Типовые ошибки в дизайне интерфейсов при переходе от “рисования” к реальной UI/UX-разработке

(и как их избежать)

Когда дизайнер начинает работать не просто над красивыми картинками, а над настоящими интерфейсами (мини-аппы, формы, продукты, панели, CRM), всплывают типичные ошибки. Это не потому, что дизайнер “плохой”, а потому что **UI — это инженерная дисциплина, а не художественная**.

Вот ключевые ошибки, которые совершают 90% начинающих дизайнеров — и рекомендации, как это исправить.

Ошибка 1. “Картинка вместо системы”

Дизайнер делает интерфейс как постер в Photoshop:

- ручные отступы,
- элементы расставлены “на глаз”,
- нет структуры, нет вложенности,
- всё в одном слое,
- блоки не связаны логически.

Почему это плохо

Такой дизайн невозможно:

- адаптировать под разные экраны,
- передавать фронтендеру,
- масштабировать,
- переиспользовать.

Как правильно

UI — не картинка, а **система элементов**:

- каждый блок — контейнер,
- внутри контейнера — логическая группа,
- всё должно быть в единой архитектуре,
- каждый элемент имеет своё место.

✔ Совет:

Представляй интерфейс не как рисунок, а как **скелет + мышцы + кожа**.
Сначала структура (скелет), потом функциональные группы (мышцы), потом визуал (кожа).

Ошибка 2. Злоупотребление автолэйаутами (“20 контейнеров ради контейнеров”)

Новичок услышал “автолэйаут — это база” и начинает:

- вкладывать блоки друг в друга без необходимости,
- делать 10 уровней вложенности,
- слепо wrap'ить всё, что видит,
- использовать контейнеры не по логике, а “потому что надо”.

Почему это плохо

- структура становится нечитаемой,
- фронтендеру страшно открывать макет,
- невозможно понять, что адаптивно, а что статично,
- всё “живёт собственной жизнью”.

Как правильно

Автолэйаут — не цель, а **инструмент**.

Использовать его нужно:

- когда элементы должны жить как группа,
- когда блок должен масштабироваться,
- когда есть логическая вертикаль/горизонталь,

- когда UI строится как компонент.

✔ Совет:

Если элемент не должен тянуться → **не делай его резиновым**.
Если элемент не является логической группой → **не кладите его в контейнер**.

Ошибка 3. Непонимание паттернов адаптивности

Новички делают:

- фиксированные ширины,
- вручную выставленные позиции,
- нулевые минимальные и максимальные размеры,
- текст, который не помещается при растяжении,
- колонки, которые ломаются.

Почему это плохо

В реальном интерфейсе:

- экраны бывают 320, 375, 414, 768, 1024, 1440, 1920…
- UI должен жить на любом разрешении.

Как правильно

Задавать:

- min-width / max-width,
- фиксированные или резиновые контейнеры по назначению,
- ограничение ширины текста,
- грамотную поддержку “узкого”, “среднего” и “широкого” вида.

✔ Совет:

Тестируй интерфейс в Figma: сожми фрейм → растяни → проверь, как ведут себя элементы.

❌ Ошибка 4. Отсутствие компонентного мышления

Новички делают:

- каждый блок уникальным,
- копируют элементы вручную,
- вносят изменения в 15 мест одновременно,
- не собирают UI KIT,
- не используют компоненты.

Почему это плохо

- правки умножаются на количество копий,
- дизайн не масштабируется,
- фронт получает 100 вариаций одной и той же кнопки.

Как правильно

Мыслить **компонентами**:

- кнопка
- поле
- карточка
- модалка
- теги
- переключатели
- и т.д.

Все вариации — в **Variants**.

✔ Совет:

Проектируй интерфейс так, будто его собирают в React — *из готовых компонентов*.

❌ Ошибка 5. Непонимание типографики

Типовые проблемы новичков:

- 7–10 разных размеров текста, без логики,

- смесь Regular / Medium / Bold в одном абзаце,
- отсутствие иерархии заголовков,
- большие расстояния между строками,
- неправильный контраст.

Как правильно

- 3–4 уровня заголовков,
- 1–2 вида параграфного текста,
- единая иерархия,
- строгая логика жирностей,
- контраст согласно WCAG (или хотя бы визуальному здравому смыслу).

✔ Совет:

Хорошая типографика = 70% ощущение “дорогого” интерфейса.

Ошибка 6. Цвет как “интуиция”, а не система

Новички:

- используют 10 оттенков синего,
- берут цвета “на глаз”,
- игнорируют брендбук,
- смешивают пастель с кислотой,
- нарушают контраст.

Правильно

- палитра: основной / вторичный / акцент / фон / текст
- строгое использование,
- никакой самодеятельности.

✔ Совет:

Цвет = язык.
Его нельзя менять “на вкус”.

❌Ошибка 7. Ручные отступы vs система отступов

Новичок ставит:

- 12px тут,
- 14px там,
- 17px снизу,
- 9px сверху.

И сам путается, фронтендер путается, всё плавает.

Правильно

Использовать единую сетку:
4pt / 8pt / 16pt — и никаких случайных чисел.

✔ Совет:

Если отступ нельзя объяснить — он неправильный.

❌Ошибка 8. Нет связи дизайнер ↔ фронтендер

Типично:

- дизайнер думает “оно понятно”,
- фронт открывает макет и видит хаос,
- дизайнер не знает, что верстать сложно,
- фронт не знает, что дизайнер хотел.

Правильно

2 мини-связки:

1️⃣ Передача макета →
дизайнер делает walkthrough:
“Вот компоненты, вот группы, вот отступы, вот логика адаптива.”

2️⃣ Перед стартом верстки →
фронтендер задаёт вопросы.

✔ Совет:

UI — командная работа, не сольная.

❌ Ошибка 9. “На глазок” вместо UX

Новички принимают визуальные решения без понимания:

- зачем элемент нужен,
- какой сценарий он поддерживает,
- какие у пользователя задачи,
- что важно, что вторично.

Правильно

Каждому элементу нужен ответ:

- какую проблему он решает?
- зачем он здесь?
- что пользователь должен сделать?
- можно ли убрать это без потери смысла?

✔ Совет:

UX начинается не в Figma, а в голове.

❌ Ошибка 10. Непонимание важности прототипа

Новички сразу рисуют дизайн, пропуская этап каркаса:
несогласованная логика, невидимые ошибки, “симпатичная каша”.

Правильно

Сначала прототип (черно-белый), потом UI.

✔ Совет:

Прототип — 80% успеха.

Резюме: что нужно, чтобы перестать допускать эти ошибки

✔ 1. Автолэйаут — да, но с головой

Не везде и не “ради галочки”.

✔ 2. Компонентное мышление

Думай как React-разработчик: системой.

✔ 3. Сетка и типографика

Строгая дисциплина, минимум хаоса.

✔ 4. Прототип → дизайн → верстка

Без прыжков через этапы.

✔ 5. Обратная связь от фронтендера

Каждый макет должен пройти “технический осмотр”.

✔ 6. Менее “рисовать”, больше “проектировать”

UI — это инженерия.

✔ 7. Постепенно: не революция, а эволюция

Каждый проект — улучшение на 5–10%.

Версия #2
Artem Garashko создал 24 November 2025 12:33:46
Artem Garashko обновил 25 November 2025 21:03:02