

## Проблемы и варианты корпоративной цифровой трансформации

Юхновский И.А. ([juhnowski@gmail.com](mailto:juhnowski@gmail.com)),

*Образовательно-научный институт ядерной энергетики и технической физики им. академика Ф.М. Митенкова Кафедра «Биоинженерия и ядерная медицина», г. Н.Новгород, ул.Минина, д.28Л, 5 корпус, ауд. 5302,*

Вначале разрешите дать объяснение, что ввиду болезненности и критичности для бизнеса, я не буду в примерах давать конкретные ссылки на корпорации в которых наблюдалось описываемые явление, считаю что они характерны для многих организаций и уверен, что вы на практике тоже с ними сталкивались, а раз признаки имеют более или менее повсеместный характер, то конкретизация будет излишней.

Бизнес-процессы корпорации могут быть описаны с помощью диаграмм, отображающих этапы, наложенные на организационную структуру. Каждому этапу присущи бизнес-системы, порождаемые ими артефакты и связанные с ними исполнители.

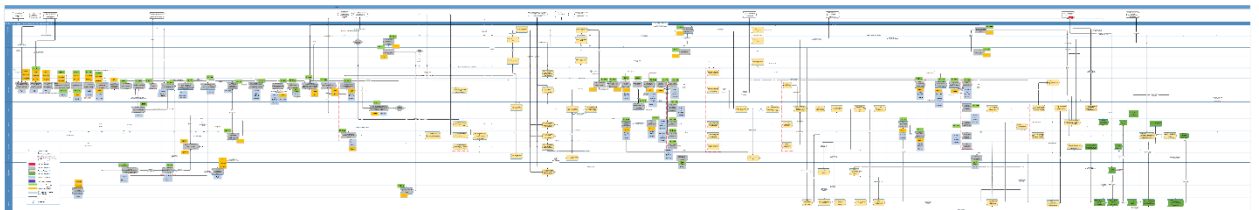


Рисунок 1 – пример визуализации реального бизнес-процесса предприятия

Взрывной характер процессов цифровизации в современной жизни имел одинаковые последствия в разных отраслях промышленности, а именно:

- слабая интеграция информационных систем между собой (экспорт импорт файлов, ручной ввод)
- отсутствие единой модели данных предметной области
- отсутствие моделей сквозных технологических конвейеров данных

Это привело к множеству проблем при построении экосистем предприятий. Как правило, все выбирают архитектурный шаблон для решения данной проблемы – разбиение по слоям: слой данных, слой приложений, интеграционный слой, ИТ ландшафт, слой безопасности и т.д. после чего стараются имплементировать их в облачном пространстве по архитектуре as a Service (PaaS, SaaS, IaaS и т.д.)

Однако на этом этапе не все гладко. Как правило, корпоративная архитектура идет в разрез с бизнес-целями отдельных подразделений. Например, с корпоративной точки зрения выгодно реализовать шаблон SSO (Single Sign On) единой точки входа пользователей для авторизации и аутентификации, но с точки зрения владельца бизнес-системы или какого-то бизнес-решения это необоснованная трата ресурсов, так как у них уже реализована система SSO, и замена локальной на корпоративную не приносит бизнес-эффекта, либо эффект незначительный. Аналогично и с другими системами. Именно поэтому происходит дублирование функциональности в различных бизнес-системах или даже самих бизнес-систем. Подразделения нацелены на максимизацию прибыли, а цифровизация может приводить к рефакторингу значительной части кода используемых информационных систем, что влечет значительных капиталовложений, причем, на первых этапах без добавления новой функциональности. Вообще, в корпоративном секторе есть правило «работает и хорошо, руками не трогай». Дублирует ПО функционал, ну и ладно, нельзя его просто взять и остановить.

Другая сложность, с которой сталкиваются корпорации – это невозможность реализовать процесс цифровизации самостоятельно: необходимо привлечение консалтинговых фирм, интеграторов, сторонних разработчиков. Частично проблема решалась созданием информационных дочерних организаций, но сталкивалась с другой проблемой – «оторванности» цифровизаторов от бизнеса. Разработчики начинали придерживаться корпоративных стандартов и процессов, что как было сказано выше, не отвечало запросам спонсирующих

их бизнес-подразделений, и, как результат, приводило либо к поглощению подобных выделенных структур, либо к снижению их роли при принятии решений в цифровой трансформации.

Как видим процесс цифровой трансформации, который идет в большинстве крупных корпорациях сложен и, если начинать проводить его комплексный анализ, то вместо ответов мы получим еще больший перечень проблем и вопросов.

Встает вопрос, так как же тогда подходить к процессам цифровизации? Опыт показал, что организации, которые используют событийную модель в локальных процессах цифровизации находятся в более выигрышном положении, чем те, кто старался на уровне корпоративной архитектуры построить цифровую систему корпорации. Проще разработать самовоспроизводящийся механизм, который построит всю систему, чем строить систему из одного центра по жестким правилам. К сожалению, многие корпоративные архитекторы не разделяют эту точку зрения и под одну гребенку начинают причесывать все бизнес-системы: деление по слоям, интеграционные шины, «сервис мэш», микросервисная структура, распиливание монолита, следование принципам корпоративной архитектуры и т.д. сегодня звучат у всех на устах. А вот такие слова как реактивная архитектура, событийная модель – редки для корпораций, а это именно тот самый самовоспроизводящийся механизм.

Рассмотрим на примере два этапа бизнес-процесса. У этапа будут события, на которые происходят реакции исполнителей этого этапа, некоторые действия, приводящие к загрузке, трансформации и передаче на следующий этап данных в виде некоторых артефактов. Также мы сможем выявить блокировщика, исполнителя, который проверяет артефакты и разрешает их передачу на следующий этап. Все эти сущности можно представить в виде абстракции – акторов. Актор это программная единица, которая реагирует на события, производит некоторые действия и создает новое событие. Соответственно, мы

получаем коммуникационную модель, в которых акторы общаются друг с другом и результатом их общения являются артефакты, содержащие данные бизнес-процесса. Разработав такую программную систему из акторов для одного этапа, не представляется сложным добавить к ней акторов второго этапа, так реализовав бизнес-процесс, который будет иметь природу его составных частей, т.е. актора, а следовательно, сможет коммуницировать на более высоком уровне с другими бизнес-процессами, бизнес-кейсами.

Не буду погружаться в детали реактивной архитектуры, а перечислю только ее основные преимущества [1]:

- реактивность (Reactive (disambiguation))
- изоляция (Isolation)
- автономия (Autonomy)
- мобильность и адресность (Mobility and Addressability)
- событийность (Message Driven vs Event Driven)
- инкапсуляция состояния (State Ownership)

Реактивная архитектура более проста в понимании и в реализации чем традиционная микро сервисная архитектура и ее не сложно реализовать в рамках импортозамещения, чем ту же «сервисную шину данных» ESB или «сервис мэш». Но есть и недостатки.

В отличии от омниканальной архитектуры придется уделять внимание безопасности передачи событий и сообщений. Для этих целей можно использовать брокер сообщений, но если каждый акт обмена сообщениями будет проверяться службами безопасности, то это может критически снизить производительность. Выделение акторов в отдельный защищенный сетевой сегмент в качестве решения проблемы приведет к автоматическому появлению демилитаризованной зоны и вопросам решения безопасного сетевого обмена между зонами.

А теперь самое главное преимущество – это бесшовность цифровой трансформации. При наличии несовершенных, постоянно меняющихся бизнес-кейсах, пусть даже и не формализованных в бизнес-процесс есть всегда возможность их автоматизировать – представить в виде актора или группы акторов, причем совершенно не важно, какой природы будет этот актер: человек или компьютерный код, главное, что есть возможность общения. Т.е. какой бы не был несовершенный процесс, сколько бы мало он не был бы автоматизирован с помощью применения реактивной архитектуры мы получаем незамедлительный результат – цифровую трансформацию, в которой также как и раньше есть несовершенство и ручной труд, но появляется возможность постепенного улучшения процесса и автоматизации, за счет оркестрации и хореографии акторов.

Не требуется для этого подхода и изменения информационных систем, поскольку события могут быть извлечены из логов.

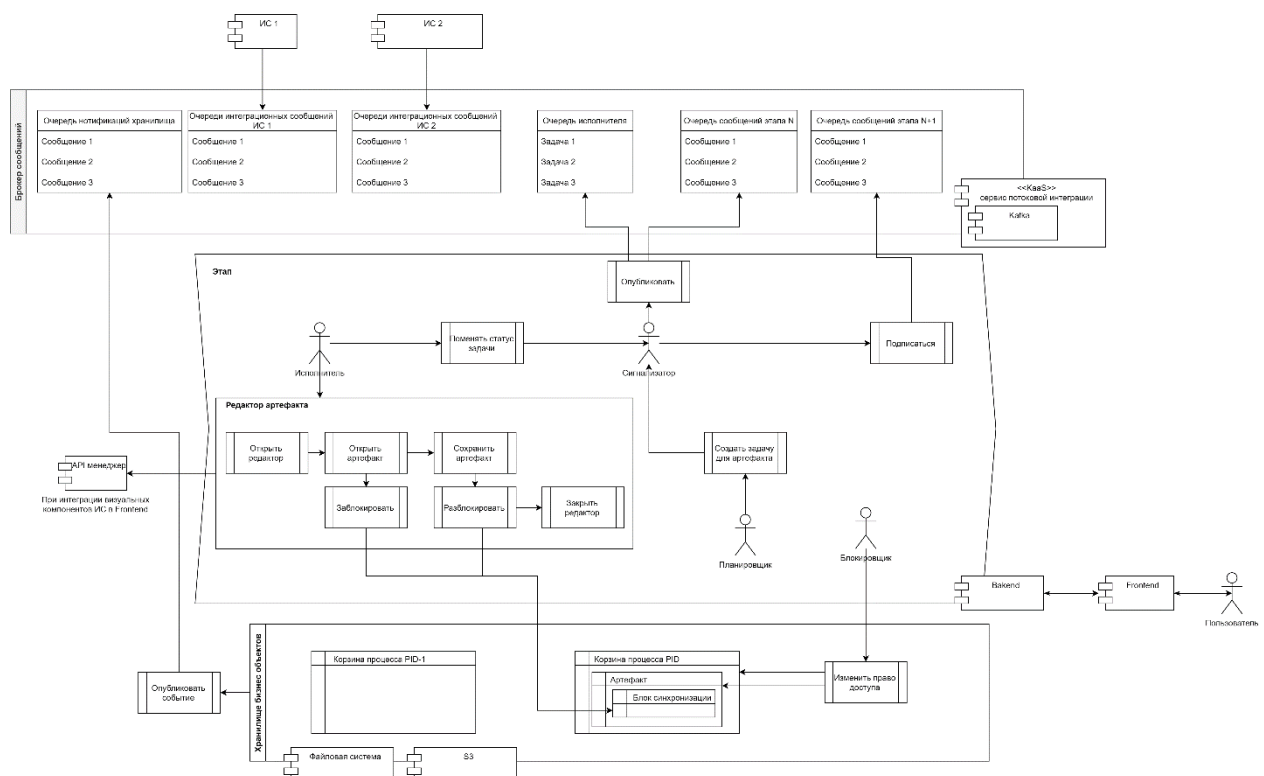


Рисунок 2 - Реактивная архитектура бизнес-процесса

И в заключении хотелось бы обратить внимание на особый случай, когда процесс цифровизации должен быть ограничен ручными операциями. Для промышленности это, наверное, будет не актуально, но важно для медицины.

При системном анализе применения искусственного интеллекта в медицине можно обнаружить эффект необходимости ограничения: с одной стороны пациент счастлив, когда ему незамедлительно оказывается медицинская помощь, но с другой стороны если это делает компьютер, то человек чувствует себя обделенным человеческим вниманием. Пациенту нравится индивидуальный подход и не нравится, когда его ставят на медицинский конвейер. Для таких ситуаций при цифровизации процессов возможны обратные процессы – переход от автоматизированного этапа к ручному труду и для актора, с его коммуникационной природой это будет сделать проще, чем для REST микросервиса, замена которого на ручной труд потребует доработки как бэкенд так и фронтенд приложений. Об этом не стоит забывать архитекторам при цифровизации медицины. Скетч прототипа можно посмотреть в [2], был разработан по материалам [1].

## Литература

[1] <https://developer.lightbend.com/docs/akka-platform-guide/concepts/index.html>

[2] [https://github.com/juhnowski/digital\\_transformation](https://github.com/juhnowski/digital_transformation)