

Концепция Enterprise Architecture. Методики разработки. Методология TOGAF. Методика Захмана. Сравнение TOGAF и TMF

Идея



Подходы

Методики описания [задают классификацию основных областей архитектуры и единые принципы для их описания во взаимной увязке друг с другом]

Изобретаются:

- Консультантами (Gartner, Open Group, NASCIO и др.)
- ГосСтруктурами (Казначейство США, МО США и др.)
- Вендорами (Microsoft, IBM, SAP и др.)
- Умными людьми (Захман, Зиндер)

Существуют различные индустриальные стандарты – от IEEE, ISO и прочие, но ни один из стандартов не занимает доминирующего положения.

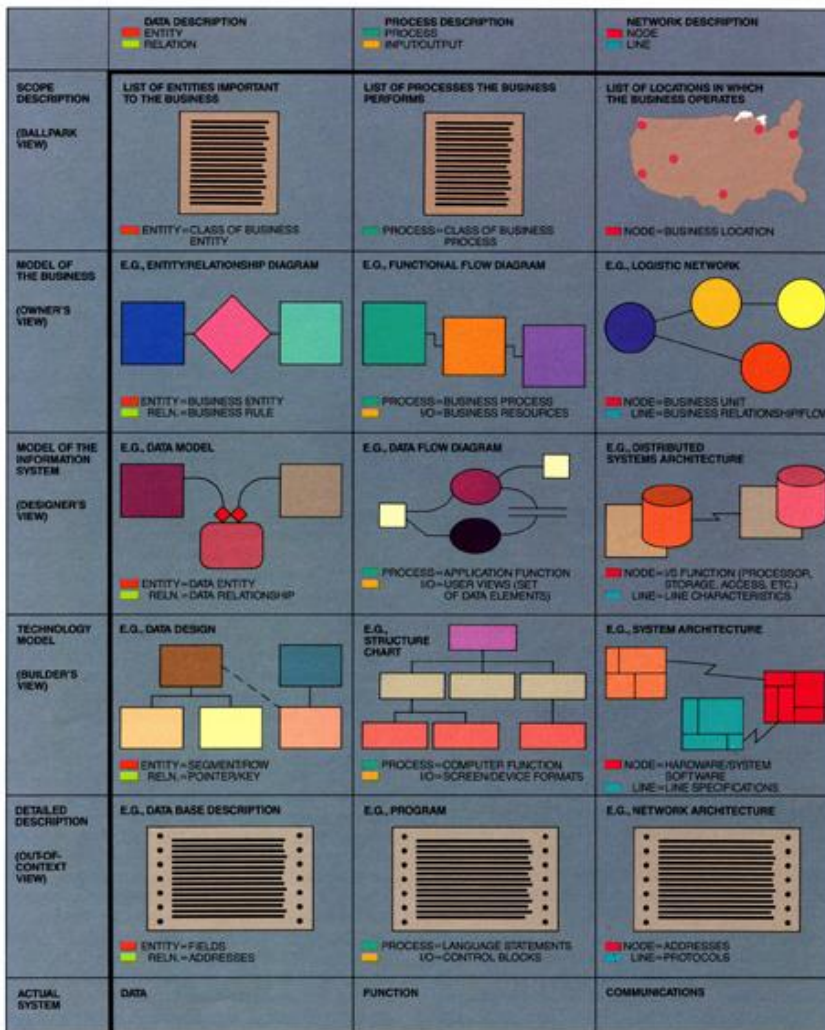
Архитектура – это одновременно и описание, и процесс.

Методика Захмана

Первая версия – 1987 год.

Изначально относилась к **архитектуре информационных систем**. Содержала измерения «данные», «функции» и «сеть»

Figure 2 Framework for information systems architecture



Расширение – 1992 год

Добавились измерения «люди», «время» и «мотивация» - **архитектура предприятия**

Figure 6 Six-column ISA framework

© 1992 John A. Zachman, All Rights Reserved

	DATA ENT RELN	FUNCTION ARG	NETWORK NODE LINK	PEOPLE AGENT WORK	TIME NODE CYCLE	MOTIVATION ENDS MEANS	
SCOPE PLANNER	LIST OF THINGS IMPORTANT TO THE BUSINESS ENT - CLASS OF BUSINESS THING	LIST OF PROCESSES THE BUSINESS PERFORMS FUNCTION - CLASS OF BUSINESS PROCESS	LIST OF LOCATIONS IN WHICH THE BUSINESS OPERATES NODE - MAJOR BUSINESS LOCATION	LIST OF ORGANIZATIONS/AGENTS IMPORTANT TO THE BUSINESS AGENT - MAJOR ORGANIZATION UNIT	LIST OF EVENTS SIGNIFICANT TO THE BUSINESS TIME - MAJOR BUSINESS EVENT	LIST OF BUSINESS GOALS/STRATEGY ENDS/MEANS - MAJOR BUS. GOAL/CRITICAL SUCCESS FACTOR	SCOPE PLANNER
ENTERPRISE MODEL OWNER	E.G. "ENT REL DIAGRAM" ENT - BUSINESS ENTITY RELN - BUSINESS CONSTRAINT	E.G. "PROCESS FLOW DIAGRAM" FUNCTION - BUSINESS PROCESS ARG - BUSINESS RESOURCES	E.G. LOGISTICS NETWORK NODE - BUSINESS LOCATION LINK - BUSINESS LINKAGE	E.G. ORGANIZATION CHART AGENT - ORGANIZATION UNIT WORK - WORK PRODUCT	E.G. MASTER SCHEDULE TIME - BUSINESS EVENT CYCLE - BUSINESS CYCLE	E.G. BUSINESS PLAN ENDS - BUSINESS OBJECTIVE MEANS - BUSINESS STRATEGY	ENTERPRISE MODEL OWNER
SYSTEM MODEL DESIGNER	E.G. "DATA MODEL" ENT - DATA ENTITY RELN - DATA RELATIONSHIP	E.G. "DATA FLOW DIAGRAM" FUNCTION - APPLICATION FUNCTION ARG - USER VIEW	E.G. DISTRIBUTED SYSTEM ARCHITECTURE NODE - I/O FUNCTION (PROCESSOR, STORAGE, ETC.) LINK - LINE CHARACTERISTICS	E.G. HUMAN INTERFACE ARCHITECTURE AGENT - ROLE WORK - DELIVERABLE	E.G. PROCESSING STRUCTURE TIME - SYSTEM EVENT CYCLE - PROCESSING CYCLE	E.G. KNOWLEDGE ARCHITECTURE ENDS - CRITERION MEANS - OPTION	SYSTEM MODEL DESIGNER
TECHNOLOGY MODEL BUILDER	E.G. DATA DESIGN ENT - SEGMENT/ROW RELN - POINTER/KEY	E.G. "STRUCTURE CHART" FUNCTION - COMPUTER FUNCTION ARG - SCREEN/DEVICE FORMAT	E.G. SYSTEM ARCHITECTURE NODE - HARDWARE/SYSTEM SOFTWARE LINK - LINE SPECIFICATIONS	E.G. HUMAN/TECHNOLOGY INTERFACE AGENT - USER WORK - JOB	E.G. CONTROL STRUCTURE TIME - EXECUTE CYCLE - COMPONENT CYCLE	E.G. KNOWLEDGE DESIGN ENDS - CONDITION MEANS - ACTION	TECHNOLOGY MODEL BUILDER
COMPONENTS SUB-CONTRACTOR	E.G. DATA DEFINITION DESCRIPTION ENT - FIELD RELN - ADDRESS	E.G. "PROGRAM" FUNCTION - LANGUAGE STATEMENT ARG - CONTROL BLOCK	E.G. NETWORK ARCHITECTURE NODE - ADDRESS LINK - PROTOCOL	E.G. SECURITY ARCHITECTURE AGENT - IDENTITY WORK - "TRANSACTION"	E.G. TIMING DEFINITION TIME - INTERRUPT CYCLE - MACHINE CYCLE	E.G. KNOWLEDGE DEFINITION ENDS - SUBCONDITION MEANS - STEP	COMPONENTS SUB-CONTRACTOR
FUNCTIONING SYSTEM	E.G. DATA	E.G. FUNCTION	E.G. NETWORK	E.G. ORGANIZATION	E.G. SCHEDULE	E.G. STRATEGY	FUNCTIONING SYSTEM

Подход применили такие организации, как GM и Bank of America

Подход наследовали и расширили такие организации как FEAF, ToGAF, DoDAF

		ЧТО (данные)	КАК (функции)	ГДЕ (сеть)	КТО (люди)	КОГДА (время)	ПОЧЕМУ (мотив)	
Бизнес	Планировщик	Словарь	Список БП (основных)	Расположение	Ключевые группы	Ключевые события	Цели и стратегия	Контекст
	Владелец, менеджер	Концепция	Модель БП	Схема логистики	Workflow	Мастер-план	Бизнес-план	Модель предприят.
ИТ (менеджеры, разработчики)	Архитектор	Логика	Архитектура софта	Распред. Архитектура	Архитектура интерфейса	Структура процессов	Роли и правила	Модель системы
	Проектировщик	«Физика»	Системный проект	Технологич Архитектура	Архитектура Представл.	Структура управления	Описания правил	Физическая модель
	Разработчик	Структуры данных	Код	Сетевая Архитектура	Архитектура Безопасн.	Временные привязки	Реализация логики	Деталь реализации
		Данные	Софт	Сеть	Реальные люди	Бизнес-события	Работающая стратегия	Работающ. Предприят.
		Данные	Функции, Процессы	Сеть, Расположение	Люди, Организации	Время, Расписание	Мотивация	

Каждая клетка – независима от других

Последовательность колонок – не важна

Внутри клетки допускается **произвольное описание**

Все клетки в одной строке = полное описание

Модели для каждой колонки - **уникальны**

Попытка пропуска строки – **расстроит Захмана**

Отсутствует контроль изменений

Отсутствует отражение динамики

[Что предложил ToGAF](#)

Предложена The Open Group (НКО, объединяющее производителей ИТ)

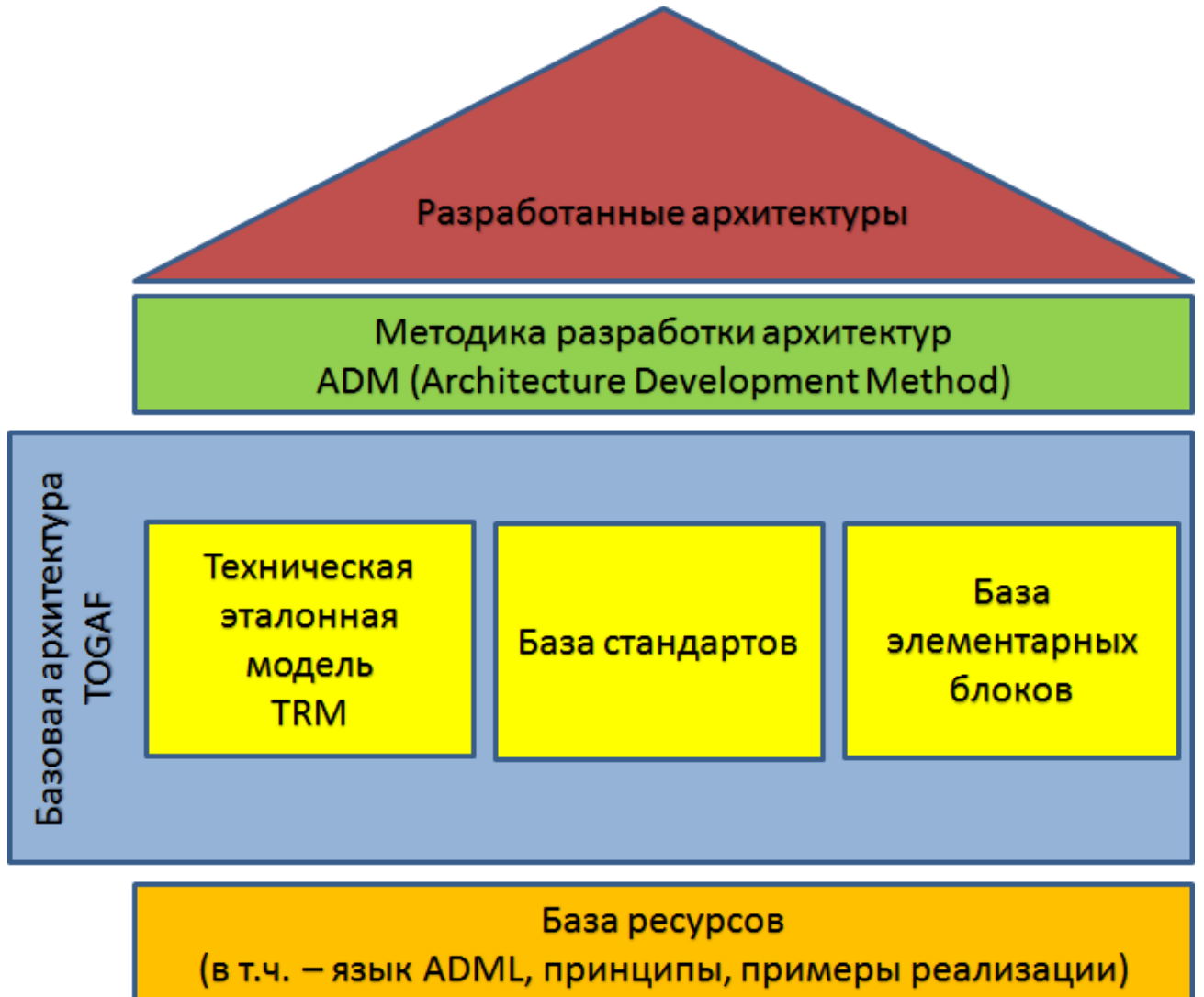
Методика TOGAF – средство для разработки архитектур информационных систем

Распространяется свободно и бесплатно (для применения на внутренних проектах)

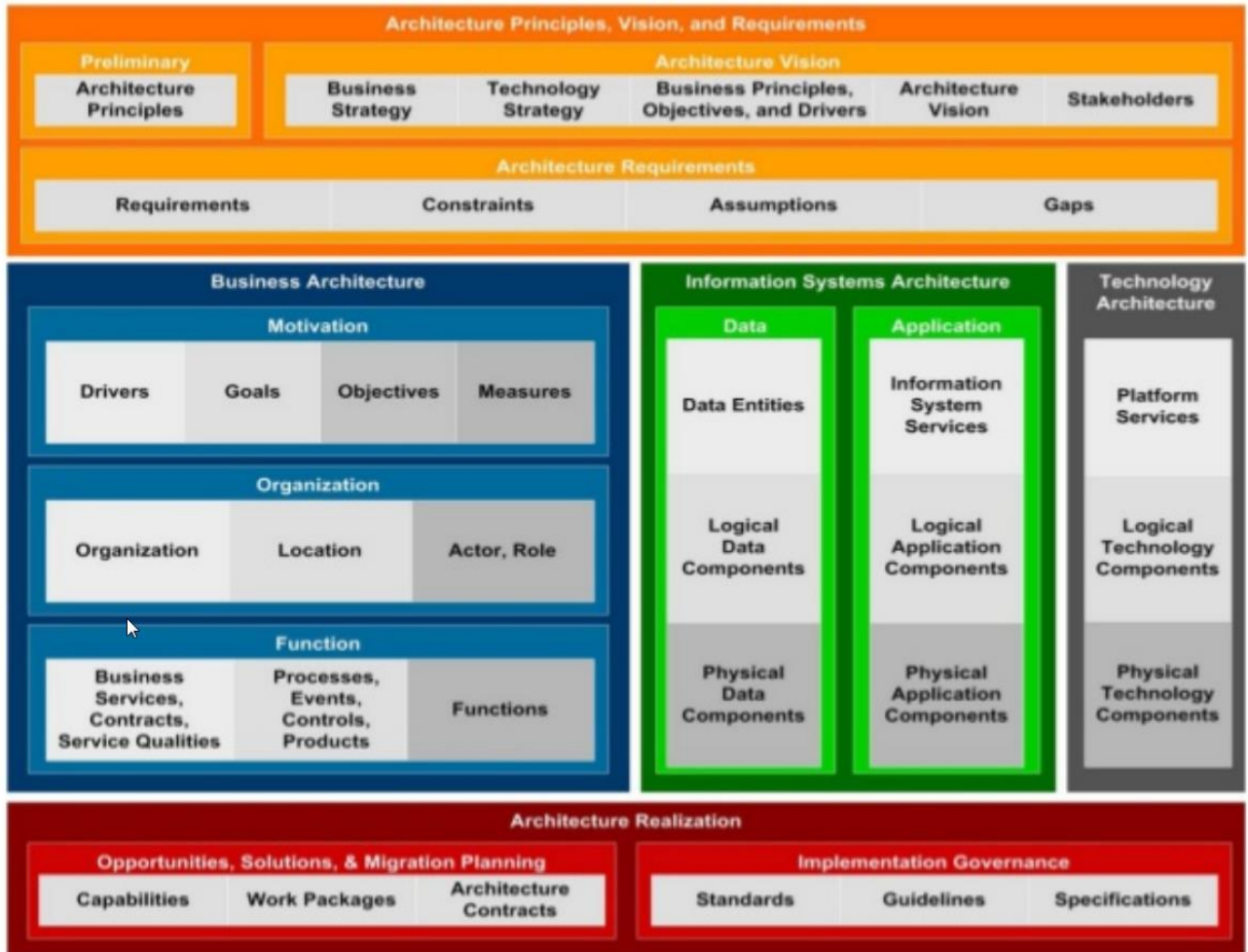
2003 (декабрь) – опубликована версия 8.1

Есть соответствие между понятиями TOGAF и Захмана

Структура ToGAF



Метамодел



Методика ADM

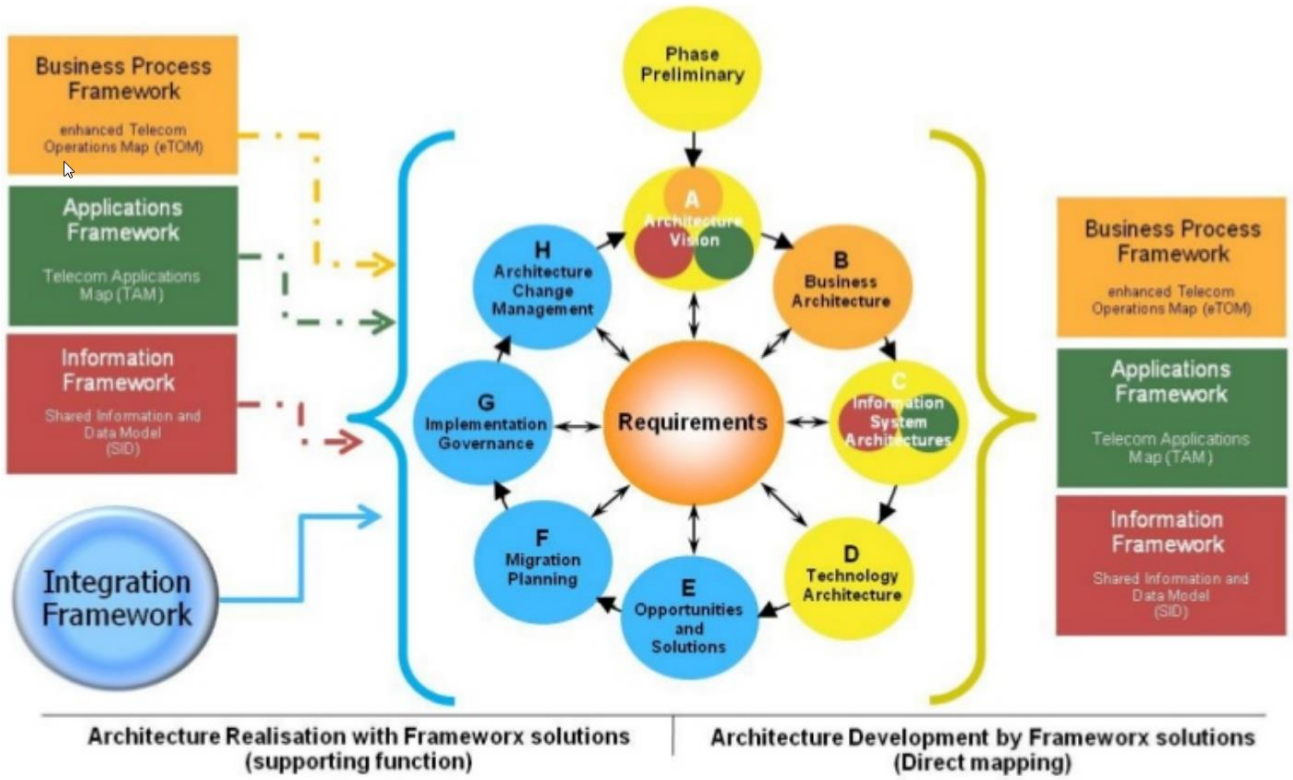
1. Scope, Vision, утверждение руководством
2. Бизнес-архитектура предприятия
3. Архитектуры данных и приложений
4. Технологическая архитектура
5. Проверка возможности реализации
6. План перехода к новой системе
7. Система управления преобразованиями
8. **Управление изменениями**

!! Допускается **изменение последовательности** фаз

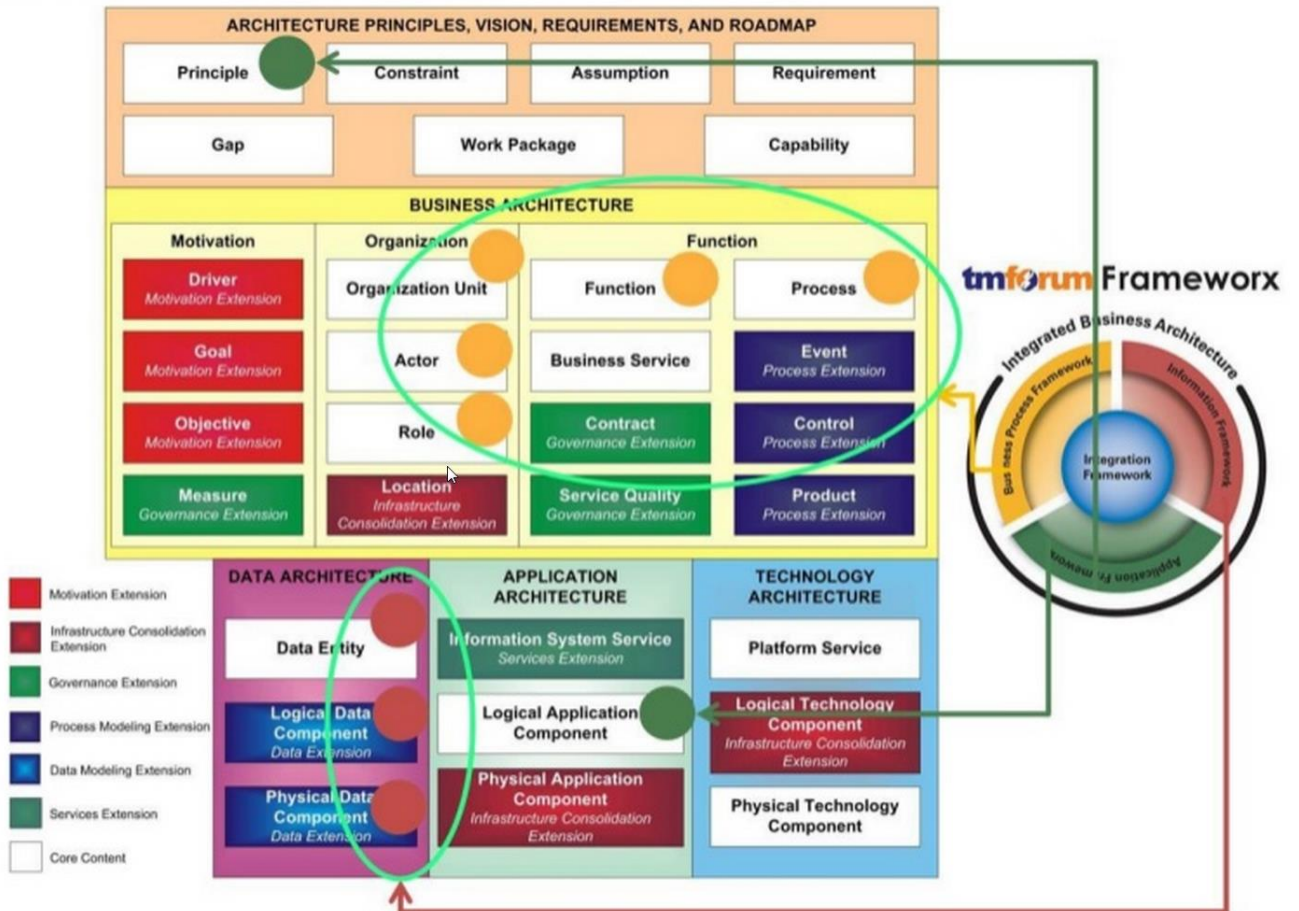
Например, при внедрении ERP-системы на предприятии – сначала будет выполнена фаза 4

И только потом – фазы 2 и 3 !

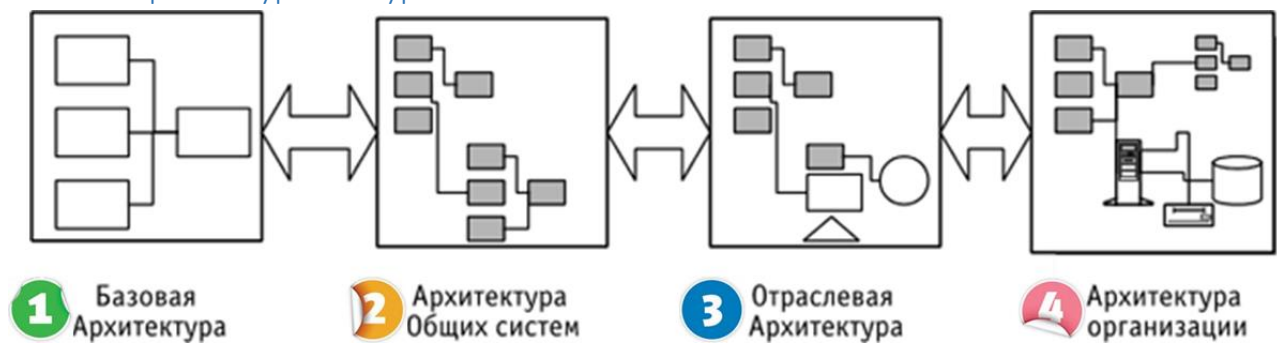
Связь ToGAF и TMF



Явные точки пересечения



Базовая архитектура и её уровни



Абстрактная реализация ИТ-системы в целом

ReUse в различных функциональных областях

Контекстно-зависимые модели (телеком...)

ИТ-архитектура конкретного предприятия (унаследованные системы, планы реализации...)

Принципы ToGAF и пример набора принципов

Принципы – отправная точка для разработки конкретных решений

«Хороший» набор принципов:

- Понятны
- Однозначны в формулировках
- Последовательны

Принцип	Формулировка
Максимальная польза	Решения в области ИТ принимаются исходя из максимума пользы для организации в целом
Непрерывность бизнеса	Деятельность предприятия должна обеспечиваться, несмотря на возможные помехи в работе ИТ
Обеспечение качества	Каждый элемент данных должен иметь ответственного за качество
Общие метаданные	Метаданные должны быть едиными в рамках предприятия и доступными для всех пользователей
Безопасность данных	Данные должны быть защищены от неавторизованного использования и распространения
Простота использования	Оптимальное выполнение бизнес-задач за счет единого интерфейса, интеграции систем, снижения вероятности неправильного использования
Минимизация разнообразия	Уменьшение числа различных вариантов применяемых платформ, продуктов и версий

Другие методики и их сравнение

Gartner

[<http://www.gartner.com/technology/research/enterprise-architecture.jsp>]

META Group [<http://www.meta-group.com/SitePages/default.aspx>]

NASCIO [<http://www.nascio.org/resources/EAResources.cfm>]

Microsoft [<http://msdn.microsoft.com/architecture>]

	Захман	TOGAF	Gartner	NASCIO	Microsoft
Иерархический подход, связь с бизнес-стратегией					
Поддержка различных уровней абстракции					
Формальный язык и система обозначений					
Описание процесса разработки архитектуры					
Рекомендации по управлению архитектурой					

Архитектура предприятия – как проект

Проект выгоднее, когда ещё ничего нет

Проект выгоднее, когда происходит миграция на новые критические системы (ERP) или выход на новые рынки

Задачи проекта:

1. Привлечение руководства, бизнес-подразделений и планирование работ.
2. Понимание стратегии развития бизнеса организации.
3. Выработка общих для бизнеса и ИТ требований к целевой архитектуре;
4. Разработка согласованного и полного набора принципов для построения архитектуры

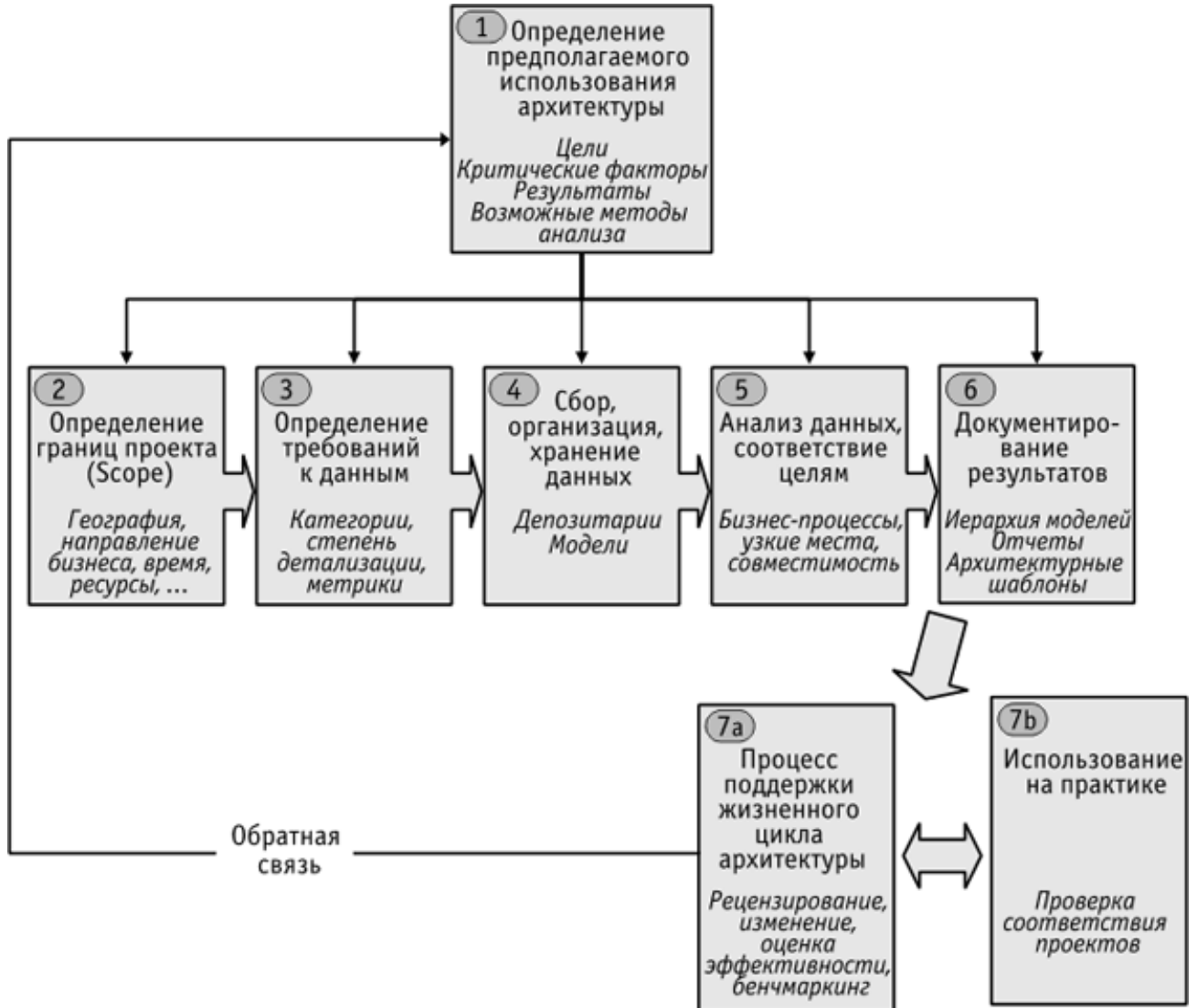
Детальнее можно раскрыть так:

1. Определить и **обосновать** цель
2. Инициировать проект (подробности дальше)
3. Определиться с существующим состоянием
4. Определиться с финальным состоянием
5. Проанализировать разницу
6. Разработать план перехода
7. Подтвердить (проверить) достижимость
8. **ВЫПОЛНИТЬ**

Как инициировать проект?

1. Определить правила и границы проекта
2. Подготовить бизнес-обоснование
3. Получить поддержку руководства
4. Определить состав рабочей группы
5. Определить необходимые документы
6. Создать локальные рабочие группы

Основные элементы проекта



Три типовых подхода:

1. Традиционный (регламент -> as is -> to be -> детальное проектирование -> разработка) – плохо, поскольку возможна ситуация «паралича анализа»
2. Сегментированный (сосредотачиваемся на главных бизнес-сферах, отбрасываем лишнее)
3. Сохранить статус-кво (проще говоря, оставить всё как есть, поскольку «ничего не выйдет»)

Общая схема



Эта схема состоит из следующих шагов:

Общим фоном для этого процесса является мониторинг существующих тенденций в области деятельности организации и тенденций в области развития информационных технологий.

Анализ на бизнес-уровне. На первом этапе проводится анализ движущих сил, которые влияют на необходимость использования ИТ с точки зрения основных функций и бизнеса организации. Определяются требования бизнеса и технологии на текущем этапе и на перспективу, которые задают требования к информационным системам. Учитываются тенденции в развитии информационных технологий и мировых аналогов с учетом перспектив развития бизнеса.

На основе этого анализа формулируются в самом общем виде требования к информационным технологиям с точки зрения информации (данных) и архитектуры ИТ.

Принимаются общие для организации стандарты и понятия о том, что такое Архитектура предприятия: принципы, общие методы описания архитектуры и ее разделы, стандарты, конкретные продукты и технологии.

Параллельно с этими процессами выполняется анализ на "системном уровне": аудит используемых информационных технологий и программно-технических средств, аудит организации процессов управления ИТ, внедрения технологий и приложений.

Результаты вышеперечисленных этапов являются основой для выполнения "Gap-анализа", т.е. выявления расхождений и различий между существующей ИТ-инфраструктурой и желаемой архитектурой предприятия.

Результаты Gap-анализа ложатся в основу Плана миграции: определяются цели создания (модернизации) информационных систем и решаемых ими задач, согласовывается стратегия разработки и внедрения информационных технологий (перечень критических процессов, подлежащих автоматизации в первую очередь и т. д.), обсуждается план детального анализа.

После этого начинается фаза реализации конкретных проектов в рамках выработанной на данный момент архитектуры предприятия.



Как разрабатывать?

	Плюсы	Недостатки
	<ul style="list-style-type: none">• Создаётся видение ситуации в целом• Сформулированы бизнес-потребности• Большой охват и поддержка руководства	<ul style="list-style-type: none">• Повышенная абстрактность• Нет явных быстрых результатов• Много малопонятных артефактов• Разочарование руководства• Необходимо обучение исполнителей• Необходимы опытные исполнители
	<ul style="list-style-type: none">• Быстрые результаты• Решение приоритетных проблем• Постепенный рост сложности• Меньше исполнителей• Ориентация на решение конкретных задач• Конкретная экономия от решений	<ul style="list-style-type: none">• Замкнутость на IT, с малым участием бизнеса• Разочарование руководства• Очерёдность решения стоящих задач

Подход "сверху-вниз" предполагает достаточно широкий охват проблем и точное следование формальному процессу. Основу этому подходу положили методики Захмана и Спивака. Он начинается со сбора информации, требующейся для описания различных доменов архитектуры "как есть". Далее следует этап, связанный с описанием и реинжинирингом бизнес-процессов, консолидации прикладных систем, выстраивание архитектуры данных и, наконец, стандартизация технологической архитектуры. Например, многие государственные проекты ориентированы на этот подход (например, в США в рамках Федеральной архитектуры FEAF).

Подход "снизу-вверх", когда процесс начинается со стандартизации инфраструктурных технологий (технологическая архитектура), а затем развивается в направлении решения проблем более высокого уровня и, в конечном итоге, решает вопросы, связанные с бизнес-архитектурой. Этот подход, видимо, имеет более широкое распространение в бизнесе и в частном секторе.

Как обосновать?

1. Понять – какие факторы толкают к EA?
2. Напомнить – наличие EA снижает затрат и расходов на ИТ, позволяет повторно определять и использовать технологии
3. Ещё напомнить – про снижение стоимости разработки, внедрения и поддержки, упрощение процессов управления системами
4. Сослаться на Gartner – отсутствие EA ведёт к увеличению эксплуатационных расходов на ИТ (до 20%)

Это могут быть, например, макроэкономические факторы, требующие переосмысления вклада ИТ в бизнес, или конкурентная ситуация, требующая новых прикладных систем и обеспечивающей инфраструктуры (например, децентрализации процесса приема заказов). Факторы могут быть связаны с изменениями в бизнес-стратегиях, например, с принятием решения об организации более индивидуализированной работы с клиентами, что потребует внедрения целого ряда новых

информационных систем. Важно понимать, как эти факторы могут быть использованы при обосновании проекта разработки архитектуры перед высшим руководством.

Экономический расчёт обоснования

ROI – может не сработать (тактический показатель, «работает» в рамках проекта)

ROA – стоит попробовать (стратегический показатель, определяет повышение эффективности основных фондов)

ROO – стоит попробовать (не является формально финансовой метрикой, но описывает влияние инвестиций на бизнес)

$$\text{ROI (\%)} = \frac{\text{Выгода - Затраты}}{\text{Затраты}}$$

$$\text{ROA (\%)} = \frac{\text{Прибыль}}{\text{Активы}}$$

$$\text{ROO (\%)} = \frac{\text{Достижение S.M.A.R.T.-целей}^*}{\text{Затраты}}$$

***SMART = Specific, Measurable, Achievable, Relevant, and Timely)**
Конкретная, Измеримая, Достижимая, Сопоставимая, Во времени

На чём основывать расчёт?

[Как вариант] Оценить изменение бизнес-процессов

$$\text{Прибыльность процесса} = \frac{\text{Прибыль на выходе}}{\text{Затраты на входе}}$$

$$\text{Эффективность процесса} = \frac{\text{Прирост прибыли}}{\text{Затраты на улучшение}}$$

$$\text{Стоимость процесса} = \sum \text{Всех затрат на выполнение одного экземпляра процесса}$$

Факторы успеха

1. Тщательное планирование
2. Адекватное финансирование
3. Обеспечение ресурсами (люди и время)
4. Мотивация и реализация («кнут и пряник»)
5. Талант и квалификация команды
6. Видение цели

Влияние психологии на успех проекта

	Высшее руководство	Менеджеры	Разработчики	ИТ-эксплуатация
Способность к взаимодействию				
Стоимость владения ИС				
Гибкость и масштабируемость				
Быстрота разработки				
Соответствие бизнесу				
Использование ресурсов				
Безопасность и риск				
Инновации в бизнесе				

Несколько критических слов о проектах Enterprise Architecture

Gartner 2007:

К 2012-му году 40% от всех проектов, которые сейчас запущены – будут свёрнуты до завершения

IDS Scheer 2008:

2/3 EA-проектов оказались неспособны улучшить взаимодействие бизнеса и ИТ-блока

Независимые консультанты 2007...2010:

90% всех проектов никогда не принесут полезных результатов... Объективные причины неудач ещё предстоит определить