



# New > Конспект > 2 урок > ПРОДУКТОВАЯ АНАЛИТИКА

## > Оглавление урока по Продуктовой аналитике

1. Приоритизация
2. Важность тестирования гипотез
3. При генерации гипотез важно понимать
4. Подходы к приоритизации
5. Метод Impact/Effort
6. Метод RICE
7. Фреймворк ICE
8. Другие фреймворки для скоринга
9. Эксперименты
10. Материалы с лекции

## > Приоритизация

Когда мы говорим про процесс управления продуктом, под приоритизацией можно подразумевать:

1. Приоритизацию идей / будущих проектов или фич в продуктовом бэклоге. Выбранные идеи формулируются в задачи и попадают roadmap продукта
2. Приоритизацию гипотез в бэклоге продуктовых гипотез. Гипотезы могут быть разного характера: как про поведение пользователя, так и ожидаемое влияние конкретного изменения в продукте.

Зачем приоритизировать?

- Время ограничено (мы сможем сделать только X работ за Y дней/месяцев/лет)
- Время = деньги (каждый эксперимент / фича / проект стоят денег аналитика, разработчика, дизайнера или маркетолога)

## > Важность тестирования гипотез

Модель продукта в голове не всегда соответствует реальности. Если что-то звучит логично, это не значит, что оно работает. Это лишь гипотеза, которая требует проверки.

Эксперименты (генерация и тестирование гипотез) нужны для того, чтобы получить больше знаний о продукте. Часто, они часто приносят неожиданные результаты (Success = Experimentation)

Мы проводим эксперименты, потому что хотим лучше понять свой продукт:

- Мы начинаем с модели продукта, которая есть у нас в голове.
- Формируем с помощью нее определенную гипотезу о том, как нам добиться конкретной цели (повысить активацию новых пользователей, конвертировать больше пользователей в платящих и т.д.)
- Придумываем эксперимент, который позволит эту гипотезу проверить.
- Получаем результаты, которые позволяют лучше узнать продукт и скорректировать планы.

Это процесс идеальной работы над развитием и улучшением продукта с помощью экспериментирования.

## > При генерации гипотез важно понимать

При генерации продуктовых гипотез важно проверять себя вопросами:

- Как знания, полученные в результате проверки этой гипотезы, помогут двинуть продукт вперед?
- Как предложенное изменение в теории может повлиять на продукт? На какие части модели продукта нацелено это изменение? Как, если оно сработает, это скажется на других узлах модели продукта?
- Является ли эта гипотеза самой важной для проверки в текущий момент?

Помните, что тест гипотез стоит денег (время разработчика, аналитика, маркетолога, дизайнера и т.д.), и если вы понимаете, что гипотеза никак не влияет на ключевые метрики, не добавляйте её.

## > Подходы к приоритизации

Можно выделить 2 подхода к приоритизации:

- Быстрая оценка – помогает отбросить наиболее неподходящие варианты
- Глубокая оценка – тщательный скоринг оставшихся опций

Представьте, что гипотезы в бэклог добавляют все сотрудники отдела и каждый хочет, чтобы его гипотеза была первой в очереди на тестирование. Чтобы избежать недопонимания и конфликтов, процесс приоритизации и скоринга гипотез должен быть максимально прозрачным. В этом помогают фреймворки приоритизации. Рассмотрим наиболее популярные.

## > Метод Impact/Effort

Фича/гипотеза оцениваются по двум критериям – влияние на продукт или ключевую метрику и трудозатраты (сложность реализации).

**Effort** может быть в человеко-часах либо в сторипointах. **Impact** может распределяться по шкале 1-5 либо по шкале high/medium/low.

Выделяются 4 группы инициатив, которые берутся в работе в порядке приоритетности:

1. **Quick Wins** - приносят больше всего ценности, но которые можно быстро и легко реализовать
2. **Big Bets** - могут принести много ценности, но их сложно реализовать.
3. **Maybes** – не дадут много ценности, но их легко внедрить. Можно оставить на потом.
4. **Time Sinks** - не достойны внимания.

## > Метод RICE

- **Reach** — охват
- **Impact** — влияние
- **Confidence** — уверенность оценке охвата, влияния и трудозатрат
- **Effort** — трудозатраты

$$RICE\ SCORE = \frac{Reach * Impact * Confidence}{Effort}$$

**Reach** — показывает, сколько пользователей будут применять новую функциональность или будут зааффецены экспериментом. Источником данных могут быть эсистемы веб-аналитики, прошлый опыт или информация о конкурентах.

**Impact** — это величина, характеризующая влияние вашей фичи. В идеале ее нужно представлять в денежном эквиваленте, так как для бизнеса это понятнее всего. Но часто останавливаются на субъективной оценке. Лучше, когда она отражает влияние на ключевые метрики или на продуктовую стратегию / стратегию компании.

**Confidence** — уверенность оценке охвата, влияния и трудозатрат. Тоже является субъективной оценкой, но можно пользоваться логикой эталонности источника

информации о Reach и Impact. Собственное мнение – наименее эталонный, результат А/Б теста – уже близкий к наиболее эталонному (подробнее в презентации).

**Effort** – трудозатраты (сложность проекта / фичи / гипотезы)

## > Фреймворк ICE

- Разработан Шоном Эллисом (автор термина Growth Hacker).
- Был предназначен для приоритизации экспериментов по росту, но позже стал использоваться и для приоритизации фичей.
- В ICE используется шкала от 1 до 10

Очень похож на RICE, но в нем нет критерия Reach и Ease оценивается по другой логике (чем легче реализация – тем больше балл).

$$ICE = Impact * Confidence * Ease$$

## > Другие фреймворки для скоринга

**Kano Model (Модель Кано)** Подход разработал японский ученый и профессор Нориаки Кано. Он основывается на эмоциональном восприятии пользователем той или иной функциональности. Кано выделяет следующие типы реакции пользователя:

- **Must Be** — минимальные требования; если их нет — пользователь не удовлетворен.
- **Indifferent** — функционал, который вызывает неоднозначную реакцию. Пользователям все равно, есть он или нет.
- **Satisfiers (Performance)** — функционал, который вызывает удовлетворенность, если он выполнен хорошо, или разочарование — если качество низкое.
- **Exciters (Attractive)** — функционал, повышающий удовлетворенность пользователя, если он есть. Если его нет — недовольства не возникает.

Сначала фокусируются на проектах / гипотезах, относящихся Must be, затем — на Satisfiers и только потом — на Exciters.

**Buy the Feature** Отлично работает, когда у продукта много стейкхолдеров. Им выдают денежные купоны, как из настольной игры «Монополия», которые приравниваются к бюджету проекта. Стейкхолдеры распределяют деньги между фичами. Приоритизация выстраивается по убыванию суммы вложенных монопольных денег.

**Karl Wiegers Method** Фреймворк придумал американский инженер, консультант и автор книги «Software requirements» Karl Wiegers. Оцениваются benefit, penalty, cost и risk проекта / фичи по шкале от 1 до 9. В оценке участвуют пользователи и разработчики. Пользователи оценивают пользу от присутствия фичи и вред от ее отсутствия. Разработчики оценивают стоимость имплементации этой фичи и риск, связанный с ее разработкой. Далее данные подставляются в формулу и рассчитывается коэффициент приоритетности.

**Feature buckets** Подход придумал бывший CEO LinkedIn Adam Nash. Все фичи условно распределяются между 3-мя ведрами:

- **Metric Movers** — функционал, который двигает основные бизнес & продуктовые метрики: Engagement, Growth, Revenue etc.
- **Customer Requests** — функционал, который запрашивают пользователи (но нужно помнить о том, что не каждое желание пользователя может быть реализовано)
- **Customer Delight** — фичи, которые не обязательно запрашивают пользователи, но которые принесут им ценность.

Распределяя функционал по этим 3-м категориям, можно честно ответить на вопрос: «Почему мы делаем этот функционал?». Потому что пользователи требуют? Потому что компания требует? Или просто потому что это killer фича?

Подробнее тут:

- Модель Кано
- Buy the Feature
- Karl Wiegers Method
- Feature buckets

## > Эксперименты

Когда гипотезы сформулированы и отприоритизированы, можно думать над дизайном эксперимента. Шаблон описания эксперимента:

1. Гипотеза (чего и как хотим добиться)
2. Что делаем в продукте (как выглядит опыт тестовой и контрольной группы)
3. На каких пользователях тестируем (какие пользователи и в какой момент становятся частью эксперимента)
4. Ключевые метрики для оценки эксперимента (что должно поменяться, как это измерить)
5. Ожидаемый эффект -> размер выборки для эксперимента\*
6. План действий в зависимости от результатов эксперимента (если X, то делаем Y, если Z, то делаем Q)

## Пример

**Наблюдение:** На этапе регистрации бóльшая доля пользователей отваливается на четвертом шаге заполнения формы из пяти, так и не завершив регистрацию. **Гипотеза:** Если мы сократим форму онлайн-заявки с 5 до 4 пунктов, то увеличим конверсию в регистрацию на 1 процентный пункт с 2% до 3%, потому что в системе аналитики мы заметили, что бóльшая часть пользователей отваливается на четвертом шаге, так и не завершив регистрацию. **Что делаем:** Контрольная версия: оставляем все, как есть. Тестовая версия: сокращаем форму онлайн-заявки с 5 до 4 пунктов. **На каких пользователях тестируем:** Только на новых пользователях. **Метрики:** Конверсия в регистрацию. **План действий:** – Если наш эксперимент будет положительным, и мы увидим статистически значимое улучшение в метрике — масштабируем на всех пользователей.— Если метрики падают, то откатываем.— Если метрики не меняются, то оставляем, как есть.

## Инструментарий для тестирования

Для тестирования гипотез есть готовые инструменты. Например, Google Optimize и Visual Website Optimizer. При этом важно ознакомиться с тем, как данные сервисы считают различные метрики (напр. конверсию), так как иногда могут встретиться различия.

- Часто аналитикам приходится считать, сколько потребуется пользователей в тестовой и контрольной группах, чтобы заметить определенное изменение в метрике (которое будет считаться стат значимым). Это нужно чтобы прикинуть, сколько дней закладывать под конкретный эксперимент (и касается тестов на относительные метрики типа конверсий).

Как прикинуть длительность эксперимента: 1. Определяемся с мощностью теста (того, с какой вероятностью мы хотим увидеть интересующее нас изменение, если оно в реальности есть) и уровнем доверия. Мощность обычно устанавливают на уровне 80%, а уровень доверия - 95%. 2. Определяем минимальный эффект, который хотим заметить в эксперименте (в %). Это будет conv2-conv1 в формуле ниже. conv2 - ожидаемая конверсия тестовой группы, conv1 - та конверсия, которую мы имеем сейчас. 3. Решаем уравнение подставляя значения в формулу и ищем X (X - необходимый минимальный размер выборки для каждой группы)

$$x = conv2 - conv1 \pm 1.96 * \sqrt{\frac{(conv1 * (1 - conv1))}{x} + \frac{(conv2 * (1 - conv2))}{x}}$$

- 1.96 - значение коэффициента для уровня доверия 95%. Он может меняться:
  - уровень доверия 90% = 1.645
  - уровень доверия 95% = 1.96
  - уровень доверия 99% = 2.575

Для задач оценки минимального размера тестовой и контрольной групп существуют готовые калькуляторы типа этого <https://www.evanmiller.org/ab-testing/sample-size.html> и гораздо проще пользоваться ими.

## > Материалы

### 1. Презентация