



Управление продуктовыми требованиями



Настя Московченко

Product Manager at FunCorp

Отвечаю за retention текущих продуктов и запуски новых.

- ex: Product manager VK
- ex: Технический менеджер Яндекс. Геосервисы
- Пишу про продакт менеджмент - <https://t.me/botaytoday>



1

Урок 1

Продуктовое мышление

2

Урок 2

Формирование продуктовых требований

3

Урок 3

Управление продуктовыми требованиями

4

Урок 6

MVP и RAT

5

Урок 5

Базовые продуктовые метрики и отслеживание результата

Что включает в себя стратегия продукта

Видение продукта

Для чего и для кого этот продукт? Успех продукта в конечном итоге достигается, если продукт выходит на новых пользователей и доставляет им определенную ценность.

1

Метрики

Метрики позволяют измерять ход и прогресс достижения целей. Для каждой цели устанавливаются свои метрики и отслеживается их прогресс

3

Цели продукта

Цели по SMART помогают менеджерам продуктов установить, чего они хотят достичь (увеличить доходы, расширить присутствие в новых странах, повысить мобильную адаптацию и т. д.)

2

Конкретный план действий

Здесь должны быть доступно расписаны все этапы и конкретные шаги по достижению целей

4

Планирование по системе GIST

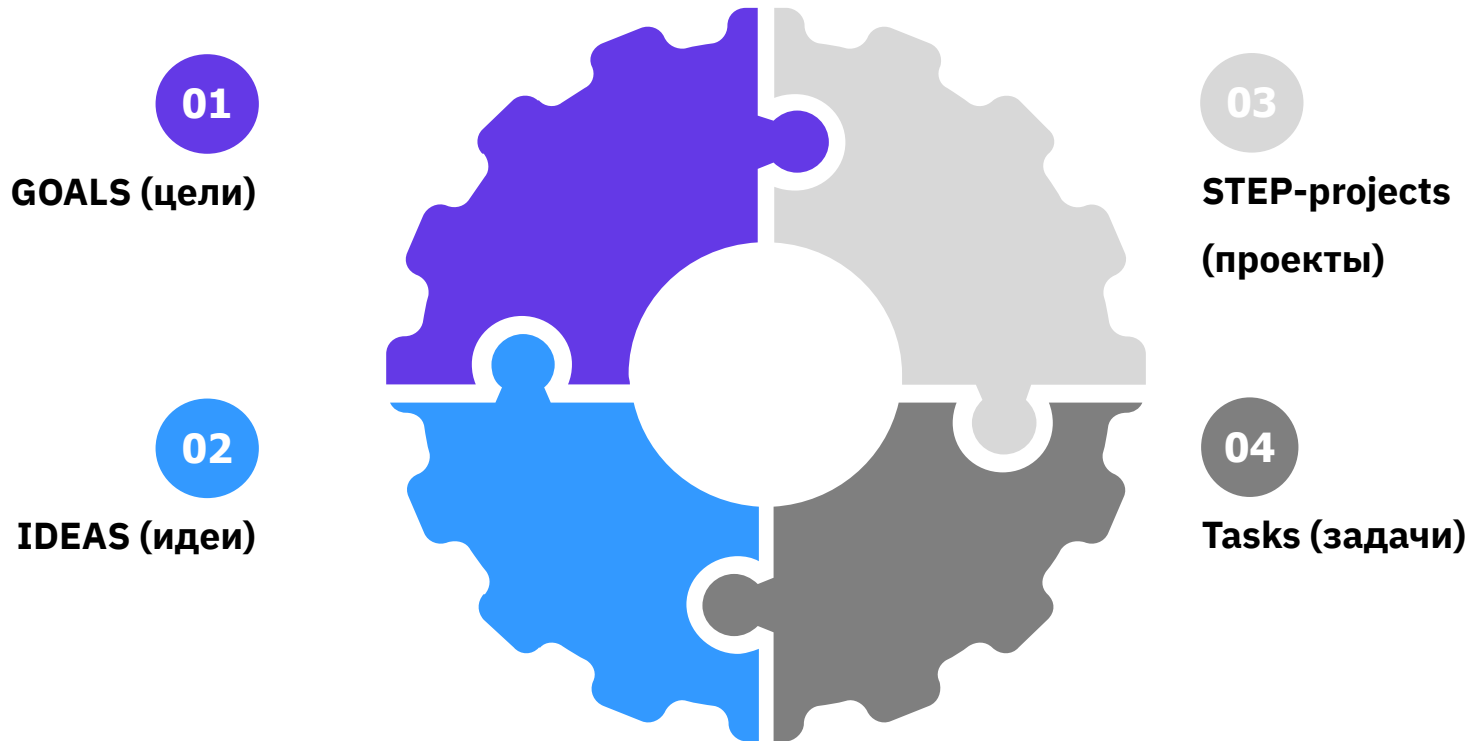
Удобный инструмент планирования, который позволяет:

- Зафиксировать стратегические цели продукта на 1-3 года
- Связать цели с продуктовыми гипотезами и конечными работами
- Не терять фокус при генерации гипотез
- Донести до команды, **зачем** нужны задачи, которые они выполняют





Система **GIST**





G – goals (цели)

Цели описывают стратегию компании с точки зрения желаемых результатов: **где мы хотим быть? когда и как мы узнаем о том, что мы их достигли?**

Всякий раз, когда кто-либо в организации задается вопросом: **«Почему мы делаем этот проект?»**, цель должна давать ясный ответ.

I – ideas (идеи)

Идеи — это гипотетический способ достижения целей. Гипотетический потому, что у вас может быть много идей для достижения заданной цели, но только 1–3 из них приведут к положительному результату (а часто соотношение и того хуже). GIST подразумевает, что вы будете:



Собирать все идеи в банк идей. При этом банк может хранить сотни идей неограниченно долго и все из них приветствуются.



Будете приоритизировать на основе фактов



Будете тестировать как можно больше идей в порядке приоритета



S – Step projects (проекты)

Большой проект разбивается на мелкие проекты, которые длятся не более 10 недель и выполняются по очереди. Например:

Детализированный статичный прототип → Интерактивный прототип → MVP → Dogfood → Beta → Launch

Каждый такой проект—это эксперимент, который проверяет идею. То есть с каждым проектом вы получаете все более полную версию идеи и проверяете ее на более широкой аудитории в течение все более длительного времени.

Конечный продукт, как правило, намного лучше, чем тот, который вы себе представляли изначально. Неработающие идеи отсеиваются рано, а идеи, которые работают, получают больше инвестиций.

T – Tasks (задачи)

Каждый проект разбивается на задачи. Эта часть системы планирования отлично помогают визуализировать Agile-ориентированные инструменты для планирования, Kanban-доски, и современные технологии для управления проектами.



Планирование с помощью GIST

многоуровневое и интерактивное

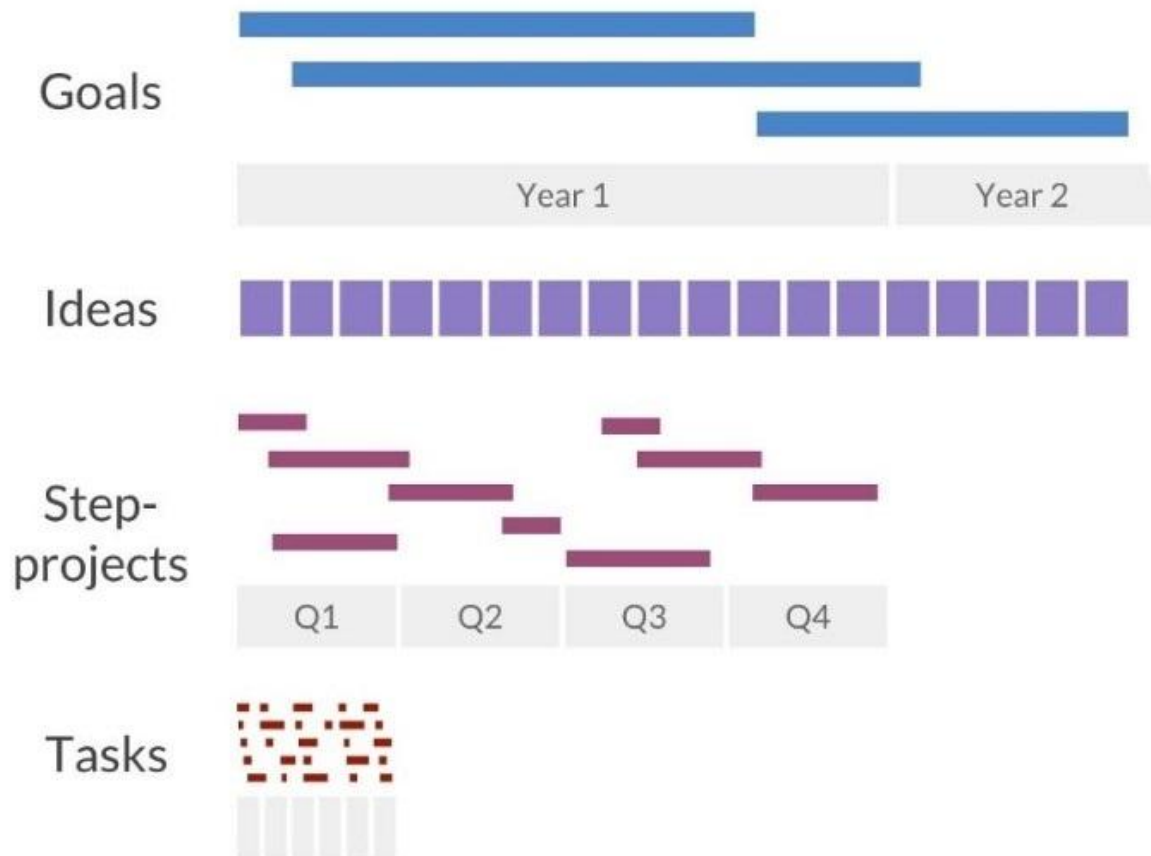


Цели обычно устанавливаются с прицелом на один год или нескольких лет.

Идеи постоянно собираются и приоритизируются. Важно не переставать искать новые идеи.

Проекты определяются в начале квартала. Команда выбирает цели и идеи, которые она хочет выполнить в этом квартале и определяет проекты.

Задачи разбиваются на 1–2х недельные итерации в соответствии с вашим методом разработки и корректируются ежедневно.



А когда наступит счастье?

Для бизнеса важно понимать, **когда** продуктовая команда достигнет стратегических целей





1

Стратегия продукта

Целевые метрики

2

Поиск гипотез

Product Backlog

3

Фокусировка и приоритезация

Product Roadmap

4

Проектирование

Релизный план

5

Разработка

Development Backlog



Product Backlog

Состоит из **гипотез**:

- Что мы можем сделать, чтобы повлиять на целевую метрику?
- Какой прогноз по изменению метрики?
- Как будем измерять результат?
- Как можем проверить без разработки продукта?

Обычно владельцем Product Backlog'а является менеджер продукта



Что лежит в Product Backlog?

Гипотезы улучшений могут направлены не только на рост целевой метрики, но и на поддержание базового качества продукта



Гипотезы, влияющие на целевые метрики

- Стратегия продукта
- Конкурентный анализ
- Feature gaps и анализ CJM
- Запросы от стейкхолдеров
- Запросы от пользователей



Поддержание базового качества

- Запросы от пользователей
- Product debt
- Баги
- Задачи на исследования
- Запросы от техподдержки и стейкхолдеров



Приоритезация Product Backlog

PRD 1

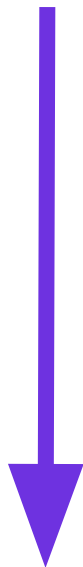
PRD 2

PRD 3

PRD 4

...

PRD N



приоритет

Зачем упорядочивать?

Чтобы в первую очередь делать самое важное = скорее принести пользу и быть раньше конкурентов

Ресурсы ограничены и нужно выбрать, чем заняться команде



RICE Scoring

$$\text{RICE score} = \frac{\text{Reach (Аудитория)} \times \text{Impact (Влияние)} \times \text{Confidence (Уверенность)}}{\text{Effort (Трудозатраты)}}$$

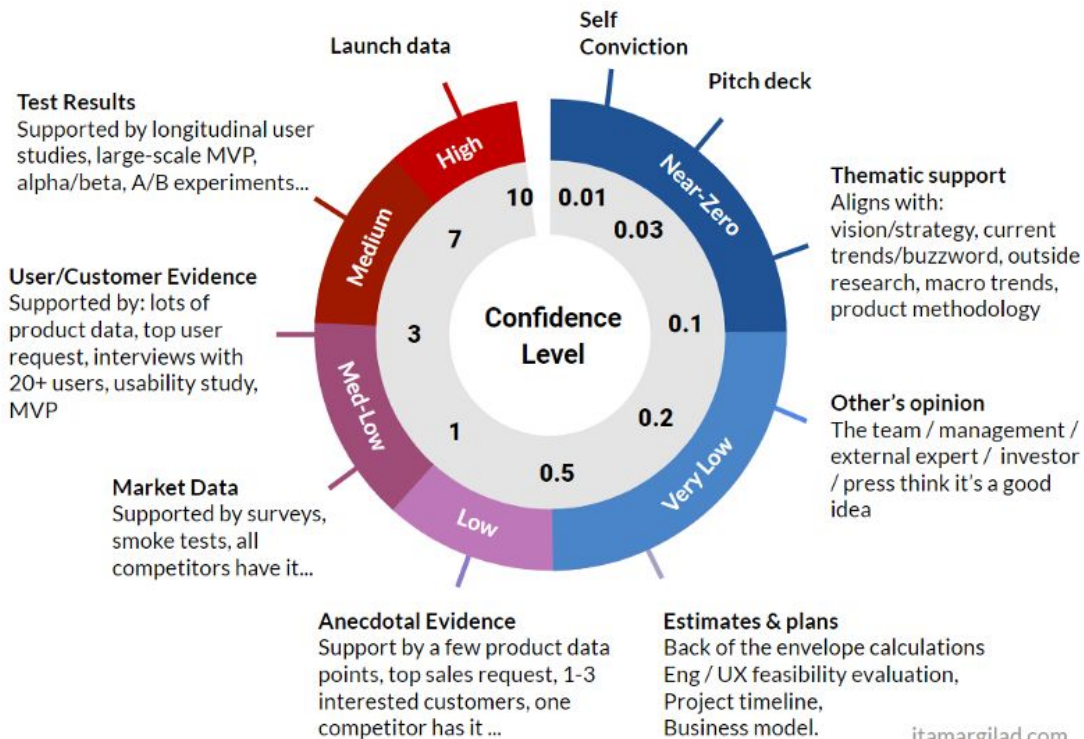
Reach абсолютными показателями аудитории (обычно в тысячах) или % от текущей аудитории сервиса

Impact зависит от бизнес-целей сервиса и может измеряться в приносимой пользе: бизнесу (доля рынка или прибыль), пользователю (“счастье пользователя” удобство использования), вкладу в retention и т.п.



Confidence в RICE

Если мы считаем, что фича может иметь огромное влияние, но у нас нет данных для доказательства этого, Confidence позволяет **проконтролировать** ЭТОТ МОМЕНТ





ICE Scoring

$$\text{ICE score} = \frac{\text{Impact} \times \text{Confidence}}{\text{Easy}}$$

Влияние × Уверенность

Легкость реализации

Изначально использовался для приоритизации гипотез роста, то есть приоритезации **внутри одной ключевой метрики**

Impact влияние на выбранную ключевую метрику бизнеса

Easy легкость реализации, например необходимость задействовать несколько отделов или суммарные трудозатраты



Приоритезация по ROI

$$\text{ROI} = \frac{\text{Доход от фичи} - \text{Расход на создание}}{\text{Расход на создание}} \times 100\%$$

Для расчета ROI обычно выбирают долгосрочный период доходов: квартал или год, реже — суммарный доход от внедрения (для активностей с конечными сроками)

В работу берутся только задачи с **ROI > 0**

- Можно привязать все инициативы к доходности бизнеса
- Но сложно оценивать маленькие фичи или поддержание базового качества



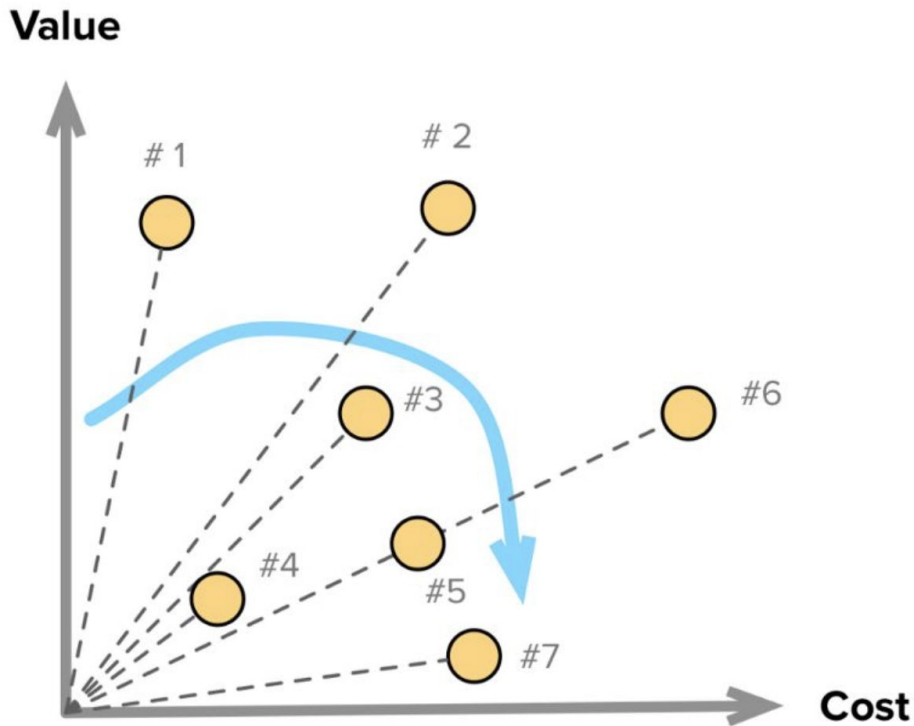
Value VS Cost

Или Bang For The Buck Prioritization

Аналогичен расчету ROI, но позволяет визуализировать приоритеты и не требует точного расчета

Шкала Value может быть не только привязана к доходам, возможны вариации с другими метриками продукта

- Акцент на скорости доставки ценности





Value VS Risk

Учитываются риски:

Планирования

Мы не сможем выполнить задачу за ожидаемое время или определенными ресурсами

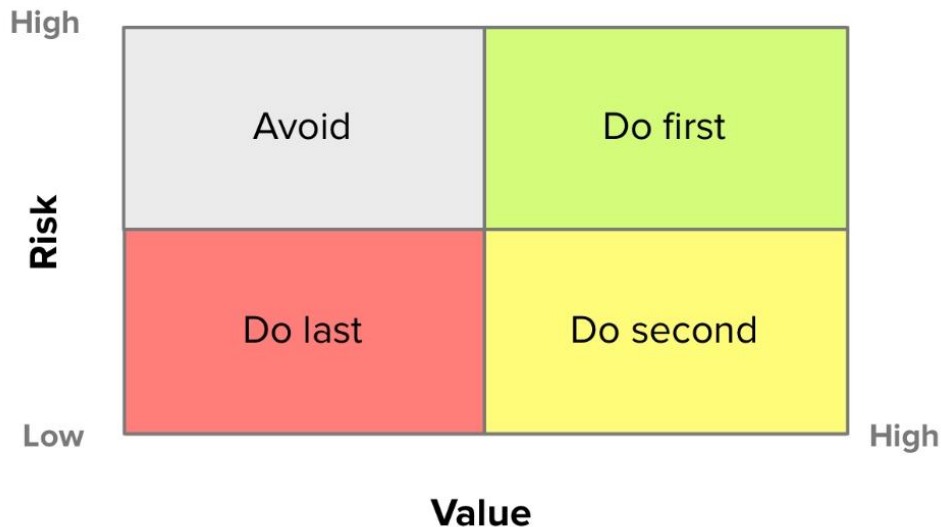
Стоимости

Запуск может стоить дороже

Функциональные

Это невозможно сделать

- Акцент на быстром росте и “захвате рынка”





Еще варианты

- Метод Weighted Scoring
- Метод Value vs Cost
- Популярная модель Kano
- Метод приоритизации MoSCoW
- Методология Buy a Feature
- Технология Feature buckets
- Игровая техника KJ Method
- Метод Quality Function Deployment (QFD)

Product Roadmap

Состоит из **сильных гипотез**

- Приоритезируем гипотезы
- Выбираем самые сильные
- Предполагаем временные промежутки, когда будет реализация

Обычно владельцем Roadmap'a является менеджер продукта или кто-то из стейкхолдеров



The 666 roadmap (Intercom)



Сейчас



Потом



Дальше



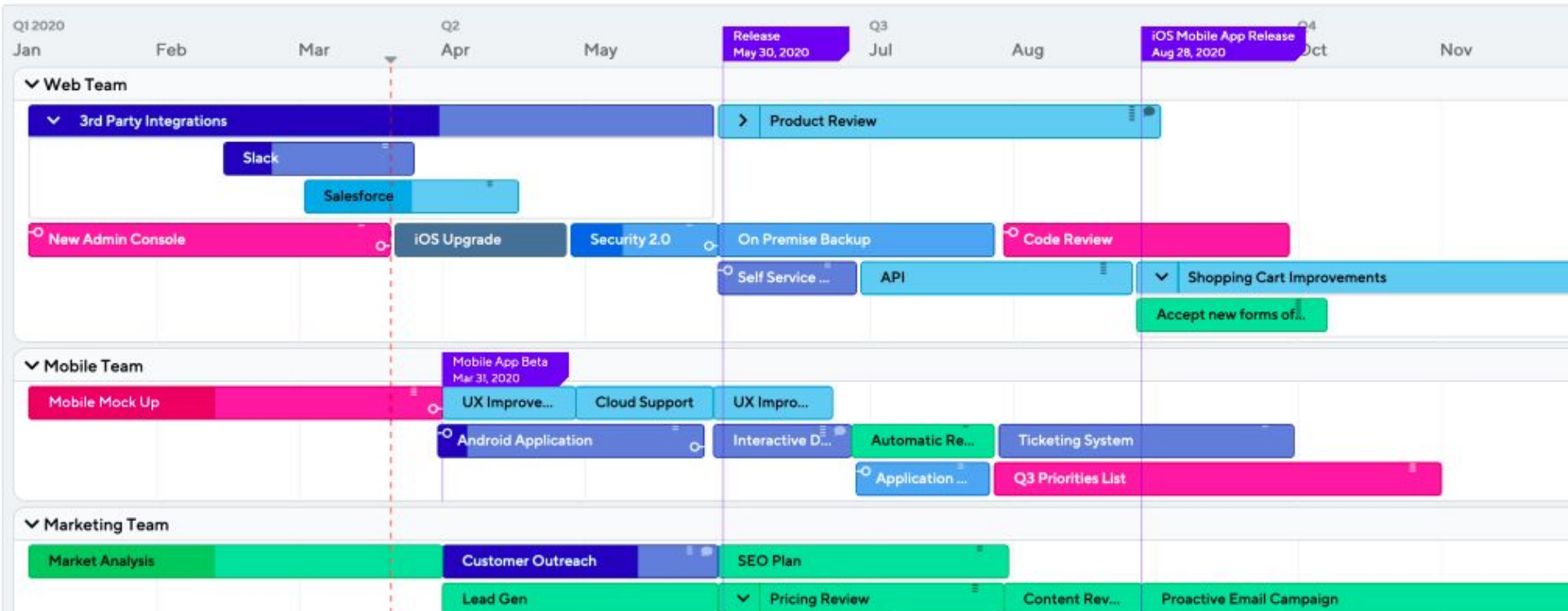
Релизный план

Состоит из **фичей, которые будут в релизах**

- Гипотезы трансформируются в фичи, которые должна реализовать продуктовая команда (epics, user stories)
- Оцениваем верхнеуровнево трудозатраты
- Следим за ресурсными конфликтами
- Оставляем “воздух”
- Определяем сроки поставки фичей

Владельцем релизного плана является менеджер проекта или техлид

Релизный план



Development Backlog

Состоит из **задач на разработку**

- Задачи декомпозированы, оценены
- Для каждой есть описание и разработчику понятно, что делать

Обычно владельцем Development Backlog'а является техлид или тимлид команды



Что лежит в Development Backlog?



Продуктовые задачи

- PRD
- Задачи на исследование



Поддержание базового качества

- Найденные дефекты, видимые пользователю



Технический долг

- Скрытые от пользователя дефекты, связанные с архитектурой и масштабируемостью
- Рефакторинг

Если планы меняются

Готовность к изменениям и возможность
быстро реагировать на изменения —
важный навык менеджера продукта





Почему меняются планы?

Продуктовая разработка связана с постоянным тестированием гипотез, и, как следствие, с реакцией на результаты тестирования



Гипотезы не подтвердились

Не все гипотезы подтверждаются, поэтому продолжать разработку не имеет смысла



Нашли более выигрышные гипотезы

В ходе продуктовой работы могут быть найдены и подтверждены гипотезы, которые имеют большее влияние на продукт и достижение целей



Рынок изменился и нам тоже пора

Выход на рынок нового игрока или перераспределение сил может повлечь за собой изменение продуктовой стратегии и продуктовых целей



Что делать?

1. Выяснить причины изменения
2. Оценить, насколько срочно нужно менять курс: есть ли время на консервацию изменений в продукте и ждут ли изменения следующего планирования спринта
3. Оценить, насколько может пригодиться то, что уже было создано: нужно удалить изменения/ законсервировать/ не начинать разработку
4. Перепланировать спринт
5. Внести изменения в релизный план



Домашнее задание

- Выбрать продукт (можно работать со своей идеей или взять любой продукт, которым пользуетесь ежедневно)
- На основании построенной на прошлом уроке CJM составить беклог продукта (5-6 фичей) и приоритезировать их по RICE

[Шаблон домашнего задания](#)

