

Что уже сделали до нас

Разбор 17 open-source проектов на GitHub — планировщики, трекеры привычек, AI-ассистенты и телеграм-боты. Данные с GitHub на 8 августа 2026: звёзды, стек, дата последнего коммита — по каждому проекту, всё проверено, ничего не выдумано.

Коротко: главный вывод

- Прямого конкурента нет. Ни один из 17 проектов не планирует жизнь раньше работы. Все без исключения начинают с задач, а отдых, семья и сон в лучшем случае вносятся как обычная задача с галочкой.
- В Telegram ниша пустая. Полноценного планировщика-ассистента в телеге нет вообще: есть либо учебные todo-боты на 0–3 звезды, либо generic GPT-боты для болтовни. Приличные планировщики живут в вебе и на десктопе. Это наша поляна.
- Что делают все и что придётся сделать нам: быстрый захват в инбокс, недельная сетка, экран «сегодня», перенос невыполненного, повторяющиеся дела, синк с календарём.
- Чего нет ни у кого — и это наши четыре отличия: брони жизни ставятся в неделю первыми; нет списка просроченного; стрик за ритуал планирования, а не за галочки; отказ от дела показан как победа.
- Главные грабли, видные по чужим issue: проекты тонут в запросах фич (у Super Productivity 1321 открытый issue), а стрики без паузы вызывают у людей злость — в трекерах привычек это самая обсуждаемая боль. Отсюда наш вывод: узкий скоуп и стрик, который прощает.

1. Недельное планирование и раскладка времени

Самая близкая к нам группа: люди раскладывают неделю руками или полуавтоматом.

Super Productivity

★ 21 196

github.com/super-productivity/super-productivity

TypeScript / Angular + Electron · живой, коммит 08.08.2026

Самый взрослый open-source планировщик: список задач + таймбоксинг + учёт времени в одном. Есть оценка времени на задачу, факт против плана, помидорный таймер, «фокус-режим», ежедневный и недельный разбор, импорт задач из календаря, Jira, GitHub. Всё локально, синк по своему облаку.

Берём: экран «сегодня» с одной задачей в фокусе; оценка времени и сравнение план/факт; ритуал разбора дня; идею, что незакрытое не висит вечно, а явно переносится вручную.

Не берём: интеграции с Jira/GitHub и учёт рабочего времени — это про наёмный труд; 1321 открытый issue показывает, куда уезжает продукт, если соглашаться на все просьбы.

Will Be Done

★ 121

github.com/will-be-done/will-be-done

TypeScript, local-first, SQLite, Docker · живой, коммит 08.08.2026

Молодой, но идейно самый близкий: весь продукт построен вокруг визуальной недели — каждый день колонка, задачи перетаскиваются между днями, сразу видно, где неделя перегружена. Плюс «Stash» — короткий постоянный список того, что держишь под рукой прямо сейчас. Работает офлайн, есть HTTP API.

Берём: неделя колонками как главный экран (сразу видно перегруз); «Stash» = наш короткий фокус-лист на неделю; принцип «открылось — сразу видно дела, без загрузки».

Не берём: vim-навигацию и клавиатурные шорткаты — у нас телефон; проектные доски и секции — для одного человека это лишний слой.

Open Sunsama

★ 46

github.com/ShadowWalker2014/open-sunsama

TypeScript, React 19, Postgres, Bun · живой, коммит 17.07.2026

Открытый клон Sunsama, заявлен как «первый AI-native task-менеджер»: задача не в том, чтобы человек кликал, а в том, чтобы его AI-ассистент управлял календарём через 24 инструмента MCP и REST API. Внутри — приоритеты P0-P3, подзадачи, таймблоки перетаскиванием, план против факта, перенос невыполненного в полночь.

Берём: правильную архитектуру: сначала честный API над задачами, ассистент ходит в него инструментами, а не пишет в базу руками. Нам это даст тот же движок и для бота, и для мини-аппки.

Не берём: автоперенос невыполненного на завтра в полночь — это ровно тот механизм, который копит долг и стыд; приоритеты P0-P3 — предпринимателю такая бюрократия не нужна.

Obsidian Day Planner

★ 2 693

github.com/ivan-lednev/obsidian-day-planner

TypeScript, Svelte, плагин Obsidian · живой, коммит 06.08.2026

Плагин, который превращает заметки дня в редактируемый таймлайн: пишешь «10:00 — созвон», и оно появляется полосой на шкале. Умеет тянуть события из Google/iCloud/Outlook календарей, показывать ближайшие 3 часа мини-полоской, вести учёт времени.

Берём: мини-таймлайн «ближайшие 3 часа» — идеальный формат ответа бота на вопрос «что сейчас»; подтягивание внешнего календаря, чтобы встречи занимали место в неделе автоматически.

Не берём: текстовый синтаксис задач — Тимур не будет писать разметку; десктопную сетку на весь день.

AutoPlan для Super Productivity

★ 30

github.com/00sapo/sp-autoplan

JavaScript, плагин · архивирован на GitHub, переехал на Codeberg

Маленький плагин, который автоматически раскладывает задачи по календарю: считает срочность из тегов, проектов, возраста задачи и дедлайна, режет крупные дела на куски и расставляет их в свободные окна по «карте времени» (когда ты вообще готов работать).

Берём: саму идею «карты времени» — окна, в которые вообще можно ставить работу, всё остальное неприкосновенно. Это технический прообраз наших броней. Ещё — предпросмотр перед применением: ассистент показывает, а не молча двигает.

Не берём: автоматическую нарезку задач на части — количество дел плодится, а это ровно то, от чего Тимур устал.

2. Классические task-менеджеры: что считается нормой

Смотрим не ради вдохновения, а чтобы понимать базовый набор, который люди ждут по умолчанию.

Taskwarrior

★ 5 996

github.com/GothenburgBitFactory/taskwarrior

C++, командная строка · живой, коммит 04.08.2026

Ветеран, командная строка. Главное в нём — формула срочности: приоритет, близость дедлайна, возраст задачи, теги, проект и блокировки складываются в одно число, и задачи сами выстраиваются в очередь. Никакого интерфейса, зато логика проработана двадцатью годами использования.

Берём: готовую формулу срочности как подсказку ассистенту — чем поднимать вверх, когда неделя трещит и надо резать.

Не берём: всё остальное. Это инструмент для инженеров, живущих в терминале.

Vikunja

★ 5 008

github.com/go-vikunja/vikunja

Go + Vue, self-hosted · живой, коммит 07.08.2026

Самая крепкая self-hosted замена Todoist/Asana: списки, канбан, таблица, гantt, календарь, напоминания, повторы, фильтры, полноценный API и CalDAV. Ставится в докер за минуту.

Берём: модель данных и CalDAV — если однажды понадобится синк с обычным календарём телефона, тут готовый образец.

Не берём: командные фишки и четыре вида одного и того же списка — для одного человека это шум.

Tasks.org

★ 5 471

github.com/tasks/tasks

Kotlin, Android · живой, коммит 07.08.2026

Зрелый мобильный task-менеджер под Android: гибкие повторы, геонапоминания, синк с Google Tasks, CalDAV, DAVx5, виджеты на экран. Хороший образец того, как мобильные повторяющиеся дела ведут себя на практике.

Берём: механику повторов (каждый вторник, каждый второй день, «через N дней после выполнения») — нам она нужна для зала три раза в неделю и массажа раз в неделю.

Не берём: мобильную нативность и настройки — у нас всё внутри телеграма.

github.com/TracksApp/tracks

Ruby on Rails, self-hosted · живой, коммит 03.08.2026

Классический GTD-веб на рельсах, живёт с 2004 года. Тут интересен сам ритуал GTD: инбокс, разбор входящего, контексты («звонки», «в городе», «дома»), список «когда-нибудь» и еженедельный обзор — тот самый weekly review, который у нас называется планёркой недели.

Берём: разделение на «сейчас», «ждёт ответа» и «когда-нибудь» — то, что не влезло в неделю, не умирает, но и не висит долгом; ритуал еженедельного обзора как отдельного события, а не отчёта.

Не берём: контексты и терминологию GTD целиком — это методология для гиков, обычного человека она отпугивает.

3. Привычки и стрики: как удерживают людей

Ровно та группа, у которой удержание получается — планировщики этому и завидуют.

Habitica

★ 14 044

github.com/HabitRPG/habitica

JavaScript, веб + мобильные · живой, коммит 07.08.2026

Привычки как ролевая игра: за выполненное дают опыт, золото и шмот, за проваленные «дейлики» персонаж получает урон и может умереть. Есть гильдии, совместные квесты, битвы с боссом, где команда теряет здоровье за пропуски.

Берём: идею внешней тяги — приложение само даёт повод вернуться; и обратную сторону: наказание работает против человека. Берём вознаграждение, выбрасываем урон.

Не берём: весь урон и смерть персонажа. Тимур уже наигрался в достигательство — механика вины его выключит за неделю.

Loop Habit Tracker

★ 10 107

github.com/iSoron/uhabits

Kotlin, Android · живой, коммит 21.07.2026

Самый уважаемый трекер привычек. Ключевая находка — «сила привычки»: вместо тупой цепочки считается плавный балл, который растёт с каждым повторением и мягко падает при пропуске. Пара пропусков после долгой серии не обнуляет прогресс, в отличие от «не разрывай цепь».

Берём: именно эту формулу вместо жёсткого стрика: пропустил планёрку в отпуске — балл просел, но не сгорел. Ещё — гибкие расписания «3 раза в неделю» без привязки к конкретным дням: это ровно формат наших броней.

Не берём: графики и статистику на пол-экрана — цифры про самого себя быстро надоедают.

Beaver Habit Tracker

★ 1 812

github.com/daya0576/beaverhabits

Python, NiceGUI, self-hosted · живой, коммит 27.07.2026

Трекер привычек с подзаголовком «без Goals» — автор сознательно выкинул цели, оставив только сам факт «сделал сегодня». Веб, ставится докером, есть API. Самая обсуждаемая просьба в issue — возможность ставить стрик на паузу, чтобы отпуск не обнулял серию.

Берём: философию «меньше сущностей» и живое подтверждение нашей идеи: стрик обязан прощать. Пауза на отпуск должна быть у нас с первого дня, а не через год по просьбам.

Не берём: минимализм до потери смысла — у нас всё-таки планирование, а не одна кнопка в день.

4. AI-ассистенты и захват на лету

Отсюда берём механику инбокса, голоса и проактивных сообщений.

Khoj

★ 36 392

github.com/khoj-ai/khoj

Python + Next.js, self-hosted · живой, коммит 02.08.2026

«Второй мозг» на LLM: чат по своим документам, семантический поиск, кастомные агенты с характером и своим набором инструментов. Важное для нас — автоматизации: агент сам, по расписанию, делает работу и присылает результат в почту или мессенджер, без запроса пользователя.

Берём: модель проактивности: ассистент пишет первым по расписанию (утро понедельника — планёрка, вечер — разбор инбокса). И агента с постоянной памятью о человеке — это наш профиль из онбординга.

Не берём: поиск по документам, RAG и корпоративную часть — нам нужен ассистент по жизни, а не по архиву файлов.

tg2obsidian

★ 154

github.com/dimonier/tg2obsidian

Python, aiogram, Whisper · живой, коммит 21.03.2026

Telegram-бот русского автора для захвата на лету: пишешь, диктуешь или пересылаешь что угодно в личную группу — бот складывает это в заметки, распознаёт голос через локальный Whisper или облачное API, делает OCR картинок, а по ключевым словам превращает сообщение в задачу. Отвечает эмодзи «ОК» на обработанное.

Берём: готовый рабочий образец нашего инбокса: голос → текст → задача, и главное — бот ничего не спрашивает, просто молча ловит и подтверждает галочкой. Разбор — потом, в спокойной обстановке. Это ровно то, о чём мы договорились с Тимуром.

Не берём: привязку к Obsidian и локальным файлам — у нас своя база.

chatgpt-telegram-bot

★ 3 460

github.com/n3d1117/chatgpt-telegram-bot

Python, python-telegram-bot · последний коммит 03.06.2025, развитие замедлилось

Самый популярный «правильно сделанный» LLM-бот в теге: потоковая печать ответа, транскрипция голосовых через Whisper, работа в группах, лимиты и учёт расходов на токены по каждому пользователю.

Берём: инженерные мелочи, на которых спотыкаются все: стриминг ответа вместо «бот думает», обработка длинных голосовых, честный подсчёт стоимости запросов.

Не берём: он просто болтает — ни памяти о человеке, ни задач, ни планирования.

Notepher Bot

★ 212

github.com/deptyped/notepher-bot

Vue, Telegram Mini App · последний коммит 12.03.2024

Заметки как мини-аппка внутри телеграма: работает офлайн, синхронизируется через Telegram Cloud Storage, интерфейс автоматически перекрашивается под тему клиента пользователя, есть теги, закрепления, поиск, чек-листы.

Берём: рабочий пример связки «бот + мини-аппка» именно на нашей платформе: как отдать данные в веб-вью, как подхватить тему телеграма, чтобы аппка выглядела родной, а не сайтом в окошке.

Не берём: хранение в Telegram Cloud Storage — нам нужна своя база, ассистент должен видеть данные на сервере.

5. И один мёртвый — как урок

Flow Dashboard

★ 1 682

github.com/onejgordon/flow-dashboard

Python 2.7, Google App Engine · НЕ ПОДДЕРЖИВАЕТСЯ, последний коммит 02.05.2024, регистрация закрыта

Идея была почти наша: не просто задачи, а личный дашборд — ежедневный дневник с настраиваемыми вопросами, привычки с недельными целями, обязательства, отслеживание главных задач года и аналитика по всему этому. Собрал 1682 звезды и умер: автор устал, платформа устарела.

Берём: две вещи: ежедневный короткий опрос-дневник (три вопроса вечером дают ассистенту материал для зеркала недели) и цели года, к которым привязываются недели.

Не берём: размах. Проект утонул в собственной ширине — дневник, привычки, цели, аналитика, интеграции. Нам нужно выбрать одно ядро и довести его.

Выводы: что забираем в наше приложение

Есть у всех — значит, придётся сделать и нам

- быстрый захват без формы: одна строка или голос, разбор потом
- экран «сегодня» с коротким списком, а не простыней
- недельный вид, где видно перегруз
- повторяющиеся дела с гибким расписанием («3 раза в неделю», без привязки к дням)
- явный перенос невыполненного руками, а не автоматом
- напоминания, которые приходят сами
- оценка «сколько это займёт» — без неё нельзя считать перегруз недели

Чего нет ни у кого из 17 — наша дельта

- Брони жизни первыми. Ни один проект не ставит семью, сон, зал и отпуск в неделю до работы. Максимум, что умеют — «карта времени» в AutoPlan, то есть окна, когда можно работать. Мы переворачиваем: сначала жизнь, дела в остаток.
- Отсутствие долга. Все до одного показывают просроченное. Open Sunsama даже автоматически переносит невыполненное в полночь — то есть копит его молча. У нас списка просроченного нет вовсе.
- Стрик за ритуал, а не за галочки. Стрики есть только в трекерах привычек и считаются по выполненным делам. Мы награждаем за то, что человек сел и спланировал — это он контролирует всегда.
- Отказ как победа. Нигде «я решил этого не делать» не считается результатом. Везде это либо удаление, либо провал.
- Разговор вместо интерфейса. Ассистент, который ведёт планёрку живым диалогом и пишет первым, есть только у Khoj — но там он про документы, а не про жизнь.

Грабли, на которые уже наступили другие

- Смерть от фич. У Super Productivity 1321 открытый issue, у Flow Dashboard размах убил проект. Люди просят интеграции, подзадачи в подзадачах, зависимости. Соглашаться нельзя.
- Стрик без паузы бесит. В Beaver Habits самая живая ветка обсуждения — «дайте поставить серию на паузу». Loop решил это давно: плавный балл вместо цепочки. Делаем сразу так.
- Наказание выключает. В Habitica персонаж получает урон за пропуск — механика вины, которая ровно противоположна нашей идеологии.
- Требовать ручной ввод оценок. AutoPlan честно пишет: без оценки времени планирование не работает. Значит, оценку должен предлагать ассистент сам, а человек только поправлять.
- Одиночная разработка на энтузиазме. Почти все хорошие проекты в этом списке — один автор. Половина из них замирает через год-два.

Что берём в стек

Для мини-аппки внутри бота — библиотека компонентов TelegramUI (github.com/telegram-mini-apps-dev/TelegramUI, 848★) и шаблон на React

(github.com/Telegram-Mini-Apps/reactjs-template, 425★): интерфейс сразу выглядит родным для телеграма. Архитектурно повторяем Open Sunsama — сервер с честным API над задачами, а бот и аппка ходят в него одинаково. Голос забираем как в tg2obsidian: Whisper на лету, молчаливое подтверждение, разбор вечером.

Как искали: GitHub API, около 50 поисковых запросов по темам (телеграм-боты и мини-аппки, LLM-планировщики, аналоги Todoist/Sunsama/Motion/Reclaim, трекеры привычек, голосовой захват). Всё, что попало в отчёт, открыто и прочитано: у каждого проекта скачан README, звёзды и дата последнего коммита взяты из API на 8 августа 2026. Найденное, но не подтверждённое, в отчёт не попало.