



MARCUS AURELIUS LTD

КРАТКОЕ ВВЕДЕНИЕ В АРХИТЕКТУРУ (V 6.1)



Краткое введение в понятие архитектуры, основанное на опыте анализа и проектирования информационных систем, бизнес-процессов, моделей данных, продуктовых каталогов, от экспертов компании «Марк Аврелий».



Г. МОСКВА 2016-2025 ГОД
Рудь Виктор +7 985 922 12 40



0. Проблема определения архитектуры: операционная и контекстная зависимость определения, операционные аспекты определения.

Определение архитектуры (границы понятия архитектура) охватывает:

- стадии проектирования от идеи до готовой конструкции,
- функциональные и эстетические аспекты конструкции,
- саму конструкцию,
- принципы создания конструкции или основополагающие решения и ограничения.

1. Грань/аспект 1. Статика

2. Грань/аспект 2. Динамика.

3. Грань/аспект 3. Документирование

4. Open Group и Agile Architecture (раздел в разработке)



0. В ЧЕМ СМЫСЛ ОПРЕДЕЛЕНИЯ?

В чем состоит смысл определения и какие бывают определения.

Данный раздел (раздел 0) можно пропустить при первом знакомстве с темой архитектуры.

КАК МЫ ПОДХОДИМ К ВОПРОСУ ОПРЕДЕЛЕНИЙ

Дефиниция (образовано от латинского слова: definitio — предел, граница). То есть наша задача состоит в том, чтобы установить границы, которыми сегодня очерчивается всё то, на что мы указываем словом «архитектура».

Дефиниция или определение — это логическая операция, которая раскрывает содержание имени (здесь – слова «архитектура») посредством описания существенных и отличительных признаков предметов или явлений (денотата имени), обозначаемых данным именем (то есть вербальным знаком).

Определить термин — значит установить границы его применения. Эти границы можно установить в виде некоего определяющего выражения/высказывания, раскрывающего смысл определяемого имени или его значение. **Термин и его определение (дефиниция) должны находиться в отношении тождества**, то есть иметь один и тот же денотат, и быть взаимозаменяемыми.

Дефиниции делят по разным основаниям, в частности по *способу раскрытия содержания* — на *явные* (указываются признаки, присущие предмету или явлению) и *неявные* (выявляются отношения, в которых находится определяемый предмет / явление с другими предметами / явлениями).

ПОДХОДЫ К СОЗДАНИЮ ОПРЕДЕЛЕНИЯ. ЯВНЫЕ ОПРЕДЕЛЕНИЯ

Явные дефиниции делятся (в зависимости от того, к какой языковой категории относится определяемый термин) на следующие виды:

Атрибутивно-реляционные дефиниции, в которых указывается ближайшее родовое отличие и видовой признак, присущий только данному виду (например, «квадрат — это ромб с прямыми углами»).

Архитектура: проектный документ, в котором...

Генетические дефиниции, в которых указывается происхождение или способ конструирования объекта, обозначаемого определяемым именем (например, «сфера — пространственная поверхность, которую описывает полуокружность при вращении её вокруг диаметра»).

Архитектура - замысел, оформляемый в виде схемы и плана и получающий воплощение в виде конструктивных и эстетических особенностей результирующего продукта.
Архитектура – результат работы архитектора.

Целевые дефиниции, в которых указывается на то, как используется определяемый объект, какие функции он выполняет, для достижения каких целей он применяется.

Архитектура – инструмент для планирования будущей деятельности или продукта, или строения.
Архитектура - упражнение для высокооплачиваемых архитекторов, живущих в башнях из слоновой кости.

Квалифицирующие дефиниции, в которых фиксируются, что определяемый объект представляет собой, то есть фиксируются какие-то его структурные особенности, атрибуты, а также особенности внешнего вида.

Архитектура – набор чертежей и/или пояснительная записка, проходящая экспертизу...., в которой раскрывается ... согласно ГОСТ ...

Перечислительные дефиниции, в которых просто перечисляются те объекты, которые подпадают под определяемый термин.

Классика, Барокко, Ампиризм, Конструктивизм, Модернизм, SOA, Микросервисы, Файл-сервер, Клиент-сервер – строго говоря, перечислены не архитектуры, а виды архитектур.

Операционные дефиниции, в которых, в качестве видовой характеристики объектов выступает указание на некоторую операцию, посредством которой эти объекты могут быть обнаружены и обозначено их отличие от других предметов (например, «кислота — вещество, окрашивающее лакмус в красный цвет»).

Архитектура (как и стратегия) – это презентация, сделанная дорогими консультантами, ее можно найти у ИТ-директора или его зама. Там все верхнеуровнево и уже давно не актуально.

ПОДХОДЫ К СОЗДАНИЮ ОПРЕДЕЛЕНИЯ. НЕЯВНЫЕ ОПРЕДЕЛЕНИЯ

Неявные дефиниции устроены весьма сложно, вследствие чего отсутствует способ простого замещения термина на некое замещаемое им выражение. Собственно термин «Архитектура» лучше всего определяется именно таким способом. Виды неявных определений:

- матричное определение логических операций в исчислении высказываний
- лингвистические конструкции вида: A есть то, что удовлетворяет условиям: $B_1, B_2, \dots B_n$.

Аксиоматические дефиниции. Посредством аксиоматических дефиниций некоторый термин определяется путём указания той совокупности аксиом, в которой он содержится. С этой точки зрения аксиомы любой системы являются синтетическими определениями тех терминов, которые в них входят. Так, в научном поиске с помощью аксиоматических определений смысл исходных (примитивных) терминов научной теории задаётся посредством введения системы постулатов, содержащих данные термины и формулирующих те условия, которым обязаны удовлетворять обозначаемые терминами объекты. В математической логике примером аксиоматической дефиниции может служить определение формулы в исчислении высказываний.

Индуктивные дефиниции. Суть таких дефиниций состоит в следующем. Если требуется задать класс предметов, подпадающих под некоторый термин, то мы прямо объявляем некоторые предметы элементами этого класса. Данный пункт определения называется базисом индукции. После этого все остальные предметы, входящие в класс, порождаются с помощью некоторых процедур. Такой пункт определения называется индуктивным шагом.

Контекстуальные дефиниции. Такие дефиниции дают возможность выяснить содержание понятия, не прибегая к толковому словарю, а через предлагаемый контекст. В этом случае говорят о некоторой контекстной зависимости определяемого термина. Определяемый термин помещается в некоторый языковой контекст, а ему приравнивается по смыслу другой контекст, не содержащий данного термина. При этом сам термин «контекстная зависимость» понимается в двух различных смыслах. С одной стороны, речь идёт о получении некоторого неявного знания об интересующем нас термине из рассмотрения некоторого конкретного контекста, в состав которого он входит. В этом случае понимание смысла контекста позволяет предположить и возможное значение соответствующего термина. С другой стороны, речь идёт об определении термина посредством определения всех контекстов, в состав которых он входит. Чтобы задать эти контексты, используют соответствующий метаязык. В первом случае говорят об определении через контекст; во втором — о контекстуальном определении.

Определения архитектур зачастую **аксиоматичны** в рамках тех или иных методологий. Но в целом это порождает **высокую контекстуальную зависимость определений архитектуры.**

БИЗНЕС-АРХИТЕКТУРА

- Архитектура концентрируется на целях конструкции/бизнеса
- Архитектура учитывает ограничения и принципы
- Архитектура проектирует исполнение функций
- Архитектура определяет состав компонентов конструкции/системы
- Архитектура учитывает и предопределяет процессы
- Архитектура учитывает и предопределяет как конструкция будет взаимодействовать со своим окружением



ЗАВИСИМОСТЬ ОТ КОНТЕКСТА И ТЕКУЧЕСТЬ ПОНЯТИЯ АРХИТЕКТУРЫ

Архитектура рассматривается нами в цикле проектирования или реконструкции некого объекта (или системы), в частности – бизнеса. Крупные бизнесы задумываются (пытаются осмыслить) свою архитектуру лишь в периоды кризиса или на пороге неизбежной трансформации.

Стадии проектирования или реконструкции: архитектура - переходный мост от начальной идеи к физической конструкции системы

Стадия-контекст №1.
Идея

Формируется целевое назначение конструкции и гипотезы по ее реализации

Архитектор отвечает на вопрос «ЗАЧЕМ»

Стадия-контекст №2.
Концепция

Формируется осмысленный замысел конструкции и убеждения в ее реализуемости

Архитектор отвечает на вопрос «ЧТО [будем создавать]»

Стадия-контекст №3.
Логически выверенное и полное решение

Формируется точная выверенная модель реализации, не зависящая от физических методов реализации

Архитектор отвечает на вопросы типа «ПОЧЕМУ и КАК»

Стадия-контекст №4.
Конструкторский «чертеж»

Формируется точная выверенная модель реализации, зависящая от физических методов реализации

Архитектор отвечает на вопрос «КАК ИМЕННО и СКОЛЬКО [стоит]»

Стадия-контекст №5.
Готовое изделие, процесс или конструкция

Физически воплощённое изделие или процесс, или бизнес-система

Архитектор отвечает на вопросы типа «Как изменить границы применения»

Архитектура – это стержень архитектора в отношении текущей проектируемой и создаваемой конструкции.

Важно: сначала появляется архитектор. Потом рождается архитектура.

АРХИТЕКТУРА ОБЕСПЕЧИВАЕТ ПРЕЕМСТВЕННОСТЬ



Архитектура соединяет:

Цель + Замысел + Чертеж + Реализация

Архитектура (или архитектор?) обеспечивает преемственность: от функционального и/или эстетического назначения к возможности изделия/системы выполнить возложенные на нее функции.

Архитектура – это необходимый и достаточный набор решений, реализация которых обеспечивает изделию (продукт, услуга, конструкция, система) полное соответствие требованиям заказчика.

Здесь слово «решение» следует понимать как контекстно-зависимое и следовательно весьма разное по своей природе:

- Решение как **намерение** «будем делать так»
- Решение как **приказ** «будем делать именно так»
- Решение как **проект** «запроектировано делать так»
- Решение как **следствие** «сделано так».

АРХИТЕКТУРА – ЭТО РЕЗУЛЬТАТ ДЕЯТЕЛЬНОСТИ АРХИТЕКТОРА

НЕТ СТРАТЕГИИ БЕЗ СТРАТЕГА.

НЕТ АРХИТЕКТУРЫ БЕЗ АРХИТЕКТОРА.

Сначала появляется «Страта» → Потом появляется «Стратег» → Затем рождается «Стратегия».

Сначала появляется Тема → Потом появляется композитор → Затем рождается композиция.

Сначала рождается Тема → Потом появляется режиссер → Затем рождается кинофильм.

Сначала возникает проблема → Потом появляется конструктор → Затем рождается конструкция.

Сначала появляется **потребность** → Потом находится **Архитектор*** → Затем рождается **Архитектура**.

Что роднит драматургию, композицию, режиссуру, архитектуру?

- неотрывность от автора
- наличие команды дизайна-воплощения (авторский коллектив) и зависимость от команды
- творческий аспект
- крайне сложное копирование.

Архитектор осуществляет **управление проектированием** и созданием результата **посредством архитектурного проекта**. В этом особенность присущего архитектору метода управления!!! Архитектору принадлежит право на использование проекта, в том числе путём разработки **документации на создание** и путём реализации архитектурного проекта. То есть **архитектор – тот тот, кто создает ПРОЕКТ проектируемого изделия/конструкции и управляет созданием запроектованного результата**.

ЗАХМАН ОБ АНАЛОГИЯХ СО СТРОИТЕЛЬСТВОМ

[Цитаты из перевода статьи](#) пионера архитектур в области информационных систем – Джона Захмана.

... В поисках объективной, независимой основы, на которой можно разрабатывать рамки для архитектуры информационных систем, кажется логичным, заглянуть в область классической архитектуры, применяемой в строительстве. Сделав это, можно перенять опыт, накопленный за 2.5 тысячи лет в этой области. Определение результатов, то есть, рабочего продукта классической архитектуры, может привести к пониманию аналогий в продуктах (результатах работ) для архитектуры информационных систем и в этом помочь классифицировать концепции и спецификации.

Эта эквивалентность должна усилить аргумент, что аналогичный набор архитектурных представлений будет применен в процессе построения любого сложного инженерного продукта, включая информационную систему.

Перед тем как идентифицировать аналоги для информационных систем, полезно сделать несколько общих наблюдений относительно архитектуры.

Первое, **существуют 3 фундаментальных представления архитектуры**, по одному для каждого «игрока». Это представления для владельца, проектировщика и конструктора. Владелец в голове имеет представление о продукте, который будет служить определенным целям. Архитектор транслирует это понимание продукта в представление владельца. Далее архитектор транслирует это представление в физический продукт, то есть, представление проектировщика. Затем конструктор учитывает ограничения, как связанные с законами природы, так и технологические, чтобы сделать продукт производимым, - это представление конструктора.

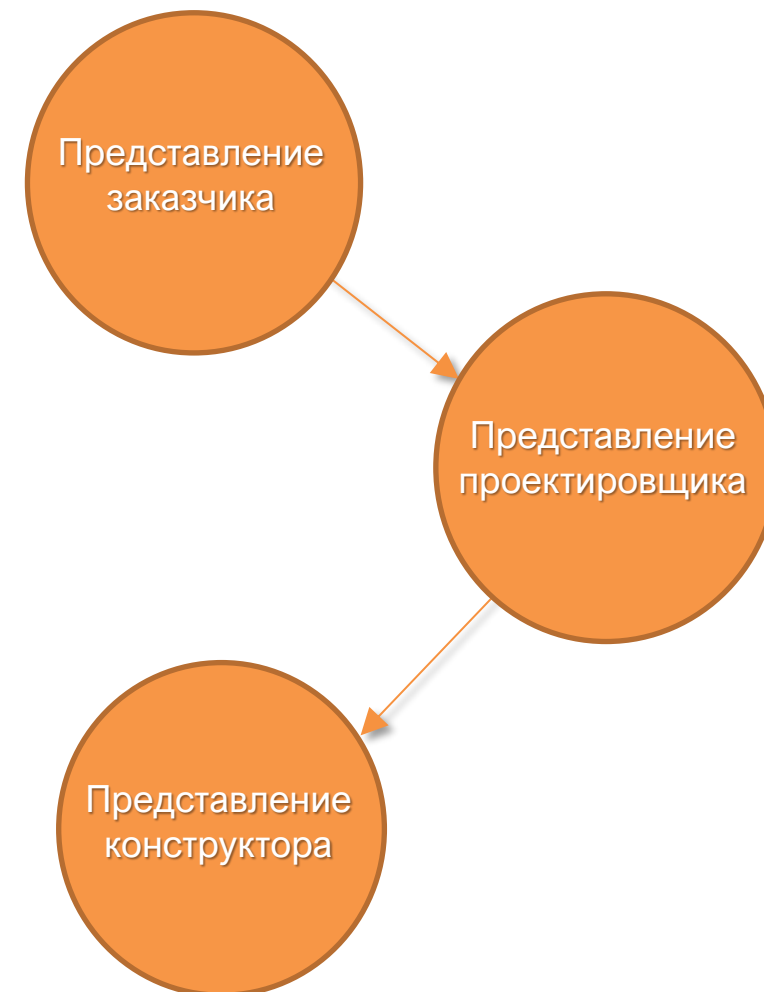
Предшествующее этим 3 фундаментальным представлениям, грубое представление о размере, форме и масштабе создается для того, чтобы установить «площадку» внутри которой все последующие архитектурные активности будут собраны воедино.

После трех фундаментальных представлений идут детальные представления вне контекста, которые технически могли бы быть архитектурами, потому что это все те же представления, только немного не дотягивающие до уровня готового физического продукта. Однако, они немного менее интересны архитектурно, поскольку они не определяют конечный продукт и, в целом, более ориентированы на актуальные задачи по реализации. Тем не менее, они включаются в дискуссию с целью обеспечения всесторонней структуры.

Существенным замечанием по отношению к этим архитектурным представлениям является тот факт, что каждое из них имеет отличную от других природу. Они являются не просто набором представлений, каждое из которых отображает больший уровень деталей чем предыдущие. Уровень деталей независимо разнообразен, варьирующий в пределах любого архитектурного представления. Например, представление проектировщика (т.е. архитектурные планы) это не просто последующий, расширенный деталями уровень представления владельца (т.е. архитектурные наброски). Это другая природа, в содержании, семантике, и так же представление другого ракурса. Уровень детализации представления проектировщика (т.е. планы) разнообразный и не зависит от уровня детализации в представлении владельца (т.е. наброски).

Таким же образом, каждое архитектурное представление отличается от других по своей сути, а не просто по уровню детализации.

Учитывая данное определение точек зрения (т.е. точек зрения владельца, проектировщика, конструктора и так далее) на различные представления архитектуры, выработанные в процессе построения сложного инженерного продукта, относительно просто идентифицировать аналоги в области информационных систем. Так как **информационные системы также являются «сложными инженерными продуктами».**



ТРИ ГРАНИ АРХИТЕКТУРЫ, КОТОРЫЕ СЛЕДУЕТ ПОСТОЯННО ДЕРЖАТЬ ВО ВНИМАНИИ

1. Архитектура. Статика. Архитектура – способ организации системы:

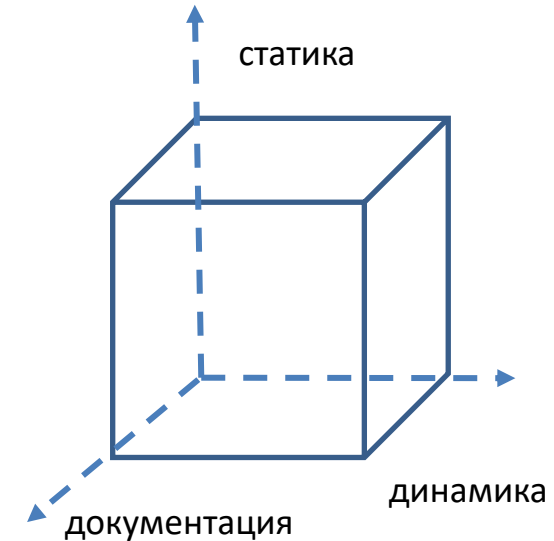
- **Архитектура** – структура **компонентов** системы, их **отношения** друг к другу.
- **Архитектура** – фундаментальная организация системы (например, бизнес-системы), встроенная или присущая ее компонентам, отношение компонентов друг к другу и к внешней среде.

2. Архитектура. Динамика. Архитектура – это процесс проектирования (архитектурирование) будущих свойств системы, как методом эволюции элементов системы так и методом революции – тотальной трансформации системы:

- **Архитектура** - **детальный план** системы на уровне компонентов для управления ее внедрением
- **Архитектура** – это структура компонентов, принципы и общие **правила управления** их проектированием и **развитием в течение времени**.
- **Архитектура** – фундаментальная организация системы, а также **принципы управления дизайном и эволюцией этой организованности**.

3. Архитектура. Набор артефактов. Архитектура – это набор **формализованных описаний** системы.

Архитектура (архитектурная документация) представляет собой описание архитектурных *строительных блоков*, организацию их в виде *архитектурных каталогов*, спецификацию взаимосвязей между этими строительными блоками в *архитектурных матрицах* и в виде *иллюстрирующих диаграмм*, наглядно и с нужной степенью детальности проясняющих ту или иную *точку зрения* для одного из *заинтересованных лиц*, озабоченных поиском решения, направленного на удовлетворение для своих *интересов (concerns)*.

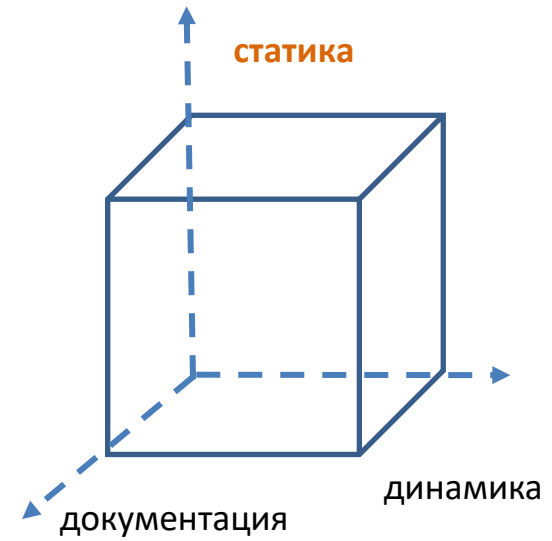


- ❖ Мы рассуждаем об архитектуре, потому что «есть потребность **понять и управлять теми элементами** рассматриваемой системы, **которые влияют на ее полезность, стоимость**, временные характеристики и риски в пределах ее окружающей среды. В некоторых случаях основным элементом являются физические или структурные компоненты системы, а также их отношения. Иногда основными элементами являются функциональные и логические элементы (в том числе явления и информация, примечание В.Рудь). В других случаях, элементами могут быть всеобщие принципы и patterns».



ГРАНЬ №1.

АРХИТЕКТУРА. СТАТИКА*



* Статика:

мы вкладываем в это слово основной архитектурный смысл: принцип декомпозиции системы на ее составляющие со всеми их взаимосвязями и взаимовлияниями. Ключевые слова: композиция-декомпозиция.

СУЩНОСТЬ СТАТИЧЕСКОГО АСПЕКТА АРХИТЕКТУРЫ

☐ Перед нами организация (предприятие) и нам нужно понять сущность ее организованности: что делает организацию организацией? Это нужно понять, чтобы можно было её аккуратно и точно совершенствовать (сегодня чаще пишут - трансформировать).

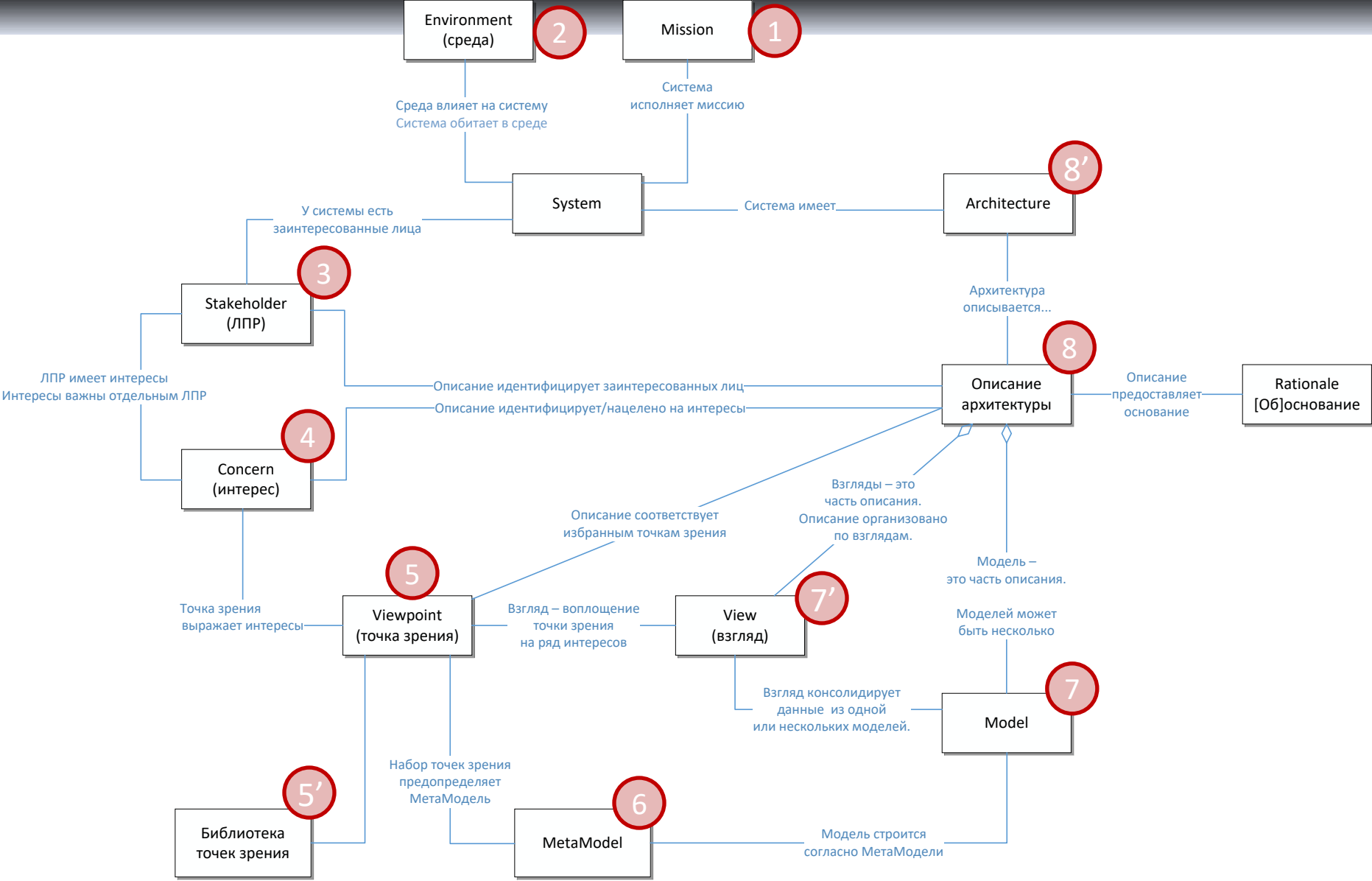
☐ Для исследования организованности компании используется системный подход, рассматривающий компанию как систему, то есть как совокупность взаимодействующих (взаимосвязанных) компонент. При этом системность компании может быть исследована с различных точек зрения.

☐ Система характеризуется/определяется архитектурой, где архитектура – это и ключевые компоненты, и определенные их ключевые свойства, а также ключевые связи между компонентами. Архитектура – это суть организованности.

☐ Методом исследования системы и ее архитектуры является моделирование. Модель отражает одну или несколько точек зрения на системность предприятия, что позволяет исследовать организованность предприятия с этих точек зрения и понять его архитектуру.

Очевидно, что точек зрения и моделей может быть несколько. Это нужно для того, чтобы было проще проводить исследование того или иного аспекта организованности.

МЕСТО АРХИТЕКТУРЫ СРЕДИ ДРУГИХ ПОНЯТИЙ



СИСТЕМА & АРХИТЕКТУРА: КЛЮЧЕВЫЕ ОПРЕДЕЛЕНИЯ

- Система** – коллекция/набор компонентов, организованных для выполнения определенной функции или набора функций.
- Stakeholder** – персоналии, которые играют ключевую роль в системе или которые имеют к ней какой-либо интерес, например, пользователи, разработчики, менеджеры. Разные персоналии в разных своих ролях имеют разные интересы.
- Concerns** – ключевые интересы, которые критически важны для stakeholder'ов системы и по которым принимается решение о приемлемости системы. Интересы могут касаться любых аспектов системы, в том числе производительности, надежности, безопасности, развертывания и т.п.
- Архитектура системы** – фундаментальная организация системы, воплощенная/присущая компонентам системы, отношения этих компонент друг к другу и к окружающей среде, а также принципы управления ее дизайном и эволюцией.
- Описание архитектуры** – набор артефактов, которые документируют архитектуру. **View** – ключевой артефакт в описании архитектуры.
- View** – репрезентация всей системы с точки зрения определенных интересов (concerns).
- Viewpoint** – определяет точку зрения, на основании которой строится и используется View, например, в виде шаблона или схемы.

Система – это набор компонентов, объединенных для выполнения определенной функции или набора функций. Термин «система» охватывает:

- отдельные приложения,
- системы в традиционном смысле,
- подсистемы или надсистемы (системы систем),
- линейки продуктов, семейства продуктов,
- компании и группы компаний (корпорации и другие агрегации).

Система существует для выполнения одной или более **миссий** в своем **окружении**.

Окружение, или **контекст**, определяет ход и обстоятельства экономических, эксплуатационных, политических и других влияний на систему.

Миссия – это применение или действие, для которого одно или несколько **заинтересованных лиц** планируют использовать систему в соответствии с некоторым набором условий

АРХИТЕКТУРА. СТРУКТУРНЫЕ ОПРЕДЕЛЕНИЯ



Архитектура – фундаментальная организация системы (например, бизнес-системы), встроенная или присущая ее компонентам, отношение компонентов друг к другу и к внешней среде.

Архитектура – формальная организация системы или детальный план системы.

Архитектура – структура компонентов, их отношения друг к другу.

Архитектура – организационная логика, лежащая в основе бизнес-процессов и ИТ-инфраструктуры компании, отражающая требования интеграции и стандартизации, вытекающие из операционной модели компании.

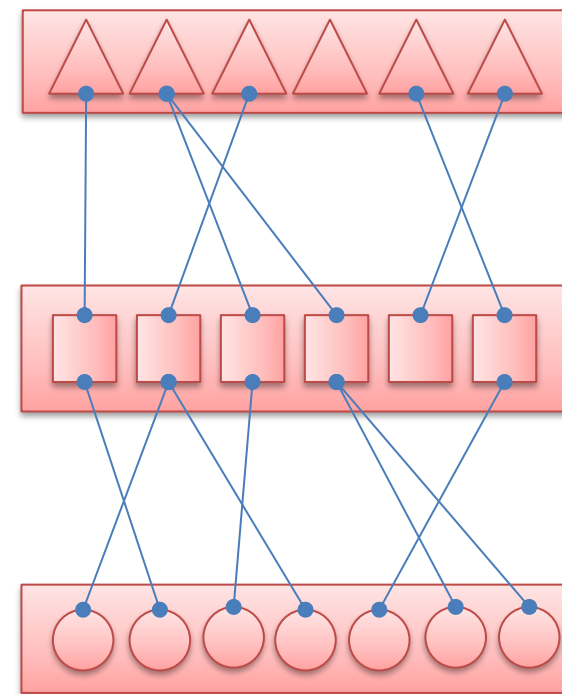
ДЕКОМПОЗИЦИЯ СИСТЕМЫ И ЕЁ АРХИТЕКТУРЫ

Бизнес-система может быть разделена на компоненты с различных точек зрения.

Бизнес-система может быть разделена на компоненты или подсистемы, каждая из которых может быть снова рассмотрена, как система. Это делает архитектурную картину весьма сложной и даже запутанной, тем более если над ней работает несколько архитекторов. Поэтому при комплексном описании архитектуры прибегают к исследованию и описанию (в том числе методом моделирования) отдельных слоёв предприятия/системы, где **слой - это совокупность компонентов однородной природы** (одного типа, вида или класса).

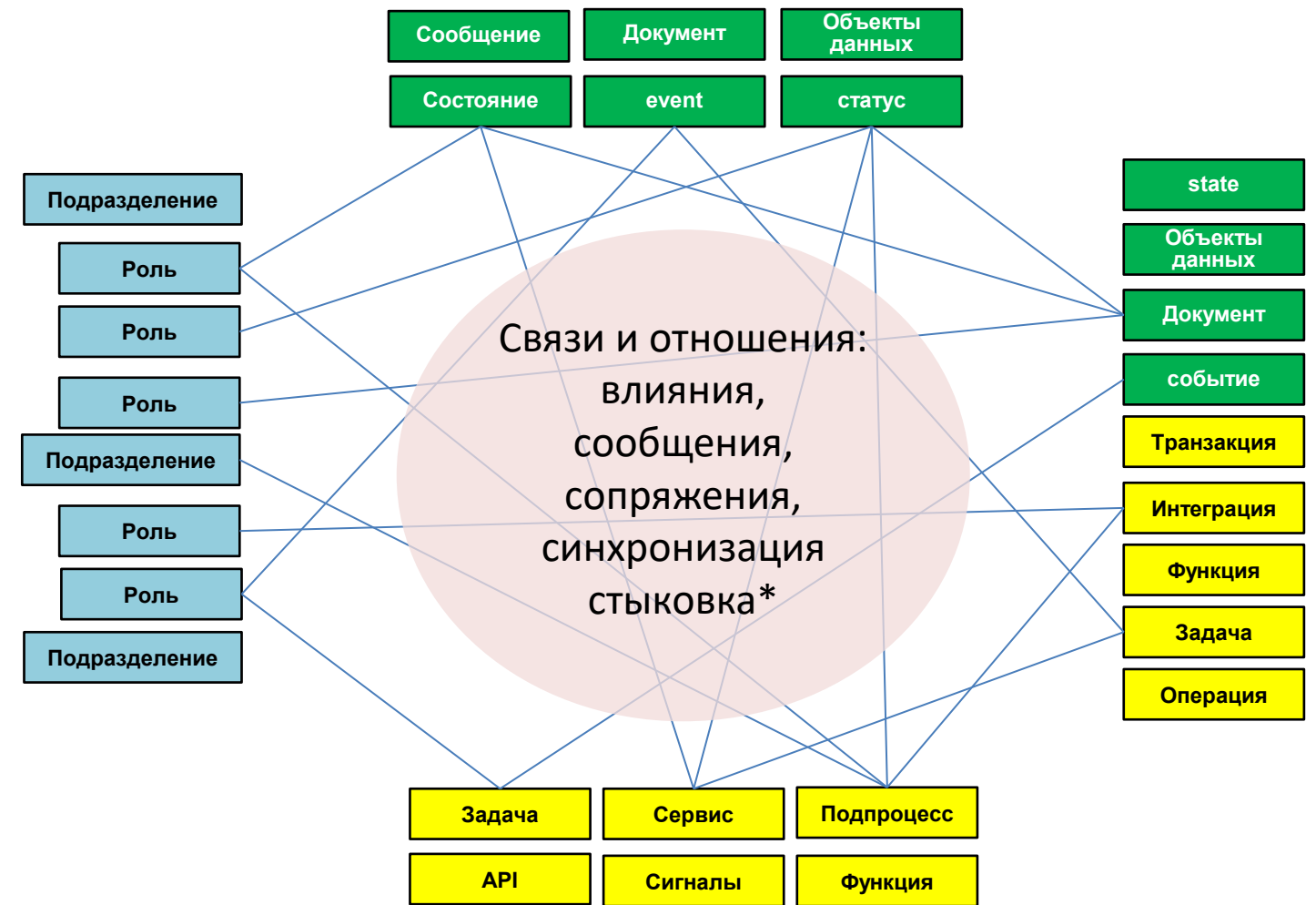
На сегодня в теории методологически хорошо проработаны подходы к описанию следующих слоёв (с целью методичности изложения слои группируют в домены):

- ❖ **Контекст и целеполагание.** Домен содержит компоненты, образующие такие слои, как цели, драйверы, ограничения, оценки ...
- ❖ **Бизнес-домен.** Домен содержит компоненты, образующие такие слои, как продукты, процессы, проекты, функции, подразделения, роли ...
- ❖ **Домен приложений.** Домен содержит компоненты, образующие такие слои, как приложения, функции, сервисы, интерфейсы...
- ❖ **Домен данных.** Слои логических и физических объектов данных. Глоссарий и онтология.
- ❖ **Домен инфраструктуры.** Домен содержит компоненты, образующие такие слои, как каналы, узлы, серверы, базы данных, станки, линии, оборудование, здания и т.п.



В ИТОГЕ АРХИТЕКТУРА СИСТЕМЫ МОЖЕТ БЫТЬ РАССМОТРЕНА, КАК СЛОЖНАЯ КОМПОЗИЦИЯ АРХИТЕКТУР ОТДЕЛЬНЫХ ДОМЕНОВ ИЛИ СЛОЁВ.

СВЯЗИ – ЭТО СТРУКТУРА / СПОСОБ «СОПРЯЖЕНИЯ»



Следует четко понимать, что архитектура это и про **компоненты** рассматриваемой системы (спектакля, здания, фильма, конструкции, бизнеса) и про **связи/отношения** этих компонентов друг с другом.

Одни и те же компоненты могут взаимодействовать разным способом. Одни и те же ноты могут породить разные мелодии. Одни и те же люди могут выдать драматически разные результаты в зависимости от того, как будут налажены взаимодействия между ними.

* Как внутри системы, так и за ее пределами

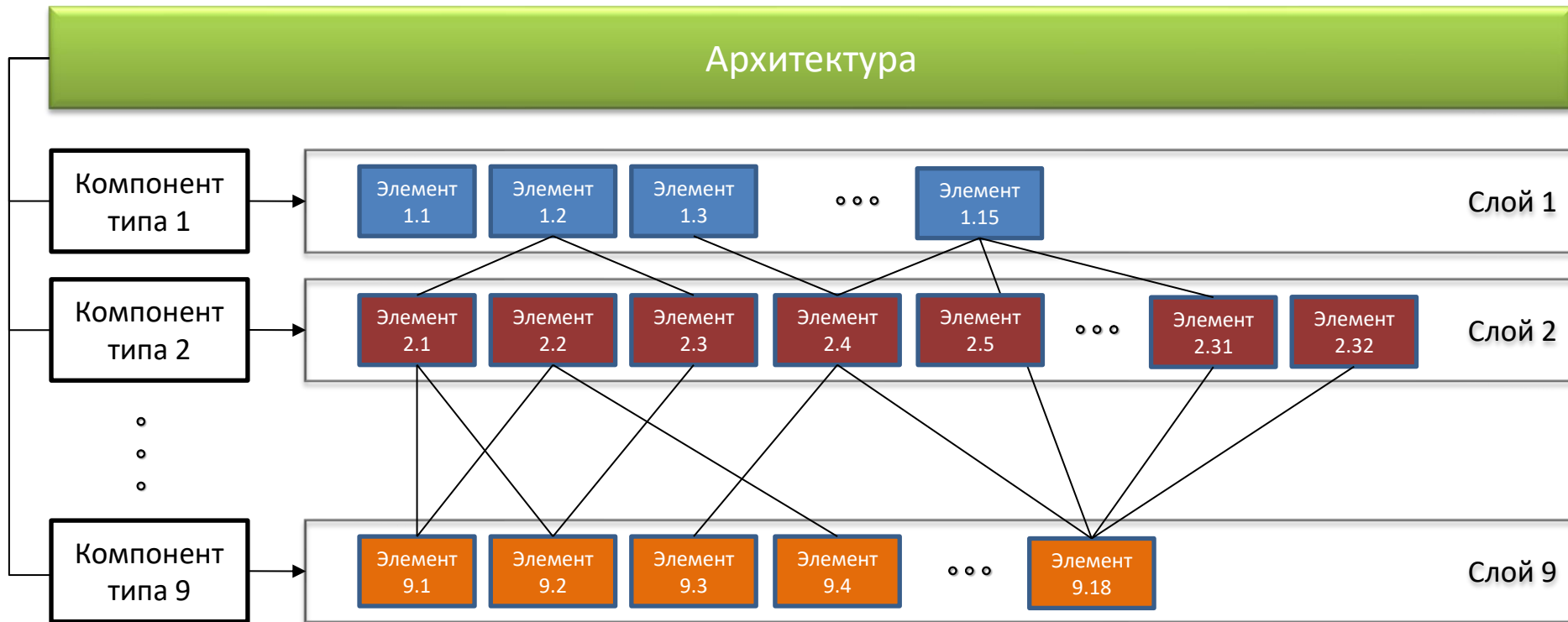
ЗАМЕЧАНИЕ ПО КОМПОНЕНТАМ



Стоит еще раз обратить внимание на упоминаемый в тексте термин – компонент архитектуры. Что следует принять во внимание и запомнить:

- Компонент архитектуры **может быть физически конкретным** – это может быть и часто есть инвентаризируемый объект реальности: система, подразделение, прибор, узел сети.
- Компонент архитектуры **может быть когнитивным (логическим)**, то есть существовать только в виде формулировки. Когнитивная стабильность достигается стабильным терминологическим аппаратом и языком, строгими договоренностями участников по терминам, четкостью суждений. Примеры: функция подразделения, цель, требование, ограничение, интерес.
- Компонент архитектуры **может быть программно-информационным**: API-метод, микросервис, функция приложения, объект данных.
- Компонент архитектуры **может быть абстрактным** – классом, категорией, типом, ролью. В таком случае от архитектора требуется понимание связи абстрактной категории с конкретными компонентами: физическими сущностями или явлениями предприятия, которые данная абстракция репрезентирует.
- Компонент архитектуры **может быть связью или отношением**. Это для многих является открытием, но связи могут быть «материальными»: канал связи, трубопровод, интеграционное взаимодействие двух приложений, совет директоров. Менее очевидное: «билет» есть связь между «пассажиром» и «поездом». Назначение Петрова на должность начальника отдела продаж выражается связующим компонентом с типом «Назначение». Корзина связывается покупателя с выбранным им товаром.
- Компонент архитектуры **должен быть паспортизирован** независимо от того, отражает он конкретное физическое явление/предмет или некую абстракцию над ними. Иначе такой компонент будет недолговечным и пустым по своей сути.

НЕ ПРОСТО СВЯЗНОСТЬ - КОНГРУЭНЦИЯ



Элементы не просто «нарезаны» в слоях. Нужно правильно выбирать уровень абстракции. Элементы между собой и между слоями не просто связаны друг с другом. Эта связь есть вид взаимодействия или зависимости.

Существует также межслоевая гармония, которую лучше всего назвать конгруэнцией.

ВИДЫ АРХИТЕКТУР СОГЛАСНО TOGAF

TOGAF рассматривает [единую] архитектуру предприятия как композицию или агрегацию следующих архитектур, методически проработанных в рамках TOGAF/Archimate:

- **Business Architecture** – описывает бизнес, как он организован, что производит, ключевые бизнес процессы и функции.
- **Data Architecture** – описывает структуру логических и физических данных.
- **Application Architecture** – описывает состав приложений, их взаимодействия, их отношений к ключевым бизнес процессам организации и данным.
- **Technology Architecture** – описывает логические возможности software & hardware, необходимые для развертывания business-, data- и application- сервисов. Это включает ИТ инфраструктуру, системное ПО, сети, коммуникации, стандарты и т.д.

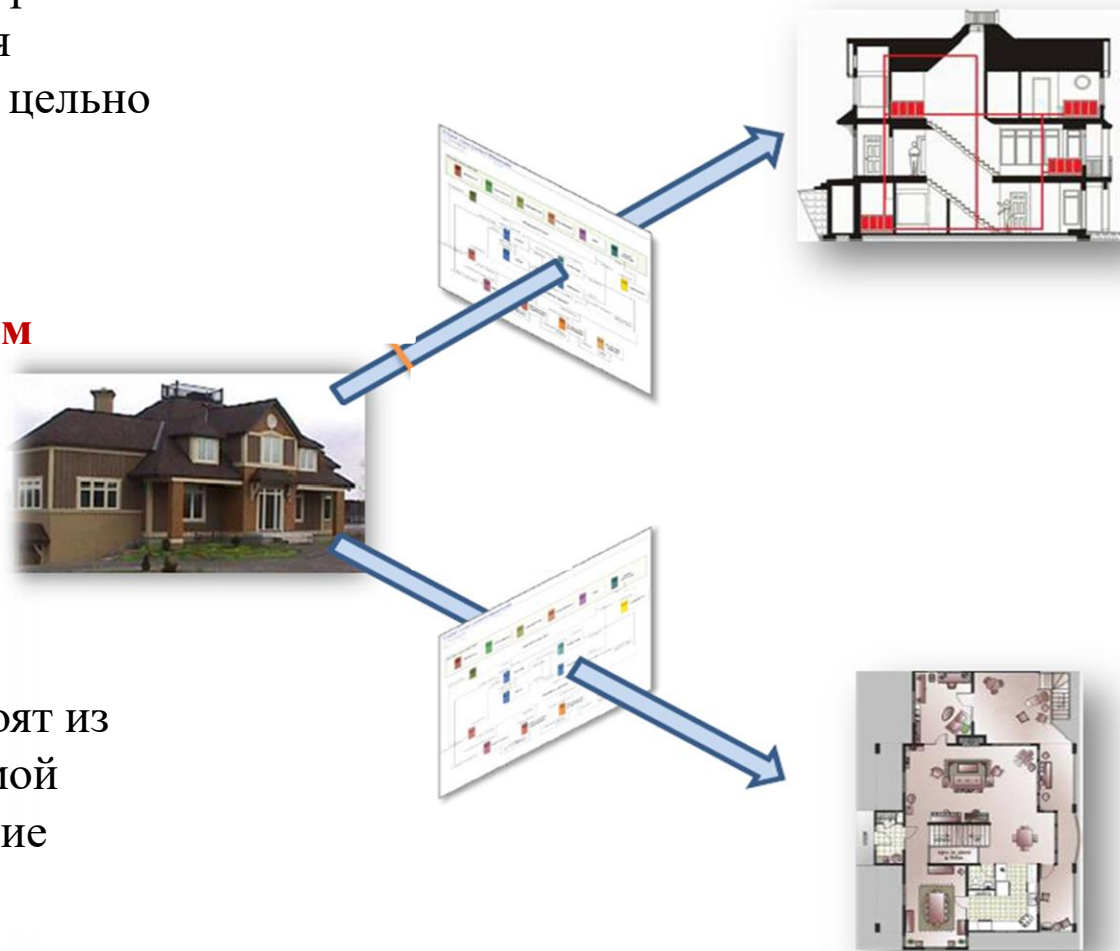
**Архитектура предприятия – это совокупность всех перечисленных выше архитектур
+ увязка их с контекстом/окружением предприятия.**

Примечание: практика компании «Марк Аврелий» расслаивает архитектуру в среднем на 10-20 слоев (до 25-30 в крупных проектах), считая перечисленные выше архитектуры (Data, Business, Application, Technology) частными видами композиции.

Чтобы понять/исследовать архитектуру системы нужно построить модель системы. Модель нужна также и для проектирования системы. Таким образом архитектура наиболее лаконично и цельно выражается в модели. **Модель – это не реальность!**

Модель – это такое **упрощенное представление о реальном** (реальности или действительности), которое позволяет делать на ее основании выводы и принимать решения, которые по качеству не уступают исследованию реального объекта.

Модель строится по определенным правилам, которые состоят из ряда продуманных упрощений, соответствующих исследуемой точке зрения. Одним из видов моделей являются графические нотации моделирования: UML, BPMN, Archimate.



Модель - M является моделью системы – S, если M может быть использована для получения ответов на определенный ограниченный круг вопросов относительно S.

Моделирование - разработка модели системы с целью решения определенного ряда задач путем прояснения свойств системы с определенной точки зрения на систему.

Целью моделирования является поиск ответов на вопросы касательно системы – S по данным модели - M.

Моделирование бизнеса - разработка модели организации (бизнес-системы) с целью решения определенных задач (concerns), стоящих перед менеджерами организации, в соответствии с избранной точкой зрения этих менеджеров на организацию.

ЯВЛЯЮТСЯ ЛИ МОДЕЛИ ПРОСТЫМИ [МОДЕЛЯМИ]?

Мы часто подчеркиваем, что модель это упрощенное описание реальности. Это может создать мнение и даже уверенность, что модели, будучи упрощенными описаниями будущего или существующего предприятия, являются ПРОСТЫМИ моделями. Однако это далеко не так.

- Во-первых, модель – это описание реальности (текущей или будущей). То есть мы должны видеть за моделью реальность. Понятное дело, что кому-то надо видеть совсем немного: сколько подразделений будет в компании или сколько процессов нужно запустить для результативной деятельности и кто их будет выполнять. Кому-то модель служит детальным чертежом, имеющим целью создание изделия, программы, конвейера или программного продукта, например, диаграмма классов UML с указанием ключей и атрибутов.
- Во-вторых, модели имеют два существенно разных метода упрощения. Первый – это убрать лишние элементы, например, директор по продажам хотел бы видеть лишь функции системы CRM и действия каких сотрудников эти функции поддержат. Второй – это повышение уровня абстракции, как метод упрощенного взгляда на реальность, например, генеральный директор хотел бы видеть лишь обобщенные функции предприятия для их распределения среди заводов: финансы, логистика, маркетинг, продажи, ведение кадров, налоговый учет.
- В-третьих, это понятие существенности и влияния. То, что одни находят существенным, для других – лишняя информация, затуманивающая взгляд.
- Четвертое. Многие не готовы воспринимать модели! Более-менее легко понять структурную модель, например, состав систем и их модулей, орг.структуру компании, но весьма сложно понять информационную модель или поведенческую. Так, например, известно, что модель процессов предприятия способны понять только те, кто ее создавал. А модели данных боятся даже создавать для широкого круга лиц.
- Пятое. Если моделей много и каждая из них великолепно отражает чей-то интерес, это не значит, что модели согласованы между собой. Это согласование моделей может стать для кого-то не просто кошмаром, но в принципе интеллектуально или инструментально не решаемой задачей.

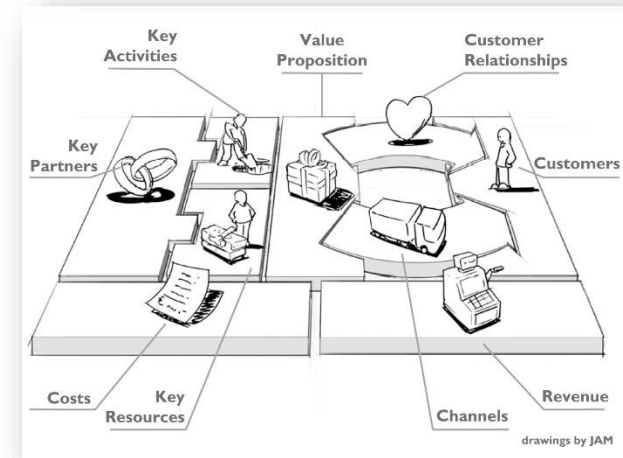
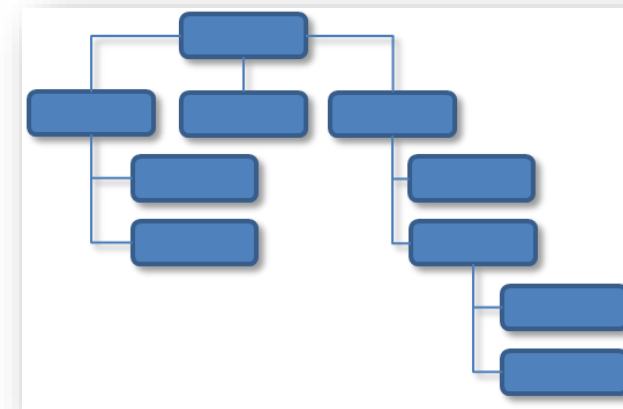
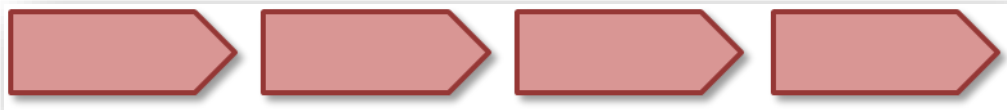
Стоит обратить внимание, что модели, которые приходится строить, не всегда можно классифицировать, как «архитектурные». Модель, в общем случае, есть проект будущего изделия: это может быть и чертеж, и макет, и даже прототип будущего изделия. Орг.структура будущей компании – это тоже модель для набора персонала с нужными компетенциями. Модель женского платья – это точная копия будущих изделий, под которые придется настроить пошивочную линию. Что же делает модель именно архитектурной? – только лишь выделение наиболее существенного. Но такое выделение может быть сделано лишь для упрощения восприятия теми лицами, которые не готовы (интеллектуально или ввиду ограничений по времени) воспринять полную модель со всеми ее деталями.

ВИДЫ МОДЕЛЕЙ

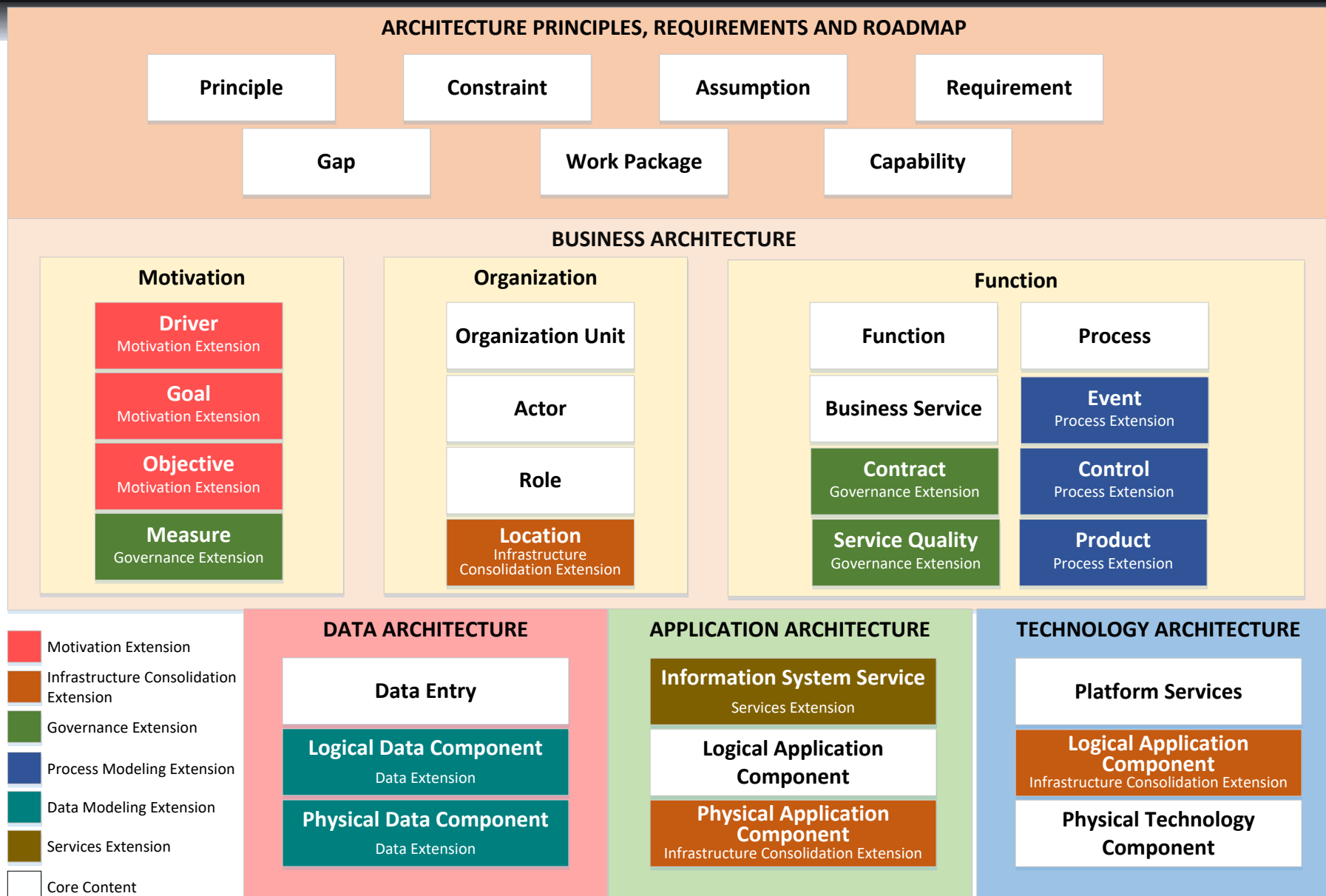
В зависимости от того, какую именно точку зрения мы собираемся адресовать методом моделирования, получится та или иная модель:

- ❖ **Бизнес-модель** - модель ключевых идей касательно устройства предприятия, позволяющих отнести его к определенной типовой группе
- ❖ **Функциональная модель** - модель функций, выполняемых в компании
- ❖ **Информационная модель** - модель информации, циркулирующей в компании.
- ❖ **Процессная модель** - модель процессов
- ❖ **Value-chain** – модель прибавки ценности

Это наиболее частые модели, используемые для анализа или инжиниринга бизнеса. Но, безусловно, это не все возможные модели.

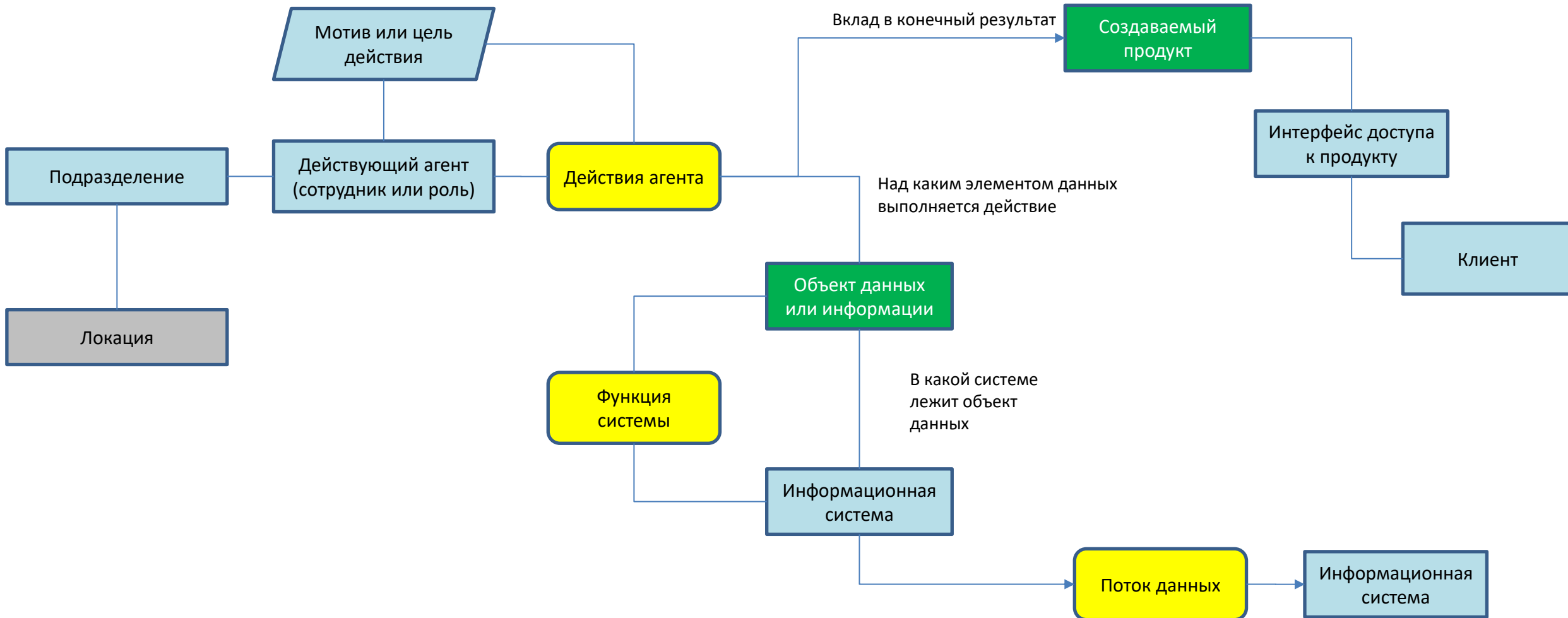


– МОДЕЛЬНЫЕ ЭЛЕМЕНТЫ



ПРИЧИНЫ ПОЯВЛЕНИЯ КОМПОНЕНТОВ

На слайде изображены уже ставшие привычными к 2020 году компоненты моделирования. Основания для подобного рода компонентизации можно прочесть в GERAM (референсный фреймворк для построения архитектур).



СТРОИТЕЛЬНЫЕ БЛОКИ (BUILDING BLOCKS) АРХИТЕКТУРЫ

Элементы и связи, перечисленные на предыдущих двух страницах, могут образовывать строительные блоки архитектуры, то есть такие законченные **блоки/кирпичики, которые можно отдельно развивать, а главное - переиспользовать.**

Умение выделить строительные блоки – это, скорее, искусство, так как выделять эти блоки можно с разной степенью гранулярности.

Задача архитектора заключается в том, чтобы развивать эти кирпичики от одного архитектурного проекта к другому, концентрируя вокруг этих кирпичиков нужную предприятию функциональность (или там capability).



Наиболее часто в качестве строительных блоков используют:

- Сервис
- Функция или пакет функций
- Процесс
- Данные (как информационные единицы, вокруг которых проектируется поведение)
- Приложение
- Актер (должность, действующее лицо)
- Node (узел или сервис инфраструктурного уровня)

Язык/нотация Archimate определяет гораздо больше кирпичиков для работы в слоях бизнеса, приложений и данных, расширяя тем самым детальность и выразительность архитектурных схем.

Задача архитектора и конструктора – создание системы, проявляющей эмерджентные свойства!

АКЦЕНТ №4. ТОЧКА ЗРЕНИЯ. ПРЕДПОСЫЛКИ

Анализ деятельности предприятия, поиск причин его не эффективной работы или поиск новых драйверов роста заставляет рассматривать/моделировать предприятие с различных точек зрения:

- ❖ Предприятие, как **процесс или совокупность процессов**
- ❖ Предприятие, как **система**
- ❖ Предприятие, как **совокупность индивидов**
- ❖ Предприятие, как **иерархия структур и функций**
- ❖ Предприятие, как **совокупность информации и данных**
- ❖ Предприятие, как **совокупность синхронизированных информационных потоков**
- ❖ Предприятие, как **набор целей и механизмов их достижения**
- ❖ Предприятие, как **целеустремленная система с набором драйверов, мотивов, рисков и ограничений**
- ❖ Предприятие, как **часть** более широкого **социального или культурного контекста**

Ни одна точка зрения не даёт универсального представления о предприятии и только совокупность представлений даёт полную картину, хотя, зачастую, субъективную ввиду ограниченных познавательных способностей аналитика или ошибочной интерпретации в коллективе моделей и модельных элементов.

ПРИМЕРЫ ТОЧЕК ЗРЕНИЯ ОТ OPEN GROUP

Business Process Viewpoint - эта точка зрения показывает высокоуровневую структуру и композицию одного или более бизнес процессов.

Product Viewpoint - эта точка зрения описывает ценность, которую один или более продуктов предлагают клиентам или другим внешним вовлеченным сторонам и показывает состав одного или более продуктов исходя из составляющих их сервисов (бизнес и приложения), а так же связанных контрактов или других соглашений.

Business Function Viewpoint - эта точка зрения показывает основные бизнес функции организации и их отношения исходя из потоков информации, ценности, или товаров между ними.

Application Cooperation Viewpoint - эта точка зрения описывает отношения между компонентами приложений исходя из информационных потоков между ними, или исходя из сервисов, которые они предлагают или используют.

Actor Co-operation Viewpoint - эта точка зрения фокусируется на отношениях действующих лиц друг с другом и их внешней средой.

Application Usage Viewpoint - эта точка зрения описывает как используются приложения для того, чтобы поддержать бизнес процессы, и как они используются другими приложениями.

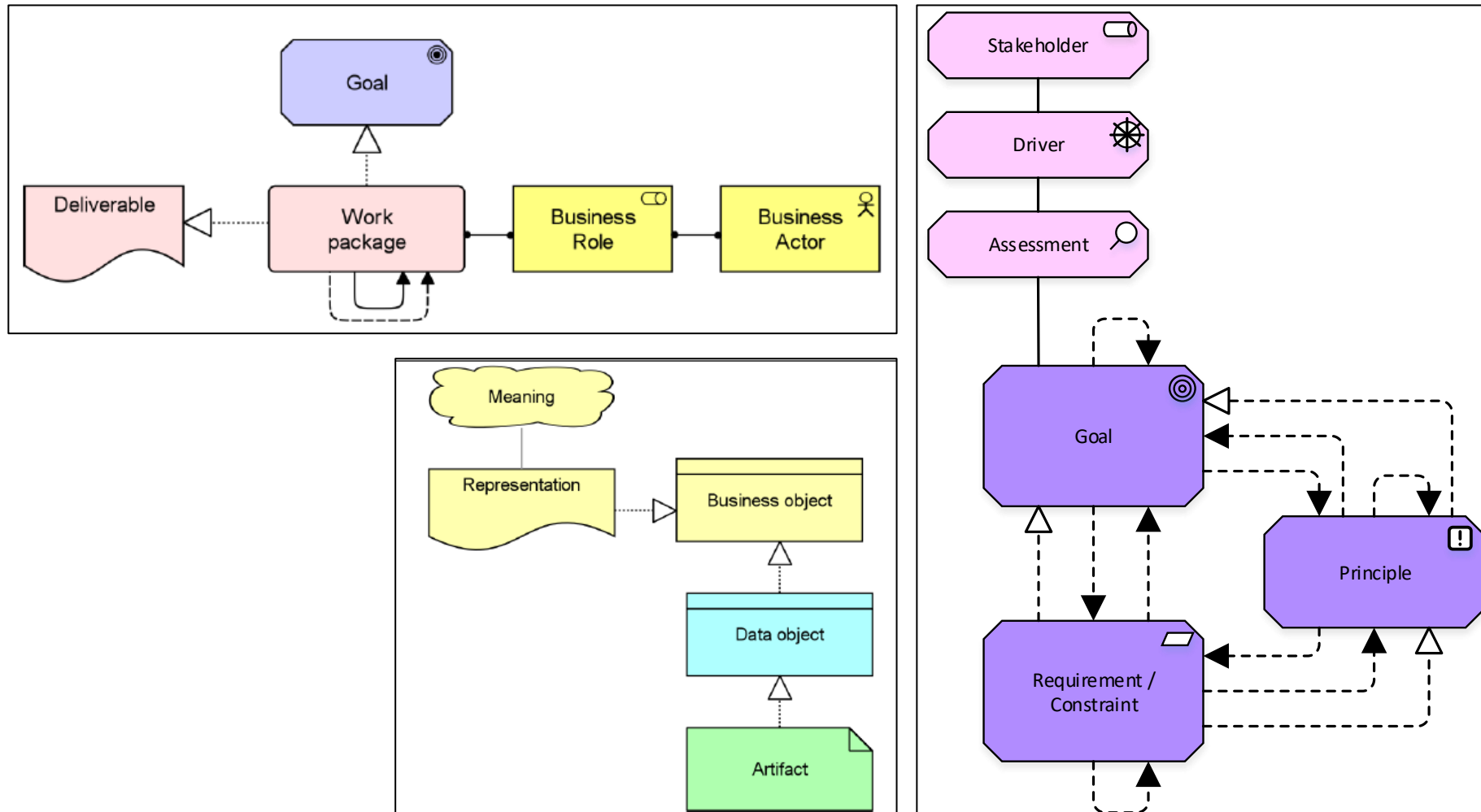
Infrastructure Viewpoint - эта точка зрения описывает инфраструктурные элементы программно-аппаратной платформы, поддерживающие Слой Приложений, такие как физические устройства, сети, или системное ПО (например, операционные системы, БД, промежуточное ПО).

Service Realization Viewpoint - эта точка зрения показывает как один или более бизнес сервисов реализуются нижележащими процессами (и иногда компонентами приложений).

Information Structure Viewpoint - эта точка зрения показывает структуру информации, используемой на предприятии или в специфичном бизнес процессе или приложении, исходя из типов данных или (объектно-ориентированной) структуре классов.

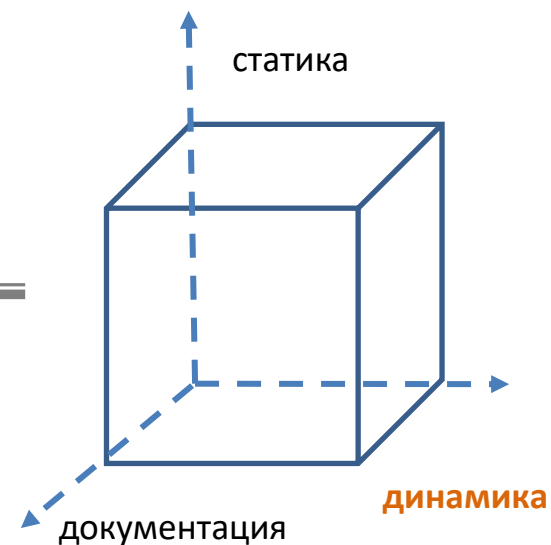
Implementation and Deployment Viewpoint - эта точка зрения показывает как одно или более приложений реализуются в инфраструктуре.

ВЫРАЖЕНИЕ ТОЧЕК ЗРЕНИЯ В НОТАЦИИ ARCHIMATE



ГРАНЬ №2.

АРХИТЕКТУРА. ДИНАМИКА*



* Динамика:

Мы вкладываем в это слово процесс перехода из текущего или базового состояния архитектуры в целевое.

Похоже, что TOGAF написан из предположения, что это и есть первичная задача архитектуры, как дисциплины.

ЗАВИСИМОСТЬ ОТ КОНТЕКСТА И ТЕКУЧЕСТЬ ПОНЯТИЯ АРХИТЕКТУРЫ

Архитектура рассматривается нами в цикле проектирования или реконструкции некого объекта, в частности – бизнеса. Содержание архитектуры меняется по мере продвижения по стадиям проектирования и создания проектируемой конструкции (конструкции бизнес-системы, например).

Стадии проектирование или реконструкции: архитектура - переходный мост от начальной идеи к физической конструкции системы



Архитектура – это стержень (компоненты, структура и функционал) проектируемой и создаваемой конструкции.

АРХИТЕКТУРА ОБЕСПЕЧИВАЕТ ПРЕЕМСТВЕННОСТЬ ОТ ЗАМЫСЛА К РЕАЛЬНОСТИ



Архитектура соединяет:

Цель + Замысел + Чертеж + Реализация

Архитектура (или архитектор?) обеспечивает преемственность: от функционального и/или эстетического назначения к возможности изделия/системы выполнить возложенные на него функции.

Архитектура – это необходимый и достаточный набор решений, реализация которых обеспечивает системе (продукту, услуге, изделию, бизнесу) полное соответствие требованиям заказчика.

Здесь слово «решение» следует понимать как контекстно-зависимое и следовательно весьма разное по своей природе:

- Решение как **намерение** «будем делать так»
- Решение как **приказ** «будем делать именно так»
- Решение как **проект** «запроектировано делать так»
- Решение как **следствие** «сделано так».

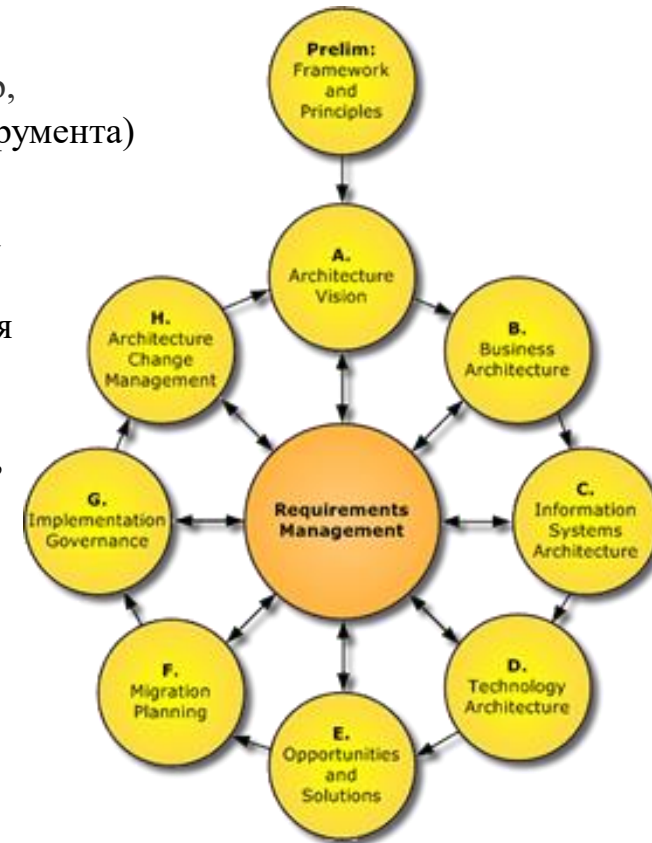
– МЕТОДОЛОГИЯ АРХИТЕКТУРНОГО МОДЕЛИРОВАНИЯ

Архитектуру, как методологию трансформации бизнеса и ИТ, развивает **Open Group** и ряд других некоммерческих институтов.

TOGAF® - The **Open Group Architecture Framework** - **стандарт** Open Group, предлагаемый **в качестве архитектурной методологии** (методического инструмента) для предприятий для целей улучшения эффективности бизнеса.

Ключевым компонентом TOGAF является Architecture Development Method – ADM (**TOGAF 9 Part II: ADM**). **ADM описывает процесс создания и развития архитектуры предприятия** и является руководством для архитекторов всех уровней/слоёв:

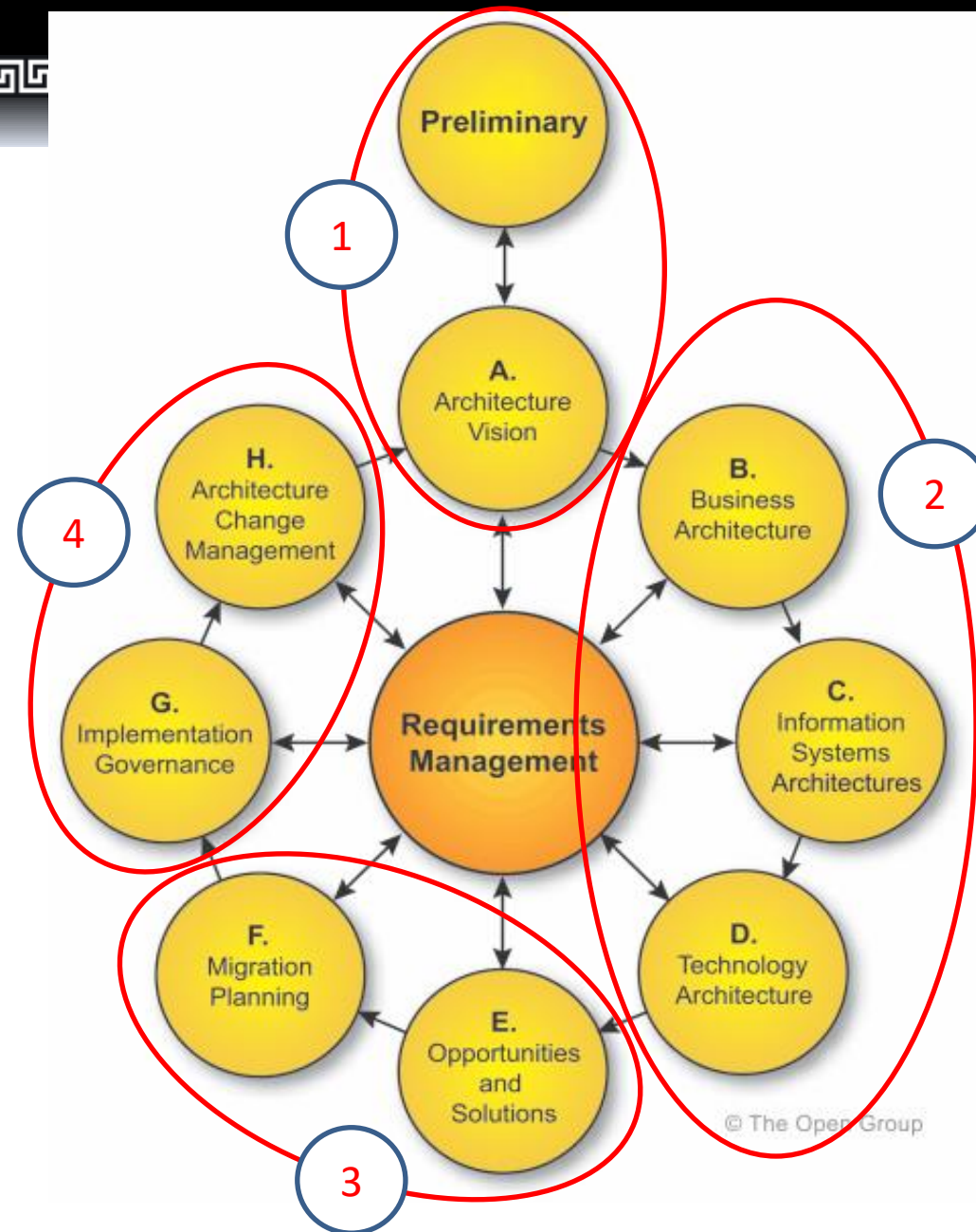
- ❖ ADM предусматривает циклический ряд фаз в разработке архитектуры, включая такие, как фаза **бизнес-архитектуры**, фаза архитектуры **информационных систем**, фаза **технологической архитектуры**.
- ❖ ADM предоставляет описание каждой архитектурной фазы (см рис справа) в виде целей фазы, подхода, входов, шагов и выходов. Входы и выходы фазы предоставляют собой определенные архитектурные артефакты.
- ❖ ADM предоставляет механизм корреляции и контроля всех фаз через центральный процесс ADM - управление требованиями.



TOGAF охватывает как сам метод ADM, так и вопросы организации архитектурных работ и контроля за процессом трансформации, а также вопросы ведения архитектурных знаний в виде репозитория.

TOGAF не столько внимания уделяет статическому аспекту архитектуры – сколько её развитию во времени, а именно: как перейти от исходной/базовой к целевой архитектуре, максимально отвечающей потребностям бизнеса? Для этого в TOGAF тщательно проработана структура (план работ) типового архитектурного проекта. Называется это ADM и состоит этот типовой план (или метод) из 10 фаз (см. рисунок справа и комментарии на следующем слайде).

1. Фазы Preliminary и A поясняют как определить **контекст архитектуры**.
2. Фазы B-C-D определяют как **спроектировать архитектуру**.
3. Фазы E-F – **поиск решения** на тему, как перейти к новой архитектуре и **планирование** перехода.
4. Фазы G-H – **внедрение архитектуры** (новой композиции компонентов бизнеса), где композиция – неразрывное сочетание компонентов и их спланированного взаимодействия.



АКЦЕНТ №5. ФАЗЫ ADM И ИХ НАЗНАЧЕНИЕ

The Preliminary Phase – описывает подготовку и инициирование мероприятий, необходимых для создания архитектурных возможностей, включая адаптацию TOGAF и определение архитектурных принципов.

Фаза А – описывает начальную фазу цикла разработки архитектуры. Включает в себя информацию об определении области охвата, выявлении заинтересованных стороны, создание Architecture Vision и получение соответствующих одобрений.

Фаза В – описывает разработку бизнес-архитектуры, поддерживающей Architecture Vision.

Фаза С – описывает разработку информационной архитектуры и/или архитектуры данных.

Фаза D – описывает разработку технологической архитектуры.

Фаза E – консолидирует результаты GAP-анализа между базовой и целевой архитектурой, полученные на фазе В-С-D, и формирует road map из блоков работ (или отдельных проектов), которые нужно выполнить для перехода к целевой архитектуре. Не исключается формирование промежуточных архитектур, как вех на пути трансформации.

Фаза F – описывает разработку детального плана реализации и миграции, который предназначен для перехода от исходной архитектуры (Baseline Architecture) к целевой.

Фаза G – архитектурный надзор за проектами, которые обеспечивают запланированный набор преобразований в компании. Цель надзора: отслеживать прогресс и этапность архитектурных изменений.

Фаза H – устанавливает процедуры управления изменениями по отношению к новой архитектуре.

Requirements Managements – управление требованиями на всех фазах (А-Н) архитектурного проекта.



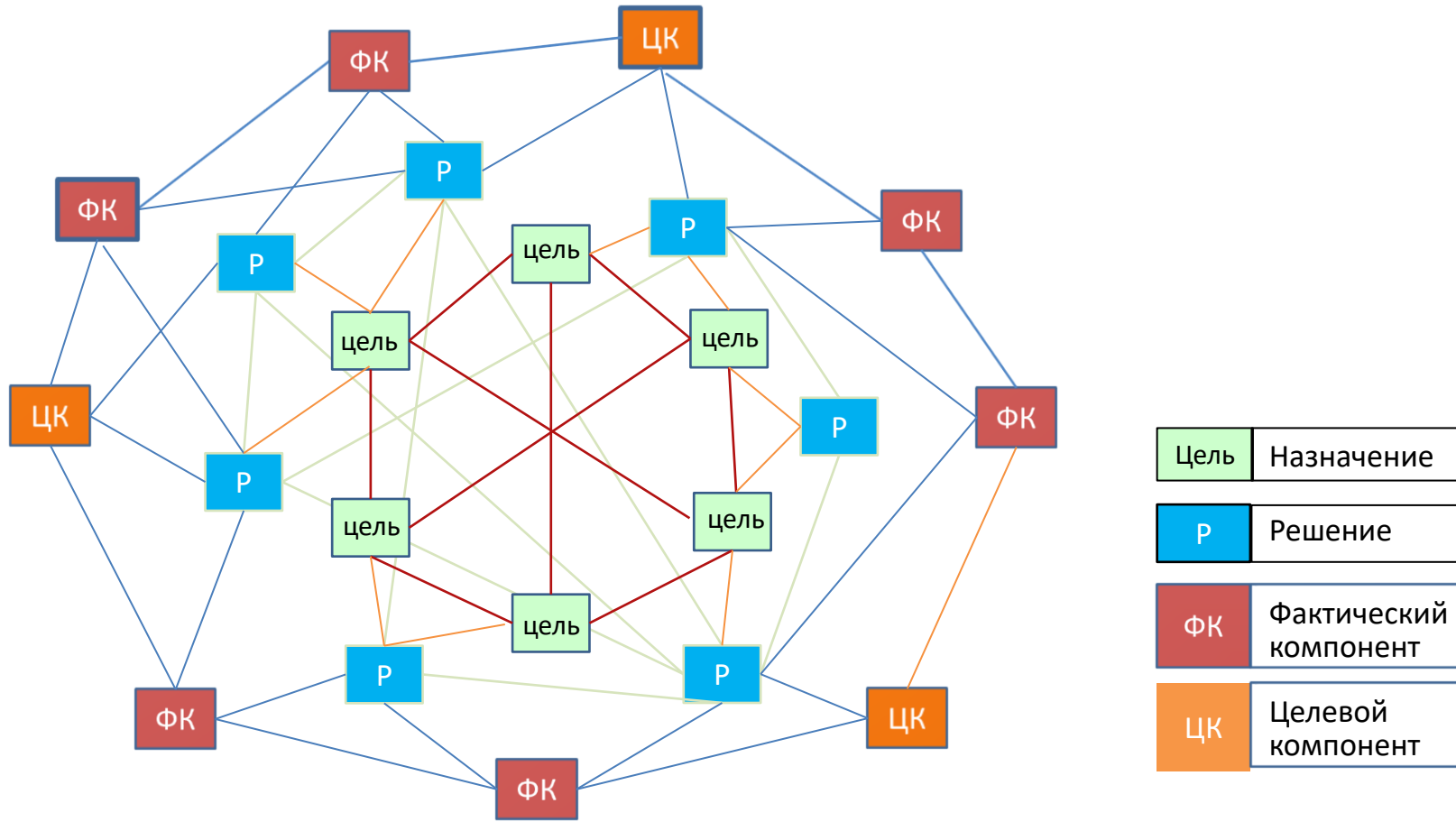
ЗНАЧЕНИЕ ДИНАМИКИ

Архитектура часто рассматривается как метод трансформации некой текущей ситуации в целевую. Было бы правильно говорить так: архитектурный метод анализа зависимостей внутри системы часто используются, а в сложных случаях являются единственным методом, позволяющим спланировать трансформацию:

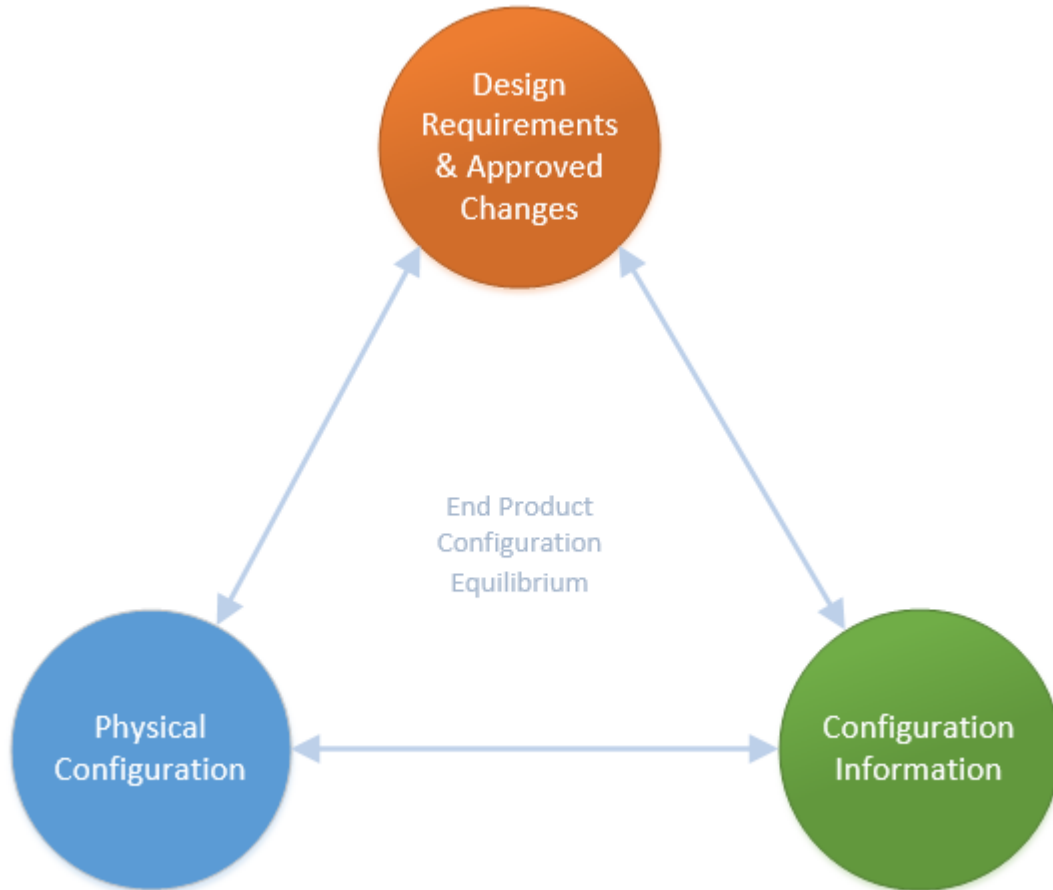
- Понять текущую «статику» (текущее состояние) системы: активные элементы, поведение элементов, пассивные элементы
- Понять будущую желаемую «статику» (будущее состояние) системы: активные элементы, поведение элементов, пассивные элементы
- Понять переход в будущее состояние: какие элементы и как необходимо изменить, включая зависимости элементов, чтобы признать наступление будущего состояния (или транзитного состояния на пути к будущему).

Примечание: в связи с тем, что изучение предприятия является затратным мероприятием, то часто требуется обоснование такого изучения. Основная причина последних лет – необходимость трансформации. Но методы трансформации нигде не описываются и не преподаются сами по себе. Поэтому и обращаются к архитектурной практике в поисках ответа на вопрос: что и как делать. Вот **поэтому архитектуру часто и ассоциируют с трансформацией.**

АРХИТЕКТУРА – ЭТО ТРИЕДИНСТВО: ЦЕЛЬ-РЕШЕНИЕ-ФАКТ



CONFIGURATION MANAGEMENT EQUILIBRIUM MODEL



Модель эквиприума
– главный принцип при дизайне АЭС

(см. подробности в документацию по проектированию АЭС от МАГАТЭ).



EA is a strategic tool that presents an approach to identify and address gaps between aspirations and reality, whatever drives the gaps. It accelerates the ability of an Enterprise to achieve its stated objectives. The tool comes with its method to use, taxonomy to support the directions, and resources needed to benefit from using the tool.

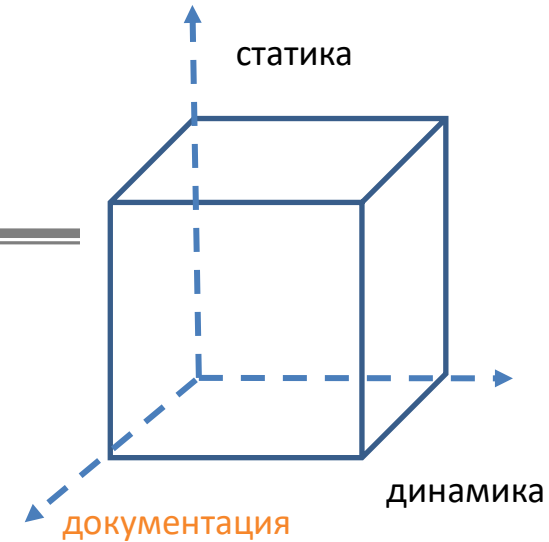
EA (если без воды) — это методический подход к выявлению и устранению разрывов между устремлениями и реальностью. Этот подход ускоряет способность предприятия достигать поставленных целей. Подход поставляется со своим методом использования, таксономией для поддержки указаний и ресурсами, необходимыми для получения выгоды от использования подхода.

Это определение наиболее близко к тому, как формулирует архитектуру Витрувий:
архитектура – это дисциплина... Важны и денотация и коннотация слова дисциплина:

- а) нужно многое знать, чтобы архитектурить;
- б) нужно архитектурить последовательно, тщательно, методично, вдумчиво, с опорой на инструменты.

ГРАНЬ №3.

АРХИТЕКТУРА. РЕПОЗИТОРИЙ ОПИСАНИЙ*



Архитектура – это схемы, плакаты, диаграммы...

* Репозиторий:

Мы вкладываем в это слово всю совокупность документации, электронной, бумажной, схемной, диаграммной, табличной, в виде моделей, специального ПО и т.п. Что есть архитектура, как не многочисленные попытки зафиксировать её с разной степенью детальности и с разных точек зрения?

ЧТО ЕСТЬ АРХИТЕКТУРА С ТОЧКИ ЗРЕНИЯ РУТИНЫ?

За пеленой высоких фраз об архитектуре, архитектура выливается в три вида основных архитектурных артефактов (= архитектурный проект):

1. **Каталоги** - инструмент учета строительных блоков архитектуры, приведенных на слайде с префиксом «Акцент №3»
2. **Зависимости** - инструмент (например, матрицы), с помощью которого связывают элементы различных каталогов, отражая тем самым отношения между ними из реальности.
3. **Диаграммы** – инструмент иллюстрирования отдельных областей предприятия с использованием данных из каталогов и их зависимости.

А есть еще и специфическая проектная документация по архитектуре, как то: architecture score, architecture vision, стратегические принципы и требования, метрики, стандарты, референсные модели, architecture roadmap, оценки (assessment) и пр.

АКЦЕНТ №6. КАТАЛОГИ

Каталоги - это списки/реестры/коллекции строительных блоков одного типа, причем каждый строительный блок в каталоге атрибутирован его отличительными свойствами. Можно сказать, что на каждый строительный блок есть свой отдельный паспорт, а значит каталог – это коллекция паспортов на элементы одного типа..

Название ИС	AC ID	Symbol	Server location	InfoF2ACsReq
Учет ИТ-активов (Assets)	СИБ.Оф.001	МО137765173	ЦОД РФ Сибирь.Оф	
ИС "Контроль табельного учета"	СИБ.Тф.006	МО137911432	ЦОД РФ Сибирь.Тф	
Система приема оплат от Агентов	СИБ.Иф.010	МО137926571	ЦОД РФ Сибирь.Иф	
формирование файла для загрузки из IUM (table nsk.x_jum_unf)	СИБ.Нф.035	МО137941240	СК РФ Сибирь.НФ схема ВЛЛНСК (НФ)	
АРМ Кросс-МРД	СИБ.Аф.010	МО137943231	ЦОД МРФ Сибирь	
Модуль автоматической обработки заданий МРД на ЭАТС филиала	СИБ.Иф.002	МО138015068	ЦОД РФ Сибирь.Иф	
Акционатор (менеджмент срока акционных услуг и перепакетирование nsk.pkg_sa)	СИБ.Нф.036	МО138113324	СК РФ Сибирь.НФ схема ВЛЛНСК (НФ)	
CRM / Бизнес центры - Учет ЮЛ	СИБ.Нф.024	МО138215322		
IS.CRM / Бизнес центры		МО1131020298		
Ядро CRM / Бизнес центры		МО1137234218		
Шлюз МРД -> Светец (АГО):данные для обзвона	СИБ.Оф.002	МО138222089	ЦОД РФ Сибирь.Оф	
HP Service-Activator (СИБ)	СИБ.012	МО138308836	ЦОД МРФ Сибирь	
Управление сервисом на базе HP OV SA		МО1991332333		
Прием и выполнение команд на управление учетными записями пользователя в LDAP каталоге (СИБ.012.Оф.011)	СИБ.012.Оф.011	МО800158174		
Определение результата выполнения операции по наряду из СУУ в LDAP каталоге (СИБ.012.Вф.022)	СИБ.012.Вф.022	МО136563643		
Создание профиля в LDAP каталоге в автоматическом режиме (СИБ.012.Вф.005)	СИБ.012.Вф.005	МО1694781198		
Откат результата действий, выполненных по наряду на предоставление доступа к услуге (СИБ.012.Вф.027)	СИБ.012.Вф.027	МО270472814		

Каталоги и зависимости являются первичным артефактом архитектуры. Диаграммы – вторичны. Если архитектор сделал лишь 1-2 диаграммы, то это пока лишь иллюстрация идеи, но не сама архитектура.

АКЦЕНТ №7. ЗАВИСИМОСТИ ИЛИ СВЯЗИ

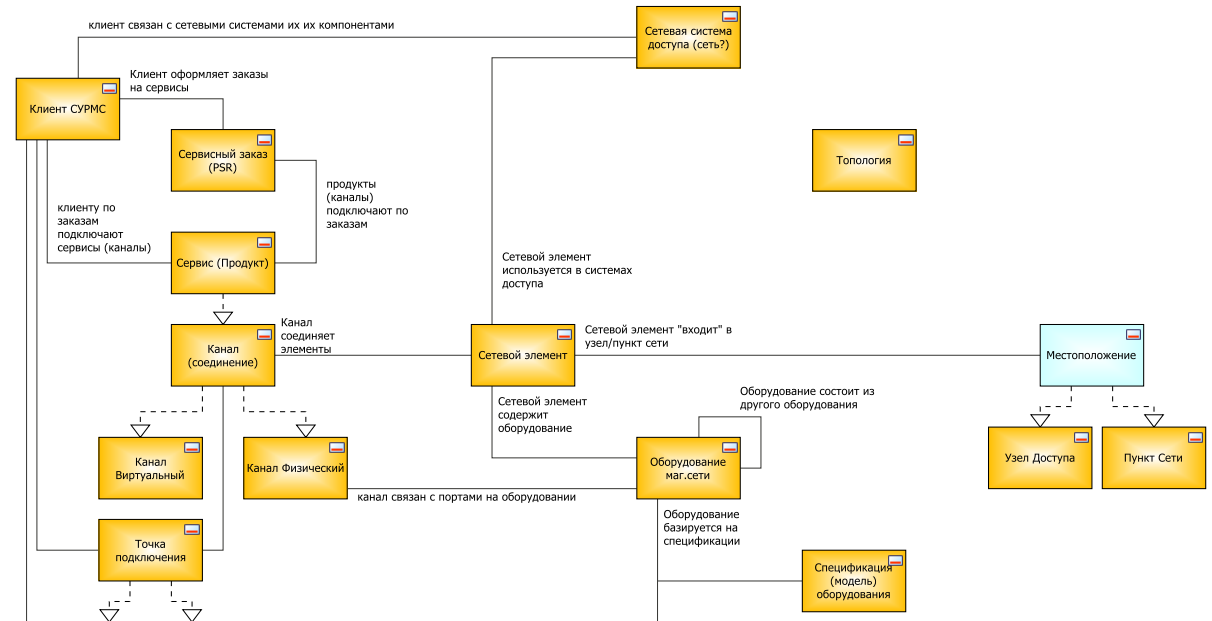
Зависимости – это таблицы или матрицы, отражающие взаимосвязи между строительными блоками в различных каталогах. Зависимости репрезентируют структуру системы в арх.модели.

Связь атрибутов с объектами данных								
Name	Symbol	AbstractBusinessObject	AbstractEntity	AbstractEntityApp	AbstractEntitySimple	AccountInfo (Wargaming)	ActivateWgServiceRequest	ActivateWgSe
associationId	AbstractBusinessObject_associationId	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
attributeEndDate	AbstractBusinessObject_attributeEndDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
attributeStartDate	AbstractBusinessObject_attributeStartDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
currentVersion	AbstractBusinessObject_currentVersion	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
deleted	AbstractBusinessObject_deleted	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
endDate	AbstractBusinessObject_endDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
incomingInitialMessage	AbstractBusinessObject_incomingInitialMessage	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
lastSnapshot	AbstractBusinessObject_lastSnapshot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
localSystemDateSnapshot	AbstractBusinessObject_localSystemDateSnapshot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
masterObject	AbstractBusinessObject_masterObject	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
startDate	AbstractBusinessObject_startDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
version	AbstractBusinessObject_version	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
app	AbstractEntityApp_app	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
localSystemId	AbstractEntityApp_localSystemId	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
owner	AbstractEntityApp_owner	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
pkID	AbstractEntitySimple_pkID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
applicationName	AbstractEntity_applicationName	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

АКЦЕНТ №8. ДИАГРАММЫ

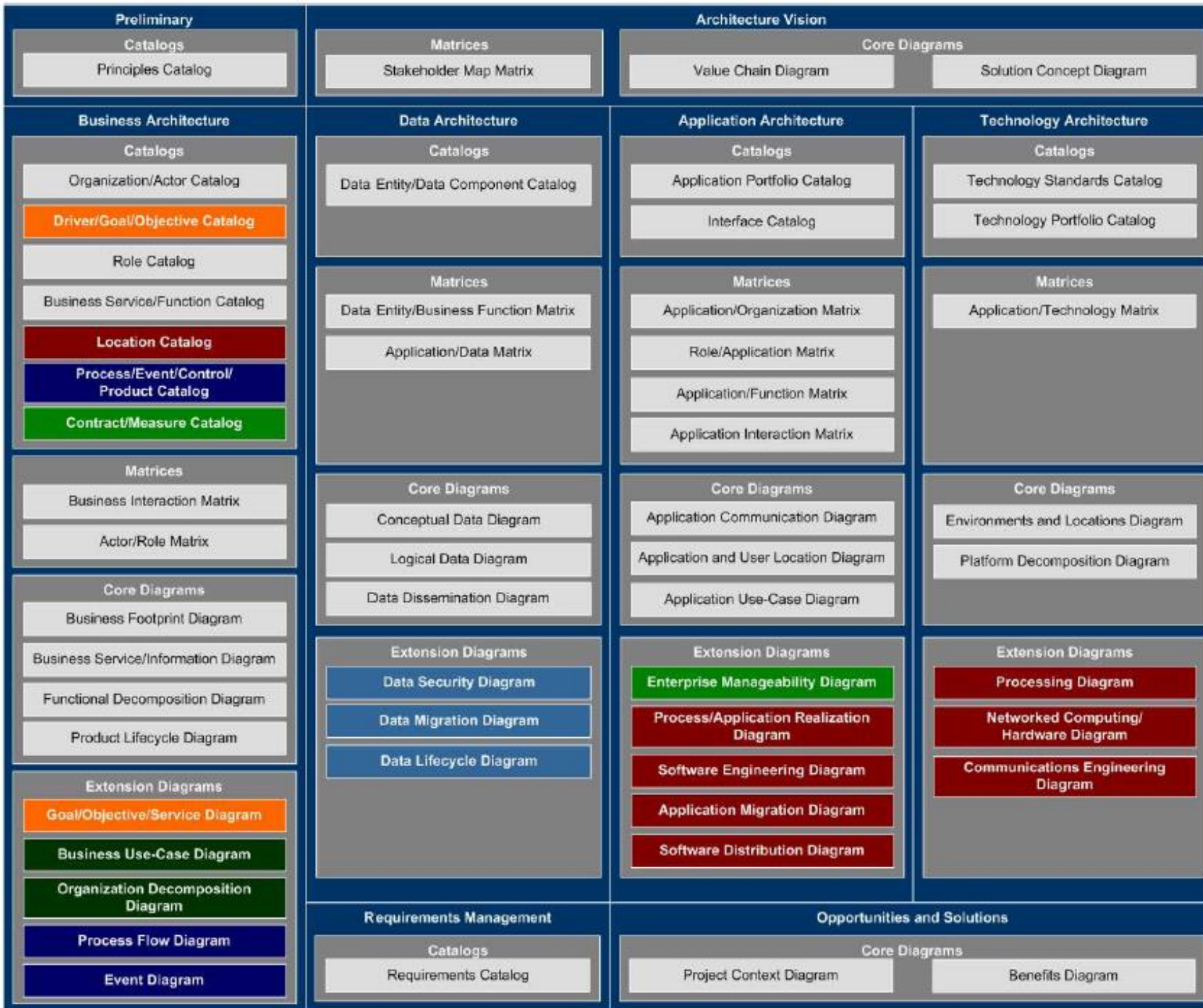
Диаграммы – это представления архитектурного контента в графическом формате, позволяющие ключевым участникам получать требующуюся информацию, так как классические представление данных в табличном виде не позволяют быстро вникнуть в моделируемое явление. Диаграмма отображает не только каталогизированные элементы конструкции, но и связи (зависимости).

Диаграммы также могут использоваться для визуальной проверки полноты и связности собранной информации.



Archimate определяет множество архитектурных диаграмм, которые должны быть созданы на каждой фазе ADM.

Виды диаграмм, каталогов и матриц в TOGAF





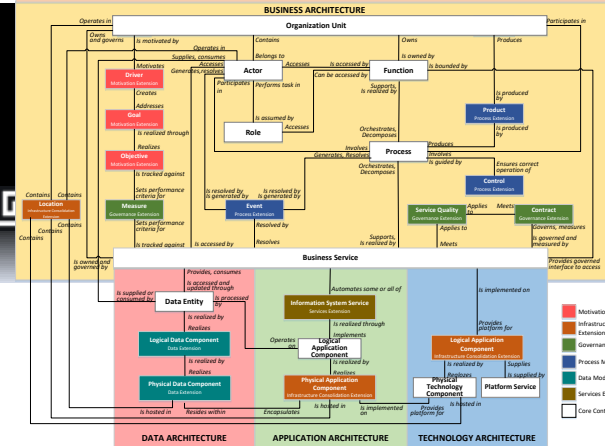
НЕ ПРОСТО СТРОИТЕЛЬНЫЕ БЛОКИ – БАЗА ДАННЫХ!

КАЖДЫЙ ЭЛЕМЕНТ В КАТАЛОГЕ АТРИБУТИРУЕТСЯ ДЛЯ ПОЛНОЦЕННОГО ОПИСАНИЯ ЭЛЕМЕНТА.
В ИТОГЕ КАТАЛОГ ЭЛЕМЕНТОВ – ЭТО В СУЩНОСТИ ИНВЕНТАРНАЯ БАЗА ЭТИХ ЭЛЕМЕНТОВ.
ПОЭТОМУ МОЖНО ГОВОРИТЬ О ПОЛНОЦЕННОЙ:

- ❖ ИНВЕНТАРИЗАЦИИ ЦЕЛЕЙ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ПРОЦЕССОВ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ СИСТЕМ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ДАННЫХ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ФУНКЦИЙ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ КАНАЛОВ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ПРОДУКТОВ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ДРАЙВЕРОВ ПРЕДПРИЯТИЯ
- ❖ ИНВЕНТАРИЗАЦИИ ТРЕБОВАНИЙ ПРЕДПРИЯТИЯ
- ❖ И Т.Д.

Атрибутированный элемент – это информационный или цифровой двойник. На стадии проектирования – это макет будущего элемента. На стадии готового изделия/системы – это цифровой паспорт готового изделия/системы.

По этой причине архитектуры часто именуют цифровыми, потому что они являются собой цифровые модели!



Итак, рассматривая тему архитектуры согласно TOGAF, нужно отслеживать в каждый момент времени, о чем именно идет речь при рассмотрении архитектуры:

- о ее **статическом** аспекте, то есть о системной декомпозиции или о том, что делает систему системой. Архитектура – это функция/метод декомпозиции;
- о ее **динамическом** аспекте, то есть как наша система будет меняться со временем;
- о том, как, где и с помощью чего архитектура **описана**/задокументирована.

Полное описание архитектуры (или полная архитектура) – это ее описание с использованием всех предложенных в TOGAF строительных блоков, включая выделение, перечисление, нормализацию, атрибутирование и связывание всех этих блоков, а также связи этих внутренних блоков/элементов системы с контекстом, в котором система работает.

Архитектурный подход и методология - самый мощный инструмент анализа и дизайна масштаба предприятия на сегодняшний день. Основные преимущества архитектурного подхода для предприятий:

- ❖ управление и трансформация, основанные на знаниях, а не интуиции
- ❖ управление изменениями с учетом всего спектра взаимосвязей и взаимовлияний на предприятии
- ❖ учет мнения и позиции ключевых заинтересованных лиц предприятия
- ❖ учет контекста, в котором предприятие работает, включая изменения контекста
- ❖ поддержка масштабных и синхронных изменений во всех слоях предприятия
- ❖ единые язык, подход, методология, позволяющие создавать большие архитектурные команды единомышленников, вовлекая в изменения всё предприятие.

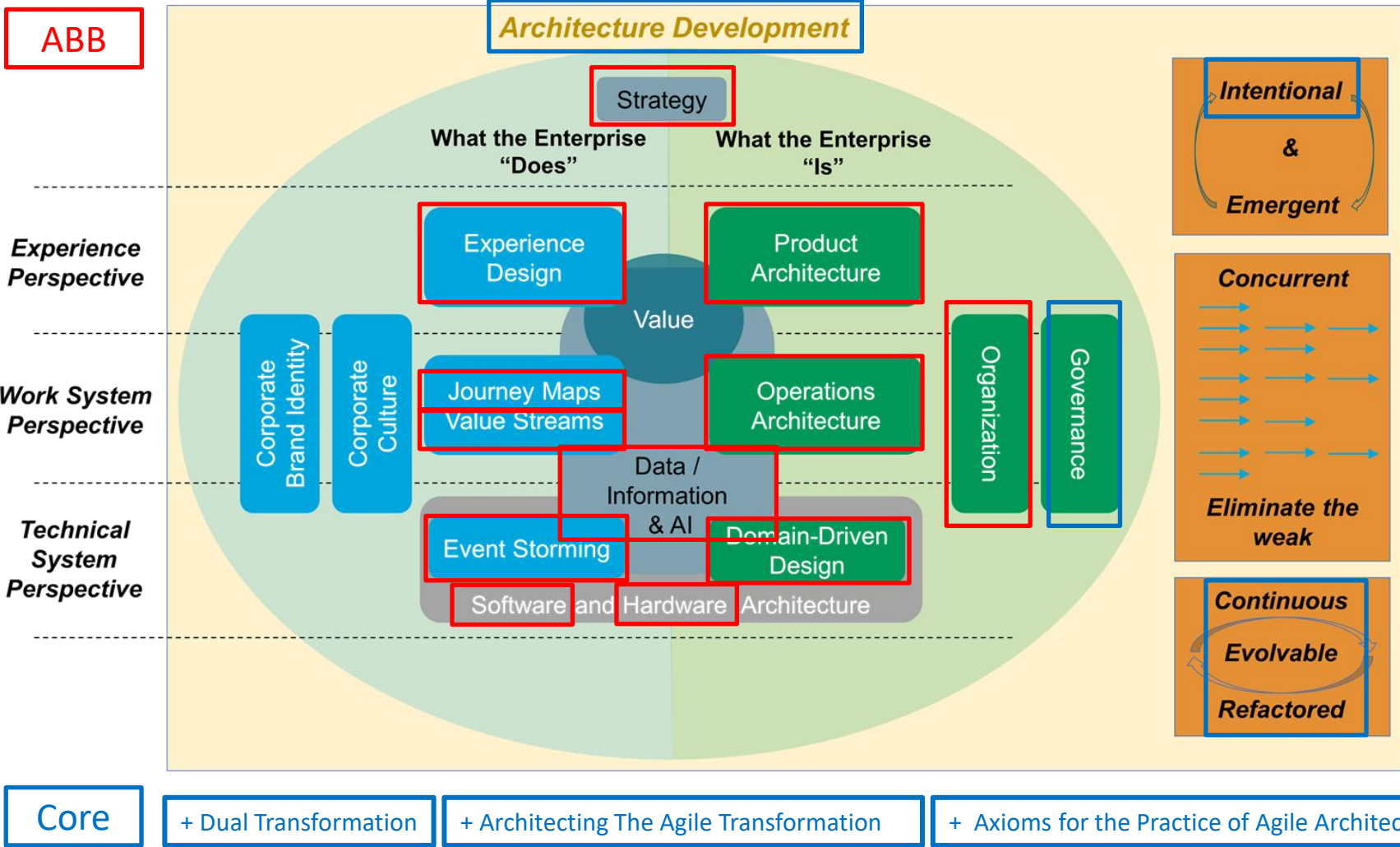
4.

AGILE АРХИТЕКТУРА.

Как повысить гибкость архитектурных решений.

По материалам Open Group. Раздел в разработке.

AGILE ARCHITECTURE: КУДА И КАК ИДЕМ?



The **Experience** perspective defines value from a client perspective. It analyzes a client's job-to-be done; their pain points and gains. It also covers the emotional dimension of the experience, starting from an outside-in view that places customer needs and problems at the center. Design thinking and market research are incorporated into the broader Agile architecting discipline.

The **Work System** perspective defines the ability of the enterprise to deliver client benefits efficiently. It starts with analyzing commonalities across value streams; this helps to identify activities that could be shared, thus contributing to architecting the operating model.

The **Technical System** perspective covers the software and hardware parts of the enterprise. It starts from analyzing domains such as payments, mortgage lending, or federated identity management and also includes architecting the physical world. The software architecture discipline is part of the broader Agile architecting discipline.

ОСНОВНЫЕ ВЫЗОВЫ, КОТОРЫЕ ДОЛЖЕН АДРЕСОВАТЬ И УЛАДИТЬ СОВРЕМЕННЫЙ АРХИТЕКТОР



Ответы на эти вопросы порождают архитектурные принципы, решения и элементы будущей системы

Ключевые аспекты процесса архитектурирования

Технологические решения и методологии.

AAF.E-01
Loosely-Coupled Systems & Organizations

- How to architect loosely-coupled systems?
- How to architect modular organizations composed of autonomous teams?
- How to design organizations that produce modular architectures?
- How to refactor highly-coupled and monolithic systems?

AAF.E-02
Business Architecture Patterns

- How to innovate business and operating models?
- Which business strategy concepts can help align the enterprise: vision, mission, purpose, ...?
- How to decompose the business into modular operating units?
- How to deploy the strategy in a non-command-and-control way?

AAF.E-03
Aligned Organizations & Systems

- How to preserve local autonomy while enforcing global alignment?
- Which organizational model and culture changes are required?
- Which governance model will keep organizations and systems aligned while preserving autonomy?
- How to enable services interoperability and composability?

AAF.E-04
Software Architecture Patterns

- How to architect highly distributed software systems that are:
 - Responsive to user requests
 - React to variable load conditions
 - Remain available?
- How to leverage big and fast data architecture patterns?
- What is the impact of AI/ML on system architecture?

AAF.E-05
Minimum Viable Architecture

- How much architecture work should be done up-front for the next agile iteration?
- How should architecture decisions be made and validated?
- How should MVA influence/impact agile teams?
- How to align MVA with MVP?

AAF.E-06
Evolvable Architecture

- Which architecture practices and patterns will facilitate future change?
- How to anticipate change and avoid unnecessary complexity?
- How to prevent the architecture from gradually degrading over time?

AAF.E-07
Maturity Model

- How many maturity levels?
- How to define maturity levels?
- How to assess the enterprise's maturity level?
- What are the pre-conditions of a successful move to the next maturity level?
- What are the key success factors?

AAF.E-08
Architect's Role & Responsibilities

- What is the architect's role as a squad member?
- Should the architect become an "über" developer?
- What is the architect's role as guardian and defender of the overall system's coherence?

AAF.E-09
The Agile Architect's Competencies & Skills

- What core set of competencies and skills should all architects possess?
- Which soft skills are needed to lead and facilitate team collaboration?
- Toward a "T-shaped" full-stack profile that includes software development skills?

AAF.E-10
Domain-Driven Design & Event-Driven Architecture

- How to identify contexts and aggregates using event storming?
- How to draw context maps using the DDD strategy patterns?
- How to protect application code from future technical debt?

AAF.E-11
Data & Information Modeling

- How do we evolve data/information modeling techniques to cater for big and fast data technology?
- How to handle data when using the μ -services architecture style?
- What are the impacts of real-time streaming analytics on system architecture?

AAF.E-12
Complex Systems Modeling

- How to model and steer the evolution of complex adaptive systems?
- How to use a Design Structure Matrix (DSM) to reveal both hierarchical ordering and cyclic groups within a complex technical system?

УПРАВЛЕНИЕ ЧЕРЕЗ АРХ.ПРИНЦИПЫ ИЛИ ПРИНЦИПЫ

Ключевые принципы Безоса.

- 1. All teams will henceforth expose their data and functionality through service interfaces.*
- 2. Teams must communicate with each other through these interfaces.*
- 3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.*
- 4. It doesn't matter what technology they use. HTTP, CORBA, Pub/Sub, custom protocols – doesn't matter. Bezos doesn't care.*
- 5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.*

Не отказывайся от помощи, особенно когда это связано с исполнением долга.

Многое из того, что не удаётся сделать в одиночку, может быть легко достигнуто, если действовать сообща...

Марк Аврелий

MARCUS AURELIUS LTD.

КОНТАКТЫ ДЛЯ СВЯЗИ

Рудь Виктор
Директор по консалтингу
ООО «МАРК АВРЕЛИЙ»

<http://www.consulo.ru>

E-mail: **v.rud @consulo.ru**

Телефон: **+7 (985) 922-12-40**

АВТОРСКОЕ ПОНИМАНИЕ АРХИТЕКТУРЫ

Виктор Рудь: рассматривая сложную систему, каковой является организация/компания/бизнес, мы задаёмся вопросом, что делает систему системой? Что делает компанию устойчивой и стабильной несмотря на окружающий ее хаос или неопределенности внешней среды?

Любой зрелый руководитель может объяснить секрет своего успеха и этот секрет лежит всегда в области оригинального эмпирически найденного баланса и связей между явлениями разной природы, как например:

- Наша компания производит уникальный продукт или портфель продуктов
- Наша компания находится в выгодном для покупателей месте
- Мы занимаем монопольное или доминирующее положение в сегменте, так сложилось исторически
- В нашей компании экономически эффективный и отлаженный в пользу клиента логистический процесс
- Наша компания имеет связи и поддержку со стороны местных властей, мы добились этого за последние 3 года
- Коллектив компании – это члены одной семьи, болеющие за общее дело, как за своё личное
- Наш производственная технология даёт наименьшее количество дефектов на единицу продукции за счет разделения на относительно автономные отдельно контролируемые участки
- Мы добились крайне низких издержек за счет переноса трудоемких частей производства в Китай
- У нас отлаженная система автоматизации, дающая точные ответы на вопросы относительно всех аспектов бизнеса и мы вовремя реагируем на все колебания спроса
- Мы способны ежегодно выводить на рынок 3 новых продукта и снимать с продажи от 2 до 3 устаревших
- Наш WEB-сайт позволяет отслеживать интерес и опыт наших клиентов в пользовании нашими продуктами
- У нас отлаженная и оптимальная организационная структура
- и т.д.

Архитектура – это инженерный способ структурно, системно и математически объяснить как всё вышеперечисленное устроено, живет, развивается и влияет друг на друга под воздействием внешних и внутренних факторов.

ЭКСПЕРТИЗА И УСЛУГИ КОМПАНИИ «МАРК АВРЕЛИЙ»

Виды услуг компании:

- Управление системной архитектурой
- Управление процессной архитектурой
- Управление архитектурой данных
- Управление архитектурой предприятия
- Концептуальное проектирование
- Реинжиниринг процессов
- Дизайн информационных систем
- Бизнес-анализ
- Дизайн и поддержание автоматизированных каталогов услуг
- Мета-моделирование
- Обучение по архитектурным методологиям
- Управление требованиями
- Разработка Требований, Тех.Заданий, Архитектурных решений и концепций

Виды помощи в больших проектах:

- Управление проектом
- Планирование проекта и ресурсов
- Создание и контроль процессной архитектуры
- Создание и контроль функциональной архитектуры
- Создание и контроль информационной архитектуры
- Разработка учебных материалов
- Разработка методологии моделирования
- Обучение ключевых пользователей
- Реинжиниринг процессов
- Ведение и поддержание архитектурного репозитория

С 2023 года основным направлением деятельности компании является разработка и развитие программного продукта для создания архитектурных моделей – СиММА (Система Многослойного Моделирования Архитектуры).