Case Study

by Joshua Fan June 08 2025

▼ Product Analysis

1. Quick overview of what's happening

- **Observation**: A lower retention rates since a month or two.
- **What I found:**

  The analysis reveals that the retention problem is not a single issue but a dual problem that affects users at different stages and for different reasons:

  - **A "Sudden Death" Problem:** The most immediate problem is a big drop in the first 3 days. about 72% of all users who are gone by Day 30 have already churned within the first 3 days. This is mainly caused by:
    - **Onboarding friction**: Found by users with an "Unknown" age, who are 6x more likely to churn immediately.
    - **Technical failures**: Focused on legacy iOS devices, which see churn rates up to 4x higher than average.
    - Systemic platform gaps: A consistent disadvantage for the Android OS as a whole.
  - **An "Accelerating Slow Death" Problem:** For users who survive the first week, the churn rate increased over time. This problem is most seen for our 18-21+ demographic and is driven by not being able to deliver on the some features:
    - **Lack of social validation**: This age group's retention is dependent on receiving a "Meaningful Interaction" on Day0. Without it, their long-term retention drops.
    - **Weak first impression**: Across all age groups, a user's first session length is an early predictor of their long-term fate. Users who spend less than 5 minutes in their first session almost never become retained, long-term users.

2. What exactly is the problem?

There are two major problems identified from the dataset:

a. Sudden churn right after signing up
b. Leaky bucket of the remaining users

Here's a breakdown:

> ⚠️ **Sudden churn right after signing up**

The headline number we are looking at here is: 71.6% churned users were long gone within the first 72 hours of the user experience. It means that every 100 user who are gone by the end of the month, nearly 72 of them had already made the decision to leave within the first 3 days.

The challenge for the big majority of the users is won or lost almost immediately.

⇒ Think next: What are the huge impacts right at the start of the user journey?

> ⚠️ **The leaky bucket of remaining users (accelerating slow death)**

This is probably the more worrying finding. Look at the churn rate for users who "survived" the initial period:

- Churn rate between D3 and D7: 41.6%
- Churn rate between D7 and D14: 47.3%
- Churn rate between D14 and D30: 56.3%

The churn rate is not decreasing over time, it is **accelerating**. The longer a "survivor" stays on the app, the more likely they are to leave in the next milestone.

⇒ Think next: The ability to retain users on the product gets worse as they get deeper into the app experience. The initial engagements/attractions are not translating into a long term sustainable value for the user.

❌: rules out the thought of - if we can just get them past the first week then the users will stick around

- **Ideas and hypothesis to follow:**
  - catastrophic impacts right at the start:
    - hardware issues - apps do not work properly on certain models
    - initial hook is not appealing to users - no meaningful interactions with other users perhaps
    - failure of long term engagement:
      - novelty of daily notification/post wears off - interactions and posting becomes a chore
      - network failure cascading - when one or two friends in a user’s small circle stop posting, the value of the app for the user drops and that churn turned into other friends churning as well
- Query of the analysis - 2 types of churning situations

```
WITH data AS (
 SELECT
  d0.*,
  retention.d0,
  d3,
  d7,
  d14,
  d30
 FROM
  `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
 JOIN
  `dl_data_analyst_case_study.retention_br_bucket` retention
 ON
  d0.keychain_udid = retention.keychain_udid),
 base_data AS (
 SELECT
```

```
  keychain_udid AS id,
  install_at,
  CASE
    WHEN CAST(age AS integer) <= 20 THEN age
    WHEN age IS NULL THEN 'Unknown'
    ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS string)
END
  AS age_group,
IF
  (country IN ('US',
      'FR',
      'GB',
      'IT',
      'CA',
      'AU'), country, 'Other') AS country_segment,
  os,
  CASE
    WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
    WHEN had_meaningful = 0 THEN 'Non-Meaningful Interaction'
    ELSE 'No Reaction'
END
  AS initial_reaction_type,
  CASE
    WHEN session_length < 60 THEN 'a. <1 min'
    WHEN session_length < 300 THEN 'b. 1-5 mins'
    WHEN session_length < 600 THEN 'c. 5-10 mins'
    WHEN session_length < 900 THEN 'd. 10-15 mins'
    WHEN session_length < 1200 THEN 'e. 15-20 mins'
    ELSE 'f. 20+ mins'
END
  AS session_length_bucket,
  -- split(device_model, ',')[0] as device_model,
  CASE
    WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
    WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
    WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
    WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
    WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
```

```
      WHEN device_model LIKE 'moto g%' THEN 'moto g'
      WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_model,
',')[0]
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_model,
',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
      ELSE locale
  END
    AS locale,
    d3,
    d7,
    d14,
    d30,
  FROM
    DATA )
SELECT
  --of the users who eventually churned wwhen did they already not
presenting in the app
  ROUND(SAFE_DIVIDE(COUNTIF(d7 = 0
        AND d3 = 0), COUNTIF(d7 = 0)) * 100, 1) AS pct_of_d7_churn_g
one_by_d3,
  ROUND(SAFE_DIVIDE(COUNTIF(d14 = 0
        AND d7 = 0), COUNTIF(d14 = 0)) * 100, 1) AS pct_of_d14_churn
_gone_by_d7,
  ROUND(SAFE_DIVIDE(COUNTIF(d30 = 0
        AND d14 = 0), COUNTIF(d30 = 0)) * 100, 1) AS pct_of_d30_chur
n_gone_by_d14,
  ROUND(SAFE_DIVIDE(COUNTIF(d30 = 0
        AND d3 = 0), COUNTIF(d30 = 0)) * 100, 1) AS pct_of_d30_churn
_gone_by_d3, -- churn on d30 but gone by d3
  --of the users who were active at one milestone the percentage lef
t by the next one
  ROUND(SAFE_DIVIDE(COUNTIF(d3 = 1
        AND d7 = 0), COUNTIF(d3 = 1)) * 100, 1) AS churn_rate_d3_to_d
```

```
7,
  ROUND(SAFE_DIVIDE(COUNTIF(d7 = 1
        AND d14 = 0), COUNTIF(d7 = 1)) * 100, 1) AS churn_rate_d7_to_
d14,
  ROUND(SAFE_DIVIDE(COUNTIF(d14 = 1
        AND d30 = 0), COUNTIF(d14 = 1)) * 100, 1) AS churn_rate_d14_t
o_d30
FROM
  base_data
WHERE
  1=1
```

3. Analytical Deep Dive

Part One: Explore a profile of the sudden churn victim

▼ Query of the analysis

```
WITH DATA AS (
  SELECT
    d0.*,
    retention.d0,
    d3,
    d7,
    d14,
    d30
  FROM
    `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
  JOIN
    `dl_data_analyst_case_study.retention_br_bucket` retention
  ON
    d0.keychain_udid = retention.keychain_udid),
  base_data AS (
  SELECT
    keychain_udid AS id,
    install_at,
    CASE
```

```
    WHEN CAST(age AS integer) <= 20 THEN age
    WHEN age IS NULL THEN 'Unknown'
    ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS string)
  END
   AS age_group,
  IF
   (country IN ('US',
     'FR',
     'GB',
     'IT',
     'CA',
     'AU'), country, 'Other') AS country_segment,
   os,
   CASE
    WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
    WHEN had_meaningful = 0 THEN 'Non-Meaningful Interaction'
    ELSE 'No Reaction'
  END
   AS initial_reaction_type,
   CASE
    WHEN session_length < 60 THEN 'a. <1 min'
    WHEN session_length < 300 THEN 'b. 1-5 mins'
    WHEN session_length < 600 THEN 'c. 5-10 mins'
    WHEN session_length < 900 THEN 'd. 10-15 mins'
    WHEN session_length < 1200 THEN 'e. 15-20 mins'
    ELSE 'f. 20+ mins'
  END
   AS session_length_bucket,
   -- split(device_model, ',')[0] as device_model,
   CASE
    WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
    WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
    WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
    WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
    WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
    WHEN device_model LIKE 'moto g%' THEN 'moto g'
    WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_model,
',')[0]
```

```
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_model,
',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
      ELSE locale
  END
    AS locale,
    d3,
    d7,
    d14,
    d30,
  FROM
    DATA ),
    d3_total_counts AS (
  SELECT
    d3,
    COUNT(id) AS total_count
  FROM
    base_data
  GROUP BY
    d3
)
SELECT
  age_group,
  -- country_segment,
  os,
  device_model_family,
  -- locale,
  d3,
  COUNT(id) AS cohort_size,
  ROUND(COUNT(id) / (SELECT total_count FROM d3_total_counts
WHERE d3 = base_data.d3) * 100, 1) AS pct_of_d3_status_group,
FROM
  base_data
GROUP BY
```

```
GROUPING SETS
  ( ()
    ,
    (os,
      d3),
    (device_model_family,
      d3),
    (age_group,
      d3)
    -- (country_segment,
    --   d3),
    -- (locale,
    --   d3)
    )
-- HAVING
--   COUNT(id) >= (
--   SELECT
--     COUNT(id)*0.01
--   FROM
--     base_data)
ORDER BY
  os,
  device_model_family,
  age_group,
  -- country_segment,
  -- locale,
  d3
```

In this part, we look into the users who churn within the first three days (sudden churn victim). This initial drop of users is not originated from one single problem but driven by two factors.

The risk factors are:

Profile A: The "Unknown" Age

The single factor of immediate churn is a user's failure to provide their age during onboarding.

> Finding: This group accounted for 30% of our D3 churn, despite only making up 5% of our retained users. They overly represented in the churn group by a factor of 6 times.

▼ Some food for thought

A quick note here to bring you through the process and analysis. Initially, I had a few hypothesis for the unknown age group churn users.

- Privacy conscious users: They are skeptical individuals, likely influenced by GDPR or other privacy related policies. However, given that our main user base is between age group of 13-18:
    - Age group 13-18 are not likely to have this privacy conscious mindset I reckon
    - Unless this group is disproportionately coming from age 20-30s and using certain devices. (really hypothesising/guessing here)
- Frustrated users: Users who encountered friction with a confusing UI during the onboarding and eventually led to the unwillingness of using the app.

I do not think privacy could be an issue here for the users of they are intended to use the app already in the first place. Therefore, I went in analyse the devices and OS profiles.

The Root Cause: Onboarding Issues on Specific Devices

▼ Query of the analysis

```
WITH DATA AS (
 SELECT
  d0.*,
  retention.d0,
  d3,
  d7,
  d14,
  d30
 FROM
  `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
 JOIN
  `dl_data_analyst_case_study.retention_br_bucket` retention
```

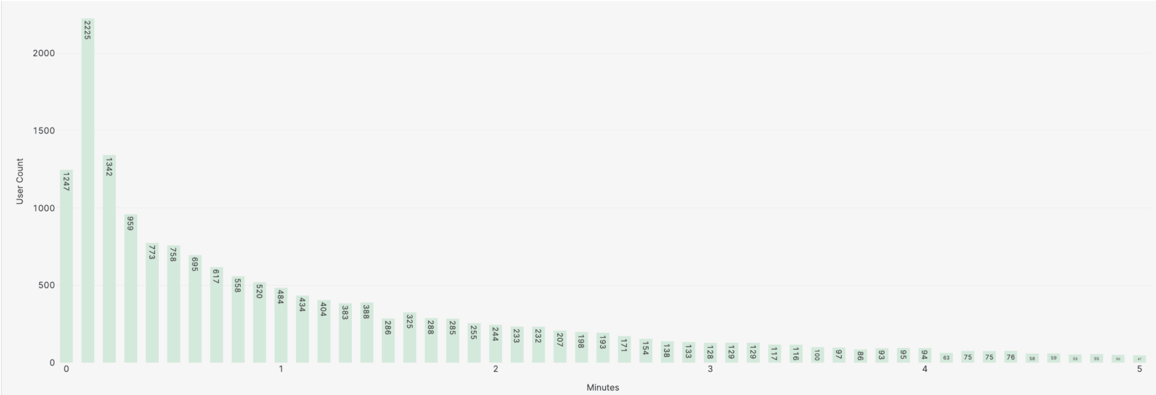
```
ON
  d0.keychain_udid = retention.keychain_udid),
base_data AS (
SELECT
  keychain_udid AS id,
  install_at,
  CASE
    WHEN CAST(age AS integer) <= 20 THEN age
    WHEN age IS NULL THEN 'Unknown'
    ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS string)
END
  AS age_group,
IF
  (country IN ('US',
      'FR',
      'GB',
      'IT',
      'CA',
      'AU'), country, 'Other') AS country_segment,
  os,
  CASE
    WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
    WHEN had_meaningful = 0 THEN 'Non-Meaningful Interaction'
    ELSE 'No Reaction'
END
  AS initial_reaction_type,
  CASE
    WHEN session_length < 60 THEN 'a. <1 min'
    WHEN session_length < 300 THEN 'b. 1-5 mins'
    WHEN session_length < 600 THEN 'c. 5-10 mins'
    WHEN session_length < 900 THEN 'd. 10-15 mins'
    WHEN session_length < 1200 THEN 'e. 15-20 mins'
    ELSE 'f. 20+ mins'
END
  AS session_length_bucket,
  -- split(device_model, ',')[0] as device_model,
  CASE
    WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
```

```
      WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
      WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
      WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
      WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
      WHEN device_model LIKE 'moto g%' THEN 'moto g'
      WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_model,
',')[0]
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_model,
',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
      ELSE locale
  END
    AS locale,
    -- ROUND(session_length/60,1) as session_length_minutes,
    d3,
    d7,
    d14,
    d30,
  FROM
    DATA ),
  d3_churn_problem_group AS (
  SELECT
    *
  FROM
    base_data
  WHERE
    age_group = 'Unknown'
    AND d3 = 0
    AND initial_reaction_type = 'No Reaction'
    AND session_length_bucket IN ('a. <1 min',
      'b. 1-5 mins' ) )
SELECT
  -- country_segment,
  -- locale,
```

```
  os,
  device_model_family,
  -- session_length_bucket,
  COUNT(id) AS cohort_size,
  ROUND( COUNT(id)/ (
    SELECT
      COUNT(id)
    FROM
      d3_churn_problem_group) * 100, 1 ) AS pct_of_d3_churn_proble
m_group
FROM
  d3_churn_problem_group
GROUP BY
GROUPING SETS
  (
    -- (country_segment),
    -- (locale),
    (os),
    (device_model_family)
    -- (session_length_bucket)
    )
ORDER BY
  os,
  device_model_family,
  cohort_size DESC
```

To find the source of this big chunk of churn users, I filtered the highest risk segment:

> "Unknown Age" users who had no interactions with other users and left in under 5 minutes. This group accounted for **25% of our D3 churn**.



A distribution by session length of [‘Unknown’ age + No Interaction + D3 churned] users ⇒ mostly stayed very shortly on the app

The device and OS of this group showed the problem:

> Finding: Within this churn group, **Android users are significantly accounted**, making up 42% of the group despite being only about 32% of our total user base. Similarly, non-standard "Other models"(mainly Android devices) are also overly represented.

- What I think: On the other hand, **iOS is under represented** in this group.
  ⇒ The problem is not universal. The onboarding flow and the start of the user journey is systematically failing at a higher rate on Android. Likely a systemic disadvantage.

Profile B: The Legacy iOS Problem
I isolated a noticeable problem on specific older iPhone models.

> Finding: Users on an **iPhone8** are overly represented in the churned group by 4 times.

- What I think: This is not an iOS problem, as analysed, other iPhone models do not over represent in the churn groups, and in fact, modern iPhones consistently show better than average retention.
  ⇒ There is a severe technical or performance issue on older hardware that creates a user experience so poor that it causes immediate abandonment and leave the app.

A Note on the "Post-Teen" Age Cliff: What I think is An Early Warning Sign
I do not consider what I discovered about age is a sudden death factor but

we see the clear beginnings of a challenge with users entering their twenties.

> Finding: Users in their twenties (age group 21 which indicates 21-29 years old) are the first cohort to show a D3 retention rate (30%) that is significantly below the platform average (34.6%).

- What I think: While most users aged 18-20 survive the first three days, this drop-off for the 21+ group is an early indicator of a mismatch.
  ⇒ I see this as an initial friction that, as we will explore next, is the tip of the iceberg for a much larger, long term retention problem.

Part Two: WHY Are They Churning in a long term?

▼ Query of this analysis - overall retention rate

```
SELECT
  ROUND(AVG(d3),3) AS d3_retention,
  ROUND(AVG(d7),3) AS d7_retention,
  ROUND(AVG(d14),3) AS d14_retention,
  ROUND(AVG(d30),3) AS d30_retention,
FROM
  `dl_data_analyst_case_study.retention_br_bucket`
```

While in the first part, I detailed the possible issues that cause users to churn within the first 72 hours, the analysis shows a second, equally important problem: **users who survive this initial filter still churn at an accelerating rate.**

Issue A: The "Age Cliff" and the Need for Interaction

This "Slow Death" can be seen clearly among the post teen/young adult audience (ages 18-21+), creating an "Age Cliff" in long-term retention. What I found is that the root cause of this is not technical but behavioural. As the age goes up, their reason for staying on the platform shifts.

Retention Becomes Dependent on "Meaningful Interaction"

The data shows a clear pattern: the older a user is, the more important a "Meaningful Interaction" on Day 0 becomes for their long term retention.

- For the major teen demographic (13-17): Their D3 retention is high (45-46%) across the board. Even with "No Reaction," their retention stays within a good 22-27%. And so does the D30 retention where we see a higher than average rate.

▼ Query of this analysis

```
-- age group retention
WITH DATA AS (
  SELECT
    d0.*,
    retention.d0,
    d3,
    d7,
    d14,
    d30
  FROM
    `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
  JOIN
    `dl_data_analyst_case_study.retention_br_bucket` retention
  ON
    d0.keychain_udid = retention.keychain_udid),
  base_data AS (
  SELECT
    keychain_udid AS id,
    install_at,
    CASE
      WHEN CAST(age AS integer) <= 20 THEN age
      WHEN age IS NULL THEN 'Unknown'
      ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS str
ing)
  END
    AS age_group,
  IF
    (country IN ('US',
        'FR',
```

```
        'GB',
        'IT',
        'CA',
        'AU'), country, 'Other') AS country_segment,
    os,
    CASE
      WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
      WHEN had_meaningful = 0 THEN 'Non-Meaningful Interactio
n'
      ELSE 'No Reaction'
  END
    AS initial_reaction_type,
    CASE
      WHEN session_length < 60 THEN 'a. <1 min'
      WHEN session_length < 300 THEN 'b. 1-5 mins'
      WHEN session_length < 600 THEN 'c. 5-10 mins'
      WHEN session_length < 900 THEN 'd. 10-15 mins'
      WHEN session_length < 1200 THEN 'e. 15-20 mins'
      ELSE 'f. 20+ mins'
  END
    AS session_length_bucket,
    -- split(device_model, ',')[0] as device_model,
    CASE
      WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
      WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
      WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
      WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
      WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
      WHEN device_model LIKE 'moto g%' THEN 'moto g'
      WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_m
odel, ',')[0]
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_mod
el, ',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
```

```
    ELSE locale
  END
    AS locale,
    d3,
    d7,
    d14,
    d30,
  FROM
    DATA ),
    d3_total_counts AS (
  SELECT
    d3,
    COUNT(id) AS total_count
  FROM
    base_data
  GROUP BY
    d3
)
SELECT
  age_group,
  -- initial_reaction_type,
  -- d3,
  COUNT(id) AS cohort_size,
  ROUND(AVG(CAST(d3 AS FLOAT64)) * 100, 2) AS d3_retention
_pct,--change this to d30 for one month retention
  -- ROUND(COUNT(id) / (SELECT total_count FROM d3_total_co
unts WHERE d3 = base_data.d3) * 100, 1) AS pct_of_d3_status_g
roup,
FROM
  base_data
WHERE
1=1
AND(
  age_group != 'Unknown'
and
cast(age_group as integer) < 31 -- remove 30s and above user d
ue to small cohort size
)
```

```
GROUP BY
GROUPING SETS
  ( (),
    (age_group)
    -- (age_group, initial_reaction_type)
    )
-- HAVING
--   COUNT(id) >= (
--   SELECT
--     COUNT(id)*0.01
--   FROM
--     base_data)
ORDER BY
  --  initial_reaction_type,
   age_group
```

```
-- age group and initial reaction type retention
WITH DATA AS (
  SELECT
    d0.*,
    retention.d0,
    d3,
    d7,
    d14,
    d30
  FROM
    `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
  JOIN
    `dl_data_analyst_case_study.retention_br_bucket` retention
  ON
    d0.keychain_udid = retention.keychain_udid),
  base_data AS (
  SELECT
    keychain_udid AS id,
    install_at,
    CASE
      WHEN CAST(age AS integer) <= 20 THEN age
      WHEN age IS NULL THEN 'Unknown'
```

```
      ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS str
ing)
  END
    AS age_group,
  IF
    (country IN ('US',
        'FR',
        'GB',
        'IT',
        'CA',
        'AU'), country, 'Other') AS country_segment,
    os,
    CASE
      WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
      WHEN had_meaningful = 0 THEN 'Non-Meaningful Interactio
n'
      ELSE 'No Reaction'
  END
    AS initial_reaction_type,
    CASE
      WHEN session_length < 60 THEN 'a. <1 min'
      WHEN session_length < 300 THEN 'b. 1-5 mins'
      WHEN session_length < 600 THEN 'c. 5-10 mins'
      WHEN session_length < 900 THEN 'd. 10-15 mins'
      WHEN session_length < 1200 THEN 'e. 15-20 mins'
      ELSE 'f. 20+ mins'
  END
    AS session_length_bucket,
    -- split(device_model, ',')[0] as device_model,
    CASE
      WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
      WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
      WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
      WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
      WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
      WHEN device_model LIKE 'moto g%' THEN 'moto g'
      WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_m
odel, ',')[0]
```

```
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_mod
el, ',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
      ELSE locale
  END
    AS locale,
    d3,
    d7,
    d14,
    d30,
  FROM
    DATA ),
    d3_total_counts AS (
  SELECT
    d3,
    COUNT(id) AS total_count
  FROM
    base_data
  GROUP BY
    d3
)
SELECT
  age_group,
  initial_reaction_type,
  -- d3,
  COUNT(id) AS cohort_size,
  ROUND(AVG(CAST(d3 AS FLOAT64)) * 100, 2) AS d3_retention
_pct, --change this to d30 for one month retention
  -- ROUND(COUNT(id) / (SELECT total_count FROM d3_total_co
unts WHERE d3 = base_data.d3) * 100, 1) AS pct_of_d3_status_g
roup,
FROM
  base_data
WHERE
```

```
1=1
AND(
  age_group != 'Unknown'
and
cast(age_group as integer) < 31 -- remove 30s and above user d
ue to small cohort size
)
GROUP BY
GROUPING SETS
  ( (),
    -- (age_group)
    (age_group, initial_reaction_type)
    )
-- HAVING
--   COUNT(id) >= (
--   SELECT
--     COUNT(id)*0.01
--   FROM
--     base_data)
ORDER BY
   initial_reaction_type,
   age_group
```

- For the post teen/young adult demographic (18-21+): This dynamic flips entirely. Their retention becomes almost entirely dependent on the quality of their first social interactions.

  > For a 15-year-old, a meaningful interaction gives a **32 point** lift in D3 retention over having no reaction (from 24.4% to 56.1%).

  > For users in their 20s, that lift increases to a good **37 point** lift (from 12.2% to 49%).

  ⇒ What I think: The reward for getting a meaningful interaction is significantly higher for users in their 20s. Their long term retention is almost dependent on it. As we have a look at the D30 retention rate, the young adult group have a lower than average retention rate (9-12% compared to average 14%). Therefore, I think meaningful interaction can be a strong early indicator that predicts the future outcome.

▼ Some Food for thought

I guess we can think of it this way: an 18-21+ year old who does not get a meaningful interaction on Day 0 does not leave immediately. They might stick around for a couple of days out of curiosity (and surviving D3), but their experience is fragile and lacks a reason to hook them on the app. Without that early social validation, they see less reason to stay and are gone by Day 7 or Day 14. So I think we can use Day 0 behaviour to predict Day 30 churn.

Issue B: The Critical First Session on the App

▼ Query of the analysis

```
WITH data AS (
 SELECT
  d0.*,
  retention.d0,
  d3,
  d7,
  d14,
  d30
 FROM
  `dl_data_analyst_case_study.d0_behaviour_br_bucket` d0
 JOIN
  `dl_data_analyst_case_study.retention_br_bucket` retention
 ON
  d0.keychain_udid = retention.keychain_udid),
 base_data AS (
 SELECT
  keychain_udid AS id,
  install_at,
  CASE
   WHEN CAST(age AS integer) <= 20 THEN age
   WHEN age IS NULL THEN 'Unknown'
   ELSE CAST(FLOOR(CAST(age AS integer) / 10) * 10 +1 AS string)
 END
  AS age_group,
 IF
```

```
    (country IN ('US',
        'FR',
        'GB',
        'IT',
        'CA',
        'AU'), country, 'Other') AS country_segment,
    os,
    CASE
      WHEN had_meaningful = 1 THEN 'Meaningful Interaction'
      WHEN had_meaningful = 0 THEN 'Non-Meaningful Interaction'
      ELSE 'No Reaction'
  END
    AS initial_reaction_type,
    -- Bucket the session length as you suggested
    CASE
      WHEN session_length < 60 THEN 'a. <1 min'
      WHEN session_length < 300 THEN 'b. 1-5 mins'
      WHEN session_length < 600 THEN 'c. 5-10 mins'
      WHEN session_length < 900 THEN 'd. 10-15 mins'
      WHEN session_length < 1200 THEN 'e. 15-20 mins'
      ELSE 'f. 20+ mins'
  END
    AS session_length_bucket,
    -- split(device_model, ',')[0] as device_model,
    CASE
      WHEN device_model LIKE 'SM-A14%' THEN 'SM-A14'
      WHEN device_model LIKE 'SM-A13%' THEN 'SM-A13'
      WHEN device_model LIKE 'SM-A03%' THEN 'SM-A03'
      WHEN device_model LIKE 'SM-A53%' THEN 'SM-A53'
      WHEN device_model LIKE 'SM-A54%' THEN 'SM-A54'
      WHEN device_model LIKE 'moto g%' THEN 'moto g'
      WHEN device_model LIKE 'iPhone%' THEN SPLIT(device_model,
',')[0]
      WHEN device_model LIKE 'iPad%' THEN SPLIT(device_model,
',')[0]
      ELSE 'Other models'
  END
    AS device_model_family,
```

```
    CASE
      WHEN locale LIKE 'en-%' THEN 'en'
      ELSE locale
  END
    AS locale,
    d3,
    d7,
    d14,
    d30,
  FROM
    data )
SELECT
  session_length_bucket,
  COUNT(id) AS cohort_size,
  ROUND(AVG(CAST(d3 AS FLOAT64)) * 100, 1) AS d3_retention_pct,
  ROUND(AVG(CAST(d7 AS FLOAT64)) * 100, 1) AS d7_retention_pct,
  ROUND(AVG(CAST(d14 AS FLOAT64)) * 100, 1) AS d14_retention_p
ct,
  ROUND(AVG(CAST(d30 AS FLOAT64)) * 100, 1) AS d30_retention_p
ct
FROM
  base_data
GROUP BY
  session_length_bucket
ORDER BY
  session_length_bucket
```

I discovered in the analysis that the first session is determining the long term fate of this user. We can see in the data that a user's retention trajectory is almost set within their first few minutes in the app. Earlier we mentioned that a short first session is not only a factor of sudden death and immediate churn, but it is also an early indicator of long-term retention failure.

To explain this, I compared the retention curve of each session length cohort against the overall average:

Milestone Days	Overall retention (%)
Day 3	34.6%
Day 7	25.9%

Day 14	19.5%
Day 30	14.1%

The data gives a clearer picture of three different user journeys based on their first session length.

- **The Unactivated Cohort who spends less than 5 minutes**

  > Finding: The users who leave in under a minute have a D3 retention of just 9.4%. Those who stayed for 1-5 minutes have a slightly better rate of 13.2%. Both are significantly lower than the overall D3 retention average of 34.6%.

  - What I think: This is already a sudden death problem and given the D30 retention for these groups is 5-6%, less than half the average. I do not think that these users are engaging with a foundation. In their short session, they probably did not complete any of the activation steps: i.e. adding close friends, testing out any possible features, and they leave before they can possibly experience a meaning interaction on the app. The path to churn in 30 days is set from the start.

- **The Mediocre Users who spend 10-15 minutes**

  > Finding: This is the first group whose retention rate begin to approach the average, with a D30 retention of 13.2% vs the 14.1% average.

- **The Good and Invested Cohort who spends 15+ minutes**

  > Finding: This group of users performs well at all the milestone days. The 20+ minute group is a great cohort, with a D3 retention of 51.5% and most importantly, a D30 retention of 20.2%, which is nearly 50% higher than the D30 overall average retention rate.

  - What I think: These users built a strong foundation in the first session. They have invested time to find friends, explore the app, and are therefore much more likely to get the social rewards that create a lower churn rate.

4. Profile of an At-Risk User

To understand churn risk, I identified two main "at-risk" profiles.

Profile 1: The Quick Appearance User (Victim of Sudden Death)

This user churns almost immediately, often within minutes of installing.

- **Who:** This user is most likely on an Android phone or an older iPhone (like an iPhone8). They also have an "Unknown" age status.
- **What Their Behaviour Looks Like:** They spend less than five minutes (often less than one) in the app. They do not interact in any meaningful way and fail to experience the core social loop.

Profile 2: The Fading User (Victim of Slow Death)

This user successfully onboards and seems engaged at first, but fades away between Day 7 and Day 30.

- **Who:** This user is typically a bit older, in the **18-21+ age range**. Their device and technical experience are generally fine.
- **What Their Behaviour Looks Like:** They survive the initial D3 churn phase and may even post or scroll for several days. However, their crucial Day 0 experience lacked a strong social hook, as they did not receive a "meaningful interaction" from a close friend.

▼ Product Feature Suggestion

Let's say based on what we have analysed, we would like to tackle the issue:

> ❓ How to ensure users in the 18-21+ age group get a "Meaningful Interaction" on Day 0 to address the "Slow Death" churn.

Product Feature Suggestion: The "First Post Nudge"

1. Description of the Feature

The idea: The first post nudge is a good way to push for a new user's first "Meaningful Interaction". Assuming that the new user has befriended with their close friends, instead of letting the friends to discover the first post themselves, this feature prompts them to welcome the new joiner at the most important moment, right after their very first post.

The problem it is trying to solve: The earlier analysis I showed that aged 18-21+ have a high long term churn rate because the retention is very much dependent on receiving “meaningful interaction” on Day 0. They need a strong and fast signal that their close friends are also on the app and ready to create this tight circle to engage. Currently we are leaving this first interaction entirely to chance.

Elaborate on the User Journey:

Let’s say we have a new user, Joshua (age 22) signs up for the app BeReal and there are two of his friends (Ben and Clara) that are already active on the app.

What triggers the feature: Joshua is prompted to post his first BeReal. He takes a quick photo of himself and what is in front of him. This is the moment of importance: He puts himself out there and is now waiting for a reaction.

“Nudge” is triggered: Once the post goes live, the system sends a very special and high prio notification to Ben and Clara. Note that this should not be a standard notification. It should be something that drives an action, for instance:

> 💬 Joshua (your super close friend) just posted his first ever BeReal!!! 🎉 Check out what he is up to and welcome him with ❤️

What Ben and Clara experience: They see this unique and visually different from standard notification on their lock screen. They tap it and instead of just opening the app to their feed, they go directly to Joshua’s first post. Ben reacted with a heart emoji and Clara commented:

> 💬 lol 😂 this can’t be real

Meaningful Interaction received: Joshua immediately receives a notification:

> 💬 Ben and Clara just showed ❤️

This completes the loop. Joshua's first post was not met with silence and non-meaningful interaction. He received quick social validation from friends and this gives him a powerful reason to come back tomorrow.

2. Events and Properties for Tracking

Here are some events to trigger:

▼ Events

- `nudge_sent`
    - Trigger: when the system sends the first post nudge to the existing user, aka Ben and Clara
    - Properties:
        - `new_user_id` : The same as the user_id of the user table. The id of the user who just made the first post, aka Joshua.
        - `recipient_user_id` : The id of the friend receiving this nudge.
        - `trigger_by` : The event name of the triggered event. In this case, we assume there is a "first_post" event.
        - `timestamp`
- `nudge_tapped` :
    - Trigger: when the recipient friend taps on the nudge notification
    - Properties:
        - `new_user_id`
        - `recipient_user_id`
        - `notification_type` : "first_post_nudge"
        - `timestamp`

- `interaction_sent`: (general `interaction_sent` event instead of `first_post_interaction_sent`)
  - Trigger: when the recipient friend sends an interaction on the first post
  - Properties:
    - `user_id` : the person sending the interaction
    - `timestamp`
    - `interaction_id` : if there is a need to analyse the interaction further we can join tables
    - `interaction_type` : comment or emoji
    - `post_author_id`
    - `source_metadata` (JSON): all the context about what triggered the interaction
      - `trigger_type`: organic or first_post_nudge
      - `triggering_user_id`: `null` if organic, or the `new_user_id` from a nudge.
      - `time_from_trigger_seconds`: `null` if organic, or the duration between `nudge_tapped` and `interaction_sent`.
- `interaction_received` (this event should be on the new user's timeline)
  - Trigger: when the recipient friend sends an interaction on the first post and `interaction_sent` was triggered
  - Properties:
    - `source`: organic or first_post_nudge
    - `time_from_post_to_interaction_seconds`
    - `interaction_type`
    - `interacting_user_id`

3. Testing & KPIs

I will go for an A/B testing to see if the new change is boosting the KPIs we are following.

The set up will be testing on all new users signing up on the platform, with the following specs:

- **A: Control Group:** The current user experience. Nothing changes.
- **B: Testing Group:** The group with the "First Post Nudge" feature enabled.
- **Duration:** I would say 2/3 weeks so that we have a significant number of users and we will track their retention the D30 of post signing up.

KPIs to track:

- **Overview Lever:**
  - **D7 Retention Rate for the 18-21+ Age Group:** I would see this as the main success metric. We are looking for a statistically significant increase in the D7 retention for this specific demographic in the treatment group compared to the control group.
  - *Other lagging KPIs to track: D14 and D30 Retention to make sure that the increase is sustained long-term. We can add this as a tooltip or secondary value if this is shown in a dashboard.*
- **Operational Lever:**

These are metrics that can help us understand why the primary KPI moved.

1. **% of New users (18-21+) receiving a D0 meaningful interaction:** This will affect directly the outcome that the feature is designed to influence. Expecting this to be a lot higher for Group B.
2. **Time to the 1st meaningful interaction:** Expecting this to decrease for Group B. This shows us how fast we can bring that social validation feeling to the new user.
3. **Funnel conversion rate of nudging:** we will see how effective nudge is at getting the friends to act.
   - `nudge_sent` ⇒ `nudge_tapped` (check the click through rate)
   - `nudge_tapped` ⇒ `interaction_sent` (check the click through rate: this is to see in-app speed)
4. **D3 retention rate:** While our main focus is on the "Slow Death" problem, we will also monitor D3 retention. A positive side effect of a better first

experience could be a reduction in "Sudden Churn" as well.

AB Test Analysis

1. The **Assumed Goal:**

The feature was designed to increase user engagement and “tackle retention issue”. In the first practice, we saw that more meaningful reactions lead to higher retention, and this is what I think we should assume.

The metrics we should follow:

1. Primary metrics: when they increase, we expect the retention to decrease.
    - `meaningful_reactions`
    - `sessions`
2. Secondary metric: even though the goal is to tackle retention issue, I think we should also expect this to go up as a downstream effect, and if not, at least keep this flat.
    - `ads_revenue`

2. Hypothesis We Make

- From a business point of view:
    - Does the new feature cause users to retain by
        - having more meaningful reactions?
        - visiting the app more frequently?
    - Does the new feature impact the ad revenue accordingly?
- Define the metrics:
    - Average of the total meaningful reactions per user over the testing period
    - Average of the total sessions per user over the testing period
    - Average of the total ad revenue over the testing period
- H0 hypothesis:

- There is **no statistically significant difference** in the
  - average number of meaningful reactions
  - average number of sessions
  - average ad revenue
- between the users in the test group and the control group.

- H1 hypothesis:
  - There is **a statistically significant difference** in the
    - average number of meaningful reactions
    - average number of sessions
    - average ad revenue
  - between the users in the test group and the control group.
- How to decide if we should reject H0?
  - We calculate the p-value using a t-test. If the p-value is less than 0.05 (95% confidence), we will reject the null hypothesis and conclude that the feature had a real effect on the said metrics.

3. First Glance at the Data

▼ Query for the analysis

```
WITH
  base_data AS (
  SELECT
    keychain_udid,
    ab_test_cohort,
    SUM(ads_revenue) AS ads_revenue,
    SUM(meaningful_reactions) AS meaningful_reactions,
    SUM(sessions) AS sessions,
    -- COUNT(DISTINCT event_date) AS active_days_per_user -- just
to make sure active days is not dropping
  FROM
    `dl_data_analyst_case_study.ab_test_br`
  GROUP BY
```

```
  1,
  2 ),
stats AS (
SELECT
  ab_test_cohort,
  'ads_revenue' AS metric,
  COUNT(ads_revenue) AS non_null_count,
  MIN(ads_revenue) AS min_val,
  ROUND(MAX(ads_revenue),2) AS max_val,
  ROUND(AVG(ads_revenue), 5) AS avg_val,
  ROUND(STDDEV(ads_revenue), 2) AS stddev_val,
  ROUND(APPROX_QUANTILES(ads_revenue, 100)[
  OFFSET
    (50)],5) AS median_val
FROM
  base_data
GROUP BY
  1,
  2
UNION ALL
SELECT
  ab_test_cohort,
  'meaningful_reactions' AS metric,
  COUNT(meaningful_reactions),
  MIN(meaningful_reactions),
  MAX(meaningful_reactions),
  ROUND(AVG(meaningful_reactions), 2),
  ROUND(STDDEV(meaningful_reactions), 2),
  APPROX_QUANTILES(meaningful_reactions, 100)[
OFFSET
  (50)]
FROM
  base_data
GROUP BY
  1,
  2
UNION ALL
SELECT
```

```
    ab_test_cohort,
    'sessions' AS metric,
    COUNT(sessions),
    MIN(sessions),
    MAX(sessions),
    ROUND(AVG(sessions), 2),
    ROUND(STDDEV(sessions), 2),
    APPROX_QUANTILES(sessions, 100)[
  OFFSET
    (50)]
  FROM
    base_data
  GROUP BY
    1,
    2 )
SELECT
  *
FROM
  stats
ORDER BY
  2,
  1
```

Is the test valid?

- Control group count: 21635 users
- Test group Count: 21751 users

⇒ Excellent news that we are comparing apples to apples. This is a nearly 50/50 split.

Metric to Metric Check

- **ads_revenue**
  - Average: Control `0.04975` vs Test `0.04714` ⇒ tiny decrease
  - Median: Control `0.00405` vs Test `0.00408` ⇒ tiny increase

> Some initial ideas: Comparing Control to Test, it does not appear to have a big impact, and rather a neutral change.

- **meaningful_reactions**
  - Average: Control `2.36` vs Test `2.51` ⇒ A clear increase. A positive 6.3% relative lift in the average number of reactions per user. This looks positive.
  - Median: Control `0` vs Test `0` ⇒ the 50th percentile, namely a typical users in both groups has zero meaningful reactions.

> Some initial ideas: The feature does not seem to succeed in getting users to start reacting meaningfully as the median user is not affected. Instead, the average increased, which seems to me that it is coming from the users at the top end of the distribution, that is those who were already having meaningful reactions are reacting more. The new feature did not activate middle ground users.

- **sessions**
  - Average: Control `85.04` vs Test `87.15` ⇒ A positive 2.5% increase.
  - Median: Control `20` vs Test `23` ⇒ A positive 15% increase.

> Some initial ideas: This looks good. The feature successfully increased average session count for the typical user (median).

4. Conduct a T-Test to decide if we want to reject H0

▼ Query for the analysis - prepare the data

```
WITH
 base_data AS (
 SELECT
  keychain_udid,
  ab_test_cohort,
  SUM(ads_revenue) AS total_revenue_per_user,
  SUM(meaningful_reactions) AS total_reactions_per_user,
  SUM(sessions) AS total_sessions_per_user
 FROM
```

```
    `dl_data_analyst_case_study.ab_test_br`
  GROUP BY
    1,
    2 )
SELECT
  ab_test_cohort,
  COUNT(keychain_udid) AS revenue_size,
  AVG(total_revenue_per_user) AS revenue_mean,
  STDDEV_SAMP(total_revenue_per_user) AS revenue_sd,
  COUNT(keychain_udid) AS reactions_size,
  AVG(total_reactions_per_user) AS reactions_mean,
  STDDEV_SAMP(total_reactions_per_user) AS reactions_sd,
  COUNT(keychain_udid) AS sessions_size,
  AVG(total_sessions_per_user) AS sessions_mean,
  STDDEV_SAMP(total_sessions_per_user) AS sessions_sd
FROM
  base_data
GROUP BY
  ab_test_cohort
```

I use the t-test calculator here for the calculation. Here are the results:

- Average Session per user ⇒ surprisingly it is not considered to be statistically significant

P value and statistical significance:

The two-tailed P value equals 0.2546

By conventional criteria, this difference is considered to be not statistically significant.

Confidence interval:

The mean of Control - Session minus Test - Session equals -2.1168000

95% confidence interval of this difference: From -5.7671256 to 1.5335256

> There is a 25% chance that we will see an addition of 2 sessions by random luck. It can be 5 more sessions per user or 1 less session per user or even 0 session. If the feature we implemented had a positive impact, then the confidence interval range would be negative, like -6 to -1.
> ⇒ Based on this, the new feature did not have a statistically

> significant impact on the number of user sessions. Even though we have a promising increase in the median and average, the variation of the data was too high for us to be confident that this change was real.

- Average meaningful reactions per user ⇒ considered to be statistically significant

P value and statistical significance:
The two-tailed P value equals 0.0130
By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:
The mean of Control - Reaction minus Test - Reaction equals -0.1452200
95% confidence interval of this difference: From -0.2601042 to -0.0303358

> Rejected the idea that the new feature led to an increase of average meaningful reactions per user is pure luck. We can be 95% sure that the increase in average meaningful reactions caused by this new feature is somewhere between 0.03 and 0.26 reactions per user.
> ⇒ Based on this, we can confidently say that the new feature works as intended and successfully causes users to have more meaningful reaction, as the test we had earlier shows that the average number of meaningful interactions per person increased from 2.36 to 2.51.

- Average ad revenue per user ⇒ considered to be statistically significant (barely below 0.05)

P value and statistical significance:
The two-tailed P value equals 0.0493
By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:
The mean of Control - Revenue minus Test - Revenue equals 0.00261000
95% confidence interval of this difference: From 0.00000197 to 0.00521803

> We can be 95% confident that the decrease in ad revenue per user caused by the newly implemented feature is somewhere around 0.00000197 and 0.0052 currency.
> ⇒ The average revenue per user went down from 0.04975 in Control to 0.04714 in Test. This represents a 5.5% relative “decrease” in average revenue per user.

Now we have the complete picture:

1. Sessions: no significant impact.
2. Meaningful reactions: a significant increase (plus 6.3% on average).
3. Ad revenue: a significant decrease (minus 5.2% on average).

Should we release this feature from test to control group?

My answer: I do not think we should release this feature to all users as it is. Instead, we should iterate on it.

Verdict "For" releasing the new feature

- It succeeded the core mission. It cause a statistically significant 6.3% increase in the average number of meaningful interactions.
- As previously discussed, meaningful interactions are a powerful driver of long term retention.
- Therefore, prioritising this metric, we are investing in long term retention to the user base and this should be more valuable than a small drop in revenue.

Verdict "Against" releasing the new feature

- It did no increase the "overall" engagement. The feature had no significant impact on average number of sessions per user. This indicates that it did not succeed to get users opening the app more frequently, leading to the failure of tackling retention issue.
- It caused a statistically significant -5.2% decrease in the average ad revenue per user. While the statistical signal was on the edge, it still meets the criteria.

Some thoughts for what's next

This new feature that was implemented led to the discovery of a mechanism that boosts meaningful interactions. However, the current implementation has an unintended and negative side effect on the revenue. So the goal now is to keep the positive effect (increase meaningful interactions) while removing the negative one.

A next move I would do in this case is to really ask: Why did driving reactions cause revenue drop?

Actions that could be helpful:

> 💡 Analyse this from the in-app user flow point of view to find out why this happened ⇒ formulate another hypothesis ⇒ propose a new feature that includes what was tested ⇒ launch a new A/B test

The goal of this new test would be to find a good feature that gives us the increase in `meaningful_reactions` without the corresponding decrease in `ads_revenue`. That would be a clear win worth releasing to all users.