ЛАБОРАТОРНАЯ РАБОТА №6

Подготовила студентка группы М4100с

Дремезова Анна

КРАТКОЕ ОПИСАНИЕ ИСХОДНОГО SLA

Проверялось SLA Дарьи Демидовой

Тип продукта: Система электронного документооборота со встроенным BPM-движком, которая реализована как веб + мобильное приложение

Частота релизов: 1 релиз каждые 3 недели.

Что фиксирует SLA?

- Аптайм: 98,5% - 99,8%
- CSAT: 3,5 - 4.6
- Реакция на P:1 30 минут – 4 часа
- Стабильность (откаты): 0-1
- Целостность данных: 100%
- Время развертывания: 2 - 4 часа
- Производительность(моб/веб): 1,5 - 3,5 сек
- Время BPM: ≤ 10 сек
- Кол-во P1-багов после релиза: ≤ 2
- Прирост обращений в поддержку: ≤ 20%
- Время формирования отчета аудита: ≤ 30 сек
- Индекс удобства редактора: ≥ 4.0
- Точность распознавания голоса: ≥85%
- CPU AI-запроса: $0.5
- Уязвимости Critical/High: 0

Должность	Роль	Количество человек
Product Owner	Формирование видения продукта, roadmap, приоритеты	1
Project Manager/Scrum Master	Организация процессов, отслеживание сроков, устранение блокеров	1
Business/System Analyst	Детализация требований, подготовка user stories и use cases	1
Tech Lead	Техническая архитектура, выбор стеков, code review	1
Backend-разработчики	Разработка ядра СЭД, BPM-движка, API, безопасность	2
Frontend-разработчик (веб)	Разработка интерфейса веб-приложения	2
Mobile-разработчики	Кроссплатформенная разработка (Flutter/React Native) или нативные (iOS, Android)	2
QA-инженеры	Ручное и автоматизированное тестирование, тест-планы	2
DevOps/SRE-инженер	Инфраструктура, CI/CD, мониторинг, deployment	1
UX/UI-дизайнер	Дизайн интерфейсов, прототипы, дизайн-система	1
Технический писатель	Написание пользовательской и внутренней документации	1
Support-инженер	Первая линия поддержки, работа с пользователями, инцидентами	2

ПЕРЕГИБЫ, НЕДОЧЁТЫ, РИСКИ

Где SLA обещает невозможное или жестко поставлено:

- Аптайм в некоторых релизах завышен (например, на первых релизах)
- Целостность данных = 100%
- 0 уязвимостей Critical/High (8 релиз)
- Стабильность = 0 откатов
- Время реакции на P1 ≤ 30 минут
- Обязательное еженедельное сканирование зависимостей избыточно

Где наоборот - недописано, слишком мягко, слишком неопределённо?

- Целостность данных: 100%, нет четкого определения понятия, критериев целостности данных
- Не указано время восстановления
- Время развертывания ≤ 2–4 ч при каких условиях такое?
- Для Аптайм можно уточнить исключения
- «CSAT: 4.0 (опрос первых 50 пользователей)», есть число пользователей, но нет периода опроса, метода расчета, исключений
- Время реакции на инцидент указано не во всех релизах, не понятно, что означает «реакция»? На что реагируют? Какое время отводится на нее? Нет разбивки на типы инцидентов
- «Время BPM ≤ 15 сек» (Релиз 3) что именно измеряется?
- «Кол-во P1-багов после релиза: ≤ 2» (Релиз 4) не определено, что считается P1-багом
- Прирост обращений в поддержку: ≤ X%. От какого числа? От среднего числа за предыдущий месяц? За аналогичный период после прошлого релиза?
- Спорная измеримость UX-показателей

Какие пункты создают риск юридического или операционного «техдолга»?

- Логирование AI-решений, без привязки к классификации данных, доступа и шифрования
- 100% целостность, 0 откатов, 0 уязвимостей Critical/High
- Интеграция с 1С, Google создает зависимость от сторонней системы и вендоров, изменения в которой могут требовать доработок, плана реакции на инциденты, связанные с ними, нет
- Фиксация цен на внешние сервисы несет финансовые затраты и юридические обязательства

ОЦЕНКА SLA В КОНТЕКСТЕ ЧЁРНЫХ ЛЕБЕДЕЙ

Какие пункты SLA помогут в кризисе?

- Правило «двух ответственных»
- Указанные время реакции на P1-инциденты, время развертывания
- Ежедневные 15-минутные синки
- Фиксация решений в общем канале в течение 1 часа
- Требование к гео-избыточности
- Пилотное тестирование рискованных функций на фокус-группе
- Обязательное документирование архитектурных решений в Confluence

Какие - наоборот усугубят ситуацию?

- «Целостность данных: 100%»
- «0 откатов» в нескольких релизах
- SLA по производительности без разделения
- Фиксированные cost-targets для внешних сервисов без правил пересмотра
- Временные промежутки для пост-релизного наблюдения (24 ч), при недостаточном кол-ве ресурсов недостаточно
- Обязательное еженедельное сканирование уязвимостей

Какие зоны ответственности определены хорошо, а какие - размыты?

Есть прописанные зоны ответственности ролей (ЛР 3, таблица в начале), но не до конца понятно кто отвечает за внешнюю коммуникацию и в какие сроки, кто принимает ключевые решения, решения об остановке сервиса или изменениях в архитектуре. Хорошо описаны Tech Lead, DevOps, Support.

Хватает ли приоритезации инцидентов?

Нет. В работе упоминается только P1, т.е. нет других инцидентов и не понятно, что к ним относится.

Ясно ли определено, кто принимает решения и в какие сроки?

Нет, можно было бы добавить матрицу распределения ответственности, дополнительно прописать сроки принятия решений

ЧТО БЫ ВЫ НЕ ДЕЛАЛИ

- Не вводили абсолютные цели вроде 0 откатов или 100% целостности данных без уточнения исключений, 0 откатов встречается часто и гарантировать их отсутствие даже при условии, что это не интеграционный релиз, сложно.
- P1 не фиксировала бы 30 минут, если нет поддержки 24/7 и достаточного количества людей на линиях поддержки, бюджета. Это касается и пост-релизного наблюдения (24 ч), нужны дежурные
- Не обещали бы «0 уязвимостей Critical/High», без бюджета на безопасность или проведения пенттестов
- Не вводили бы финансовые метрики (CPU AI-запроса ≤$0.5) без долгосрочных контрактов с вендорами, включенных заранее в SLA ограничений ответственности в случае изменений со стороны провайдера
- Не обещали обязательное еженедельное сканирование зависимостей безопасности, если не подготовлена инфраструктура, люди, планы реакций для этого
- Мы бы не хранили полные AI-логи 90 дней без шифрования, ролевого доступа, политики по безопасности
- Не указывали во всех релизах CSAT как метрику SLA, а вынесли в отдельный раздел бизнес-метрик
- Не указывали бы метрики SLA без формул, периода измерения, исключений, откуда получаем метрики (например, Grafana, Zabbix)
- Не подгоняли каждый раз SLA, а ввели правила его пересмотра
- Не использовали бы экономические метрики (TCO, ROI) без горизонта расчета, методики расчета
- При росте цен AI-сервиса (релиз 7) внедрили кэширование, fallback на правила, начали разработку собственной упрощенной модели, но не стали пересматривать контракт или искать альтернативы, возможно стоило это сделать

ЧТО БЫ ВЫ СДЕЛАЛИ ПО-ДРУГОМУ

- Ввели отдельный SLA для работы с внешними API
  - Прописали бы допустимую деградацию, порядок отключения функционала
- Добавили регламент коммуникации в момент аварии
  - Так как сейчас не прописано время первого уведомления/реакции на тикет, не определено кто связывается с клиентом/бизнесом, когда вернемся с ответом или новой информацией по проблеме, нет деления на внутреннюю и внешнюю коммуникацию (через что взаимодействуем: почта и т.п.)
- Матрицу инцидентов с их приоритетами
  - Уточнили бы что считается P0, P1 и т.д. инцидентом
- Время реакции на инциденты для всех релизов указывали, а не выборочно. Также подробно расписывала тип инцидента + время реакции на него
- Прописали бы что уязвимостей Critical/High не 0, а допускала бы их наличие и прописывала условия, время устранения
- Стабильность = 0 откатов не обещали бы в большинстве релизов, странно увеличивать их кол-во до 1 только там где точно знаем, что они понадобятся (релизы с интеграциями: 5, 7)
- Описали бы порядок отмены релиза, так как не всегда получится выпустить его вовремя
- Добавили бы проведение ревью другими членами команды, чтобы сотрудники были в контексте на случай ухода члена команды
- Время развертывания указывали бы с условиями
  - время развертывания ≤4ч составит при отсутствии сбоев инфраструктуры провайдера, в противном случае *фиксируются причины по которым можно потребовать возмещение или нет*
- Задачи передавали бы с четким указанием их приоритетов (относится к ЛР4)
- Дизайн-ревью ввели бы раньше (ЛР4) 6 или 7 релиза, хотя мобильную версию запустили во 2, получили конфликт
- Прописали бы границы веб и мобильной версии
- Время развертывания разделили
  - например так: простые патчи - ≤ 2 ч, миграционные релизы - ≤ 24 ч с планом отката
- Производительность указывали бы с разделением на мобильную и веб версию (сейчас не везде указано, не очевидно)

ВЫВОД

Оценка: 4/10

Обоснование:

- Требования к отсутствию уязвимостей и откатов неконструктивны и юридически опасны, так как делают SLA источником потенциальных судебных претензий. Критериев по тому когда делать откат тоже нет
- Хранение AI-логов без четкой политики безопасности и доступа к ним несут юридические и регуляторные риски
- Финансовые обещания по тарифам внешних поставщиков несут экономические уязвимости, в целом нет мер защиты от проблем с работой с внешними поставщиками (их в работе несколько)
- Недостаточно сотрудников для некоторых реакций на черных лебедей (например, релиз 8: пострелизное наблюдение 24 часа или релиз 10: время реакции на P1: ≤ 30 мин, при 2-х Support-инженерах) и в целом нет достаточного числа сотрудников для предотвращения черных лебеде, связанных с болезнями, увольнениями, отпусками. Явно не указано изменение их числа.
- В таблицу релизов в ЛР3 не всегда вносились новые метрики из релизов (например, на релизе 7 ввели «Стоимость владения фичей (TCO)», на релизе 4 «Время самостоятельного освоения новой фичи» но далее нигде не указывали)
- Нет достаточной приоритезации инцидентов (только P1). Не ясно, что считать P1, а что P0, аналогично и с багами, нет времени реакции на них, механизмов коммуникации
- Многие меры внедрялись уже после инцидента, нет никаких мер по их предупреждению
- Нет границы между веб и мобильной версией и полного контекста продукта (ЛР4, проблемы на релизах 6,7)
- В таблицах релизов смешаны метрики разного уровня: аптайм (SLA/SLI), CSAT (KPI), время реакции (SLO), количество обращений (KPI)
- Измеримость метрик не ясна, пример: «CSAT: 4.0 (опрос первых 50 пользователей)», есть число пользователей, но нет периода опроса, метода расчета, исключений
- Некоторые пункты (например, инциденты) имеют недостаточную детализацию, сотрудники могут начать подгонять показатели, чтобы в отчетах выглядело так, что нет проблем
- Метрики некоторые поставлены жестко, сотрудники начнут подгонять результаты под них, скрывая проблемы
- Не всегда ясно, кто принимает решение, все обобщенно написано, есть роли, но кто именно сказал принял такое решение не указано
- Не указаны какие-то конкретные артефакты, внедрялось документирование архитектурных решений в Confluence, но что вносится туда (артефакты: архитектурные схемы в C4, диаграммы развертывания и т.п.), по каким шаблонам

СПАСИБО ЗА ВНИМАНИЕ