

База болей

- [Как это сделано](#)
- [Подключение агента к базе исследований](#)
- [База промтов](#)

Как это сделано

Предпосылки и подход

За несколько лет по Элли накопилось много исследований, пользовательских болей и наблюдений, но инфа была распределена по разным местам: отдельным документам, Figma/Miro, записям исследований, Яндекс Трекеру и другим артефактам.

Поэтому мы начали собирать все боли в отдельную очередь в Трекере

<https://tracker.yandex.ru/PMALLY>

Одна задача в этой очереди = одна пользовательская боль с цитатами, ссылкой на исследования, сырые данные, расшифровки и т.д. Если есть несколько источников в разное время, но проблематика одна — всё агрегируется в одной карточке боли.

При проведении исследований приходилось расшифровывать боли и вручную заводить их в Трекер. Но теперь все боли автоматом создаются на сервере при расшифровке, а затем улетают в Трекер.

Таким образом, все набранные ранее боли перенесены из Трекера на сервер, а все новые боли изначально создаются на сервере и отправляются в Трекер, чтобы обе базы были актуальны.

С болями на сервере можно работать через агента. Боли в трекере привязываются как Эпик к производственным тикетам в Трекере.

Что находится внутри

1. Исследования

Каждое исследование хранится отдельным `.md`-файлом в:

```
docs/ux/researches/
```

В карточке исследования могут быть:

- контекст и потребность;

- цели;
- дизайн исследования;
- метод;
- гипотезы;
- респонденты;
- результаты и находки;
- исходные материалы;
- артефакты результата;
- принятые решения;
- следующие действия;
- связанные исследования;
- роли и другие сущности базы.

Дополнительно фиксируются метаданные: период, тип исследования, платформы, роли, подразделения, типы торговых точек и другие параметры, если они известны.

2. Боли

Пользовательские боли вынесены из исследований в отдельные карточки:

`docs/ux/pains/`

Каждая пользовательская боль хранится отдельной карточкой. Мы стараемся фиксировать не только саму формулировку проблемы, но и контекст, силу подтверждения и связь с продуктовыми изменениями.

По каждой боли собираем:

- **ID и ссылку на Яндекс Трекер** — чтобы связать Markdown-карточку с рабочей карточкой боли в Трекере. Внутренний числовой ID совпадает с номером `PMALLY`-задачи.
- **Категорию** — собственно боль, идея или наблюдение. Это помогает не смешивать пользовательскую проблему с предложенным решением или просто зафиксированным фактом поведения.
- **Статус боли** — открыта, в работе, на проверке, решена или больше не актуальна. Статус относится именно к пользовательской проблеме, а не к связанной продуктовой задаче.
- **Дату последнего обновления** — чтобы понимать свежесть информации.
- **Контекст возникновения**: для каких ролей проблема подтверждена, для каких типов торговых точек, на какой платформе и в какой продуктовой области она возникает. Боль не распространяется автоматически на другие роли или платформы только потому, что сценарии похожи.
- **Темы** — например, «Графики», «Расхождения», «Окна», «Прогноз». Они нужны для группировки и поиска проблем по предметным областям.

- **Частоту / распространённость** — насколько часто проблема встречается у конкретной роли или когорты, если это вообще удалось определить.
- **Эмоциональность** — насколько сильно пользователи переживают или эмоционально реагируют на проблему, если такие данные есть. Частота и эмоциональность хранятся отдельно: редкая проблема может быть очень болезненной, а частая — восприниматься спокойно. Структура этих полей заложена непосредственно в шаблон карточки боли.
- **Описание боли** — сама проблема пользователя, без подмены её предполагаемым решением.
- **Цитату пользователя**, если сохранилось первичное свидетельство.
- **Подтверждение** — что именно позволяет считать проблему существующей: например, сколько участников исследования её отметили, была ли она найдена в одном или нескольких исследованиях, есть ли количественные данные. Наличие нескольких источников усиливает подтверждение, но само по себе ещё не означает известную распространённость проблемы.
- **Источники** — откуда вообще появилась боль: исследование, интервью, личное сообщение, стратегическая сессия, наблюдение и т. д. Для источника по возможности сохраняем исследование, дату, респондента, ссылку на расшифровку или запись и таймкод.
- **Связанные продуктовые задачи** — что уже делалось или планируется сделать в продукте в ответ на проблему. При этом закрытая задача не означает автоматически, что сама боль решена: решение могло закрыть её только частично или у одной боли может быть несколько связанных задач.

Например, для боли «Неудобный список Окон» в базе отдельно зафиксированы роли Старшего продавца и Старшего кассира-комплектовщика, типы ТТ, только веб-платформа, тема «Окна», описание проблемы, подтверждение от пяти участников исследования и две связанные продуктовые задачи.

Таким образом, карточка боли отвечает не только на вопрос **«что неудобно?»**, но и на более важные:

У кого возникает проблема? В каком контексте? Чем она подтверждена? Насколько она распространена? Откуда мы это знаем? Что с ней уже делали?

Да — кроме `pains` и `researches` у вас база уже содержит ещё три смысловых слоя и один служебный файл. Я бы описала это в статье так:

3. Подразделения

Сейчас в базе есть:

- Розница;
- Последняя миля;

- Горячая линия;
- Бэкофис.

Подразделение нужно, чтобы понимать более широкий контекст: где работает пользователь, к какому контуру относятся его сценарии и исследования.

4. Роли пользователей

Каждая карточка роли описывает тип пользователя Элли и его реальный рабочий контекст.

В карточке роли могут храниться:

- ID роли внутри базы;
- системный код роли в Элли;
- тип роли — менеджерская или линейная;
- альтернативные названия и сокращения;
- подразделение;
- применимые типы торговых точек;
- используемые платформы;
- количество пользователей;
- права в Элли;
- описание рабочего контекста.

Это позволяет анализировать исследования не абстрактно «по сотрудникам», а понимать, **кто именно сталкивается с конкретной проблемой и какие возможности у него есть в системе.**

Отдельно учитываются альтернативные названия. Например, сокращение **СП** можно сопоставить со Старшим продавцом, а не создавать ещё одну сущность только потому, что в одном исследовании роль названа сокращённо.

Есть и специальные агрегирующие роли — например, **Сотрудник** для линейных сотрудников торговых точек в целом. Она используется, когда знание относится сразу к широкой группе и перечислять все конкретные роли отдельно не имеет смысла.

5. Типы торговых точек

Это важно, потому что одна и та же роль или процесс может работать по-разному в разных форматах ТТ. Например, боль может быть подтверждена только для даркстора, только для вайтстора или сразу для обоих. Также здесь хранятся разные варианты названий. Например, **Вайт**, **Вайт стор**, **white store**, **whitestore** и **BC** могут быть сопоставлены с одной сущностью **Вайтстор**. Это особенно полезно для старых исследований, где терминология была не унифицирована.

6. Правила работы с базой AGENTS.md

По сути, это инструкция для AI-агентов: словарь + модель данных + правила интерпретации.

В ней описано:

- какие сущности есть в базе;
- как они связаны;
- что означают роли, подразделения, типы ТТ, платформы и боли;
- какие значения считаются допустимыми;
- как интерпретировать wikilinks;
- какие данные нельзя автоматически переносить между ролями, платформами или подразделениями;
- как отличать боль от идеи, наблюдения и продуктовой задачи;
- как работать с историческими исследованиями;
- как трактовать неполные или неизвестные данные.

Связи в базе

Все сущности хранятся обычными Markdown-файлами.

При этом они связаны между собой через wikilinks — ссылки вида:

```
[[roles/starshiy-prodavets|Старший продавец]]
```

или:

```
[[pains/103|Боль 103]]
```

Поэтому в совокупности получается не просто набор документов, а **граф знаний**.

Исследование может ссылаться на роль и боль, боль — на исследование и пользователя, а обратные ссылки позволяют посмотреть на эту же связь с другой стороны.

Например:

```
Исследование → нашло боль → боль относится к Старшему → подтверждается ещё двумя исследованиями
```

Связи должны быть явными: база специально запрещает автоматически приписывать исследованию роль, подразделение или проблему только потому, что такая связь кажется логичной.

Итого структура сейчас такая:

docs/ux/

└─ divisions/ — подразделения и операционный контекст

└─ pains/ — отдельные пользовательские боли

└─ researches/ — исследования и их результаты

└─ roles/ — роли пользователей, права и рабочий контекст

└─ shop-types/ — типы торговых точек

└─ AGENTS.md — правила понимания и обработки всей базы

Чем это полезно

Главный эффект базы — переход от архива исследований к накопительной системе знаний.

Раньше исследование в основном отвечало на вопрос:

Что мы узнали в этом конкретном проекте?

Теперь можно задавать вопросы уровнем выше.

Например:

Какие боли Старших повторяются в разных исследованиях?

Какие проблемы подтверждены несколькими независимыми источниками?

Что мы уже знаем про работу с расхождениями?

Какие боли Окон возможностей остаются нерешёнными?

Есть ли исследования, которые противоречат друг другу?

Какие роли исследованы хорошо, а про какие у нас почти нет данных?

Какие проблемы встречаются и в мобильном приложении, и в вебе?

Что мы уже знаем по теме, прежде чем запускать новое исследование?

Для исследователя база сокращает ручной поиск по старым материалам и помогает не исследовать повторно то, что уже известно.

Для продуктовой команды позволяет пройти от конкретной боли назад к доказательствам и понять, насколько хорошо она подтверждена.

Для новых участников команды становится картой уже накопленного знания об Элли.

Для AI-агентов Markdown и явные связи позволяют работать сразу со всем массивом знаний: искать паттерны, сопоставлять исследования, находить противоречия, собирать срезы по ролям или темам и при этом возвращаться к конкретным источникам.

По сути мы сохраняем исследования не только как историю проведённой работы, а превращаем их в **переиспользуемую память продуктовой и исследовательской команды**.

Подключение агента к базе исследований

База представляет собой обычную папку с Markdown-файлами на сервере Элли.

Вы разово настраиваете обращение агента к базе и затем можете работать с актуальным списком болей.

Первое подключение к базе

Перед началом работы нужны:

- активное подключение к VPN ВкусВилл ([см. п. 6.1. инструкции](#))
- Доступ к репозиторию Элли ([см. п. 3 инструкции](#))
- установленный AI-агент: Cursor, Claude Code, Codex, Gemini CLI и пр. на ваше усмотрение

С базой можно работать двумя способами — зависит от того, как настроен ваш AI-агент.

Вариант 1. Агент умеет работать с репозиторием напрямую

Если среда уже подключена к Git или умеет открывать репозиторий по ссылке, достаточно перейти к папке:

```
docs/ux
```

Вариант 2. Агент работает с локальными файлами

Если вы не подключаете агента к репозиторию напрямую, нужно сначала получить локальную копию файлов на компьютер.

Самый распространённый способ — клонировать репозиторий через Git. После этого на компьютере появится обычная папка проекта, внутри которой будет нужная:

```
docs\ux
```

Дальше агенту нужно дать доступ к этой локальной папке

File → Open Folder

Что в итоге получает агент

В обоих случаях результат одинаковый:

агент видит файлы репозитория

↓

видит docs/ux

↓

читает AGENTS.md

↓

может работать с исследованиями, болями, ролями и остальной базой

Работа с болями — только на чтение

Важно: папка с болями — это источник данных, а не рабочая зона для агента.

AI-агенту нужно настроить работу с `docs/ux/pains` в режиме **только чтения**.

Он может:

- искать боли;
- читать карточки;
- сопоставлять их между собой;
- анализировать источники и подтверждения;
- использовать их как контекст для ответа.

Но агент **не должен**:

- создавать новые файлы в `pains`;
- редактировать существующие боли;
- удалять карточки;
- менять статусы, формулировки, связи или источники;
- «дополнять» базу своими выводами;
- автоматически заводить новые боли на основании анализа.

Все изменения в базе болей должны проходить через штатный процесс и источник данных, а не появляться как побочный результат работы AI.

Для дополнительной защиты это правило лучше зафиксировать и в настройках/permissions агента, и в `AGENTS.md`.

Например:

Работа с болями

Папка ``docs/ux/pains/`` доступна только для чтения.

Разрешено:

- читать файлы;
- искать по ним;
- использовать данные для анализа.

Запрещено:

- создавать новые файлы;
- редактировать существующие;
- удалять файлы;
- менять статусы, формулировки, связи и источники;
- записывать в базу собственные выводы агента.

Если задача пользователя предполагает изменение боли, не вноси изменение самостоятельно — сообщи, что для этого нужен отдельный рабочий процесс.

База промтов

Все промты ниже предполагают работу с UX-базой в `docs/ux`.

Для папки `docs/ux/pains` действует режим **только чтение**: агент может искать, читать и анализировать карточки, но не должен создавать, редактировать или удалять боли.

Если хочется дополнительно застраховаться, к любому запросу можно добавлять:

Работай с docs/ux/pains только на чтение.
Ничего в базе более не создавай, не редактируй и не удаляй.
Не додумывай отсутствующие данные.
Для выводов показывай, на каких исследованиях и болях они основаны.

Быстро разобратся в базе

Что вообще есть

Изучи docs/ux и объясни, какие знания уже накоплены в базе.

Покажи:

- основные темы исследований;
- основные группы пользователей;
- наиболее часто встречающиеся продуктовые области;
- какие типы исследований представлены;
- какие крупные группы более можно выделить.

Не пересказывай каждый файл отдельно — дай карту базы.

Краткий онбординг нового человека

Представь, что я впервые работаю с Элли.

По материалам docs/их подготовь мне 10-минутный онбординг:

- кто основные пользователи;
- в каких контекстах они работают;
- какие ключевые сценарии исследовались;
- какие пользовательские проблемы уже известны;
- какие термины мне обязательно понимать.

Все утверждения основывай только на базе.

Что команда уже знает

Собери краткое резюме того, что исследовательская команда уже знает о пользовательском опыте Элли.

Не перечисляй исследования по очереди.

Синтезируй знания поперёк исследований.

Поиск информации по теме

Что уже знаем по теме

Найди всё, что база знает про [ТЕМА].

Собери:

- связанные исследования;
- связанные боли;
- роли;
- платформы;
- типы торговых точек;
- основные выводы;
- открытые вопросы.

Отдельно отметь, что подтверждено хорошо, а где данных мало.

Все боли по теме

Найди все боли, связанные с [ТЕМА].

Для каждой покажи:

- ID;
- краткую формулировку;
- роль;
- платформу;
- тип ТТ;
- подтверждение;
- источник;
- текущий статус.

Не объединяй разные боли только потому, что у них похожие формулировки.

Всё про конкретный сценарий

Что мы уже знаем про сценарий [СЦЕНАРИЙ]?

Найди связанные исследования, боли, роли и наблюдения.

Опиши сценарий глазами пользователя:

- задача;
- контекст;
- основные шаги;
- затруднения;
- известные потребности;
- связанные продуктовые изменения.

Поиск конкретного знания

Проверь, есть ли в базе подтверждение утверждения:

"[УТВЕРЖДЕНИЕ]"

Найди аргументы за и против.

Если данных недостаточно — так и скажи.

Не интерпретируй отсутствие данных как опровержение.

Работа с болями

Самые подтверждённые проблемы

Найди боли, которые подтверждаются несколькими независимыми источниками или исследованиями.

Для каждой покажи:

- сколько разных источников её подтверждают;
- какие это источники;
- насколько разнообразны роли и контексты;
- есть ли количественное подтверждение.

Не делай вывод о распространённости только по числу файлов.

Слабые боли

Найди боли, у которых самая слабая доказательная база.

Например:

- единственный источник;
- единичная цитата;
- нет исследования;
- неизвестна выборка;
- давно не обновлялись;
- нет подтверждения распространённости.

Объясни, чего именно не хватает для более уверенного вывода.

Старые открытые боли

Найди самые давно не обновлявшиеся боли со статусом open или in-progress.

Не делай вывод, что они неактуальны.

Покажи, какие из них стоит перепроверить на актуальность и почему.

Похожие боли

Найди пары или группы болей, которые потенциально описывают одну и ту же пользовательскую проблему разными словами.

Для каждой группы сравни:

- пользователя;
- контекст;
- причину;
- проявление;
- источники.

Не объединяй их автоматически.

Покажи только кандидатов на ручную проверку.

Боль целиком

Разбери боль [ID].

Покажи:

- что именно является проблемой;
- у кого она возникает;
- в каком контексте;
- чем подтверждается;
- что неизвестно;
- какие решения уже предпринимались;
- связанные исследования;
- возможные риски неправильной интерпретации.

Проверить распространённость

Что база позволяет сказать о распространённости проблемы [БОЛЬ/ТЕМА]?

Отдельно покажи:

- качественные свидетельства;
- количество участников, если известно;
- количественные данные;
- ограничения выборки.

Не превращай количество упоминаний в процент пользователей.

Старые и новые данные

Сравни более старые и более новые исследования по теме [ТЕМА].

Что изменилось со временем?

Какие старые проблемы сохраняются?

Какие исчезли?

Какие появились позже?

Не считай изменение автоматически результатом продуктовых доработок, если это не подтверждено.

Роли, подразделения и типы ТТ

Всё про роль

Собери UX-профиль роли [РОЛЬ].

Используй roles, pains и researches.

Покажи:

- рабочий контекст;
- платформы;
- права и ограничения;
- основные сценарии;
- ключевые боли;
- связанные исследования;
- темы, по которым знаний мало.

Сравнить две роли

Сравни пользовательский опыт ролей [РОЛЬ 1] и [РОЛЬ 2].

Покажи:

- общие сценарии;
- различия;
- общие боли;
- боли, характерные только для одной роли;
- различия в правах и контексте;
- где данных недостаточно.

Не переноси результаты одной роли на другую.

Что характерно именно для роли

Какие проблемы подтверждены именно для роли [РОЛЬ], а не для пользователей Элли вообще?

Исключи боли, где связь с этой ролью только предполагается.

Срез по подразделению

Собери срез пользовательских проблем для подразделения [ПОДРАЗДЕЛЕНИЕ].

Раздели результат по ролям.

Не распространяй боль одной роли на всё подразделение.

Вайтстор против даркстора

Сравни подтверждённые боли Вайтсторов и Дарксторов.

Раздели:

- общие проблемы;
- специфичные для Вайтстора;
- специфичные для Даркстора;
- проблемы, где тип ТТ неизвестен.

Не переносись с одного типа ТТ на другой без явных данных.

Специфика конкретного контекста

Какие пользовательские проблемы появляются только в контексте [ТИП ТТ / ПОДРАЗДЕЛЕНИЕ]?

Покажи доказательства специфичности этого контекста.

Web и mobile

Сравнение платформ

Сравни боли пользователей web и mobile.

Покажи:

- проблемы, подтверждённые на обеих платформах;
- только web;
- только mobile;
- случаи, где платформа не определена.

Не считай наличие похожей функции доказательством одинаковой боли.

Один сценарий на разных платформах

Сравни пользовательский опыт сценария [СЦЕНАРИЙ] в web и mobile.

Покажи только подтверждённые различия.

Если по одной платформе нет данных — отметь пробел.

Работа над продуктовой задачей

Контекст перед проектированием

Мы собираемся менять [ФУНКЦИЮ / СЦЕНАРИЙ].

Собери для продуктовой команды существующий UX-контекст:

- роли;
- боли;
- предыдущие исследования;
- известные ограничения;
- предыдущие попытки решения;
- риски, которые уже обнаруживались.

Какие боли затрагивает фича

По описанию продуктовой задачи:

[ОПИСАНИЕ]

найди существующие боли, которые она потенциально затрагивает.

Не создавай новые связи в базе.

Покажи кандидатов и объясни, почему они могут быть связаны.

Проверить решение против известных болей

Вот предлагаемое решение:

[ОПИСАНИЕ]

Сравни его с накопленными исследованиями.

Покажи:

- какие боли оно потенциально закрывает;
- какие не затрагивает;
- какие известные пользовательские риски может создать;
- какие утверждения требуют проверки с пользователями.

Какие пользователи могут пострадать

Для изменения [ОПИСАНИЕ] найди роли и сценарии, на которые оно потенциально может повлиять по данным базы.

Не придумывай технические зависимости.

Используй только связи пользовательского контекста.

Подготовка материалов

Подготовка к встрече

Мне предстоит встреча по теме [ТЕМА].

Собери шпаргалку на одну страницу:

- что мы точно знаем;
- где есть спорные моменты;
- какие боли связаны;
- какие решения уже предпринимались;

— какие вопросы стоит задать команде.

Ответ на вопрос

Коллега спрашивает:

"[ВОПРОС]"

Подготовь ответ исключительно на основании UX-базы.

Сначала дай краткий ответ.

Затем покажи исследования и боли, на которых он основан.

Отдельно обозначь ограничения данных.