначе. Он не пытается каждый раз целиком перенести проект в один стартовый запрос, а удерживает знания в контексте проекта и позволяет собирать из них рабочий контекст под конкретный этап. За счёт этого повторяемость достигается не только между запусками агента, но и на уровне организации работы разработчика. Именно такая форма управляемости и проявляется в рассматриваемой последовательности регенераций.

Устойчивая ошибка

На том же примере проявляется и граница такой устойчивости. Контекст удерживал не только полезный диапазон решений, но и повторяющийся дефект. В первых трёх версиях агент стабильно возвращался к устаревшей схеме Spotify Web API. Он воспроизводил более вероятный паттерн, усвоенный при обучении на более ранних источниках, вместо актуальной схемы, уже требуемой платформой.

В первой версии это проявилось прямо. Создание плейлиста было построено через `POST /v1/users/{user_id}/playlists` вместо `POST /v1/me/playlists`, а добавление треков через `POST /v1/playlists/{playlist_id}/tracks` вместо `POST /v1/playlists/{playlist_id}/items`. Во второй и третьей версиях агент возвращался к тому же классу решения. Ошибка повторялась между независимыми регенерациями, хотя код в каждом случае собирался заново.

Для анализа здесь важен не сам endpoint, а характер явления. Речь идёт не о разовой локальной ошибке, а о воспроизводимом выборе старого паттерна. Такое поведение указывает не на случайный дефект текущего запуска, а на пробел в текстовой конфигурации проекта. Контекст удерживал общую форму приложения, но не содержал достаточно явного барьера, который отделял бы актуальный API-контракт от статистически более вероятного исторического паттерна, усвоенного моделью при обучении.

Именно поэтому этот эпизод важен для ADSM. Он показывает, что воспроизводимость относится не только к правильным решениям. Если в когнитивном контексте отсутствует значимое ограничение, проект может воспроизводимо получать и неправильный паттерн. Для разработки это особенно полезный сигнал, потому что он показывает, где именно текстовая опора проекта ещё недостаточна.

Изменение контекста

Переход к четвёртой версии не сопровождался сменой продукта, платформы или общей логики задачи. Управляющее изменение было внесено именно в когнитивный контекст. Ключевым здесь стал коммит `be51f2274354ab1fef2c4b14e5ee15fb3f6e38df` от 2026-04-20.

В этой правке контекст был усилен сразу в нескольких местах. На уровне описания продукта были явно зафиксированы авторитетные ссылки на документацию Spotify, требование опираться на актуальную неустаревшую документацию и допустимая поверхность API для данного продукта. Там же были прямо запрещены устаревшие схемы работы с плейлистами. На уровне кодовой документации появилась верификационная опора, закрепляющая обязательный выбор endpoint и требующая, чтобы тестовое покрытие удерживало именно этот контракт.

Содержательно эта правка фиксировала типовые отклонения, проявившиеся в первых трёх версиях. Во всех трёх случаях агент возвращался к устаревшему Spotify API. Кроме этого, были зафиксированы и отклонения, проявлявшиеся не во всех версиях: отсутствие чтения .env и неустойчивость веб-сервера после запуска. Важен сам характер такого шага. Изменение касалось не кода продукта, а текста, который должен управлять будущей генерацией. Архитектура не переписывалась, агент не подменялся, постановка задачи не менялась. Проект лишь сделал явными те ограничения, которые раньше существовали как знания автора, но не были достаточно жёстко выражены в когнитивном контексте.

Так и проявляется управление в логике ADSM. Когда проект замечает устойчивый сбой, полезно не ограничиваться локальной правкой исходников. Более устойчивый шаг состоит в том, чтобы вернуть наблюдение в документацию и сделать его частью ограничений, которые будут действовать и в следующем цикле, и в следующей регенерации.

Повторная генерация

После этой правки была выполнена новая независимая регенерация, ставшая v4.0.0. Остальные условия эксперимента сохранялись настолько, насколько это вообще возможно в реальной работе: тот же проект, тот же класс задачи, та же модель GPT-5.4, та же ADSM-организация контекста. Существенно изменилось только одно: в проекте появились дополнительные текстовые ограничения против типовых отклонений предыдущих версий.

Смысл этого повтора состоял в проверке причинной связи. Если поведение агента действительно определяется доступным ему текстом, изменение текста должно изменить и характер регенерации. Если бы решающим оставался только случайный стиль запуска, такая связь выглядела бы слабее.

Поэтому четвёртая версия важна здесь не просто как следующий релиз. Она выступает как контрольный проход после изменения когнитивного контекста. В такой постановке можно проверить, изменилось ли поведение агента без замены модели и без вмешательства в код прошлых реализаций. Поскольку код всех релизов опубликован в pmt, а полный репозиторий с историей изменений доступен читателям книги по запросу, этот тезис остаётся проверяемым и вне рамок самого текста. При желании такой эксперимент можно воспроизвести самостоятельно и буквально проверить формулу: Code is cheap. Show me the spec.

Наблюдаемое изменение

В v4.0.0 агент сразу перешёл на требуемую актуальную схему Spotify API. Создание плейлиста использовало POST /v1/me/playlists, а добавление треков POST /v1/playlists/{playlist_id}/items. Старый паттерн больше не воспроизводился.

Практический смысл этого результата состоит в характере изменения. Поведение изменилось после изменения текста, а не после изменения модели. Проект не получил новую “более умную” конфигурацию агента. Он уточнил различия, существенные для данного приложения, и сделал их частью когнитивного контекста. После этого вероятность выбора нужных обращений к Spotify API и других требуемых решений стала выше, а поведение агента стало более определённым в значимых для проекта точках.

Здесь полезно различать два уровня влияния. Изменение уровня рассуждения в ранних версиях влияло на стиль кода и локальную форму реализации, но не устраняло возврат к старому паттерну. Добавление явного ограничения в контекст, напротив, сразу изменило поведение в значимой для проекта точке. Это означает, что часть сбоев следует искать не в “силе” модели, а в точности проектных границ и формулировок.

По этой причине четвёртая версия важна не только как успешный результат. Она служит наблюдаемым подтверждением того, что документация действительно управляет поведением агента. Пока требование существовало только в памяти автора, агент воспроизводил историческое решение. Когда это требование стало явной частью проектного текста, поведение изменилось уже в следующей регенерации.

Связь с ADSM

Этот пример практически связывает между собой несколько основных положений книги. Прежде всего он подтверждает первую аксиому: агент работает в текстовом режиме и строит результат из доступного ему текста. Код в этой схеме выступает производным артефактом. Когда меняется текстовая опора, меняется и результат генерации.

Не менее важна здесь и управляемость через текст. Разработчик влияет на поведение агента не через ожидание скрытого “понимания” модели, а через явное изменение когнитивного контекста проекта. Для разработки это означает смещение центра тяжести: существенная часть управления переносится с постфактум-исправления кода на организацию текста, который задаёт границы следующих реализаций.

На примере хорошо видна и роль формулировок. До правки проект содержал знание о продукте, но не отделял актуальный API-контракт от исторического паттерна достаточно явно. После правки это различие стало текстово выраженным и проверяемым. Для проекта это не косметическое улучшение документации, а усиление управляющего слоя.

Наконец, здесь проявляется и спиральная логика разработки. Ошибка была обнаружена в работе, затем наблюдение было возвращено в документацию, после чего обновлённый контекст стал основанием следующей регенерации. Так проект накапливает управляемость между витками: он не только исправляет код, но и усиливает условия будущего исполнения.

Итог

Практический смысл этого примера сводится к одному наблюдению. Агент не удерживал проект через внутреннюю память о прошлых версиях и не “понимал” его сам по себе. Он воспроизводил тот диапазон решений, который был допустим внутри доступного ему текста. Пока контекст не запрещал старый Spotify API-паттерн явно, этот паттерн стабильно возвращался. Когда запрет и правильная опора были закреплены в когнитивном контексте, поведение изменилось без смены модели и без ручной пересборки логики приложения.

Именно поэтому в ADSM документация не является вторичным описанием уже готового решения. Для разработки она выступает рабочим механизмом управления. На примере [@flancer32/spotify-playlist](#) это видно в сжатой и наблюдаемой форме: неизменный контекст удерживает инвариант результата, но способен удерживать и устойчивую ошибку, а уточнённый контекст изменяет поведение следующей генерации.

Этим практический пример завершает книгу. Всё, что раньше вводилось как инженерная рамка, здесь проявляется в одном конкретном наблюдении: разработка с агентом становится управляемой тогда, когда проект умеет переводить значимое знание в текст, удерживать его в когнитивном контексте и возвращать туда результаты обратной связи.