

## **Практическая работа №5. Trade-off и метрики релиза**

Подготовила студентка группы М4100с Дремезова Анна.

**Цель работы:** научиться принимать решения о приоритетах (trade-off переговоры), исходя из метрик, ресурсов и неожиданных событий. Связать качество релиза с бизнес-эффективностью.

### **Что нужно сделать:**

1. Возьми релиз из предыдущей работы и определи, какие задачи можно выполнить в срок, а какие придётся отложить по тем или иным обстоятельствам.
2. Построй **матрицу приоритетов** (Value vs Effort, WSJF и т.п.). Обоснуй свои решения.
3. Рассчитай ключевые показатели:
  - **TCO (Total Cost of Ownership)** - совокупная стоимость релиза;
  - **CPU (Cost per Unit)** - стоимость одной функции или запроса;
  - **ROI** - ожидаемая отдача от релиза.
4. Добавь чёрного лебедя: например, падение метрик производительности, перерасход бюджета или неожиданный запрос на фичу от инвестора. Пересчитай приоритеты с учётом этого события.
5. Сделай вывод: что стоит изменить в релизе X+N, чтобы улучшить метрики и стабильность.

### **Что оформить в отчёте:**

- Матрицу приоритетов с учётом непредвиденных факторов.
- Таблицу метрик релиза (TCO, CPU, ROI, Run vs Grow и др.).
- Анализ влияния рисков и итоговый план корректировок.

### Матрица приоритетов с учетом непредвиденных факторов

Релиз 3: выпустить мобильную версию с полным CRUD для задач, офлайн-буфер. Обеспечить стабильную систему уведомлений с fallback-провайдером и ретраями. Запустить базовую геймификацию.

На этот релиз были следующие черные лебеди: через неделю после начала спринта уходит ключевой mobile-разработчик, отвечавший за архитектуру и сложные взаимодействия с бэкендом. Одновременно с этим основной провайдер пуш-уведомлений объявляет о массовом сбое, который длится несколько часов.

Высокая ценность	1. Базовый CRUD для мобильных приложений (Создание, просмотр, редактирование задач онлайн). 2. Устранение критических багов Релиза 2.	1. Доработка сервиса уведомлений: добавление fallback-провайдера, механизм повторных отправок (retries), мониторинг доставки 2. Механизм начисления очков (Базовая геймификация).
	1. Составляет фундамент мобильного приложения. 2. Повышает стабильность работы продукта.	1. Ценно за счет гарантии доставки уведомлений, влияет на SLA (аптайм и стабильность). 2. Повышение вовлеченности.
Низкая ценность	Drag&Drop в Kanban (мобильные)	1. Офлайн-буфер (синхронизация при появлении сети) 2. Расширенная таблица лидеров с историей, различные типы очков за разные действия.
	Улучшение UX, но не срочно.	1. Ценность может быть высокой в долгосрочной перспективе, но в кризисной ситуации не является первостепенной задачей. Затраты высокие и требуется экспертизы, которой нет. 2. Ценность низкая, а сил потратим много.
	Низкие затраты	Высокие затраты

### Таблица метрик релиза (TCO, CPU, ROI, Run vs Grow и др.)

ФОТ на месяц для команды = 7 982 000 руб.

Доля ФОТ за 10 дней:  $(7\,982\,000 / 21) \times 10 = 3\,800\,952$  руб.

Доп. Затраты составят приблизительно 25%, то есть  $3\,800\,952 * 0,25 = 950\,238$  руб.

Итого TCO:  $950\,238 + 3\,800\,952 = 4\,751\,190$  руб.

Метрика	Плановое значение	Фактическое значение
TCO (Total Cost of Ownership)	4 751 190,00 руб.	4 751 190,00 руб.
CPU (Cost per Unit)	1 187 797,50 руб.	1 583 730,00 руб.
	(за 4 фичи)	(за 3 фичи)
ROI (Return on Investment)	$=((200000*3)/4751190)*100\% = 12,6\%$	$=((120000*3)/4751190)*100\% = 7,6\%$
Run vs Grow Ratio	20%/80%	40%/60%
SLA: апптайм	99,7%	99,5%
SLA: критические баги	$\leq 2$	3
SLA: время реакции	$\leq 8$ ч	6 ч
SLA: перенос задач	$\leq 10\%$	25%
SLA: NPS	$\geq 85\%$	72%

### Анализ влияния рисков и итоговый план корректировок

Последствия от черных лебедей:

- Работа над самой сложной частью мобильного приложения останавливается. Оставшиеся мобильные разработчики не обладают такой же экспертизой.
- Падает доставляемость уведомлений, что прямо угрожает целевым SLA по апптайму и пользовательскому опыту.
- Разработчики тратят время на попытки «тушения пожара» с уведомлениями, параллельно пытаясь перераспределить задачи ушедшего коллеги.

Очевидно, что выполнить все запланированное в срок невозможно.

С точки зрения финансов была потеряна стоимость 1 187 797 руб., снижен ROI с 12,6% до 7,6%. увеличена доля Run с 20% до 40%.

Снижена производительность, нужно устранять инциденты, увеличено число критических багов и переноса задач.

Реакция команды и корректирующие действия

Техлид проводит совещание с РО, РМ и всеми лидами направлений. Project Manager показывает, какие задачи блокированы, и как сбой у провайдера влияет на текущие дедлайны. Product Owner принимает ключевое продуктивное решение: офлайн-режим для мобильных приложений - это «желательная» функция, а стабильность уведомлений и базовый CRUD «обязательные». Команда мобильной разработки временно усиливается за счет Frontend-разработчика, который помогает с UI-компонентами, освобождая мобильных разработчиков для работы над бизнес-логикой.

Команда принимает решение о том, что офлайн-режим переносится в другой релиз. Высвободившиеся ресурсы бросаются на стабилизацию уведомлений и завершение базового CRUD. Целевой показатель «Перенос

задач» пересматривается с  $\leq 10\%$  до  $\leq 25\%$ , что честно отражает масштаб инцидента. При этом показатели по багам и времени реакции остаются неизменными. Для реализации fallback-механизма используется заранее заложенный 20% буфер.

План улучшений для релиза X+N может быть следующим. В работу команды вводится короткое собрание для мониторинга состояния команды и выявления рисков перегрузки или ухода сотрудников. Определяется четкий процесс на случай ухода ключевого специалиста, кого сразу ставить в известность, как перераспределять задачи, кого привлекать для срочного найма. Техлид инициирует выделение времени в течении недели, когда каждый разработчик обязан задокументировать ключевые части своей работы, чтобы снизить риск единственной точки отказа. Буфер на непредвиденные работы должен быть увеличен с 20% до 30%. Архитектура системы требует повышения отказоустойчивости через автоматическое переключение на резервные провайдеры. Можно усилить мониторинг, чтобы узнавать о проблеме с уведомлениями, не когда пользователи начали жаловаться в поддержку. Необходимо увеличить покрытие автотестами.

В итоге, несмотря на «черных лебедей», команда смогла выпустить Релиз 3. Хотя офлайн-режим и был перенесен (фактический перенос 25%), основные цели по стабильности уведомлений и мобильному CRUD были достигнуты. Фактические значения SLA (аптайм 99,5%, крит. баги 3, время реакции 6ч) частично достигли целей.