Как понимать контекст проекта

Этот материал объясняет:

- разницу типов проектов,
- что такое mission critical / non-mission-critical,
- когда дизайнер должен общаться с заказчиком,
- а когда избыток коммуникаций — во вред,
- как отличается клиентский опыт в продуктах и в мини-апах,
- как всё это классифицировать,
- как команда может ориентироваться в этих режимах и не путать одно с другим.

□□ Два мира разработки: продукты vs мини-аппы. Как отличать, как работать и какие процессы нужны

Мы сталкиваемся с разными типами проектов. Где-то мы делаем полноценный продукт, который живёт годами, влияет на бизнес-процессы и проходит десятки согласований. А где-то — небольшой мини-апп, простой инструмент или небольшую коммуникационную вставку внутри чат-бота.

Чтобы команда работала эффективно и не пыталась применить “тяжёлые” процессы там, где они только мешают, важно понимать фундаментальную разницу между **двумя классами проектов**:

- **Полноценные продукты / Mission Critical**
- **Мини-аппы и поддерживающие интерфейсы / Non-Mission-Critical**

Это два разных мира, требующих двух разных подходов — в дизайне, разработке, коммуникациях и управлении ожиданиями.

□□ 1. Что такое Mission Critical, и почему эти проекты другие

Mission Critical — это продукты, без которых бизнес не сможет выполнять свою основную функцию.

Примеры:

- Яндекс.Такси — приложение не работает → такси не существует
- Интернет-банк — ошибка → бизнес клиента остановился
- Личный кабинет госуслуг → критичен для доступа к услугам
- Управляющая панель в промышленности (логистика, производство, расчёты)

У промышленной компании, например фармацевтической, mission critical — это производство таблеток, логистика, ERP, склад, документоборот.

Чат-боты, мини-формы, калькуляторы, маркетинговые инструменты — не mission critical.
Их можно улучшать, допиливать, полировать, но они не остановят бизнес.

Отличительные признаки Mission Critical:

- UI = бизнес-операция, ошибка стоит дорого
- требуется глубокая выверенная архитектура
- десятки согласований
- UX-исследования
- строгая типографика, модульные сетки, дизайн-системы
- дизайнер обязан общаться напрямую с заказчиком
- множество рабочих встреч
- нужно понимать роли, сценарии, ограничения
- всё должно быть документировано
- много уровней ответственности

Там нет “примерно нормально” — там только **идеально и надёжно**.

□□ 2. Что такое Non-Mission-Critical (наши текущие мини-аппы)

Мини-аппы, калькуляторы, формы, промо-страницы, быстрые B2B-калькуляторы в мессенджерах — это **дополняющие** интерфейсы.

Они помогают:

- сократить путь пользователя,
- поддержать маркетинг,
- объяснить продукт,
- загрузить данные,
- собрать обратную связь.

Но **не являются ядром бизнеса**.

Отличительные признаки таких проектов:

- одна или две встречи достаточно
- можно опираться на вкус и здравый смысл
- нет 50 согласований
- нет десятков UX-сценариев
- проще требования к цветам, сетке, блокам
- допускаются разумные компромиссы
- UI — не синоним бизнес-операции, а инструмент коммуникации

Это не значит “делать плохо”, это значит **не переусложнять там, где это не нужно**.

□□ 3. Почему процессы должны быть разными

Если на non-mission-critical проекты натянуть процессы от больших продуктовых студий, произойдёт следующее:

- сроки вырастут x3–5
- стоимость выйдет за рамки бюджета
- ценность UI станет ниже цены его производства
- клиент удивится, почему такие простые задачи делаются так дорого
- команда будет перегружена ненужными встречами
- дизайнеры будут ловить лишние правки и вкусовщину
- фронтендеры погрязнут в овердизайне

Но если на mission critical проект натянуть подход “как для мини-аппа” — бизнес словит катастрофу.

Поэтому нужен **двухрежимный подход**.

□□ 4. Должен ли дизайнер встречаться с заказчиком? Да — но не всегда.

Два типа проектов = два типа участия дизайнера.

□□ Тип 1. Полноценные продукты — дизайнер участвует активно

Тут дизайнер — не художник и не исполнитель.
Он — **аналитик**, **переговорщик**, **продуктовый партнёр**.

Его обязанности:

- собирать требования
- уточнять задачи
- выяснять сценарии
- согласовывать логику
- прототипировать
- решать UX-проблемы
- участвовать в созвонах
- защищать решения с цифрами и логикой

Без прямой коммуникации тут работать невозможно.

□□ Тип 2. Мини-аппы — дизайнеру достаточно 1-2 встречи

В этих проектах:

- нет сложной логики,
- нет рискованных пользовательских сценариев,
- нет десятков ролей и use-case'ов,
- нет mission critical-функций,
- интерфейс не решает судьбу компании.

Поэтому дизайнеру достаточно:

- 1 вводной встречи,

- 1 обсуждения прототипа,
- остальная коммуникация — через менеджера.

Прямой контакт с клиентом после этого **ничего не улучшит, а часто только ухудшит**:

- начинается вкусовщина,
- “попробуйте зелёный”,
- “давайте кнопку побольше”,
- “а сделайте фон потемнее”,
- дизайнер теряет фокус,
- дизайн разваливается,
- сроки растут.

Дизайнер защищает эстетику и логику →
Менеджер фильтрует запросы →
Команда работает спокойно.

Это правильный процесс.

□□ 5. Разные типы клиентского опыта (CX): “деловой” vs “приятный”

Чтобы правильно строить интерфейсы, важно понимать разницу между двумя видами клиентского опыта.

□□ Тип 1. “Деловой” клиентский опыт (B2B, сервисный, утилитарный)

Цель: **дать результат быстро, чётко и без лишних действий.**

Примеры:

- B2B калькуляторы
- формы заказа
- заявки
- рабочие панели
- кабинки операторов
- функциональные мини-аппы

Тут интерфейс должен быть:

- быстрым

- простым
- прямым
- без флейвора
- без украшательств
- минимум шагов
- минимум когнитивной нагрузки

Это “инструмент”, а не “впечатление”.

□□ Тип 2. “Приятный” клиентский опыт (развлечения / эмоции / smacking experience)

Цель: **вовлечь, развлечь, удержать внимание.**

Примеры:

- рестораны, еда, сервис, доставка
- игры
- развлекательные проекты
- брендинг
- storytelling-интерфейсы

Здесь можно:

- растягивать путь,
- играть с эмоцией,
- добавлять анимации,
- давать “вкус” и “медленность”,
- делать journey красивым.

□□ 6. Как классифицировать проект за 1 минуту

Задаём два вопроса:

□ Если интерфейс сломается — пострадает ли основная миссия бизнеса?

Если **да** → это mission critical → полноценный продуктовый процесс.
Если **нет** → это мини-апп → упрощённый процесс.

□□ 7. Что важно

1. **Не надо натягивать продуктовые процессы на мини-аппы.**
   Это убивает скорость и ценность проекта.
2. **Не надо упрощать миссион-критичные проекты.**
   Там нужна глубина.
3. **Дизайнер не должен общаться с заказчиком там, где это во вред.**
   Коммуникация — ресурс, его надо дозировать.
4. **Нужно уметь классифицировать тип задачи с первых минут проекта.**
5. **Наша цель — гибкость.**
   Мы должны одинаково уверенно вести оба типа проектов.

□□ 8. Тонкое различие, которое важно осознать

Мы не “студия, которая делает всё одинаково”.
Мы команда, которая умеет **менять подход под контекст**.

Где-то нужен продуктовый UX, десятки встреч и прототипы.
Где-то достаточно одного звонка, одного прототипа и менеджерского фильтра.

Зрелость = умение различать.

Версия #1
Artem Garashko создал 2025-11-24 13:13:53 UTC
Artem Garashko обновил 2025-11-25 21:03:02 UTC