

Слайд 1

В данной работе SLA рассматривается не как формальный документ для клиента, а как внутренний контракт команды, фиксирующий договорённости о качестве, сроках и ответственности. SLA используется как инструмент управления продуктом и командой, позволяющий оценивать качество релизов, управлять ожиданиями внутри команды, анализировать последствия «чёрных лебедей» и постепенно повышать зрелость процессов разработки.

Продукт представляет собой веб- и мобильное приложение «1С:Финансовый ассистент», ориентированное на бухгалтеров и предпринимателей. Это карманный помощник, который помогает пользователям не пропускать важные финансовые события и изменения.

Основные функции продукта:

- напоминания о плановых платежах (налоги, аренда, кредиты) и сроках сдачи отчётности;
- уведомления об изменениях в налоговом и бухгалтерском законодательстве, релевантных пользователю;
- интеграция с продуктами «1С:Предприятие» для автоматического получения данных о платежах и сроках.

Цитата из работы:

«карманного помощника для бухгалтеров и предпринимателей, который напоминает о платежах и сроках сдачи отчетности, а также уведомляет об изменениях в законодательстве.»

Слайд 2

Одним из ключевых перегибов данного SLA является включение метрик, на которые команда не имеет прямого операционного контроля. В документе зафиксированы требования к актуальности данных из 1С, доставке push-уведомлений и своевременности обновлений законодательства. Однако фактически выполнение этих показателей напрямую зависит от внешних систем: серверов 1С у клиентов, государственных порталов, а также платформ Apple и Google. Без отдельных контрактов и SLA с этими сторонами команда не может гарантировать заявленные значения, что создаёт риск систематического нарушения SLA даже при корректной работе команды.

Ярким примером такой «красной зоны» является релиз 1.2, где была зафиксирована метрика «Актуальность данных из 1С — не более одного часа». На практике этот показатель оказался невыполнимым, так как сервер 1С у клиента был отключён на техническое обслуживание. В данной ситуации команда формально нарушает SLA, хотя не имеет возможности повлиять на доступность внешней системы. Это нарушает баланс между качеством сервиса и операционным реализмом. Более устойчивым решением было бы формулировать SLA асимметрично — описывая именно зону ответственности команды. Например, фиксировать время обработки и отображения данных после их получения, а доступность внешнего источника выносить в отдельное соглашение по интеграции.

Похожий перегиб наблюдается и в релизе 1.3, связанном с доставкой push-уведомлений. SLA с требованием доставки push за пять минут не учитывает архитектурные ограничения мобильных операционных систем, особенно iOS, где фоновые уведомления могут задерживаться или агрегироваться для оптимизации энергопотребления. В таких условиях SLA становится заведомо невыполнимым и теряет свою управленческую ценность, превращаясь в источник демотивации команды.

Оптимальный баланс в данном случае мог бы быть достигнут за счёт дифференциации SLA по типам уведомлений. Для критичных транзакционных сообщений (например, уведомлений о просроченных платежах) допустимо сохранять строгие требования с оговоркой о работоспособности платформенных сервисов. Для менее срочных информационных уведомлений, таких как изменения в законодательстве, более реалистичным и управляемым является SLA с увеличенным временным окном доставки. Такой подход позволяет сохранить фокус на действительно важных пользовательских сценариях и повысить исполнимость SLA.

Слайд 3

Что в SLA сработало

SLA хорошо выявляет проблемные зоны.

Используемые метрики (доступность, стабильность, актуальность данных, CSAT) позволяют:

- быстро зафиксировать факт деградации сервиса;
- понять, в каком именно блоке произошёл сбой — инфраструктура, интеграции, пользовательский опыт или контент.

SLA запускает управленческие и процессные решения.

При возникновении «чёрных лебедей» команда не игнорирует нарушения, а:

- пересматривает объём релиза (scope);
- меняет приоритеты задач;
- корректирует процессы разработки и эксплуатации.

Таким образом, SLA используется как триггер для действий, а не как формальная отчётность.

Что в SLA не сработало

SLA слабо работает с внешними зависимостями.

Во всех работах критическое влияние на выполнение SLA оказывают:

- платёжные системы,
- интеграции с 1C,
- государственные порталы,
- инфраструктура облачного провайдера.

При этом эти зависимости формально остаются внутри общего SLA команды, хотя команда не контролирует их работу напрямую.

Почему это происходит

SLA изначально спроектирован как:

- технический,
- операционный,
- внутrikомандный инструмент.

Из-за этого он не покрывает в полной мере:

- коммуникационные риски,
- юридические сценарии,
- финансовые ошибки,
- кризисы пользовательского доверия.

Юридические и финансовые риски проявляются не как технические баги, а как штрафы, иски и репутационные потери. Такие инциденты не отражаются в технических метриках SLA и остаются вне его поля зрения.

Отдельным ограничением является то, что в процессах формально представлен только QA Lead, тогда как команда тестирования как полноценный участник delivery-цикла отсутствует. Это снижает вероятность раннего выявления критичных сценариев и усиливает эффект «чёрных лебедей».

Он не покрывает полностью:

- коммуникационные риски,
- юридические сценарии,
- финансовые ошибки,
- кризисы доверия.

Юридические риски проявляются не как баг, а как штрафы, иски и репутационные потери SLA их просто не видит.

Слайд 4

Мы бы не брали на себя SLA за то, что не находимся под нашим контролем

Мы бы не обещали сроки и качество для данных из 1С, государственных порталов, push-уведомлений, зависящих от платформ Apple и Google, а также для платёжных шлюзов без наличия отдельных соглашений и чётко прописанных деградационных сценариев. Взятие на себя подобных обязательств без реального контроля над источниками данных приводит к систематическим нарушениям SLA и искажает реальную оценку работы команды.

Мы бы не запускали монетизацию без отдельного финансового SLA

Мы бы не выкатывали платёжные функции в продакшен, если не проработаны сценарии возвратов средств, отсутствует staged rollout, не зафиксирован отдельный SLA на корректность платёжных операций и не подготовлен коммуникационный план для пользователей.

Ошибки, связанные с деньгами, являются самым дорогим типом «чёрных лебедей», так как напрямую затрагивают доверие пользователей, юридическую ответственность и финансовую устойчивость бизнеса.

Слайд 5

Разделили бы SLA по зонам ответственности

Что изменить:

Ввели отдельные SLA для внутренних сервисов, внешних API (1С, государственные порталы, платформы Apple/Google), а также для платёжных и финансовых операций.

Почему:

Такой подход позволил бы корректно оценивать зону ответственности команды, не «наказывать» SLA за сбои во внешних системах и быстрее локализовывать источник проблемы. SLA в этом случае отражал бы реальные обязательства команды, а не итоговый результат, зависящий от сторонних факторов.

Ввели бы отдельный финансово-бизнес SLA

Что изменить:

Для платёжных функций зафиксировали отдельные метрики:

- корректность платежей (0% ошибок как блокирующее требование);
- допустимое время возврата средств;
- обязательный staged rollout для финансовых фич;
- наличие коммуникационного плана при любых изменениях, связанных с деньгами.

Почему:

Ошибки, связанные с деньгами и доверием пользователей, обходятся бизнесу значительно дороже любых технических провалов. Такие риски требуют отдельного уровня контроля и не должны «растворяться» в общем техническом SLA.

Пересобрали бы взаимодействие команд и роль QA

В текущей схеме QA представлен только на уровне QA Lead и отделён от команды разработки. Отсутствует полноценная команда тестирования, а также непрерывный цикл «написали → проверили → уточнили».

Я бы встроил QA-команду в delivery-процесс на ранних этапах — при проработке требований и планировании релиза. Это позволило бы выявлять критичные сценарии (отмена, возврат, сбой) до начала разработки и существенно снизить вероятность появления «чёрных лебедей».

Слайд 6

Оценка 8 / 10

SLA хорошо развернут в плане управлеченческого инструмента, что и было целью всех работ.

Есть смысловая динамика по релизам.

