CUSTIS

Обеспечиваем устойчивость интеграции

Максим Цепков
Главный архитектор решений CUSTIS
Навигатор по миру Agile, бирюзовых организаций и спиральной динамики

http://mtsepkov.org

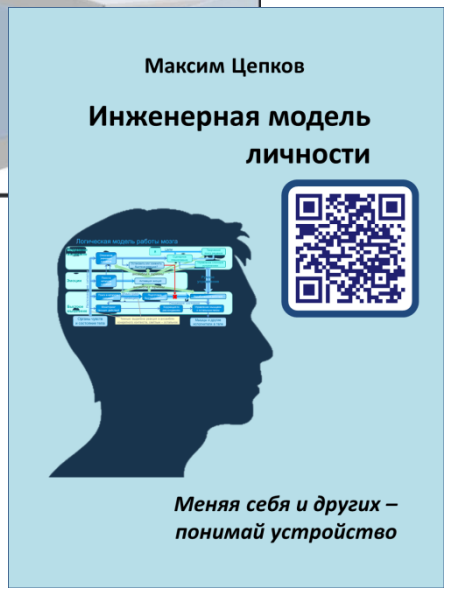
SQA Days #36

25–26 апреля 2025, Петербург

36 sqa days

Немного обо мне

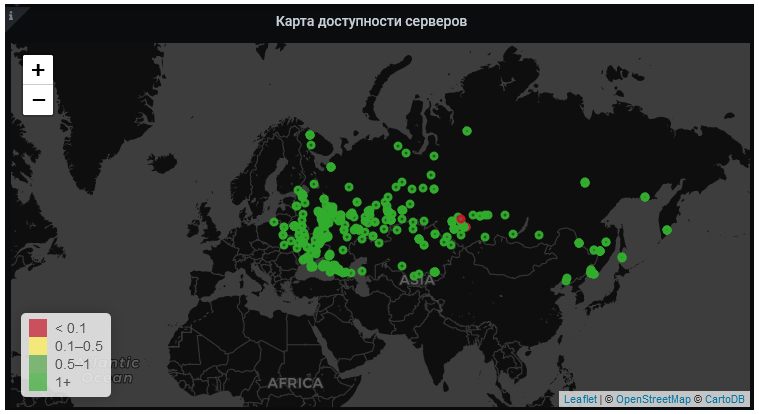
- Создание и внедрение больших корпоративных систем (более 25 лет)
  - Знание практик операционного управления и ведения проектов в крупных коммерческих и государственных организациях и банках
  - Опыт управления проектами в IT: от инженерного подхода и PMBOK – к современным Agile-методам (с 2007 года)
  - Опыт перестройки процессов организаций при внедрении систем
- Навигация в менеджменте цифрового мира
  - Agile и менеджмент самоуправления: бирюзовые организации, холакратия и социократия (**книга**, статьи и выступления)
  - Модель спиральной динамики (с 2013 года) и другие модели soft skills, **модели личности** и самоопределения (**книга**, статьи и выступления)
  - СМД-методология и развитие СРТ при переходе в цифровой мир



На моем сайте mtsepkov.org – мои выступления и много других материалов

Мой опыт большой интеграции

1997 Банковская система, интеграция по модемным каналам: документы, справочники, метаданные



2003 Мастер-данные: каталог 1M+ товаров с ценами, 1000+ точек установки, развивается до сих пор

Раз в 5–7 лет аудит технологий — нет ли чего готового лучше? Нет!

2008 Подключение к интеграционной шине банка: НСИ и документы

Интеграцию делали несколько подрядчиков для своих проектов, можно сравнить

2020 Централизованная ферма кассовых аппаратов (ККМ) для печати чеков торговой сети, повышенные требования надежности

И это не считая множества других проектов

Что полезного в этом докладе?

- Хорошие тестировщики **отвечают за качество продукта,**
  а не просто проверяют функционал тикета
- **Качество интеграции закладывается на этапе проектирования**,
  и его тоже надо проверять
- Я буду говорить не про тестирование того, что сделано, а про конструкции хорошей интеграции, и вопросы, которые стоит задать при проектировании
- А тест-кейсы должны проверять, что конструктивные решения работают как ожидается, в том числе в ситуациях инцидентов

Что такое хорошая интеграция?

Вариант 1

Делаем быстро и дешево,
часто на коленке

Вариант 2

Делаем на выбранных
«правильных» технологиях

Правильный вариант: **оптимизируем стоимость эксплуатации**

- Хорошая админка, позволяющая решать инциденты
- Шаблоны программирования, устойчивые к ошибкам и потерям при передаче
- Хорошая расширяемость

И важна архитектура интеграции, которая позволит сделать все это!

Хорошая админка

Как сделать хорошую админку

Разработчики и тестировщики должны понимать **реалистичный** сценарий:

- Проблемы будут: сообщения будут теряться, путаться, неверно обрабатываться
- Сначала проблемы **они будут решать сами** — надо предусмотреть средства
- Когда средства будут хороши — они смогут передать проблемы своим инженерам
- И только потом, в кооперации с инженерами — отдать сотрудникам заказчика
- И так повторяется для каждой новой внедряемой фичи

Этот же сценарий знают руководители проектов — им надо заложить ресурсы

Этот сценарий надо поддерживать практически — кто-то из опытных членов команды должен своевременно задавать вопросы о разборе будущих инцидентов при обсуждении технических решений интеграции и админки в целом

Вопросы от тестировщика и инженера

- Если произошел инцидент, требующий ручного разбора
  - Увижу ли я содержание сообщения, с раскрытием всех конвертов, и структурированно, а не потоком xml или json?
  - Если сообщение касается существующих объектов – какие средства поиска этих объектов через интерфейс системы, а не в базе данных?
  - Если причина инцидента – ошибка в коде и пришел хотфикс – можно ли запустить обработку дальше, в том числе – отобрав все подобные инциденты, а не по одному?
  - Если причина на стороне отправителя – то каков сценарий устранения проблемы
- Про логическую связь сообщений
  - Если сообщения имеют логическую последовательность обработки, то какие средства ее обеспечивают, особенно при многопоточном чтении сообщений из очереди?
  - Как избегают кумулятивного накопления ошибок?
  - Если передача на много инстансов – как избежать исправления ошибок на каждом?

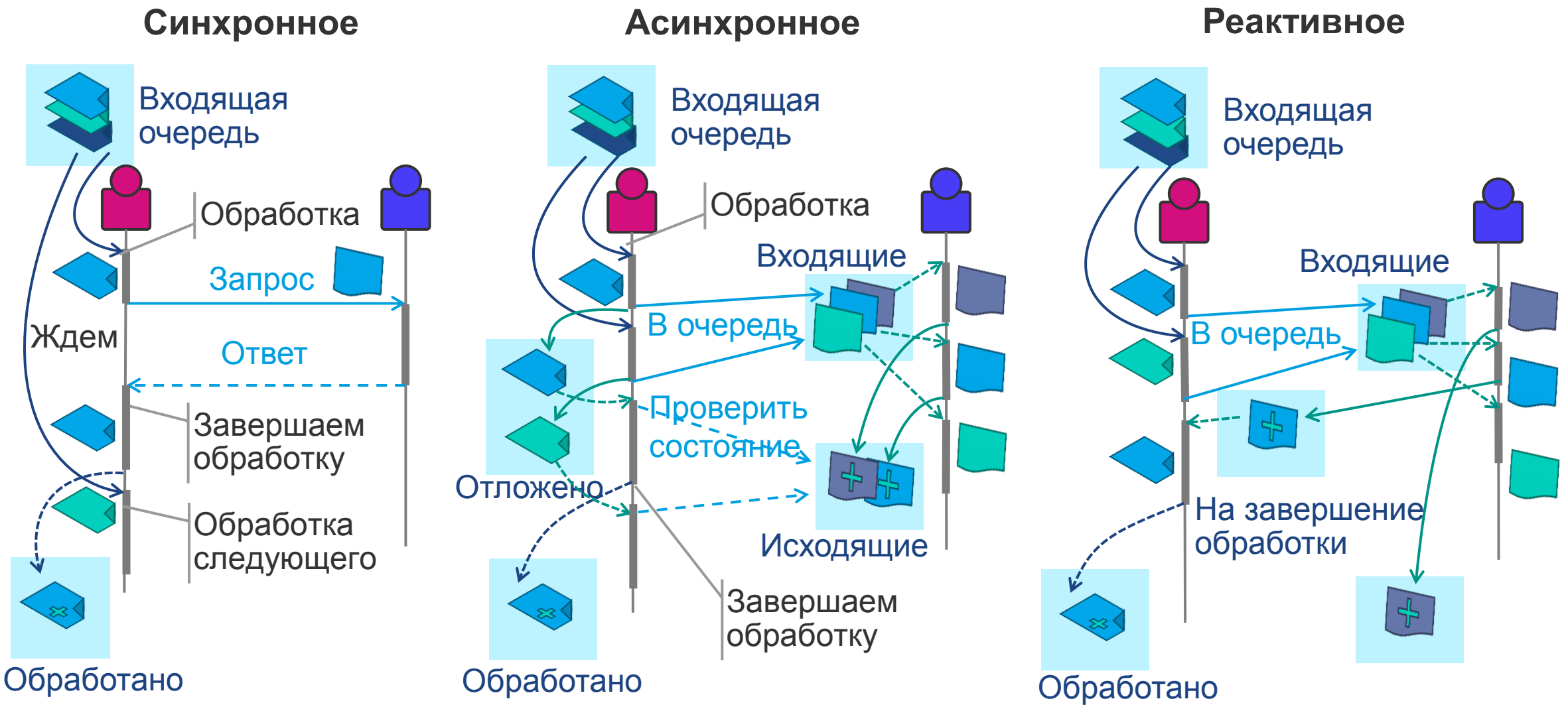
Хорошая админка

- Не только показывает ошибку, но и позволяет с ней разобраться
  - Показывает содержание сообщения
  - Позволяет заново обработать сообщение или заново отправить
  - Позволяет при необходимости переслать целый объект или набор объектов
  - Сравнивает объект в системе с передаваемым, показывая различия
- Позволяет массовую сверку или передачу данных между системами
- Обеспечивает решения при различии ограничений бизнес-логики в системах
- Имеет свои или стандартные средства мониторинга и настройки уведомлений

Хорошая админка предотвращает ночные авралы с разбором инцидентов

Устойчивые шаблоны

Внутри интеграции – передача сообщений



Как действует ваш транспорт?

- Протоколов передачи, обеспечивающих exactly once – не существует!
  - Если протокол обеспечивает гарантированную доставку – как игнорируем дубли?
  - Если протокол теряет сообщения – как работаем с пропусками?
  - Квитанция о доставке – это постановка в очередь внутри приемника или обработка им?
  - А если сообщение доставлено, а при обработке произошла ошибка?
  - А если сообщение обработано, а ошибка – на доставке и обработке уведомления?
- Устойчивость на большом потоке сообщений
  - Не останавливаем прием и обработку на первой же ошибке
  - Как отслеживаем цепочки сообщений, для которых важен порядок обработки?
  - Как обеспечена нумерация в цепочке, всегда ли ей соответствует фактический порядок?
- Выясняем проблемные месте и думаем про их покрытие тест-кейсами!

Передаем объекты или вызовы?

- Передача объектов — транспортный объект и его образы в системах
  - Возможен контроль соответствия данных в разных системах, устранение расхождений
  - Возможна загрузка и передача объектов в полуручном режиме
  - Но! Объекты большие, зачем передавать целиком, когда менялась только часть?
- Чистая передача вызовов — вызов API
  - Сообщение приводит к удаленному вызову процедуры, содержание не фиксировано
  - Дает гибкость в проектировании и независимость систем между собой
  - Но! Нет соответствия между данными в различных системах, невозможна сверка
- Композитные варианты
  - Передача изменений объектов (CRUD, REST API)
  - Вызовы для массовых операций над объектами
- Event Sourcing

Критерий выбора — не простота реализации,
а **устойчивость и разбор инцидентов**

Устойчивые шаблоны

- Идемпотентные операции — аналог UPSERT
  - CRUD и REST — не идемпотентны!
- Идемпотентные операции при передаче документов и сложных объектов
- Создание уникальных идентификаторов
- Потери и дубли сообщений
- Идемпотентные операции в публичных протоколах

Транзакции — нужны ли они? Консистентность без транзакций

О транспорте данных

- Файловый транспорт — хорошая штука
- Http и его аналоги
- MQ и другие сообщения
- Шины данных
- Удаленный вызов
- Транспорт поверх БД

Любой из транспортов можно использовать для любых целей

✓ Передача объектов, удаленные вызовы или event sourcing

✓ Синхронное, асинхронное, реактивное взаимодействие

Интеграция под нагрузкой – как проверять?

- Необходимо решить несколько проблем
  - Тестовый стенд, нормированный по производительности относительно промышленного
  - Репрезентативный поток тестовых данных, учитывающий влияние сообщений
- Можно тестировать на проде, урезая ресурсы в период малой нагрузки
  - Такое тестирование заодно может проверять отказы части оборудования
- В потоке данных могут быть особые случаи, ломающие масштабирование: большие документы, пачки документов с взаимной блокировкой и т.п.
- Отклик может быть задержан сложной обработкой на получателе
- Надо учитывать задержки межсетевых экранов и аналогичные проблемы
- Проблемы могут быть в смежных системах или транспорте

Итоги

- Устойчивость интеграции определяется проектированием
- Тестирование интеграции начинается с анализа проектных решений на чувствительные проблемы
- Необходим комплексный взгляд: ваша система, смежные системы, уровень транспорта
- Залог устойчивой работы – хорошая админка для быстрого решения инцидентов

Вопросы по докладу и обратная связь

Максим Цепков

http://mtsepkov.org

@MaximTsepkov

На сайте много материалов по анализу и архитектуре, Agile и менеджменту самоуправления, моделям soft skill, мои доклады, статьи и конспекты книг

Вакансии

Пишите на hr@custis.ru, подходите с вопросами!