Тестовое задание на позицию Product Analyst
Бересневой-Батт Валерии

Теоретическая часть

1. Что такое LTV? Если у нас есть данные за первый месяц жизни приложения, как бы вы рассчитали LTV?

LTV - это выручка, которую приносит клиент за всё время его взаимодействия с продуктом.

В данном случае, для начала, я бы посмотрела на пользователей, которые пришли в первую неделю, м.б. первые две недели:
- разделила бы их на когорты по дате первого входа;
- посчитала бы их удержание за 2-3 недели на каждый день лайфтайма;
- посмотрела бы на их средний лайфтайм.
Это даст ориентир для расчёта Lifetime.
Затем посчитала бы ARPDAU за каждый день месяца как выручку за день / DAU, взяла бы среднее значение ARPDAU.
А затем применила бы формулу LTV = Lifetime * ARPDAU.

Здесь важно учитывать, что это только первый месяц жизни приложения. Предполагается, что выручка будет постоянно расти, как и Lifetime, и ARPDAU, поэтому данные будут меняться в будущем.

2. Что такое ARPPU? В результате изменений в продукте ARPPU снизился. Это хорошо или плохо? Поясните ответ.

ARPPU — это средний доход на одного платящего пользователя.
ARPPU = выручка за период / количество платящих пользователей за период.

Тут может быть несколько гипотез:

- Были акции / скидки / триал-периоды. Скорее всего, это снизило средний чек или количество покупок на пользователя, но увеличило количество платящих пользователей. Это неплохо, потому что могло помочь в привлечении новых пользователей, которые могут стать платящими, или миграции имеющихся пользователей из неплатящих в платящих. Смотря, какая была цель.

- Изменилась система монетизации (или в целом что-то изменилось):
Тут два основных варианта:
Изменения позволяют пользователям меньше платить за какие-то фичи, или есть какой-то фрод, или новая система не нравится пользователям и они уходят. Это довольно неприятно и нужно что-то менять.
Средний чек уменьшился, но увеличилось количество платящих пользователей. Это не самый плохой вариант, потому что в долгой перспективе это может дать бо́льшую выручку за счёт массовости низких чеков по сравнению с более редкими, но высокими чеками. Можно попробовать посчитать перспективу для старого и нового варианта и сравнить.

Таким образом, всё зависит от того, что было сделано и какой эффект ожидался. Не всегда снижение ARPPU является плохим сигналом.

3. При тесте новой маркетинговой компании значительно снизилась стоимость установки, но упал и ретеншн 1 дня. Какие метрики стоит смотреть и какую картину по этим метрикам вы ожидаете увидеть чтобы понять, это хорошо или плохо?

Стоимость установки могла снизиться за счёт увеличения количества установок, или конверсии из просмотра в установку, поэтому это первое, что стоит посмотреть. Повышение конверсии - это признак того, что новая кампания привлекает качественную аудиторию, которым интересен продукт. Вопрос, будут ли они играть в долгую.

Как доп. вариант - это использование другого агента для проведения кампаний (например, переход с Facebook Ads на Яндекс.Директ). Просто у нового агента услуги стоят дешевле, а эффект может быть таким же. Но, судя по описанному условию, просто поменялась кампания, а не агент.

Падение удержания в первый день лайфтайма, как правило, говорит о том, что новые пользователи реже возвращаются в приложение после установки и первого входа.
Первая причина, которая приходит в голову, - это несоответствие ожиданий от того, что было в рекламе, с тем, что пользователи видят в приложении.
Вторая - это технические проблемы, что отталкивает пользователей от дальнейшего входа. Это можно проверить, сравнив аналогичные метрики на новых пользователях, которые пришли органично в этот же период.
Сказать о том, что кампания привлекла некачественную аудиторию довольно сложно, если конверсия из просмотра в установку действительно увеличилась. Можно понаблюдать и сравнить удержание на 3, 5 день лайфтайма с предыдущей кампанией. Возможно, что пользователи возвращаются позже или удержание резко падает именно в 1 день, а в последующие дни падение незначительное.

Аналогично можно сравнить и churn rate для убедительности. Вероятнее всего, он растёт по мере снижения удержания.

Также стоит посмотреть показатели LTV и ROI. Возможно, что кампания привлекает меньше пользователей, но они более платёжеспособны. Или, если LTV не снизился, то новые пользователи окупаются быстрее за счёт снижения затрат на их привлечение.

Весь этот анализ следует провести, разделив пользователей на когорты по дате установки (может быть, какой-то сбой произошёл в конкретный день-два, что даёт сильный эффект на общие результаты) и по каким-то группам, например, ОС, версии ОС, типу девайса, возраста или локации, если есть такие данные. Это может показать, что проблема в каком-то конкретном сегменте.

Можно посмотреть отзывы в магазине, появились ли новые, что изменилось в настроении этих отзывов. Иногда это хорошая почва для поиска гипотез.

Практическая часть

Вовлечение

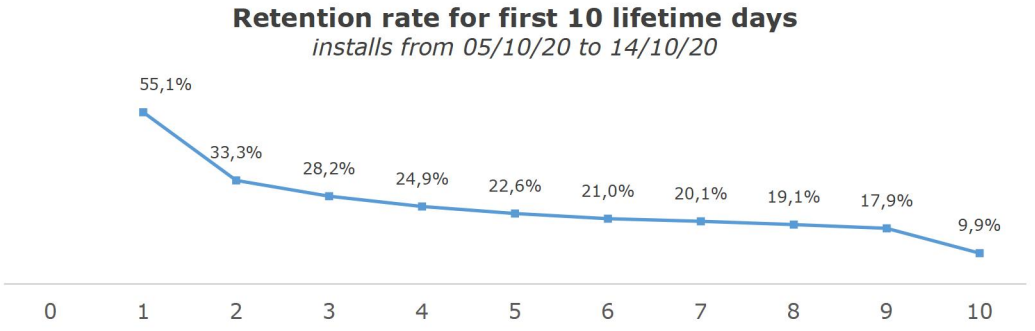
Ретеншн и время, которое игроки проводят в игре

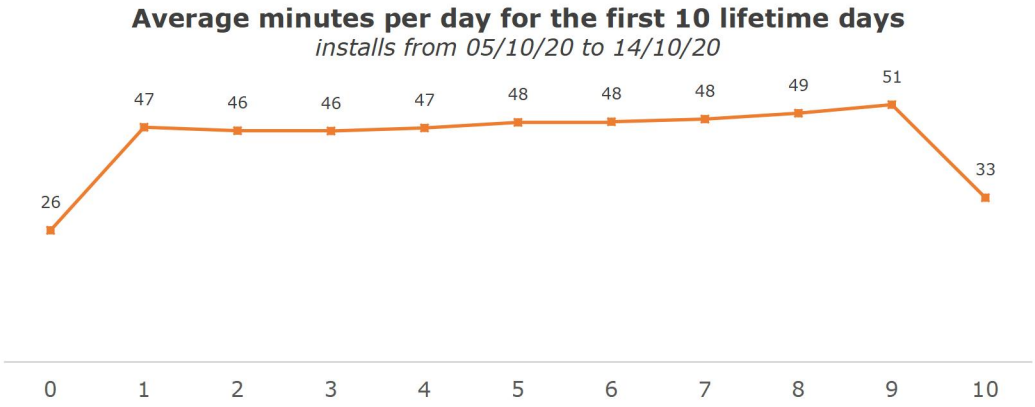
** скрипты написаны на SQLite*

```
WITH raw_1 AS ( -- installs & sessions dates
        SELECT
                DISTINCT
                i.user_id
                , DATE(i.reg_time) AS dt
                , NULL AS duration_sum
        FROM install AS i

        UNION ALL

        SELECT
                sc.user_id
                , DATE(sc.open_time) AS dt
                , SUM(sc.duration) AS duration_sum
        FROM session_close AS sc
        GROUP BY 1, 2
        )
, raw_2 AS ( -- lifetime calculation
        SELECT
                DISTINCT
                user_id
                , dt
                , SUM(duration_sum) OVER(PARTITION BY user_id, dt) AS duration_sum
                , MIN(dt) OVER(PARTITION BY user_id) AS reg_dt
                , julianday(dt) - MIN(julianday(dt)) OVER(PARTITION BY user_id) AS
lifetime
        FROM raw_1
        ORDER BY 1, 2
)
-- retention rate
SELECT
        r.lifetime
        , a.users_total_cnt
        , COUNT(DISTINCT r.user_id) AS users_cnt
        , COUNT(DISTINCT r.user_id) * 1.00 / a.users_total_cnt AS retention_rate
        , SUM(r.duration_sum) / COUNT(DISTINCT r.user_id) / 60 AS duration_min_avg
FROM raw_2 AS r
LEFT JOIN ( -- installs count
        SELECT
                lifetime
                , COUNT(DISTINCT user_id) AS users_total_cnt
        FROM raw_2
        WHERE lifetime = 0
        GROUP BY 1
        ) AS a
GROUP BY 1
```

Больше всего пользователей не возвращается на следующий день после установки (rr = 55%), на второй день лайфтайма (rr = 33%, -22 п.п.) и на десятый день использования приложения (rr = 9.9%, -8 п.п. по сравнению с 9-м днём).
Это может говорить о том, что в первые два дня пользователи знакомятся с игрой, проходят онбординг. Возможно, для кого-то первые квесты кажутся сложными или в целом не заинтересованы в игре, поэтому на втором дне теряется 22 п.п. по сравнению с первым.
Аналогичная ситуация может происходить и на 9-м дне: прохождение последующих квестов становится сложнее, поэтому на 10-м дне удержание на 8 п.п. меньше по сравнению с предыдущим.
Это можно проверить, посмотрев, на каких квестах уходят пользователи.

Продолжительность прохождения игры отчасти может подтвердить выдвинутую гипотезу о снижении rr на десятый день лайфтайма. На графике видно, что после того, как наименее заинтересованные пользователи ушли в 0 день лайфтайма, и их меньшая продолжительность нахождения в игре перестала занижать среднее значение, средняя продолжительность игры с каждым днём увеличивалась и достигла 51 минуты на 9-й день, а затем резко упала до 33 минут. Скорее всего что-то с одним из квестов.



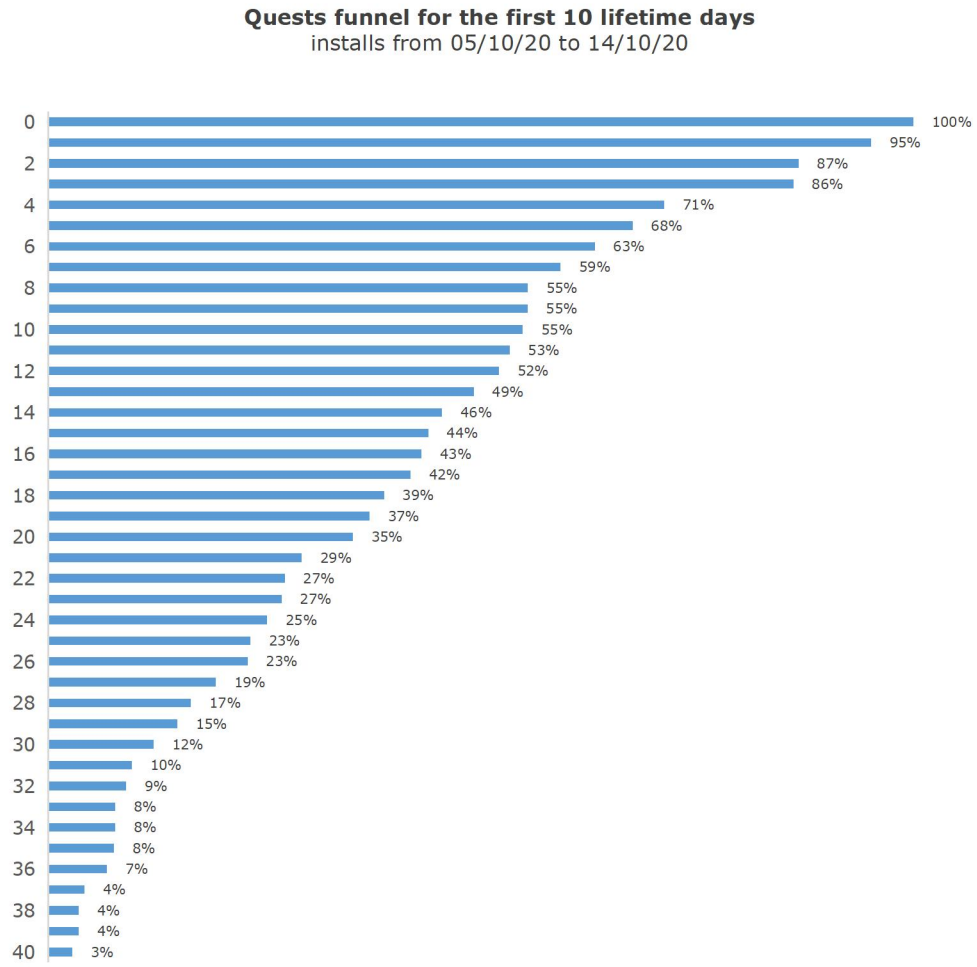


Воронка прохождения квестов

```
WITH
quest_0_start AS ( -- count of quest_0 starts
        SELECT
                CAST(SUBSTR(quest, INSTR(quest, '_') + 1) AS integer) AS quest_num
                , SUM(CASE WHEN time IS NOT NULL THEN 1 ELSE 0 END) AS
quest_0_started_count
        FROM quest_start
        WHERE CAST(SUBSTR(quest, INSTR(quest, '_') + 1) AS integer) = 0
        GROUP BY 1
),
raw AS ( -- funnel calculation
        SELECT
                CAST(SUBSTR(qs.quest, INSTR(qs.quest, '_') + 1) AS integer) AS
quest_num
                , (SELECT quest_0_started_count FROM quest_0_start) AS
quest_0_started_count
                , SUM(CASE WHEN qs.time IS NOT NULL THEN 1 ELSE 0 END) AS
quest_started_count
                , SUM(CASE WHEN qc.time IS NOT NULL THEN 1 ELSE 0 END) AS
quest_ended_count
                , SUM(CASE WHEN qs.time IS NOT NULL THEN 1 ELSE 0 END) * 1.00 / (SELECT
quest_0_started_count FROM quest_0_start) AS funnel_cr
        FROM quest_start AS qs
        LEFT JOIN quest_complete AS qc
                ON qc.user_id = qs.user_id
                AND qc.quest = qs.quest
        GROUP BY 1
```

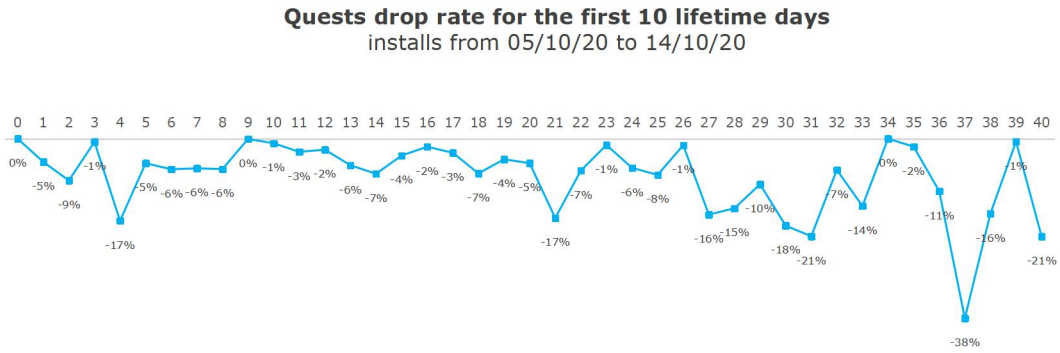
```
        UNION ALL

        -- line with count of quest_0 starts
        SELECT
                -1 AS quest_num
                , quest_0_started_count AS quest_0_started_count
                , quest_0_started_count AS quest_started_count
                , quest_0_started_count AS quest_ended_count
                , 1 AS funnel_cr
        FROM quest_0_start
)
-- add a drop rate & precent of quest ending
SELECT
        quest_num
        --, quest_0_started_count
        , quest_started_count
        , quest_ended_count
        , funnel_cr
        , quest_started_count * 1.00 / LAG(quest_started_count) OVER(ORDER BY quest_num)
- 1 AS quest_drop_rate
        , quest_ended_count * 1.00 / quest_started_count AS quest_ended_percent
FROM raw
ORDER BY 1
```

Для квестов можно смотреть стандартную воронку, на которой будет видно, что после 3-его квеста есть провал с 86% до 71% пользователей. Это не выглядит как технические проблемы, т.к. drop не такой большой. Возможно, квест затянут по времени. Или на этом квесте многие новые игроки понимают, интересна ли им игра или нет. Т.е. это может быть привычной ситуацией.



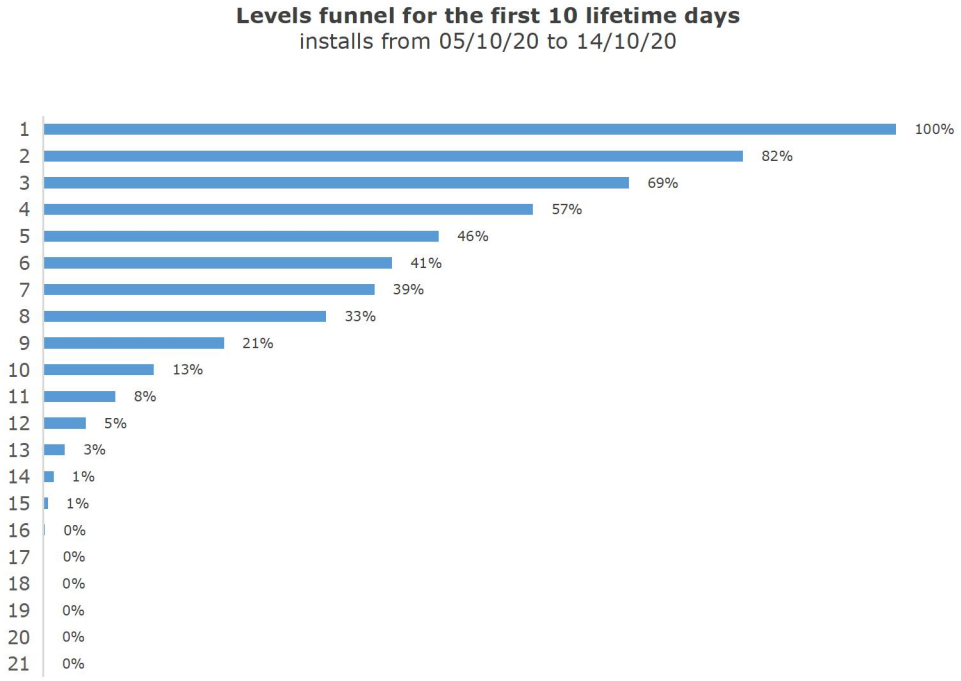
Но мне более показательным воронка становится в сочетании с drop (churn) rate, т.е. долей пользователей, которая не доходит до следующего квеста. Тут

тоже хорошо видно, что 17% пользователей не доходит до 4 квеста после 3-его. 9-ый квест - это как будто момент стабильности, что видно и на воронке, которая остаётся на одном уровне (55%). Скорее всего к этому моменту остаются те, кто действительно заинтересован в игре.
Далее видно, что что-то происходит на 20-м квесте, т.к. здесь второй drop в 17%. На воронке это тоже заметно - резкое уменьшение на 6 п.п. Либо в этот момент уходят сомневающиеся, либо этот квест сложно проходить, либо здесь начинается какая-то монетизация.
Очень беспокоит то, что происходит после 26 квеста, когда воронка явно стремилась к стабильности, но что-то привело к дальнейшему падению.

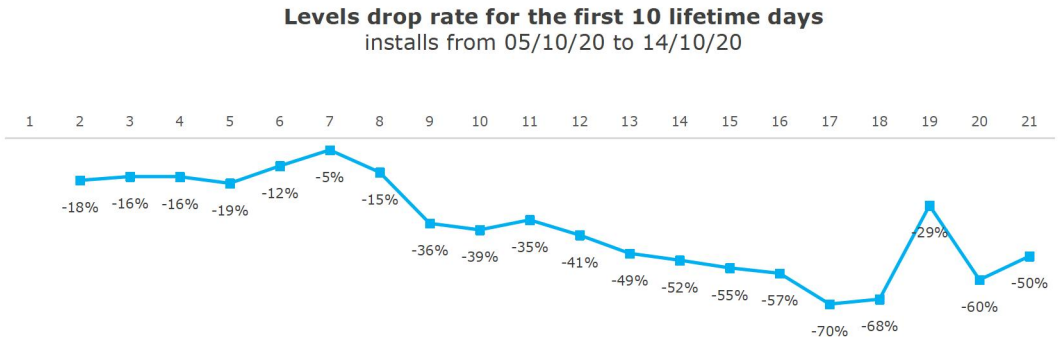


Воронка получения уровня

```
WITH
raw AS (
        SELECT
                lvl
                , COUNT(user_id) AS users_cnt
        FROM (
                SELECT
                        DISTINCT
                        i.user_id
                        , 1 AS lvl
                FROM install AS i

                UNION ALL

                SELECT
                        DISTINCT
                        lu.user_id
                        , level AS lvl
                FROM level_up AS lu
                ) AS a
        GROUP BY 1
)
SELECT
        lvl
        , MAX(users_cnt) OVER() AS users_cnt_total
        , users_cnt
        , users_cnt * 1.00 / MAX(users_cnt) OVER() AS funnel_cr
        , users_cnt * 1.00 / LAG(users_cnt) OVER(ORDER BY lvl) - 1 AS lvl_drop_rate
FROM raw
```



С 5-го уровня drop становится лучше, что может говорить о пороге для тех, кто действительно заитересован в игре. На воронке видно, что последующие 3 уровня она находится примерно на одно уровне.
На 9 уровне происходит резкий дроп с постепенным снижением. Возможно, что-то происходит на 8-9 уровне.



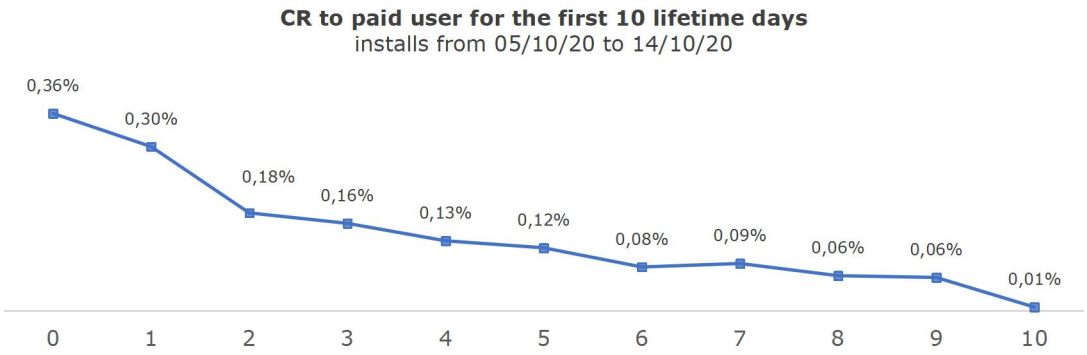
Монетизация

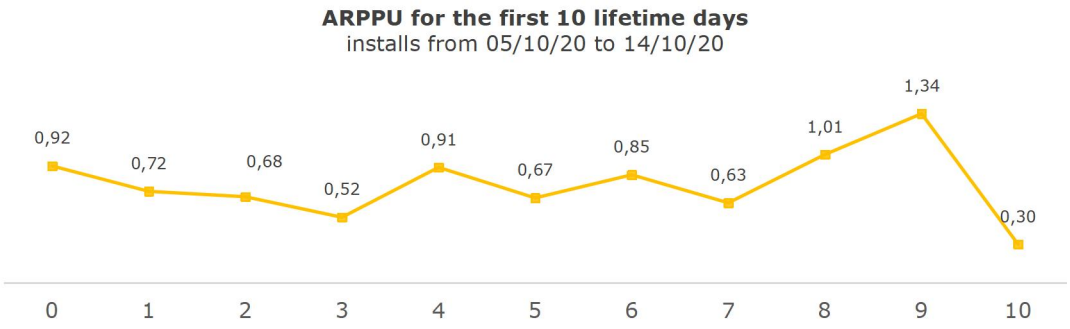
```
WITH raw_1 AS ( -- installs & sessions dates
        SELECT
                DISTINCT
                i.user_id
                , DATE(i.reg_time) AS dt
                , NULL AS amount_sum
        FROM install AS i

        UNION ALL

        SELECT
                p.user_id
                , DATE(p.time) AS dt
                , SUM(p.amount) AS amount_sum
        FROM payment AS p
        GROUP BY 1, 2
        )
, raw_2 AS ( -- lifetime calculation
```

```
        SELECT
                DISTINCT
                user_id
                , dt
                , SUM(amount_sum) OVER(PARTITION BY user_id, dt) AS amount_sum
                , MIN(dt) OVER(PARTITION BY user_id) AS reg_dt
                , julianday(dt) - MIN(julianday(dt)) OVER(PARTITION BY user_id) AS
lifetime
        FROM raw_1
        ORDER BY 1, 2
)
-- ltv & arpu
SELECT
        lifetime
        , users_total_cnt
        , paid_users_cnt
        , conversion_rate
        , SUM(amount_sum) OVER(ORDER BY lifetime) / users_total_cnt AS ltv
        , amount_sum / users_total_cnt AS arpu
        , amount_sum / paid_users_cnt AS arppu
FROM (
        -- revenue & conversion rate
        SELECT
                r.lifetime
                , a.users_total_cnt
                , COUNT(DISTINCT r.user_id) FILTER(WHERE amount_sum IS NOT NULL) AS
paid_users_cnt
                , COUNT(DISTINCT r.user_id) FILTER(WHERE amount_sum IS NOT NULL) * 1.00
/ a.users_total_cnt AS conversion_rate
                , SUM(r.amount_sum) AS amount_sum
        FROM raw_2 AS r
        LEFT JOIN ( -- installs count
                SELECT
                        lifetime
                        , COUNT(DISTINCT user_id) AS users_total_cnt
                FROM raw_2
                WHERE lifetime = 0
                GROUP BY 1
                ) AS a
        GROUP BY 1
        ) AS r
```

Если посмотреть на конверсию в платящих пользователей, то видно, что на 2-ой день лайфтайма она падает. Предполагаю, что первые два дня - это важный период, когда пользователь лучше конвертируется в платящего. Возможно, в последующие дни требуют больших усилий, чтобы сконвертировать пользователя (акции или, наоборот, немного усложнить прохождение, что требовало бы доп. ресурсов).

Судя по ARPPU на 7-8 дней лайфтайма пользователи, возможно, сталкиваются со сложность прохождения, что увеличивает денежные вливания. Но на 10-ый день происходит резкий drop показателя, как и CR. Если исключить технические проблемы, то, вероятнее, игроки разочаровываются в игре и уходят (снижение продолжительности игры на 10-в день обсуждалось выше).





Проблемные точки

Какие наиболее проблемные квесты/уровни вы бы выделили и предложили к доработкам? Почему?

Если говорить о конкретных квестах, то я бы посмотрела на 3 и 20 квесты, потому что у них drop rate = -17%, что существенно в сравнении с остальной динамикой.

Как выглядели бы целевые метрики в аб-тесте правок этих проблемных точек?

Основная метрика - это CR из начала 3/20 квеста в начало 4/21 квеста.
Как дополнительную метрику можно установить среднюю продолжительность прохождения 3/20 квеста.

Какие участки игры вы бы предложили изменить и протестировать по итогу анализа? Почему?

Помимо квестов я бы повнимательнее посмотрела, что происходит с пользователями на 9 день лайфтайма. Судя по анализу, что это платящие лояльные пользователи, но по какой-то причине они не доходят до 10-го дня. Здесь можно попробовать сопоставить квесты, платежи и дни лайфтайма и найти конкретную проблемную точку. После этого уже можно обсуждать возможные изменения.

Какие ещё метрики и внутриигровые фичи вы бы отслеживали / анализировали в первую очередь?
Дополните таблицу выше событиями, которые вам в этом помогут.
Опишите стратегию дальнейшего анализа на основании предложенных вами событий.

В принципе, для любого приложения основная метрика - это DAU / MAU. Чем больше активных пользователей возвращается, тем большее количество пользователей будут переходить в платящих, а выручка будет увеличиваться даже если конверсия в платящего будет оставаться на одном уровне. Ещё DAU моментально реагирует на технические сбои: если есть резкое падение, то явно где-то что-то произошло. С этой целью можно смотреть и на количество сессий, и на среднюю продолжительность сессии. Для отслеживания этих метрик вполне достаточно уже имеющейся таблицы sessions.

Поскольку продвижение по квестам зависит от наличия энергии, то можно отслеживать средний расход энергии в день/сессию/квест/конкретное действие (в квест, наверное, будет самым показательным, т.к. будет хорошо видно, какие квесты следует упростить, а какие усложнить). Для этого нужно трекать действия расхода и пополнения энергии и время его совершения для каждого игрока. По времени можно будет это сопоставить с квестами/уровнями/покупками.

Доля открытия приложения после получения нотификации или доля игроков, совершивших действие, указанное в нотификации, - хорошая метрика для отслеживания заинтересованности игроков. Например, нотификации о восполнении энергии способствуют игроку зайти в приложение и совершить в нём несколько действий. Для расчёта нужно трекать получателя, время отправки нотификации, её текст и статус (отправлено, ошибка, свайп, нижатие).

Для анализа внутриигровых покупок нужно добавить данные о конкретной покупке: что куплено, кто купил, стоимость, количество. Это поможет отслеживать популярные и непопулярные позиции и работать над улучшением ассортимента.

В целом, стратегия анализа примерно одинаковая для любых метрик и событий:
- определение конкретных метрик для отслеживания/анализа;
- сбор данных и их чистка;
- собственно анализ (посмотреть на картину в целом, разбить на когорты, постараться найти особенности);
- формулирование выводов и предложений;
- обсуждение возможных изменений, их тест и внедрение.

Запросы по SQL

1. Необходимо рассчитать ретеншн по первым 7 дням. За активность игрока в определенный день считать старт сессии.
Результат: день - процент активных игроков.

```
WITH raw_1 AS ( -- installs & sessions dates
        SELECT
                DISTINCT
                i.user_id
                , DATE(i.reg_time) AS dt
        FROM install AS i

        UNION ALL

        SELECT
                sc.user_id
                , DATE(sc.open_time) AS dt
        FROM session_close AS sc
        GROUP BY 1, 2
        )
, raw_2 AS ( -- lifetime calculation
        SELECT
                DISTINCT
                user_id
                , dt
                , MIN(dt) OVER(PARTITION BY user_id) AS reg_dt
                , julianday(dt) - MIN(julianday(dt)) OVER(PARTITION BY user_id) AS
lifetime
        FROM raw_1
        ORDER BY 1, 2
)
-- retention rate
SELECT
```

```
        r.lifetime
        --, a.users_total_cnt
        --, COUNT(DISTINCT r.user_id) AS users_cnt
        , COUNT(DISTINCT r.user_id) * 1.00 / a.users_total_cnt AS retention_rate
FROM raw_2 AS r
LEFT JOIN ( -- installs count
        SELECT
                lifetime
                , COUNT(DISTINCT user_id) AS users_total_cnt
        FROM raw_2
        WHERE lifetime = 0
        GROUP BY 1
        ) AS a
WHERE r.lifetime <= 7
GROUP BY 1
```

2. Необходимо рассчитать среднее кол-во сессий на активного игрока по дням с момента регистрации.
Результат: день - среднее кол-во сессий.

```
WITH raw_1 AS ( -- installs & sessions dates
        SELECT
                DISTINCT
                i.user_id
                , DATE(i.reg_time) AS dt
                , NULL AS sessions_cnt
        FROM install AS i

        UNION ALL

        SELECT
                sc.user_id
                , DATE(sc.open_time) AS dt
                , COUNT(sc.open_time) AS sessions_cnt
        FROM session_close AS sc
        GROUP BY 1, 2
        )
, raw_2 AS ( -- lifetime calculation
        SELECT
                DISTINCT
                user_id
                , dt
                , SUM(sessions_cnt) OVER(PARTITION BY user_id) AS sessions_cnt
                , MIN(dt) OVER(PARTITION BY user_id) AS reg_dt
                , julianday(dt) - MIN(julianday(dt)) OVER(PARTITION BY user_id) AS
lifetime
        FROM raw_1
        ORDER BY 1, 2
)
-- avg sessions cnt
SELECT
        r.lifetime
        --, COUNT(DISTINCT r.user_id) AS users_cnt
        --, SUM(sessions_cnt) AS sessions_cnt
        , SUM(sessions_cnt) / COUNT(DISTINCT r.user_id) AS avg_sessions_cnt
FROM raw_2 AS r
GROUP BY 1
```

3. Необходимо рассчитать среднюю длительность сессии в зависимости от порядкового номера сессии.
Результат: номер сессии - средняя продолжительность.

```
SELECT
        session_rn
        , ROUND(AVG(duration_min)) AS avg_duration_min
FROM (
        SELECT
                user_id
                , ROW_NUMBER() OVER(PARTITION BY user_id ORDER BY open_time) AS
session_rn
                , duration / 60 AS duration_min
        FROM session_close AS sc
        ) AS r
```

```
GROUP BY 1
ORDER BY 1
```

4. Необходимо рассчитать среднее кол-во сессий перед первым платежом.
Результат: среднее кол-во сессий перед первым платежом.

```
WITH
first_payment AS (
        SELECT
                user_id
                , MIN(time) AS first_payment_ts
        FROM payment AS p
        GROUP BY 1
)
SELECT
        AVG(REPLACE(session_cnt, 0, 1)) AS avg_session_cnt
FROM (
        SELECT
                p.user_id
                , COUNT(sc.user_id) AS session_cnt
        FROM first_payment AS p
        LEFT JOIN session_close AS sc
                ON sc.user_id = p.user_id
                AND sc.open_time < p.first_payment_ts
        GROUP BY 1
        ) AS raw
```

5. Необходимо рассчитать кол-во сконвертированных игроков в плательщики по уровням.
Результат: level - кол-во игроков, кто совершил первый платеж на данном уровне.

```
WITH
first_payment AS (
        SELECT
                user_id
                , MIN(time) AS first_payment_ts
        FROM payment AS p
        GROUP BY 1
),
levels AS (
        SELECT
                user_id
                , lvl
                , lvl_start
                , COALESCE(LEAD(lvl_start) OVER(PARTITION BY user_id ORDER BY lvl),
datetime('now')) AS lvl_end
        FROM (
                SELECT
                        user_id
                        , 1 AS lvl
                        , reg_time AS lvl_start
                FROM install AS i

                UNION ALL

                SELECT
                        user_id
                        , level AS lvl
                        , time AS lvl_start
                FROM level_up
                ) AS l
)
SELECT
        lvl
        , COUNT(DISTINCT fp.user_id) AS first_payment_users_cnt
FROM levels AS lv
LEFT JOIN first_payment AS fp
        ON fp.user_id = lv.user_id
        AND fp.first_payment_ts BETWEEN lv.lvl_start AND lv.lvl_end
GROUP BY 1
```

6. *Необходимо рассчитать суммарное ревенью игроков в зависимости от квестов, которые были активны в момент совершения покупки. При наличии нескольких активных квестов разделить ревенью в равной степени на каждый квест.
Результат: quest - суммарное кол-во ревенью.

```
WITH
raw AS (
	SELECT
		qs.user_id
		, qs.quest
		, qs.time AS quest_start_ts
		, COALESCE(qc.time, datetime('now')) AS quest_end_ts
		, p.time AS payment_ts
		, p.amount AS payment_amount
	FROM quest_start AS qs
	LEFT JOIN quest_complete AS qc
		ON qc.user_id = qs.user_id
		AND qc.quest = qs.quest
	LEFT JOIN payment AS p
		ON p.user_id = qs.user_id
		AND p.time BETWEEN qs.time AND COALESCE(qc.time, datetime('now'))
),
several_quests_payments AS (
	SELECT
		user_id
		, payment_ts
		, payment_amount / COUNT(quest) AS one_quest_payment_amount
	FROM raw
	WHERE payment_ts IS NOT NULL
	GROUP BY 1, 2
	HAVING COUNT(quest) > 1
)
SELECT
	raw.quest
	, SUM(COALESCE(sqp.one_quest_payment_amount, raw.payment_amount)) AS revenue
FROM raw
LEFT JOIN several_quests_payments AS sqp
	ON sqp.user_id = raw.user_id
	AND sqp.payment_ts = raw.payment_ts
GROUP BY 1
ORDER BY CAST(SUBSTR(quest, INSTR(quest, '_') + 1) AS integer)
```