

# UX/UI ДИЗАЙН КАК СИСТЕМА

КАК ПРИНИМАТЬ РЕШЕНИЯ,  
РАБОТАТЬ С ДАННЫМИ  
И НЕ БОЯТЬСЯ КОМАНДЫ

Екатерина  
Маркевич

Екатерина Маркевич

**UX/UI дизайн как система. Как  
принимать решения, работать с  
данными и не бояться команды**

«Автор»

2026

## **Маркевич Е.**

UX/UI дизайн как система. Как принимать решения, работать с данными и не бояться команды / Е. Маркевич — «Автор», 2026

Устали от бесконечных правок? Боитесь критики? Чувствуете, что просто «рисуете картинки», а не решаете задачи? Эта книга — про системный подход в UX/UI дизайне. Она превращает хаос в порядок, а исполнителя — в эксперта. Вы получите: Уверенность, что ваши решения работают. Вы освоите работу с данными, метриками и проверенными гипотезами. Вы научитесь говорить с продактами, разработчиками и заказчиками. Это про работу в команде и продуктивное мышление. Опора в любой ситуации. Когда нет ТЗ, когда противоречивая обратная связь — вы будете знать, что делать. Навык задавать правильные вопросы. Не «что вы хотите?», а вопросы, которые экономят недели переделок. Система, а не набор техник. От первого эскиза до карьерного роста и управления командой. Внутри — 18 глав, 6 частей и ни капли лишнего. Только живые кейсы, чёткие алгоритмы и система, которая работает в реальных проектах. Для кого: для начинающих дизайнеров, продактов, тимлидов, предпринимателей и всех, кто устал от хаоса.

© Маркевич Е., 2026

© Автор, 2026

# Содержание

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Введение                                                                             | 6  |
| Как перестать рисовать картинки, и начать решать задачи?                             | 6  |
| О чём эта книга                                                                      | 7  |
| Для кого эта книга                                                                   | 8  |
| Как устроена эта книга                                                               | 9  |
| Как читать эту книгу                                                                 | 11 |
| Хотите больше?                                                                       | 12 |
| ГЛАВА 1. СИНДРОМ DRIBBBLE: КАК ПЕРЕСТАТЬ РИСОВАТЬ<br>КАРТИНКИ И НАЧАТЬ РЕШАТЬ ЗАДАЧИ | 13 |
| 1.1. Вступление                                                                      | 13 |
| 1.2. История первая. Максим                                                          | 14 |
| 1.3. История вторая. Алиса                                                           | 15 |
| 1.4. Что такое «синдром Dribbble» и почему он опасен                                 | 16 |
| 1.5. Погружение в проблему                                                           | 17 |
| 1.6. Решение и инструменты                                                           | 18 |
| 1.7. Итог: алгоритм действий                                                         | 23 |
| 1.8. Резюме                                                                          | 24 |
| 1.9. Практическое задание                                                            | 25 |
| 1.10. Чек-лист для самопроверки после доработки                                      | 26 |
| 1.11. FAQ                                                                            | 27 |
| 1.12. Связка с другими материалами                                                   | 28 |
| ГЛАВА 2. ГИПОТЕЗЫ ВМЕСТО ЛЮДЕЙ: КАК ПЕРЕСТАТЬ<br>ГАДАТЬ И НАЧАТЬ ИССЛЕДОВАТЬ         | 29 |
| 2.1. Вступление                                                                      | 29 |
| 2.2. История первая. Максим                                                          | 30 |
| 2.3. История вторая. Алиса                                                           | 31 |
| 2.4. Что такое «гипотезы вместо людей» и почему это опасно                           | 32 |
| 2.5. Погружение в проблему                                                           | 33 |
| 2.6. Решение и инструменты                                                           | 35 |
| 2.7. Итог: алгоритм действий                                                         | 38 |
| 2.8. Резюме                                                                          | 39 |
| 2.9. Практическое задание                                                            | 40 |
| 2.10. Чек-лист для самопроверки                                                      | 41 |
| 2.11. FAQ                                                                            | 42 |
| 2.12. Связка с другими материалами                                                   | 43 |
| ГЛАВА 3. СЛЕПОЕ СЛЕДОВАНИЕ ТЗ: КАК ПЕРЕСТАТЬ БЫТЬ<br>ИСПОЛНИТЕЛЕМ И СТАТЬ ПАРТНЁРОМ  | 44 |
| 3.1. Вступление                                                                      | 44 |
| 3.2. История первая. Максим снова ошибается                                          | 45 |
| 3.3. История вторая. Алиса                                                           | 46 |
| 3.4. Что такое «слепое следование ТЗ» и почему это опасно                            | 48 |
| 3.5. Погружение в проблему                                                           | 49 |
| 3.6. Решение и инструменты                                                           | 51 |
| 3.7. Итог: алгоритм действий                                                         | 54 |
| 3.8. Резюме                                                                          | 55 |
| 3.9. Практическое задание                                                            | 56 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 3.10. Чек-лист для самопроверки                                    | 57 |
| 3.11. FAQ                                                          | 58 |
| 3.12. Связка с другими материалами                                 | 59 |
| ГЛАВА 4. ДВОРЕЦ ВМЕСТО ДОМА: КАК ПЕРФЕКЦИОНИЗМ<br>УБИВАЕТ ПРОДУКТЫ | 60 |
| 4.1. Вступление                                                    | 60 |
| Конец ознакомительного фрагмента.                                  | 61 |

# **Екатерина Маркевич**

## **UX/UI дизайн как система. Как принимать решения, работать с данными и не бояться команды**

### **Введение**

#### **Как перестать рисовать картинки, и начать решать задачи?**

Ты когда-нибудь чувствовал, что твой дизайн похож на красивую открытку? Все восхищаются, но никто не понимает, как это работает.

Ты сдаёшь макет руководителю на проверку. Он смотрит на работу и задаёт вопросы, на которые у тебя нет ответов:

— Почему кнопка здесь? Почему такой цвет? Зачем этот элемент вообще нужен? Ты поворачиваешься к своему экрану, и внутри появляется холодок. Ты не знаешь ответа. Ты просто нарисовал «красиво».

А через месяц к верстке приступают разработчики. У них нет времени приходить к тебе с вопросами. Они просто додумывают то, что ты не отрисовал в макете: сами решают, каким будет состояние кнопки при нажатии, придумывают поведение формы при ошибке, адаптируют элементы под мобильные устройства. Они не дизайнеры, они инженеры. Их решения продиктованы тем, как проще написать код, а не тем, как удобнее пользователю. Дизайн начинает разваливаться в руках разработчиков, потому что он был картинкой, а не системой.

Позже ты открываешь готовый продукт и не узнаёте свой дизайн. Кнопки другие, отступы съехали, состояний ошибок нет. Продукт выглядит сырым, и совсем не таким каким был на макете в Figma.

Знакомо?

Я видела это слишком много раз. В разных командах, на разных проектах, у специалистов разного уровня. Ошибки повторяются с пугающим постоянством. Не потому, что дизайнеры глупые или ленивые, а потому, что нас никто не учит мыслить системно.

Эта книга — именно о системе.

## О чём эта книга

Эта книга — не просто о дизайне. Это система: пошаговый алгоритм, который превращает хаос в порядок.

Здесь нет абстрактных теорий и сложной терминологии — только конкретные ошибки, проверенные решения и план действий. Она для тех, кто хочет не просто рисовать макеты, а принимать осознанные продуктивные решения. Для новичков, которым нужна опора, и для опытных специалистов, которые стремятся систематизировать свою работу.

Я — UX-методолог. Моя первая книга, *«Методы UX-исследований. 25 методов с примерами и метриками»*, стала справочником по инструментам работы с данными UX-исследований.

Эта — совсем другая книга. Без лишней теории я покажу главные барьеры и страхи, которые мешают дизайнерам расти, и дам алгоритмы их преодоления. Это путеводитель по реальным проблемам профессии: от непонимания бизнес-задач до страха перед AI и неопределенностью на рынке.

Внутри — не догадки, а выверенная методология. Система, которую я собрала по крупицам из реальных проектов, ошибок и побед.

## Для кого эта книга

**Для начинающих дизайнеров**, которые хотят строить карьеру системно, а не наугад. Для тех, кто хочет научиться аргументировать свои решения так, чтобы работу принимали с минимальным количеством правок.

**Для продуктовых менеджеров и тимлидов**, которые хотят понимать, где в дизайн-процессе скрываются риски и как их предотвращать. Эта книга поможет тебе говорить с дизайнерами на одном языке и выстраивать эффективные процессы в команде.

**Для предпринимателей и основателей стартапов**, которые хотят понимать, как сделать дизайн так, чтобы он работал на бизнес, а не просто «красиво выглядел». Ты узнаешь, как оценивать качество дизайна и принимать правильные решения без лишних затрат.

**Для всех, кто чувствует, что застрял**— боится критики, не знает, как защищать свои решения, не понимает, как расти в профессии. Если ты хочешь перестать быть «дизайнером-исполнителем» и стать «дизайнером-партнёром» — эта книга тоже для тебя.

## Как устроена эта книга

Книга состоит из шести частей, которые ведут от базовых ошибок к стратегическому мышлению и профессиональному росту.

### **Часть 1. Пять ошибок, которые убивают продукты (главы 1–5)**

Это то, с чего мы начинаем. Пять системных ошибок, которые я вижу в работе дизайнеров снова и снова. Каждая глава — это один кейс. Я показываю, как ошибается Максим — дизайнер, который учится на своих провалах, и как правильно делает Алиса — дизайнер, который использует систему. Вы увидите контраст и поймёте, какой путь ведёт к результату. После прочтения первой части, ты:

- перестанешь рисовать картинки и начнёшь решать задачи (глава 1);
- научишься проверять гипотезы, а не выполнять задачи наугад (глава 2);
- начнёшь задавать вопрос «Зачем?» и предлагать альтернативы (глава 3);
- будешь начинать с простого, а не строить дворцы (глава 4);
- станешь продумывать полноценные сценарии, а не только «счастливый путь» (глава 5).

### **Часть 2. Инструменты, которые делают тебя экспертом (главы 6–8)**

Здесь я рассказываю о том, как перестать бояться исследований, задавать правильные вопросы и читать метрики. Как работать в условиях неопределённости и как принимать обратную связь так, чтобы она делала вас сильнее, а не больше.

Ты научишься:

- не бояться исследований — пять страхов и способы с ними справиться (глава 6);
- задавать правильные вопросы — пять типов вопросов для любой ситуации (глава 7);
- принимать решения на основе данных — пять основных метрик для дизайнера + дополнительные (глава 8).

Это превращает тебя из «дизайнера-исполнителя» в «дизайнера-партнёра», к мнению которого прислушиваются.

### **Часть 3. Как работать с неопределённостью и обратной связью (главы 9–10)**

Здесь мы разбираем два самых сложных навыка, которые отличают джуниора от мидла и синьора: умение работать в условиях неопределённости (когда нет чёткого ТЗ) и умение принимать обратную связь (когда заказчик говорит «мне не нравится»).

Ты научишься:

- работать с неопределённостью — пять страхов и способы с ними справиться (глава 9);
- работать с обратной связью — пять типов критиков и стратегии работы с ними (глава 10).

### **Часть 4. Карьера и рынок (главы 11–14)**

Это отдельный блок для тех, кто хочет не просто выжить в профессии, а строить карьеру осознанно. В 2026 году рынок дизайна изменился, и важно понимать, как оставаться востребованным.

Ты узнаешь:

- пять побед Макса — как навыки превращают хаос в результат (глава 11);
- как не выгореть на рынке 2026 года — карьерная стратегия и работа с тревогой (глава 12);
- как собрать портфолио, которое продаёт — структура кейсов и работа с бизнес-контекстом (глава 13);
- как использовать AI как инструмент — а не бояться, что он заменит тебя (глава 14).

### **Часть 5. Работа в команде (главы 15–16)**

Здесь мы разбираем, как выстраивать эффективное взаимодействие с разработчиками и организовывать работу так, чтобы не утонуть в хаосе.

Ты узнаешь:

как подружиться с разработчиками — и говорить на одном языке (глава 15);

как создавать дизайн-системы — и не утонуть в компонентах (глава 16).

### **Часть 6. Продуктовое мышление (главы 17–18)**

Это финальный блок, который превращает тебя из дизайнера, который «делает красиво», в дизайнера, который решает бизнес-задачи.

Ты узнаешь:

как мыслить, как продакт — понимать бизнес, формулировать гипотезы и влиять на продукт (глава 17);

как делать доступные интерфейсы — и почему это больше не опция, а обязанность (глава 18).

Эта часть — про твою ценность для бизнеса, про влияние на продукт и про то, как оставаться востребованным в меняющемся мире.

## Как читать эту книгу

Можешь читать её последовательно — главу за главой, или открыть любую главу, которая сейчас интересует больше всего. Каждая глава — это самостоятельная история. Ты не потеряешь смысл, если начнёшь с середины.

В конце каждой главы я даю:

резюме — чтобы ты запомнил главное;

практическое задание — чтобы ты сразу применил знания;

чек-лист — чтобы проверить себя;

FAQ — я ответила на самые частые вопросы.

Это делает книгу не просто «интересным чтивом», а рабочим инструментом, который можно использовать в повседневной практике.

И ещё кое-что важное

Эта книга — не про то, как стать идеальным дизайнером. Идеальных не бывает. Но есть системные подходы, которые сводят количество ошибок к минимуму.

Эта книга — про то, как начать проектировать осознанно. Как превратить дизайн из «картинки» в рабочий инструмент, который решает задачи пользователей и приносит пользу бизнесу.

## Хотите больше?

Если после прочтения появятся вопросы или ты захочешь обсудить что-то из книги — я всегда открыта для диалога.

А если хочешь углубиться в теорию — моя первая книга «Методы UX-исследований» уже ждёт на [Litres](#). В ней я разобрала 25 методов, которые помогут вам принимать решения на основе данных, а не интуиции.

Две книги — два уровня.

Первая — про инструменты.

Вторая — про систему.

Вместе они дают полную картину: что делать и как думать. Как принимать решения и как их защищать. Как строить карьеру и как оставаться востребованным.

Ну что, поехали?

# **ГЛАВА 1. СИНДРОМ DRIBBBLE: КАК ПЕРЕСТАТЬ РИСОВАТЬ КАРТИНКИ И НАЧАТЬ РЕШАТЬ ЗАДАЧИ**

## **1.1. Вступление**

Эта глава — о самой массовой ошибке, которую я наблюдаю у дизайнеров на всех уровнях: от новичков до вполне опытных специалистов. Ошибка настолько типична, что в профессиональном сообществе у неё есть устойчивое название — «синдром Dribbble».

Сначала я расскажу две истории. В первой — дизайнер делает всё, как его учили на курсах по «красивому дизайну». Во второй — дизайнер делает всё, как учит системный подход. Посмотри, в ком ты узнаешь себя. И решай, какой путь тебе ближе.

## 1.2. История первая. Максим

Максим — начинающий дизайнер. Он получил первый серьёзный заказ — интерфейс для системы управления складским учётом. Не красивый сайт, а сложный технический инструмент: таблицы, графики, отчёты, фильтры.

Максим открывает Figma и начинает творить.

Ему кажется, что интерфейс должен быть стильный, современный, с градиентами, тенями и плавными анимациями. Он вдохновляется Dribbble и Pinterest: окошки с неоновыми подсветками, кнопки с эффектом стекла, анимированные иконки складов, которые летают при наведении.

Он работает три дня. Макет выглядит потрясающе. Это не просто интерфейс — это арт-объект.

Максим с гордостью отправляет макет заказчику.

Заказчик — мужчина лет под 60, с двадцатилетним стажем в логистике — открывает макет. Несколько секунд молчит. Разглядывает. Щурится. Потом нажимает на кнопку «Добавить товар» — она светится неоном, плавно увеличивается и издаёт воображаемый «пшик». А рядом нет ни поля для ввода артикула, ни кнопки «Сохранить», ни даже простой таблицы с товарами.

Ни-че-го.

Заказчик трёт переносицу:

— Красиво, но... где таблица? Где столбцы с артикулами и количеством? А где кнопка «Сохранить»? Мне не нужно угонять звездолёт, — шёпотом произносит он, переводя взгляд с одной светящейся и мигающей кнопки на другую. — Мне нужен экран, по которому мои кладовщики быстро проведут тысячу позиций. Чтобы я открыл — и увидел всё. Сразу. Без анимации. Без неона.

Максим смотрит на свой макет свежим взглядом — и его передёргивает. Он нарисовал новогоднюю ёлку, а не интерфейс для склада.

Он перерисовывает. Убирает неоновые подсветки, добавляет таблицу, делает кнопки плоскими. Но теперь у него 15 цветов на экране: зелёные кнопки «Добавить», красные «Удалить», синие «Редактировать», жёлтые предупреждения, фиолетовые баннеры... Глаза разбегаются.

Заказчик снова пишет: «Красиво, но я ничего не вижу. Всё такое яркое, ничего не могу разглядеть».

Максим убирает 70% декора. Оставляет чёрно-белую таблицу с четырьмя столбцами, серые кнопки, чёткую структуру.

Заказчик: «Вот теперь всё отлично. Спасибо».

Максим смотрит на свой минималистичный интерфейс и понимает: именно это и должно было быть с самого начала. Три дня потеряны. А могли бы быть три часа.

## 1.3. История вторая. Алиса

Алиса — дизайнер с системным подходом. Она тоже получила заказ на интерфейс для складского учёта. Но она начинает не с Figma, а с вопросов.

Она садится с заказчиком и выясняет:

**Кто пользователи?** Кладовщики, 20–50 лет, большая часть с бухгалтерским образованием. Им важна скорость. Им привычны таблицы.

**Какая главная задача?** Быстро найти товар по артикулу и отметить его количество.

**Какие данные критичны?** Артикул, наименование, остаток, место на складе.

Алиса берёт лист бумаги и рисует простую таблицу в четыре столбца. Без теней, без градиентов, без анимации. Показывает заказчику. Тот говорит: «Да, вот это понятно. А можно ещё кнопку "Редактировать" рядом с каждой строкой?»

Алиса добавляет кнопку. Всё.

Она открывает Figma и за 20 минут собирает чистый, понятный интерфейс: таблица, кнопка «Добавить товар», фильтры сверху, форма редактирования в модальном окне. Единый стиль. Чёткая иерархия.

Заказчик принимает макет с первого раза.

Алиса потратила на проект 4 часа. Максим — 3 дня и 2 итерации.

В чём разница?

|                  | Максим                        | Алиса                    |
|------------------|-------------------------------|--------------------------|
| <b>Начало</b>    | Открыл Figma и начал рисовать | Задала вопросы заказчику |
| <b>Подход</b>    | «Сделаю красиво»              | «Решу задачу»            |
| <b>Результат</b> | Переделка, потеря времени     | Принято с первого раза   |
| <b>Ощущение</b>  | Разочарование                 | Уверенность              |

Максим проектировал картинку. Алиса проектировала решение.

И эту разницу видят все: заказчик, разработчик, продукт-менеджер. Они не говорят: «Какая красивая страница!» Они говорят: «Я быстро нашёл, что хотел, и сделал то, за чем пришёл». Это и есть победа.

## 1.4. Что такое «синдром Dribbble» и почему он опасен

В профессиональном сообществе это явление называют «синдромом Dribbble» — и не случайно.

Dribbble — это платформа, где дизайнеры выкладывают свои «шедевры»: красивые, стильные, но часто абсолютно нежизнеспособные решения. Там можно зависнуть на час, вдохновляясь идеями, и незаметно для себя начать проектировать картинку, а не решение.

Это известная проблема, о которой говорят на конференциях, в блогах и в вакансиях. Работодатели всё чаще пишут: «Понимать, что дизайн — это не картинка, а инструмент решения задач». Заказчики жалуются на «красивые, но бесполезные макеты». Разработчики устали спрашивать: «Что здесь вообще происходит?»

В этом главная опасность: ты перестаёшь думать о пользователе и начинаешь думать о том, как бы сделать «вау». А «вау» — это не про UX. Это про искусство.

UX — про то, чтобы пользователь сделал то, зачем пришёл, не напрягаясь, без бесконечных поисков и переходов от одной страницы к другой.

Как выглядит синдром Dribbble на практике:

Элементы не связаны с задачами — кнопка есть, но она не видна или не понятна.

Визуальный шум — 15 цветов, 7 шрифтов, анимации на каждом элементе.

Декоративные элементы перебивают функциональные — картинка важнее кнопки «Купить».

Нет иерархии — глаза разбегаются, не понятно, куда смотреть.

Нет состояний — кнопка есть, но hover/active/disabled не проработаны.

## 1.5. Погружение в проблему

### 1.5.1. Почему дизайнеры попадают в «синдром Dribbble»

#### Причина №1. «Меня учили делать красиво»

На курсах и во многих обучающих методиках акцент часто делают на визуальную составляющую. Студентов учат подбирать цвета, работать с типографикой, создавать композиции, но редко учат задавать вопросы: «Кто пользователь?», «Какая задача?», «Что для них критично?»

Как с этим справиться:

Переключи фокус. Вместо «Как сделать красиво?» спроси себя: «Как сделать полезно?» Красота — это дополнение к решению, а не самоцель. UX-дизайн начинается с вопросов, а не с Figma.

#### Причина №2. «Вдохновляюсь картинками»

Дизайнер открывает Dribbble или Behance, видит красивые работы и начинает копировать стиль, не вникая в контекст. Он не знает, кто пользователи, какая задача, какие ограничения.

Как с этим справиться:

Вдохновляйся решениями, а не картинками. Смотри не на то, как красиво, а на то, как это работает. Какие задачи решает данный интерфейс? Как он помогает пользователю? Если ты не знаешь ответа — это просто картинка.

#### Причина №3. «Хочу впечатлить»

Дизайнер боится, что его работу сочтут слишком простой. Он добавляет сложные анимации, нестандартные шрифты, множество цветов — чтобы показать, что он «профессионал», что ему платят не зря его зарплату.

Как с этим справиться:

Простота — признак профессионализма. Самый сложный навык — сделать интерфейс понятным и чистым. Заказчик и разработчики оценят не сложность, а удобство.

## 1.6. Решение и инструменты

### 1.6.1. Шаг первый. Тест на пять секунд

Это самый простой, но самый мощный инструмент, который я знаю. Он не требует специальных знаний, дорогих программ или подготовки респондентов. Он занимает ровно пять минут, но спасает от недель переделок и неловких моментов на презентациях.

Как это работает:

Ты показываешь свой макет человеку, который никогда его не видел, ровно на пять секунд, а потом закрываешь и задаёшь вопросы.

За пять секунд человек не успевает разобраться в деталях — он видит только общую картину. И именно это нам и нужно: мы проверяем, понятно ли с первого взгляда, что тут происходит и куда нажимать.

Пошаговая инструкция:

Найди человека, который не участвовал в проекте. Это может быть коллега из другого отдела, друг, любой человек, который согласится потратить пять минут. Важно: он не должен был видеть этот макет раньше и знать, о чём проект. Это даст честную реакцию.

Открой макет на весь экран.

Засеки пять секунд и покажи макет. Ничего не объясняй! Человек должен увидеть всё сам.

Закрой макет и задай три вопроса:

«**Что ты запомнил?**» — Если запомнил кнопку «Купить», цену или заголовок — ты молодец. Если запомнил красивую картинку или анимацию — ты создал шум.

«**Как ты думаешь, что здесь самое главное?**» — Если говорит про то, что ты считаешь главным — иерархия работает. Если говорит про второстепенное — пересматривай приоритеты.

«**Куда бы ты нажал?**» — Если на нужную кнопку — отлично. Если на декоративный элемент — переделывай.

Три сценария, когда тест на пять секунд спасает тебя.

**Сценарий первый.** Перед показом заказчику.

Ты уверен, что всё круто. Но тест на пять секунд может выявить то, чего ты сам не замечаешь, потому что привык к своему макету.

Если друг говорит: «Я не понял, куда нажимать» — ты слышишь это до того, как показал заказчику. Ты можешь спокойно переделать, не краснея на презентации.

Когда ты показываешь макет заказчику впервые, у тебя есть один шанс произвести впечатление профессионала. Если заказчик сразу увидит, что главная информация потерялась, он подумает: «Дизайнер не видит очевидных вещей». А если он увидит чистый, понятный макет — он скажет: «Вот это профессионал».

**Сценарий 2.** Перед передачей разработчикам.

Разработчики — твои главные партнёры. Они искренне радуются, когда открывают макет и видят не кучу разбросанных цветов, шрифтов и карточек, а чёткую систему.

Когда ты заранее продумал все состояния, поставил отступы и определил главные элементы — разработчики не задают вопросов. Они просто берут и делают.

А когда макет перегружен, стили разрознены, а главная кнопка спрятана — разработчики начинают задавать вопросы. Много вопросов. Каждый вопрос — твоя недоработка. Ты теряешь время на объяснения и переделки. Переживаешь и нервничаешь, всё ли в порядке теперь или будут еще вопросы. И в этом режиме проводишь каждый день до полной сдачи проекта! Команда начинает сомневаться в твоём профессионализме, как и ты сам.

Тест на пять секунд помогает избежать этого задолго до того, как макет попадёт в руки разработчиков.

**Сценарий 3.** Когда ты сам сомневаешься.

Ты сидишь над макетом несколько часов. Перекладываешь элементы, меняешь цвета. Ты перестаёшь видеть дизайн свежим взглядом.

Тест на пять секунд — твой спасательный круг. Найди любого человека, покажи на пять секунд. Ответы могут подтвердить, что ты всё сделал правильно, — и ты выдохнешь. Или покажут, что ты упустил очевидное, — и ты исправишь это до того, как кто-то ещё увидит макет.

Это отрезвляет. Когда ты сомневаешься, мозг рисует страшные картинки: «А вдруг я ничего не понимаю?» Тест на пять секунд даёт объективные данные. Ты понимаешь: либо всё хорошо, либо есть конкретная точка для доработки. Это снимает тревогу и возвращает уверенность.

Нюансы и подводные камни

Что делать:

Показывать макет новому человеку, который не видел его раньше.

Ничего не объяснять перед показом — человек должен увидеть всё сам.

Записывать ответы дословно, чтобы не потерять детали, лучший вариант — на диктофон или видео.

Тестировать на трёх–пяти добровольцах, чтобы увидеть паттерны.

Чего не делать:

Не показывать макет тому, кто уже видел его — они не будут объективны.

Не говорить: «Сейчас проверим, видна ли кнопка» — это наводит на ответ.

Не интерпретировать ответы «на ходу» — это искажает результаты.

Не останавливаться на одном мнении — оно может быть случайным.

Как интерпретировать ответы, «что запомнил/куда нажал бы»:

| Ответ                              | Что значит                          | Что делать                                        |
|------------------------------------|-------------------------------------|---------------------------------------------------|
| «Кнопка "Купить"» или «Цена»       | Всё отлично.<br>Главное заметно.    | Молодец, работай дальше.                          |
| «Красивая картинка» или «Анимация» | Ты создал шум.<br>Главное не видно. | Уменьши декор, выдели главное действие.           |
| «Не понял, куда смотреть»          | Хаос. Иерархия не работает.         | Начни с чистого листа, расставь приоритеты.       |
| «Нажал бы на картинку»             | Картинка оказалась важнее кнопки.   | Сделай картинку второстепенной, кнопку — главной. |

Чек-лист для самопроверки перед тестом:

Перед тем как показывать макет, проверь себя по этому списку. Если хоть один пункт вызывает сомнение — лучше внести изменения сразу и тестируй максимально приближённую к рабочей версию проекта.

Я знаю, какое действие пользователь должен совершить в первую очередь.

Это действие визуально выделено (цвет, размер, положение).

На макете не больше 2-3 акцентных цветов.

Все элементы имеют свою иерархию (заголовки, подзаголовки, текст).

Кнопки понятны без подписей (или подписаны понятно).

Я продумал хотя бы одно состояние ошибки или пустого экрана.

Я могу объяснить, зачем каждый элемент на экране.

### 1.6.2. Шаг второй. Визуальная иерархия

После теста на пять секунд ты узнаешь, есть ли проблема. А этот шаг поможет тебе исправить её, даже если ты не учился на художника.

Законы визуальной иерархии (для тех, кто не любит теорию):

| Что делать                                                           | Как проверить                                                                  |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------|
| <b>Размер. Самое важное — самое большое.</b>                         | Сравни заголовок, подзаголовок и остальной текст. Что крупнее?                 |
| <b>Цвет. Самое важное — самое яркое / контрастное.</b>               | Выключи цвет (включи ч/б режим). Всё ещё видно главное?                        |
| <b>Положение. Самое важное — в центре или в начале пути взгляда.</b> | Посмотри на макет 1 секунду. Куда упал взгляд?                                 |
| <b>Отрицательное пространство. Вокруг важного — больше воздуха.</b>  | Есть ли отступы вокруг кнопки? Или она сливается с другими элементами дизайна? |
| <b>Консистентность. Одинаковые элементы — одинаково выглядят.</b>    | Все кнопки одного типа выглядят одинаково?                                     |

Как проверить иерархию без специальных знаний?

Самый простой способ — отвернуться от экрана на две секунды, быстро повернуться и посмотреть на макет ровно одну секунду. А потом закрыть глаза и ответить на вопрос: «Что я запомнил?»

Если это кнопка «Купить» или другая важная информация — иерархия работает.

Если это второстепенная картинка, анимация или фон — что-то точно нужно менять.

### 1.6.3. Шаг третий. Сетка и модульная система

Когда ты работаешь над интерфейсом, хаос возникает там, где нет системы. Сетка — твой главный помощник в борьбе с хаосом.

Раздели экран на колонки (8, 12 или 16 — зависит от сложности) — это помогает выровнять элементы и не плодить случайные отступы.

Задай базовый шаг (4, 8 или 10 пикселей) — все отступы и размеры должны быть кратны базовому шагу.

Всё, что можно выравнивать — выравнивай — это снижает когнитивную нагрузку на пользователя.

Не используй больше 2 шрифтов — для читаемости и консистентности.

Все кнопки одного типа — одинаковые — пользователь привыкает и не переучивается.

### 1.6.4. Шаг четвёртый. Состояния элементов

Это то, что отличает профессиональный макет от любительского: продуманные состоя-

| Состояние               | Когда применять                     |
|-------------------------|-------------------------------------|
| <b>Default</b>          | Кнопка / поле просто есть на экране |
| <b>Hover</b>            | Пользователь навёл курсор (десктоп) |
| <b>Active / Pressed</b> | Пользователь нажал                  |
| <b>Focused</b>          | Поле ввода активно                  |
| <b>Disabled</b>         | Элемент недоступен                  |
| <b>Error</b>            | Пользователь ошибся                 |
| <b>Success</b>          | Всё сделано правильно               |
| <b>Empty state</b>      | Нет данных                          |

ния.

#### 1.6.5. Шаг 5. Проверка на доступность (Accessibility)

Это не «дополнительная опция». Это обязательная часть работы. 15–20% пользователей имеют особенности зрения, и, если ты игнорируешь доступность — ты теряешь до 20% аудитории.

Минимальный набор для проверки:

| Что проверить                                    | Как проверить                                               | Норма                                                 |
|--------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------|
| <b>Контрастность текста</b>                      | Используй плагин в Figma (Contrast, Stark) или онлайн-чекер | WCAG AA (4.5:1 для обычного текста, 3:1 для крупного) |
| <b>Размер шрифта</b>                             | Посмотри, читается ли текст с расстояния 50 см              | Не менее 16–18px для основного текста                 |
| <b>Кликабельные области</b>                      | Проверь, что кнопки и ссылки не меньше 44x44px              | Минимум 44x44px (особенно для мобильных)              |
| <b>Цвет как единственный носитель информации</b> | Обязательно проверь, понятно ли без цвета                   | Информация должна быть понятна без цвета              |

## 1.7. Итог: алгоритм действий

Вот чёткая последовательность шагов, которая поможет тебе не попасть в число жертв «синдрома Dribbble»:

Начни с вопросов. Не с Figma. Кто пользователь? Какая главная задача? Какие данные критичны?

Нарисуй на бумаге. Простая схема структуры. Только блоки, без декора.

Собери прототип в Figma без декора. Только структура, кнопки, текст. Минимум цветов.

Проведи тест на пять секунд. Найди человека, покажи на пять секунд, задай три вопроса.

Исправь по результатам теста. Если кнопка не видна — выдели её. Если слишком много шума — убери лишнее.

Проверь доступность. Контрастность. Размер шрифта. Кликабельные зоны.

Покажи заказчику / разработчикам. Теперь ты уверен, что интерфейс понятен.

## 1.8. Резюме

«Синдром Dribbble» — это когда ты проектируешь картинку, а не решение. Это известная проблема в индустрии, о которой говорят профессионалы по всему миру.

UX-дизайн — это не про красоту. Это про то, чтобы пользователь достиг своей цели без лишних усилий.

Тест на пять секунд — самый простой и эффективный способ проверить, понятен ли твой дизайн с первого взгляда.

Визуальная иерархия — размер, цвет, положение и пространство вокруг элемента определяют, куда посмотрит пользователь.

Сетка и модульная система — основа порядка в интерфейсе.

Состояния элементов — отличают профессиональный макет от любительского.

Доступность — не опция, а обязательная часть работы.

## 1.9. Практическое задание

Возьми свой текущий проект (или любой макет, над которым ты работаешь).

Проведи тест на 5 секунд с 3 разными людьми.

Запиши их ответы на три вопроса: что запомнил, что главное, куда нажал бы.

Проанализируй: совпадает ли их восприятие с твоей задумкой?

Если нет — переделай макет, используя чек-лист ниже.

## 1.10. Чек-лист для самопроверки после доработки

- ☐ Главное действие видно с первого взгляда.
- ☐ На макете не больше 3 акцентных цветов.
- ☐ Все элементы имеют чёткую иерархию.
- ☐ Кнопки понятны без подписей (или подписаны понятно).
- ☐ Есть состояния: hover, active, disabled (где необходимо).
- ☐ Текст читается без усилий (минимальный размер 16–18px, контрастность 4.5:1).
- ☐ Кликабельные области не меньше 44x44px (для мобильных).

## 1.11. FAQ

— А если заказчик сам просит «красиво»?

Покажи ему альтернативы: твой вариант с акцентом на задачу и их вариант. Объясни, что «красиво» без пользы — это потеря денег. Приведи примеры из этой главы.

— А если Dribbble — это моё вдохновение?

Вдохновляйся. Но бери оттуда идеи для UI, а не для UX. Когда ты работаешь над структурой — Dribbble не помощник. Когда ты работаешь над визуалом — да, можно посмотреть на тренды.

— А если я работаю в агентстве и клиент хочет «вау-эффект»?

Сделай «вау» в тех местах, где это не мешает функциональности. Например, приветственный экран, анимация загрузки, иконки. Но основная задача — кнопка «Купить» — должна быть видна.

— Как понять, что я перестарался с декором?

Проведи тест на 5 секунд. Если человек запоминает только декор, а не действие — ты перестарался.

## 1.12. Связка с другими материалами

Это только первый шаг. Мы разобрали «синдром Dribbble» — ошибку, с которой начинается путь почти каждого дизайнера. Ты уже знаешь, как проверять дизайн на 5 секунд, как управлять вниманием пользователя и почему красота не должна быть самоцелью.

Но есть ещё четыре ошибки впереди. И каждая из них поджидает тебя там, где ты меньше всего ожидаешь. Не в визуале. Не в туториалах по Figma. А в том, как ты думаешь, принимаешь решения и общаешься с командой.

Следующая ошибка — самая коварная, потому что она маскируется под уверенность. Ты начинаешь верить, что знаешь пользователей лучше, чем они сами. Ты перестаёшь спрашивать и начинаешь придумывать. И в этот момент ты снова наступаешь на грабли.

Что это за ошибка? Узнаешь в следующей главе.

А пока — подумай: в своём последнем проекте ты действительно знала, что нужно пользователю? Или просто предполагала?

## ГЛАВА 2. ГИПОТЕЗЫ ВМЕСТО ЛЮДЕЙ: КАК ПЕРЕСТАТЬ ГАДАТЬ И НАЧАТЬ ИССЛЕДОВАТЬ

### 2.1. Вступление

Открой свои старые макеты. Те, которым уже год или два, или те, которые были сделаны, в самом начале твоего пути. Посмотри на них свежим взглядом — с теми знаниями, которые у тебя есть сейчас. Уверена, ты сразу заметишь: там, где раньше казалось «круто и стильно», теперь видишь перегруженный шум, а кнопка, которую ты тогда прятал между иллюстрациями, чтобы она не бросалась в глаза и не портила всю красоту, теперь воспринимается как очевидная и серьёзная ошибка.

Это не стыдно. Это значит, что ты растёшь.

В первой главе мы разобрались, как перестать рисовать картинки и начать решать задачи. Мы научились проверять дизайн за 5 секунд, управлять вниманием пользователя и отличать UI от UX.

Но есть ещё одна проблема.

Даже если ты научилась делать чистые, понятные интерфейсы, ты всё равно можешь наступить на вторые грабли. Они лежат не в визуале, а в голове. Там, где мы начинаем придумывать, а не исследовать. Где мы думаем, что знаем пользователя лучше, чем он сам.

Помнишь Максима из первой главы? После того как он переделал свой интерфейс для склада, он успокоился. Ему казалось, что теперь он всё делает правильно. Он даже немного загордился: «Всё, я понял, как надо. Теперь я профессионал».

А потом он получил новый проект. И снова попал в ловушку. Но на этот раз — другую.

## 2.2. История первая. Максим

«Я знаю, что им нужно. Я же сам пользователь».

Сколько раз я слышала эту фразу от дизайнеров, менеджеров и даже себя самой. И каждый раз, когда она прозвучала, внутри меня что-то ёкает. Потому что это путь в никуда.

После истории со складом Максим выдохнул, закрыл тот злополучный макет и подумал: «Всё, теперь я понял. Больше никаких неонов и градиентов, если в них нет необходимости. Только польза, только данные». Он даже записал себе на стикере: «Дизайн — это решение проблем», прилепил на монитор и почувствовал себя почти гуру.

Прошло время. Максиму дали новый проект — мобильное приложение для доставки еды. Он открывает Figma, смотрит на чистое поле и думает: «Ну, я же сам заказываю еду. Я знаю, что людям нужно. Всё просто: меню, корзина, оплата. Погнали».

И он рисует макет за два дня, добавляет «фишки»: голосовой поиск, анимацию корзины, кнопку «Повторить заказ», потому что это же удобно, лично ему бы пригодилось, значит многие оценят это изменение. Он чувствует себя гением. На этот раз точно всё круто.

Максим с гордостью показывает макет заказчику. Тот смотрит, кивает и говорит:

— Красиво, но... у нас целевая аудитория — люди 45+, они не пользуются голосовым поиском. И почему ты убрал фильтр по кухням? Мы его тестировали — он увеличивает конверсию на 20%.

Максим замирает. Он опять... Снова... Его «гениальные идеи» оказались никому не нужны. Он гадал. А заказчик — знал, потому что проводил исследования, и все данные приложил отдельным файлом к техническому заданию. Тот файл под скрепочкой... В самом конце письма.

И вот наш дизайнер снова сидит, смотрит на свой макет и чувствует, как внутри нарастает то самое холодное: «Ну как так? Я же хотел как лучше...»

## 2.3. История вторая. Алиса

Алиса получает точно такой же проект — мобильное приложение для доставки еды.

Но она не открывает Figma. Она открывает данные.

Она смотрит аналитику: кто пользователи? Как они заказывают? Какие у них боли?

Она видит: целевая аудитория — люди 45+. Они не используют голосовой поиск. Им важны фильтры по кухням. Они привыкли листать меню.

Она спрашивает продакта: «Какие гипотезы вы уже проверяли? Что сработало, а что нет?»

Она узнаёт, что фильтр по кухням уже тестировали — он дал +20% конверсии.

Алиса делает простой, понятный интерфейс: меню, корзина, фильтр по кухням. Без голосового поиска, без лишней анимации.

Она показывает заказчику. Тот смотрит и говорит: «Да. Это то, что нам нужно. Ты поняла, кто наши пользователи».

Алиса потратила на исследование 2 часа, на дизайн — один день. Максим потратил 2 дня на рисование и получил переделку.

В чём разница?

|                  | Максим                         | Алиса                                 |
|------------------|--------------------------------|---------------------------------------|
| <b>Начало</b>    | Открыл Figma и начал рисовать  | Открыла данные и начала анализировать |
| <b>Подход</b>    | «Я знаю, что им нужно»         | «Давайте узнаем, что им нужно»        |
| <b>Результат</b> | Переделка, потеря времени      | Принято с первого раза                |
| <b>Ощущение</b>  | Разочарование, чувство провала | Уверенность, доверие команды          |

Максим гадал. Алиса исследовала, и эту разницу видит вся команда. Когда ты приходишь с данными, тебя слушают. Когда ты приходишь с догадками — тебя переделывают.

## **2.4. Что такое «гипотезы вместо людей» и почему это опасно**

В профессиональном сообществе эту проблему называют «гаданием вместо исследований» или «дизайном на основе интуиции». Это вторая по частоте системная ошибка в работе дизайнеров.

Как выглядит «гадание вместо исследований» на практике:

Ты добавляешь фишки, которые нравятся тебе: «Это же удобно, я бы так сделал».

Ты не проверяешь гипотезы на пользователях: «Я и так знаю, что им нужно».

Ты игнорируешь данные: «Цифры не важны, главное — мой опыт».

Ты не умеешь формулировать гипотезы: «Мне кажется, они хотят голосовой поиск».

Ты не понимаешь, какие вопросы задавать: «Ну, что вам нужно?» — в ответ: «Не знаю».

На этом заканчивается весь диалог.

Почему это опасно:

Ты тратишь время на ненужные фишки — создаёшь то, что никто не использует.

Ты теряешь доверие команды — твои решения не работают.

Ты подводишь команду — они ждут результатов, а ты гадаешь.

Ты чувствуешь себя самозванцем — не можешь объяснить, почему сделал именно так.

Бизнес теряет деньги — клиенты уходят к тем, кто их понимает.

В этой главе мы разберём:

почему дизайнеры гадают вместо того, чтобы исследовать;

как перестать придумывать и начать узнавать;

три простых метода, которые заменят гадание на знание;

как защитить свои решения перед командой.

## 2.5. Погружение в проблему

### Почему дизайнеры гадают вместо того, чтобы исследовать?

Причин несколько. И все они очень человеческие.

#### Причина №1. «Исследования — это скучно и долго»

Давай признаемся: исследования кажутся чем-то из разряда «научной фантастики». Это же надо составлять вопросы, искать респондентов, проводить интервью, анализировать данные... Гораздо веселее сразу открыть Figma и начать рисовать! Там же результат виден сразу, а в исследованиях — непонятно что.

Как с этим справиться:

Перестань воспринимать исследования как «большой и сложный проект». Начни с малого. Один опрос двух человек занимает 15 – 20 минут. Один разбор тикетов поддержки — 30 минут. Одно наблюдение за пользователем — 10 минут.

Это не «научная фантастика». Это просто разговор с людьми, которые пользуются твоим продуктом. И он занимает меньше времени, чем ты думаешь.

Попробуй провести одно мини-исследование в своём текущем проекте и посмотреть, как изменится твой дизайн. Ты удивишься, насколько это увлекательно — видеть, как твои гипотезы подтверждаются или опровергаются.

#### Причина №2. «Я же сам пользователь»

Это самая коварная причина. Мы все считаем, что знаем, что нужно другим. Но мы — не другие. Мы — дизайнеры, которые сидят в Figma по восемь часов в день. Хорошо если только по восемь. Наш опыт и опыт обычного пользователя — это две разные вселенные.

Как с этим справиться:

Напомни себе: ты — не пользователь. Твой опыт, привычки, знания — всё это делает тебя уникальным, и не типичным. Чтобы не попасть в эту ловушку:

Сформулируй гипотезу как вопрос. Вместо «Я знаю, что пользователям нужен голосовой поиск» — скажи: «Я предполагаю, что голосовой поиск поможет пользователям быстрее находить товары. Проверим?»

Найди хотя бы одного реального пользователя. Поговори с ним. Спроси, что ему нужно. Не придумывай — узнай.

Запиши свои предположения и проверь их. Если ты думаешь, что пользователи хотят X — проверь это на 5–10 людях. Если они скажут «нет» — ты сэкономил неделю работы.

И помни: «Я бы так сделал» — это про тебя. «Они так делают» — это про пользователей. Не путай.

#### Причина №3. «У меня нет времени»

Дедлайны горят, заказчик торопит, разработчики уже начали писать код. Некогда проводить исследования, нужно делать макеты. Знакомо?

Но спешка — это путь к переделкам.

Как с этим справиться:

На самом деле у тебя есть время. Просто ты его неправильно распределяешь.

Один день на исследование → спасает неделю переделок.

Пятнадцать минут на опрос → спасает день на перерисовку.

Тридцать минут на анализ тикетов → спасает два дня на обсуждения с командой.

Возьми таймер. Установи себе пятнадцать минут на одно мини-исследование. Если ты уложишься — ты уже победитель. А если не уложишься — ты всё равно потратишь меньше времени, чем на переделку.

#### Причина №4. «Я боюсь услышать правду»

А вдруг пользователи скажут, что моя идея — полный провал? Вдруг они не понимают мой дизайн? Это больно. Проще не спрашивать и продолжать жить в иллюзии, что всё круто, хотя бы пока.

Как с этим справиться:

Это самый сложный страх. Он лежит не в логике, а в эмоциях. Но есть один приём, который помогает.

Спроси себя: «Что хуже — узнать правду сейчас или узнать её после того, как я потратил недели на дизайн?»

Если ты узнаешь правду сейчас:

ты сэкономишь время и нервы;

ты сможешь исправить ошибку до того, как её увидят заказчик и разработчики;

ты почувствуешь облегчение, потому что перестанешь двигаться в неизвестности.

Если ты узнаешь правду позже:

ты потратишь недели на то, что не работает;

ты потеряешь доверие команды;

ты будешь чувствовать себя самозванцем.

Правда — это не враг. Это твой союзник. Она говорит тебе: «Ты можешь сделать лучше». И это нормально — ошибаться. Ошибки создают рост там, где на них учатся. Это не нормально — игнорировать ошибки и продолжать идти в никуда с гордо поднятой головой и делать вид, что так и надо.

## 2.6. Решение и инструменты

### 2.6.1. Шаг первый. От «гипотезы» к «вопросу»

Первый шаг — перестать думать в формате «Я знаю, как надо» и начать думать в формате «А что на самом деле?».

Что делать:

Вместо того чтобы сразу рисовать макет, сформулируй гипотезу в формате:

**«Я предполагаю, что если мы сделаем [действие], то [пользователь] получит [результат], потому что [причина].»**

Или, если проще:

**«Я думаю, что пользователям нужен [функционал], потому что [причина]. Давайте проверим.»**

Примеры:

Гадание: «Пользователям нужен голосовой поиск».

Гипотеза: «Мы предполагаем, что голосовой поиск увеличит конверсию среди пользователей 18–25 лет, потому что они привыкли к голосовым командам в соцсетях. Проверим это через опрос десяти пользователей этой группы».

Разница очевидна: второе — это не просто «мне кажется», это запрос на проверку.

Как формулировать гипотезу: простой шаблон

Я использую этот шаблон всегда — он спасает от бесконечных споров:

**«Если мы [сделаем X], то [пользователи] смогут [сделать Y], что приведёт к [изменимому результату Z], потому что [причина].»**

Пример для интернет-магазина сумок:

«Если мы добавим фильтр по цвету на главную страницу, то пользователи смогут быстрее находить нужную сумку, что сократит время поиска на 30% и увеличит конверсию, потому что 60% пользователей указывают цвет как главный критерий выбора».

Теперь это не гадание, а гипотеза, которую можно проверить. И если она не подтвердится — мы не потеряли недели на дизайн, мы просто поняли, что это не работает.

### 2.6.2. Шаг второй. Три простых метода для проверки гипотезы

У нас нет бюджета на дорогие исследования? Не проблема. Есть три способа, которые работают и ничего не стоят.

**Метод №1:** Опрос пяти–десяти человек

Это самый простой способ. Ты берёшь свой вопрос и задаёшь его пяти–десяти людям из твоей целевой аудитории.

Что делать:

Сформулируй вопрос: «Пользуешься ли ты функцией X?» или «Что для тебя важно при выборе Y?».

Задай его пяти–десяти людям.

Запиши ответы.

Сделай вывод.

Пример:

Я хочу проверить, нужен ли голосовой поиск в приложении для доставки еды. Я спрашиваю десять друзей: «Пользуешься ли ты голосовым поиском в приложениях для заказа еды?» Все десять отвечают: «Нет, я просто листаю меню». Вывод: голосовой поиск не нужен.

Нюансы:

Это не полноценное исследование, но оно даёт первичную информацию.

Важно спрашивать честно, не наводя на ответ.

Если восемь из десяти говорят одно — это уже сигнал.

### **Метод №2: Анализ тикетов поддержки**

Это мой любимый способ, потому что он даёт реальные данные о болях пользователей. Ты открываешь тикеты в службе поддержки или чаты с пользователями и ищешь паттерны.

Что делать:

Попроси доступ к тикетам поддержки (обычно это легко сделать).

Собери пятьдесят–сто тикетов.

Выпиши повторяющиеся вопросы и проблемы.

Сделай вывод: что чаще всего беспокоит пользователей?

Пример:

Я анализирую тикеты поддержки интернет-магазина. Вижу: 30% вопросов — «Как отменить заказ?», 20% — «Не пришло письмо с подтверждением», 15% — «Не могу найти товар». Вывод: нужно сделать кнопку «Отменить заказ» заметнее, проверить работу писем и улучшить поиск.

Почему это работает:

Это бесплатно.

Это реальные данные от реальных пользователей.

Это экономит время, потому что ты сразу видишь, что болит, и не гадаешь.

**Метод №3: «Shadowing»** — посмотри на пользователя со стороны

Этот метод прост до безобразия: ты наблюдаешь за тем, как пользователь взаимодействует с продуктом, и просто записываешь.

Что делать:

Найди одного–трёх пользователей твоего продукта.

Попроси их выполнить конкретную задачу (например, «Найди товар и добавь в корзину») и комментировать свои действия.

Смотри молча. Не помогай. Не объясняй. Просто наблюдай.

Записывай моменты, где пользователь зависает, ошибается или делает не то, что ты ожидала.

Пример:

Я смотрю, как пользователь ищет телефон в интернет-магазине. Он набирает «iPhone 15» в строке поиска, жмёт Enter, видит двести результатов и зависает. Он не понимает, как отсортировать по цене. Я записываю: проблема — пользователь не видит сортировку. Через минуту я добавила кнопку «Сортировать по цене» в заметное место.

Почему это работает:

Ты видишь реальные проблемы пользователя, которых ты не замечала, сидя в Figma.

Это бесплатно.

Это занимает пятнадцать минут, а спасает от недель переделок.

### **2.6.3. Шаг третий. Как защитить свой дизайн перед командой, используя исследования**

Это самый важный навык, который отличает джуниора от синьора. И он строится на том, что ты не гадаешь, а знаешь.

Вот твой скрипт для защиты дизайна:

**«Мы проверили эту гипотезу через [метод], и выяснили: [факты]. Поэтому я предлагаю сделать [решение], потому что это решает проблему [пользователей].»**

Пример:

«Мы проверили гипотезу через опрос десяти пользователей и анализ тикетов поддержки. Выяснили, что 60% пользователей не находят фильтр по цвету и уходят с сайта. Поэтому я предлагаю разместить фильтр на главной странице, чтобы пользователи сразу видели его и могли выбрать нужный цвет. Это сократит количество уходов и увеличит конверсию».

Такой подход делает тебя неуязвимым. Все твои решения основаны на данных, а не на личных предпочтениях и поэтому имеют обоснование.

## 2.7. Итог: алгоритм действий

Сформулируй гипотезу — не «я знаю», а «я предполагаю, проверим?»

Проверь гипотезу простым методом — опрос пять–десять человек, анализ тикетов поддержки или наблюдение за пользователем.

Сделай вывод. Если гипотеза подтвердилась — двигайся дальше. Если нет — формулируй новую.

Защити своё решение перед командой — используй данные, а не фразу «мне кажется».

## 2.8. Резюме

Гадание — путь в никуда. Ты тратишь время на то, что никому не нужно, и подводишь команду.

Перестань придумывать, начни узнавать. Формулируй гипотезы в формате «Если мы сделаем X, то получим Y, потому что Z».

Не надо сложных исследований. Достаточно опроса пяти–десяти человек, анализа тикетов поддержки или наблюдения за пользователем.

Исследования — твоя броня. Они делают твой голос весомым и защищают от бесконечных правок.

Когда ты приходишь с данными, тебя слушают. Когда ты приходишь с догадками — тебя переделывают.

## 2.9. Практическое задание

Возьми свой текущий проект (или любой старый макет) и сделай три вещи:

**Задание 1.** Сформулируй гипотезу

Сформулируй одну гипотезу по шаблону:

*«Если мы сделаем [X], то пользователи смогут [Y], что приведёт к [Z], потому что [причина].»*

**Задание 2.** Проведи опрос

Задай свой вопрос пяти–десяти людям из целевой аудитории. Запиши их ответы. Сделай вывод: подтвердилась гипотеза или нет?

**Задание 3.** Проанализируй тикеты (если есть доступ)

Открой тикеты поддержки и выпиши топ три повторяющиеся проблемы. Подумай, как их можно решить дизайном.

## 2.10. Чек-лист для самопроверки

- ☐ Я сформулировал гипотезу, а не просто нарисовал макет.
- ☐ Я проверил гипотезу на реальных пользователях (опрос, тикеты, наблюдение).
- ☐ Я могу объяснить, почему я сделал именно так, опираясь на данные.
- ☐ Я не добавляю фичи, которые «кажутся» удобными.
- ☐ Я знаю, кто мои пользователи, и понимаю их боли.

## 2.11. FAQ

— А что, если у меня нет доступа к тикетам поддержки?

Попроси у менеджера или продакта. Обычно дают, если объяснить, зачем. Если не дают — проведи простой опрос. Он тоже даёт информацию.

— А если у меня нет времени на исследования, дедлайн горит?

Тридцать минут на опрос пяти человек сэкономит тебе дни на переделках. Лучше потратить тридцать минут сейчас, чем неделю потом.

— А что, если я проведу опрос, а он скажет, что моя гипотеза неверна? Это же провал.

Это не провал. Это успех! Ты узнал правду до того, как потратил недели на дизайн. Ты сэкономил время и нервы. Двигайся дальше с новыми знаниями.

— А как быть, если команда не верит в исследования и требует сразу рисовать?

Ты можешь сказать: «Давайте потратим тридцать минут на опрос пяти пользователей, и если они скажут, что это им нужно, я нарисую альтернативный вариант, если нет — работает с этим макет, который есть». Обычно команда соглашается, потому что это быстро и легко.

— А как понять, что я действительно исследовал, а не просто спросил у коллеги?

Коридорный тест — отличный инструмент для быстрой проверки. Поймать коллегу в коридоре, показать макет на пять секунд и спросить «что запомнил?» — это ценно. Это помогает увидеть очевидные проблемы до того, как ты покажешь макет заказчику.

Но есть важное различие: коллеги — не пользователи. Они знают контекст, они дышат одним с тобой воздухом. Их ответы — это мнение людей из индустрии, а не реальных пользователей.

Как отличить коридорный тест от настоящего исследования:

Если ты просто спросил у коллеги «как тебе?» и он сказал «нормально» — это не исследование. Это мнение.

Если ты показал макет трём-пяти коллегам на пять секунд, записал, что они запомнили, и сделал выводы — это уже коридорный тест. Он даёт первичную информацию, но не заменяет разговор с реальными пользователями.

**Настоящее исследование начинается там, где ты выходишь за пределы своей команды и разговариваешь с теми, для кого ты проектируешь.**

Если не знаешь, где найти пользователей — начни с друзей и знакомых, которые не связаны с разработкой. Они не знают контекста — и это делает их мнение честнее.

## 2.12. Связка с другими материалами

Мы разобрали вторую ошибку. Ты уже знаешь, что гадание — это не метод, а ловушка. Ты научился формулировать гипотезы, проверять их на пользователях и защищать свои решения данными, а не «мне кажется».

Максим тоже научился. После истории с голосовым поиском он твёрдо усвоил: сначала спросить, потом рисовать. Он даже перестал добавлять фичи «на всякий случай» — теперь он сначала проверяет, нужны ли они. Он горд собой. Ему кажется, что теперь он защищён от ошибок.

Но есть ещё одна проблема.

В следующей главе мы разберём, как перестать быть просто исполнителем и начать быть партнёром. Как превратить «сделай попап» в осмысленную задачу. И как задавать правильные вопросы, чтобы не переделывать макеты по три раза.

А пока — подумай: в своём последнем проекте ты просто делал, что сказали? Или сначала понял, зачем это нужно?

## **ГЛАВА 3. СЛЕПОЕ СЛЕДОВАНИЕ ТЗ: КАК ПЕРЕСТАТЬ БЫТЬ ИСПОЛНИТЕЛЕМ И СТАТЬ ПАРТНЁРОМ**

### **3.1. Вступление**

В первых двух главах мы разобрали две ошибки: «синдром Dribbble» (когда дизайнер рисует картинку вместо решения) и «гипотезы вместо людей» (когда дизайнер гадает вместо того, чтобы исследовать). Ты научился проверять дизайн на пять секунд, формулировать гипотезы и проверять их на реальных пользователях.

Но есть третья ошибка. Она лежит там, где дизайнер получает задачу и делает ровно то, что сказали. Без вопросов. Без сомнений. Просто исполняет.

Эту проблему называют «слепое следование ТЗ», «дизайн по команде» или «режим исполнителя». Это третья по частоте системная ошибка в работе дизайнеров.

Приходит задача — и дизайнер сразу открывает Figma. «Сделай попап», — говорят ему, и он делает. Без вопросов. Без попытки понять, зачем это нужно. Просто открывает макет и начинает рисовать, потому что «сказали — значит, надо».

А через месяц выясняется, что попап этот никому не нужен, или убивает конверсию, или раздражает пользователей, но дизайнер сделал, что просили. Он же не виноват? Он просто исполнил ТЗ. И команда смотрит с недоумением: «Ты же дизайнер, ты должен был предложить лучшее решение».

И если предыдущие две ошибки (перегруженный визуал и гадание без исследований) ещё можно списать на неопытность, то слепое следование ТЗ — это уже про профессиональную позицию.

Сегодня рынок дизайна изменился. Всё меньше компаний ищут «дизайнеров-исполнителей» — тех, кто просто рисует по команде. Всё больше — партнёров, которые задают вопросы, предлагают альтернативы и помогают решать бизнес-задачи. В вакансиях всё чаще можно встретить формулировки вроде:

- «Уметь аргументировать свои решения»;
- «Навык работы с продуктовыми метриками»;
- «Опыт участия в стратегических обсуждениях».

Потому что слепое исполнение стоит денег. И времени. И доверия команды.

### 3.2. История первая. Максим снова ошибается

Помнишь Максима? Он уже прошёл через две ошибки — перестал рисовать неон и научился проверять гипотезы. Он даже загордился: «Всё, теперь я настоящий профессионал».

А потом получил новое задание от продакта:

— Срочно! Заказчик хочет чат-бота с ИИ на главной. Сделай красиво, похоже на ChatGPT.

Максим открывает Figma. Начинает рисовать. А внутри уже шевелится сомнение: «Чат-бот? Зачем? У нас же есть поиск и фильтры. Зачем пользователю спрашивать у бота то, что он может найти за пару кликов?»

Но он молчит. Продакт сказал «срочно» и «заказчик хочет». Значит, надо делать.

Он рисует. Отправляет. Чат-бота запускают.

Проходит месяц. Аналитика: ботом воспользовались 3% пользователей. Остальные просто не поняли, зачем он нужен. Заказчик звонит:

— Мы потратили деньги, а результат — ноль. Почему вы не предложили другое решение?

Продакт смотрит на Максима:

— Ты же дизайнер. Почему ты не сказал, что бот — плохая идея?

Максим замирает. Он сделал, что сказали. Или он должен был спросить?

### 3.3. История вторая. Алиса

Пока Максим рисовал чат-бота, Алиса получила точно такое же задание:

— Срочно! Заказчик хочет чат-бота с ИИ на главной.

Но Алиса не открыла Figma. Она открыла рабочий чат и написала продакту:

— Хорошо, я сделаю, но прежде, чем начать, давай уточним: зачем нам чат-бот? Какую задачу мы решаем?

Продакт замялся:

— Ну... заказчик хочет казаться современным. Чтобы было как у всех.

Алиса:

— Поняла. А какая у нас цель? Мы хотим увеличить продажи? Улучшить поддержку?

Сократить время поиска товара?

Продакт:

— Ну, наверное, чтобы пользователи быстрее находили товары.

Алиса:

— Отлично. А сейчас пользователи как ищут товары? Через поиск и фильтры, да?

Продакт:

— Да.

Алиса:

— Тогда давай посмотрим на данные. У нас есть аналитика по поиску и фильтрам?

Сколько времени пользователи тратят на поиск? Какие фильтры используют чаще всего?

Продакт скинул данные. Алиса посмотрела и увидела:

80% пользователей находят товар через поиск за 10–15 секунд;

фильтры используют 60% пользователей;

жалоб на сложность поиска — почти нет.

Алиса сделала вывод: пользователям не нужен чат-бот. Они и так быстро находят товары.

Она написала продакту:

— Я посмотрела данные. У нас нет проблемы с поиском — пользователи находят товары быстро. Если мы добавим чат-бота, мы потратим ресурсы на фичу, которую никто не будет использовать. Но я вижу другую проблему: 20% пользователей бросают корзину на этапе оформления заказа. Может, вместо чат-бота сделаем упрощённую форму оформления?

Продакт посмотрел на данные, подумал и сказал:

— А знаешь, ты права. Давай сделаем форму.

Проект запустили через две недели. Упрощённая форма сократила число брошенных корзин на 15%. Заказчик был доволен.

Алиса потратила на анализ данных 2 часа. Максим потратил 3 дня на дизайн чат-бота и получил переделку.

В чём разница?

|                      | Максим                                | Алиса                                         |
|----------------------|---------------------------------------|-----------------------------------------------|
| <b>Начало</b>        | Открыл Figma и начал рисовать         | Открыла данные и задала вопросы               |
| <b>Реакция на ТЗ</b> | «Сказали — делаю»                     | «Сначала пойму, а потом сделаю»               |
| <b>Результат</b>     | Чат-бот, который никто не использовал | Упрощённая форма, которая увеличила конверсию |
| <b>Время</b>         | 3 дня + переделка                     | 2 часа анализа + 2 дня дизайна                |

Алиса — профессионал, который знает, как работать с системой. Она понимает, что дизайн начинается не с Figma, а с вопросов и понимания задачи. Максим — всё ещё исполнитель. И эта разница видна в результатах.

### 3.4. Что такое «слепое следование ТЗ» и почему это опасно

Суть одна: ты делаешь то, что сказали, не пытаясь понять, зачем это нужно.

Перегруженный визуал или гадание без исследований — это ошибки, которые пройдут с опытом. Но слепое следование ТЗ — это не ошибка. Это выбор. Ты либо берёшь ответственность за результат, либо перекладываешь на того, кто написал ТЗ. Ты либо задаёшь вопросы и ищешь лучшее решение, либо просто делаешь, что сказали.

Профессиональный рост начинается в тот момент, когда ты перестаёшь быть «тем, кто рисует», и начинаешь быть «тем, кто решает». Когда ты смотришь на задачу и думаешь не «как сделать», а «зачем мы это делаем и можно ли лучше».

Как выглядит слепое следование ТЗ на практике:

Ты не задаёшь вопросов: «Сказали — делаю».

Ты не предлагаешь альтернатив: «Как в ТЗ написано — так и рисую».

Ты не знаешь бизнес-цели: «Мне дали задачу, я её выполняю».

Ты не вникаешь в контекст: «Какая разница, кто пользователь?»

Ты не защищаешь свои решения: «Мне так сказали».

Почему это опасно:

Ты становишься «исполнителем» — тебя не воспринимают как эксперта.

Ты теряешь время — делаешь то, что не нужно.

Ты теряешь доверие — команда не видит в тебе партнёра.

Ты подводишь команду — твой дизайн не решает реальную проблему.

Ты теряешь интерес к работе — рисовать по команде скучно.

В этой главе мы разберём:

почему дизайнеры боятся задавать вопросы;

как перевести «сделай попап» в осмысленную задачу;

как задавать правильные вопросы, чтобы не подвести команду;

как превратиться из исполнителя в партнёра.

## 3.5. Погружение в проблему

### 3.5.1. Почему дизайнеры боятся задавать вопросы

Причины те же, что и в прошлой главе — человеческие, понятные, но опасные.

#### Причина №1. «Я не хочу казаться глупым»

Новичок боится, что, если он задаст вопрос «А зачем мы это делаем?», то его сочтут некомпетентным. Вдруг это очевидно для всех, а я один не понимаю? Лучше промолчу и нарисую.

Как с этим справиться:

Задавать вопросы — это не слабость. Это профессионализм. Тот, кто задаёт вопросы, выглядит как человек, который вникает в суть. Тот, кто молча рисует ерунду, выглядит как исполнитель, которому всё равно.

Попробуй переформулировать свой страх. Вместо «Я боюсь показаться глупым» скажи себе: «Я хочу понять суть задачи». Вместо «Они подумают, что я не знаю» скажи: «Они увидят, что я хочу сделать хорошо».

Запомни: если ты не задаёшь вопросов — ты не управляешь процессом. Ты просто плывёшь по течению. А потом оказывается, что ты нарисовал не то.

#### Причина №2. «Мне дали ТЗ — я делаю»

Новичок воспринимает ТЗ как «закон». Так написал менеджер/продакт/заказчик — значит, так надо. Спорить нельзя, а то уволят.

Как с этим справиться:

ТЗ — это не закон. Это гипотеза. Это попытка решить проблему. И твоя задача — помочь найти лучшее решение, а не просто нарисовать то, что написано.

Попробуй воспринимать ТЗ как «входные данные», а не как «приказ». Спроси себя: «Какую бизнес-задачу мы решаем?», «Какая у нас цель?», «Что мы хотим изменить?». А потом уже думай, как это сделать лучше.

Ты — эксперт. Ты знаешь, как делать правильно. Твоё мнение ценно. Не бойся его предлагать.

#### Причина №3. «У меня нет времени»

Дедлайн через два дня, а я буду задавать вопросы и тормозить процесс. Лучше быстрее нарисую и сдам.

Как с этим справиться:

Один вопрос «Зачем?» занимает 1 минуту. Одна альтернатива — 10 минут. Переделка — дни, а иногда и недели.

пять минут на уточнение → экономит пять дней переделок.

десять минут на альтернативу → экономит неделю споров с командой.

один вопрос «А что, если?» → спасает от провала в аналитике.

Время на вопросы — это не потеря времени. Это инвестиция в то, чтобы не делать лишнюю работу.

#### Причина №4. «Я не знаю, какие вопросы задавать»

Это самая честная причина. Новичок не знает, с какой стороны подойти к задаче. Он просто не умеет задавать правильные вопросы.

Как с этим справиться:

Этому можно научиться. И мы сейчас научимся.

Начни с простого. Задавай 5W-вопросы (их мы разберём в следующем разделе):

Что? — Что мы делаем?

Зачем? — Какую проблему решаем?

Кто? — Для кого?

Когда? — В какой момент?

Где? — На каких устройствах?

Эти пять вопросов покрывают 80% того, что нужно знать, чтобы не нарисовать ерунду.

И ещё один совет: если ты не знаешь, что спросить — спроси про цель. «Какая у нас цель?», «Что мы хотим изменить?», «Как мы поймём, что задача решена?». Цель — это всегда самое важное.

## 3.6. Решение и инструменты

### 3.6.1. Шаг 1. Задавай вопрос «Зачем?» (5W-вопросов)

Самый простой способ перестать слепо следовать ТЗ — задавать правильные вопросы. Я использую классическую схему 5W-вопросов:

**What (Что?)**— Что мы делаем? Что это за задача?

**Why (Зачем?)**— Зачем мы это делаем? Какую проблему решаем?

**Who (Кто?)**— Для кого мы это делаем? Кто наши пользователи?

**When (Когда?)**— Когда это будет использоваться? В какой момент?

**Where (Где?)**— Где это будет работать? На каких устройствах?

Пример. «Сделай попап с подпиской»

Разбираем её по пяти вопросам:

**Что мы делаем?**— Попап с подпиской на новости.

Это задача, но не цель.

**Зачем мы это делаем?**— Чтобы собрать базу подписчиков.

Это реальная цель, которая стоит за задачей.

**Для кого мы это делаем?**— Для новых пользователей, которые зашли на сайт впервые.

Это наша целевая аудитория.

**Когда это будет использоваться?**— Сразу после загрузки главной страницы.

Это момент, когда пользователь видит интерфейс.

**Где это будет работать?**— На десктопе и на мобильных устройствах.

Это технические условия использования.

Теперь задача выглядит иначе: не «нарисуй попап», а «реши задачу по сбору базы подписчиков среди новых пользователей при входе на сайт, учитывая все устройства».

### 3.6.2. Шаг 2. Переводи ТЗ в задачу

Простой алгоритм, который я использую:

Прочитай ТЗ.

Задай вопрос: «Какую проблему мы решаем этим функционалом?»

Переформулируй ТЗ в задачу: «Мы хотим [решить проблему] для [пользователей]».

Предложи 2–3 способа решения (не только тот, что в ТЗ).

Пример:

Было: «Нарисуй попап с подпиской».

Стало: «Собрать базу подписчиков для новых пользователей на главной странице. Варианты решения: попап, баннер, встроенная форма, текстовый блок с кнопкой в хедере».

Теперь ты — дизайнер, который предлагает решения, а не просто «рисует по команде».

### 3.6.3. Шаг 3. Предлагай альтернативы

Самый сильный ход, который показывает, что ты эксперт, — предлагать альтернативы.

Алгоритм:

Нарисуй вариант из ТЗ («как просили»).

Нарисуй 1–2 альтернативных наброска («а что, если сделать иначе?»).

Сравни их по критериям: какое решение быстрее в реализации, какое лучше для пользователя, какое лучше для бизнеса.

Аргументированно предложи лучший вариант.

Пример для попапа:

| Вариант                           | Плюсы                         | Минусы                                            |
|-----------------------------------|-------------------------------|---------------------------------------------------|
| Попап                             | Заметный, привлекает внимание | Раздражает пользователей, высокий уровень отказов |
| Баннер на главной                 | Менее навязчивый              | Меньше конверсия, чем у попапа                    |
| Встроенная форма в футере         | Совсем не навязчивая          | Низкая конверсия                                  |
| Текстовый блок с кнопкой в хедере | Виден всегда, не навязчив     | Требуется тестирование                            |

Твой аргументированный выбор:

«Я предлагаю сделать баннер на главной, потому что он менее навязчив, чем попап, и даёт пользователю время ознакомиться с контентом. Попап я тоже нарисовал, но он может снизить конверсию на 30% по нашим данным».

#### 3.6.4. Шаг 4. Как отстаивать своё решение перед командой

Этот навык приходит с опытом, но есть несколько проверенных приёмов.

**Приём №1.** «Давайте проверим гипотезу»

Вместо того чтобы спорить, предложи проверить.

«Я предлагаю сделать баннер вместо попапа. Давайте запустим А/В-тест: половине пользователей покажем попап, половине — баннер. Через неделю посмотрим, где выше конверсия и ниже отказы. Если баннер проиграет — мы всегда можем вернуться к попапу».

Это снимает напряжение и делает тебя партнёром, а не спорщиком.

**Приём №2.** «Давайте посмотрим на данные»

Если у тебя есть данные — ты неуязвим.

«По данным наших аналитиков, 70% пользователей закрывают попап, не читая. Баннер на главной даёт больше вовлечения. Поэтому я предлагаю его».

**Приём №3.** «Давайте спросим у пользователей»

Если данных нет — проведи мини-исследование.

«Давайте спросим у пяти пользователей, что им больше нравится — попап или баннер. Это займёт пятнадцать минут, а мы примем решение на основе фактов, а не догадок».

#### 3.6.5. Шаг 5. Скрипт для диалога с заказчиком

Вот готовый скрипт, который я рекомендую использовать, когда получаешь ТЗ (техническое задание), с которым не согласен.

Пример диалога:

Заказчик: «Нарисуй попап для подписки на главной».

Ты: «Хорошо, я нарисую, но, прежде чем начать, давай уточним: зачем нам попап? Какую задачу мы решаем?»

Заказчик: «Мы хотим собрать базу подписчиков».

Ты: «Понял. А для кого мы это делаем? Какие пользователи заходят на главную?»

Заказчик: «Новые пользователи, которые впервые на сайте».

Ты: «Я вижу два решения. Первое — попап, как ты просил. Второе — баннер на главной, который не перекрывает контент и даёт пользователю время ознакомиться с сайтом. Давай я нарисую оба варианта, и мы выберем лучший. Или, если хочешь, мы можем провести А/В-тест и посмотреть, что работает лучше. Что ты думаешь?»

Заказчик: «Звучит разумно. Давай оба варианта».

Разница заметна? Ты не просто «рисуй попап» — ты стал частью процесса принятия решения. Ты — партнёр, а не исполнитель.

### **3.6.6. Шпаргалка: что спросить у заказчика перед тем, как начать**

Держи эту памятку под рукой — она экономит часы.

**Какую проблему мы решаем?** Без этого ты рисуешь фиши, а не решения.

**Кто наша целевая аудитория?** Без этого ты рисуешь для себя, а не для пользователей.

**Какие у нас есть данные?** Опирайтесь на факты лучше, чем на догадки.

**Какие есть ограничения?** По времени, бюджету, технологиям.

**Как мы будем измерять успех?** Метрики — твоя защита и доказательство.

### **3.6.7. Шаг 6. Что делать, если заказчик настаивает на своём**

Иногда заказчик категоричен. Он сказал «попап» — и будет попап.

Что делать в таком случае:

Сделай как он просит.

Но добавь параллельный тест: «Я нарисую попап, как ты просишь. Но давай параллельно протестируем альтернативный вариант — баннер. Через неделю сравним результаты».

Если тест невозможен — сделай как просят, но в портфолио этот проект можешь не показывать.

Пример Алисы в такой ситуации:

Заказчик настаивал на чат-боте на главной странице, чтобы собирать контакты пользователей. Алиса сказала:

— Я сделаю чат-бота, как ты просишь. Но давай посмотрим на это шире: наша общая цель — собирать контакты пользователей. Чат-бот — один из способов, но он не всегда работает, потому что пользователи часто уходят, не оставив контактов. Поэтому я предлагаю добавить ещё один канал — форму подписки, встроенную в корзину. Она появится в момент, когда пользователь уже выбрал товары. Это повысит шансы на сбор контакта, независимо от того, воспользуется ли пользователь чат-ботом. Мы не противопоставляем решения — мы дополняем их.

Заказчик согласился. Чат-бота запустили, он показал низкие цифры. Через месяц запустили форму в корзине — и она собрала в 3 раза больше контактов, потому что пользователи уже были вовлечены в процесс покупки.

Алиса не спорила, но показала альтернативу, которая решала ту же бизнес-задачу. И в итоге оказалась права.

### 3.7. Итог: алгоритм действий

Вот чёткая последовательность шагов, которая поможет тебе перестать быть исполнителем и стать партнёром:

Прочитай ТЗ.

Задай вопрос «Зачем?» — какую проблему мы решаем?

Переформулируй ТЗ в задачу — «Мы хотим решить проблему X для пользователей Y».

Предложи 2–3 альтернативы — не только ту, что в ТЗ.

Аргументированно предложи лучший вариант — с данными, если они есть.

Если заказчик настаивает — сделай как просит, но предложи параллельный тест.

## 3.8. Резюме

ТЗ — не закон. Это гипотеза. Твоя задача — проверить её и предложить лучшее решение.

Задавай вопрос «Зачем?». Это превращает тебя из исполнителя в партнёра.

Предлагай альтернативы. Покажи, что ты думаешь, а не просто рисуешь.

Умей отстаивать своё решение. Используй данные, гипотезы и A/B-тесты.

Ты — дизайнер-стратег, а не «рисовальщик». И команда относится к тебе соответственно.

### 3.9. Практическое задание

Возьми свой текущий проект (или любой старый макет) и сделай три вещи.

**Задание 1.** Проанализируй ТЗ

Если у тебя есть ТЗ — выпиши из него все требования. Теперь задай по каждому пункту вопрос «Зачем?». Что ты узнал? Запиши свои выводы.

**Задание 2.** Предложи альтернативу

Для одного из пунктов ТЗ предложи 2 альтернативных решения. Опиши, почему они лучше или хуже. Запиши свои аргументы.

**Задание 3.** Разыграй диалог

Представь, что заказчик просит тебя сделать что-то, с чем ты не согласен. Напиши свою реплику по скрипту из этой главы. Запиши, что бы ты сказал и как бы аргументировал свою позицию.

### 3.10. Чек-лист для самопроверки

- ☐ Я задал вопрос «Зачем?» и понял, какую проблему мы решаем.
- ☐ Я знаю, кто моя целевая аудитория.
- ☐ Я знаю, какие данные у нас есть по этой задаче.
- ☐ Я предложил альтернативы, а не просто сделал «как в ТЗ».
- ☐ Я могу аргументировать, почему выбрал именно это решение.
- ☐ Я готов защищать своё решение перед командой.

## 3.11. FAQ

— А если заказчик не хочет слушать альтернативы и просто говорит «делай как я сказал»?

Если заказчик категоричен — сделай как он сказал. Но добавь к своему ответу фразу: «Я нарисую, как ты просишь, но параллельно предлагаю протестировать и альтернативный вариант. Через неделю мы сравним результаты». Иногда заказчики соглашаются на параллельный тест, потому что это безопасно.

— А если у меня нет времени на вопросы и альтернативы?

Один вопрос «Зачем?» занимает одну минуту. Альтернатива — десять минут. Переделка — день. Считай сам.

— А если я задаю вопросы, а коллеги раздражаются, что я торможу процесс?

Если коллеги раздражаются — значит, они не привыкли к такому подходу. Объясни им: «Я задаю вопросы, чтобы мы сделали правильно с первого раза. Это экономит нам время и нервы». Обычно это снимает раздражение.

**Но здесь важно чувствовать баланс.** Вопросы должны помогать процессу, а не останавливать его. Если ты задаёшь их слишком много или слишком глубоко копаешь в очевидных вещах — это начинает раздражать. Задавайся вопросами, когда видишь реальный риск ошибки. И помни: иногда лучший способ проверить гипотезу — не бесконечно спрашивать, а сделать простой прототип и показать команде. Уважай время коллег и дай им возможность увидеть решение, а не только слышать вопросы.

— А что, если я не знаю, какую альтернативу предложить?

Используй ChatGPT или почитай, как делают у конкурентов. Или спроси у продакта: «А какие ещё варианты ты видишь?». Обычно они сами подсказывают, если их спросить.

— А если я предложил альтернативу, но команда всё равно выбрала вариант из ТЗ? Как быть, если я знаю, что это неправильно?

Ты сделал свою работу — предложил альтернативу и аргументировал. Если команда выбрала другой путь, это её ответственность. Твоя задача — не продавливать свою точку зрения любой ценой, а **быть услышанным**.

В большинстве случаев лучшее, что можно сделать — это предложить параллельный тест: «Давайте запустим оба варианта и посмотрим, что покажут данные». Это снимает напряжение и переводит спор в конструктивное русло. Если тест невозможен, а команда всё равно настаивает на своём — прими решение большинства.

Запись несогласия в протоколе имеет смысл только в двух случаях:

Когда цена ошибки критически высока (например, юридические или финансовые риски).

Когда это входит в твои прямые обязанности (например, если ты архитектор решения или тимлид).

В остальных случаях главное — ты не промолчал. Ты предложил альтернативу, аргументировал, предложил проверить. Дальше — доверяй команде. Иногда они видят контекст, который не виден тебе.

### 3.12. Связка с другими материалами

Мы разобрали третью ошибку. Теперь ты знаешь, как не быть слепым исполнителем, как задавать вопрос «Зачем?» и предлагать альтернативы. Ты научился смотреть на ТЗ не как на приказ, а как на гипотезу, которую можно и нужно проверять.

Максим тоже научился. Он уже не тот новичок, который боялся задавать вопросы и просто рисовал, то, что скажут. Теперь он интересуется целью и задачами, предлагает альтернативы, когда видит, что решение не работает. Он горд собой. И правильно — это большой шаг.

Но есть ещё одна проблема.

Он всё ещё слишком много рисует. Слишком детально. Слишком продуманно. Слишком идеально.

Получив задачу, Максим сразу начинает прорабатывать все состояния, все экраны, все микроинтеракции. Он хочет сделать идеально с первого раза. Он строит хрустальный дворец, хотя заказчику нужен просто дом. Он тратит недели на то, что можно проверить за пару часов.

И это — четвёртая ошибка. Она называется «Замок вместо MVP» (или «дворец вместо дома», как мы будем её называть в этой книге). Это про перфекционизм, который убивает продукты. Про страх показать сырой прототип. Про неумение вовремя остановиться и спросить: «А что, если мы проверим гипотезу на самом простом решении?»

В следующей главе мы разберём, как перестать строить дворцы и начать с простого. Как понять, что такое MVP и почему он спасает продукты. И как убедить команду, что «просто» — это нормально.

А пока — подумай: не строишь ли ты дворец там, где достаточно дома?

## ГЛАВА 4. ДВОРЕЦ ВМЕСТО ДОМА: КАК ПЕРФЕКЦИОНИЗМ УБИВАЕТ ПРОДУКТЫ

### 4.1. Вступление

В первых главах мы разобрали три ошибки: «синдром Dribbble» (когда дизайнер рисует картинку вместо решения), «гипотезы вместо людей» (когда дизайнер гадает вместо того, чтобы исследовать) и «слепое следование ТЗ» (когда дизайнер просто исполняет, не задавая вопросов). Ты научился проверять дизайн за пять секунд, формулировать гипотезы и превращать ТЗ в осмысленные задачи.

Но есть четвёртая ошибка. Она там, где дизайнер начинает рисовать слишком много. Где он хочет сделать идеально с первого раза. Где он строит хрустальный дворец, хотя достаточно было бы простого дома.

Это явление называют «дворцом вместо дома», «синдромом перфекциониста» или «over-engineering в дизайне». Это четвёртая по частоте системная ошибка в работе дизайнеров.

«Я хочу сделать идеально. С первого раза. Чтобы никто не мог придраться».

Этот внутренний голос знаком каждому дизайнеру. Ты получаешь задачу и сразу начинаешь прорабатывать всё: все состояния, анимации, адаптив, микроинтеракции. Ты работаешь неделями, доводя каждую иконку до совершенства.

А потом выясняется, что гипотеза не работает. Или заказчик передумал. Или разработчики говорят, что это слишком сложно в реализации.

Первые две ошибки — «синдром Dribb

## **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.