

# Всё из ничего



Surf Summer School '18

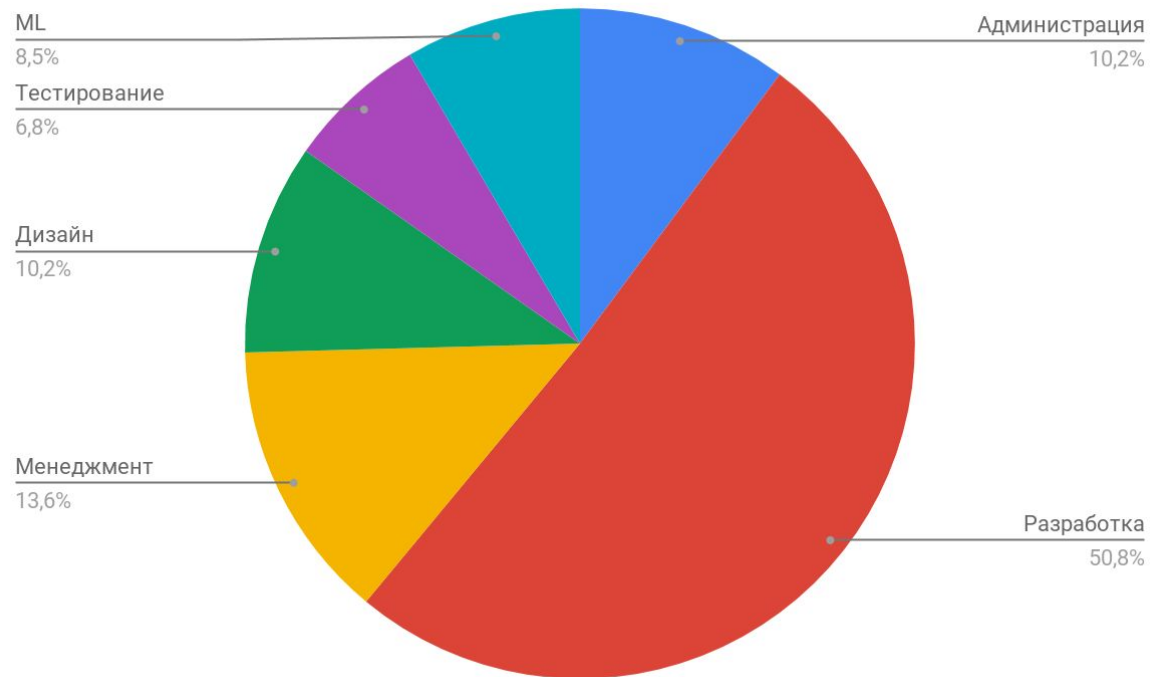
***“- Сколько нужно хипстеров,  
чтобы зарелизить одно  
приложение?”***

**Целая студия.**

**59 человек.**

**8 городов**

**2 страны**



# Административный отдел

Директор, продавники, пиарщики - эти люди обеспечивают работой всю остальную часть студии.

Кадровая служба, офисный менеджмент и системные администраторы - служат всеобщему благополучию, комфорту и процветанию студии.

# Отдел менеджмента

Вся организация рабочего процесса команды, общение с заказчиком, взаимодействие между отделами в рамках реализации проекта, бумажная работа, сметы и составление документации - всё на них.

Зануды и бюрократы, без которых проект бы напоминал велосипед без колёс, зато с кондиционером и реактивным двигателем. При том, что заказчик изначально просил нас построить трёхэтажный дом.



# Отдел дизайна

Стильные ребята, без которых наши приложения были бы скучными и однотипными.

**Выработать общий стиль и визуальную концепцию, нарисовать красивый дизайн, адаптировать под все возможные типы устройств - это ещё не всё. Хороший дизайнер - прежде всего системный архитектор и проектировщик UX.** Из разрозненных мыслей заказчика он собирает продукт, от удобства которого пользователь не сможет отказаться.

# Отдел разработки

Главная задача этих ребят - написать такой код, за который им будет не стыдно. Ни перед начальником, ни перед собой же через два года.

Уметь **мыслить абстрактно**, тренировать **глобальное видение проекта**, **шустро разбираться в чужом коде** и **выявлять главное**, не гнушаться **копаться в глубоких исходниках** и **не бояться сказать “я не знаю, мне надо разобраться”** - всё это должен уметь хороший разработчик.

# Отдел тестирования

В проекте нет дефектов? Значит вы его ещё не начинали писать.

Или их пока не нашли ребята из отдела тестирования. Их работа состоит в том - чтобы в проекте появились дефекты. Но главное, чтобы эти дефекты не попали в релизную сборку.

# Отдел ML

Мы мало что знаем об этих ребятах.

Но если вам надо предсказать колебания биткоинов, посчитать морских котиков на фотографии или выработать наиболее прибыльную бизнес-модель - вам сюда.

***“- Откуда берутся заказы?”***

# Источники заказов

- Тендеры;
- Сарафанное радио;
- Рейтинги студий и входящие с них;
- Постоянные клиенты.

***“- Какие заказы особенно перспективны?”***

***“- За которыми стоит большой бизнес.”***



# Перспективные сферы

- Банкинг;
- E-commerce;
- Крупные социальные продукты;
- Госструктуры;
- Зарубежные стартапы;
- Российские крупные стартапы.

***“У заказчика есть проблема и  
деньги.***

***У нас есть мозги, опыт и время.”***

***“- По рукам! Что дальше?”***

# Развитие идеи

- Формулировка проблемы;
- Выявление целевой аудитории;
- Описание основных сценариев использования;
- Мозговой штурм;
- Формулировка концепции;
- Отрисовка прототипа.

# Ч/б прототип в Invision App



# Взаимодействие с заказчиком

Идёт на всём протяжении проекта. Но на старте проекта особенно интенсивно.

Заказчик - хозяин проекта. Он может ошибаться, а задача команды профессионалов сделать так, чтобы он сам признал свою ошибку.

# Написание ТЗ

ТЗ - это много букв о том, что представляет собой проект для пользователя.

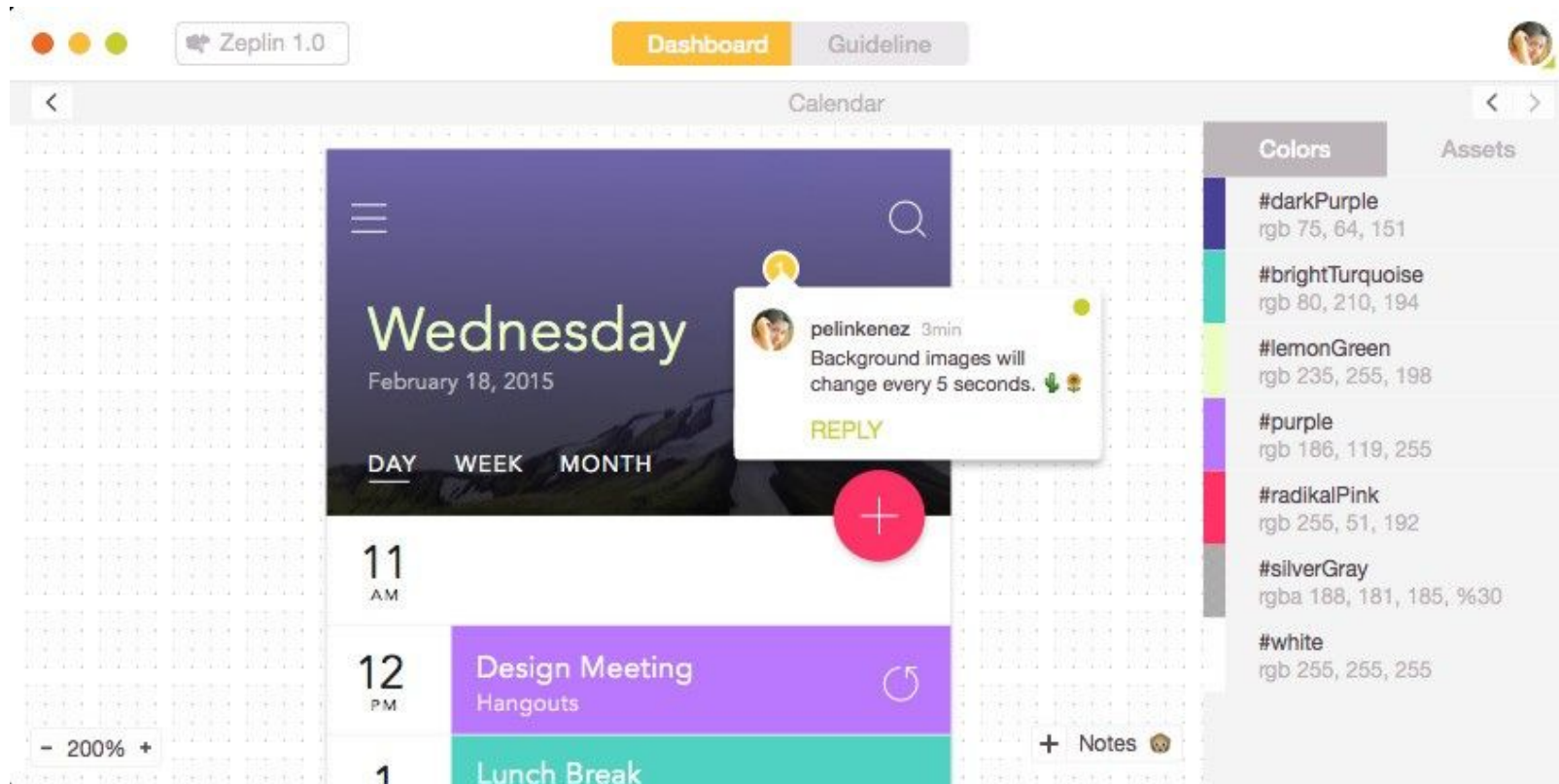
Какие будут экраны, какова логика переходов между ними, какие данные на них будут отображаться, а какие - вводятся пользователем - всё описано в этом документе.

# Отрисовка дизайна

Итоговый результат должен до каждого пикселя совпадать с отрисованными макетами. Макеты, в свою очередь, должны покрывать все возможные состояния экранов.



# Отрисовка дизайна



# Проектирование API

Клиент и сервер должны разговаривать на одном языке. Добиться этого помогает исчерпывающая документация.

# Проектирование API

## Транзакции

### История транзакций пользователя

#### История транзакций пользователя

Требуется авторизация.

С пагинацией.

Метод используется для получения истории транзакций пользователя.

Тип транзакции регламентируется значением поля `BalanceOperation/type`:

- 0 - Вывод;
- 1 - Пополнение;
- 2 - Оплата за продвижение поста;
- 3 - Заработок за модерацию;
- 4 - Заработок за комментарии;
- 5 - Заработок за посты;
- 6 - Пополнение через активацию купона.

×Switch to Console

#### Attributes

limit	number
	10
offset	number
	0

#### Request

Production

Raw

Try

#### Response

##### ATTRIBUTES

userBalance	number
	текущий баланс пользователя
	100

# Формирование команды

В команде разработки может быть от 1 до 6 человек.

Всего над проектом могут принимать участие до 20 человек.

# Уровни участников команды

В каждой команде есть тим-лид - он играет роль технического лидера команды и технического координатора работы над проектом.

Все остальные - разработчики одного уровня компетенции.

# Путь разработчика

- Джуниор;
- Мидл;
- Сеньор;
  
- Исполнитель;
- Тим-лид (архитектор);
- Технический лидер платформы;
- Лидер направления разработки.

# Проектирование проекта

Проектирование проекта выполняется лидером.

Главная цель - выявление и продумывание сложных мест, приведение будущего кода к единообразному виду, инициализация кодовой базы.

# Написание документации

Артефакт для будущих поколений, описывающий основные технические вехи, на которых базируется кодовая база проекта.



# Анализ артефактов

Никогда не удаётся всё продумать без ошибок с первого раза. Ошибки всегда есть в ТЗ, в дизайне и даже в исходной задумке.

Задача каждого члена команды - выявить как можно больше таких проблем на самом раннем этапе.

# Кодинг

Пишем качественно, быстро, чисто.

Советуемся с коллегами, проверяем работу других и учимся каждый день.

# Скрам-митинги

Общение - ключевой фактор для достижение успешности проекта.

Митинги могут проводиться ежедневно или по необходимости. Все члены команды синхронизируются друг с другом, рассказывают о своих успехах и проблемах.

# Цикличность

Из Agile мы позаимствовали цикличность процессов. Все вышеперечисленные шаги повторяются до тех пор, пока проект не будет готов к релизу.

# Тестирование

Тестировщики ломают то, что создали разработчики.

В этом им помогают написанные тест-кейсы, дотошно описывающие все нюансы функционирования программного продукта.

# Отладка

То что сломано - надлежит починить. Отладка может занимать до 30% от всего времени разработки проекта.

Мы стремимся к тому, чтобы понижать этот процент.

# Цикличность

Из Agile мы позаимствовали цикличность процессов. Все вышеперечисленные шаги повторяются до тех пор, пока проект не будет готов к релизу.

# Приёмка

Сборку, готовую к релизу, принимает начальник отдела дизайна, техлид отдела, начальник отдела тестирования, ПМ, а также продуктовик.

Все претензии, возникшие в ходе приёмки, имеют высочайшую важность и должны быть обязательно поправлены до релиза.



# Ретроспективы

Полностью работоспособная версия появляется в Google Play.

Часто раскатка версии происходит поэтапно на часть активной аудитории.

# Релиз

Полностью работоспособная версия появляется в Google Play.

Часто раскатка версии происходит поэтапно на часть активной аудитории.

# Сбор аналитики

Нужно понимать, как ведёт себя пользователь.

Важно выявить проблемное место до того, как оно приведёт к финансовым убыткам.

- Firebase Analytics;
- Flurry;
- Yandex Метрика.

# Всё сначала

Проект живёт до последнего пользователя.

Пока проект развивается, все вышеперечисленные шаги повторяются вновь и вновь.

# Как быть на волне?

- Не зацикливайтесь на локальной задаче;
- Пытайтесь разобраться в том, что делают другие;
- Делитесь знаниями и получайте знания с благодарностью;
- Иногда забывайте о том, что вы пишете код;
- Стремитесь быть в команде авторитетом;
- Критикуйте ваших коллег (но аккуратно);
- Никогда не сомневайтесь, что вы знаете, как лучше поступить;
- Будьте инициативными;
- Умейте говорить “Я не знаю этого”;
- Беритесь за задачи, к которым не знаете, как подступиться.

# Как развиваться?

- Общение с теми, кто опытнее вас;
- Общение с такими же, как вы;
- Статьи в интернете и SOF;
- Курсы в интернете;
- Курсы в оффлайне;
- Обучение тех, кто знает меньше вас;
- Книги?
- Изучение исходников опен-сорс проектов;
- Разработка пет-проекта;
- Стажировки (следите за соцсетями).