ЦИФРОВОЙ МОЗГ

Твоя внешняя память

Как организовать заметки, знания, идеи и проекты в единую систему

ВАСИЛИЙ ЗАМЕТКИН



ПРАКТИЧЕСКОЕ РУКОВОДСТВО

шаблоны · AI-промпты

Василий Заметкин

Цифровой мозг

«Автор»
2026

Заметкин В.

Цифровой мозг / В. Заметкин — «Автор», 2026

Заметки, идеи, ссылки и проекты быстро превращаются в информационный хаос. «Цифровой мозг» поможет построить собственную систему внешней памяти: организовать знания, связать информацию и быстро находить нужное. В книге — пошаговая практическая часть, готовые шаблоны, AI-промпты и примеры, с которыми систему можно собрать самостоятельно без скачивания дополнительных материалов.

© Заметкин В., 2026
© Автор, 2026

Содержание

Выходные данные 6
От автора 7
Как читать эту книгу 8
- Что означают выделенные блоки 9
- Практические блоки 10
- Практический комплект 11

Введение. Почему мы сохраняем всё и не используем почти ничего 12
- Тысяча заметок, из которых ничего не найти 13
- Информации не слишком много. Её слишком много непереваренной 14
- Память — плохое место для обязательств 15
- Что обещает эта книга — и чего не обещает 16
- Кому это нужно 17
- Алексей 18
- Узнаёте ли вы себя? 19

Система, а не приложение 20
- Ваш мозг не обязан быть жёстким диском 21
  - Почему в голове помещается так мало 21
  - Выгружать наружу — нормально 21
  - Система, которой доверяют 22
  - Хранение — это не мышление 22
  - Что мы будем строить 22
  - Что дальше 23
- Что такое второй мозг и PKM 24
  - Идея не новая 24
  - Три уровня ценности 24
  - Чем второй мозг не является 25
  - Что должно быть внутри 26
  - При чём тут Алексей 26
- Почему прочитать — не значит выучить 27
  - Пять состояний информации 27
  - Идея первая: вспоминать полезнее, чем перечитывать 27
  - Идея вторая: интервалы 28
  - Идея третья: сформулированное своими словами держится лучше 28
  - Идея четвёртая: объяснять — значит понимать 29
  - Идея пятая: перемешивание 29
  - Что со всем этим делать на практике 29
- От потребителя информации к создателю знаний 31
  - Цикл, который не работает 31
  - Цикл, который работает 31
  - Фильтр на входе 32
  - Информационная диета без фанатизма 32
  - Система самообразования 33
  - Проектное обучение 33
  - Как это будет выглядеть в системе 34

Жизненный цикл информации 35
Шесть шагов 35
Где ломается 35
Как это соотносится с известными системами 36
Сколько живёт запись 36
Как из строчки в дневнике получилась эта книга 37
Путь одной мысли 37
Inbox: перестаньте решать всё в момент появления 38
Что такое входящие 38
Три двери вместо одной заметки дня 38
Мимолётные заметки 39
Сколько живёт входящее 39
Правило разбора 39
Схема триажа 40
Как выглядит хороший разбор 40
Частые возражения 41
Что дальше 41
Проекты, сферы, знания, жизнь, архив 42
Пять категорий 42
Конец ознакомительного фрагмента. 43

Василий Заметкин
Цифровой мозг

Выходные данные

Независимость издания. Эта книга — независимый образовательный материал. Она не аффилирована с Obsidian и не является официальной документацией продукта. Obsidian, а также названия других упомянутых программ и сервисов принадлежат их правообладателям и используются только для обозначения этих продуктов.

Версия и актуальность. Все инструкции проверены на Obsidian 1.14.2 (сентябрь 2026) на настольной версии. Интерфейс, названия пунктов меню и набор возможностей меняются от версии к версии: если что-то выглядит иначе, сверьтесь с официальной справкой программы (адрес — в разделе «Источники»).

Ответственность. Материалы книги носят образовательный характер. Автор не несёт ответственности за потерю данных: перед экспериментами со структурой хранилища сделайте резервную копию (см. главу о бэкапах).

Права на текст. Идеи и методики сторонних авторов — PARA и CODE Тиаго Форте, Zettelkasten в изложении Зонке Аренса, GTD Дэвида Аллена и другие — упоминаются со ссылкой на источник и не выдаются за разработку автора этой книги. Полный список источников — в конце книги.

Правообладателям. Если вы считаете, что какой-либо материал книги нарушает принадлежащие вам авторские, смежные или иные интеллектуальные права, укажите спорный материал и основание обращения. Автор рассмотрит обращение и при наличии оснований внесёт необходимые изменения или удалит соответствующий материал из последующих версий издания. Обращение можно направить через магазин, в котором приобретена книга.

От автора

Эта книга появилась не из желания написать ещё одно руководство по кнопкам Obsidian. Руководств хватает, а порядка в головах и папках больше не становится.

У любого человека, который хоть немного интересуется чем-то за пределами работы, каждый день появляется поток: мысли, идеи, ссылки, скриншоты, книги, которые хочется прочитать, планы, которые хочется однажды осуществить. Всё это выглядит ценным в момент появления. Поэтому мы сохраняем: в заметки телефона, в «Избранное», в закладки браузера, в переписку с самим собой.

А потом не возвращаемся. Не потому, что ленивы, а потому, что возвращаться некуда: сохранённое расползлось по десятку приложений, и ни одно из них не отвечает на вопрос «что у меня вообще есть по этой теме».

Мне кажется, нормальное желание — иметь одно место, которому можно доверять. Такое, где идея не теряется, проект не забывается, а прочитанное через год можно найти и использовать, а не вспоминать как смутное ощущение «я где-то это видел».

Эта книга — про то, как такое место построить. Сначала мы разберёмся, как вообще устроена работа с информацией: почему одни системы живут годами, а другие разваливаются за две недели. Потом соберём систему — на бумаге, в виде понятных правил. И только затем реализуем её в Obsidian: папка за папкой, шаблон за шаблоном, до рабочего результата.

Obsidian здесь — инструмент, а не цель. Если через год вы перейдёте в другую программу, принципы останутся с вами, а заметки — обычные текстовые файлы — переедут вместе с вами.

И последнее. Настраивать систему можно бесконечно. Это самая приятная форма прокрастинации: кажется, что работаешь, а на деле красишь стены в комнате, где никто не живёт. Поэтому книга устроена так, чтобы вы как можно раньше получили что-то работающее, а улучшали потом — по мере того, как жизнь покажет, чего не хватает.

Василий Заметкин

Как читать эту книгу

Первая часть книги не требует никакой программы: это принципы работы с информацией, которые останутся с вами при любом инструменте. Всё, что дальше, — практическая часть: готовая реализация этой системы в Obsidian, от структуры папок до баз данных и главного экрана. Obsidian расширяет книгу, но не является условием, без которого она бесполезна.

Книга рассчитана на два разных чтения, и оба нормальные.

Если вы пришли за порядком в цифровой жизни и про Obsidian только слышали — читайте подряд: в первой части мы разбираемся, как работает внимание, память и накопление информации, и собираем систему из принципов. Дальше вы увидите, как она превращается в конкретные файлы.

Если у вас уже установлен Obsidian и вопрос звучит как «и что теперь делать» — можно начать с части III, где мы собираем хранилище с нуля, а к первой части вернуться позже. Она короткая и объясняет, почему структура именно такая.

Что означают выделенные блоки

Идея

Короткая мысль, которую стоит унести с собой, даже если остальное забудется.

Сделайте сейчас

Действие на 2–10 минут. Не обязательное, но книга работает заметно лучше, если его делать.

Частая ошибка

Ошибка, которую совершают чаще всего именно в этом месте.

Почему это работает

Объяснение механизма: почему приём работает, а не просто «делайте так».

Готовый шаблон

Готовый шаблон или фрагмент кода, который можно скопировать в своё хранилище.

AI-промпт

Промпт для языковой модели с указанием, что подавать на вход и как проверять результат.

Дополнительно

Необязательная глубина: сторонние плагины, продвинутые приёмы. Без этого система работает.

Практические блоки

Технические инструкции устроены одинаково: *что мы делаем — зачем — шаги — готовый код — что должно получиться — частая ошибка — если хотите проще*. Последний пункт важен: у каждой конструкции в этой книге есть упрощённая версия, и она честно рабочая.

Практический комплект

Всё, что нужно для сборки, лежит в самой книге. В конце есть приложения: полная структура хранилища, все тринадцать шаблонов, словарь свойств, промпты, чек-листы и шпаргалка. Последнее из них — «Практический комплект» — собирает это в один маршрут и добавляет примеры заполненных заметок, по которым видно, как система выглядит в работе. Ничего скачивать не нужно.

Введение. Почему мы сохраняем всё и не используем почти ничего

Проверьте одну вещь прямо сейчас. Откройте на телефоне переписку с самим собой — «Избранное», «Сохранённые», заметку «разное», неважно, как это у вас называется. Пролистайте на тридцать сообщений назад.

Скорее всего, вы увидите примерно такое: ссылка без подписи, скриншот таблицы, чей-то совет про сон, адрес мастерской, название книги, полторы строчки про идею, которая тогда казалась отличной.

А теперь вопрос: что из этого вы использовали?

Обычно ответ — ничего. Не потому что мы ленивы или неорганизованны. А потому что в момент сохранения мы делаем ровно половину работы. Мы фиксируем, что это было важно. И не фиксируем, чем именно, для чего и что с этим делать дальше. Через месяц открыть такую запись — всё равно что найти чужой чек: понятно, что кто-то что-то купил, непонятно зачем.

Тысяча заметок, из которых ничего не найти

Я пишу эту книгу не с позиции человека, у которого всегда был порядок. Скорее наоборот.

До того как появилась система, у меня в заметках телефона лежало больше тысячи записей. Я честно пытался их организовать: заводил папки, внутри папок — подпапки. В одном месте оказались рабочие записи, цитаты, выписки из книг, фильмы и сериалы, отдельные списки рекомендаций от разных людей, расчёты, задачи, покупки, бытовые напоминания, идеи. Какое-то время там же жили пароли — пока не переехали в отдельный менеджер.

Показательнее всего были покупки. Утром я мог записать, что нужно что-то купить. Днём, уже перед магазином, искать ту запись было некогда — и появлялся новый список. А дома выяснялось, что утренний список существовал и в нём было самое важное.

Проблема была не в том, что я не записывал. Записывал. Проблема была в том, что запись нельзя было надёжно вернуть в тот момент, когда она нужна.

Параллельно та же информация жила в закладках браузера, в группах вкладок, в нескольких браузерах под разные контексты, в сообщениях самому себе, в сохранённых постах, в коротких видео, в скриншотах и в голове. Хуже всего были именно скриншоты: находишь такой через полгода при чистке фотоплёнки и понимаешь ровно одно — «в тот момент мне это почему-то показалось интересным». Что именно показалось, вспомнить уже невозможно.

Информации не слишком много. Её слишком много *непереваренной*

Про информационную перегрузку написано столько, что она сама стала формой перегрузки. Поэтому сразу к сути: проблема не в объёме входящего потока. Мы неплохо умеем не читать лишнее. Проблема в том, что у сохранённого нет следующего шага.

Посмотрите, где сейчас живут ваши материалы:

- заметки в телефоне — то, что записывалось на ходу;
- «Избранное» в мессенджере — то, что кто-то прислал;
- закладки браузера — то, что «прочитаю потом»;
- скриншоты в галерее — то, что было проще сфотографировать, чем записать;
- почта — то, что нужно будет найти по слову «договор»;
- рабочие документы — то, что лежит в пятнадцати папках;
- голова — всё остальное.

Ни одно из этих мест не отвечает на простой вопрос: «что у меня вообще есть по этой теме?» И когда тема наконец становится актуальной — вы начинаете собирать материал заново. С нуля. Из интернета. Хотя половина уже лежала у вас.

Идея

Мы редко страдаем от нехватки информации. Мы страдаем от невозможности достать нужное в тот момент, когда оно нужно.

Память — плохое место для обязательств

Отдельная история — то, что мы держим в голове. Не знания, а мелкие обязательства: позвонить, продлить, уточнить, не забыть спросить.

Голова с этим справляется плохо, но делает вид, что справляется. Она не теряет задачу молча — она напоминает о ней в самый неподходящий момент. В три часа ночи. Перед важным разговором. В душе́.

Это не поломка. Это нормальная работа механизма, который не рассчитан на календарь и список покупок одновременно. Именно поэтому любая приличная система продуктивности начинается с одного и того же действия: выгрузить всё наружу, чтобы голова перестала охранять список.

Что обещает эта книга — и чего не обещает

Начнём с того, чего здесь не будет.

Здесь не будет обещания, что за семь дней ваша жизнь изменится. Не будет утверждения, что правильные заметки сделают вас продуктивнее в три раза. И не будет красивых скриншотов графа из тысячи связанных заметок — потому что такой граф обычно означает, что человек больше настраивал систему, чем ей пользовался.

Что будет.

Вы соберёте одно место, которому можно доверять. Место, где входящее не теряется, проекты видны, знания находятся через год, а список «хочу посмотреть» не живёт в четырёх приложениях сразу. Сначала мы разберёмся с принципами — они работают в любой программе, хоть в бумажной тетради. Потом реализуем их в Obsidian: папка за папкой, шаблон за шаблоном.

К концу книги у вас будет рабочее хранилище, а не понимание того, как оно могло бы выглядеть.

Кому это нужно

У книги два типичных читателя, и они пришли с разных сторон.

Первый хочет навести порядок в цифровой жизни. Он может вообще не знать, что такое Obsidian. У него много сохранённого и ноль системы. Его вопрос: «как перестать терять важное?»

Второй уже установил Obsidian, посмотрел пару видео, создал хранилище — и застрял. Пустое окно, мигающий курсор, тридцать плагинов в списке рекомендованных. Его вопрос: «и что теперь делать?»

Первому нужна логика. Второму — готовые решения. В этой книге есть и то и другое, и они идут по порядку: сначала объяснение, потом конкретный файл, который можно скопировать.

Отдельно скажу про тех, кто «не технарь». Obsidian выглядит как программа для программистов ровно первые двадцать минут. Всё, что мы будем делать, — это создавать обычные текстовые файлы и расставлять между ними ссылки. Никакого кода писать не придётся: те несколько блоков, которые выглядят как код, можно просто скопировать.

Алексей

Чтобы разговор не был абстрактным, через всю книгу пройдёт один человек.

Алексей, 29 лет, аналитик. Читает нон-фикшн, иногда до конца. Ведёт два рабочих проекта и один домашний. Собирается осенью в Тбилиси. Сохраняет места, куда хочет сходить, в четырёх разных местах и ни одного не может найти в нужный момент. Думает про свой проект уже второй год.

Алексей — вымышленный, но узнаваемый. На его примере мы посмотрим, как мысль проходит путь от случайной записи до результата: попала во входящие, превратилась в проект, обросла заметками, оказалась в базе, вышла на главный экран и закончилась чем-то, что можно показать.

Узнаёте ли вы себя?

Небольшая проверка. Честно отметьте то, что про вас.

1. У меня больше двухсот открытых вкладок или закладок, к которым я не возвращался.
2. Я сохраняю ссылки «на потом» и почти никогда не открываю их потом.
3. Я читал книгу, которая мне понравилась, и сейчас не могу назвать три идеи оттуда.
4. Я искал свою же заметку и не нашёл — проще было загуглить заново.
5. Я держу в голове несколько мелких дел, чтобы не забыть.
6. У меня есть проект, о котором я думаю больше года и ничего не начал.
7. Мои материалы лежат минимум в четырёх разных приложениях.
8. Я не смогу за минуту сказать, какие проекты у меня сейчас активны.
9. Я делаю скриншоты вместо записей и потом их не разбираю.
10. Я уже пробовал «навести порядок» и бросил через пару недель.

Три-четыре отметки — вам хватит первой части и минимальной версии системы из главы 33. Семь и больше — читайте подряд, вы целевая аудитория этой книги.

Попробуйте сейчас

Не пролистывайте этот тест. Отметьте пункты вслух или в заметках телефона — прямо сейчас, за минуту. Дальше будет проще: вы уже будете знать, какую именно проблему решаете.

И последнее перед стартом. Эту книгу можно читать с телефона в метро, но хотя бы один вечер стоит провести за компьютером с открытым Obsidian. Система, собранная руками, отличается от системы, о которой прочитали, примерно так же, как поездка отличается от карты.

Система, а не приложение

Принципы, которые работают в любой программе — и даже в бумажной тетради. Здесь мы собираем систему на уровне решений, ещё не открывая Obsidian.

Ваш мозг не обязан быть жёстким диском

Представьте, что вам позвонили и сказали: «Завтра в одиннадцать, не забудьте документы». Вы отвечаете: «Конечно», кладёте трубку и возвращаетесь к работе.

С этой секунды часть вас занята охраной. Не самой задачей — до неё ещё сутки, — а тем, чтобы про неё не забыть. Вы будете вспоминать про эти документы вечером, перед сном, утром в душе́ и ещё раз в метро. Работу это не продвинет ни на шаг, но внимание съест.

А теперь представьте, что вы записали: «Завтра 11:00, встреча, взять документы» — в календарь, куда точно посмотрите. Напряжение исчезает почти мгновенно. Задача никуда не делась, но охранять её больше не нужно: это делает внешняя запись.

Вот вся идея этой главы, дальше будут только подробности.

Почему в голове помещается так мало

У нас отличная долговременная память. Вы наверняка помните запах бабушкиной квартиры, сюжет фильма, который смотрели десять лет назад, и стыдную сцену из школы, которую предпочли бы забыть.

А вот с оперативной памятью — с тем, что мы удерживаем прямо сейчас, — дела обстоят скромнее. Исследователи спорят о точной цифре, но сходятся, что одновременно человек удерживает в активном состоянии очень немного элементов: в обзоре Нельсона Коуэна речь идёт примерно о четырёх[1]. Не сорока. Четырёх.

Поэтому «держать в голове» пять проектов, восемь мелких дел и адрес мастерской — это не сила воли, а перегрузка. Каждое новое «не забыть» вытесняет что-то предыдущее. Знакомое ощущение, когда заходишь в комнату и не помнишь, зачем, — из этой оперы.

Почему это работает

Когда обязательство записано в место, которому вы доверяете, голова перестаёт тратить ресурс на его удержание. Освобождается не память — освобождается внимание. Именно поэтому после получаса «выгрузки» списка дел на бумагу становится легче, хотя сами дела никуда не делись.

Выгружать наружу — нормально

Мы постоянно перекладываем умственную работу на внешний мир, и это не признак слабости. Мы считаем на калькуляторе, ставим будильник, поворачиваем карту, чтобы она совпала с направлением движения, загибаем пальцы. Исследователи называют это когнитивной разгрузкой и изучают как обычную, повсеместную стратегию, а не как дефект[2].

Интересно другое: чем надёжнее внешнее хранилище, тем охотнее мы им пользуемся. И наоборот — если вы однажды записали важное в блокнот, а потом не нашли эту запись, вы больше не будете доверять блокноту. Вы начнёте дублировать: и записать, и запомнить. Худший из миров.

Отсюда первое требование к системе, которую мы будем строить: она должна быть достаточно надёжной, чтобы в неё поверили. Не идеальной. Надёжной.

Есть и обратная сторона, о которой честно стоит сказать. Когда мы знаем, что информация останется доступной, мы охотнее запоминаем, *где* она лежит, чем её саму. В известном исследовании Спэрроу, Лю и Вегнера участники хуже воспроизводили факты, если верили,

---

[1] Cowan, N. (2001). The Magical Number 4 in Short-Term Memory. *Behavioral and Brain Sciences*, 24(1), 87–114.

[2] Risko, E. F., & Gilbert, S. J. (2016). Cognitive Offloading. *Trends in Cognitive Sciences*, 20(9), 676–688.

что запись сохранится в компьютере[3]. Правда, позже к этой работе возникли вопросы по воспроизводимости, поэтому не будем делать из неё драму. Практический вывод всё равно полезный: внешняя система хорошо хранит детали, но не думает за вас. Если вы просто складываете ссылки, вы получите склад ссылок, а не знания. К этому мы вернёмся в главе 3.

Система, которой доверяют

В подходе GTD Дэвида Аллена есть точное выражение — *trusted system*, система, которой доверяешь. Смысл в том, что разгрузка головы работает только при одном условии: вы уверены, что записанное не потеряется и попадётся вам на глаза в нужный момент.

Проверить доверие легко. Ответьте на три вопроса:

1. Если прямо сейчас вам придёт важная мысль — вы знаете, куда её записать? Одно место, без раздумий?

2. Если вы запишете её туда — вы точно увидите её снова? Когда?

3. Если через полгода понадобится что-то по этой теме — вы найдёте запись?

Три «да» — у вас уже есть система, и эта книга поможет её усилить. Хотя бы одно «нет» — как раз этим мы и займёмся. Обычно проваливается второй вопрос: записать-то люди умеют, а вот возвращаться к записанному некуда.

Частая ошибка

Самая частая ошибка на старте — завести много мест для записи: одно для идей, другое для рабочего, третье для ссылок. Через неделю вы не помните, куда положили. Точка входа должна быть одна.

Хранение — это не мышление

Ещё одно различение, без которого дальше будет каша.

Хранить — значит положить так, чтобы потом достать. С этим справляется любая папка.

Думать — значит соединять: сравнивать, находить противоречия, доставать старое в новом контексте, формулировать своё. Для этого нужно, чтобы записи стояли не по одной в разных углах, а рядом и были связаны.

Большинство «систем заметок» — это про хранение. Они устроены как шкаф: аккуратные полки, подписанные коробки. В шкафу удобно, пока вы точно знаете, что ищете. Но шкаф никогда не скажет вам: «слушай, а вот эта мысль из мартовской книги хорошо ложится на твою сентябрьскую задачу».

Наша цель — не самый аккуратный шкаф. Наша цель — рабочий стол, на котором нужное оказывается рядом само.

Идея

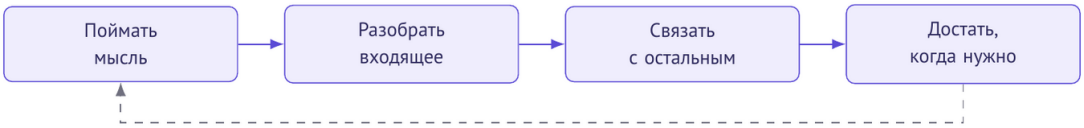
Хорошая система не та, в которой всё разложено. Хорошая система — та, в которой нужное само попадается вам на глаза в правильный момент.

Что мы будем строить

Дальше в книге всё будет крутиться вокруг четырёх функций. Держите их в уме, когда будете видеть очередную папку или шаблон.

---

[3] Sparrow, B., Liu, J., & Wegner, D. M. (2011). Google Effects on Memory. *Science*, 333(6043), 776–778. Результат обсуждался и воспроизводился неоднозначно.



Четыре функции второго мозга. Любой элемент системы должен обслуживать хотя бы одну из них — иначе он лишний.

Если папка, свойство или плагин не помогает ни поймать, ни разобрать, ни связать, ни достать — этот элемент не нужен. Это правило сэкономит вам десятки часов настройки.

Попробуйте сейчас

Пять минут, таймер. Выпишите в любое место всё, что «надо не забыть»: дела, звонки, мысли, покупки, недоделанное. Не сортируйте, не оценивайте — просто выгружайте, пока мысли не закончатся.

Большинство людей на этом упражнении выписывают от двадцати до сорока пунктов и заметно удивляются. Этот список пригодится нам в главе 6, когда мы будем разбирать входящие. Не выбрасывайте его.

Что дальше

Мы договорились: голова — плохое хранилище и отличный процессор. Значит, хранение нужно вынести наружу, а голову оставить для того, что она делает хорошо.

Место, куда мы это выносим, обычно называют вторым мозгом. Термин красивый и слегка обманчивый — в следующей главе разберёмся, что за ним стоит и почему из двух людей с одинаковым набором заметок один получает пользу, а другой — только чувство вины.

Что такое второй мозг и PKM

У термина «второй мозг» плохая репутация, и заслуженно. Он звучит как обещание суперспособности, которую продают на вебинаре за девять тысяч рублей. Поэтому договоримся сразу, что мы имеем в виду.

Второй мозг — это внешняя система, в которой живёт то, что не должно жить в голове: детали, обязательства, чужие идеи, ваши собственные мысли и материалы, которые понадобятся позже. Ничего мистического. Записная книжка Леонардо, картотека исследователя, папка «Проекты» на рабочем столе — всё это второй мозг разной степени удобства.

Меня самого этот термин долго отталкивал. Он звучал либо как очередная концепция из индустрии продуктивности, либо, наоборот, как фантастика: цифровая копия человека, искусственный интеллект, который однажды станет твоим вторым мозгом. Ни то ни другое к делу не относится.

Когда система у меня наконец появилась, понимание стало гораздо приземлённее. Второй мозг — это не второй человек и не замена вашему мозгу. Это внешний склад памяти.

Представьте, что у мозга есть свой склад, и он не бесконечный. Часть товара можно хранить у себя, а часть — отдать на внешний склад, как продавец отдаёт товары на склад маркетплейса. Важно не держать каждую коробку под рукой. Важно знать, что она существует, где лежит и как быстро её вернуть.

Английская аббревиатура PKM — personal knowledge management, личное управление знаниями — описывает ту же область: как один человек собирает, хранит, связывает и использует информацию, которая важна лично ему. Не корпоративная база знаний, не библиотека, а ваш личный слой.

Слово «управление» тут немного сбивает с толку. Речь не про менеджерские таблицы, а про очень бытовую вещь: сделать так, чтобы вы могли пользоваться тем, что уже знаете и уже нашли.

Идея не новая

Люди вели личные сборники задолго до появления компьютеров. В Европе XVI–XVIII веков существовала традиция *commonplace book* — тетради, куда переписывали цитаты, наблюдения, рецепты, полезные факты. Это был не дневник, а рабочий инструмент: человек собирал у себя то, к чему собирался возвращаться.

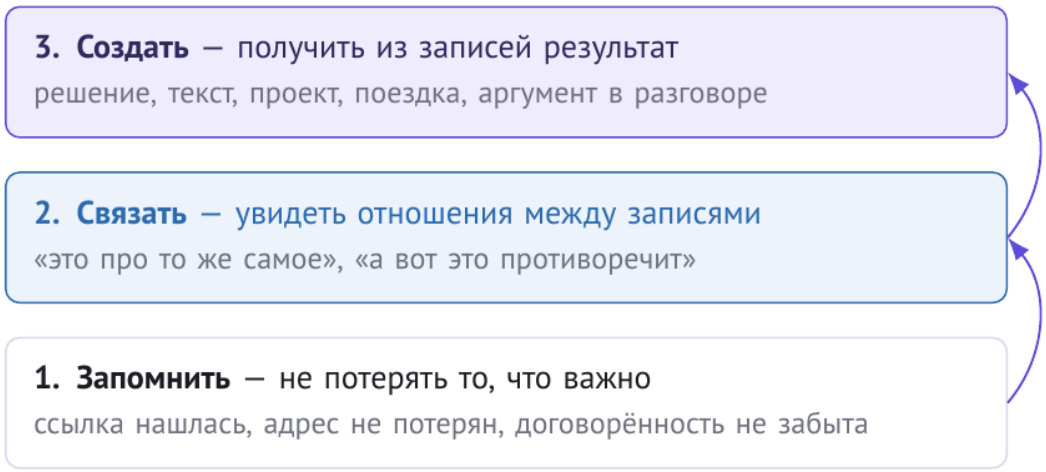
Отсюда полезный вывод: ценность такой тетради никогда не была в количестве страниц. Она была в том, что владелец её перечитывал, дополнял и использовал. С цифровыми заметками ровно та же история, только собирать стало настолько легко, что объём сам по себе перестал что-либо значить.

Идея

Сохранить стало бесплатно. Поэтому теперь ценность создаётся не сохранением, а всем, что происходит после него.

Три уровня ценности

Из двух людей с одинаковой коллекцией заметок один получает пользу, а другой — чувство вины. Разница обычно в том, на каком уровне остановился человек.



Три уровня ценности второго мозга. Первый доступен сразу, третий требует привычки.

Запомнить. Самый базовый уровень, и он уже полезен. Вы сохранили адрес мастерской — и нашли его через полгода. Ничего интеллектуального, просто работает.

Связать. Здесь начинается то, ради чего вообще стоит городить систему. Заметка про отток клиентов оказывается рядом с заметкой про когорты, а та — рядом с мыслью из книги, которую вы читали весной. Ни одна из них по отдельности не стоила многого. Вместе они дают готовый ход мысли.

Создать. Уровень, на котором система окупается. Вы садитесь писать отчёт, готовить встречу или планировать поездку — и обнаруживаете, что половина материала уже есть, причём в переваренном виде.

Большинство людей застревают между первым и вторым уровнем: сохранённого много, связей нет. Именно поэтому в этой книге так много внимания уделяется не сбору, а обработке.

Чем второй мозг не является

Четыре распространённых заблуждения, каждое из которых стоило кому-то месяца жизни.

Это не архив всего. Цель — не сохранить максимум, а сохранить то, что понадобится. Разница огромная: архив требует полноты, рабочая система требует полезности.

Это не замена мышлению. Заметки не думают за вас. Они убирают из головы то, что мешает думать, и подсовывают нужное в нужный момент — не больше.

Здесь же проходит граница, которую я считаю важной. Если выгружать наружу вообще всё и перестать что-либо понимать и помнить самому, получится не расширенное мышление, а зависимость от заметок. Наружу стоит отдавать контекст, факты, которые важно восстановить, результаты изучения, связи, идеи и проекты. Основные концепции своей области человек всё-таки должен держать в голове — иначе он не сможет даже сформулировать, что именно ему искать в собственном хранилище.

Это не про красоту. Граф связей, цветные теги, аккуратные иконки — приятно, но к результату отношения не имеет. Самые полезные системы, которые я видел, выглядят скучно.

Это не проект на выходные. Систему нельзя «настроить и забыть», как нельзя один раз убраться в квартире. Но и обслуживания она требует немного: пятнадцать минут в неделю при нормальной настройке.

Частая ошибка

Симптом, по которому легко узнать заблудившегося: человек полгода «настраивает второй мозг», меняет плагины, переносит заметки между приложениями — и ни разу не исполь-

зовал его для реальной задачи. Настройка приятнее работы, потому что даёт ощущение прогресса без риска.

Что должно быть внутри

Независимо от программы, у любой работающей системы есть пять частей.

1. **Точка входа** — одно место, куда попадает всё новое, не разбираясь.
2. **Правило разбора** — понятный способ решить, что делать с каждой записью.
3. **Структура** — небольшое число категорий, которые отражают вашу жизнь, а не теорию.
4. **Связи** — возможность соединять записи между собой и находить их с разных сторон.
5. **Ритм возвращения** — регулярный момент, когда вы смотрите на систему целиком.

Если хоть одного пункта нет, система разваливается предсказуемым образом. Нет точки входа — записи расползаются. Нет правила разбора — входящее копится. Нет структуры — всё лежит в одной куче. Нет связей — получается склад. Нет ритма — вы просто перестаёте туда заходить.

Мы будем строить все пять частей по очереди. Вот что чему соответствует:

Часть системы	Где в книге	Что это будет в Obsidian
Точка входа	главы 6, 16	папка `00 Inbox`
Правило разбора	глава 17	дерево решений и сессия на 10 минут
Структура	главы 7, 13	восемь папок верхнего уровня
Связи	главы 8, 22	ссылки, обратные ссылки, карты содержания
Ритм возвращения	глава 31	недельный и месячный обзор

При чём тут Алексей

У Алексея сейчас есть первый уровень, и то частично. Ссылки он сохраняет, но в трёх местах. Адрес мастерской нашёл, а материалы по прошлому похожему проекту — нет, хотя точно что-то читал.

К концу книги у него будет так: мысль про отток клиентов попадает во входящие за пять секунд, при разборе превращается в заметку с понятной формулировкой, эта заметка связывается с прошлогодней про когорты, обе всплывают, когда он начинает проект по дашборду, а после завершения проекта то, что он понял, остаётся в системе и переживёт смену работы.

Это не фантазия — это обычная работа системы, у которой есть все пять частей.

Попробуйте сейчас

Ответьте себе на один вопрос: на каком уровне вы сейчас? Сохраняете (1), связываете (2) или используете (3)? Ответ «ни на каком» тоже нормальный — значит, начинаем с чистого листа, и это даже проще.

Почему прочитать — не значит выучить

Вспомните последнюю нон-фикшн книгу, которая вам понравилась. Не пролистанную — именно понравившуюся, ту, после которой вы кому-то её советовали.

А теперь, не подглядывая, назовите три идеи оттуда. Своими словами, без формулировок автора.

Обычно получается одна. Иногда полторы. При этом ощущение от книги осталось отличное: «сильная вещь, многое взял». Между этим ощущением и тем, что реально осталось в голове, — пропасть, и именно в ней теряется большая часть нашего чтения.

Эта глава — про то, откуда берётся пропасть и что с ней делать. Без курса когнитивной психологии: нам нужны пять идей, каждая из которых сразу превращается в практику.

Пять состояний информации

Между «увидел» и «умею» лежит несколько ступеней, и мы регулярно путаем соседние.

Потребление. Информация прошла перед глазами. Видео посмотрено, статья пролистана. Это ноль, хотя ощущается как работа.

Узнавание. Вы видите текст и думаете: «да, знакомо». Самое коварное состояние: узнавание почти неотличимо от знания изнутри, но при этом бесполезно. Именно поэтому перечитывание конспекта перед экзаменом даёт такое приятное и такое обманчивое чувство готовности.

Понимание. Вы можете объяснить идею другому человеку своими словами, не заглядывая в источник.

Воспроизведение. Вы можете достать эту идею из памяти без подсказки, когда её никто не упоминал.

Применение. Вы использовали идею в реальной ситуации и получили результат — или не получили и знаете почему.

Большая часть того, что мы «изучаем», застревает на второй ступени. Причём чем удобнее формат, тем вероятнее застревание: хорошо смонтированное видео легко смотрится и плохо запоминается.

Идея

Наша задача — не собрать больше информации. Наша задача — сделать так, чтобы информация меняла то, что мы знаем и умеем.

У меня это ощущение хорошо отработано на экзаменах. Готовишься, читаешь, всё понимаешь и думаешь: записывать нечего, я и так разобрался. А потом на самом экзамене ловишь себя на вопросе: «это решалось вот так — или всё-таки иначе?» Понимание было настоящим, но за две недели оно потеряло опору.

Из этого у меня выросло практическое правило: даже когда кажется, что всё понял, полезно сохранить скелет знания — название концепции, ключевую формулу, отличие одного подхода от другого, несколько опорных пунктов. По такому скелету остальное восстанавливается за минуты. Без него приходится учить заново.

Идея первая: вспоминать полезнее, чем перечитывать

Самый надёжно подтверждённый приём в этой области звучит скучно: попытка вспомнить материал без подсказки укрепляет память сильнее, чем повторное чтение того же текста.

В классической работе Рёдигера и Карпике студенты читали научно-популярные тексты, а потом одна группа перечитывала их, другая — пыталась воспроизвести по памяти. Через

пять минут перечитывание выглядело лучше. Через неделю картина переворачивалась: те, кто вспоминал, помнили заметно больше[4].

Позже Карпике и Блант сравнили практику вспоминания с развёрнутым изучением и построением схем — и практика вспоминания снова выиграла на отложенной проверке[5].

Что это значит для заметок. Когда вы дочитали главу и хотите её «зафиксировать», есть два пути. Первый — открыть текст и аккуратно переписать оттуда ключевые места. Второй — закрыть книгу и написать по памяти, что вы поняли, а потом сверить с источником и дописать пропущенное.

Второй путь неприятнее и медленнее. Именно поэтому он работает.

Почему это работает

Каждый раз, когда вы достаёте воспоминание, оно записывается заново — и достать его в следующий раз становится легче. Перечитывание такой работы не требует: текст перед глазами, доставать нечего.

Идея вторая: интервалы

Второй приём — распределять повторение во времени. Три подхода по десять минут с паузами в несколько дней дают лучший результат, чем полчаса подряд. Это подтверждено на большом количестве экспериментов; обзор Сепеды с коллегами свёл их вместе[6].

Отсюда не следует, что вам нужны карточки и приложение для повторений. Для большинства читателей этой книги хватит другого: регулярного возвращения к собственным заметкам. Недельный обзор из главы 31 — это и есть распределённое повторение, просто без слова «повторение» и без расписания.

Идея третья: сформулированное своими словами держится лучше

Эффект генерации: то, что человек произвёл сам, запоминается лучше, чем то, что он просто прочитал. В исходных экспериментах Сламецки и Графа разница проявлялась даже на уровне отдельных слов — участники лучше помнили те, которые достраивали сами, а не читали готовыми[7].

Практический перевод: скопированная цитата почти ничего не даёт, кроме иллюзии работы. Та же мысль, пересказанная вашими словами, — даёт.

Есть и побочная выгода. Когда вы пытаетесь сформулировать идею сами, вы моментально обнаруживаете, что не поняли её до конца. Цитата это скрывает, пересказ — обнажает.

Частая ошибка

Самая частая ошибка в заметках по книгам — выделить сорок фрагментов и выгрузить их в файл. Через год вы откроете этот файл и увидите чужой текст без контекста. Лучше три абзаца своими словами, чем сорок цитат.

Обратный пример у меня тоже есть. Я проходил курс по системной аналитике. Пока шли занятия, материал был понятен, и записи казались достаточными. Когда я вернулся к ним спу-

---

[4] Roediger, H. L., & Karpicke, J. D. (2006). Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention. *Psychological Science*, 17(3), 249–255.

[5] Karpicke, J. D., & Blunt, J. R. (2011). Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. *Science*, 331(6018), 772–775.

[6] Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed Practice in Verbal Recall Tasks. *Psychological Bulletin*, 132(3), 354–380.

[7] Slamecka, N. J., & Graf, P. (1978). The Generation Effect: Delineation of a Phenomenon. *Journal of Experimental Psychology: Human Learning and Memory*, 4(6), 592–604.

стя время, выяснилось, что по ним трудно восстановить смысл: остались формулировки без контекста, и часть изученного фактически потерялась. Дело не в том, что курс был плохим, — дело в том, что мои записи не сохранили того, что требовалось для возвращения.

Похожая история случается и вне учёбы. Возвращаешься к игре, которую забросил два месяца назад, и не помнишь ни сюжета, ни персонажей, ни на чём остановился, — и начинаешь заново не потому, что хочется, а потому, что потерян контекст. Пять строк, записанных перед паузой, экономят вечер.

Идея четвёртая: объяснять — значит понимать

Развёрнутое объяснение (в литературе — elaboration) означает простую вещь: связать новое с тем, что вы уже знаете, и объяснить, почему это так работает. Не «отток считается по когортам», а «отток считается по когортам, потому что иначе новые клиенты маскируют уход старых — как в классе, куда каждый год приходят новички и средний возраст не меняется, хотя все взрослеют».

Такое объяснение занимает на две минуты больше и держится на порядок дольше, потому что цепляется сразу за несколько уже существующих узлов.

Отсюда формат заметки, который мы будем использовать в главе 9: не только «что», но и «почему это важно» и «с чем связано».

Идея пятая: перемешивание

Последний приём пригодится не всем, но знать о нём стоит. Когда задачи разных типов перемешаны, а не идут блоками, обучение идёт медленнее, зато результат при отложенной проверке выше. Рорер и Тейлор показали это на школьной математике: группа, решавшая перемешанные задачи, на занятиях выглядела хуже, а на отложенном тесте — заметно лучше[8].

Причина в том, что при перемешивании вам приходится каждый раз выбирать подход, а не применять один и тот же по инерции. В жизни это выглядит так: изучать формулы курса подряд — легко и бесполезно; применять их к разным своим задачам — трудно и работает.

Что со всем этим делать на практике

Соберём пять идей в один рабочий приём, который дальше в книге станет основой работы с источниками.

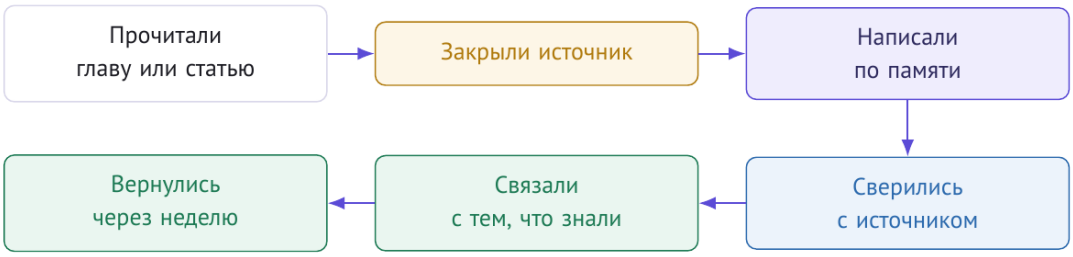


Схема работы с источником. Ключевой шаг — второй: закрыть источник до того, как начали писать.

Шаг «закрыли источник» выглядит мелочью, но именно он превращает конспектирование в обучение. Всё остальное — обвязка.

[8] Rohrer, D., & Taylor, K. (2007). The Shuffling of Mathematics Problems Improves Learning. *Instructional Science*, 35(6), 481–498.

Попробуйте сейчас

Возьмите статью, которую прочитали на этой неделе. Не открывая её, напишите в заметках телефона пять предложений: о чём она и что вы из неё берёте. Потом откройте и сверьтесь.

Два наблюдения, которые обычно случаются: вы вспомнили меньше, чем ожидали, и вспомнили не то, что автор считал главным. Оба наблюдения полезны.

Три оговорки, чтобы не уехать в противоположную крайность. Не превращайте в учёбу всё чтение: роман и нон-фикшн для удовольствия обрабатывать не нужно. Не обрабатывайте всё подряд — из десяти статей заметки заслуживают одна-две. И не гонитесь за системой карточек: для большинства задач достаточно того, что вы регулярно возвращаетесь к своим заметкам по работе.

В следующей главе мы посмотрим на это шире: как перейти от режима «потребляю информацию» к режиму «делаю из неё что-то своё» — и почему для этого нужен не столько метод чтения, сколько повод.

От потребителя информации к создателю знаний

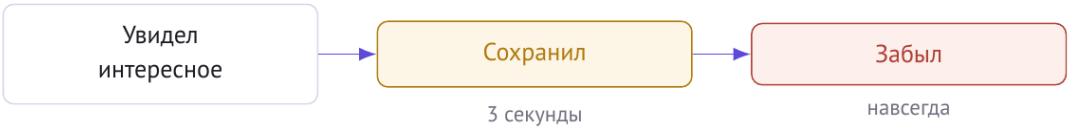
У Алексея в «Избранном» лежит сто сорок сообщений. Он пролистал их однажды в отпуске и обнаружил забавное: примерно треть — статьи про продуктивность, которые он сохранил, чтобы стать продуктивнее, и не прочитал.

Это не ирония судьбы, а механика. Сохранение приносит маленькое удовольствие прямо сейчас: галочка поставлена, ощущение продвижения получено. Чтение и обдумывание требуют усилия и наград не обещают. Поэтому при прочих равных мы всегда будем сохранять больше, чем обрабатывать.

Значит, полагаться на «буду дисциплинированнее» бессмысленно. Нужно поменять сам цикл.

Цикл, который не работает

Вот как выглядит обычный маршрут информации у человека без системы:



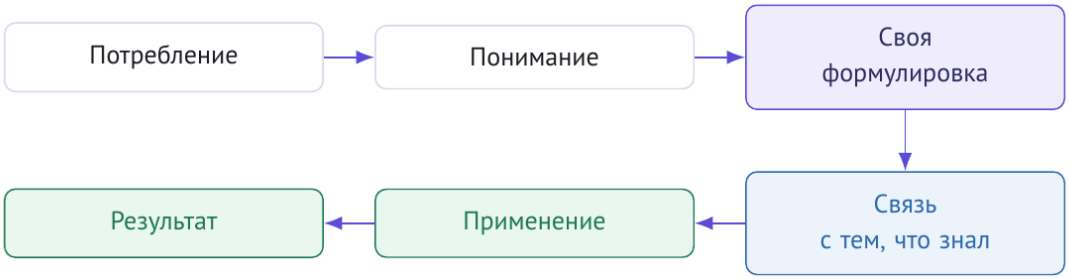
Цикл сохранения. Единственный результат — растущее чувство, что «где-то это было».

У меня этот цикл дольше всего жил в виде скриншотов. Увидел что-то интересное — сфотографировал экран. Дальше скриншот попадал в фотоплёнку, и следующая встреча с ним происходила через полгода, во время очередной чистки галереи. В этот момент от записи оставалось ровно одно сообщение: в тот день мне это почему-то показалось интересным. Ни темы, ни причины, ни того, что я собирался с этим делать.

Проблема даже не в том, что мы забываем. Проблема в том, что сохранённое накапливает долг. Каждая непрочитанная сохранённая статья — это маленькое невыполненное обещание себе. Сто сорок таких обещаний создают фоновую тяжесть, из-за которой люди в какой-то момент просто удаляют всё и начинают заново.

Цикл, который работает

Рабочий маршрут длиннее и требует усилия ровно в одном месте — но зато заканчивается результатом, а не долгом.



Цикл знания. Между «понял» и «своя формулировка» проходит граница между потребителем и создателем.

Моя нынешняя цель — чтобы поток шёл по этому маршруту: увидел, зафиксировал, вернулся, решил, нужно ли это вообще, и при необходимости развил и связал с проектом или знанием. Не могу сказать, что через него проходит вся информация, — но это направление, и разница с прежним «сохранил и забыл» заметна уже сейчас.

Обратите внимание, где именно усилие: переход от «понял» к «сформулировал». Всё, что до него, приятно и бесплатно. Всё, что после, уже катится само, потому что своя формулировка легко связывается, легко применяется и легко превращается в результат.

Идея

Пятьдесят сохранённых цитат почти всегда менее полезны, чем три собственные мысли, записанные своими словами. Цитаты принадлежат автору. Мысли — вам.

Фильтр на входе

Первый способ разгрузить систему — меньше в неё класть. Но не по принципу «сохраняй меньше интересного», а по другому вопросу.

Обычный вопрос при сохранении: **«Это интересно?»** На него почти всегда ответ «да» — иначе бы вы не смотрели.

Рабочий вопрос: **«Зачем это может понадобиться будущему мне?»**

Разница огромная. Первый вопрос про текущую эмоцию, второй — про будущее использование. Попробуйте прогнать через него последние сохранения, и половина отвалится сама.

Если ответ на второй вопрос есть — сохраняйте, но вместе с ответом. Одна строчка «зачем» превращает мёртвую ссылку в живую заметку:

```
ссылка на статью про удержание
Зачем: хочу сравнить их определение удержания с нашим,
когда будем спорить о метриках на следующем созвоне.
```

Через месяц вы откроете эту запись и сразу поймёте, что с ней делать. Без строчки «зачем» — скорее всего, не поймёте.

Попробуйте сейчас

Откройте последние тридцать сохранений (заметки, «Избранное», закладки — что ближе). По каждому ответьте одним словом: «зачем?».

Считайте, на скольких вы застряли. Обычно застревают на двух третях. Это не повод расстраиваться — это калибровка входного фильтра.

Информационная диета без фанатизма

Про информационную гигиену написано много воинственного: удалить соцсети, отписаться от всего, читать только книги. Обычно это заканчивается через неделю, потому что борется с симптомом.

Практичнее принять три вещи как данность.

Вы не прочитаете всё сохранённое. И не должны. Watch Later — это не очередь, а свалка, и относиться к ней надо соответственно: раз в месяц удалять не глядя всё старше трёх месяцев. Если что-то было действительно нужным, оно вернётся.

FOMO не лечится потреблением. Страх упустить важное растёт от количества входящего, а не уменьшается. Единственное, что реально помогает, — наличие собственных проектов: когда вы чем-то заняты, чужая повестка автоматически теряет вес.

Сохранение ради ощущения продуктивности — нормальный человеческий рефлекс. С ним не надо бороться, его надо обезвредить: пусть всё падает в одно место, откуда

потом легко удалить пачкой. Именно для этого нужны входящие, о которых пойдёт речь в главе 6.

Частая ошибка

Не устраивайте «большую чистку» всего сохранённого за прошлые годы. Это занимает выходные, выматывает и почти ничего не даёт. Начните с потока, который идёт сейчас, а старые залежи разберите по требованию — когда что-то из них понадобится.

Система самообразования

Теперь про то, ради чего многие вообще открывают такие книги: как учиться так, чтобы что-то оставалось.

Вот цикл, который работает у большинства людей, кого я видел за этим занятием:

```
Хочу разобраться в теме
→ формулирую вопрос или результат
→ выбираю два-три источника, а не двадцать
→ изучаю
→ фиксирую только то, что отвечает на вопрос
→ объясняю своими словами
→ связываю с тем, что уже знаю
→ применяю на реальной задаче
→ получаю результат, который можно показать
```

Ключевой — второй шаг. «Изучить SQL» не заканчивается никогда, потому что нет критерия завершения. «Переписать наш отчёт по удержанию так, чтобы он считался одним запросом» заканчивается ровно тогда, когда отчёт работает.

Сравните две формулировки одного и того же намерения:

Бесконечное	Конечное
Изучить фотографию	Снять серию из 15 кадров и выбрать 5 лучших
Изучить Python	Написать скрипт, который собирает мои расходы в таблицу
Читать про переговоры	Подготовиться к переговорам по аренде и провести их
Разобраться в дизайне	Переделать одну страницу и показать двум людям
Подтянуть английский	Провести рабочий созвон на английском без подготовки

Левая колонка порождает бесконечный список материалов. Правая — порождает проект, у которого есть конец. В части V мы увидим, что это ровно то же самое различие, что между сферой жизни и проектом.

Проектное обучение

Идея простая: учиться вокруг результата, а не вокруг темы.

Что это даёт на практике:

Фильтр появляется сам. Когда есть конкретная задача, становится очевидно, какие главы читать, а какие пропустить. Без задачи вы честно читаете всё подряд и устаёте на трети.

Появляется обратная связь. Вы пробуете — не получается — вы понимаете, чего именно не хватает. Это самый быстрый способ учиться из всех известных.

Остаётся артефакт. После «изучения фотографии» не остаётся ничего. После серии из пятнадцати кадров остаются кадры, а вместе с ними — понимание, что вы умеете и что нет.

Знание закрепляется применением. Мы уже говорили в главе 3: применение — последняя ступень, и достигается она только в деле.

Идея

Если вы не можете придумать результат, ради которого изучаете тему, скорее всего, вам не нужно её изучать. Нужно просто про неё почитать — и это тоже нормально, если не притворяться, что это обучение.

Как это будет выглядеть в системе

Забегая вперёд, вот путь, который мы построим в Obsidian, на примере Алексея:

1. Алексей решает разобраться с оконными функциями. Заводит проект «Разобраться с оконными функциями SQL» с результатом: три рабочих запроса, которые он написал сам.
2. Статьи и видео по теме попадают во входящие как заметки-источники, привязанные к проекту.
3. Из каждого источника он вытаскивает одну-две мысли своими словами: «оконная функция считает агрегат, не схлопывая строки».
4. Эти мысли связываются с тем, что уже лежит: с заметкой про когортный анализ, с рабочим проектом по оттоку.
5. Он применяет их к настоящему отчёту. Отчёт работает — проект закрыт.
6. Источники уходят в архив, три сформулированные мысли остаются в знаниях навсегда и переживут смену работы.

Заметьте, чего здесь нет: нет конспекта курса на сорок страниц. Нет папки «SQL» с восемнадцатью сохранёнными ссылками. Есть три заметки и один работающий отчёт.

Попробуйте сейчас

Выберите одну тему, которую вы «давно собираетесь изучить». Переформулируйте её в результат по образцу таблицы выше: что должно существовать, чтобы вы сказали «готово»?

Если результат придумать не получается — это ценный ответ. Возможно, тему стоит вычеркнуть и вернуться к ней, когда появится настоящий повод.

Дальше нам нужен общий язык: как вообще движется информация от появления до пользы. Об этом следующая глава — и она даст скелет, на который мы будем вешать все остальные главы книги.

Жизненный цикл информации

Любая запись в вашей системе — от адреса мастерской до идеи на миллион — проходит один и тот же путь. Если вы этот путь видите, вы понимаете, где именно у вас всё ломается. А ломается у большинства людей в одном и том же месте.

Шесть шагов



Схема 1. Жизненный цикл информации. Пунктирная стрелка — возврат: при обзоре часть записей снова попадает в обработку.

Capture — поймать. Зафиксировать мысль или материал до того, как он исчез. Требование одно: быстро. Если фиксация занимает больше десяти секунд, вы не будете ею пользоваться в те моменты, когда она нужнее всего.

Process — разобрать. Решить, что это такое: мусор, задача, материал для проекта, знание, объект личной базы. На этом шаге запись получает место и, если нужно, форму.

Connect — связать. Поставить ссылки на то, с чем запись связана. Один шаг, который отделяет систему от склада.

Use — использовать. Достать в нужный момент и применить: написать текст, принять решение, подготовить поездку, собрать отчёт.

Review — пересмотреть. Регулярно смотреть на систему целиком: что активно, что застряло, что уже неактуально.

Archive — убрать. Отправить в архив то, что закончилось. Не удалить, а убрать с глаз.

Где ломается

Теперь интересное. Почти все проблемы, с которыми люди берутся за такие книги, — это поломка на конкретном шаге.

Симптом	Где поломка
«Записываю в четыре разных места»	Capture: нет единой точки входа
«Записал и не нашёл»	Process: записи не разбираются
«Всё лежит, но я этим не пользуюсь»	Connect: нет связей, только хранение
«Каждый раз собираю материал заново»	Use: система не участвует в работе
«Система разъехалась за месяц»	Review: нет момента возвращения
«Не могу найти нужное среди старого»	Archive: завершённое не убирается

Обратите внимание: почти никто не жалуется на Capture. Записывать умеют все. Поэтому большая часть книги — про шаги со второго по шестой.

Частая ошибка

Типичная ловушка — усиливать тот шаг, который и так работает. Человек с проблемой на Process покупает третье приложение для быстрых заметок. Стало ещё хуже: теперь входящие в трёх местах.

Как это соотносится с известными системами

Наш цикл — не новая теория, а общий знаменатель нескольких хорошо известных подходов. Если вы что-то из них читали, вот карта соответствий.

Наш шаг	GTD (Дэвид Аллен)	CODE (Тиаго Форте)	Smart notes (Зонке Аренс)
Capture	Collect	Capture	Fleeting notes
Process	Process / Organize	Organize	Literature notes
Connect	—	—	Permanent notes, связи
Use	Engage	Express	Написание текста
Review	Weekly Review	—	Регулярная работа с картотекой
Archive	Someday / Reference	Archive в PARA	—

Видно, что подходы дополняют друг друга. GTD силён на входе и в обзорах: там придумана идея доверенной системы и привычка регулярно к ней возвращаться. CODE у Форте хорошо описывает работу с материалами и доведение их до результата. Аренс детально разбирает середину — как из чужого текста получается ваша собственная мысль.

Мы возьмём сильные стороны каждого, но структуру построим свою, потому что ни одна из трёх не рассчитана на человека, которому одновременно нужны рабочие проекты, список фильмов и заметки по книгам.

Дополнительно

Ни PARA, ни CODE, ни Zettelkasten не являются разработкой этой книги. PARA и CODE предложены Тиаго Форте, Zettelkasten восходит к практике Никласа Лумана и описан в современном виде Зонке Аренсом, GTD — метод Дэвида Аллена. В книге они используются как источники идей с указанием авторства; полные ссылки — в разделе «Источники».

Сколько живёт запись

Полезная мысль, которая экономит много места: у разных записей разный срок жизни, и относиться к ним надо по-разному.

Минуты и часы. «Купить батарейки», «перезвонить в три». Это не знание, это оперативная память. Такие записи должны исчезать из системы после выполнения — им место в задачах или в ежедневной заметке, а не в базе знаний.

Недели и месяцы. Материалы текущего проекта: выгрузки, черновики, ссылки по теме. Они нужны, пока идёт проект, и уходят в архив вместе с ним.

Годы. Ваши собственные сформулированные мысли, выводы, проверенные подходы. Это ядро системы, и его должно быть немного — десятки заметок, а не тысячи.

Справочное. Данные, которые не устаревают и нужны по требованию: список любимых мест, прочитанные книги, контакты, чек-листы. Живут в личных базах.

Идея

Большая часть того, что кажется знанием, на самом деле оперативная память проекта. Не тащите её в вечное хранилище: она умрёт вместе с проектом, и это нормально.

Как из строчки в дневнике получилась эта книга

Эта книга — сама по себе пример цикла, и я буду возвращаться к ней по ходу глав.

Начиналось всё предельно скромно. В обычный день среди прочих записей в ежедневной заметке появилась строчка: сделать электронную книгу. Ни плана, ни темы, ни оценки, стоит ли за это браться. Одна строка среди покупок, задач и ссылок.

Если бы на этом всё закончилось, вы бы сейчас это не читали. Но у строчки было продолжение — и оно начинается со следующего шага цикла, обработки. О том, что было дальше, — в главах 6, 19 и 23.

Путь одной мысли

Посмотрим на конкретный пример — как одна запись Алексея проходит весь цикл.

Capture. На созвоне Алексей слышит от коллеги: «мы считаем отток по-другому». Он за пять секунд пишет во входящие: «у продукта и у нас разные определения оттока — разобраться».

Process. Вечером при разборе входящих он понимает, что это не отдельная задача, а часть проекта про дашборд. Запись превращается в пункт журнала проекта, а сама формулировка проблемы — в заметку «Отток — это не одна метрика».

Connect. Он ставит из этой заметки ссылки на «Когортный анализ без боли» (там объясняется, почему определение влияет на расчёт) и на проект. Заодно видит в обратных ссылках, что полгода назад уже писал что-то похожее про удержание.

Use. Через неделю на встрече он открывает заметку и предлагает готовое определение. Спор занимает пятнадцать минут вместо часа.

Review. На недельном обзоре видит, что проект сдвинулся, а в следующем шаге появилось «согласовать определение с продактом» — вычёркивает.

Archive. Когда дашборд сдан, проект уезжает в архив вместе с рабочими выгрузками. Заметка «Отток — это не одна метрика» остаётся в знаниях. Через два года на новой работе она всплывёт снова.

Весь путь занял у Алексея минут двадцать суммарно, размазанных по трём неделям. Именно так это и должно работать: не «час на ведение системы», а по чуть-чуть внутри обычной работы.

Попробуйте сейчас

Возьмите любую запись из вашего списка «выгрузки головы» (упражнение из главы 1) и пройдите по шагам: что это, куда относится, с чем связано, когда понадобится, когда можно будет убрать.

Если на каком-то шаге вы зависли — скорее всего, именно там у вас и проседает система.

Следующие три главы разбирают шаги подробно: сначала точку входа, потом категории, в которые всё раскладывается, потом связи между ними.

Inbox: перестаньте решать всё в момент появления

Представьте почтовое отделение, где каждое письмо сортируют в момент, когда его принесли. Курьер стоит и ждёт, пока сотрудник решит, в какой ящик положить письмо, а если подходящего ящика нет — пока он смастерит новый. Очередь из курьеров растянется до соседней улицы.

Так работает большинство людей со своими мыслями. Возникла идея — и в ту же секунду нужно решить: это задача? проект? заметка? куда её? в какую папку? а может, завести новую? Пока идёт это решение, мысль успевает померкнуть, а вы — сбиться с того, чем занимались.

Отсюда простое правило, которое лежит в основе всех приличных систем: **фиксация и разбор — разные операции, и делать их надо в разное время.**

Что такое входящие

Входящие (Inbox) — это одно место, куда всё падает без раздумий. Никакой структуры, никаких правил, никакой сортировки. Единственное требование к записи: она должна там оказаться.

Три свойства правильных входящих:

Одни. Не «рабочие входящие» и «личные входящие». Одни. Как только их становится двое, появляется решение «куда положить» — то самое, от которого мы избавлялись.

Быстрые. Две секунды на открытие, десять на запись. Всё, что медленнее, не переживёт столкновения с реальной жизнью — в метро, на ходу, в середине разговора.

Временные. Входящие должны пустеть. Это перевалочный пункт, а не склад. Записи, которые живут там третий месяц, — сигнал, что разбор не случается.

Идея

Входящие — это разрешение не думать прямо сейчас. Вы записали и вернулись к делу, а решение приняли потом, когда у вас будет на это голова.

Три двери вместо одной заметки дня

До системы у меня роль входящих играла одна заметка на день. В ней подряд оказывались покупки, важная информация, задача, ещё один список покупок, идея, ссылка, что-то из просмотренного и снова задача. Запись существовала — структуры не было никакой.

Сейчас у меня три пути, и решение занимает секунду:

- быстрая мысль, идея или событие дня — в ежедневную заметку;
- запись, назначение которой пока непонятно, — во входящие;
- конкретное действие — в задачи.

Всё остальное — куда это в итоге ляжет, нужно ли вообще — решается позже. Кстати, идея этой книги пришла именно первым путём: она была одной строчкой в ежедневной заметке, среди прочих дел.

В первые дни после перехода у меня накопилось около семнадцати таких записей. Разбор занял один спокойный вечер: часть я связал с существующими заметками, часть разложил, часть удалил без сожаления. Это не статистика и не обещание — просто иллюстрация масштаба: обычно речь идёт о десятках записей, а не о сотнях.

Мимолётные заметки

У Зонке Аренса для таких записей есть точное название — *fleeting notes*, мимолётные заметки. Смысл в том, что они по определению одноразовые: их задача — дожить до момента разбора, и не более.

Из этого следуют два послабления, которые сильно облегчают жизнь.

Их можно писать плохо. Три слова без глагола, ссылка без объяснения, «спросить М. про ту штуку». Мимолётная заметка не обязана быть понятной через год — она обязана быть понятной сегодня вечером.

Их можно удалять. Большая часть входящих после разбора не превращается ни во что. Это норма, а не потеря.

Второе почему-то даётся людям тяжелее первого. Мы относимся к собственным записям как к ценности просто потому, что мы их сделали. Полезно помнить: ценность создаёт не запись, а то, что вы с ней сделали.

Сколько живёт входящее

У записи во входящих есть срок годности, и он короче, чем кажется.

• **День-два.** Всё помнится, разбор занимает секунды.

• **Неделя.** Половина записей требует усилия, чтобы вспомнить контекст.

• **Месяц.** «Посмотреть про ту штуку с графиками» — уже загадка.

• **Три месяца.** Практически мусор, что бы там ни было написано.

Отсюда практическое следствие: разбирать входящие нужно не «когда-нибудь», а достаточно часто, чтобы контекст ещё был жив. Мне удобнее разбирать записи прямо в течение дня, пока помню, зачем они появились; кому-то удобнее один спокойный заход в конце недели. Ритм индивидуален, и превращать его в жёсткое правило бессмысленно. Важно другое: входящие не должны стать новым кладбищем информации — тем самым, от которого вы уходили.

Частая ошибка

Самая частая причина смерти системы — входящие, которые доросли до двухсот записей. В этот момент разбор перестаёт быть делом на десять минут и превращается в проект, который всё время откладывают. Не доводите: лучше разобрать двадцать записей плохо, чем двести идеально.

Правило разбора

Разбор — это не «навести порядок». Это последовательность быстрых решений по каждой записи. Полное дерево решений мы разберём в главе 17, когда всё будет собрано в Obsidian, а пока — логика, которая за ним стоит.

По каждой записи вы отвечаете на вопросы по порядку, пока не получите «да»:

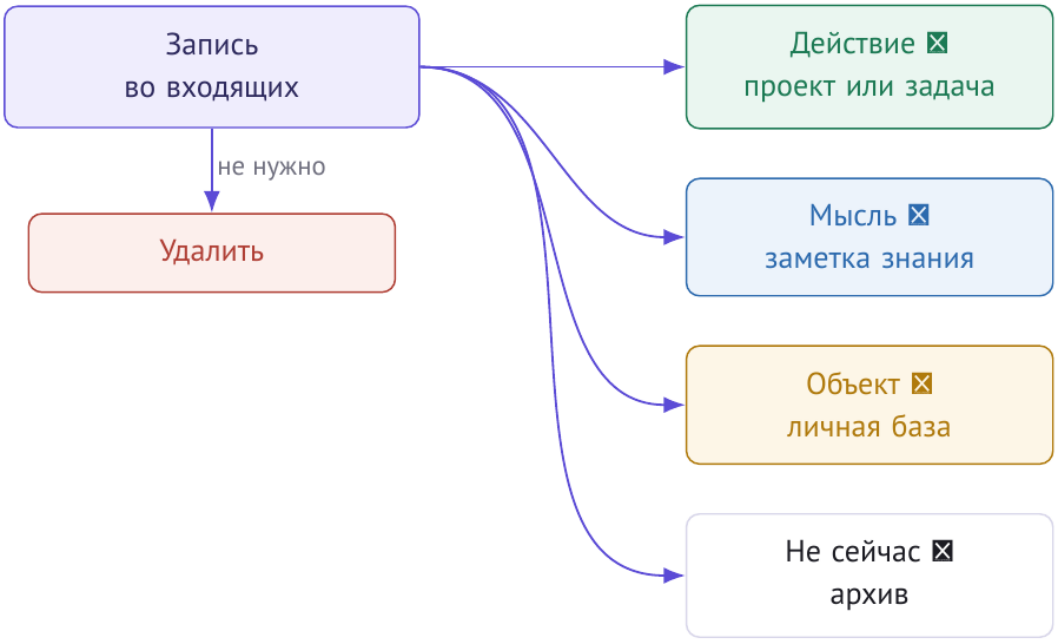
1. **Это ещё нужно?** Нет — удалить. Это самый частый и самый полезный ответ.
2. **Это действие?** Да — превратить в задачу или проект.
3. **Это относится к идущему проекту?** Да — положить в проект.
4. **Это про постоянную область жизни?** Да — в соответствующую сферу.
5. **Это мысль или знание?** Да — превратить в заметку своими словами.
6. **Это объект вашей коллекции?** (книга, фильм, место, человек, желание) Да — в личную базу.
7. **Ничего из перечисленного, но жалко?** В архив. Не в «разное».

Первый пункт важнее остальных вместе взятых. У большинства людей на разборе удаляется от трети до половины записей — и это отличная новость, а не признак того, что вы зря сохраняли.

Почему это работает

Быстрое удаление работает потому, что отделяет два разных чувства: интерес в момент находки и потребность в момент применения. Первое обманывает почти всегда. Второе — почти никогда.

Схема триажа



Разбор входящих. Левая ветка — самая частая, и это нормально.

Как выглядит хороший разбор

Вечер среды, у Алексея во входящих четыре записи. Он открывает папку и идёт сверху вниз.

«Кофейня рядом с офисом». Что с ней делать? Название он не помнит, адрес не записан. Удалить. Пять секунд.

«Курс по SQL, 6 недель». Нужен ли ему курс? У него уже есть проект про оконные функции с конкретным результатом. Курс туда не помещается — это отдельная большая штука, к которой он вернётся, если проект покажет, что знаний не хватает. Отправляет в архив одной строкой: «курс, если своими силами не выйдет».

«Идея: утренний блок под аналитику». Это не задача и не знание, это изменение рабочего режима — то есть его сфера «Карьера». Переносит туда, дописывает одно предложение: «пробую две недели, смотрю, сколько успеваю до обеда».

«Подкаст про интервальные повторения». Это источник, который он собирается послушать. Заводит заметку-источник со статусом «в очереди» и связывает с темой обучения.

Итого: четыре записи, около трёх минут, входящие пусты. Ничего героического.

Попробуйте сейчас

Найдите любое место, где у вас скопились сохранения, и разберите ровно десять записей по правилу выше. Не все — десять. Засеките время.

Скорее всего, уйдёт меньше пяти минут, и минимум четыре записи вы удалите. Это ощущение — «оказывается, это быстро» — важнее, чем сам результат.

Частые возражения

«А если я удалю что-то важное?» Практически исключено. Важное возвращается: если тема действительно ваша, она всплывёт снова — в разговоре, в работе, в чужой статье. Удалённая ссылка на статью — потеря нулевая, статью можно найти. Гораздо дороже обходится противоположное: система, в которую страшно заглядывать.

«Мне удобнее сразу раскладывать по папкам». Если вы действительно так делаете и это работает — прекрасно, не ломайте. Но проверьте честно: раскладываете ли вы по папкам в момент, когда мысль пришла в очереди в магазине? Обычно выясняется, что в этот момент запись всё равно улетает в заметки телефона, а «раскладываю сразу» относится только к спокойным ситуациям за компьютером.

«У меня нет времени на разбор». Разбор двадцати записей занимает пять-семь минут. Времени нет не на разбор, а на разбор двухсот записей — то есть на последствия того, что разбора долго не было.

Что дальше

Мы договорились, что у системы должна быть одна дверь и регулярная уборка за ней. Но чтобы разбирать, нужно понимать, куда раскладывать. В следующей главе — пять категорий, которых хватает практически для всего, что появляется в жизни обычного человека.

Проекты, сферы, знания, жизнь, архив

Самый распространённый способ организовать информацию — по темам. Папка «Работа», папка «Здоровье», папка «Финансы», внутри — подпапки, внутри них — ещё подпапки.

Выглядит логично и разваливается примерно через месяц. Причина в том, что тема не отвечает на главный вопрос: **что мне с этим делать и когда?**

Заметка про интервальное голодание и заметка про запись к врачу обе относятся к теме «здоровье». Но первая — это знание, к которому вы вернётесь когда-нибудь, а вторая — обязательство на этой неделе. Лежат они рядом, а требуют совершенно разного.

Гораздо практичнее раскладывать информацию по её роли в вашей жизни. Именно так устроена система PARA Тиаго Форте, откуда мы возьмём идею и слегка перестроим её под задачи обычного человека, а не только работника умственного труда.

Пять категорий

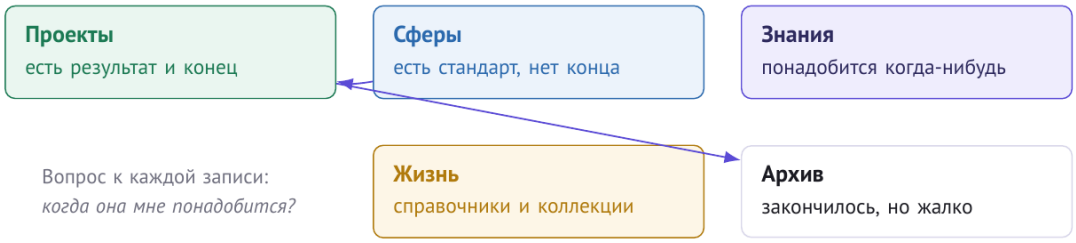


Схема 2. Пять категорий по применимости, а не по теме. Стрелки: сферы порождают проекты, завершённые проекты уходят в архив.

Проекты — то, что имеет результат и конец. «Сделать дашборд по оттоку», «подготовить поездку в Тбилиси», «поменять права». Проект можно закрыть и вычеркнуть — в этом его главное свойство.

Сферы — постоянные области ответственности. Здоровье, финансы, карьера, дом, отношения. У сферы нет финиша, зато есть стандарт: как выглядит «всё в порядке». Здоровье нельзя завершить, но можно понимать, что три тренировки в неделю — это нормально, а ноль — уже нет.

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.