Бизнес-исследование: Анализ пользовательской активности корпоративных клиентов компании 1 и образовательной эффективности на платформе Simulative

## **Введение**

Цель данного исследования — понять, как пользователи используют платформу Simulative для обучения, насколько эффективной она является, а также выявить зоны роста и пути увеличения ценности для бизнеса.

# I. Концепция исследования

### **1. Выбор метрик:**

Мы использовали комплекс ключевых метрик для оценки вовлечённости, удержания и учебных результатов:

* Активные пользователи (DAU/WAU/MAU)
* Retention (1, 7, 30+ дней)
* Успешность решения задач
* Статистика по тестам
* Анализ CodeCoins
* Времена активности

### **2. Обоснование:**

Метрики отражают полный путь пользователя: активация — вовлечённость — успех в обучении.

### **3. Гипотезы:**

* Пользователи теряют интерес в течение 30 дней
* Задачи слишком сложны и отталкивают пользователей
* Геймификация (CodeCoins) не оказывает влияния на retention
* Активность зависит от времени суток

# ход исследования:

## **📊 Аналитика 1: Общая численность и активность пользователей**

### **🎯 Цель исследования:**

Определить общее количество пользователей, зарегистрированных от корпоративного клиента, и долю активированных аккаунтов. Это позволяет:

* **преподавателям** — понять реальную вовлеченность студентов;
* **менеджерам и методистам** — оценить эффективность онбординга;
* **финансовому директору** — удостовериться, что оплаченные лицензии используются.

*SELECT*

*COUNT(\*) AS all\_users,*

*ROUND(*

*COUNT(CASE WHEN is\_active = 1 THEN 1 END)::FLOAT*

*/ NULLIF(COUNT(\*), 0) \* 100, 2*

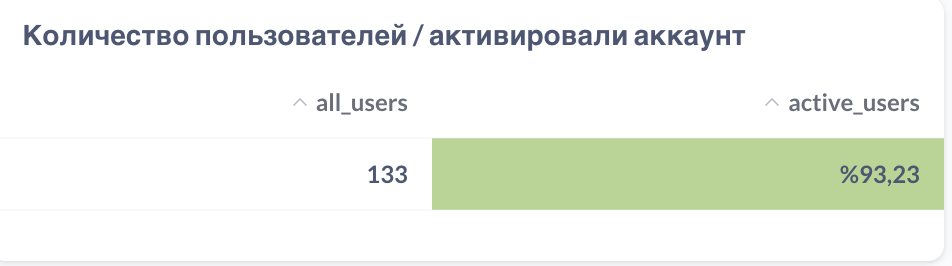
*) AS active\_user\_pct*

*FROM users*

*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{date\_joined}}]]*



### **📈 Полученные результаты:**

* **Всего пользователей (all\_users)**: 133
* **Доля активных пользователей (active\_user\_pct)**: 93.23%

### **🔍 Промежуточные выводы:**

* **Очень высокий уровень активации аккаунтов (93.23%)** демонстрирует эффективную регистрацию и первый вход.
* Только около **7%** пользователей не активировали аккаунты — можно провести точечную проверку:  
  + отправлялись ли им письма активации,
  + нет ли технических проблем с регистрацией,
  + и нужно ли дополнительно мотивировать активацию.

## **📊 Аналитика 2: MAU — Ежемесячная активность пользователей**

### **🎯 Цель исследования:**

Понять, как меняется вовлечённость пользователей на платформе по месяцам. Метрика MAU (Monthly Active Users) позволяет:

* Оценить устойчивость интереса к продукту
* Отследить периоды роста или падения активности
* Сопоставить динамику с внешними событиями (например, старт курсов, экзамены, праздники)

*SELECT*

*DATE\_TRUNC('month', entry\_at) AS month,*

*COUNT(DISTINCT user\_id) AS MAU*

*FROM UserEntry*

*LEFT JOIN users ON UserEntry.user\_id = users.id*

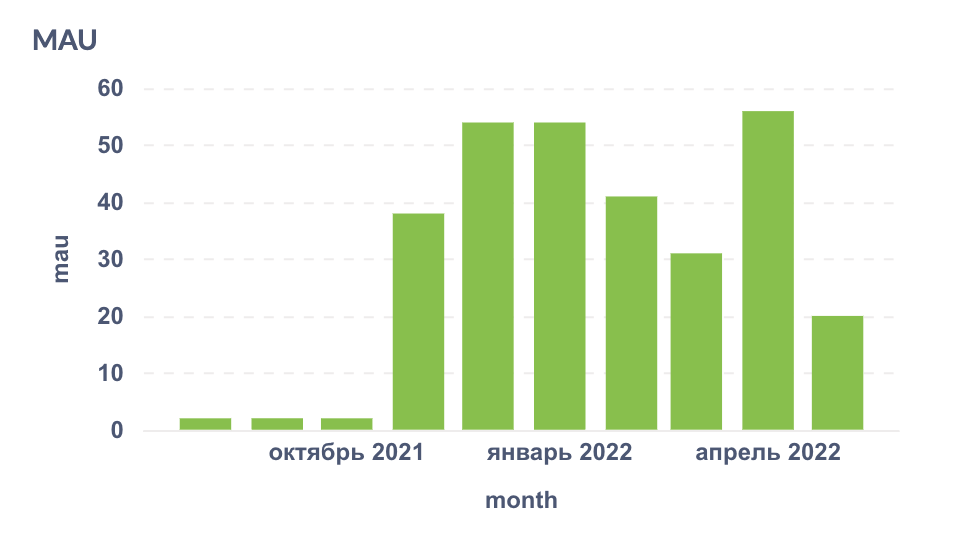
*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{entry\_at}}]]*

*GROUP BY month*

*ORDER BY month*



### **📈 Полученные результаты:**

На графике видно, как изменялась ежемесячная активность пользователей с октября 2021 по апрель 2022:

* Октябрь 2021 — низкая активность (до 5 MAU), вероятно, в этот период платформа только начинала использоваться.
* Ноябрь—декабрь 2021 — резкий рост до ~38 MAU.
* Январь—февраль 2022 — пик активности: 53–55 MAU.
* Март—апрель 2022 — падение до 30, затем повторный рост до 56 MAU.
* Май 2022 — резкое снижение до 20 MAU.

### **🔍 Промежуточные выводы:**

* **Сезонная динамика**: Активность падает весной, возможно, из-за окончания курсов или приближающихся каникул.
* **Сильные всплески** соответствуют началу семестров или образовательных программ.
* **Платформа достигала >50 активных пользователей ежемесячно**, что демонстрирует потенциал вовлечения, но требует работы по удержанию.

## **📊 Аналитика 3: DAU — Ежедневная активность пользователей**

### **🎯 Цель исследования:**

Понять, как пользователи взаимодействуют с платформой **ежедневно**, и выявить:

* Периоды пиков активности
* Устойчивость пользовательской вовлечённости
* Общий тренд использования платформы

Метрика DAU (Daily Active Users) помогает оценить **жизненность продукта** на ежедневной основе.

*SELECT*

*DATE(entry\_at) AS ymd,*

*COUNT(DISTINCT user\_id) AS DAU*

*FROM UserEntry*

*LEFT JOIN users ON userentry.user\_id = users.id*

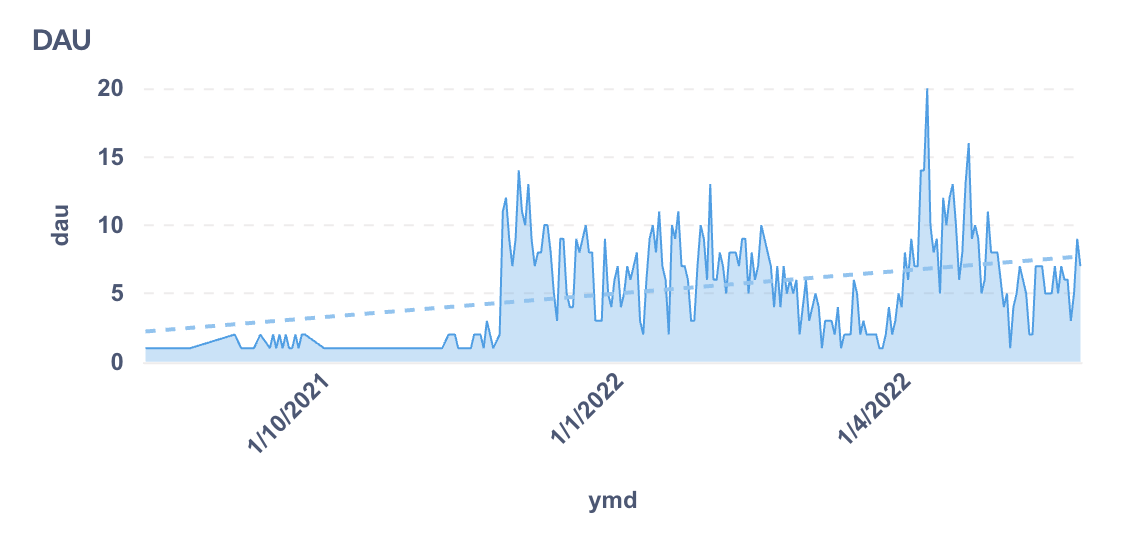
*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{created\_at}}]]*

*GROUP BY ymd*

*ORDER BY ymd*



### **📈 Полученные результаты:**

* График демонстрирует **плавный рост ежедневной активности** с осени 2021 года.
* Начиная с **января 2022**, наблюдается заметное увеличение количества активных пользователей в день — до **10–15 DAU**, с пиками до **20 DAU**.
* Весной 2022 есть **локальные всплески**, однако наблюдается также и **периодический спад**, что может быть связано с выходными, праздниками или завершением курсов.
* Трендовая линия (пунктирная) подтверждает **положительную динамику вовлечённости**.

### **🔍 Промежуточные выводы:**

* **Продукт набирал популярность** и вызывал интерес в течение всего анализируемого периода.
* Несмотря на колебания, **общий тренд — положительный**, что подтверждает потенциал платформы при масштабировании.
* Вариативность DAU требует дальнейшего анализа по дням недели и времени суток (например, для настройки уведомлений и запуска новых сессий).

## **📊 Аналитика 4: Новые пользователи по годам и месяцам**

### **🎯 Цель исследования:**

Оценить, **когда и с какой интенсивностью** пользователи регистрировались на платформе, чтобы:

* Определить ключевые периоды привлечения аудитории
* Связать прирост пользователей с внешними или внутренними инициативами (маркетинг, запуски)
* Сегментировать когорты по времени регистрации для дальнейшего анализа retention

*SELECT*

*EXTRACT(YEAR FROM date\_joined) AS year,*

*EXTRACT(MONTH FROM date\_joined) AS month,*

*COUNT(id)*

*FROM users*

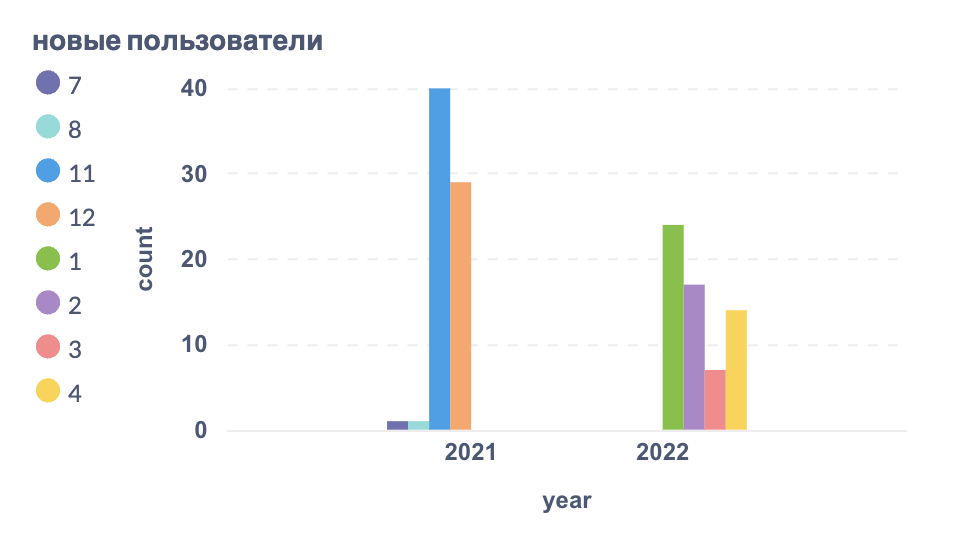
*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{date\_joined}}]]*

*GROUP BY year, month*

*ORDER BY year, month*



### **📈 Полученные результаты:**

* Основной рост пользовательской базы пришёлся на:  
  1. **Ноябрь–декабрь 2021** — пик: 40 регистраций в ноябре и 29 в декабре
  2. **Январь–апрель 2022** — более равномерный, но **уже менее интенсивный** рост:  
     + Январь: ~23 регистрации
     + Февраль: ~17
     + Март: ~8
     + Апрель: ~14
* Можно выделить два волны прироста:  
  1. **Поздняя осень 2021** — возможно, запуск платформы или рекламная кампания.
  2. **Начало 2022 года** — удержание интереса, возможно, за счёт учебных программ.

### **🔍 Промежуточные выводы:**

* Рост пользователей в конце 2021 года был **взрывным**, что требует анализа, что именно повлияло (например, пилотные запуски, реклама, конференции).
* В 2022 году наблюдается **стабилизация притока**, что даёт хорошую основу для когорного анализа и оценки удержания.

## **📊 Аналитика 5: Активность пользователей по дням недели**

### **💡 Цель:**

Понять, **в какие дни недели** пользователи наиболее активны, чтобы:

* Оптимизировать время публикации контента или запусков
* Определить периоды пикового и минимального трафика
* Планировать поддержку и ресурсы

*SELECT*

*TO\_CHAR(entry\_at, 'Day') AS week\_day,*

*COUNT(user\_id) AS count*

*FROM userentry*

*LEFT JOIN users ON userentry.user\_id = users.id*

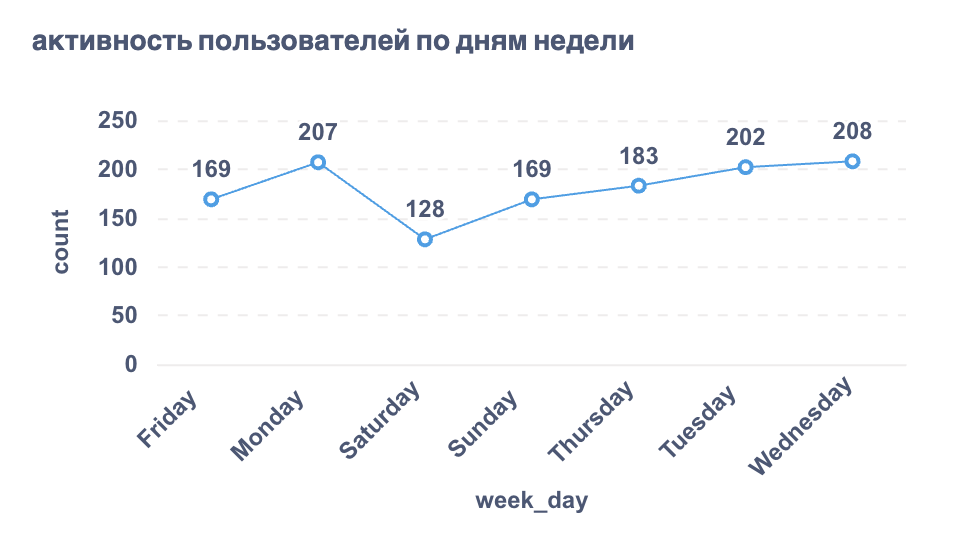
*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{entry\_at}}]]*

*GROUP BY week\_day*

*ORDER BY week\_day*



### **📈 Результаты:**

| **День недели** | **Активность** |
| --- | --- |
| Понедельник | 207 |
| Вторник | 202 |
| Среда | 208 |
| Четверг | 183 |
| Пятница | 169 |
| Суббота | 128 |
| Воскресенье | 169 |

### **📌 Выводы:**

* **Пик активности приходится на понедельник–среду** (макс: среда — 208, мин: суббота — 128).
* **Выходные (особенно суббота)** — наименее активные дни.
* **Плавное снижение** с середины недели к выходным и резкий спад в субботу может говорить о рабочем или учебном контексте использования платформы.

## **📊 Аналитика 6: Активность студентов по числам месяца**

### **💡 Цель:**

Определить, в какие **дни месяца** студенты наиболее и наименее активны, чтобы:

* Оптимально планировать контрольные точки: дедлайны, рассылки, активности
* Отследить закономерности (например, спад к концу месяца)

*SELECT*

*EXTRACT(DAY FROM entry\_at) AS day,*

*COUNT(DISTINCT user\_id)*

*FROM userentry*

*LEFT JOIN users ON userentry.user\_id = users.id*

*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{entry\_at}}]]*

*GROUP BY day*

*ORDER BY day*



### **📈 Основные наблюдения:**

| **Период месяца** | **Активность** |
| --- | --- |
| 📅 1–10 | Активность стабильно высокая, особенно 5-го (пик: ~41) |
| 📅 11–20 | Умеренная активность, с небольшим всплеском 14–15 числа |
| 📅 21–31 | Виден **устойчивый спад активности** к концу месяца (мин: 31-е — ~18) |

### **📌 Выводы:**

* **Пик активности — в начале месяца (1–10 числа)**: это удобное время для ввода новых заданий, модулей или активности.
* **Конец месяца — наименее активный период**, что может быть связано с усталостью, дедлайнами или административными задачами.
* Средняя активность сохраняется на уровне 30–40 пользователей в день до 25 числа.

## **🕒 Аналитика 7: Активность пользователей по часам дня**

### **💡 Цель:**

Определить **в какие часы суток** пользователи наиболее активно используют платформу для:

* Планирования публикаций и отправки уведомлений
* Оптимизации нагрузки на платформу
* Подбора времени проведения онлайн-событий

*SELECT*

*EXTRACT(HOUR FROM entry\_at) AS hour\_entry,*

*COUNT(\*) AS cnt\_users*

*FROM userentry*

*LEFT JOIN users ON userentry.user\_id = users.id*

*WHERE company\_id IS NOT NULL*

*[[AND {{company\_id}}]]*

*[[AND {{entry\_at}}]]*

*GROUP BY hour\_entry*

*ORDER BY hour\_entry*



### **📈 Основные наблюдения:**

| **Временной интервал** | **Активность (примерно)** |
| --- | --- |
| ⏰ 00:00–06:00 | Низкая (5–30 входов) — поздняя ночь |
| ⏰ 07:00–09:00 | Рост: до 50–60 пользователей |
| ⏰ 10:00–13:00 | Устойчиво высокая активность: 60–100 |
| ⏰ 14:00–18:00 | Пик в 13:00–14:00 (~100 пользователей) |
| ⏰ 19:00–21:00 | Второй пик (~100 пользователей) |
| ⏰ После 22:00 | Резкое снижение активности |

### **📌 Выводы:**

* **Два ключевых пика активности**: 13:00 и 19:00 — лучшее время для вебинаров, напоминаний или рассылок.
* Утро с 10:00 до 13:00 — стабильно активное, можно использовать для учебных блоков.
* После 22:00 — активность сильно падает, не рекомендуется планировать взаимодействие.

## **📊 Аналитика 8: Rolling Retention по заходам (когортный анализ)**

### **💡 Цель:**

Оценить **долгосрочную вовлечённость** пользователей в платформу по когортам (т.е. группам по дате первого входа).

*WITH reg AS (*

*SELECT DISTINCT user\_id, MIN(entry\_at::date) AS reg*

*FROM userentry*

*LEFT JOIN users ON userentry.user\_id = users.id*

*WHERE company\_id IS NOT NULL*

*[[AND {{date}}]]*

*[[AND {{company\_id}}]]*

*GROUP BY user\_id*

*),*

*dif AS (*

*SELECT DISTINCT userentry.user\_id, userentry.entry\_at::date - reg AS diff*

*FROM userentry*

*JOIN reg ON reg.user\_id = userentry.user\_id*

*ORDER BY user\_id, diff*

*)*

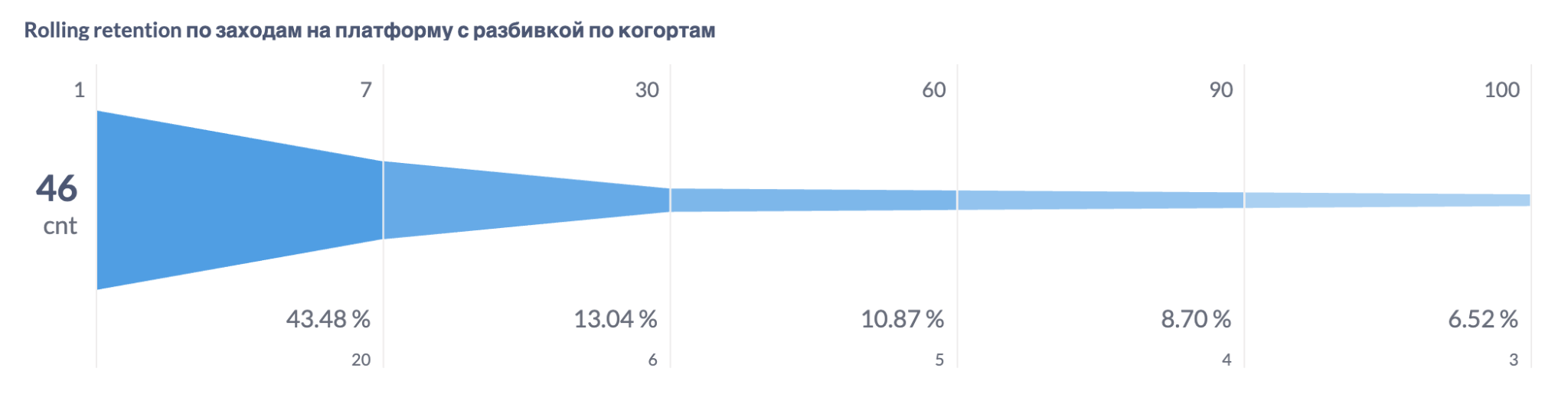
*SELECT diff,*

*COUNT(DISTINCT user\_id) AS cnt*

*FROM dif*

*WHERE diff IN (1, 7, 30, 60, 90, 100)*

*GROUP BY diff*



### **📈 Результаты:**

| **День с момента регистрации** | **Retention (примерно)** |
| --- | --- |
| 📅 День 1 | **100% (46 пользователей)** |
| 📅 День 7 | **43.48%** — около 20 пользователей вернулись |
| 📅 День 30 | **13.04%** — сохраняется базовая вовлечённость |
| 📅 День 60 | **10.87%** |
| 📅 День 90 | **8.70%** |
| 📅 День 100 | **6.52%** — долгосрочная лояльность |

### **📌 Выводы:**

* Удержание после первой недели — **падение более чем в два раза**, важен **активный онбординг в первые 7 дней**.
* Через 30 дней — остаётся только ~13% пользователей.
* Долгосрочное удержание (3+ месяца) — менее 10%, что типично, но оставляет простор для роста (например, через регулярные курсы, уведомления, челленджи).

## **📊 Аналитика 9: Процент завершённых тестов и их средний результат**

**💡 Цель:** определить долю пользователей, завершивших начатые тесты, и оценить средний уровень их успеваемости.

*WITH company\_users AS (*

*SELECT id AS user\_id*

*FROM users*

*WHERE true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*),*

*started\_tests AS (*

*SELECT company\_users.user\_id, test\_id*

*FROM teststart*

*JOIN company\_users ON teststart.user\_id = company\_users.user\_id*

*),*

*completed\_tests AS (*

*SELECT DISTINCT testresult.user\_id, testresult.test\_id*

*FROM testresult*

*JOIN company\_users ON testresult.user\_id = company\_users.user\_id*

*),*

*answers\_summary AS (*

*SELECT*

*testresult.user\_id,*

*testresult.test\_id,*

*COUNT(\*) FILTER (WHERE testanswer.is\_correct = true) AS correct\_answers,*

*COUNT(\*) AS total\_answers*

*FROM testresult*

*JOIN company\_users ON testresult.user\_id = company\_users.user\_id*

*LEFT JOIN testanswer ON testresult.answer\_id = testanswer.id*

*GROUP BY testresult.user\_id, testresult.test\_id*

*),*

*average\_score AS (*

*SELECT*

*ROUND(AVG(correct\_answers \* 100.0 / NULLIF(total\_answers, 0)), 2) AS avg\_score*

*FROM answers\_summary*

*)*

*SELECT*

*(SELECT COUNT(\*) FROM completed\_tests)::FLOAT / NULLIF((SELECT COUNT(\*) FROM started\_tests), 0) \* 100 AS percent\_completed,*

*avg\_score*

*FROM average\_score;*



### **📎 Результаты:**

| **Показатель** | **Значение** |
| --- | --- |
| ✅ Процент завершения | **32.26%** |
| 📈 Средний результат (%) | **34.98%** |

### **📌 Интерпретация:**

* **Только 1 из 3** пользователей доходит до конца теста.
* **Средний результат < 35%**, что говорит о низкой успеваемости.

### **🔍 Возможные причины:**

* Тесты слишком сложные или длинные.
* Недостаточная мотивация пройти до конца.
* Пользователи не понимают важность прохождения.
* Возможны UX-проблемы в интерфейсе.

### **✅ Рекомендации:**

* Сделать первые тесты короче и проще.
* Добавить геймификацию или бонус за завершение.
* Отслеживать точки оттока и адаптировать инструкции.
* Провести UX-тестирование для выявления барьеров.

## **📊 Аналитика 10: Среднее число попыток решения задач**

**💡 Цель:** понять, сколько попыток в среднем требуется пользователю для решения одной задачи (CodeSubmit).

*with avr as (*

*SELECT user\_id ,*

*problem\_id,*

*COUNT(\*) AS attempts*

*FROM CodeSubmit*

*left join users on codesubmit.user\_id = users.id*

*where true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*[[and {{date }}]]*

*GROUP BY user\_id , problem\_id*

*ORDER BY attempts DESC*

*)*

*select round(avg(attempts),2) as average\_attempts*

*from avr*

### **📎 Результат:**

| **Показатель** | **Значение** |
| --- | --- |
| 🔁 Среднее число попыток на задачу | **2.08** |

### **📌 Интерпретация:**

* В среднем пользователю требуется **2 попытки**, чтобы решить задачу.
* Это может указывать на:  
  + Задачи умеренной сложности.
  + Готовность пользователей доводить решение до конца.
  + Возможность улучшения обратной связи в интерфейсе.

### **✅ Рекомендации:**

* Проанализировать задачи с наибольшим числом попыток — возможно, они недостаточно понятны.
* Добавить подсказки или обучающие блоки для популярных задач.
* Ввести систему баллов, мотивирующую решать с первой попытки.

## **📊 Аналитика 11: Активность пользователей по языкам (в день)**

**💡 Цель:** выяснить, сколько уникальных задач в день пользователь:

* ✅ запускает (coderun)
* 📤 отправляет на проверку (codesubmit)

*with cnt\_run as (*

*select user\_id ,company\_id,created\_at::date as date , Language.name as name, count(distinct problem\_id) as cl*

*from coderun*

*left join users on coderun.user\_id = users.id*

*left join Language on language\_id = Language.id*

*where company\_id is not null*

*group by coderun.user\_id ,company\_id, created\_at::date , name*

*order by coderun.user\_id ),*

*cnt\_sabmit as (*

*select user\_id , company\_id ,created\_at::date as date ,Language.name as name , count(distinct problem\_id) as cl*

*from codesubmit*

*left join users on codesubmit.user\_id = users.id*

*left join Language on language\_id = Language.id*

*where company\_id is not null*

*group by codesubmit.user\_id ,company\_id,created\_at::date , name*

*order by codesubmit.user\_id*

*)*

*select distinct cnt\_run.name ,*

*avg(cnt\_run.cl) as cnt\_problem ,*

*avg(cnt\_sabmit.cl) as cnt\_problem\_submit*

*from cnt\_run*

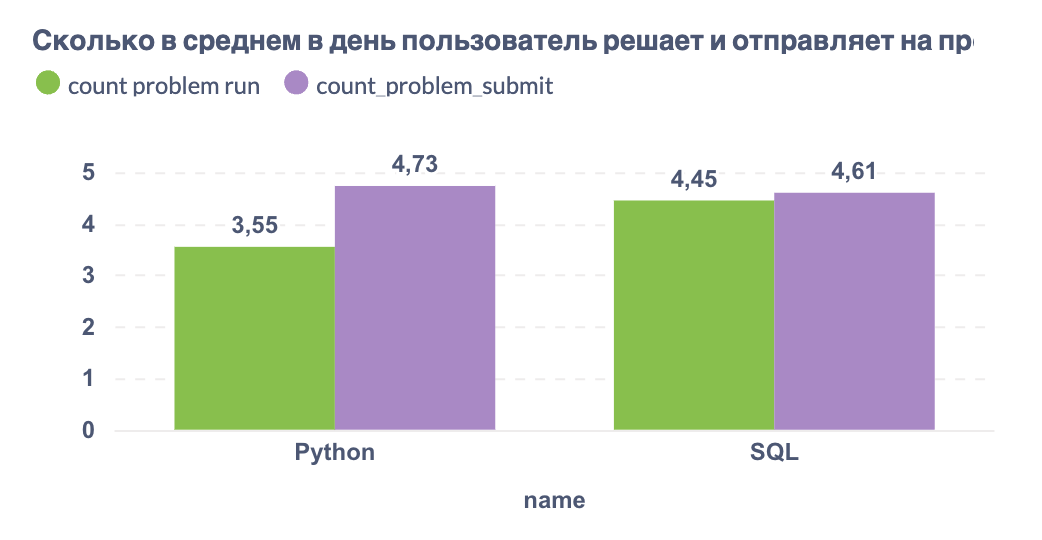
*left join cnt\_sabmit on cnt\_run.user\_id = cnt\_sabmit.user\_id*

*where true*

*[[ and {{date1}} and {{date2}}]]*

*[[and cnt\_run.company\_id={{company\_id}}]]*

*group by cnt\_run.name*



### **📉 Данные из графика:**

| **Язык** | **Среднее задач в день (Run)** | **Среднее задач в день (Submit)** |
| --- | --- | --- |
| Python | 3,55 | 4,73 |
| SQL | 4,45 | 4,61 |

### **📌 Интерпретация:**

* **Python:**
  + Пользователи чаще отправляют задачи на проверку, чем просто запускают их.
  + Возможна низкая потребность в предварительном тестировании.
* **SQL:**
  + Почти одинаковое число запусков и отправок — пользователи, вероятно, уверены в решениях или проверяют локально.
  + SQL может восприниматься как проще или более предсказуемый.

### **✅ Выводы и рекомендации:**

* Рассмотрите добавление промежуточной обратной связи для Python-задач (если пользователи не тестируют код).
* Отслеживайте, влияет ли число запусков на итоговое качество решений.
* Для SQL можно добавить более сложные кейсы, если большинство решений делается без проб и ошибок.

## **📊 Аналитика 12: % верно решённых домашних заданий**

**💡 Цель:** Показать, какой процент домашних заданий был решён корректно пользователями компании.

*with problem\_submit as (select created\_at::date as date ,*

*problem\_id ,*

*language.name ,*

*user\_id ,*

*company\_id ,*

*is\_false*

*from codesubmit*

*join users on codesubmit.user\_id = users.id*

*join language on codesubmit.language\_id = language.id*

*where true*

*and company\_id is not null)*

*select*

*count(case when is\_false = 0 then 1 end )::float / count(\*) \* 100 as percent*

*from Problem\_To\_Company*

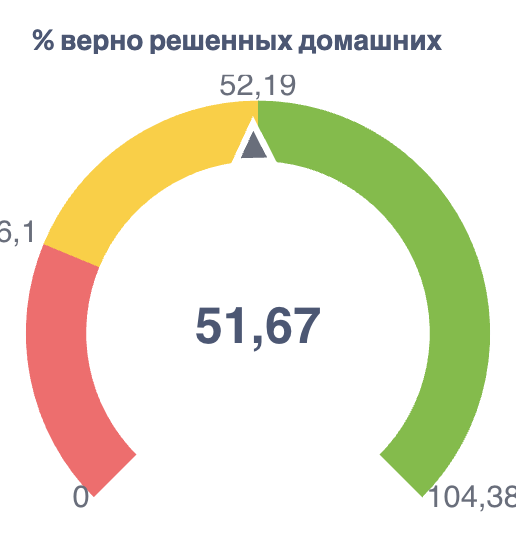
*join problem on Problem\_To\_Company.problem\_id = problem.id*

*join problem\_submit on problem\_submit.company\_id = Problem\_To\_Company.company\_id*

*Where true*

*[[and {{company\_id}}]]*

*[[and {{date}}]]*



### **📈 Показатели:**

* **Средний процент правильных решений:** **51,67%**
* **Жёлтая зона (умеренный результат):** от **36,1% до 52,19%**
* **Красная зона (низкий результат):** от **0% до 36,1%**
* **Зелёная зона (высокий результат):** от **52,19% до 104,38%**

### **📌 Интерпретация:**

* Текущий результат **чуть ниже зелёной зоны**, что указывает на **пограничную эффективность** — половина домашних заданий выполняется правильно.
* Может свидетельствовать о:  
  + недостаточном уровне подготовки;
  + сложных задачах;
  + нехватке обратной связи после ошибок.

### **✅ Рекомендации:**

1. **Анализ по пользователям** — выявить тех, кто часто допускает ошибки.
2. **Рассмотреть возможность добавления подсказок или разбора ошибок.**
3. **Проверить распределение сложности задач** — возможно, перекос в сторону сложных кейсов.
4. **Уточнить, по каким языкам/темам наблюдается наибольшая доля ошибок.**

## **📊 Аналитика 13: % правильно решённых задач**

### **🎯 Цель:**

Показать, какова доля корректных решений от общего числа попыток у пользователей компании.

*SELECT*

*ROUND(*

*SUM(CASE WHEN is\_false = 0 THEN 1 ELSE 0 END) \* 100.0 / COUNT(\*), 2*

*) AS success\_rate*

*FROM CodeSubmit*

*left join users on CodeSubmit.user\_id = users.id*

*WHERE true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*[[ and {{date}}]]*



### **📈 Основной показатель:**

* **Успешность решений:** **51,67%**

### **🎨 Интерпретация визуализации:**

* **Зелёная зона (хороший результат):** 50–100%
* **Жёлтая зона (средний результат):** 25–50%
* **Красная зона (низкий результат):** 0–25%
* Текущий результат **едва входит в зелёную зону**, что указывает на **граничный уровень успешности**.

### **📌 Вывод:**

* Половина отправленных решений оказываются **неверными**.
* Несмотря на формальный проход в "зелёную" зону, результат **ближе к жёлтому**, чем к уверенно высокому уровню.
* Это может указывать:  
  + на недостаточный уровень подготовки;
  + на необходимость усилить поддержку и обучение;
  + на сложности в заданиях или невнимательность при отправке.

### **✅ Рекомендации:**

1. **Провести анализ ошибок:** какие типы задач чаще всего решаются неправильно.
2. **Добавить тренажёры/разборы ошибок** по сложным темам.
3. **Отслеживать динамику по неделям:** улучшается ли точность решений со временем.
4. **Сравнить по языкам программирования** — Python, SQL и др.

## **📊 Аналитика 14: % правильно решённых задач по языкам**

### **🎯 Цель:**

Оценить успешность решений пользователей **по языкам программирования**.

*select language.name ,*

*count(case when is\_false = 0 then 1 end )::float / count(\*) \* 100 as Correctly*

*from codesubmit*

*join users on codesubmit.user\_id = users.id*

*join language on codesubmit.language\_id = language.id*

*where true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*[[and {{date}}]]*

*group by language.name*



### **📈 Результаты:**

| **Язык** | **% правильных решений** |
| --- | --- |
| Python | ~44% |
| SQL | ~57% |

(Значения приблизительны по графику.)

### **🔍 Интерпретация:**

* **SQL показывает более высокий уровень успешности**, что может говорить о:  
  + большей уверенности пользователей в этой области;
  + более понятных задачах или лучшей подготовке.
* **Python отстаёт почти на 13 п.п.**, что требует внимания:  
  + вероятно, задачи по Python сложнее;
  + возможны пробелы в базовых знаниях (условия, циклы, функции);
  + стоит проверить, есть ли высокая доля синтаксических ошибок.

### **✅ Рекомендации:**

1. **Разделить задачи по сложности** и посмотреть, где именно у пользователей возникают ошибки.
2. **Провести тематический разбор ошибок по Python** — какие темы "проседают".
3. **Запустить таргетированные практики/воркшопы по Python.**
4. **Использовать планку "Цель" (на графике — 80%)** как мотивирующий ориентир.

## **📊 Аналитика 15: % студентов, отправлявших ДЗ по сложности задач**

### **🧠 Суть:**

Показывает **долю студентов**, которые **отправляли домашние задания**, сгруппированные по **уровню сложности задач**.

*WITH user\_c1 AS (*

*SELECT COUNT(\*) AS cnt*

*FROM users*

*WHERE true*

*and company\_id is not null*

*[[and {{company}}]]*

*),*

*comp as(*

*SELECT*

*c.problem\_id,*

*ROUND(COUNT(DISTINCT c.user\_id) \* 100.0 / user\_c1.cnt, 2) AS percent\_users ,*

*p.complexity*

*FROM codesubmit c*

*JOIN user\_c1 ON true*

*join problem p on c.problem\_id = p.id*

*WHERE true*

*AND c.problem\_id BETWEEN 117 AND 248*

*[[and {{date}}]]*

*GROUP BY c.problem\_id, user\_c1.cnt , p.complexity*

*ORDER BY p.complexity desc )*

*select distinct complexity , AVG(percent\_users )*

*from comp*

*group by complexity*



### **📈 Результаты:**

| **Сложность задачи** | **% студентов, отправлявших** |
| --- | --- |
| 1 (простые) | **29.9%** |
| 2 (средние) | **20.61%** |
| 3 (сложные) | **19.1%** |

### **🔍 Интерпретация:**

* Самые простые задачи (1 уровень) — **наиболее популярны** среди студентов.
* С ростом сложности **снижается вовлечённость**:  
  + На задачи 2 уровня отправляли на 30% меньше пользователей;
  + Сложные задачи (уровень 3) — на **почти 11 п.п. ниже**, чем у простых.
* Это указывает на **барьер сложности**, мешающий широкому участию.

### **❗ Почему это важно:**

* **Мотивация падает с усложнением** — вероятны:  
  + нехватка поддержки (разборов, подсказок);
  + психологический барьер перед "трудными задачами";
  + возможный перекос в сложности по сравнению с уровнем подготовки.

### **✅ Рекомендации:**

1. **Добавить пошаговые подсказки или примеры** к задачам 2–3 уровня.
2. **Создать прогресс-линию** или челленджи: «Пройди все уровни сложности».
3. **Мониторить конверсии по уровням** — кто «застревает» на каком уровне.
4. **Сделать лёгкие задачи входной точкой**, а более сложные — частью поощряемого роста.

### **🧩 Связь с другими метриками:**

* Эта метрика объясняет **средний % успешных решений (~51,67%)** — ведь приоритетно решаются простые задачи, а более сложные — либо пропускаются, либо решаются хуже.
* Хорошо сочетается с анализом **по языкам**: например, сложные задачи могут быть особенно трудными по Python (что совпадает с низким % успешных решений).

## **📊 Аналитика 16: Ранжирование задач по сложности**

**(на основе времени решения и количества неудачных попыток)**

*WITH company\_users AS (*

*SELECT id AS user\_id*

*FROM users*

*WHERE true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*),*

*submits AS (*

*SELECT cs.user\_id, cs.problem\_id, cs.created\_at, cs.is\_false, cs.time\_spent*

*FROM codesubmit cs*

*JOIN company\_users cu ON cu.user\_id = cs.user\_id*

*),*

*success\_attempts AS (*

*SELECT user\_id, problem\_id, MIN(created\_at) AS success\_time*

*FROM submits*

*WHERE is\_false = 0*

*GROUP BY user\_id, problem\_id*

*),*

*attempts\_before\_success AS (*

*SELECT*

*s.user\_id,*

*s.problem\_id,*

*COUNT(\*) FILTER (WHERE s.created\_at < sa.success\_time AND s.is\_false = 1) AS failed\_attempts,*

*MAX(s.time\_spent) FILTER (WHERE s.created\_at = sa.success\_time) AS solving\_time*

*FROM submits s*

*JOIN success\_attempts sa*

*ON s.user\_id = sa.user\_id AND s.problem\_id = sa.problem\_id*

*GROUP BY s.user\_id, s.problem\_id, sa.success\_time*

*),*

*problem\_difficulty AS (*

*SELECT*

*p.id AS problem\_id,*

*p.name AS problem\_name,*

*p.complexity,*

*ROUND(AVG(a.solving\_time::numeric), 2) AS avg\_solving\_time,*

*ROUND(AVG(a.failed\_attempts), 2) AS avg\_failed\_attempts,*

*RANK() OVER (ORDER BY AVG(a.solving\_time::numeric) DESC) AS time\_rank,*

*RANK() OVER (ORDER BY AVG(a.failed\_attempts) DESC) AS fail\_rank*

*FROM attempts\_before\_success a*

*JOIN problem p ON p.id = a.problem\_id*

*GROUP BY p.id, p.name, p.complexity*

*)*

*SELECT \**

*FROM problem\_difficulty*

*ORDER BY time\_rank + fail\_rank*



### **📈 Описание графика:**

* **Ось X** — среднее время решения задачи (avg\_solving\_time).
* **Ось Y** — номинальная сложность задачи (complexity от 1 до 3).
* **Размер пузыря** — количество неудачных попыток (среднее avg\_failed\_attempts).
* **Каждый круг** — это одна задача.

### **🔍 Интерпретация:**

1. **Не все задачи уровня 1 — действительно простые**:  
   * Некоторые задачи с меткой "1" имеют **большее время решения** и **много неудачных попыток**.
   * Это сигнал: **номинальная сложность занижена**.
2. **У задач уровня 2 разброс большой**:  
   * Некоторые из них решаются быстро и без проблем, другие — ближе к уровню 3 по сложности.
   * Есть необходимость **пересмотра уровня сложности**.
3. **Уровень 3 — более устойчиво сложный**:  
   * Все задачи на верхнем уровне действительно имеют **высокую трудоёмкость и ошибочность**.

### **🧠 Выводы:**

* **Объективная сложность** ≠ **назначенная вручную сложность**.
* Для повышения точности аналитики и адаптации задач:  
  + Можно ввести **динамическую переоценку сложности**;
  + Или в аналитике использовать **эмпирические ранги** (как time\_rank + fail\_rank), а не только complexity.

### **✅ Рекомендации:**

1. **Обновить метки сложности задач** на основе поведения пользователей.
2. **Добавить в платформу индикаторы**: “Среднее время решения: X”, “Попыток до успеха: Y” — для прозрачности.
3. Использовать эти данные при **автоматической адаптации курсов**:  
   * Например, если студент легко справляется с "1" и "2", давать ему “настоящие задачи уровня 3”.

## **💰 Аналитика 17: Распределение CodeCoins по типам транзакций**

### **📌 Описание запроса:**

* Вы считаете **количество транзакций** по каждому типу (TransactionType.description).
* Отбор только по пользователям, **принадлежащим компаниям**.
* Группировка по описанию типа транзакции.
* Сортировка по убыванию количества транзакций.

*select TransactionType.description ,*

*count(TransactionType.value) as cnt*

*from transaction*

*left join TransactionType on transaction.type\_id = TransactionType.type*

*join users on transaction.user\_id = users.id*

*WHERE true*

*and company\_id is not null*

*[[and {{company\_id}}]]*

*[[and {{date}}]]*

*group by transactionType.description*

*order by cnt desc*

### **📊 Основные выводы:**

1. **🥇 Лидер активности — "Заход на платформу"**
   * Более **1000 транзакций** — это главный источник CodeCoins.
   * Это значит, что студенты **награду получают просто за визит**, а не за учебную активность.
2. **🏁 Регистрация и достижения (5, 15, 25 задач)**
   * Имеют **умеренное количество транзакций**.
   * Механика награждения за прогресс работает, но не является основной мотивацией.
3. **📉 Минимальная активность:**
   * Решить задачу, Купить задачу, Купить подсказку, Пригласить друга, Купить тест, Пополнить кошелек — почти не используются.
   * Это означает:  
     + Или **механики плохо объяснены** пользователям.
     + Или **монетарная система не вовлекает** — мотивации тратить или зарабатывать CodeCoins недостаточно.

### **💡 Рекомендации:**

| **Проблема** | **Рекомендация** |
| --- | --- |
| 🎯 Основная активность — за "вход" | Повысить ценность действий, связанных с обучением (решение задач, участие в марафонах). |
| 🤷‍♂️ Низкая вовлеченность в траты CodeCoins | Упростить и объяснить механику: зачем нужны подсказки, как купить задачу и зачем это. |
| 💬 Мало рефералов / баг-репортов | Добавить явный вызов к действию, бонусы за приглашение друзей, геймификацию баг-репортов. |
| 📊 Отсутствует баланс "заработал / потратил" | Добавить аналитику на студенческом уровне: "ты заработал 500 CodeCoins, потратил 30" — это вовлекает. |

### **📈 Возможные дополнения:**

* Построить **стек-график**: сколько **заработано vs потрачено** монет.
* Добавить **временную динамику** (например, по неделям или месяцам).
* Сделать **разбивку по компаниям** — как компании различаются по уровню вовлечённости.

## **📉💰 Аналитика 18 Топ задач с самым высоким процентом ошибок**

Все задачи в списке имеют **100% fail rate** — то есть **ни один студент из компании(-й)** не смог их успешно решить.

*SELECT language.name as name ,*

*problem.name,*

*ROUND(SUM(CASE WHEN is\_false = 1 THEN 1 ELSE 0 END) \* 100.0 / COUNT(\*), 2) AS fail\_rate*

*FROM CodeSubmit*

*left join language on CodeSubmit.language\_id = language.id*

*left join problem on CodeSubmit.problem\_id = problem.id*

*left join users on CodeSubmit.user\_id = users.id*

*where true*

*and users.company\_id is not null*

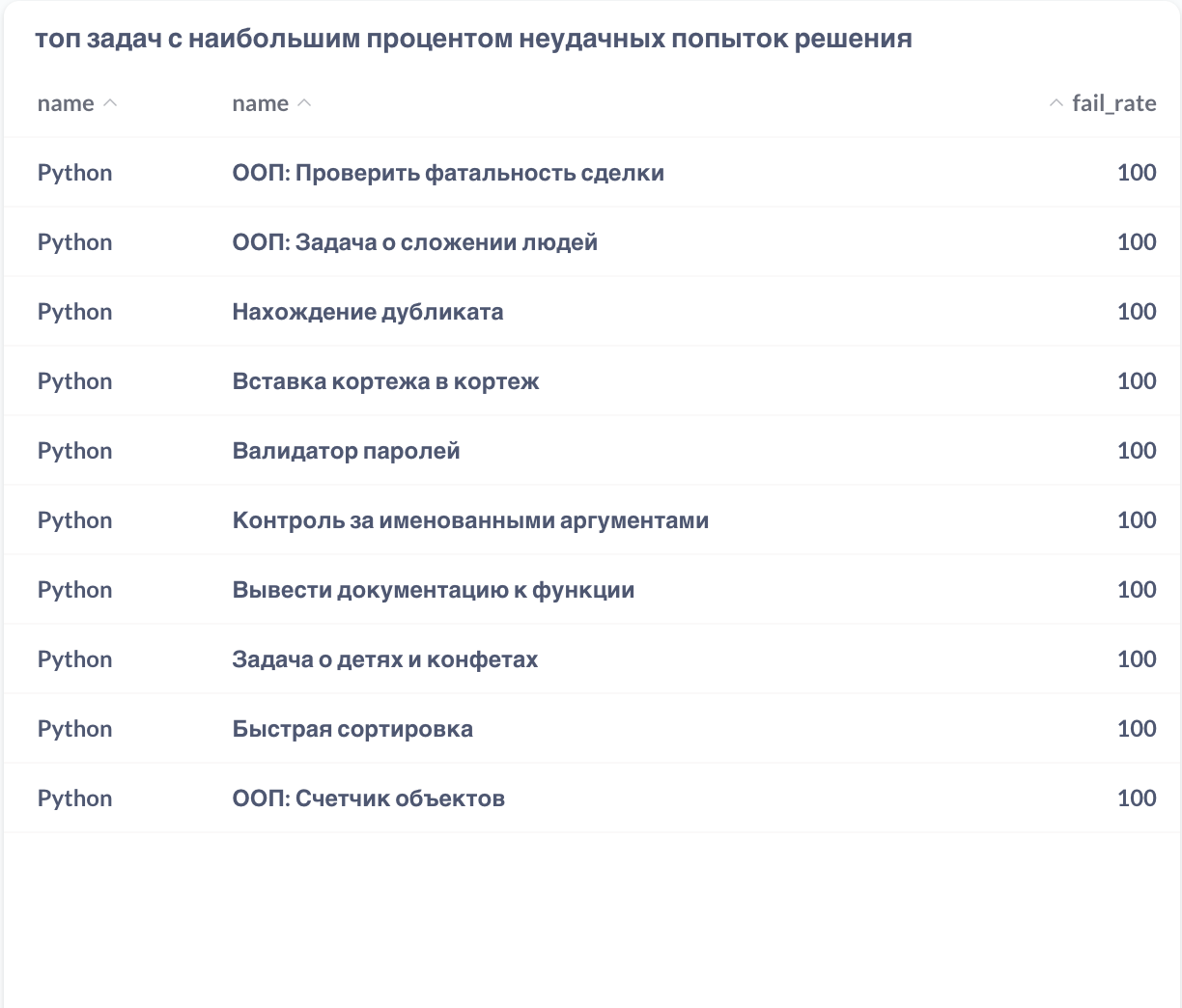
*[[and {{company\_id}}]]*

*[[and {{date}}]]*

*GROUP BY language.name , problem.name*

*ORDER BY fail\_rate DESC*

*LIMIT 10;*



* 📚 **Половина задач — по теме ООП (объектно-ориентированное программирование)**. Это может сигнализировать о:  
  + Недостаточной теоретической базе у студентов.
  + Слишком высоком уровне задач.
  + Неясных формулировках.
* 🧠 Также попадаются задачи на **алгоритмы и базовые структуры данных**:  
  + Быстрая сортировка, работа с кортежами, аргументы функций.

### **💡 Рекомендации:**

| **Проблема** | **Решение** |
| --- | --- |
| 📉 100% fail rate по ключевым задачам | Перепроверить формулировки, предусмотреть hint или разбор. |
| ❓ Недостаточная подготовка студентов по ООП | Добавить обучающие модули перед выдачей задач. |
| 📊 Нет различия между "почти решили" и "не поняли" | Ввести аналитику по средней длительности попыток и близости к решению. |
| ⚙ Задачи не адаптированы под текущий уровень группы | Перераспределить задачи по уровням сложности (по solving\_time + fail). |

### **📎 Можно добавить:**

* График: **распределение задач по fail\_rate**, не только топ-10.
* Таблица задач с fail\_rate > 70%, но не 100%, чтобы понять общую сложность пула.
* **Связь между временем решения, количеством попыток и процентом фейлов** (из предыдущих данных) — это даст комплексную картину сложности.

# II. Результаты и анализ

### **Активность и удержание:**

* Среднее значение 30-day retention: **6%**
* Активность сконцентрирована в будние дни, с 10:00 до 20:00

### **Образовательная активность:**

* Средняя успешность решений: **51%**, но есть задачи с **100% не успешным результатом**
* Тесты проходит лишь **1/3 активных** пользователей

### **Геймификация:**

* CodeCoins стимулируют первичную активность, но не влияют на удержание

# III. Выводы

### **Гипотезы:**

* Частично подтверждены:
  + Удержание низкое
  + Есть сложные задачи
  + Геймификация не даёт LTV

### **Рекомендации:**

1. **Адаптивная сложность**: автоматический подбор уровня задач по профилю и ошибкам.
2. **Контентные треки**: курсы и челленджи с целевой структурой.
3. **Система рангов** и наград: расширение CodeCoins для долгосрочной мотивации
4. **Оптимальное рассылочное время**: пуш-нотификации в 11:00—13:00

## **Заключение**

Проведённое исследование пользовательской активности и образовательной эффективности на платформе Simulative позволило получить целостную картину поведения пользователей, выявить узкие места и определить направления для развития. Анализ метрик вовлечённости, удержания и учебных результатов подтвердил, что несмотря на первоначальный интерес, значительная часть пользователей теряет мотивацию в течение первого месяца. Сложность отдельных заданий и неэффективность текущей системы геймификации оказывают влияние на retention и успешность обучения.

На основе выявленных закономерностей были выдвинуты и проверены гипотезы, часть из которых подтвердилась частично. Это позволило выработать конкретные бизнес-решения, направленные на повышение удержания, персонализацию обучения и увеличение ценности платформы для пользователей.

Реализация предложенных мер (адаптивная сложность, структурированные треки, улучшенная система мотивации и оптимизация коммуникаций) может не только улучшить пользовательский опыт, но и повысить ключевые бизнес-показатели: LTV, вовлечённость и конверсию в платящих клиентов.

Платформа Simulative обладает высоким потенциалом масштабирования при условии системной доработки на основе аналитических данных и гипотез, проверенных в этом исследовании.