Как работать с Feature flag?

1. Что такое Feature flag?

Фича-флаги (Feature Flags) — модуль GrowthBook, позволяющий гибко управлять включением и выключением функциональной сущности в приложении. По сути своей, это IF-блок, который запускает кусок кода при выполнении некого условия. Фича-флаги позволяют управлять контентным содержимым: в интерфейсе вы можете присвоить то или иное значение в зависимости от каких-либо условий.

```
def провести_ab_тест(тоггл_включен):
    if тоггл_включен:
        # Логика для группы A (новая функциональность)
        print("Группа A: Новая функциональность включена")
    else:
        # Логика для группы B (старая функциональность)
        print("Группа B: Используется старая функциональность")
```

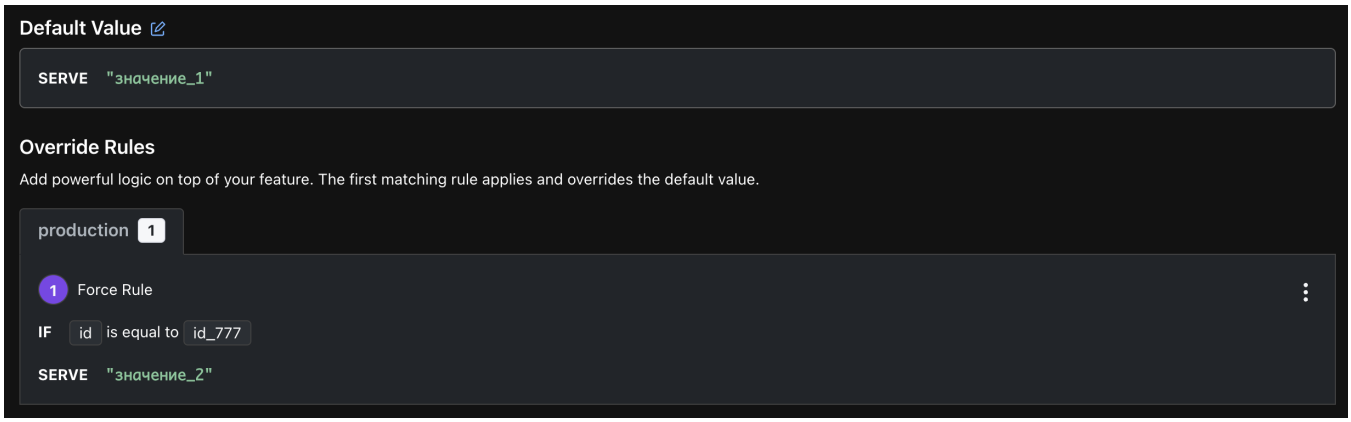
2. Tracking key (feature key)

Каждый фича флаг определяется уникальным ключом - tracking key. Этот ключ используется для идентификации эксперимента в SDK, соответственно на него и будет ссылаться команда разработчиков в своем коде при проверке значения фичи. Напомню, что именно tracking key используется вместе с атрибутом хеширования пользователя в алгоритме хеширования, чтобы гарантировать, что пользователи всегда будут рандомизированы в одну и ту же группу. Tracking key также передается в callback-функцию отслеживания в аппметрике. По умолчанию, tracking key совпадает с названием фича флага. **При изменении ключа эксперимента, пользователи перехешируются и попадут в иной бакет.**

> ⓘ Если задавать одно и то же значение ключа отслеживания для двух разных фича-флагов, то пользователи будут распределяться в группы эксперимента одинаково для обоих фича-флагов. Это может быть полезно в тех случаях, если необходимо, чтобы пользователи всегда находились в одной и той же группе для нескольких флагов.

3. Default Values и Targeting Conditions

Значения фича флага, если нет соответствующих предоопределенных правил для таргет атрибута. Мы оставляем пустым значение, так на данный момент не используем таргетинг. Если вы указываете что-то в **defaultValue**, это значение будет использоваться, когда никакие другие правила не применяются к текущему пользователю. Какие правила могут применяться? Например, мы можем задать условия **Targeting Conditions** в Forced Value или в A/B Experiment (Type of Rule).

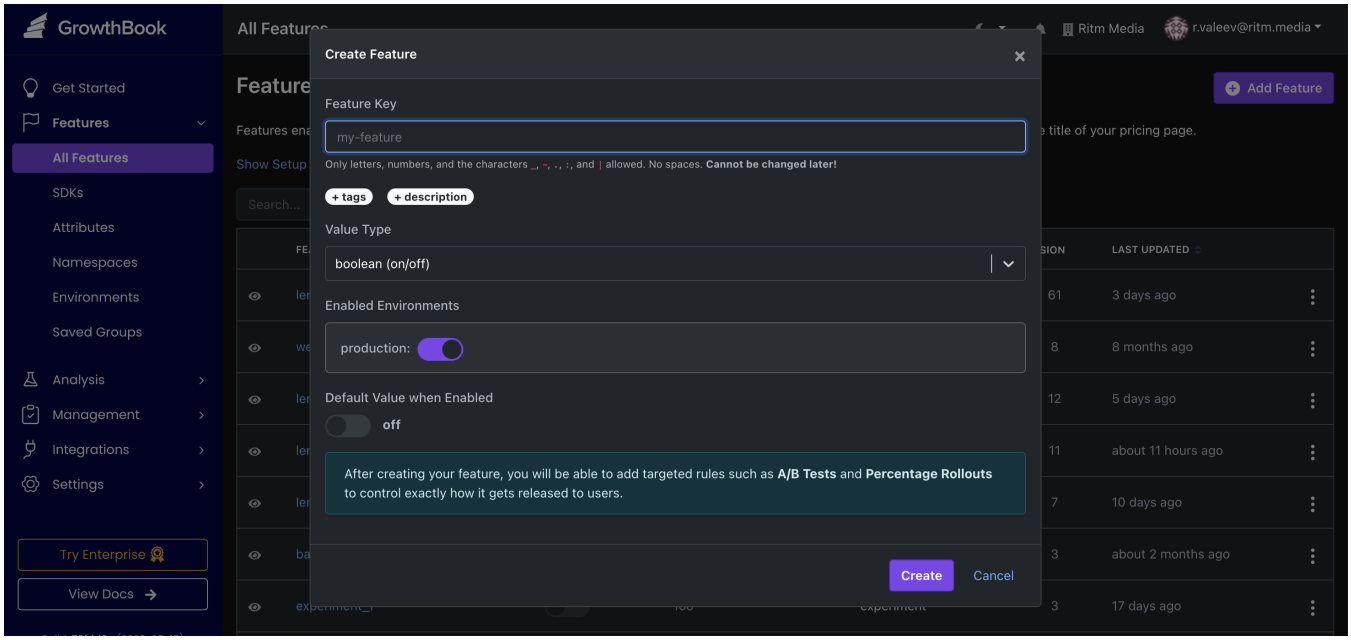


4. Алгоритм создания feature flag

После того, как эпик переходит в стадию "Формализация ФТ", автоматически создаются задачи на заведение фича флагов для теста в рамках данного эпика.

Для создания нового фича флага,

- Авторизоваться в клиенте https://gb.ritm.media/
- На навигационной панели (слева) выбрать **All Features**
- Кликнуть на кнопку **Add Feature**
- Введи название фича флага и его тип (обычно выбираем string)
- Кликнуть **Create** для создания пустого фича флага
- Задать группы эксперимента для данного фича флага
- Сформировать задачу в jira на создание фича флагов с группами экспериментов на команду ios/android (через проджекта АБ-тестов, https://t.me/ElizavetaKosta)

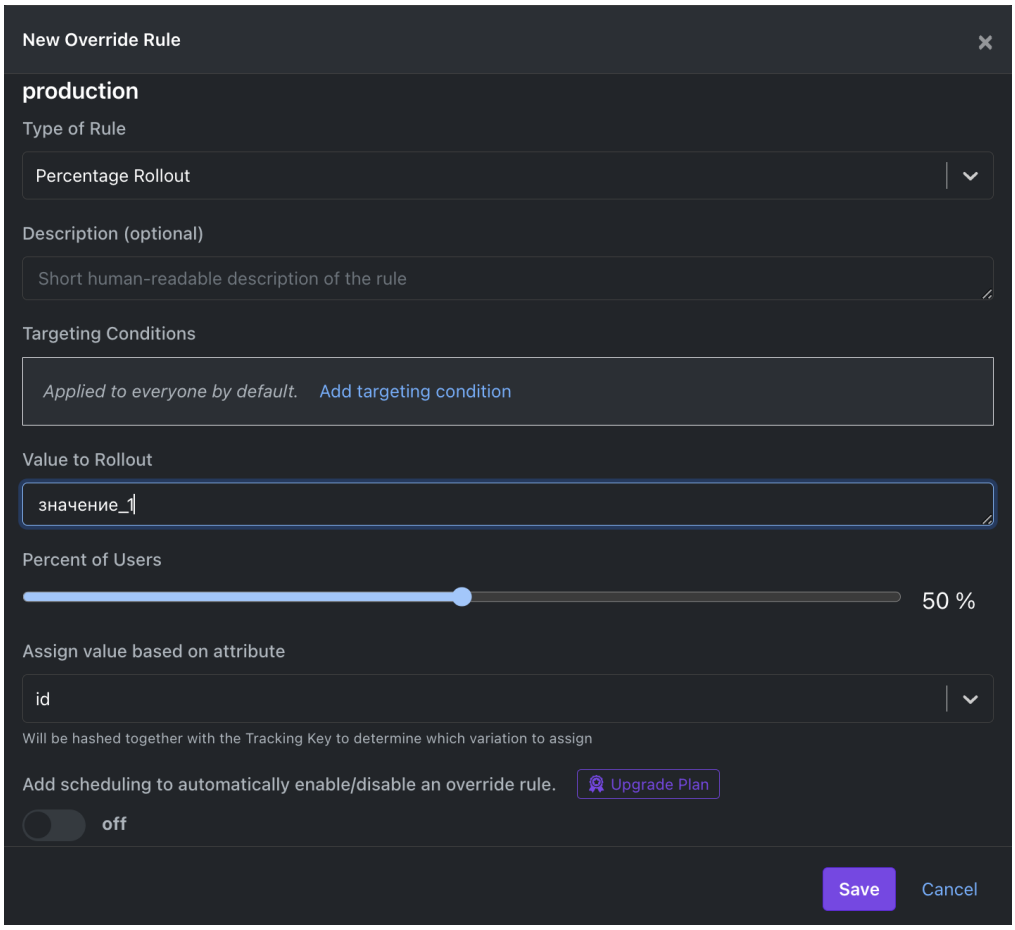


5. Почему нейминг feature flag имеет значение?

Есть несколько причин, по которым нейминг фича флага имеет важное значение. Первое, и самое важнейшее, исходит из алгоритма хеширования данных в сплитилке - идентификатор эксперимента (feature flag, tracking name) является одним из входных данных используемых в хэш-функции (Хэш функция используемая в Growthbook). То есть уникальность фича флага для каждого эксперимента минимизирует риск конфликтов и коллизий в хэш-значениях (случаи, когда пользователь попадает в один и тот же бакет). Проще говоря, **если не менять фича флаг, то для одних и тех же пользователей значение хэш-функции не будет меняться в разных теста**, что ведет к тому, что одни и те же пользователи будут постоянно попадать в одни и те же группы тестирования. Вторая причина - мы повышаем читаемость флага и прозрачность экспериментов, что важно для масштабируемости и долгосрочного управления фичами.

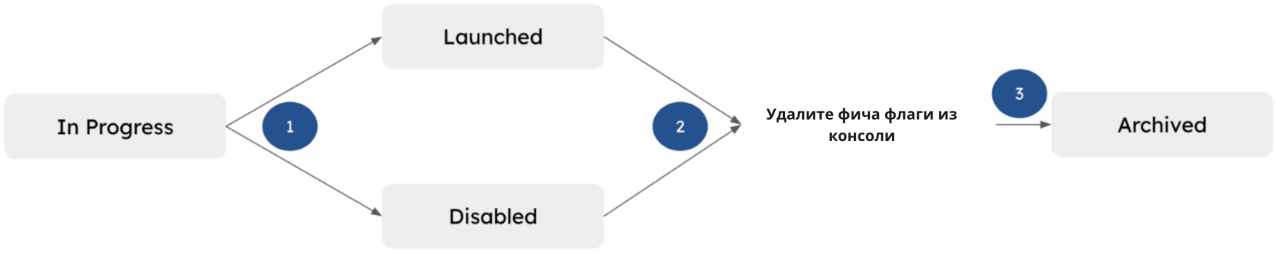
6. Как запускать канареечные релизы?

Percentage Rollout позволяет постепенно выпускать новую фичу для случайной выборки пользователей. Для правил Percentage Rollout вы выбираете атрибут пользователя, по которому будет производиться случайная выборка и значение для этой выборки.



7. Жизненный цикл Feature Flags

Фича флаг проходит через различные фазы на протяжении своего жизненного цикла — может быть, она еще тестируется на небольшой группе пользователей, или уже была полностью развернута и тд. Независимо от статуса, в которой находится фича флаг, её состояние должно быть четко отражено в консоле. Это важно так как может предотвратить потенциальные конфликты: ситуации, когда старый код для устаревшего фича флага используется повторно.



Рекомендации:

- Не использовать повторно один и тот же фича флаг для разных тестов;
- Фича флаги лучше не удалять, а архивировать;
- Регулярно проверять, какие фича флаги прилетают в систему мониторинга (appmetrica).