

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 автоматически 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**.

Revision #2

Created 25 September 2026 14:29:01 by Ксения Морозова

Updated 25 September 2026 15:04:22 by Ксения Морозова