

Анализ SLA веб и мобильного приложения

Михаил Гусир, М4200с

Описание исходного SLA

- Доступность (Uptime) $\geq 99.5\%$
- Время отклика $\leq 200\text{ms}$ для 95% запросов
- Время исправления багов: Critical $\leq 4\text{ч}$, High $\leq 24\text{ч}$, Medium $\leq 72\text{ч}$
- Стабильность релиза: ≤ 3 критических бага на релиз
- Rollback Rate: $\leq 5\%$ релизов требуют откат
- Удовлетворенность пользователей (CSAT) $\geq 4.2/5.0$
- Частота падений мобильного приложения $\leq 0.5\%$

Какие задачи решает

1. Определение обязательных метрик качества сервиса (7 ключевых показателей)
2. Фиксация нормативов времени реакции и восстановления
3. Установление целевых значений для пользовательского опыта
4. Формализация обязательств команды перед бизнесом

Перегибы, недочеты, риски

1. Трудновыполнимые обещания:

1. Uptime $\geq 99.5\%$ при сильной зависимости от внешних API (R6: факт 95.1%)
2. Payment Success Rate $\geq 98\%$ без contract-testing с банками (R3: упал до 63%)
3. Bug Fix Time ≤ 4 ч для кросс-платформенных критических багов

2. Перегибы в требованиях:

1. Response Time ≤ 150 ms для R2 (оптимизация) — требовало радикальных изменений
2. Release Stability ≤ 3 критбага для функционально сложных релизов (R3: 8 багов)

3. Размытые формулировки:

1. Нет SLA на время реакции на инциденты
2. Отсутствие процедуры отката релиза при Rollback Rate $\leq 5\%$
3. Не определены сроки получения PCI DSS сертификации

4. Риски

1. Юридический: отсутствие SLA на уведомление об изменениях внешних API
2. Операционный: нет требований к мониторингу времени обнаружения инцидентов

Оценка SLA в контексте чёрных лебедей

- Что работает
 - Чёткое время на исправление критических багов (R7: уязвимость исправлена за 2.5ч)
 - Наличие целевых метрик как триггеров действий (падение Payment Success Rate в R3)
 - Ответственность за метрики распределена (DevOps → Uptime, Security Engineer → PCI DSS)
- Что не работает
 - Жёсткие SLA на Response Time при росте нагрузки ×100 (R9: пришлось отключать функционал)
 - Release Stability при мажорных изменениях (R3, R6 — превышения в 2-3 раза)
 - Отсутствие SLA на восстановление данных (R10: потеря 2 часов данных)
- Проблемы
 - Не ясно, кто принимает решение об откате релиска
 - Нет приоритизации инцидентов (P1, P2, P3)
 - Размыта ответственность за коммуникацию с пользователями

Что можно было бы не делать

- Не стал бы фиксировать Payment Success Rate $\geq 98\%$ без fallback-механизмов и contract-testing с внешними провайдерами
- Не ввел бы штрафы за нарушение Bug Fix Time в первых релизах MVP, где команда сталкивается с непредвиденными сбоями инфраструктуры
- Не требовал бы Uptime $\geq 99.5\%$ для сервисов с непроверенной масштабируемостью и внешними зависимостями
- Не зафиксировал бы Response Time $\leq 150\text{ms}$ без проведения нагрузочного тестирования на пиковые сценарии
- Не прописывал бы SLA по безопасности (PCI DSS) без участия внешнего аудитора с самого начала проекта

Что можно доработать

- Матрица приоритетов инцидентов: P1 (критический): реакция ≤ 15 мин, восстановление ≤ 4 ч P2 (высокий): реакция ≤ 1 ч, восстановление ≤ 12 ч P3 (средний): реакция ≤ 4 ч, восстановление ≤ 3 рабочих дня
- Регламент коммуникации при авариях: Уведомление пользователей в течение 2 часов для P1-инцидентов. Ответственный: Product Owner + Support Lead
- Отдельный SLA для внешних API: Обязательное уведомление об изменениях за 7 дней. Автоматическое переключение на fallback при недоступности > 15 мин
- Процедура отката релиска: Rollback window: не более 2 часов с момента обнаружения проблемы. Решение принимает Tech Lead по согласованию с Product Owner
- Operational readiness checklist: Проверка мониторинга, fallback-механизмов, результатов load testing. Ответственные: DevOps + QA Lead

Вывод

- Оценка: 7/10

- ☒ Покрыты ключевые метрики качества сервиса
- ☒ Есть чёткие числовые значения (не "максимально быстро")
- ☒ Команда учится на ошибках и корректирует SLA после инцидентов
- ☒ Не хватает процедурных SLA (реакция, эскалация, откат)
- ☒ Не учтены риски внешних зависимостей в изначальных метриках

Рекомендация: Дополнить SLA операционными регламентами, ввести матрицу приоритетов инцидентов и формализовать процедуры взаимодействия при авариях.