

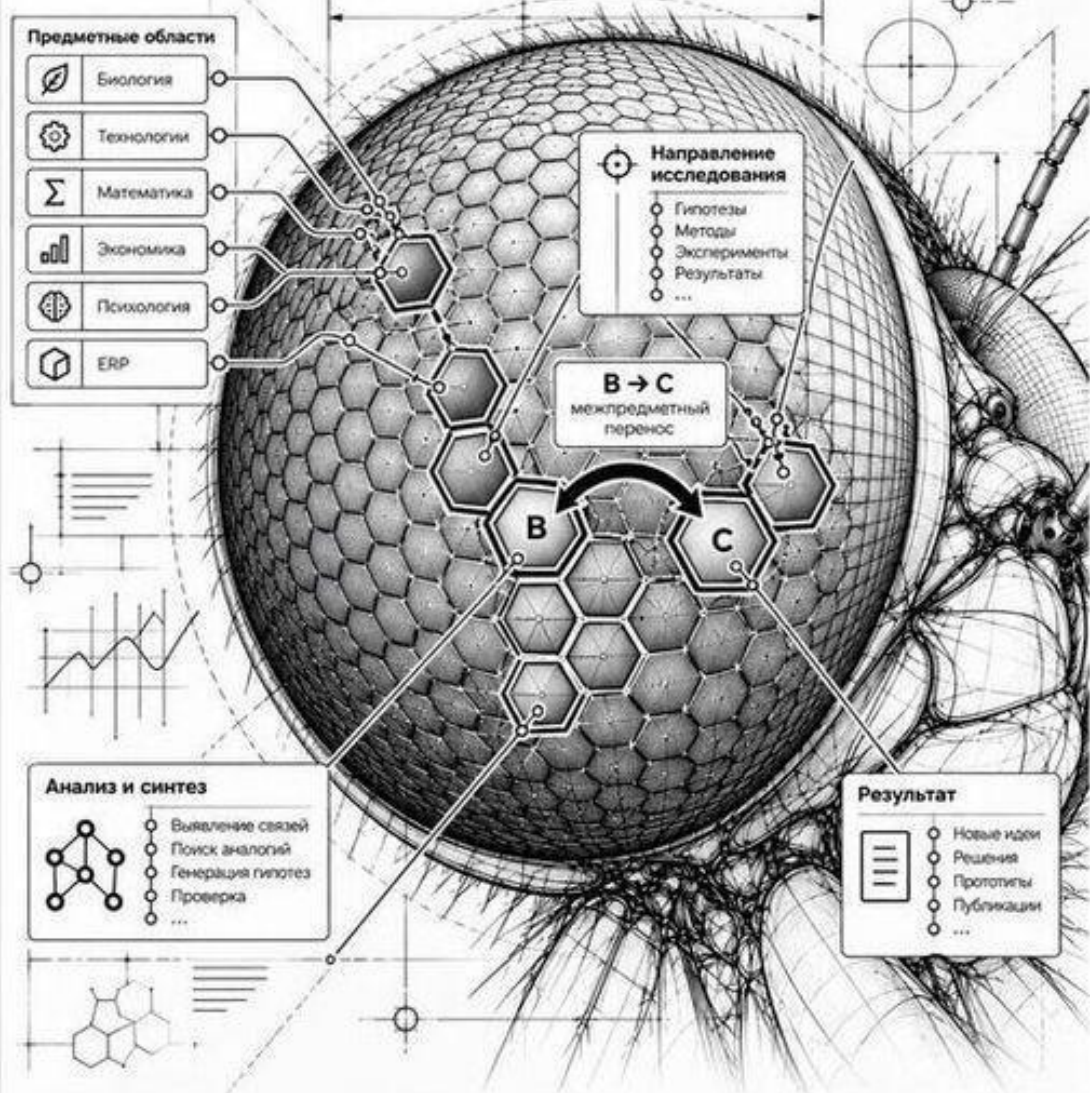
12+

Кирилл Ледовский

Глаз стрекозы

Практическая технология исследования с ИИ

Таксоны, фасеты и межпредметный перенос



Кирилл Ледовский

Глаз стрекозы

«Издательские решения»

Ледовский К.

Глаз стрекозы / К. Ледовский — «Издательские решения»,

«Глаз стрекозы» — практическая книга о многопредметном инженерном исследовании в эпоху ИИ. Она показывает, как одиночный исследователь может собирать несколько предметных миров, различать таксоны, находить переносимые механизмы, защищаться от ложной аналогии, связывать утверждения с доказательствами и вовремя останавливать поиск.

© Ледовский К.

© Издательские решения

Содержание

Аннотация	7
Введение. Зачем вообще нужен «Глаз стрекозы»	8
Часть I. Что это за методология	10
Глава 1. Когда ИИ слишком хорошо отвечает	11
1.2. Первый холодный душ: предприятие не хочет отдавать себя наружу	11
1.3. Агент и цифровая реплика — не одно и то же	12
1.4. Обычный ИИ начинает везти исследователя дальше	12
1.5. Достаём «Глаз стрекозы»	12
1.6. Теперь можно вернуться к донорам — но уже с вопросом	13
1.7. А если взять сильную методику и прогнать через неё всё?	13
1.8. Первый большой результат — не расширение, а сокращение	14
Глава 2. «Глаз стрекозы»: от метафоры к инженерной архитектуре	15
2.1. Короткая карта терминов	15
2.3. Назовём методологию строго	15
2.4. Общая архитектура	16
2.5. Это не конвейер	16
2.6. Пять дисциплин	16
2.7. Чего методология не обещает	16
2.8. Теперь вернёмся к нашему продукту	17
Глава 3. Product Brief и Research Direction: сначала продукт, потом исследование	18
3.1. Product Brief появляется не из методологии, а из тупика	18
3.3. Теперь появляется второй вопрос: что мне ещё неизвестно?	19
3.5. Границы: где исследование обязано остановиться	19
3.6. Доказательную стратегию лучше выбрать до того, как понравился ответ	20
3.7. Output Contract: чем направление обязано закончиться	20
3.8. Product Reframe: если исследование изменило сам продукт	20
3.9. Product Brief не выбирает науки заранее	20
3.10. От прототипа к промышленной реализации	21
3.11. Две короткие карточки вместо бюрократии	21
3.12. Теперь можно открывать предметные миры	21
Часть II. Захват предметной реальности	22
Глава 4. Предметные фасеты: как не смешать разные миры	23
4.1. Тема — ещё не предметный мир	23
4.2. Что должно быть внутри фасеты	23
4.3. Одинаковое слово — разные фасеты	24
4.4. Фасета не обязана совпадать с научной дисциплиной	24

4.5. Роль фасеты относительно продукта	25
4.6. Как фасета появляется из реального GAP	25
4.7. Что происходит, если границы фасеты не удержать	25
Конец ознакомительного фрагмента.	27

Глаз стрекозы

Кирилл Ледовский

© Кирилл Ледовский, 2026

ISBN 978-5-0071-4688-3

Создано в интеллектуальной издательской системе Ridero

Аннотация

«Глаз стрекозы» — таксоно-онтологическая методология многопредметного инженерного исследования с управляемым межпредметным переносом.

«Глаз стрекозы» — практическая книга о многопредметном инженерном исследовании в эпоху ИИ. Она показывает, как одиночный исследователь может собирать несколько предметных миров, различать таксоны, находить переносимые механизмы, защищаться от ложной аналогии, связывать утверждения с доказательствами и вовремя останавливать поиск. Сквозной кейс Universal DR демонстрирует, как продуктовая постановка меняет сам Рабочий Глаз исследования. Книга предназначена для ИТ-архитекторов, аналитиков, инженеров и руководителей, которым нужен не ещё один набор промптов, а воспроизводимая дисциплина работы с неизвестным.

Введение. Зачем вообще нужен «Глаз стрекозы»

Эта книга выросла из довольно простого раздражения. Современный ИИ очень хорошо умеет продолжать мысль. Стоит задать ему сложный инженерный вопрос, и через несколько минут на экране уже появляется целый мир: подходы, архитектуры, аналоги, фреймворки, статьи, инструменты, варианты развития. Иногда ответ настолько связный, что хочется просто идти по предложенному маршруту.

Именно в этот момент исследователь становится особенно уязвим. Не потому, что ИИ обязательно ошибается. Наоборот, опаснее другое: он часто предлагает слишком много разумного сразу. Из каждого разумного ответа естественно вытекает следующий. Через несколько часов исходная задача обрастает десятками предметных областей, красивых аналогий и потенциально полезных технологий. Где-то среди них действительно находится нужная конструкция. Но чем дальше, тем труднее понять, что относится к продукту, что является лишь интересной ветвью, а что вообще держится на одном удачном слове.

«Глаз стрекозы» появился как попытка дисциплинировать именно этот режим работы. Не запретить ИИ генерировать и не заменить специализированные методы новой сверхметодикой, а научиться управлять исследовательской оптикой: какие предметные миры сейчас нужны, что именно мы пытаемся перенести между ними, как не перепутать одинаковые слова с одинаковыми предметами, чем подтвердить вывод и когда уже достаточно.

В книге много английских терминов: Product Brief, Working Eye, Evidence, Claim, Transfer. Они используются не ради моды. Часть понятий родилась внутри нашей проектной технологии, часть пришла из сложившейся инженерной практики. При первом появлении я стараюсь давать короткий человеческий смысл, чтобы читатель не выпадал из сюжета.

Книга не требует предварительно знать онтологии, системную инженерию или теорию инноваций. Но она предполагает другое: читателю интересно, как устроена сложная инженерная мысль. Это может быть архитектор, аналитик, руководитель ИТ-направления, инженер, исследователь, консультант — человек, который уже заметил, что хороший ответ ИИ и хорошее инженерное решение не одно и то же.

Повествование построено немного необычно. Основной текст остаётся нормальным инженерным non-fiction. Но иногда мы будем вставать за спиной одиночки-исследователя и слышать короткий внутренний разговор. Не художественную сцену, а одну-две реплики, которые обычно и возникают в реальной работе: «а что это вообще значит?», «слишком красиво», «не прошло», «ну-ка посмотрим». После такой перебивки мы возвращаемся к спокойному разбору и только потом даём формальное определение.

Это сознательный порядок. Термин лучше запоминается, когда читатель уже почувствовал проблему, которую он решает. Таксон не нужен до тех пор, пока одинаковое слово не начало означать разные предметы. Anti-analogy не нужна до тех пор, пока красивая аналогия не стала слишком убедительной. Stop не нужен до тех пор, пока исследование не научилось продолжаться бесконечно.

В качестве сквозного кейса используется Universal DR — наша авторская линия универсальной Digital Replica, цифровой реплики профессиональной деятельности. Кейс выбран не потому, что он завершён и идеален. Наоборот, он интересен тем, что в нём исследовательская оптика реально менялась под давлением продукта: широкие предметные ветви сокращались, появлялись новые GAP, менялись роли доноров, семантическое ядро отделялось от runtime. Это позволяет показать методологию не в виде учебной схемы, а в работе.

При этом книга не является руководством по Universal DR. Тот же маршрут можно применить к производственной технологии, медицинскому изделию, информационной системе,

новому компоненту ERP, исследовательской гипотезе. Нас интересует не конкретный продукт, а инженерная дисциплина вокруг неизвестного.

Главный вопрос книги можно сформулировать так: если ИИ сделал дешёвыми поиск, генерацию и переход между областями, что теперь становится главным ограничением исследователя? Мой ответ — умение различать предметы, выбирать оптику, управлять переносом, проверять утверждения и вовремя останавливаться.

Ну что, начнём? Для начала попробуем сделать самого обычного «умного агента». Именно там всё и сломается.

Часть I. Что это за методология

Начнём не с определения. Представим человека, который пытается собрать серьёзного цифрового помощника для предприятия. ИИ охотно объясняет ему, что почти всё уже существует. Чем дальше он читает, тем убедительнее выглядит ответ. И именно в этот момент начинается настоящая проблема: красивый ответ ещё не означает, что перед нами нужный продукт.

Глава 1. Когда ИИ слишком хорошо отвечает

Представим, что наш одиночка-исследователь решил собрать интеллектуального агента общего назначения для серьёзной профессиональной работы. В первом разговоре он может даже назвать его AGI-агентом. AGI — Artificial General Intelligence, искусственный общий интеллект. Здесь это пока не техническое обещание настоящего AGI, а удобная исходная мечта: «хочу умного универсального помощника, который сам понимает задачу, помнит контекст, вызывает инструменты и доводит работу до результата».

Так. Нужен умный агент для предприятия.

Ну-ка, расскажи, что уже придумали умные люди.

1.1. Первый ответ выглядит почти слишком хорошо

Современная языковая модель на такой запрос отвечает уверенно. Она перечисляет agent frameworks, память, вызов инструментов, оркестрацию, планирование, approvals, retrieval, долговременный контекст. Подсказывает крупные экосистемы, где уже есть готовые части. Показывает, как связать модель с инструментами и как сделать цикл «получил задачу — выбрал действие — вызвал инструмент — проверил результат».

Если человек только входит в тему, эффект сильный. Возникает ощущение, что поезд уже ушёл: всё придумано, всё собрано, остаётся выбрать библиотеку и написать несколько интеграций.

Ничего себе. Так я, похоже, опоздал. Уже всё сделали.

И вот здесь полезно не спорить с ИИ, а просто продолжать задавать вопросы. Хороший ответ редко ломается на первом уровне. Он начинает трещать, когда в него входят реальные ограничения будущего изделия.

1.2. Первый холодный душ: предприятие не хочет отдавать себя наружу

Возьмём тяжёлый, но совершенно обычный для промышленности случай: продукт должен работать в закрытом корпоративном контуре. Внутри — договоры, цены, технологические документы, производственные планы, переписка, нормативы. В некоторых организациях внешний интернет ограничен или отсутствует вовсе. Для оборонного предприятия вопрос может закрыться ещё быстрее: внешний облачный сервис просто не является допустимым эксплуатационным контуром.

Так. Внешний сервис отпадает. Можно всё это поставить внутрь?

Ответ «можно сделать локально» звучит просто, пока мы не разберём, что именно означает «всё это». Локально должны жить не только модель и интерфейс. Нужны профессиональная память, активный контекст, работа с инструментами, маршрутизация, состояния, полномочия, версии, контроль исполнения и восстановление после сбоев. Одна фраза «работаем без внешнего облака» внезапно превращается в самостоятельную инженерную программу.

Понятно. Я думал, мне нужна модель. А мне, кажется, нужна целая система.

Именно здесь заканчивается разговор о выборе очередного чат-бота и начинается разговор о продукте.

1.3. Агент и цифровая реплика — не одно и то же

На этом месте возникает ещё один соблазн: назвать будущую систему AI-agent и считать вопрос закрытым. Но агент прежде всего описывает способ исполнения: принять цель, выбрать действие, вызвать модель или инструмент, получить наблюдение и продолжить цикл. Для длительной профессиональной работы этого мало.

В этой книге используется авторское понятие Digital Replica, сокращённо DR — цифровая реплика. Под ним понимается не просто агент, а устойчивый цифровой профессиональный субъект или реплика профессиональной деятельности: с собственной идентичностью, профессиональной памятью, версионизируемым состоянием, границами полномочий, рабочим контекстом, происхождением принятых решений и контролируемым исполнением.

То есть агент умеет действовать. А мне ещё нужно, чтобы он не забывал, кто он, что уже принято и что ему вообще разрешено.

Эта разница кажется словесной, пока система живёт один вечер. Но как только она должна продолжать работу завтра, восстановить состояние после переноса, различить черновую гипотезу и утверждённую версию, передать результат другому исполнителю и не изменить канон без полномочия, разница становится архитектурной.

Конкретную линию такой универсальной цифровой реплики дальше будем называть Universal DR. Это рабочее авторское название продукта, а не общепромышленный стандарт.

1.4. Обычный ИИ начинает везти исследователя дальше

Теперь задача стала интереснее. Нужна уже не просто модель, а Digital Replica. Что дальше делает обычный ИИ? Он продолжает помогать. Для памяти предлагает одни архитектуры, для автономности — другие, для безопасного исполнения — третьи, для планирования — четвёртые. Можно заглянуть в когнитивные системы, организационную память, системную инженерию, safety-critical autonomy, agent runtime, устойчивость.

Это тоже полезно. И это. И вот это, кажется, вообще обязательно.

Через некоторое время на столе уже не один вопрос, а целая россыпь умных областей. Каждая что-то обещает. Именно это делает современный ИИ опасно убедительным: он редко останавливает исследователя словами «вам это сейчас не нужно». Он умеет продолжать.

Погодите. Я вообще ещё свой продукт строю — или уже изучаю половину человеческого знания?

Вот теперь у нас впервые появляется проблема, ради которой и нужна специальная исследовательская дисциплина. Не недостаток идей, а их избыток. Не отсутствие хороших методов, а отсутствие критерия: что действительно нужно именно этому продукту.

1.5. Достаём «Глаз стрекозы»

Предположим, у исследователя уже есть экспериментальная технология, которой он пользуется именно тогда, когда обычный поиск начинает расплзаться. Она называется «Глаз стрекозы» — Dragonfly Eye.

Так не пойдёт. У меня как раз есть штука, которая умеет охлаждать слишком красивые ответы.

Название связано с фасеточным зрением. У стрекозы глаз собран из множества отдельных зрительных элементов. В нашей методологии аналогия очень ограниченная, но полезная:

сложная задача рассматривается через несколько самостоятельных предметных миров. Они не смешиваются в общую кашу, а сохраняют собственные границы. Позже мы назовём такие предметные миры фасетами.

Важно: «Глаз стрекозы» не просит ИИ придумать ещё больше областей. Он сначала задаёт дисциплину различения. Что за предмет перед нами? Что именно мы хотим перенести? В чём механизм? Где доказательства? Что изменится в продукте? Что произойдёт, если эту ветвь вообще убрать?

Ну-ка ещё раз. Не «что ещё можно почитать?», а «что именно мне сейчас нужно и почему?»

И происходит неприятная, но полезная вещь: часть прежнего великолепия начинает исчезать. Некоторые направления оказываются дублирующими. Некоторые — интересными, но преждевременными. Некоторые дают только метафору, но не механизм. А несколько областей, наоборот, неожиданно становятся центральными, потому что закрывают реальный разрыв продукта.

1.6. Теперь можно вернуться к донорам — но уже с вопросом

После такой очистки снова имеет смысл идти смотреть чужие решения. Например, если требуется механика agent runtime, можно изучать архитектуры OpenAI и других разработчиков агентных систем. Если нужна диагностика готовности системы к действию — полезно посмотреть, как подобные задачи решаются там, где ошибочный зелёный статус дорого стоит. Если нужна долговременная профессиональная память — стоит заглянуть в исследования памяти и организационной памяти.

Вот теперь понятно, зачем я туда иду.

Не за чужим продуктом. За конкретной конструкцией.

Это принципиальная смена режима. Раньше чужая область могла увлечь исследователя сама по себе. Теперь она рассматривается как донор: какую конкретную конструкцию она может дать, что из неё можно абстрагировать, что обязано сохраниться при переносе и где находятся границы применимости.

1.7. А если взять сильную методику и прогнать через неё всё?

Хорошо. А может, не нужен весь этот Глаз? Возьмём одну мощную методику — и всё.

Такой соблазн естественен. Но специализированные методики сильны именно потому, что хорошо организуют определённый тип работы.

ТРИЗ — теория решения изобретательских задач — нужна, когда приходится работать с инженерным противоречием и искать изменение конструкции, которое снимает конфликт требований. DOE, Design of Experiments, планирование эксперимента, нужно для системного изучения факторов и их влияния. Системная инженерия удерживает требования, архитектуру, интерфейсы и проверку сложной системы. Систематический обзор дисциплинирует доказательный поиск.

«Глаз стрекозы» находится уровнем выше. Он не заменяет эти методы, а решает, какой предметный мир нужен, какой метод вызвать, какой контекст ему передать, как вернуть результат и как не позволить локальному инструменту захватить всё исследование.

Противоречие? Зовём ТРИЗ. Эксперимент? Зовём DOE. Но мастерскую отвёрткой не заменяют.

1.8. Первый большой результат — не расширение, а сокращение

В исследовании Universal DR именно это и произошло. Сначала был широкий запрос и широкий набор областей, которые казались полезными. После продуктовой постановки этот набор пришлось пересмотреть. Часть направлений стала INCLUDE, часть EXCLUDE, часть DEFER — включить, исключить или отложить.

Только теперь можно назвать получившуюся конфигурацию термином методологии: это первая версия Working Eye — Рабочего Глаза, то есть временной, версионизируемой сборки предметных фасет под конкретное направление исследования и конкретный продукт.

Вот теперь название имеет смысл. Не «чем шире глаз, тем лучше», а «вижу ровно то, что нужно для этой задачи».

Рабочий Глаз не должен быть максимально широким. Он должен быть минимально достаточным для создания и проверки конкретного продукта при текущем горизонте доказательств.

Ладно. А из чего этот Глаз вообще собирается — и почему именно «стрекозы»?

Глава 2. «Глаз стрекозы»: от метафоры к инженерной архитектуре

Теперь у названия появился контекст. Мы пришли к «Глазу стрекозы» не потому, что захотелось красивой метафоры, а потому, что обычный AI-поиск слишком легко разрастается и смешивает разные предметные миры. Прежде чем разбирать архитектуру подробно, договоримся о нескольких словах.

2.1. Короткая карта терминов

Термин	Рабочий смысл в этой книге
Одиночка-исследователь	Человек, который ведёт исследование с помощью ИИ, но сам сохраняет постановку задачи, ответственность за смысл и решение о достаточности.
AI-agent	Исполнительная конструкция, способная получать цель, использовать модель и инструменты, наблюдать результат и продолжать цикл действий.
Digital Replica / DR	Авторская цифровая реплика профессиональной деятельности: идентичность, память, версии, полномочия, контекст и контролируемое исполнение.
Фасета	Самостоятельный предметный мир со своей границей, словарём, таксонами, механизмами и доказательствами.
Working Eye / Рабочий Глаз	Временная версионированная сборка фасет под конкретный продукт и Research Direction.
Межпредметный перенос	Не перенос слов, а перенос конкретной предметной конструкции с проверкой инварианта, границ и доказательств.

Фасеточный глаз хорошо передаёт одну важную особенность многопредметного исследования. Несколько предметных миров не должны растворяться в общей «междисциплинарности». Каждая фасета сохраняет собственную границу, словарь, таксоны, механизмы и доказательства. Исследователь получает не одно размытое изображение, а рабочую сборку различных предметных граней.

*То есть фасеты не надо смешивать.
Надо научиться смотреть ими одновременно — и понимать, откуда
пришла каждая конструкция.*

Поэтому выражение «посмотреть на проблему с разных сторон» здесь недостаточно. Фасета должна быть оформлена так, чтобы её можно было версионировать, включать, исключать и повторно использовать, а конкретный перенос из одной фасеты в другую — проследить и попытаться опровергнуть.

Важно различать два понятия: «Глаз стрекозы» — вся методология; Рабочий Глаз — конкретная временная сборка фасет для одного направления исследования.

2.3. Назовём методологию строго

Полное название — таксоно-онтологическая методология многопредметного инженерного исследования с управляемым межпредметным переносом.

«Таксоно-онтологическая» означает, что работа начинается не с ассоциаций слов, а с различения предметов, границ, состояний, отношений и механизмов. «Многопредметная» — что одной инженерной задаче могут понадобиться несколько самостоятельных предметных миров. «Инженерное исследование» — что исследование обслуживает создаваемый результат.

«Управляемый межпредметный перенос» — что чужая конструкция проходит через абстракцию, инвариант, целевую интерпретацию, анти-аналогию и проверку.

2.4. Общая архитектура

Книжная версия методологии начинается с Product Intent / Product Brief — компактного описания целевого инженерного результата. Следующий узел — Research Direction / Input Contract: какой вопрос надо решить, какие границы действуют, какие доказательства допустимы и какой выход требуется.

Далее формируются фасеты и собирается Working Eye. Затем возникает Transfer — конкретный перенос между предметными мирами. Перенос входит в Evidence-контур. При локальной задаче вызываются специализированные Methods / Generator. После существенного результата выполняется Product Recheck: изменилось ли наше понимание самого продукта?

Затем выбирается Next Check — следующая проверка, которая ещё способна изменить решение. Stop / Reopen закрывает или открывает цикл, а Research Memory сохраняет версии продукта, фасет, переносов, доказательств, отказов и причин пересборки.

2.5. Это не конвейер

Слева направо получилось слишком аккуратно. Неужели опять waterfall?

Нет. Новый источник может изменить определение предмета. Контрпример — разрушить перенос. Испытание — открыть GAP. Изменившийся Product Brief — потребовать убрать старую фасету или добавить новую. Даже Stop относится только к текущей версии продукта и текущему evidence horizon.

Поэтому правильнее говорить о версионизируемом замкнутом цикле, а не о линейном маршруте.

2.6. Пять дисциплин

Методология одновременно удерживает пять дисциплин. Продуктовую — зачем исследование вообще ведётся. Предметную — что именно означает каждый объект. Дисциплину переноса — что можно и нельзя унести из соседней области. Доказательную — почему мы считаем утверждение принятым. И дисциплину памяти и остановки — что уже проверено, что отвергнуто и когда дальнейший поиск перестаёт менять решение.

2.7. Чего методология не обещает

Она не обещает, что один исследователь с ИИ эквивалентен профессиональной команде. Не обещает, что любой межпредметный перенос является изобретением. Не отменяет предметную экспертизу и внешнюю проверку там, где она нужна. И не объявляет единственное правильное разбиение знания на фасеты.

Один человек с ИИ не стал НИИ. Но он может быстрее понять, куда ему действительно нужен НИИ.

2.8. Теперь вернёмся к нашему продукту

Мы уже знаем, зачем понадобился Глаз и из каких крупных узлов состоит методология. Но у нашего исследователя остаётся практический вопрос. Он решил строить локальную Digital Replica. Что именно считать продуктом? Что считать обязательным свойством? Где граница между продуктом и исследовательской гипотезой?

«Глаз стрекозы» — это не набор взглядов, а система управления тем, какие предметные миры мы включаем, что между ними переносим и что считаем доказанным.

Хорошо. Теперь надо наконец записать: что именно я строю?

Глава 3. Product Brief и Research Direction: сначала продукт, потом исследование

Вернёмся к тому месту, где внешний облачный сервис уже отпал. Исследователь понял, что ему нужна локальная Digital Replica, но это пока всё ещё слишком широкая фраза. Если сейчас снова спросить ИИ «как её сделать», он опять развернёт десятки технологий. Значит, прежде чем искать дальше, надо зафиксировать сам продукт.

Так. Хватит пока искать. Сначала запишу, что у меня должно получиться.

3.1. Product Brief появляется не из методологии, а из тупика

Product Brief — короткий продуктовый контракт. Он нужен не потому, что так красивее устроена схема, а потому, что без него исследователь уже один раз почти утонул в хороших советах. Теперь каждая новая область должна отвечать на вопрос: какое обязательное свойство продукта она помогает реализовать или проверить?

Для нашего кейса Product Brief начинает складываться естественно. Продукт должен работать внутри локального контура. Не зависеть от одного внешнего облачного сервиса. Сохранять профессиональную память. Различать утверждённое и рабочее состояние. Вызывать только разрешённые способности. Не изменять каноническое состояние без контролируемого перехода. Восстанавливаться и переноситься между средами.

Вот теперь это уже не «хочу умного агента». Это вполне конкретный продукт с неприятными обязательствами.

3.2. Что должно быть в минимальном Product Brief

Product Brief в этой книге намеренно короткий. Это не техническое задание на сотни страниц и не backlog. Его задача — удержать критерий полезности исследования.

Поле	Что фиксируем
Целевой продукт	Не тему, а результат, который должен существовать после работы.
Пользователь / владелец	Кто применяет продукт и кто имеет право принимать ключевые решения.
Среда эксплуатации	Закрытая сеть, локальный контур, допустимость или запрет внешних сервисов.
Обязательные способности	Например: восстановить рабочий контекст, выбрать разрешённый инструмент, сохранить проверенный результат.
Запреты	Что продукт не имеет права делать: например, молча менять каноническое состояние.
Неотменяемые ограничения	Свойства, нарушение которых превращает решение в другой продукт.
Критерий готовности	Какой наблюдаемый результат позволяет перейти к следующей стадии.
Архитектурные качества	Локальность, воспроизводимость, прослеживаемость, управляемые изменения, единый контур полномочий.
Граница реализации	Концепт, архитектура, демонстрационный прототип, локальный runtime или проверенный компонент.

Здесь полезен ещё один практический приём. На ранней стадии не обязательно сразу поднимать промышленное хранилище. Если нужно проверить семантику, версии, контракты и переходы, переносимое машинное ядро можно сначала собрать на обычном Excel-файле. Excel

здесь не «база будущей системы», а прозрачный канонический носитель смысла для прототипа. Позже этот смысл можно перенести в промышленное хранилище, не меняя саму предметную модель.

Набросаю сначала на Excel. Если смысл живой — в промышленную БД переползём потом.

Это важное различие между смыслом и реализацией. Хранилище можно заменить. Канонические определения, версии, полномочия и контракты должны пережить замену.

3.3. Теперь появляется второй вопрос: что мне ещё неизвестно?

Как только Product Brief записан, возникает Research Direction — направление исследования. Это уже не описание продукта, а контракт на неизвестное: что именно требуется узнать, доказать, сравнить или опровергнуть, чтобы продвинуть продукт.

Продукт записал. Теперь — что именно мешает мне его построить?

Research Direction фиксирует вопрос или цель, объект, границы, исходные факты, ограничения, критерии результата и доказательную стратегию. В product-directed версии добавляется происхождение вопроса: какой пункт Product Brief или какой продуктовый GAP породил это исследование.

Например, Product Brief требует локального контролируемого исполнения. Тогда Research Direction может звучать так: «Какая архитектурная организация обеспечивает локальное исполнение Digital Replica, при котором активный контекст, вызов инструментов, полномочия и запись в профессиональную память остаются управляемыми и воспроизводимыми?»

Вот это уже вопрос. А не «докажи, что мой любимый фреймворк правильный».

3.4. Research Direction не должен подсказывать ответ

Плохой исследовательский вопрос содержит желаемый результат: «доказать, что архитектура X подходит». Хороший фиксирует требование и оставляет пространство для конкурирующих решений.

Именно здесь Product Brief защищает исследование от вкусовщины. Нам может нравиться определённый agent framework или конкретный поставщик. Но в Research Direction мы должны проверять не симпатию к реализации, а способность конструкции удовлетворить продуктовые ограничения.

OpenAI красиво устроено? Возможно. Но я пришёл туда за механизмом, а не за правом скопировать весь продукт.

3.5. Границы: где исследование обязано остановиться

В AI-среде отсутствие границы почти гарантированно превращается в расширение вопроса. Поэтому Research Direction фиксирует system-of-interest, IN-SCOPE и OUT-OF-SCOPE.

Если сейчас мы исследуем контролируемое локальное исполнение, философия сознания может быть интересна, но она не обязана находиться внутри текущего контура. Если позже обнаружится конкретный GAP, который без неё нельзя закрыть, граница будет пересмотрена новой версией.

А давайте ещё сознание посмотрим. Нет. Пока не за что его сюда тащить.

3.6. Доказательную стратегию лучше выбрать до того, как понравился ответ

Evidence Strategy задаётся до генерации итоговых выводов. В базовой форме есть два контура: внешний поиск опор и собственные проверки — расчёты, стенды, исполнение, воспроизводимые эксперименты.

Разные утверждения требуют разных доказательств. Что определённая конструкция существует у донора — вопрос источников. Что она работает в нашем runtime — вопрос исполнения. Что другой человек сможет воспроизвести сборку — вопрос независимого повторного прохода. Что система допустима в конкретной регулируемой среде — может потребовать внешнюю экспертизу.

Сначала понравился ответ, потом придумал проверку?

Нет. Так можно доказать себе что угодно.

3.7. Output Contract: чем направление обязано закончиться

У входного контракта должен быть выходной. Output Contract перечисляет, что должно существовать, когда мы решим закрыть Research Direction: проверяемые выводы, альтернативы, отрицательные результаты, evidence trace, использованные фасеты, состоявшиеся и отклонённые переносы, результаты собственных проверок, GAP и следующий вопрос.

Product-directed версия добавляет Product Delta — что именно исследование изменило в продукте. Иногда мы подтверждаем способность. Иногда удаляем её. Иногда Product Brief вообще не меняется, но закрывается критический долг проверки. Все три результата нормальны.

3.8. Product Reframe: если исследование изменило сам продукт

Серьёзное исследование иногда меняет исходную постановку. Это не обязательно ошибка. Управляемый Product Reframe имеет причину: новый факт, противоречие, провал испытания, новый GAP или доказанный архитектурный предел.

Тогда создаётся новая версия Product Brief. Старый набор предметных областей не переносится автоматически. Каждая ранее включённая область проходит повторную проверку: всё ещё ли она нужна новому продукту?

Знания не исчезли.

Просто продукт изменился — и старый набор вопросов больше не святыня.

3.9. Product Brief не выбирает науки заранее

Это правило принципиально. Product Brief не содержит списка «правильных областей». Он задаёт критерий полезности. Новая фасета может появиться позже, если она закрывает реальный продуктовый GAP.

В Universal DR именно так позднее появились архитектурные и agent-runtime доноры. Они вошли не потому, что хотелось расширить обзор, а потому, что обнаружился конкретный execution GAP: семантическое ядро было развито лучше, чем контролируемая механика его эксплуатации.

Так вот зачем я снова полез к агентам.

Не потому, что они модные. Потому что у меня здесь дырка.

Это и есть ключевое различие между product-directed исследованием и заранее закрытой продуктовой декомпозицией. Продукт определяет, зачем искать; исследование сохраняет свободу обнаружить неожиданное.

3.10. От прототипа к промышленной реализации

На этой стадии полезно ещё раз развести семантику и реализацию. Допустим, первая версия канонического ядра живёт в Excel. Это удобно: структура прозрачна, версии сравнимы, контракты можно читать глазами и машинно. Позже промышленный продукт может хранить те же сущности в СУБД, специализированном хранилище или другом локальном контуре.

Если при переносе приходится заново придумывать смысл сущностей, значит, в прототипе мы моделировали не канон, а случайную реализацию. Хороший Product Brief и Research Direction помогают заметить эту ошибку заранее.

Хранилище поменяем.

Смысл менять не должны.

3.11. Две короткие карточки вместо бюрократии

Практический старт можно свести к двум карточкам. В Product Brief фиксируются продукт, пользователь, среда, обязательные способности, запреты, неотменяемые ограничения, критерий готовности и граница реализации. В Research Direction — вопрос, объект, границы, факты, ограничения, Evidence Strategy, критерии результата и Output Contract.

Новая область просится в исследование?

Какой пункт одной из этих карточек требует её присутствия?

Если ответа пока нет, направление не обязано быть ошибочным. Но оно остаётся кандидатом и не расширяет исследовательскую оптику автоматически.

3.12. Теперь можно открывать предметные миры

Мы наконец удержали обе стороны задачи. Product Brief фиксирует, что строим. Research Direction — что именно надо узнать и доказать. Теперь можно идти в соседние области уже не туристом, а исследователем с конкретным вопросом.

Продукт задаёт критерий полезности. Research Direction задаёт вопрос и доказательный контракт. Но этого ещё мало: теперь надо понять, как вообще устроен самостоятельный предметный мир и почему два одинаковых слова могут означать совершенно разные вещи.

Хорошо. Что искать — понятно.

А как не перепутать чужой предмет со своим?

Часть II. Захват предметной реальности

К этому моменту у исследователя уже есть две вещи, которых не было в самом начале: он понимает, какой продукт строит, и знает, какой исследовательский вопрос должен закрыть. Но перед ним остаётся самая неприятная часть работы. Мир вокруг продукта по-прежнему не организован. В документах одни и те же слова означают разное, в соседних дисциплинах одинаковые механизмы называются по-разному, а ИИ очень охотно склеивает всё это в гладкий текст. Поэтому следующая задача «Глаза стрекозы» — не искать больше знаний, а захватить предметную реальность так, чтобы с ней вообще можно было работать.

Глава 4. Предметные фасеты: как не смешать разные миры

Представим, что наш исследователь продолжает разбираться с локальной Digital Replica. Он уже понял, что ему нужна профессиональная память, контролируемое исполнение, восстановление состояния и работа в закрытом контуре. Обычный поиск тут же начинает подбрасывать знакомые слова: memory у AI-agent, working memory в когнитивных архитектурах, organizational memory в теории организаций, configuration baseline в системной инженерии. Все четыре направления похожи. Но похожи чем?

Так. Везде написано «память».

А память-то у них одна и та же?

4.1. Тема — ещё не предметный мир

Самая простая ошибка в многопредметном исследовании — назвать тему и решить, что предметная область уже определена. «Память», «устойчивость», «агенты», «безопасность», «риски» звучат как области знания. Но для исследовательской работы этого недостаточно. Тема не объясняет, какие сущности существуют внутри неё, какие состояния для них допустимы, какие механизмы считаются нормальными, какие источники доказывают утверждения и где заканчивается область применимости.

Если мы просто создадим папку «Память» и сложим туда всё найденное, получится аккуратный архив. Но уже через несколько дней в нём рядом окажутся память языковой модели, оперативная память вычислительной системы, рабочая память когнитивной архитектуры, архив документов, организационная память предприятия и профессиональная память Digital Replica. ИИ без труда напишет между ними переходы. Проблема только в том, что эти переходы могут существовать лишь на уровне слова.

Поэтому «Глаз стрекозы» требует более жёсткой единицы. Мы будем называть её предметной фасетой. Фасета — это самостоятельный предметный мир, у которого есть явная граница, собственный словарь, набор таксонов, отношения, механизмы, процессы, доказательная база и области возможного переноса.

Понятно.

Не папка с текстами. Маленький мир со своими законами.

4.2. Что должно быть внутри фасеты

У зрелой фасеты есть несколько обязательных слоёв. Первый — Boundary, граница. Нужно заранее написать, что считается внутри предметного мира, что остаётся снаружи и по какому правилу мы отличаем одно от другого. Граница не строится по ключевым словам. Слово «memory» может встретиться в десятке дисциплин, но это не делает их одной фасетой.

Второй слой — Specification, спецификация. Она отвечает на простые вопросы: что за объект мы рассматриваем, какой масштаб допустим, какие типовые вопросы здесь имеют смысл, какие единицы наблюдения используются и какие ограничения нельзя забывать. Такая спецификация нужна не для бюрократии. Она позволяет другому исследователю собрать тот же предметный мир и не получить совершенно другую область под тем же названием.

Третий слой — Vocabulary, словарь. Он фиксирует критические термины именно в контексте данной фасеты. Если одинаковое слово имеет два значения, это должно быть видно.

Если два разных слова обозначают один и тот же предмет, это тоже нужно показать. Синонимы и омонимы для ИИ особенно опасны: языковая модель умеет красиво поддерживать оба значения, не всегда замечая момент подмены.

Четвёртый слой — Тахоному, таксономия. Здесь появляются нормализованные предметы, которые позже мы разберём отдельно. Пока достаточно понимать: фасета не может жить только терминами. Внутри неё должны быть различимы сущности, их классы, состояния и отношения.

Пятый слой — Mechanisms и Processes. Предметный мир должен объяснять не только «что существует», но и «как это живёт». Механизм связывает условия, взаимодействующие предметы, преобразование и результат. Предметный процесс показывает динамику этого мира. Он не обязан совпадать с бизнес-процессом предприятия и тем более с экраным маршрутом ERP.

Шестой слой — Evidence. Существенные положения фасеты должны иметь опоры: источники, наблюдения, испытания, расчёты или явно зафиксированный GAP. Фасета, состоящая только из правдоподобных объяснений, удобна для разговора и опасна для переноса.

И, наконец, SUBJECT_TRANSFER_AREA — область возможного предметного переноса. Она не говорит, что перенос уже состоялся. Она лишь фиксирует, с какими соседними предметными мирами у этой фасеты есть потенциально полезная связь, какие конструкции могут оказаться переносимыми и какие ограничения надо проверить.

4.3. Одинаковое слово — разные фасеты

Вернёмся к памяти. В агентной архитектуре память часто рассматривается как механизм хранения и извлечения контекста, который нужен исполнителю между действиями. В когнитивной архитектуре working memory связана с текущим состоянием задачи, доступным для выбора следующего оператора. В организационной памяти обсуждаются способы, которыми организация удерживает опыт, решения, процедуры и знания независимо от конкретного человека. В нашей Digital Replica профессиональная память должна ещё хранить версии, происхождение решений, принятые результаты, отклонённые ветви и возможность восстановить рабочее состояние.

Эти предметные миры связаны. Но они не совпадают. Если мы сразу сведём их в один общий термин «память», то незаметно импортируем свойства одного мира в другой. Например, то, что запись можно извлечь из векторного хранилища, ещё не означает, что она является принятой профессиональной памятью. И наоборот, то, что организационное решение формально сохранено, не означает, что оно доступно агенту в текущем рабочем контексте.

Вот оно.

Одно слово, а четыре разных предмета. И каждый тянет за собой свои правила.

Фасета нужна именно затем, чтобы временно удержать эти миры раздельно. Позже мы сможем строить между ними мосты. Но сначала надо знать, с каких берегов этот мост вообще начинается.

4.4. Фасета не обязана совпадать с научной дисциплиной

Есть ещё одна ловушка. Увидев слово «предметный мир», легко решить, что каждая фасета должна совпадать с университетской дисциплиной: психология, теория управления, системная инженерия, биология. Это не так.

Граница фасеты определяется исследовательской задачей. Для одного продукта нам может понадобиться широкая фасета организационной памяти. Для другого — гораздо уже:

жизненный цикл профессиональной памяти цифрового субъекта. В исследовании устойчивости предприятия отдельной фасетой может стать управление жизнеспособными продолжениями при нарушении, хотя такой университетской кафедры не существует.

Это не произвол. Фасета должна оставаться воспроизводимой и предметно самостоятельной. Но она строится под исследовательскую оптику, а не под каталог научных факультетов. В этом смысле фасеты purpose-relative: их границы зависят от вопроса, но внутри границы правила должны быть строгими.

4.5. Роль фасеты относительно продукта

Product-directed версия «Глаза стрекозы» добавляет к фасете ещё один вопрос: зачем она нужна текущему продукту? Это не часть её предметного определения. Фасета не перестаёт быть самостоятельным миром. Но в конкретном Working Eye у неё должна быть роль.

Роль может быть разной. Одна фасета даёт semantic contribution — помогает понять, какие предметы и состояния вообще существуют. Другая приносит mechanism contribution — объясняет механизм, которого не было в исходной области. Третья даёт evidence contribution — независимый способ проверки. Четвёртая помогает архитектурой, пятая — реализационной конструкцией, шестая выявляет риск или запрет.

Такой Product Role защищает исследование от собирательства. Соседняя область может быть очень интересной и даже хорошо разработанной, но если мы не можем объяснить, что именно она добавляет продукту сейчас, она остаётся кандидатом.

Красиво? Очень.

Нужно продукту? Пока не вижу. Значит, не включаю.

Это не означает, что фасета навсегда выброшена. Может появиться новый GAP, и её роль изменится. Но Working Eye не должен платить за чужую красоту постоянным расширением.

4.6. Как фасета появляется из реального GAP

Лучший момент для появления новой фасеты — не когда исследователю стало любопытно, а когда он встретил разрыв, который не умеет объяснить текущей оптикой.

Так произошло в Universal DR. На раннем этапе можно было долго обсуждать семантическое ядро, память и версионирование. Но когда задача дошла до реального исполнения, обнаружился другой вопрос: как субъект понимает, что он вообще готов выполнить конкретную работу? Достаточен ли контекст? Загружена ли правильная версия технологии? Есть ли нужные полномочия? Не устарели ли источники? Может ли он выбрать допустимую способность, а при невозможности честно остановиться?

Такой разрыв выводит исследователя в соседние миры: системная диагностика, executive control, autonomic computing, когнитивное управление. Они входят не потому, что звучат современно. Они вошли потому, что текущий Product Brief создал конкретный GAP.

Вот теперь понятно, зачем идти наружу.

У меня вопрос есть. Не просто любопытство.

4.7. Что происходит, если границы фасеты не удержать

Представим обратную ситуацию. Исследователь увидел у NASA понятие health monitoring и решил: отлично, это и есть наша «готовность». Затем он нашёл confidence в когнитивной психологии и назвал это уровнем доверия системы к себе. Потом добавил MAPE-K из

autonomic computing и объявил его контроллером Digital Replica. На уровне текста всё выглядит очень убедительно.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.