

Вадим Алджанов

ИТ-архитектура

Практическое руководство от А до Я. Первое издание

```
mirror_mod = modifier_ob.modifiers.new("mirror_mod", mirror_ob)
# Add mirror object to mirror_ob
mirror_mod.mirror_object = mirror_ob

# operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
# operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
# operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

# Selection at the end -add back the deselected objects
mirror_ob.select = 1
modifier_ob.select = 1
key.context.scene.objects.active = modifier_ob
print "selected" + str(modifier_ob) # modifier_ob
mirror_ob.select = 0
key = key.context.selected_objects[0]
key.data.objects[one.name].select = 1

print "please select exactly two objects, we"

OPERATOR CLASSES
```

Вадим Алджанов

**ИТ-архитектура.
Практическое руководство
от А до Я. Первое издание**

«Издательские решения»

Алджанов В.

ИТ-архитектура. Практическое руководство от А до Я. Первое издание / В. Алджанов — «Издательские решения»,

ISBN 978-5-44-905650-4

В книге собраны и обобщены теоретические знания и примеры реализации в таких областях ИТ, как построение архитектуры предприятия, информационной безопасности, управление ИТ-сервисами, управление проектами, качеством и рисками, бизнес-процессами. Даются системное описание ИТ, модели и подходы по формированию ИТ-стратегии, требования к системам обеспечения, выбор компонентов ИТ-инфраструктуры, расчет стоимости владения и организационные аспекты управления ИТ.

ISBN 978-5-44-905650-4

© Алджанов В.
© Издательские решения

Содержание

Предисловие	7
Об авторе	7
Цели книги	8
Сферы, охваченные книгой	9
Благодарность	11
Юридическое уведомление	12
Авторские права	13
Отказ от ответственности	14
Часть I: Теоретические Основы	15
Концепция Архитектуры Предприятия	16
Общие Положения	16
Основные аспекты Архитектуры Предприятия	18
Определение Структуры Предприятия	18
Определение роли ИТ в составе организации	19
Базовые принципы построения Архитектуры Предприятия	22
Уровни зрелости ИТ	25
Критерии выбора методологии	27
Архитектура Предприятия на основе методологии «ZAHMAN FRAMEWORKS»	29
Архитектура Предприятия на основе методологии «FEAF»	30
Архитектура Предприятия на основе методологии «GARTNER»	31
Архитектура Предприятия на основе методологии «TOGAF»	32
Базовые принципы	33
Метод Развития Архитектуры (Architecture Development Method ADM) TOGAF	34
Процесс Управления Архитектурной Практикой	38
Ключевые документы и шаблоны методологии TOGAF	40
Подходы к построению ИТ Стратегии	41
Стратегия Управления	46
Стратегия финансового контроля	46
Стратегия административных регуляторов	46
Стратегия Компонентов ИТ Архитектуры	48
Стратегия выбора платформа решения	48
Стратегия выбора платформа развертывания	49
Стратегия выбора аппаратного и программного обеспечения	49
Стратегия лицензирования	50
Стратегия построения инженерных систем	51
Стратегия тестирования	51
Стратегия уровня избыточности	51
Стратегия по системам резервирования и архивирования	52
Требования к резервным сайтам	52
Подходы по внедрению Архитектуры Предприятия	53
Инструменты внедрения Архитектуры Предприятия	55
Рекомендации по внедрению	56
Заключение	57

Концепция Управления Проектами	58
Общие Положения	58
Ключевые аспекты Управления Проектами	59
Подходы и методы управления проектами	62
Процессно-ориентированное проектное управление (Process Based Project Management PBPM)	62
Классическая Методология Управления Проектами «ВОДОПАД» (WATERFLOW)	63
Гибкая Методология Управления Проектами (Agile Project Management Methodology)	65
Метод SCRUM семейства Гибких Методов Управления Проектами	68
Метод KANBAN семейства Гибких Методов Управления Проектами	70
Гибридная Методология Управления Проектами	72
Методология Быстрой Разработки Приложений (Rapid Application Development RAD)	73
Методология Экстремального Программирование (Extreme Programming XP)	75
Методология моделирования событий Event Chain Methodology (ECM)	76
Метод Адаптивные Рамки Проектов APF (Adaptive Project Framework)	77
Метод «Реализация Выгоды» (Benefit Realization BF)	78
Методология PRISM (проекты со встроенными устойчивыми/жизнеспособными методами)	79
ПРОЧИЕ МЕТОДЫ И СПЕЦИФИЧЕСКИЕ ПОДХОДЫ К УПРАВЛЕНИЮ ПРОЕКТАМИ	80
Методология Управления Проектами в Контролируемой Среде (PRINCE2)	81
Принципы методологии PRINCE2	81
Темы методологии управления проектами PRINCE 2	81
Семь процессов управления PRINCE 2	82
Управление проектами на основе методологии «PROJECT MANAGEMENT INSTITUTE PMI»	83
Общие положения	83
Группа процессов инициации проекта (Initiating)	83
Группа процессов планирования проекта (Planning)	83
Группа процессов реализации проекта (Executing)	88
Конец ознакомительного фрагмента.	89

ИТ-архитектура
Практическое руководство
от А до Я. Первое издание

Вадим Алджанов

© Вадим Алджанов, 2018

ISBN 978-5-4490-5650-4

Создано в интеллектуальной издательской системе Ridero

Предисловие

Об авторе



Вадим Алджанов (англ. Vadim Aldzhanov) [Microsoft MCP, MCSA Security, MCSE Security, MCTS, MCITP, MCITP SQL Database Administrator, Cisco CCNA, VMware VCP4, CompTIA A+, Network+, Security+, EC—Council CEH и ECSA, SNIA Certified Storage Professional SCSP, Wireless Technology CWTS, CWNA, CWSP, IT Management ITILv3, Apple Certified Associate – Integration | Management].

В руководстве собраны и обобщены знания и опыт за более чем 17+ лет работы в ИТ. В течении 14 лет проработал в банковской сфере, большую часть времени на позиции руководителя ИТ департамента. На данный момент являюсь ИТ Архитектором в одном из крупных холдингов страны. Имею степень бакалавра по специальности «Радиотехника» и степень магистра по направлению «Компьютерные Информационные Системы (CIS)». Продолжаю образование на получение докторской степени по направлению «Менеджмент Информационных Систем (MIS)». Кроме этого имеется порядка тысячи часов обучения на специализированных курсах по направлениям системное администрирование, компьютерные сети, беспроводные сети, системы хранения, системы виртуализации, информационная безопасность, управление ИТ сервисами, управление проектами, банковское дело, пластиковые карты, стратегическое планирование, проведение аудита и прочие. Профиль в LinkedIn: <https://www.linkedin.com/in/vadim-aldzhanov-623a7b44/>

Цели книги

Цели книги, помочь специалистам, руководителям и директорам ИТ в построении Архитектуры Предприятия, организации процессов управления, расчете стоимости внедрения и сопровождения ИТ инфраструктуры, в выборе оптимального архитектурного решения, как с точки зрения бизнеса, так и с точки зрения ИТ. Книга поможет организовать коммуникацию между бизнесом и ИТ, позволив им общаться на одном «языке», а также задает вектор развития для организации в целом и ИТ в частности.

Книга не является обязательным руководством по выбору того или иного продукта или решения, а выражает точку зрения автора. Материал изложен в логической последовательности, дополнен теоретическими сведениями и снабжен наглядными примерами реализации. Это дает возможность использования данного руководства для методичного изучения всех аспектов деятельности ИТ, наряду с использованием его в качестве справочного пособия при работе с конкретными системами.

Сферы, охваченные книгой

Книга представляет собой руководство на русском языке, в котором собраны и обобщены как теоретические знания, в области построения Архитектуры Предприятия, Управления Проектами, Информационной Безопасности, Организации и Управления ИТ сервисами и ИТ аудита, так и рассмотрена последовательность применения их на практике, позволяющий полностью обеспечить потребности организации в процессе создания и управления ИТ архитектурой и инфраструктурой.

Книга разделена на две части: Первая часть содержит информацию по общим вопросам связанными с функционирование ИТ. Описывает общие положения, концепции, сервисы и задачи, требуемые для построения ИТ архитектуры. Вторая часть содержит рекомендации по организации работы ИТ, ключевые вопросы и процессы его функционирования, а также детализированную информацию по развертыванию ИТ сервисов на конкретных примерах.

В книге рассматриваются вопросы разработки и использования Архитектуры Предприятий, включающие в себя такие аспекты, как Бизнес-архитектура, Архитектура информации, Архитектура прикладных систем и Технологическая архитектура. способы объединения функциональных и бизнес-потребностей организаций с возможностями информационных технологий. В книге, на основе обширного материала, дается систематичное описание современного состояния ИТ компании, приводятся основные модели и подходы по формированию ИТ стратегии, управлению проектами, управлению ИТ сервисами, анализу рисков и управлению качеством, Контроль и аудит ИТ, интеграции и взаимодействие различных подходов и методов. Помимо теоретической базы, в книге представлены практические примеры реализации ИТ инфраструктуры, требования ИТ предъявляемые к системам обеспечения, структурированной кабельной системе, вопросы выбора ИТ решений для бизнеса и многое другое. Рассмотрены требования к элементам ИТ инфраструктуры и анализ альтернативных ИТ решений.

Книга предназначена для широкого круга читателей и будут полезна:

Топ-менеджерам, кураторам ИТ, ИТ директорам крупных и средних компаний так как позволит лучше понимать Архитектуру Предприятия (Enterprise Architecture) на базе подхода (TOGAF), роль и вовлеченность ИТ в бизнес. В книге содержится информация по таким ключевым вопросам, как теоретические и практические показатели стоимости информационных технологий для организации, методы расчета и распределения расходов. Показатели распределения финансовых инвестиций на ИТ сервисы. Представители бизнеса смогут понять общие аспекты по функционированию ИТ инфраструктуры, технические термины, принципиальные отличия различных архитектурных решений, принципы построения и сопровождения технических решений. Позволяет сформировать понимаемые обоими сторонами метрики и отчеты по оценке эффективности и результативности функционирования ИТ инфраструктуры.

Руководителям ИТ подразделений, ИТ архитекторам, менеджеров среднего звена ИТ департамента, а также менеджерам проектов книга предоставляет теоретические основы управления ИТ сервисами (ITSM) с применением рекомендаций практик ITIL, интеграцию методологии Управление Проектами (PMI) в ИТ, вопросы аудита ИТ (CobiT) и Информационной Безопасности. Вся теоретическая база рассматривается на практических примерах реализации ИТ архитектуры и инфраструктуры.

ИТ специалистам помимо теоретических основ внедрения и сопровождения ИТ сервисов предоставляется возможность рассмотрения построения компонентов ИТ инфраструктуры на примере таких продуктов как Windows 2016 Server и Windows 10, комплексного решение Microsoft System Center 2016, корпоративного портала Microsoft SharePoint Server 2016 и Microsoft Project Server 2016, Exchange 2016 и Skype for Business 2015, функции Direct Access, Hyper-V и другие.

Материал изложен в логической последовательности, дополнен теоретическими сведениями и снабжен наглядными примерами реализации, что дает возможность для методичного изучения всех аспектов деятельности ИТ, наряду с использованием его в качестве справочного пособия при работе с конкретными системами.

Благодарность

Выражаю благодарность друзьям, учителям, руководителям и коллегам за помощь в написании книги, а также бесценный опыт и знания полученный от общения с такими людьми как Александр Буслаев («AIG Group»), Иршад Гулиев («SINAM»), Фазиль Маммедов («ROTABANK»), Яна Хмельницкая и Karsten Stellner («LFS Financial Systems GmbH»), Thomas Engelhardt («Microfinance Bank of Azerbaijan»), Andrew Pospelovsky («ACCESSBANK») и Alan Crompton («Baku European Games Operation Committee BEGOC 2015»).

Юридическое уведомление

Информация, содержащаяся в книге, не несет в себе никакой коммерческой тайны или иной конфиденциальной информации. Материалы собраны из открытых источников, переработаны автором, используя имеющийся опыт и знания. Некоторые рассмотренные примеры приведены только для справки и являются вымышленными. Любое сходство с реально существующими людьми или организациями является случайным. Все упоминающийся в книге названия компаний и продуктов могут быть торговыми марками, принадлежащими соответствующим владельцам.

Авторские права

Информация, указанная в книге не может воспроизводиться, дублироваться, копироваться, передаваться, распространяться, храниться или использоваться иным образом для любого коммерческого и не коммерческого использования без письменного согласия автора.

Отказ от ответственности

Автор не дает никаких гарантий или заявлений о точности, пригодности или полноте информации, ссылок или других предметов, которые содержатся в настоящем документе. Книга доступна всем читателям «как есть» без каких-либо заявлений или гарантий любого рода, явных или подразумеваемых, включая гарантии в отношении товарности или пригодности для определенной цели. Документ может содержать неточности или орфографические ошибки.

Автор не несет никакой ответственности за прямые, косвенные, случайные или прочие убытки при использовании данного руководства. Читатель данного руководства проинформирован.

Посвящается моим родителям, любящей жене и двум прекрасным дочерям

Часть I: Теоретические Основы

Книга разбита на две части. Содержание первой части книги включает в себя рассмотрение теоретических основ по построению ИТ компании. В частности, рассмотрены:

Глава 1: Концепции построения Архитектуры Предприятия – В данной главе рассматриваются вопросы построения Архитектуры Предприятия, построения ИТ стратегии и т.п.

Глава 2: Концепция Управления Проектами – в главе рассматриваются основы Управления Проектами, применяемые методологии, принятые методы и т.п.

Глава 3: Концепция Управления Рисками – в главе рассматриваются методики и методы оценки рисков, классификация рисков и типы реагирования на риски.

Глава 4: Концепция Управления Качеством – в главе рассмотрены базовые принципы и методы управления проектами при использовании методов, основанных на принципах управления качеством и бережливого производства.

Глава 5: Концепция Управление Бизнес Процессами и Модели Деловой Активности различных сфер бизнеса, Рассмотрены основы построения бизнес процессов, виду организации бизнеса и связь с информационными системами.

Глава 6: Концепция Интеграции Данных и централизация ИТ – В данной главе рассмотрены вопросы по интеграции данных между различными информационными системами, различные архитектурные решения, проблемы и возможности. Рассмотрены уровни централизации Автоматизированных Систем Управления.

Глава 7: Концепция выбора ИТ решений – Рассмотрены проблемы выбора ИТ решений, а также предложены механизмы по выбору и обоснованию того или иного решения.

Глава 8: Концепция Информационной Безопасности – Рассмотрены вопросы информационной безопасности, порядок организации взаимодействия информационной безопасности и ИТ.

Глава 9: Концепция Управления ИТ сервисами – Рассмотрены процессы построения управления ИТ сервисами с применением практики ITIL.

Глава 10: Концепция Контроля и аудита ИТ – рассмотрены общие вопросы по контролю ИТ и проведению аудита

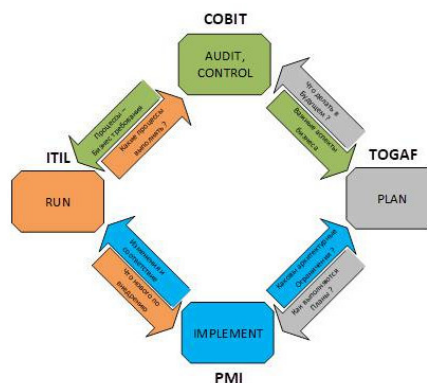
Глава 11: Концепция Определение стоимости и модели финансирования ИТ – Рассмотрены модели финансирования, принципы оценки ИТ проектов, методы и практики расчета стоимости ИТ сервисов и т.п.

Концепция Архитектуры Предприятия

Общие Положения

В данной главе описывается общая информация по Архитектуре Предприятия (Enterprise Architecture). Обобщенное определение можно представить, как:

Архитектура Предприятия – это набор принципов, методов и моделей, который используется в проектировании и реализации организационной структуры, бизнес-процессов, информационных систем и технологий. Она представляет из себя управленческую практику, направленную на максимизацию отдачи от ресурсов предприятия, инвестиций в ИТ, деятельности по разработке систем в процессе достижения целей предприятия, перевод видения и стратегии бизнеса в эффективное изменение компании посредством создания, обсуждения и улучшения ключевых требований и принципов, которые описывают будущее состояние компании и делают возможным её развитие.



Так как Архитектура Предприятия представляет из себя сложное и комплексное решение, включающая в себя переплетение различных методологий и техник, при построении Архитектуры Предприятия должны быть учтены, но не ограничены, рекомендациями следующих стандартов:

- TOGAF – Enterprise Architecture
- ISO/IEC 20000 – Quality in IT Service Management
- ISO/IEC 27000 – Best Practice IT Security Standards
- CobiT v5 – Audit and Control Framework
- ITIL v3 – Best practices in IT Service Management
- MOF – Microsoft Operations Framework
- PMI – Project Management Institute

Архитектура призвана ответить на такие вызовы и проблемы организации как:

- Недовольство бизнеса ИТ-службой. Причины могут быть объективные или субъективные.
- Невозможность оценить эффективность использования информационных технологий в бизнесе.
- Отсутствие порядка в «зоопарке» ИТ решений и систем, внедренных в организации.
- Сложность принятия решений, связанных с информационными технологиями

- Сложность согласование ИТ-бюджета и запуск ИТ-проектов.
- Рост ценности «Информации» и связанности бизнеса и ИТ.
- Отсутствие прозрачных и понятных связей бизнеса и ИТ?
- Можно ли решить актуальные задачи бизнеса с использованием ИТ?
- Как заставить ИТ давать компании большую ценность?
- Как нужно менять ИТ при изменениях в бизнесе?
- ИТ системы сложны, не управляемы и дороги в обслуживании
- ИТ системы сдерживают организацию от адекватного реагирования на изменения в бизнесе
- Критичная для бизнеса информация не своевременна и не адекватна
- Отсутствует культура общения бизнеса и ИТ

Как результат бизнес не видит ценности в информационных технологиях. ИТ директору тяжело продвинуть новые идеи, если он говорит о технологиях. Его не понимают. Все, что ему остается, это поддерживать то, что есть, и делать те задачки, которые подбрасывает бизнес. Возникают серьезные сложности с обоснованием ИТ бюджетов. ИТ директор по факту больше похож на прораба, латающего дыры, а не на топ менеджера, который работает над развитием компании. Топ менеджеры быстро теряют интерес к ИТ проектам, как следствие, теряют финансирование и проваливаются. На место ИТ департамента приходят различные системные интеграторы по внедрению «супер-пуперских» решений, которые «спасут» бизнес. Или возникают идеи забрать все ИТ активы и услуги компании и передать на аутсорсинг. Борьба ИТ департаменту с интеграторами будет сложно и результат будет предсказуем, ключевая компетенция у интеграторов одна – технологии, и здесь им равных нет. ИТ департамент превращается в «болото», и его покидают лучшие сотрудники, а вместе с ними будут утеряны уникальные знания и навыки. Цели у интегратора или аутсорсинг компании такая же, как и у вашей компании – получение прибыли. Но в отличие от ИТ департамента, интересы которого совпадают с интересами вашей компании, интересы интегратора могут не совпадать с вашими интересами, или уникальными идеями и видением. В лучшем случае будет «как у всех», и бизнес потеряет идентичность (если он неразрывно связан с ИТ) или говоря словами персонажа одного из кинофильмов: «... будет все у нас все по-новому оставаясь все по-старому...». Конец печален.

Основные аспекты Архитектуры Предприятия

Для борьбы с выше сказанными проблемами и последствиями, Архитектура Предприятия помогает сформулировать следующие важные критерии.

Определение Структуры Предприятия

При построении Архитектуры Предприятия самый первый и наиболее важный аспект – понимание организационной структуры предприятия, принципов управления, принятия решения и т.п.

Структура организации – фиксированное упорядоченное множество объектов и связей между ними.

По характеру специализации и видам деятельности различают:

Вертикальное разделение – по уровню подчинения

Горизонтальное разделение – по функциям и специфике деятельности

Соответственно в структуре управления различают линейные связи по типу «начальник – подчинённый» и функциональные «профессиональная интеграция на основе специфики деятельности». Как результат, тип структуры организации можно представить, как следующие основные типы и их разновидности:

Плоская – наиболее простая структура. Подходит для рабочих или проектных групп, или небольшой организации.

Иерархическая (бюрократическая) – формирование структуры исходя из структуры организации, разделение труда на функции и ответственности работников.

Линейная – управление по прямой (руководитель – подчинённый), общение между подразделениями только через руководителей подразделений

Функциональная – взаимодействие идет по функциональному признаку

Линейно-функциональная – взаимодействие совмещается по линейному и функциональному типу (наиболее используемая модель)

Линейно-штабная – отдельные функциональные группы (штабы), самостоятельно ведущие работу с подразделениями или организациями. Как пример группа компаний в составе холдинга.

Дивизионная – характеризуется центральной координацией с децентрализованным управлением. Ключевые фигуры в данном случае не руководители функциональных подразделений, а менеджеры отдельных объединений (менеджеры филиалов, заводов и т.п.)

Органические (адаптивные) – формирование структуры исходит из необходимости приспособления к изменениям. Отношения формируются, исходя не из структуры, а характера поставленных задач.

Проектные – организуется при ведении проектов

Матричные (программно-целевые) – принцип двойного подчинения, непосредственное подчинения руководителю, и проектному менеджеру

Бригадные (кросс-функциональные) – работа обособленными группами, с самостоятельным управлением и принятием решений (противоположность иерархическому типу)

Существование той или иной организационной структуры может зависеть от ряда факторов:

- Специфика и степень разнообразия деятельности
- Географическое размещение

- Уровень централизации в организации
- Стратегия организации
- Количества и спектра предоставляемых услуг

Определение роли ИТ в составе организации

Одним из главных критериев при построении Архитектуры Предприятия можно выделить задачу по определению роли ИТ в составе организации. Если отбросить множество статей, рекомендаций различных консультантов о значимости ИТ в современном мире и «бла бла бла», необходимо четко и конкретно зафиксировать роль ИТ в составе организации, задачи, права возможности и степень ответственности.

Для понимания роли ИТ в конкретной организации нужно ответить на следующие вопросы:

Вовлеченность ИТ в бизнес организации. Насколько бизнес организации зависит от ИТ. Многие бизнес процессы и функции завязаны на сложных, централизованных или специфических ИТ решениях. Развитие бизнеса невозможно без быстрой и качественной работы ИТ. Есть целые отрасли, которые зависят от ИТ: банки, страховщики, прочие финансисты, сервисные компании, госорганы, технологические компании, энергетика и прочие (как пример в банке две трети сотрудников так или иначе работают с централизованной банковской программой, в то время как для сельского хозяйства, лишь небольшая часть организации использует ИТ решения, да и те по большей части автономные – большинство сотрудников работают на поле или уровень компьютеризации компании низкий).

Вашу организацию можно отнести к крупному или среднему бизнесу. На мой взгляд, до тех пор, пока все технологии компании со всеми подробностями помещаются в одной голове, а ИТ директор напрямую общается с руководством организации об архитектуре предприятия думать рано. Для малого бизнеса архитектура предприятия вообще, как телега пятое колесо.

Компания активно развивает ИТ. В компании идет несколько ИТ проектов в год, или хотя бы один проект по внедрению ERP, CRM или прочего сложного решения. Формирование Архитектуры Предприятия поможет с принятием решений и убережет от большинства переделок, ошибок, не состыковок, задержек и прочих проблем. У кого-то, и желательно не одного, должна быть общая картинка будущего и понимание процесса развития. Иначе кусочки мозаики, созданные в разных проектах, не сойдутся.

У компании периодически происходят кризисы в ИТ, которые имеют существенное влияние на бизнес. В информационных системах происходят сбои, которые негативно влияют на бизнес, вплоть до его остановки. Сбои вызваны проблемами в интеграции, просчетами в инфраструктурных решениях, временными решениями и просто бардаком. Не все проекты, в том числе и ИТ, заканчиваются успешно, уложившись в сроки, бюджет и заявленные требования. Если руководство вашей компании устало от провалов, задержек, превышения бюджетов ИТ проектов, то стоит задуматься.

Компании важна скорость, качество и эффективность развития ИТ. Одна из сторон (Бизнес или ИТ) развивается значительно быстрее чем другая. В случае, если бизнес развивается быстрее чем ИТ, то ИТ становится тормозом в развитии компании. И наоборот, если ИТ развивается быстрее чем необходимо бизнесу, то владельцы бизнеса несут убытки как финансовые (хорошее ИТ стоит денег), так и упущенная прибыль (бизнес не использует все потенциальные возможности ИТ). Необходимо организовать работу обеих сторон «на одной волне» для гармоничного развития всех элементов компании.

Выстраивание взаимоотношения между бизнесом и ИТ

Сфокусировать действия ИТ на целях и задачах бизнеса. Бизнес и ИТ стороны по-разному видят задачи, цели и ожидания по отношению друг к другу. Видение ИТ сотрудников «... мы прекрасно разбираемся в технологиях, нам платят деньги за умение программировать, настраивать, устанавливать и решать технические проблемы и т.п. Наша работа начинается после получения технического задания...». Видение со стороны бизнеса – «... столько новых ИТ технологий, ИТ должен нам внедрить нам какое-то решение, чтобы увеличить объем продаж. Или что еще хуже, нам нужно решение ХХХ, оно есть у конкурентов и поэтому продажи у них выше... внедряйте мы уже все решили вы ничего не понимаете бизнесе...». Задача ИТ директора, участие на равных в обсуждении стратегии бизнеса. Общие принцип можно определить, как: «бизнес описывает свои требования и ожидания (Бизнес требования), ИТ в свою очередь формирует техническое задание для достижения целей.»

Наладить сотрудничество между бизнесом и ИТ.

Из выше сказанного вытекает вторая важная проблема – «вакуум» в коммуникации между сотрудниками бизнеса и ИТ. Задача руководства организации, и ИТ директора в том числе, наладить общение между сотрудниками организации не только на высшем уровне, но и между сотрудниками среднего уровня, и непосредственных исполнителей бизнеса и ИТ. Как говорится в популярной фразе «дьявол кроется в мелочах». Любая классная идея в общих чертах должна быть спущена вниз для детальной проработки. На этом этапе особенности мышления ИТ специалистов с его жесткими связями «причина – следствие», вопросами «... что будет если ...», «... как контролировать...», «... как измерить...», анализ сценариев предельных состояний, поможет выработать оптимальное решение. Кроме этого такие вопросы помогут представителям бизнеса лучше понять и проработать решение, с точки зрения бизнеса. Понять, что можно требовать от ИТ, текущие возможности решения и их перспективные возможности или ограничения.

Получить максимальную ценность от ИТ.

Для большинства организаций, если только это не ИТ компания, ИТ является лишь инструментом достижения целей бизнеса, вторичным сервисом, таким же, как и бухгалтерия или администрация, обеспечивающих сопровождение основных направлений и процессов. ИТ оценивает решения с точки зрения технической зрелости, законченности, функциональности и т.п. В то время как бизнес интересуется лишь возможностью извлечения прибыли. Идеально проработанное техническое решение может свести на «нет» бизнес преимущества самой идеи, сделать ее сложной в эксплуатации пользователями, высокой стоимости как при внедрении, так и при сопровождении, что снижает финансовую выгоду, не удобную для клиентов и т.п. Задачи ИТ директора – разработка не самого «лучшего» решения с точки зрения ИТ, а наиболее «правильного». Наиболее «правильное» решение будет сформировано формулой и из следующих ключевых составляющих:

ЦЕННОСТЬ = ВЫГОДА – ЗВТРАТЫ

С точки зрения бизнеса: Получить максимальную ценность от ИТ. Ценность информационных технологий – это разница между выгодами использования информационных технологий и затратами на них).

С точки зрения ИТ: Соблюдение ИТ ценностей и требований (технологичность, безопасность и управляемость ИТ).

Перенос часть работ на сторону ИТ департамента, или отказ от автоматизации ряда элементов, может снизить стоимость решения, повысить простоту эксплуатации и удобство для клиентов.

Одна из первостепенных задач – определить границы, где можно искать возможности, а где та грань, переход которой ломает основы управляемости ИТ и информационной безопасности.

Управлять изменениями.

В контексте данной главы имеется ввиду готовность к изменениям инициаторами которых является бизнес. Обстановка в бизнесе может меняться быстро и радикально. Появление новых ниш для бизнеса, разработка новых продуктов, слияния и поглощения и т.п. Это может привести к тому, что техническое решение, используемое в организации, перестает удовлетворять требованиям организации. Задача ИТ – адаптация имеющегося решения или разработка нового в кратчайшие сроки и с минимальным бюджетом для соответствия требованиям бизнеса. Как следствие при разработке ИТ стратегии и ИТ решений в частности, необходимо закладывать определённую гибкость и универсальность.

Навести порядок и управлять развитием ИТ.

Современные реалии ведения бизнеса и развития технологий приводит к тому, что бизнес требует внедрения все новых и новых технологических решений. Различные методы и модели их внедрения (самостоятельная разработка, покупка «коробочного» решения, внедрение и сопровождение со стороны третьей стороны и т.п.) приводит к тому, что в ИТ инфраструктуре появляется большое количество различного аппаратного и программного обеспечения, частично дублирующего друг друга, устаревших решений, и т.п. Кроме этого, постоянная зависимость от специалистов, обладающих «уникальными» знаниями и опытом. Задача ИТ директора – организация управления ИТ, непрерывное обучение сотрудников новым технологиям, выбор перспективных направлений, которые могут принести выгоду бизнесу.

Базовые принципы построения Архитектуры Предприятия

Комплексная автоматизация функции управления требует создания единого информационного пространства на любом современном предприятии, в котором обычные сотрудники и руководство смогут осуществлять свою деятельность, руководствуясь едиными правилами доступа, представления и обработки информации. Современные методы построения ЕИП предприятия базируются на проведении комплексного реинжиниринга бизнес-процессов, на создании информационно-логической модели и последующей реализации соответствующего программного и информационного обеспечения с использованием новых технологий. Разработка ИТ-архитектуры позволяет ясно представить:

- Какая информация/данные критичны для бизнеса компании и как они организованы;
- Какие приложения будут поддерживать бизнес;
- Смогут ли эти приложения эффективно взаимодействовать между собой и с внешними системами партнеров и поставщиков;
- Соответствуют ли используемые технологии требованиям поддержки бизнес-процессов;
- Достаточно ли обеспечена информационная безопасность систем;
- Смогут ли сотрудники компании получить своевременный доступ к нужным данным из любого необходимого места;
- Какие стандарты должны использоваться при разработке и закупке компонент системы;

Формирование архитектуры производится с использованием распространенных методологий, рамочных моделей (frameworks) описания архитектуры и инструментальных средств моделирования (например, ARIS IT Architect) – с учетом имеющихся у Заказчика опыта и предпочтений. В ходе проекта формируются набор архитектурных принципов, определяются используемые и перспективные стандарты и разрабатываются модели архитектуры в целом и ее отдельных областей (приложения, данные, интеграция, техническая инфраструктура, безопасность и др.)

Примерный порядок построения единого информационного пространства предприятия следующий:

Обследование предприятия



- Цели и основные задачи;
- Четкое определение задач и целей проекта;
- Формулирование функционально-технических требований к проекту;
- Инфраструктурный аудит ИТ-системы предприятия для проектирования архитектуры решения;
- Формализация существующих на предприятии бизнес-процессов, автоматизируемых в рамках проекта;
- Получение данных для обоснования количества лицензий;
- Оценка ресурсов, которые потребуются для реализации проекта;

- Оценка рисков проекта;
- Разработка предварительного плана проекта, графика работ и спецификации поставок по каждому этапу и проекту в целом.

Результаты обследования:

- Отчет с фиксацией ситуации в виде «Как есть»;
- Предложение по созданию информационной системы в виде «Как необходимо»;
- Предложение по «Функционально-техническим требованиям к проекту»;
- План реализации проекта;
- Оценка рисков проекта и предложения по их минимизации;
- Ориентировочная стоимость пилотного проекта и конечного решения.

Поставка необходимых предприятию аппаратного обеспечения и функциональных решений по автоматизации подразделений:

- Адаптация и настройка базовых, при необходимости разработка новых решений в соответствии с функционально-техническими требованиями, полученными на этапе обследования;
- Развертывание платформы с базовыми сервисами обработки и хранения документов;
- Опытная эксплуатация решения;
- Интеграция с ERP-системой;
- Ввод системы в промышленную эксплуатацию.

Обучение технического персонала и пользователей.

Обеспечение дополнительного функционала – работы, выделенные на этапе обследования в отдельные этапы.

Архитектурный Фреймворк – это готовая методология и набор поддерживающих инструментов, которые адаптируются для использования в конкретной компании. В Фреймворке есть типовые архитектурные процессы, рекомендации по их адаптации для конкретной компании, рекомендации по формированию шаблонов архитектурных артефактов, требования к их заполнению, требования к архитекторам и многое другое.

Стратегия ИТ Архитектуры Предприятия		
Охват: все-организация		
Уровень детализации: поверхностный		
Архитектура сегмента 1	Архитектура сегмента 2	Архитектура сегмента N
Охват: Бизнес-подразделение, функция, направление	Охват: Бизнес-подразделение, функция, направление	Охват: Бизнес-подразделение, функция, направление
Уровень детализации: средний	Уровень детализации: средний	Уровень детализации: средний
Архитектура решение 1	Архитектура решение 2	Архитектура решение N
Охват: ИТ-решение, Проект, Бизнес-процесс, функция	Охват: ИТ-решение, Проект, Бизнес-процесс, функция	Охват: ИТ-решение, Проект, Бизнес-процесс, функция
Уровень детализации: подробный	Уровень детализации: подробный	Уровень детализации: подробный

Модель Архитектуры Предприятия можно разделить на несколько ключевых уровней (категорий):

- **Архитектура бизнеса** – содержит стратегию компании, подход к управлению, организационную структуру и ключевые бизнес процессы.
- **Архитектура информационных систем** – описывает, как устроены или будут устроены информационные системы компании. Архитектура информационных систем может быть разбита на две подгруппы:
 - **Архитектура приложений** – показывает бизнес приложения, развернутые в компании, их взаимодействие друг с другом, а также их связь с бизнес процессами компании

- *Архитектура данных* – содержит описание логической и физической структуры данных компании, а также подход и средства управления данными.

- *Техническая архитектура* – описывает программное обеспечение и оборудование, которое необходимо для развертывания бизнес сервисов, данные и приложения, ИТ инфраструктуру, сервера приложений, сети, телекоммуникации, стандарты и т. д.

Следующим уровнем иерархии может быть по:

- *Стратегическая архитектура* описывает всю компанию в целом. На этом уровне вы не погружаетесь в особенности конкретных подразделений, направлений бизнеса и т. д. Стратегическая архитектура сконцентрирована на общих для всей компании стратегии, бизнес процессах, инвестициях, данных, системах и технологиях. Она прежде всего ориентирована на реализацию стратегии компании. На этом уровне определяются принципы и приоритеты, создаются общие для всей компании ИТ сервисы.

- *Архитектура сегмента*. Это архитектура направления деятельности компании, программы проектов или отдельного подразделения. Таких сегментов у компании будет несколько. На этом уровне вы погружаетесь в специфические особенности и требования конкретного подразделения, функции или компании в составе организации. Прорабатываете бизнес кейсы использования ИТ с руководством подразделения. Архитектура сегмента должна иметь ту же структуру и общие сервисы, что и стратегическая архитектура.

- *Архитектура решения* – архитектура конкретного ИТ решения. Она используется для реализации новых или доработки существующих ИТ решений сервисов или их частей. В ней учтены как общие для всей компании требования, так и специфические потребности, выявленные на уровне архитектуры сегмента. Рамки архитектуры решения ограничены одним проектом, процессом или функцией. Архитектура решения прорабатывается максимально детально. Это позволяет избежать неприятных сюрпризов в будущем при реализации проекта.

Уровни зрелости ИТ

В процессе роста и развития организации, департамент ИТ также проходит несколько этапов своего развития. При построении Архитектуры Предприятия следует принимать во внимание уровень зрелости организации в целом, так и ИТ в частности. Можно выделить следующие этапы и симптомы состояния ИТ департамента:

Начальный или «локализация»

В компании быстро разрабатывают или закупают информационные системы, которые дают ценность отдельным бизнес подразделениям или функциям.

Основные симптомы данного этапа развития:

- Роль ИТ департамента – второстепенная
- Стратегия ИТ департамента – отсутствует
- Бюджет ИТ департамента – отсутствует
- Тип деятельности ИТ департамента – пассивный, только по запросу.
- Задачи менеджмента – максимальная экономия
- Место в организационной структуре компании – служба в составе бизнес департамента

(администрация, финансы)

Развития или «стандартизация»

Компания переходит от закрытия потребностей отдельных бизнес подразделений или функций и повышает эффективность ИТ за счет стандартизации технологий и инфраструктуры. Основные симптомы данного этапа развития:

- Роль ИТ департамента – не стратегический ресурс
- Стратегия ИТ департамента – отсутствует или на начальном уровне
- Бюджет ИТ департамента – отсутствует или управляемый со стороны
- Тип деятельности ИТ департамента – слабо активный по ИТ инфраструктуре и пассивный, только по запросу для бизнеса
- Задачи менеджмента – минимальные инвестиции

Место в организационной структуре компании – служба или отдел в составе бизнес департамента (администрация, финансы) или под курацией.

Становления или «оптимизация»

Компания строит сквозные бизнес процессы, используя общие данные и информационные системы. Централизует по возможности бизнес процессы и функции подразделений в централизованных информационных системах по предварительно описанных операционных моделях. Основные симптомы данного этапа развития:

- Роль ИТ департамента – важная
- Стратегия ИТ департамента – частично отвечает бизнес требованиям
- Бюджет ИТ департамента – существует и управляем
- Тип деятельности ИТ департамента – реактивная реакция на бизнес требования,
- Задачи менеджмента – получение бизнес выгоды в краткосрочной перспективе
- Место в организационной структуре компании – выделенный департамент, но ИТ менеджер не входит в состав совета директоров.

Зрелый

Повторное использование имеющихся бизнес процессов и компонентов для создания новых сервисов и возможностей. ИТ работает на опережение и является не отъемлющей частью бизнеса. Основные симптомы данного этапа развития:

- Роль ИТ департамента – стратегический ресурс организации
- Стратегия ИТ департамента – интегрированная в бизнес стратегию компании
- Бюджет ИТ департамента – существует и рассматривается как инвестиции в бизнес
- Тип деятельности ИТ департамента – активная реакция (превентивная) на бизнес требования
- Задачи менеджмента – долгосрочные инвестиции в бизнес
- Место в организационной структуре компании – департамент, ИТ директор в составе совета директоров

Каждый из этих этапов имеет плюсы и минусы. Этап развития ИТ должен соответствовать этапу развития компании. Опыт показывает, что перескочить хотя бы через один из этапов сложно и требуются колоссальные затраты ресурсов (времени, денег, энергии сотрудников и т. д.), при этом эффект будет обратный. Большинство компаний проходят их один за другим. Переход на следующую стадию переходит при осознании руководством компании или остроты возникших вопросов:

- Проблемы поддержки роста и изменений бизнеса
- Дублирование бизнес процессов
- Многообразие платформ и систем
- Неудовлетворенность текущем состоянием ИТ

Критерии выбора методологии

Так как методологии сильно отличаются друг от друга, то следует задать критерии для их сравнения.

Полнота таксономии, определяет, насколько методология пригодна для классификации различных архитектурных артефактов. Полностью сосредоточена на фреймворке Захмана.

Полнота процесса, определяет, насколько детально представлен процесс создания архитектуры предприятия.

Руководство по эталонным моделям, определяет полезность методологии в создании адекватного набора эталонных моделей. На этом практически полностью сосредоточена методология FEA.

Практическое руководство определяет, насколько методология позволяет воплотить в жизнь умозрительное представление об архитектуре предприятия и сформировать культуру, в которой эта архитектура будет использоваться. На этом практически полностью сосредоточена методология Gartner.

Модель готовности определяет, насколько методология позволяет оценить эффективность использования архитектуры предприятия в различных подразделениях.

Ориентированность на бизнес определяет, ориентирована ли методология на использование технологии для повышения ценности бизнеса, где ценность бизнеса определяется как снижение затрат или увеличение доходов.

Руководство по управлению определяет, насколько методология полезна в понимании и создании эффективной модели управления для архитектуры предприятия.

Руководство по разбиению определяет полезность методологии в эффективном разбиении предприятия на отделы, что весьма важно при управлении сложностью.

Наличие каталога определяет, насколько эффективно методология позволяет создать каталог архитектурных активов, которые можно будет использовать в дальнейшем.

Нейтральность по отношению к поставщикам услуг определяет вероятность того, что при внедрении методологии вы окажетесь привязанными к конкретной консалтинговой организации. Высокая оценка означает низкую степень привязки к конкретной организации.

Доступность информации определяет количество и качество бесплатных или относительно недорогих материалов по данной методологии.

ремя окупаемости инвестиций определяет продолжительность периода, в течение которого вы будете использовать данную методологию, прежде чем сможете построить на ее основе решения, обеспечивающие высокую ценность бизнеса.

Построение Архитектуры Предприятия может быть выполнено с применением различных методов и практик. В данной книге мы вкратце рассмотрим несколько и сфокусируем свое внимание на одной из них:

- «*Zahman Framework*» – наиболее ранняя и известная методология. Идеально подходит для «классификации» элементов архитектуры.

- «*TOGAF (The Open Group Architecture Framework)*». Методология получила широкую известность и распространение, во многом за счет открытости. Представляет из себя каркас «построения процессов».

- «*Gartner*» – методология экспертного анализа с использованием «лучших» практик.

- «*FEAF*» – методология построения архитектуры, использующая «сервис ориентированный» подход.

При построении ИТ Архитектуры Предприятия необходимо выделить следующие состояния архитектуры организации:

- *Текущее состояние «As-is» или «Baseline Architecture».*
- *Переходное состояние «Transition Architecture».*
- *Будущее состояние «To-be» или «Target Architecture».*
- *План перехода «Enterprise Architecture Management Plan & Roadmap»*

В общем случае Архитектура Предприятия представляет из себя план перехода от «Текущего» к «Будущему» состоянию организации. Архитектурный проект длится несколько лет и инициирует множество ИТ проектов. Эти проекты будут разной продолжительности, у них разные даты начала и окончания. Их нужно сгруппировать таким образом, чтобы изменения в бизнесе и ИТ происходили в нужное время, с минимальными рисками и без проблем с совместимостью. То есть архитектура может переходить из одного работоспособного состояния в другое несколько раз за время архитектурного проекта. Промежуточное состояния называют «переходной архитектурой» (Transition Architecture).

Архитектура Предприятия на основе методологии «ZAHMAN FRAMEWORKS»

Наиболее ранняя методология и на мой взгляд наиболее полная и структурированная. Изначально разрабатывалась как архитектура Информационных Систем. Основная идея заключается в том, чтобы обеспечить возможность последовательного описания каждого отдельного аспекта системы в координации со всеми остальными. Модель «Захмана» используется для описания предприятия в целом, так что предложенная модель, вообще говоря, может использоваться как средство для описания архитектур сложных производственных систем любого типа. Основные характеристики данной модели:

- целостность в отношении предприятия;
- обсуждений сложных вопросов с использованием относительно небольшого количества нетехнических понятий;
- применимость для решения задач, то есть возможность работать с абстракциями и сущностями, выделяя и изолируя отдельные параметры системы без потери восприятия предприятия как целого;
- простота описания

Модель Захмана имеет такие преимущества перед другими моделями:

- Отличается своей простотой понимания как техническими, так и нетехническими специалистами.
- Более детальными описаниями компонентов архитектуры
- Возможность применения для планирования, позволяющего лучше принимать решения.
- Наиболее наглядным представлением компонентов компании.
- Описанием сложной архитектуры небольшим количеством нетехнических понятий.
- Независимостью конкретных инструментов описания.
- Может использоваться как средство для описания архитектур сложных производственных систем любого типа.

3 Матрица Захмана						
	МОТИВЫ	ЛЮДИ	ГРАФИКИ	ДААНЬЕ	ФУНКЦИИ	СЕТЬ
Потребности цели	Конкуренты Товары	Партнеры	События			
Бизнесмодель	Бизнесплан					
Логическая (системная) модель АС	Бизнес-правила					
Тактическая архитектура ИС	Условия/действия					
Детальная реализация	TRIGGER ALARM	read string	on event t > t1 ...	CREATE TABLE	BEGIN BLOCK	C->PING
Практика использования		Умения			Мемо	Wait, please

Архитектура Предприятия – матрица «Zahman Framework»

Архитектура Предприятия на основе методологии «FEAF»

FEAF- фреймворк разработанный правительством США, как некий подход для развития информационных технологий правительственных учреждений, приведенный к использованию единой архитектуры.

В основе FEAF лежат пять эталонных моделей:

- Исполнительная модель.
- Бизнес-модель.
- Сервисная модель Компонента.
- Техническая эталонная модель.
- Эталонная модель данных.

Одно из полезных свойств фреймворка FEA – принцип сегментного подхода, дает возможность ускорить внедрение «Архитектуры предприятия». Процесс разработки архитектуры предприятия по методологии FEA включает в себя следующие этапы:

- Анализ Архитектуры
- Архитектурное определение
- Стратегия инвестиций и финансирования
- План управления программой и реализация проекта

Архитектура Предприятия на основе методологии «GARTNER»

Методология Gartner – эту модель можно описать как набор рекомендаций по созданию архитектуры предприятия.

Модель Gartner 2002 года сформулирована в виде четырех связанных, взаимозависимых и усложняющихся уровней:

- Среда бизнес-взаимодействия (Business Relationship Grid);
- Бизнес-процессы и стили бизнес-процессов;
- Шаблоны;
- Технологические строительные блоки (кирпичики – bricks).

Методология Gartner – по сути своей не является методологией, как например структурированная модель Захмана, ни процессом как TOGAF, ни как FEA. Gartner – является набором практических рекомендаций. Данная методология является сборником советов по построению архитектуры предприятия от одной из наиболее известных в мире консалтинговых ИТ-компаний – Gartner.

Данный фреймворк представляет собой трехмерный куб, состоящий из слоев:

- Горизонтальные слои.
- Вертикальные домены.
- Вертикальные элементы технической архитектуры.

Архитектура Предприятия на основе методологии «TOGAF»

Основным полем для применения TOGAF является, прежде всего, программная инфраструктура информационной системы. Она в наилучшей мере подходит для описания интеграционных компонент, использующихся для поддержки широкого спектра корпоративных приложений, прежде всего, критичных для бизнеса. Описание модели:

- Обзор бизнес – архитектуры
- Описание существующей системы с необходимой степенью детализации
- Выявление и описание элементарных архитектурных блоков – кандидатов на использование в новой архитектуре
- Разработка черновика технического отчета
- Направление черновика отчета на рецензирование

TOGAF позиционируется не как некоторая эталонная модель, а как «средство для разработки архитектур информационных систем». Основное назначение – ускорить и облегчить процесс разработки архитектуры конкретной организации, обеспечивая при этом возможность будущего развития. Рассмотрим построение Архитектуры Предприятия на основе методологии «TOGAF» более детально, который на мой взгляд имеет ряд преимуществ:

- Подход TOGAF позволяет построить весь архитектурный процесс – от запуска практики до результатов.
- TOGAF – это де-факто является стандартом. Имеется программа сертификации по TOGAF.
- TOGAF – абсолютно бесплатен. Множество открытых ресурсов, скачивайте и используйте.
- TOGAF содержит полный набор инструментов для создания и развития архитектурной практики в организации. Есть пошаговый процесс для разработки описания Архитектуры Предприятия и полный набор инструментов, шаблонов и т. д.
- TOGAF совместим с другими Фреймворками, например, с «Zahman Framework». Как архитектурный процесс модель TOGAF дополняет модель Захмана – которая, классифицируется как архитектурная таксономия. Захман показывает, как следует классифицировать артефакты. Модель TOGAF описывает процесс создания артефактов.

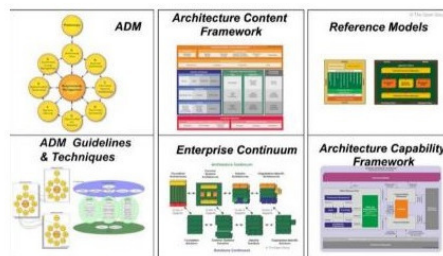


Рисунок: Структура Архитектуры Предприятия «TOGAF»

Методология TOGAF и инфраструктура Захмана хоть и объединены к категории «инфраструктур предприятия», но имеют отличия в своих принципах, структурах и компетенциях. TOGAF- представляет собой функциональную и динамичную инфраструктуру, которая включает руководящие принципы моделей процесса их использования. В то время как фреймворк Захмана представляет собой статичную структуру архитектуры, наиболее эффективна

для применения анализа и метаанализа фреймворков инфраструктур. Несмотря на значительные отличия данных фреймворков их можно использовать совместно.

Базовые принципы

Процесс создания конкретной архитектуры предприятия рассматривается как переход от общей архитектуры к специализированной. Методика разработки архитектуры в модели TOGAF представляет собой процесс осуществления такого перехода. В модели TOGAF наиболее обобщенные архитектуры называются фундаментальными архитектурами. Эти принципы построения архитектуры теоретически могут использоваться практически любой ИТ-организацией в мире.

Следующий уровень специализации в модели TOGAF называется общесистемными архитектурами. Эти принципы прослеживаются во многих – возможно, не во всех – типах предприятий. Далее идет отраслевой уровень архитектурой. Эти принципы характерны для предприятий, занятых в одной сфере деятельности. И наконец финальный уровень – это уровень архитектуры организации. Это самый высокий уровень специализации в модели TOGAF. Это архитектуры конкретных предприятий.

В состав модели TOGAF входят две основные компоненты:

- методика ADM (Architecture Development Method), определяющая процесс разработки архитектуры.

- Базовая Архитектура (Foundation Architecture). Она дополняется соответствующей базой данных ресурсов, включающей описания архитектурных принципов, примеров реализации, а также специализированный язык ADML.

Описание TOGAF включает в себя 7 частей:

- Introduction. Содержит высокоуровневое описание ключевых концепций Архитектуры в целом и TOGAF в частности.

- Architecture Development Method (ADM). Ключевая часть TOGAF, описывает пошаговую методику разработки Архитектуры Предприятия.

- ADM Guidelines and Techniques. Включает в себя описание правил и техник, которые используются в TOGAF ADM.

- Architecture Content Framework. Описывает подход к описанию Архитектуры Предприятия. Содержит метамодель архитектурных артефактов, структуру и описание типовых архитектурных артефактов.

- Enterprise Continuum & Tools. Описан подход к категоризации и хранению результатов архитектурных активностей.

- TOGAF Reference Models. Описание эталонных моделей, которые вы можете использовать в ваших проектах.

- Architecture Capability Framework. Подход к организации архитектурной практики в компании. Структура, процессы, роли, навыки и полномочия, требуемые для работы архитектурной практики в компании.

В качестве основных процессов построения Архитектуры Предприятия важно внедрить всего четыре ключевых процесса:

- Создания и развития Архитектуры Предприятия.
- Управления изменениями.
- Контроль реализации архитектурных решений.
- Управления практикой.

Метод Развития Архитектуры (Architecture Development Method ADM) TOGAF

В TOGAF процессы создания и развития, управления изменениями, контроля реализации архитектурных решений интегрированы в единый архитектурный цикл Метод Развития Архитектуры (Architecture Development Method ADM). Данный метод можно и нужно адаптировать под задачи вашей компании на всех уровнях разработки архитектуры. При этом, нет необходимости делать все возможные документы. Не нужно погружаться во все детали. На каждом этапе ADM предлагает готовый набор техник, инструментов, шаблонов и чек листов. Метод Развития Архитектуры (ADM) содержит десять фаз. Каждая фаза, в свою очередь разбивается на под-процессы (этапы), отдельные работы и так далее.

Например, фаза D включает следующие основные под-процессы:

Описание существующей технологической архитектуры.

- Обзор бизнес-архитектуры, архитектуры данных и приложений для определения начальных данных и необходимой степени детализации.

- Описание существующей системы с необходимой степенью детализации, которая выбирается для того, чтобы можно было выявить необходимые изменения при формировании целевой архитектуры. Формирование реестра используемых платформ программного и аппаратного обеспечения.

- Выявление и описание элементарных архитектурных блоков – кандидатов на использование в новой архитектуре. Фактически, речь идет о возможных архитектурных шаблонах.

- Разработка черновика технического отчета, резюмирующего основные результаты изучения существующего состояния и возможности использования типовых блоков.

- Направление черновика отчета на рецензирование, анализ комментариев и внесение, при необходимости, поправок.

Формирование целевой технологической архитектуры.

- Описание существующей системы в терминах TOGAF.

- Определение перспектив (представлений) архитектуры.

- Формирование модели целевой архитектуры.

- Определение ИТ-служб (сервисов).

- Подтверждение учета бизнес-требований.

- Определение архитектуры и используемых блоков (шаблонов).

- Проведение анализа расхождений (gap analysis).



Рисунок: «TOGAF» и Метод Развития Архитектуры (Architecture Development Method ADM).

Фазы и задачи Метода Развития Архитектуры

Предварительная фаза. Основные задачи:

- Создать архитектурную практику,
- подготовить компанию к запуску архитектурных проектов,
- заручиться поддержкой руководства,
- сформулировать архитектурные принципы,
- адаптировать методологию под цели и задачи компании

Фаза А – Видение архитектуры. Основные задачи:

- Запустить архитектурный проект, определить цели и задачи, определить рамки, предположения и ограничения проекта, разработать видение архитектуры, определить всех заинтересованных лиц,
- разработать «Устав проекта» и получить формальное подтверждение старта проекта.

Фаза В – Бизнес Архитектура. Основные задачи:

- Разработать архитектуру с описанием текущей и целевой архитектуры,
- Анализ расхождений

Фаза С – Архитектура Информационных Систем. Задачи:

- Разработать архитектуру с описанием текущей и целевой архитектуры,
- Анализ расхождений

Фаза D – Техническая Архитектура. Основные задачи:

- Разработать архитектуру с описанием текущей и целевой архитектуры,
- Анализ расхождений

Фаза E – Возможности и решения. Основные задачи:

- Выполнить начальное планирование реализации задач проекта.
- Идентифицировать основные проекты внедрения и сгруппировать их в переходные архитектуры.

Фаза F – Планирование миграции. Основные задачи:

- Анализ затрат и рисков.
- Разработка детального плана внедрения и миграции.

Фаза G – Управление реализацией. Основные задачи:

- Архитектурный надзор за проектами внедрения.
- Подготовить архитектурные контракты.
- Обеспечить соответствие архитектуре результатов проектов внедрения.

Фаза H – Управление изменениями архитектуры. Задачи:

- Подготовиться к следующему витку жизненного цикла архитектуры.
- Процесс управления изменениями должен обеспечить соответствие архитектуры актуальным потребностям бизнеса и дать максимальную ценность бизнесу.

Управление требованиями. Основные задачи:

- На каждой фазе архитектурного проекта собираете и согласуете бизнес требования.

- Требования должны быть идентифицированы, сохранены, определены приоритеты и использованы на соответствующих фазах архитектурного проекта.

Спецификация TOGAF также позволяет гибко работать с этапами. В самой спецификации говорится следующее:

- Перед применением методики разработки архитектуры необходимо проверить компоненты на применимость, а затем связать их с конкретными обстоятельствами отдельного предприятия. Это позволяет создать методику разработки архитектуры для конкретного предприятия.

- Модель TOGAF позволяет выполнять этапы частично, пропускать их, объединять, изменять порядок и вносить изменения в соответствии с конкретными требованиями. Неудивительно, что два сертифицированных консультанта по TOGAF могут разработать два совершенно различных процесса – даже при работе с одной и той же организацией.

- Модель TOGAF обладает еще большей гибкостью в отношении созданной архитектуры. Фактически TOGAF, как это ни удивительно, «ничего не знает» об архитектуре. Окончательная архитектура может с одинаковым успехом быть хорошей, плохой или неопределенного качества. В TOGAF описывается, как создать архитектуру предприятия, но не описывается, как создать хорошую архитектуру. Качество конечного продукта зависит от опыта персонала компании и консультанта по TOGAF. Те, кто внедряет TOGAF в надежде получить чудодейственное средство, будут жестоко разочарованы (впрочем, как и при использовании любой одной методологии).

Базовая Архитектура (Foundation Architecture).

Базовая Архитектура, включает в себя:

- набор наиболее общих служб и функций, объединенных в Техническую Эталонную Модель (Technical reference model – TRM);
- набор элементарных архитектурных элементов, которые используются как «строительные блоки» при построении конкретных решений;
- база данных стандартов (Standards Information Base).

Концепция использования Базовой архитектуры определяется в соответствии с иерархией архитектур, входящих в общий континуум определений. В TOGAF техническая эталонная модель рекомендуется к использованию, но не является обязательной. В общем техническая эталонная модель, не лишена недостатков по следующей причине: она направлена на обеспечение переносимости приложений в ущерб их способности к взаимодействию и автономности. Для организаций модель TOGAF в значительной степени сводится к методике разработки архитектуры. В этом смысле компонента Базовой Архитектуры, содержащая набор служб и стандартов, является некоторой абстрактной реализацией ИТ-системы в целом. Архитектура Общих Систем реализуется путем выбора и интеграции определенных служб для формирования выделенных блоков, которые могут (возможно, повторно или в различных комбинациях) использоваться в различных функциональных областях, таких как архитектура безопасности, сетевая архитектура и т. п.

Следующая степень детализации реализуется на уровне Отраслевой Архитектуры, которая добавляет специфичные для каждой индустрии модели данных, приложения, стандарты, бизнес-правила, а также, при необходимости, процедуры взаимодействия различных отраслевых систем между собой. Наконец, на последнем уровне Архитектуры Организации формируется архитектура ИТ-систем конкретного предприятия, учитывающая все его особенности,

в том числе наличие унаследованных систем, планы и возможности реализации, организацию данных на физическом уровне и т. п.

В состав Эталонной Модели, в свою очередь, входит система (таксономия) общих служб, включающая такие службы, как Обмен и преобразование данных, Управление данными, Поддержка интернационализации, Службы Каталогов и т. п.

Для всех используемых в архитектуре служб, наряду с функциональным назначением, необходимо определить и уровень качества реализации, то есть такие характеристики как управляемость, гибкость, гарантированность, удобство использования и т. п. При этом следует учитывать, что некоторые службы являются в этом плане взаимозависимыми. Например, для обеспечения заданного качества службы интернационализации могут потребоваться специализированные компоненты службы разработки программного обеспечения для создания и тестирования соответствующих программных продуктов.

Архитектурные принципы представляют собой как бы фундаментальные «аксиомы», которые используются в качестве «отправных точек» как для оценки существующей системы, так и для разработки отдельных архитектурных решений. Вообще говоря, архитектурные принципы являются подмножеством более общего понятия ИТ-принципов, которые определяют основные

аспекты всей деятельности, связанной с применением информационных технологий. ИТ-принципы, в свою очередь, являются детализацией еще «более общих» принципов, определяющих деятельность предприятия в целом.

В состав набора принципов могут входить обоснования для формирования системы требований или критериев оценки тех или иных решений. Например, такой принцип, как «минимизация числа поставщиков программного обеспечения», может быть в дальнейшем конкретизирован в зависимости от особенностей предприятия, как требование «единой СУБД для всех критичных для бизнеса приложений» или же как «использование той же СУБД, что и уже применяемая». Архитектурные принципы могут также использоваться для обоснования значимости самого понятия Архитектуры и необходимости ее разработки для бизнеса предприятия, а также для выбора вариантов реализации этого процесса.

Принципы являются взаимозависимыми и должны применяться в целостном наборе. «Хороший» набор принципов должен удовлетворять таким естественным критериям, как доступность для понимания, точность формулировок, полнота, последовательность и стабильность (не нужно путать с неизменяемостью!) Обычно число принципов не превышает 20, чтобы не ограничивать гибкость архитектуры или чтобы избежать чисто формального определения принципов, которые неработоспособны на практике.

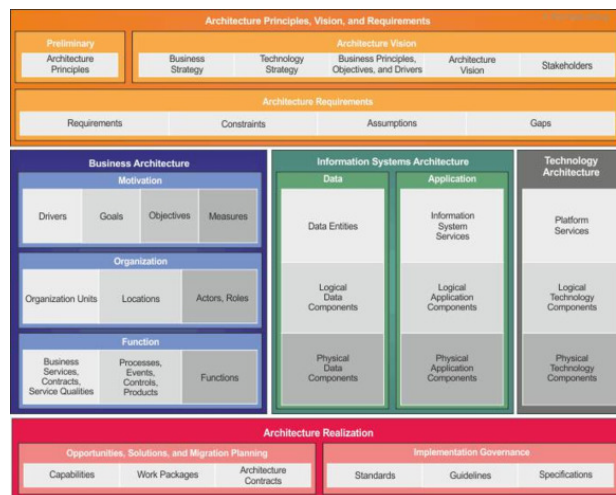


Рисунок: Архитектуры Предприятия «TOGAF»

Примеры принципов, используемых при создании архитектуры TOGAF (Название и содержание):

Пример использования – Сформулированные принципы управления ИТ применимы для всех случаев и подразделений организации.

Максимальная польза – Решения в области ИТ принимаются исходя из максимума пользы для организации в целом.

Привлечение всех – Управление информацией есть дело каждого.

Непрерывность бизнеса – Деятельность предприятия должна обеспечиваться, несмотря на возможные помехи в работе ИТ.

Общее использование – Предпочтение должно отдаваться разработке или внедрению приложений, применимых в масштабах всего предприятия, а не отдельных его подразделений.

Соответствие законодательству – Управление ИТ не должно противоречить применяемому законодательству и принятым регламентам, однако это не есть препятствие к улучшению бизнес-процессов, влекущему за собой изменение этих регламентов.

Ответственность ИТ-службы – ИТ-служба является ответственным владельцем ИТ-ресурсов и исполнителем процессов для удовлетворения требований бизнеса.

Защита интеллектуальной собственности – Обеспечение защиты интеллектуальной собственности организации должно быть реализовано на уровне архитектуры, процессов эксплуатации и управления ИТ.

Данные являются активом – Данные в ИТ-системе предприятия имеют определенную ценность и должны соответственно управляться, быть общими и доступными для пользователей с учетом их прав доступа.

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.

Технологическая независимость – Прикладное ПО не должно зависеть от специфичных моделей оборудования и системного ПО.

Простота использования – Приложения позволяют сконцентрироваться на выполнении бизнес-задач за счет единого интерфейса, минимизации специфики работы, интеграции систем, снижения вероятности неправильного использования.

Обоснованность и своевременность изменений – Изменения в информационной системе и приложениях производятся только в соответствии с запросами бизнеса, но в случае появления такой необходимости – в нужное время.

Взаимодействие – Взаимодействие Компоненты программного и аппаратного обеспечения должны обеспечивать интеграцию между собой в соответствии с общими стандартами.

Минимизация разнообразия – Уменьшение числа различных вариантов применяемых платформ, продуктов и версий.

Процесс Управления Архитектурной Практикой

Архитектурная практика представляет из себя практику внедрения архитектурного проекта в организации. Архитектурная практика состоит из четырех ключевых элементов:

•*Люди.* Они основа любой деятельности в компании. Если люди не знают, не умеют, не используют, не участвуют, не делают, не хотят, то все остальное бесполезно. Забыли про людей – забудьте про результаты.

•*Артефакты.* В процессе работы люди должны достигать заранее определенных результатов. Также они создают артефакты, которые позволяют обмениваться информацией, обсуждать задачи и проблемы, сохраняя идеи для последующего воплощения, контролировать достижение результатов и т. д.

•*Процессы.* Для достижения результатов люди должны делать правильные действия в правильной последовательности. Все люди ошибаются, но если они следуют заранее определенным процессам, то вероятность ошибок снижается, а их последствия удастся быстро устранить. Процессы помогают превратить хорошие идеи в результаты. Так что от них никуда не уйдешь.

•*Управление.* Без правильного управления архитектурная практика обречена на провал. Требуется заранее определить рамки и правила практики, взяв за основу стандартные процессы, артефакты, роли и т. д. Придумывать правила по ходу игры очень опасно. Люди будут дезориентированы, процессы будут сбоить, результаты будут не те и не тогда, когда нужно.



Рисунок: Управление Архитектурной Практикой «TOGAF»

Процесс управления архитектурной практикой необходим для того, чтобы исключить обычные ошибки, совершаемые людьми в процессе работы над задачей или проектом. Люди чаще всего не достигают результатов, если:

- залезают в дебри теорий и исследований;
- выполняют бесполезную работу;
- долго готовятся к выполнению работы;
- изобретают велосипед;
- «оптимизируют» свою работу вместо её выполнения;
- излишне теоретизируют;
- ищут ответственных и виноватых;
- считают себя самыми умными;
- избегают неприятной работы.

Подход к управлению архитектурной практикой состоит из шести основных элементов: *Методология* — это основной элемент подхода. Он определяет процессы компании для разработки, обновления и реализации Архитектуры Предприятия. Роли и их обязанности.

Артефакты — набор, шаблоны и правила заполнения документов, таблиц, схем, при помощи которых описана Архитектура Предприятия.

Стандарты — это стандарты (законы, правила) ведения бизнеса и ИТ, которые использует компания в своей работе. Это могут быть международные стандарты, российские стандарты, стандарты отрасли, региона, компании.

Лучшие практики и готовые модели — доказанные способы реализации решений, проверенные в вашей или в других компаниях.

Регламенты и правила – документы, в которых описаны цели, задачи, организационная структура, правила работы и границы архитектурной практики. Правила работы с другими подразделениями. Полномочия архитекторов. Регламенты должны быть интегрированы с другими регламентами компании, особенно ИТ департамента.

Управленческие воздействия со стороны менеджеров практики. Они направлены на то, чтобы компания получила практические результаты. Нужно планировать, заставить людей следовать процессам, стартовать архитектурные проекты, решать конфликты, контролировать промежуточные результаты и т. д. Все прочие элементы не будут работать без управления.

Порядок внедрения архитектурной практики. Во-первых, разработать все эти элементы для компании с нуля – это неподъемная задача. Поэтому для её решения нужно взять уже созданные методологии и адаптировать их к нуждам компании. Во-вторых, внедряйте их постепенно, как часть развития практики. Внедрение каждого элемента должно давать ценность практике. В-третьих, соблюдайте баланс между бюрократией и личной инициативой. И последнее – экспериментируйте. Проверяйте новые подходы. Если они дают ценность, опишите их в регламенте и используйте в ваших следующих проектах.

Ключевые документы и шаблоны методологии TOGAF

Для внедрения методологии TOGAF возникает вопрос какой минимальный набор документов необходим для ведения архитектурного проекта? Лишние документы – лишние затраты времени и денег. По исходя из собственного опыта и теории (при подготовке к сертификации), минимальный «джентельменский набор» может состоять из восьми документов:

Основы методологии (Templates, Business Principles, Goals, Drivers). Необходим для понимания миссию, цели, стратегию компании. Зафиксировать бизнес принципы. Шаблон «TOGAF 9 Templates, Business Principles, Goals, Drivers.doc»

Архитектурные принципы (Architecture Principles) – это правила, которыми руководствуются в работе над архитектурой. На их основе принимают архитектурные решения. Принципы нужно сформулировать на основе примеров из TOGAF. Использование принципов при работе над архитектурой доказало свою эффективность. Шаблон – «TOGAF 9 Template Architecture Principles.doc»

Видение архитектуры (Architecture Vision) – это высокоуровневое описание желательного конечного продукта архитектурного проекта. То есть это те результаты, которых нужно достичь. Описание решения тех проблем и задач, ради которых стартуют проект. Этот документ важен для взаимодействия со спонсором проекта и другими заинтересованными лицами. Шаблон – «TOGAF 9 Template Architecture Vision.doc»

Устав проекта (Statement of Architecture Work) – соглашение между спонсором и проектной командой о выполнении работ. В него включают все рамки, ограничения, предположения, сроки, бюджет, правила проекта. В нем конкретно назначают менеджера проекта и прописывают его права и обязанности. Сюда же включают как приложение видение архитектуры как описание рамок проекта. Шаблон – «TOGAF 9 Template Statement of Architecture Work.doc»

Описание архитектуры (Architecture Definition) – это представление текущей и целевой архитектуры. Он охватывает каждый из архитектурных доменов (бизнес, данные, приложения, технологии). А также анализ расхождений между текущим и будущим состоянием. Шаблон – «TOGAF 9 Template Architecture Definition.doc»

Спецификация требований к архитектуре (Architecture Requirements Specification) – документ, в котором собраны все требования, ограничения, предположения, критерии достижения. Шаблон – «TOGAF 9 Template Architecture Requirements Specification.doc»

Переходная архитектура (Transition Architecture) – Реализация целевой архитектуры проходит в несколько этапов. Каждое промежуточное состояние должно быть работоспособно и давать компании большую ценность. В этом документе сгруппированы проекты по каждому из таких этапов. Шаблон – «TOGAF 9 Template Transition Architecture.doc»

План реализации и миграции (Implementation and Migration Plan) – это сводный план реализации проектов, направленных на достижение целевой архитектуры. В него также включают выгоды, рамки, сроки, стоимость, риски, контрольные точки проектов. Шаблон – «TOGAF 9 Template Implementation and Migration Plan.doc»

Такой набор документов можно сделать максимально быстро и без больших затрат. Если предполагается воспользоваться услугами третьих сторон, то рекомендую включить ещё один документ – архитектурный контракт. Это соглашение между архитекторами и исполнителями ИТ проекта. Шаблон – «TOGAF 9 Template – Architecture Contract.doc» При приемке проекта вам будет легче получить нужный результат, если вы о нем заранее договорились.

Подходы к построению ИТ Стратегии

Подытоживая выше сказанное попробуем применить на практике полученный теоретические знания.

Первый шаг – оцениваем текущее состояние Архитектуры Предприятия (Baseline Architecture). Концентрируем внимание на вопросах бизнеса (Business Architecture) и начинаем с общих высокоуровневых вопросов. На данном этапе желательно провести интервью или просто побеседовать с руководством компании, топ менеджерами. Для данной задачи хорошо бы было участие двух специалистов уровня экспертов. Причем один в области управления, консалтинга или аудита, с хорошими аналитическими способностями и пониманием бизнес составляющей (ИТ Директор, Chief Information Officer CIO), а второй, эксперт в области информационных технологий (ИТ Архитектор, Chief Technology Officer CTO). Их главная задача – слушать. Ключевые вопросы, которые помогут удерживать направление беседы в бизнес сфере, сводятся к определению следующих составляющих:

- Основной вид деятельности организации
- Организационная структура
- Особенности ведения бизнеса
- Миссия и видения организации
- Цели и задачи
- Текущие проблемы организации
- Организация и оценка текущего состояние ИТ по мнению руководства

Следующий шаг – формируем видение Целевой Архитектуры Предприятия (Target Architecture). Для этого, задавая наводящие вопросы, пытаемся определить видение руководства, цели бизнеса, ожидания и чаяния. Видение будущего ИТ, его роли, вовлеченность в бизнес и т.п. Важный момент данного этапа общения – Не прерывайте их!!! Дайте им высказаться, помечтать. Даже если их фантазии будут время от времени противоречить друг другу, и здравому смыслу. Ваша задача – собрать как можно больше информации. Руководство организации является заказчиком. А как говорится: «... кто музыку заказывает, тот ее и танцует...». Запомните, а затем запишите и проанализируйте всю информацию.

По окончанию первых двух шагов, у нас будет воздушный замок высокоуровневой «Архитектуры Бизнеса» для «Текущей» и «Целевой» Архитектуры Предприятия конкретной организации:

Теперь повторяем те же шаги, но общаясь с руководством среднего звена, экспертами и ИТ компании. На данном этапе можно и нужно углубиться в детали, активно задавать вопросы, понять проблемы, рекомендации и опыт сотрудников. Как правило, можно подчерпнуть ценную информацию, понять ограничения и причины проблем, совместно набросать планы решения проблем. В результате получаем практически полную картинку:

Текущая Архитектура

- Архитектура Бизнеса
- Архитектура сегментов
- Информационные Системы
- Техническая Архитектура

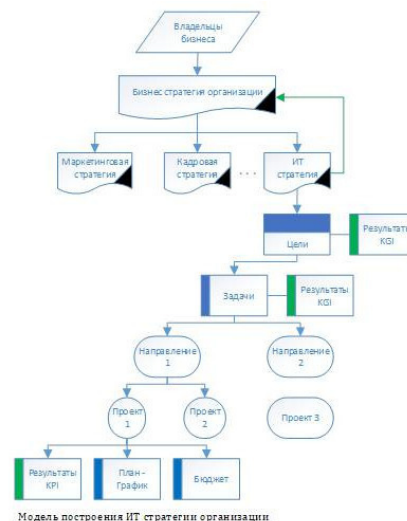
План Перехода

- Наброски и идеи

Целевая Архитектура

- Архитектура Бизнеса
- Архитектура сегментов
- Информационные Системы
- Техническая Архитектура

Процесс итерации можно повторять бесконечное число раз пока не добьетесь 100% полноты данных или забудете зачем вам это нужно. Полнота картины и количество циклов зависит от навыков и уровня экспертизы консультантов с одной стороны, и общение с правильными сотрудниками, с другой стороны. В конечном итоге мы формируем необходимый нам план перехода.



В настоящее время во многих компаниях, особенно крупных, внедрены формализованные процессы стратегического менеджмента. Согласно классике стратегического управления, стратегии делятся на 3 уровня:

- корпоративная стратегия (на уровне корпорации, холдинга),
- бизнес-стратегия (на уровне отдельной бизнес-единицы)
- функциональные стратегии (на уровне отдельных функциональных направлений в бизнес-единице).

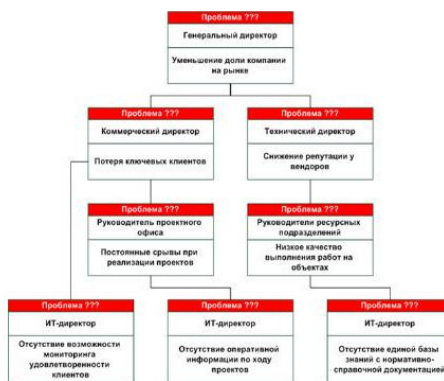


Диаграмма: Модель дерева проблем – решений» – Проблемы

В рамках этой классификации ИТ-стратегия является одной из функциональных стратегий, наряду, допустим, с финансовой стратегией или маркетинговой стратегией.

Как и любая другая стратегия, ИТ-стратегия по сути представляет из себя совокупность некоторого целевого видения будущей архитектуры ИТ и укрупненного описания направления движения к этой целевой архитектуре.

Как и любая другая функциональная стратегия, ИТ-стратегия направлена на достижение целей, обозначенных в бизнес-стратегии организации. Для удобства и наглядности соответствия планов развития ИТ планам развития бизнеса в первой части ИТ-стратегии формулируются цели для ИТ в привязке к целям, сформулированным в бизнес-стратегии.

С точки зрения объема и наполнения документа «ИТ-стратегия», то здесь все зависит от размеров организации и личных предпочтений ИТ-директора. Некоторые руководители предпочитают описать стратегию в виде общих лозунгов на несколько печатных страниц, другие считают, что необходима более глубокая проработка. Однако, практически все они сходятся во мнении, что сам по себе такой документ должен быть на любом более-менее крупном предприятии. Более того, это должен быть «живой» документ, который должен корректироваться при изменении внешней и внутренней среды организации, бизнес-целей для того, чтобы в каждый момент времени отражать актуальный вектор развития ИТ.

Можно предложить следующие информационные разделы для ИТ-стратегии:

- Цели для ИТ в привязке к бизнес-целям организации.
- Направления развития ИТ для достижения обозначенных целей.
- Проекты, которые должны быть реализованы в рамках каждого направления.
- Каждый реализуемый проект, в свою очередь, характеризуется определенным набором целей.
- Основные этапы по каждому проекту (краткое описание результатов, сроки, стоимость).
- Набор КРІ для мониторинга развития ИТ и достижения соответствующих целей.
- Бюджеты ИТ-проектов, направлений и общий бюджет ИТ.

Не смотря на очевидную пользу стратегии в области ИТ, на многих даже достаточно крупных предприятиях стратегический процесс в области ИТ отсутствует. В этом случае может быть целесообразно привлечение Консультанта для помощи в составлении ИТ-стратегии и внедрении процесса ее актуализации.

Построении ИТ Архитектуры и ИТ Стратегии в организации можно графически представить в виде диаграммы.

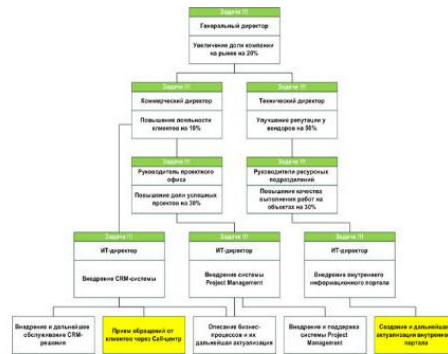


Диаграмма: Модель дерева «проблем – решений» – Решение

Для понимания проблем бизнеса и перевод их на язык ИТ можно воспользоваться методикой «дерево проблем – решений». Как видно из диаграммы, бизнес формулирует свои проблемы на языке бизнеса. Задача ИТ директора перевести язык бизнеса на технический язык, для дальнейшего формирования ИТ проектов и постановки задач сотрудникам ИТ департамента. Для того, чтобы бизнес и ИТ понимали, что необходимо требовать друг от друга, необходимо определить критерии достижения целей и задач, а также метрики их измерения.

При формировании ИТ Стратегии и определении роли ИТ в структуре организации необходимо ответить на ряд ключевых вопросов:

Для чего необходим департамент ИТ (миссия департамента). Миссия ИТ должна отвечать следующим параметрам:

- Соответствие миссии организации
- Соответствовать внутренним и внешним бизнес требованиям организации
- Будет ли актуальна в долгосрочной перспективе
- Стратегия централизации информационных систем (одна для всех задач, различные для разных задач)
- Стратегия централизации данных (функциональный, процессный или приложения)
- Стратегия централизации технологической архитектуры (платформа, резервирование и т.п.)

Что хотим достигнуть (цели). Основные требования к целям компании:

- Отражать специфику работы ИТ департамента
- Реалистичными
- Измеряемыми

Какие пути выбрать для достижения цели (стратегия). Под стратегией можно предполагать «выбор путей для создания конкурентного преимущества и достижения определённых показателей организации на рынке». Можно рассматривать два основных направления ИТ стратегии или их комбинацию:

- Нарращивание производства или продаж за счет инноваций в области ИТ
- Сокращение издержек за счет оптимизации и реинжиниринга с привлечением ИТ

Каким образом реализовать (проекты). Для эффективной реализации проектов, необходимо, чтобы соблюдались минимальные требования:

- Список проектов (операционный план) должен быть утвержден на уровне совета директоров организации
- Выделен необходимый бюджет, ресурсы и возможности

- Для каждого проекта рассчитаны преимущества (финансовые показатели, ROI, выгоды, риски и т п)
- Проводится мониторинг состояния операционного плана (проектов)

Каким образом измерять (ключевые показатели).

Кто реализует (сотрудники или аутсорсинг)

Стратегия Управления

При формировании ИТ Стратегии можно определить стратегию по вопросам «Технической Архитектуры» и «Архитектура Информационных Систем (сервисов)» ряда компонентов ИТ инфраструктуры, а также рекомендации по выбору того или иного решения.

Стратегия финансового контроля

Стратегия финансирования может предполагать поведение организации в вопросах финансирования ИТ. В частности, желательно иметь финансовую стратегию по таким вопросам как:

При выборе решения по ИТ проекту, как приоритет руководствоваться стоимостью решения, даже если это идет в ущерб функциональности. Возможности решения должны соответствовать требованиям бизнеса с минимальной стоимостью.

Ориентироваться на использование свободно распространяемого программного обеспечения

Выравнивание по стоимости или функционалу. Например, стоимость аренды каналов связи для разных филиалов банка может отличаться в зависимости от расстояния или географического расположения. Предположим, что стоимость 10 Mbps канала (стандартные и достаточные ИТ требования) для регионального филиала обходится 500 долларов. За ту же стоимость городской филиал может получить скорость канала до 100 Mbps. На текущий момент у этого филиала стандартная скорость 10 Mbps, но обходится за 100 долларов в месяц. Средняя стоимость канала связи для всех филиалов порядка 250 долларов. Руководство должно иметь определенную стратегию по данному вопросу: предоставить возможность филиалам наращивать скорость в пределах средней стоимости, максимальной стоимости или выравнивание всех по установленной скорости канала (10 Mbps).

Использование по возможности, ресурсы аутсорсинга при внедрении и сопровождении Информационных Систем и проектов или наращивать возможности и потенциал сотрудников ИТ департамента.

Стратегия административных регуляторов

Стратегия изменений может предполагать поведение организации в вопросах реакции на изменения требований и т.п. Как пример рассмотрим ситуацию, когда количество сотрудников остается не измененным, но скорость доступа в интернет значительно упала по причине активного использования. Как решение можно пойти по пути увеличения скорости канала и соответственно увеличением расходов. И так с определенной периодичностью данный вопрос будет возникать пока не дойдет до критической точки. Можно выбрать альтернативный «репрессивный» метод. ИТ департамент проведя анализ сетевого трафика определил, что вырос процент нецелевого использования интернета. Проведя ряд показательных казней руководство компании может добиться улучшения скорости использования интернета без увеличения стоимости используя административные рычаги управления.

Уровни управления	Название	Действия	Степень детализации
Стратегический	Стратегия ИТ	Вопрос «Что делать?»	Низкая
Тактический	Архитектура ИТ	Вопрос «Как делать?»	Средняя
Операционный	ИТ-процессы ИТ-проекты	Выполнить!	Высокая

Диаграмма: Уровни управления, задачи и детализации

Стратегия Компонентов ИТ Архитектуры

Стратегия выбора платформа решения

В качестве платформы решения могут быть рассмотрены следующие решения:

«*On-premises*» – ИТ активы физически располагаются на территории организации.

Преимущества:

•ИТ инфраструктура располагается на территории организации и контролируются ИТ сотрудниками организации.

•Относительно низкой стоимостью сопровождения (ОРЕХ) ИТ инфраструктуры.

•Автономность решений и более высокий уровень безопасности

Недостатки:

•Относительно высокие первоначальные вложения (САРЕХ) в ИТ инфраструктуру.

•Внедрение новых сервисов, или резкие скачки (роста) имеющихся сервисов трудно поддается планированию.

•Необходимость иметь в составе ИТ специалистов по поддержанию инфраструктуры (ремонт и обслуживание физических серверов, сетевого оборудования и т п). Все это ведет к дополнительным расходам.

•Косвенные расходы на инженерные системы ИТ инфраструктуры.

«*Cloud based*» – ИТ активы располагаются в «облаке».

Преимущества:

•Относительно низкие первоначальные вложения (САРЕХ) в ИТ инфраструктуру.

•Высокий уровень планирования расходов на сопровождение ИТ инфраструктуры.

•Хорошая связь, линейная зависимость используемых ресурсов и их стоимости.

•Простота внедрения новых решений и расширения имеющихся ИТ сервисов.

•Нет необходимости в дополнительных сотрудниках.

•Нет необходимости в косвенных расходах на системы инженерные.

Недостатки:

•Относительно высокая стоимость сопровождения (ОРЕХ) ИТ инфраструктуры.

•Более высокие требования к каналам связи интернета и наличию резервных каналов.

«*Hybrid*» – комбинированное решения. Используются преимущества двух первых решений.

Преимущества:

•Использует достоинства обоих решений.

Недостатки:

•Более высокая стоимость внедрения (САРЕХ) и сопровождения (ОРЕХ)

Рекомендации по выбору:

Для начальных проектов, небольших организаций, организаций с развитой географией и т п предпочтительно использование «Cloud based» решения. Для крепкой организации, финансовых институтов, организаций где выход в интернет не является требованием бизнеса

и т.п. предпочтительно использование «On-premises» решения. «Hybrid» решения могут как дополнять ИТ инфраструктуру, так и заменять часть компонентов ИТ архитектуры.

Стратегия выбора платформа развертывания

В качестве платформы для развертывания инфраструктуры, при условии, что выбрано «On-premises» решение, могут быть рассмотрены следующие варианты:

«**Физические сервера**» – ИТ сервисы располагаются на физических серверах.

Преимущества:

- Относительно низкой стоимостью внедрения (CAPEX) ИТ сервисов для малых решений.

- Ресурсы физического сервера полностью выделены по задачи конкретного сервиса.

Недостатки:

- Сложность сопровождения, с ростом инфраструктуры.

- Скорость развёртывания и восстановления.

- Не оптимальное использование вычислительных ресурсов.

«**Платформа виртуализации**» – ИТ сервисы располагаются на платформе виртуализации в виде виртуальных машин.

Преимущества:

- Относительно низкая стоимость сопровождения (OPEX) в ИТ инфраструктуру.

- Фактически является «де-факто» стандартов развертывания «On-premises» решений.

Недостатки:

- Относительно высокая стоимость внедрения (CAPEX) ИТ инфраструктуры.

Рекомендации по выбору:

Для небольших, отдельно стоящих или удаленных ИТ сервисов, или же высоконагруженных систем предпочтительно использование «физических серверов» решения. Для всех прочих случаев, использование платформы виртуализации предпочтительнее.

Стратегия выбора аппаратного и программного обеспечения

Стратегия выбора программного и аппаратного обеспечения определяет подход к выбору производителя, стандартизации и т.п. В качестве вариантов, могут быть рассмотрены следующие варианты:

Использование определённого производителя для каждой категории ИТ активов. Использование стандартов аппаратного и программного обеспечения в организации.

Преимущества:

- Внедрение стандартов ИТ активов упрощает процесс обеспечения сотрудников организации ИТ активами.

- Облегчает внедрение и сопровождение ИТ инфраструктуры

- Повышает уровень информационной безопасности организации.

Недостатки:

- Относительно высокая стоимость и зависимость от производителя и/или поставщика.

Использование произвольного производителя. Использование рекомендаций вместо стандартов для аппаратного и программного обеспечения в организации.

Преимущества:

- Относительно низкая стоимость и быстрота приобретения ИТ активов.

Недостатки:

- Процесс обеспечения сотрудников организации ИТ активами сложный и более долгий.
- Усложняет внедрение и сопровождение ИТ инфраструктуры
- Снижает уровень информационной безопасности организации.

Рекомендации по выбору:

Для организаций имеющих тесную интеграцию и зависимость бизнеса и Информационных Технологий или большую численность сотрудников рекомендуется использование первого варианта лицензирования.

Стратегия лицензирования

Стратегия лицензирования определяет подход к методам лицензирования. В качестве вариантов, могут быть рассмотрены следующие варианты:

Использование лицензионного соглашения уровня «Предприятия» с ежегодным продлением возможности обновления программного обеспечения. Лицензирование является непрерывным процессом в ИТ.

Преимущества:

- Большая степень свободы в вопросах лицензирования.
- Поддержка ИТ инфраструктуры и информационной безопасности на высоком уровне за счет использование более современных версий систем и решений.
- Обновление систем происходит плавно, без всплесков требований в ИТ ресурсах, людях и времени

Недостатки:

- Относительно высокая стоимость.

Покупка «коробочных» лицензий без продления обновления программного обеспечения. Лицензирование «по требованию».

Преимущества:

- Относительно дешевле по стоимости

Недостатки:

- Меньшая степень свободы в вопросах лицензирования.
- Поддержка ИТ инфраструктуры и информационной безопасности снижается по причине использования не самых современных версий систем и решений.
- Обновление систем происходит скачками (раз в три, четыре года), и требует наличие дополнительных ИТ ресурсов, людей и времени.

Рекомендации по выбору:

Для организаций имеющих тесную интеграцию и зависимость бизнеса и Информационных Технологий рекомендуется использование первого варианта лицензирования.

Стратегия построения инженерных систем

Для «On premise» и «Hybrid» решений требуется определиться с требованиями к инженерным системам.

К требованиям по инженерным системам можно отнести:

- Физические требования к помещению дата центра, серверной комнаты, коммуникационных шкафов и т.п.
- Требования к Структурированной Кабельной Системе (избыточность, резервирование и т.п.)

Стратегия тестирования

Тестирование один из важных элементов при построении ИТ архитектуры. Уже на этапе проектирования ИТ архитектуры требуется определить вопросы по тестированию. Различают следующие площадки:

•«*Test или Development*» – площадка с развернутыми отдельными ИТ сервисами. Используется для разработки сервисов или отладка элементов ИТ сервиса.

•«*Pre-production*» – уменьшенная копия «Production» площадки со всеми компонентами ИТ архитектуры. Используется для отработки взаимодействия ИТ сервисов между собой. Так же позволяет эмулировать ситуации при поиске неисправностей, расчет производительности и т.п.

•«*Production*» – площадка с развернутым вычислительными ресурсами предоставляющие ИТ сервисы.

Рекомендации по выбору:

В зависимости от предъявляемых требований со стороны бизнеса и финансовых возможностей организации возможны различные решения. Детальное описание рассмотрено далее в данном руководстве.

Стратегия уровня избыточности

Определение уровня избыточности компонентов ИТ инфраструктуры указывает требования к дублированию компонентов. Может рассматриваться на уровне:

Компонентов инженерных систем (каналы и кабели связи, коммутационные стойки и т.п.). Резервирование (дублирование или избыточность) на уровне критически важных элементов, таких как:

- Каналы и кабели связи (два и более проходящие по разным шахтам),
- Избыточное количество рабочих точек (обеспечивает рост организации и резерв на отказ)

Компонентов устройства (сервера, коммутатора и т.п.). Резервирование (дублирование) на уровне критически важных элементов устройства, таких как:

- Процессоры (два и более процессоров),
- память (две и более «банки» памяти),
- жесткие диски (два диска в RAID1 массиве + один резервный)
- сетевые карты и адаптеры (две карты с одним и более портами)
- блоки питания (N+N или N+1)

Компонентов ИТ сервиса. Резервирование на уровне устройств и типу их включения. Как пример:

- Два сервера в отказоустойчивом кластере
- Два сервера выполняющих одну и ту же роль (два контролера домена)
- Два сервера выполняющих разные роли в штатном режиме, но могут принять роли соседа в случае отказа (Файловый сервер и сервер печати. В случае отказа можно установить роль на соседний сервер).

Географически разнесенные устройства в пределах сайта. Расположение пары устройств в разных стойках, помещениях или зданиях в пределах одного сайта.

Географически разнесенные устройства на уровне сайтов. Расположение пары устройств на разных сайтах.

Определение уровня резервирования зависит от критичности бизнеса к отказу или простою. Анализируется в процессе Управления Рисками, формализуется в документах по стандартах оборудования.

Стратегия по системам резервирования и архивирования

К системам резервирования и архивирования ИТ инфраструктуры можно определить следующие требования:

- Компоненты систем должны быть установлены на выделенных физических элементах (серверах, СХД и т п)
- Возможно совмещение систем резервирования и архивирования на одних компонентах.
- Права доступов для учетных записей (автоматического) резервного копирования по возможности не должны предоставлять возможность удаления данных или же их перезапись.

Требования к резервным сайтам

Наличие компонента резервного сайта, один из важных элементов при построении ИТ архитектуры. Он имеет отношение к концепции информационной безопасности, отказоустойчивости и восстановления. Различают следующие виды отказоустойчивости:

- Отказоустойчивость на уровне отдельных компонентов ИТ сервиса
- Отказоустойчивость на уровне дублирования компонентов ИТ сервиса
- Отказоустойчивость в пределах сайта
- Отказоустойчивость с географически распределённым сайтом

Различают следующие категории *специализированных* резервных площадок:

- «Горячий» – готовность в течении секунд/минут/часов
- «Теплый» – готовность в течении часов/дней (порядка 48 часов)
- «Холодный» – готовность в течении дней/не полное восстановление

Не специализированные площадки – использование имеющуюся площади организации (филиал и т п)

Подходы по внедрению Архитектуры Предприятия

При разработке архитектуры вашей компании необязательно начинать с разработки стратегической архитектуры. Безусловно, *подход «Сверху – Вниз»* (Стратегическая архитектура – > Архитектура сегмента – > Архитектура решений) самый распространенный и имеет множество преимуществ:

- Будет понятен общий вектор развития организации.
- На уровне сегментов и решений будет меньше разброда и шатаний.
- Общие правила и подходы на уровне организации, а затем их трансляция на уровень сегментов и решений.
- Общие для компании информационные системы и сервисы, повторно используемые на уровне сегментов и решений.

Основная идея движение от «общего к конкретному» (Generic- to – specific), а также непрерывное совершенствование ИТ архитектуры на основе требований бизнеса. Но вы можете использовать и другие подходы:

«Снизу – Вверх» (Архитектура решений – > Архитектура сегмента – > Стратегическая архитектура). От архитектуры конкретных проектных решений к стратегической архитектуре. Компания начинает с архитектуры решения. До старта проекта решение прорабатывается и описывается. Затем – более высокие уровни Архитектуры Предприятия. Этот метод позволяет быстро получить ценность от методов Архитектуры Предприятия. Но есть вероятность, что без стратегической архитектуры, как основы, архитектуры различных решений «разъедутся». Придется потратить время и ресурсы на приведение их к общему знаменателю.

«От Сегмента» (Архитектура Сегмента – > Архитектура решений и стратегическая архитектура) Когда компании важно решить проблемы в конкретном подразделении, запустить новое направление бизнеса, либо если компания не готова к крупномасштабным внедрениям Архитектуры Предприятия и хочет «потренироваться на кошечках». Обкатать идеи на одном подразделении или направлении бизнеса. В зависимости от целей и сроков их достижения вам нужно выбрать подход для вашей компании.

Информационные системы вашей компании возможно далеки от идеала. Но выбросить их и переделать будет стоить очень дорого и займет много времени. Ценность такой инициативы сильно ниже нуля. Опытный архитектор «полечит» проблемы с интеграцией, информационной безопасностью, инфраструктурой, заплатками, а саму систему заставит давать большую ценность вашей компании. Реализация будет идти небольшими, точно выверенными доработками. Он будет решать конкретные проблемы, а не «делать все по уму». Замена системы – это серьезное изменение для компании, и для него должны быть веские аргументы.

Следующий важный вопрос при внедрении проекта построения Архитектуры Предприятия: Стоит ли привлекать консультантов или делать все самим (MAKE or BUY)? Однозначного ответа на него нет. Все зависит от конкретной организации, целей и возможностей. Для примера, представим ситуацию:

Мы заключаем контракт с уважаемой компанией «СУПЕР ПУПЕР КОНСУЛЬТАНТЫ и Ко». Через несколько месяцев получаем набор документов под названием «Архитектура компании БАНАНАС и Ко», в котором расписана правильная система, множество схем, описаний и т.п. Так сказать решение «под ключ». Заманчиво? Да скорее всего дорого, но быстро, профессионально, не один «комар» (читаем «аудитор») носу не подточит. Возможно.

Конечно, использование внешних ресурсов, чужого опыта и знаний – это один из самых быстрых способов достижения результатов. но есть несколько минусов:

- План перехода будет подозрительно похож на список товаров и услуг этой уважаемой компании. Любой внешний исполнитель в рамках проекта будет продавать другие свои товары и услуги. Это общая практика и один из главных законов продаж.

- Вы получите только набор документов, а не архитектурную практику. У вас останется только описание. Процессы, люди и управление уйдут вместе с консультантами.

- Ваши люди так и не понюхают пороху, консультанты сделают всю работу. А опыт и знания будут использовать у нового заказчика. Ваши специалисты будут архитекторами только на бумаге. У них не будет ключевого навыка – умения принимать технические решения.

На мой взгляд, чтобы выстроить процессы в организации, нужны сильные и грамотные менеджеры, знакомый с культурой компании, имеющие теоретические и практические навыки, умеющие и заинтересованные в передаче опыта и знания коллегам и сотрудникам. Они должны играть главную роль и являться движущей силой организации.

Я бы рекомендовал привлечение внешних консультантов, но не отдавал бы архитектурную работу внешним исполнителям на 100%. В компании должны оставаться не только результаты проектов, но и люди, знания и опыт. Поэтому большую часть работы должны выполнять сотрудники вашей организации. Но под присмотром эксперта и с его помощью. Обязательно в проект внедрения должны быть включены ресурсы по обучению сотрудников. Как гласит народная мудрость «практика без теории слепа, теория без практики мертва».

Инструменты внедрения Архитектуры Предприятия

Как и какие выбрать инструменты для разработки Архитектуры Предприятия. На рынке представлены десятки инструментов. От бесплатных до дорогих. Внедрение некоторых из них может стоить сотни тысяч долларов. Я рекомендую использовать по возможности бесплатные или дешевые инструменты (если не имеются в наличии) по следующим причинам:

- На старте архитектурной практике ещё предстоит доказать свою эффективность, поэтому значительные инвестиции и упущенное время будут серьезным препятствием на пути ее создания.

- На начальном этапе лучше использовать простые и знакомые инструменты, не требующие дополнительного внедрения и обучения. Скорость внедрения определяет успех.

- Нужно наработать опыт и получить первые результаты, чтобы сформировать адекватные требования к дорогой системе.

Только после того, как практика запущена, получены опыт и первые результаты, время переходить на специализированные продукты.

Какие минимальные требования к инструментам? Что они должны помогать вам делать?

- Писать и редактировать тексты, составлять схемы, вести таблицы, делать презентации и т. д.

- Размещать их на общедоступном ресурсе, регулировать права доступа к информации, обсуждать эти материалы.

Набор инструментов:

- Для составления документов, схем, таблиц и презентаций можно использовать стандартный офисный пакет, например, MS Office. Для него есть готовые шаблоны для документов. Есть расширения для Visio, при помощи которых можно нарисовать все нужные схемы.

- Для совместной работы можно использовать корпоративный портал, систему управления документами, корпоративную почтовую систему, корпоративную систему передачи сообщений или выделенный файловый ресурс в корпоративной сети организации. Выбор решения будет зависеть от того, что уже есть у вашей компании.

Рекомендации по внедрению

На основе требований и ограничений, определённых выше, можно суммировать основной подход при построении ИТ архитектуры компании:

- Сервис ориентированный подход к построению ИТ архитектуры
- Минимизировать риски за счет использования проверенных и хорошо зарекомендовавших себя технологий, и решений
- Снижение расходов за счет максимального и оптимального использования текущих ИТ активов и решений
- Снижение сложности и разнородности имеющейся инфраструктуры
- Максимально возможный рациональный подход к консолидации ИТ активов (железо, программное обеспечение), оставляя на периферии уровень поддержки пользователей и/или специализированных систем
- Множественное использование ИТ активов (ИТ персонал) на проектах внутри компании
- Стандартизация ИТ активов (железо, программное обеспечение и решения) для снижения стоимости владения, сложности сопровождения и повышения информационной безопасности
- Введение единых ИТ политик и процедур в компании для снижения стоимости владения, сложности сопровождения и повышения информационной безопасности
- Автоматизация рутинных ИТ процессов для снижения стоимости владения, сложности сопровождения и повышения информационной безопасности
- Формирование единой политики информационной безопасности в организации
- Формирование проектной команды с высоким уровнем компетенции

Совместимость

Везде где это возможно, необходимо учитывать наличие имеющихся систем и по возможности максимально использовать выгоды данных систем. Это позволит снизить издержки на замену, или внедрению и сопровождению новых систем.

Следование основным принципам

При внедрении новых или промежуточных решений, или же при разворачивании бизнеса (или его расширении), рекомендуется руководствоваться основными требованиями, изложенными в данном документе. Стоимость решения и сопровождения ИТ инфраструктуры напрямую зависит от выбора того или иного решения.

Заключение

В заключение можно сказать, что построение Архитектуры Предприятия один из важнейших аспектов построения эффективного механизма корпоративного управления с интеграцией информационных технологий для получения наилучших результатов. Интегрирование механизмов Управления ИТ сервисами (ITSM), систему Управления Проектами (PMM) методов аудита и контроля (COBIT) позволит бизнесу добиться исключительных результатов и максимальной отдачи от ИТ.

Концепция Управления Проектами

Общие Положения

Управление проектом рассматривается как метод и набор процедур, основанных на принятых принципах управления, которые используются для планирования, оценки и контроля рабочих заданий, с целью получить желаемый конечный результат в установленные сроки, в рамках выделенных средств и в соответствии с требованиями к проекту. Данный раздел содержит основные принципы управления проектами, которые должны быть приняты во внимание при построении ИТ Архитектуры Предприятия. Любые проекты, в которые вовлечены ИТ (влияют на ИТ или зависят от ИТ), рассматриваются как ИТ проекты. В процессе планирования, разработки и внедрения различных сервисов ИТ архитектуры необходимо руководствоваться принципами «проектного» подхода.

Управление Проектами очень обширная тема, затрагивающие и тесно переплетающаяся с различными направлениями деятельности, такими как управление качеством, рисками, финансами и т.п. В данной главе мы кратко рассмотрим основные аспекты, методы и техники управления проектами.

Проект – это одноразовая, не повторяющаяся деятельность или совокупность действий, за определенное время, направленная на создание уникальных продуктов, услуг, результатов или четко поставленных целей.

Признаки проекта:

- Есть конкретная дата начала. Ключевой признак проекта «временная составляющая», т.е. есть начало и конец проекта.

- Есть конкретная дата конца. Дата установить предполагаемый срок окончания проекта, но зафиксировать условие окончания проекта по получению конечного результата или достижения целей проекта.

- Результат проекта уникален. Это второе отличие проекта от процесса или регулярной деятельности. «Уникальный» не означает абсолютно новый для всех, он может быть уникальным для организатора или команды.

- Ресурсы и бюджет – ограничены.

- Специфический порядок выполнения заданий. Он может предполагать временное изменения в иерархии подчинения и управления от установленного в организации.

Программа – отличается от проекта тем, что она больше по масштабу и может состоять из множества проектов. Так, например, у организации есть программа перехода к централизованной системе управления ИТ, которая может включать в себя несколько отдельных проектов.

Задача или набор рабочих заданий – представляет из себя набор действий, которые могут быть выполнены одним или несколькими лицами с помощью простого списка, задающего последовательность действий.

Ключевые аспекты Управления Проектами

Тройственная ограниченность или Треугольник проекта – описывает баланс между содержанием (объем работ или score), стоимостью (cost) и временем (time, schedule) проекта, что влияет на конечный результат – качество (quality). Качество – это четвертый элемент проектного треугольника, который находится в центре, и любое изменение сторон влияет на него.

Как говорится в популярном слогане «Сделаем хорошо, быстро, дешево. Нужно подчеркните».

Вывод данного постулата сводится к одному: невозможно изменить один из факторов (деньги, перечень работ, время или качество) не повлияв по крайней мере на один из других факторов. Как пример:

Чтобы приблизить дату окончания (время) проекта, вы можете потратить больше ресурсов (деньги) или убрать некоторые задачи (область охвата) из проекта.

Чтобы сделать проект в рамках бюджета (деньги), вы можете либо сократить некоторые задачи (область охвата), что повлияет на возможности продукта (качество).

Чтобы добавить в продукт новые возможности (качество), вы можете продлить срок проекта, чтобы выделить время на новые задачи (время), тем самым добавив новые задачи (область охвата) и привлечь новых людей, чтобы работать быстрее (затраты).

Идея всех методик и подходов сводится к фокусированию внимания на одном или нескольких факторах с контролем воздействия на оставшиеся.



Треугольник проекта

Критерии успешности проекта – чтобы проект считался успешным фактические показатели, должны совпадать с запланированными в плане завершения проекта в срок, в рамках установленного бюджета и в соответствии с требованиями, предъявляемыми к конечному результату. Эти требования могут и должны быть сформулированы и включать в себя измеряемые критерии – показатели успеха проекта.

Цель руководителя проекта – достижения критериев успешности проекта.

Основная задача руководителя проекта – соблюдение баланса треугольника проекта.

Как показывает практика лишь четверть всех проектов достигает критериев успеха. Поэтому при планировании проекта желательно указывать основные факторы успеха проекта (например, бюджет проекта и неизменность результата), а также возможные допуски – разрешенные отклонения связанных факторов (например, превышение срока реализации проекта, но не более чем на десять процентов и т.п.).

Для удобства управления проектами, в организации можно ввести классификацию проектов по различным категориям. Категории и признаки проекта каждая организация определяет самостоятельно. Как пример общей классификации проекта по сложности:

•*Простой* – Проект, целью которого являются новые или улучшение существующих бизнес процессов, информационных систем, программного обеспечения, документов, сервисов, машин, оборудования. Будет изменяться или создаваться только один бизнес-процесс только в одном подразделении компании. Характеризуется как правило малой стоимостью проекта, небольшим охватом работ, коротким сроком и вовлечением не большого количества сотрудников.

•*Сложный* – Проект, целью которого являются новые или улучшение существующих бизнес процессов, информационных систем, программного обеспечения, документов, сервисов, машин, оборудования. Будут изменяться или создаваться несколько бизнес-процессов или задействовано несколько подразделений компании. Характеризуется как правило высокой стоимостью проекта, большим перечнем задач, продолжительным сроком и вовлечением большого количества сотрудников.

Классификация по приоритету, позволяет определить порядок выполнения проектов:

- Низкий,
- Средний
- Высокий.

Проекты могут быть классифицированы по назначению в контексте взаимодействия с ИТ:

Административные проекты – связаны с изменениями в организации как правило не влияющие на состояние ИТ инфраструктуры, или без привлечения ИТ ресурсов.

Внутренние проекты – происходящие, как правило, внутри одного подразделения организации и характеризуются незначительными изменениями в ИТ инфраструктуре или с незначительным привлечением ИТ ресурсов

ИТ проекты – проекты, происходящие внутри организации, в значительной степени влияют на состояние ИТ инфраструктуры, имеющие цель изменить ИТ инфраструктуру организации, или с привлечением значительных ИТ ресурсов организации.

Самая первая фаза (этап) проекта, вне зависимости от выбранного метода управления проектом, является начальная стадия или инициализация проекта. На этом этапе принимается решение по старту проекта. В контексте данной книги, инициаторами проекта могут выступать одна из двух сторон организации:

- Бизнес подразделения
- ИТ департамент.

Вводной информацией на данном этапе может служить стратегия организации или бизнес план.

Бизнес план – это документ, дающий развернутое обоснование проекта и возможность всесторонне оценить эффективность принятых решений, планируемых мероприятий, ответить на вопрос, стоит ли вкладывать деньги в данный проект.

Определение целей проекта – является одним из важнейших аспектов процесса управления проектом. Каждый проект должен иметь хотя бы одну основную цель и возможно несколько частных, вспомогательных целей. Цель имеет следующие функции:

- Задаёт направление работ по реализации проекта.
- Определяет конечные результаты в терминах конечных продуктов или услуг.

- Служить в качестве источника информации для разрешения спорных вопросов, касающихся проекта.

Формулировка основной цели должна быть ориентирована на действия, быть краткой и простой, а также как можно более понятной. Важно помнить, что цель должна быть сформулирована с использованием таких терминов, которые не вызовут непонимание у тех, кто будет участвовать в проекте, принимать решения или читать документы по проекту. На начальном этапе формулировка целей может быть размыта и выражать общее направление.

Определение заинтересованных сторон (Stakeholders) – позволяет определить, кто является непосредственным заказчиком, имеет права принятия окончательного решения и т. п. Кроме этого выявляются общие вопросы по организации проекта, такие как организация, состав рабочей группы по направлениям (на уровне руководителей), порядок взаимодействия, принятия решений, механизмы коммуникации и т. п.

Выходом данной фазы проекта является определение заинтересованных сторон, постановка целей проекта и общие вопросы организации. В зависимости от выбранной методики управления проектом могут быть сформированы дополнительные выводы и результаты. В качестве первичных документов начального этапа можно рассматривать бизнес план, устав проекта или протокол собрания.

Следующим этапом управления проектом может быть **этап планирования**. Планирование является наиболее важным и сложным этапом управления проектами. На данном этапе происходит основная работа, по детальной проработке проекта и формируется ответ на главный вопрос: готова ли организация к выполнению проекта и будет ли проект успешен. Уровень сложности зависит от выбранной методологии и сложности проекта. На данном этапе может быть собраны бизнес требования, конкретизация целей, определение критериев успешности, детализация задач, планирование ресурсов, времени и т. п. Основные документы на выходе этапа планирования могут представлять из себя план проекта, техническое задание и т. п.

Согласование и утверждение проекта – Данный этап или более верное определение «веха» или контрольная точка проекта, характеризуется ответом на главный вопрос: начинаем проект или нет. После положительного ответа руководства, проект переходит на стадию исполнения. Для различных методологий управления проектами может находиться на различных этапах проекта.

Фаза реализации проекта. Как только проект утвержден, начинается реальная работа. В терминах цикла управления проектом, мы завершили этапы планирования и переходим к этапам реализации. Реализация проекта начинается с формирования рабочей группы по проекту, после этого следует детальная разработка и распределение рабочих заданий, сметы. Реализация завершается, когда конечный продукт, полученный в результате выполнения проекта, удовлетворяет требованиям проекта. В процессе реализации проекта, вне зависимости от методологии, параллельно выполняется фаза или процессы по мониторингу и контролю состояния проекта. Еще один из важных процессов данного этапа – управление изменениями. Основные документы на данном этапе – отчеты по статусу проекта, выполненным работам, использованию ресурсов.

После реализации, проект подходит к **финальной стадии – завершению**. Проект является завершенным с точки зрения достижения поставленных целей проекта и получения ожидаемых конечных результатов. Кроме этого владельцы проекта или руководство организации могут принять решение по досрочному завершению проекта, или изменение целей проекта приводящие к его завершению.

Подходы и методы управления проектами

Выбор правильной методологии управления проектами (Project Management Methodologies PMM) – первый шаг к успеху вашей команды. Управление проектами помогает улучшать их реализацию с точки зрения эффективности и затрат, одновременно снижая риски. Но одного лишь декларирования приоритетов для этого недостаточно. Нужно хорошо понимать, какое позитивное воздействие оказывает каждая из методологий управления проектами и чем она может помешать успешной реализации проекта.

В этой главе мы коротко рассмотрим наиболее популярные методологии управления проектами и более подробно остановимся на наиболее интересных с моей точки зрения.

Процессно-ориентированное проектное управление (Process Based Project Management PBPM)

На сегодняшний день является основным подход к концепции управления проектом, основанный на процессном подходе к управлению проектами. Данный подход гарантирует того, что каждый проект будет направлен на продолжение следованию миссии компании. Перед стартом (инициацией) проекта, план проекта анализируется с целью определения его соответствия утвержденной или установленной миссии. Если же результат анализа отрицателен, то все стратегии и цели корректируются. Каждое действие добавляет ценность к стратегическому видению организации. Данные методы управления проектами также подходят и для административных проектов в компаниях.

Классическая Методология Управления Проектами «ВОДОПАД» (WATERFLOW)

«ВОДОПАД» (*WATERFLOW*) – Традиционная, классическая методология управления проектом. Она также носит название «каскадной» или «поточной» модели, вследствие того, что предлагаемая ею последовательность фаз напоминает водопад. Наиболее очевидный способ сделать свой проект более управляемым – это разбить его процессы реализации на последовательные этапы. Именно на такой линейной структуре базируется традиционное проектное управление. Данный подход не предполагает возврата к предыдущим этапам по их завершению и принятию, или внесение изменений в требования проекта. Данная методология управления проектами предполагает разбиение проекта на ряд последовательных задач, с четким определением целей и сроков. Члены проекта выполняют задания в установленном порядке, завершая каждое задание перед тем, как преступать к последующему. В «водопадной» модели, в применении к ИТ проектам разработки программного обеспечения, стадии идут в следующем порядке:

- Определение требований
- Проектирование
- Конструирование («реализация» или «кодирование»)
- Интеграция
- Тестирование и отладка («верификация»)
- Инсталляция
- Поддержка

При этом разработчик не может перейти к следующей стадии, не закончив предыдущую. Сначала полностью завершается этап «определение требований», в результате чего получается список требований к программному обеспечению. После того как требования полностью определены, происходит переход к проектированию, в ходе которого создаются документы, подробно описывающие для программистов способ и план реализации указанных требований. После того как проектирование полностью выполнено, программистами выполняется реализация полученного проекта. На следующей стадии процесса происходит интеграция отдельных компонентов, разрабатываемых различными командами программистов. После того как реализация и интеграция завершены, производится тестирование и отладка продукта; на этой стадии устраняются все недочёты, появившиеся на предыдущих стадиях разработки. После этого программный продукт внедряется и обеспечивается его поддержка – внесение новой функциональности и устранение ошибок.



Сильные стороны классического проектного менеджмента – является то, что он требует от заказчика или руководства компании определить, что же они хотят получить, уже на первом этапе проекта. Раннее включение привносит определённую стабильность в работу проекта, а планирование позволяет упорядочить реализацию проекта. Кроме того, этот подход подразумевает мониторинг показателей и тестирование, что совершенно необходимо для реальных проектов. Потенциально, классический подход позволяет избежать стрессов ввиду наличия запасного времени на каждом этапе, заложенного на случай каких-либо осложнений и реализации рисков. Кроме того, с правильно проведённым этапом планирования, руководитель проектов всегда знает, какими ресурсами он обладает, даже если эта оценка не всегда точная. Основные преимущества: возможность заранее посчитать стоимость реализации решения и контроль состояния на всех этапах жизненного цикла управления проектом.

Слабые стороны классического проектного менеджмента – не толерантность к изменениям. Если в вашем проекте ресурсы и время не являются ключевыми ограничениями, а содержание проекта подвержено изменениям – возможно вам стоит присмотреться к другим системам управления проектами. Кроме этого, в реальных ИТ проектах довольно сложно на начальном этапе сформулировать детальные требования к конечному проекту.

Гибкая Методология Управления Проектами (Agile Project Management Methodology)

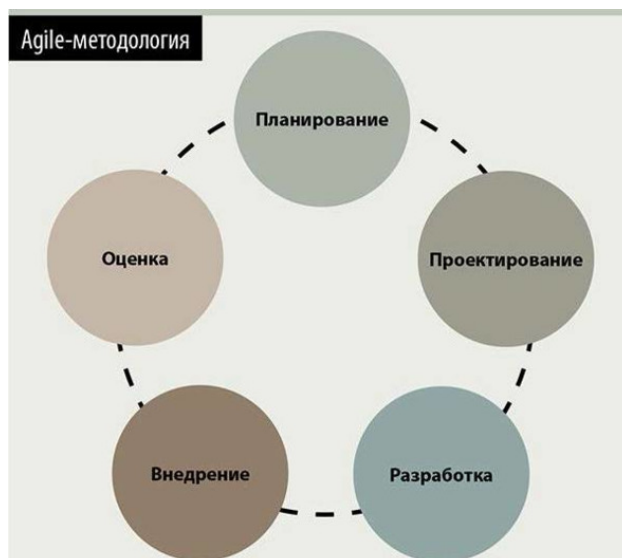
AGILE – семейство процессов и методов разработки гибких методов управления и ведения проектов. Сам по себе Agile – скорее набор идей и принципов того, как нужно реализовывать проекты. Уже на основе этих принципов и лучших практик были разработаны отдельные гибкие методы или, как их иногда называют – фреймворк (frameworks). Примеры: Scrum, Kanban, Crystal, LeSS, SAFe, Nexus и многие другие. Эти методы могут достаточно сильно отличаться друг от друга, но они следуют одним и тем же принципам. Гибкие методы предполагают изменения требований к продукту в течении всего времени проекта. Такое проектирование подразумевает совершенно иной подход к управлению проектами. Изначально методология разрабатывалась для проектов, которым требуются высокая гибкость и быстрая реализация. Гибкое управление проектом представляет собой поступательную и итеративную проектную методологию. Ее главной особенностью является то, что в начале выполнения проекта точно неизвестно, каким должен быть конечный продукт и каким будет жизненный цикл проекта. Вместо этого, проектная деятельность разбивается на несколько итеративных фаз – короткие циклы, называемые «спринтами». Каждый спринт состоит из множества задач и имеет свой конечный продукт и результат. Методология Agile позволяет менеджерам проектов постоянно получать обратную связь и улучшать продукт после каждой итерации. В соответствии с данной методологией управления проектами, ответственность за результат делится между тремя ролями:

- *Владелец продукта* – определяет проектные цели, разрабатывает оптимальный график при заданных проектных параметрах, адаптирует процесс выполнения проекта к изменившимся требованиям и устанавливает приоритеты в характеристиках продукта

- *Scrum мастер* – устанавливает приоритеты в выполнении задач командой проекта и устраняет возникающие затруднения, препятствующие этому

- *Члены команды* – выполняют большинство поставленных задач, осуществляют ежедневный менеджмент, создают отчеты о ходе выполнения проекта, контролируют качество продукта.

Методология Agile является гибкой и позволяет легко изменить параметры проекта, что является значимым для таких сервисно-ориентированных проектов, как разработка программного обеспечения или графический дизайн. Но это методология не подходит для проектов со строго заданными параметрами и требованиями.



В процессе управления проектом важно умение быстро адаптироваться к изменениям, отслеживать последние тенденции развития и уметь извлекать из них выгоду. Не менее важным является и человеческий ресурс. Поэтому и необходимо умение создавать динамичную команду проекта на основе сотрудничества и гибкости, возможности нахождения компромисса. Немаловажную роль играют заинтересованные стороны (Stakeholders). Они контролируют и проверяют проект на каждой стадии, а члены команды, с свою очередь, правильно и своевременно корректируют проект, создавая высококачественные продукты или услуги, соответствующие потребностям и запросам потребителей.

AGILE-проектирование лучше подходит для проектов, требующих интенсивного взаимодействия в реальном времени и реализуемых высокомотивированными командами, не нуждающимися в дополнительном контроле. Методология AGILE отличается высокой интерактивностью, дает возможность быстро подстраиваться под проект. Одно из главных ее преимуществ в том, что можно быстро выявлять спорные моменты и вносить необходимые изменения на ранней стадии разработки, не дожидаясь завершения тестирования. AGILE – проектирование обеспечивает применение повторяющихся процессов, снижение рисков, оперативную обратную связь, быструю оборачиваемость и уменьшение сложности.

Сильные стороны Agile – их гибкость и адаптивность. Он может подстроиться под практически любые условия и процессы организации. Именно это обуславливает его нынешнюю популярность и то, сколько систем для различных областей было создано на его основе. Один из принципов Agile: «Реакция на изменения важнее следования плану». Вотчина Agile – разработка новых, инновационных продуктов с «открытым концом». В проектах по разработке таких продуктов высока доля неопределённости, а информация о продукте раскрывается по ходу проекта. В таких условиях реализовывать проект по «водопаду» становится невозможно – нет информации для планирования. Второй сильной стороной подхода является возможность «совершать ошибки» и быстро их исправлять без существенного влияния на статус проекта в целом.

Слабая сторона состоит в том, что каждой команде придётся самостоятельно составлять свою систему управления, руководствуясь принципами Agile. Это непростой и длительный процесс, который потребует изменений всей организации, начиная процедурами и заканчивая базовыми ценностями. Это тернистый путь и не всем организациям он под силу. Этот путь потребует от лидера изменений не только знаний и упорства, но и серьёзных административных ресурсов, а также затрат. Кроме этого – плавающая оценка сроков и бюджетов, неопределенность планирования целей и задач, недостаточное документирование и как след-

ствии увеличение вероятности расхождения поставленных задач и фактической реализации, сложность ретроспективного анализа проекта.

Метод SCRUM семейства Гибких Методов Управления Проектами

SCRUM — («схватка») классический метод, с описанием процессов планирования, контроля и анализа на всех этапах ведения проекта, базирующийся на идеях Agile. Методология реализации agile-разработки предполагает использование интерактивного подхода. «Скрам-сессии», или «30-дневные спринты», используются для определения приоритетных задач. Роль менеджера проекта для упрощения передается скрам-мастеру. Для независимого решения конкретных задач формируются небольшие команды. В ходе встреч со скрам-мастером оцениваются достигнутые результаты, после чего определяется приоритетность невыполненных задач. Основная особенность методики:

- Совещания и анализ по определённым отрезкам времени «спринтов» (sprints).
- Небольшая команда
- Ограничение по определённым промежуткам времени (sprint) для выполнения WIPs.
- Определённых «кусоч» продукта (Work in Process WIPs)

Следуя заветам Agile, Scrum разбивает проект на части, которые сразу могут быть использованы Заказчиком для получения ценности, называемые заделами продуктов («W» – product backlog). Затем владельцем продукта – представителем заказчика в команде определяются приоритеты этих частей. Самые важные «кусочки» первыми отбираются для выполнения в спринте – так называются итерации в Scrum, длящиеся от 2 до 4 недель. В конце спринта заказчику представляется рабочий инкремент продукта – те самые важные «кусочки», которые уже можно использовать. Например, сайт с частью функционала или программа, которая уже работает, пусть и частично. После этого команда проекта приступает к следующему спринту. Длительность у спринта фиксированная, но команда выбирает её самостоятельно в начале проекта, исходя из проекта и собственной производительности.

Чтобы удостовериться в том, что проект отвечает требованиям заказчика, которые имеют свойство изменяться со временем, перед началом каждого спринта происходит переоценка ещё не выполненного содержания проекта и внесение в него изменений. В этом процессе участвуют все – команда проекта, лидер команды проекта (Scrum Master) и владелец продукта. И ответственность за этот процесс лежит на всех. Scrum Мастер призван помочь участникам проекта лучше понять и принять ценности, принципы и нормы практики Scrum. Он лидер и посредник между внешним миром и командой. Его задача – следить, чтобы никто не мешал команде самостоятельно и комфортно работать над поставленными задачами. Команда же отвечает за то, чтобы в конце спринта все необходимые задачи были сделаны, а поставки – выполнены. Основная структура процессов Scrum вращается вокруг 5 основных встреч:

- упорядочивания заделов (backlog),
- планирования Спринта,
- ежедневных летучек
- подведения итогов Спринта
- ретроспективы Спринта.

Встреча по упорядочиванию заделов (Backlog Refinement Meeting, «Backlog Grooming»): Эта встреча аналогична фазе планирования в классическом проектном управлении, и проводится в первый день каждого Спринта. На ней рассматривается – что уже было сделано по проекту в целом, что ещё осталось сделать и принимается решение о том, что же делать дальше. Владелец продукта определяет, какие задачи на данном этапе являются наиболее при-

оритетными. Данный процесс определяет эффективность Спринта, ведь именно от него зависит, какую ценность получит Заказчик по итогам спринта.

Планирование Спринта: После того, как Владелец продукта определил приоритеты, команда совместно решает, что же конкретно они будут делать во время грядущей итерации, как достигнуть поставленной на предыдущей встрече цели. Команды могут применять различные инструменты планирования и оценки на данном этапе, лишь бы они не противоречили принципам и логике Scrum. Планирование Спринта проводится в самом начале итерации, после Встречи по упорядочиванию продукта.

Ежедневные летучки: Каждый день спринта, в идеале, в одно и то же время, члены команды тратят 15 минут на то, чтобы поделиться информацией о статусе задач и состоянии проекта. На ней не происходит обсуждений проблем или принятия решений – если после встречи возникают вопросы и конфликты, Scrum Мастер и вовлечённые участники обсуждают их отдельно. Летучка же нужна для обмена информацией и поддержания всех членов команды в курсе состояния проекта.

Подведение итогов Спринта: Цель этапа – обследование и адаптация создаваемого продукта. Команда представляет результаты деятельности всем заинтересованным лицам. Основная задача – убедиться, что продукт этапа соответствует ожиданиям участников и согласуется с целями проекта.

Ретроспектива Спринта: Проводится сразу после Подведения итогов спринта и до планирования следующего спринта. На нём команда выясняет, насколько чётко и слаженно проходил процесс реализации этапа. Обследованию подвергаются возникшие проблемы в работе, методологии и взаимодействии. Именно этот этап позволяет команде провести рефлекссию и следующий Спринт провести эффективнее.

Сильные стороны – был разработан для проектов, в которых необходимы «быстрые победы» в сочетании с толерантностью к изменениям. Кроме того, этот фреймворк подходит для ситуаций, когда не все члены команды имеют достаточный опыт в той сфере, в которой реализуется проект – постоянные коммуникации между членами командами позволяют недостаток опыта или квалификации одних сотрудников за счёт информации и помощи от коллег. На мой взгляд основное преимущество Scrum в том, что он позволяет «быстро ошибаться». Вместо того, чтобы долго и с большими затратами готовить крупный релиз, поставки раз в две недели по Scrum имеют небольшой размер. Их легко отслеживать и, если что-то идёт не так, быстро исправлять.

Слабые стороны – очень требователен к команде проекта. Она должна быть небольшой (5—9 человек) и кросс-функциональной – то есть члены команды должны обладать более чем одной компетенцией, необходимой для реализации проекта. Например, разработчик ПО должен обладать познаниями в тестировании и бизнес-аналитике. Делается это для того, чтобы часть команды не «простаивала» на разных этапах проекта, а также для того, чтобы сотрудники могли помогать и подменять друг друга. Кроме того, члены команды должны быть «командными игроками», активно брать на себя ответственность и уметь само-организовываться.

Подобрать такую зрелую команду очень непросто! Scrum подходит не для всех команд и организаций ещё и потому, что предлагаемый процесс может не подойти для разработки конкретного продукта – например промышленного станка или постройки здания.

Метод KANBAN семейства Гибких Методов Управления Проектами

KANBAN — также представляет из себя гибкий, итеративно-инкрементальный подход к управлению проектами базирующийся на идеях Agile. Является противоположностью «SCRUM» метода. Основные особенности методики:

- Каждый участник проекта самостоятельно берет на себя ограниченное количество задач, а не по указанию менеджера
- Работа вносится в карточку (Sticker)
- Количество «незавершённой» работы (WIPs) ограничено для каждой стадии
- Новая работа берется только тогда, когда существующая выполнена или «вытянута» (LEAN).
- Больше внимания к управлению изменениями, визуализация узким мест, незавершенной работы и т п
- Ограничения по количеству WIPs и их статусы

Lean выглядит немного абстрактным сам по себе, но в комбинации с Kanban его становится гораздо проще использовать для построения собственной системы управления проектами. Kanban очень похож на схему промышленного производства. На входе в этот процесс попадает кусочек металла, а на выходе получается готовая деталь. Также и в Kanban, инкремент продукта передается вперед с этапа на этап, а в конце получается готовый к поставке элемент. Руководство принципом – «держи на полках только то, что нужно клиенту». А потому в Kanban разрешается оставить неоконченную задачу на одном из этапов, если её приоритет изменился и есть другие срочные задачи – всё это нормально для работы по Kanban.

Kanban намного менее строгий, нежели Scrum – он не ограничивает время спринтов, нет ролей, за исключением владельца продукта. Kanban даже позволяет члену команды вести несколько задач одновременно, чего не позволяет Scrum. Также никак не регламентированы встречи по статусу проекта – можно делать это как Вам удобно, а можно не делать вообще.

Для работы с Kanban необходимо определить этапы потока операций (workflow). В Kanban они изображаются как столбцы, а задачи обозначают специальные карточки. Карточка перемещается по этапам, подобно детали на заводе, переходящей от станка к станку, и на каждом этапе процент завершения становится выше. На выходе мы получаем готовый к поставке заказчику элемент продукта. Доска со столбцами и карточками может быть, как настоящей, так и электронной – даже здесь Kanban не накладывает никаких ограничений на пользователей. Ваша собственная система Kanban может быть настолько гибкой, насколько Вы сами того пожелаете – ведь во многом Kanban является визуализацией идеи Agile.

Но у Kanban есть четыре столпа, на которых держится вся система:

•*Карточки:* Для каждой задачи создается индивидуальная карточка, в которую заносится вся необходимая информация о задаче. Таким образом, вся нужная информация о задаче всегда под рукой.

•*Ограничение на количество задач на этапе:* Количество карточек на одном этапе строго регламентировано. Благодаря этому сразу становится видно, когда в потоке операций возникает «затор», который оперативно устраняется.

•*Непрерывный поток:* Задачи из беклога попадают в поток в порядке приоритета. Таким образом, работа никогда не прекращается.

•*Постоянное улучшение* («кайзен» (kaizen)): Концепция постоянного улучшения появилась в Японии в конце XX века. Её суть в постоянном анализе производственного процесса и поиске путей повышения производительности.

Сильные стороны – Как и Scrum, Kanban хорошо подходит для достаточно сплочённых команды с хорошей коммуникацией. Но в отличие от Scrum, в Kanban нет установленных чётких временных рамок, что хорошо подходит для замотивированных и опытных команд. При правильной настройке и управлении, Kanban может принести большую пользу команде проекта. Точный расчёт нагрузки на команду, правильная расстановка ограничений и концентрация на постоянном улучшении – всё это позволяет Kanban серьёзно экономить ресурсы и укладывать в сроки и рамки бюджета. И всё это в сочетании с гибкостью.

Слабые стороны – Часто можно слышать, что по Kanban, в отличие от Scrum, можно работать с практически любой командой. Но это не совсем так. Kanban лучше всего подходит для команд, навыки членов которых пересекаются друг с другом. Таким образом они могут помогать друг другу преодолевать трудности при решении задач. Без этого Kanban будет не так эффективен, как мог бы быть. Также, как уже было сказано, Kanban лучше подходит в тех случаях, когда нет жёстких сроков. Для жёстких сроков проекта лучше подходит классический подход или Scrum.

Гибридная Методология Управления Проектами

Несмотря на то что многие команды отдают предпочтение либо методологии водопада, либо agile-проектированию, преимущества обоих подходов привели к появлению гибридной методологии, когда этапы планирования и определения требований выполняются согласно методологии водопада, а этапы проектирования, разработки, внедрения и оценки соответствуют гибкому подходу. Данная методология является попыткой применения сильных сторон каждого из основных подходов, а также снижения негативного воздействия слабых сторон. Гибридная методология может хорошо вписаться в сферу информационных технологий за счет того, что на уровне руководства организации проект имеет четко поставленные цели, хорошо документированы, предоставляет возможность мониторинга статуса проекта для предоставления руководству. На этапе выполнения за счет высокого уровня коммуникации между представителями бизнеса и непосредственно исполнителями задачи по проекту глубоко детализованы, понятны разработчикам. Возможность учиться «на лету», совершать ошибки и совершенствовать продукт под текущие требования бизнеса одинаково хорошо для разработчиков и бизнеса. Конечно за все надо платить. Подход предъявляет требования к как к техническому уровню подготовки сотрудников, так и к его персональным качествам.



Из собственной практики управления проектами в ИТ департаменте, данный подход не стоит внедрять в самом начале, но по мере совместной работы сотрудников ИТ департамента и бизнеса по нескольким проектам, так сказать «притирки» сотрудников данный метод будет является логическим продолжением и будет наиболее эффективным, и результативным.

Методология Быстрой Разработки Приложений (Rapid Application Development RAD)

Быстрая разработка приложений (RAD) – это специфическая проектная методология, чаще всего используемая в проектах по разработке ПО, основной целью которых является быстрое и качественное создание приложения.



Диаграмма: Сравнение WATERFLOW и RAD подходов

Суть методологии – подход к созданию средств разработки программных продуктов, уделяющая особое внимание скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. Практическое определение: RAD – это жизненный цикл процесса проектирования, созданный для достижения более высокой скорости разработки и качества ПО, чем это возможно при традиционном подходе к проектированию. Данная методология управления проектами выделяет четыре стадии проекта:

- Планирование
- Пользовательское проектирование
- Быстрое конструирование
- Переключение

Основные преимущества применения различных подходов RAD:

Применение *итеративного подхода* к разработке решений. Итеративный подход предполагает выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы, то есть своего рода обратную связь. Проект при этом подходе периодически проходит повторяющийся цикл Планирование-Реализация-Проверка-Оценка. Благодаря применению итеративного подхода, RAD может быстро реагировать на изменяющиеся требования бизнеса.

Инкрементальный подход предполагает разработку услуги «от куска к куску», то есть последовательно. При этом каждый «кусок» может поддерживать одну из бизнес-функций, для которых предназначена услуга в целом. Для бизнеса инкрементальный подход дает возможность использования какой-то значимой части услуги до того, как она будет разработана полностью. Дополнительные преимущества: продукт быстрее поступает на рынок, более широкие возможности для разработки устраивающего пользователей интерфейса, большая адаптивность к изменяющимся требованиям бизнеса и простота развития и изменения функциональности решения. Методология быстрой разработки приложений, с одной стороны, помогает улучшить показатели результативности проекта и повысить качество риск-менеджмента. Но с другой стороны, данная методология не подходит для масштабных ИТ проектов, может привести к низкому качеству кода и требует постоянного вовлечения клиента в процесс

исполнения всего проекта. RAD является более современным и гибким подходом к проектированию.

Недостатки метода – слабая документная база может приводить к недопониманию, ошибкам при формировании и разработки, сложность контроля и аудита процесса разработки. При проектировании услуг возможно комбинирование инкрементального и итеративного подходов. Начинают с определения требований для услуги в целом, продолжают путем инкрементальной разработки отдельных ее частей.

Методология Экстремального Программирование (Extreme Programming XP)

Данная методология управления проектами оказывает особенности (возможности) коротких циклов развития, частые релизы и открытое взаимодействие со стейкхолдерами. Команды сосредотачиваются на сотрудничестве, эффективности и производительности, написании наиболее простых из всех возможных кодов для достижения желаемого качества, при этом избегая истощения и низкокачественного конечного результата.

Методология моделирования событий Event Chain Methodology (ECM)

Эта методология управления проектами помогает выявить и спрогнозировать потенциальные риски. Анализ проекта при помощи метода Монте Карло и Диаграммы цепочки событий помогает определить вероятность некоторых рисков и их возможное влияние на проект в целом. Визуализация связей между внешними событиями и работами проекта помогает создать план, максимально приближенный к реальности.

Метод Адаптивные Рамки Проектов APF (Adaptive Project Framework)

Использование адаптивных/регулируемых рамок проектов APF (Adaptive Project Framework) позволяет улучшать проект на каждом этапе, основываясь на полученном опыте от предыдущих результатов. Определив цели проекта и постоянно контролируя работы проекта, менеджер может обеспечить успех максимально возможной стоимости бизнеса и создать бизнес-ценность для потенциального потребителя.

Метод «Реализация Выгоды» (Benefit Realization BF)

Цель метода «Реализация Выгоды» (Benefit Realization BF) – выгода от реализации проекта. Успех определяется как достижение желаемой/ожидаемой выгоды. Если клиенты хотят увеличить продажи CRM (система управления взаимоотношениями с клиентами – customer relationship management software), проект не будет выполнен/реализован до того момента, пока продажи не повысятся на 15% – даже если Вы установили и наладили работу CRM вовремя и с соблюдением/в соответствии с бюджетом.

Методология PRISM (проекты со встроенными устойчивыми/жизнеспособными методами)

Сочетание проектного планирования с экологической устойчивостью мер. Хотите пойти в «зеленом» направлении? В таком случае, PRISM точно для Вас! Сокращение расходования энергии и издержек обращения (распределение затрат), все это при одновременном снижении Вашего воздействия на окружающую среду.

ПРОЧИЕ МЕТОДЫ И СПЕЦИФИЧЕСКИЕ ПОДХОДЫ К УПРАВЛЕНИЮ ПРОЕКТАМИ

Помимо перечисленных, существуют и другие методологии управления проектами:

- функционально-ориентированная разработка (feature driven development, FDD),
- разработка динамических систем (dynamic systems development, DSDM),
- адаптивная разработка программного обеспечения, Rational Unified Process (RUP),
- Концепции Шесть сигм (six sigma) и Бережливого Производства (LEAN). Управление

проектом на основе принципов «контроля качества». Будут детально рассмотрены в следующей главе.

Методология Управления Проектами в Контролируемой Среде (PRINCE2)

PRINCE2 (Projects in Controlled Environments PRINCE2) так же является структурированной методологией к проектному управлению. Это одна из самых популярных методологий управления проектами, широко используемая в Великобритании в управлении как в бизнесе, так в органах власти. PRINCE2 – это процессно-ориентированная проектная методология (PBPM), которая фокусируется на процессах верхнего уровня (управление, организация, контроль), а не на низших задачах (декомпозиция работ, разработка графиков).

Методология PRINCE2 базируется на семи принципах, семи темах и семи процессах. Принципы являются центральным элементом методологии: если хотя бы один из них не выполняется, то нельзя говорить, что проект выполняется в рамках PRINCE2.

Принципы методологии PRINCE2

Постоянная оценка экономической необходимости – остается ли неизменной экономическая выгода от проекта на протяжении всего жизненного цикла проекта.

Обучение на опыте – команда проекта должна постоянно искать и изучать опыт предыдущих проектов

Определение ролевой модели – команда проекта должна иметь ясную организационную структуру и вовлекать подходящих людей для решения нужных задач

Управление по этапам – необходимо, чтобы проекты были спланированы, а также подвергались мониторингу и контролю на каждом этапе выполнения;

Управление по отклонениям – следует четко обозначить допустимые границы отклонений в проекте, чтобы установить границы ответственности.

Фокус на продуктах – необходимо концентрироваться на определении и достижении результатов проекта.

Адаптация к проектной среде – следует адаптировать процессы и инструменты управления проектом к требованиям проектной среды, а также к масштабу работ, их сложности, важности, квалификационным требованиям и степени риска.

PRINCE2 – гарантирует, что каждый проект имеет бизнес обоснование и способствует созданию ценности. Планирование начинается с четкого определения: потребностей, запрашиваемых потребителем, реальной выгоды и точной оценки затрат.

Темы методологии управления проектами PRINCE 2

Обоснование проекта: какую ценность проект принесёт организации?

Организация: каким образом необходимо распределить роли и ответственность между членами проектной команды для того, чтобы эффективно управлять проектом.

Качество: какие имеются требования и критерии к качеству и каким образом можно их обеспечить

Планы: шаги, требуемые для разработки плана, и инструменты PRINCE2, необходимые к использованию

Риски: каким образом менеджмент проекта будет разрешать проблему наличия неопределённости в плане проекта и во внешней среде.

Изменение: как руководство проекта будет оценивать влияние непредвиденных задач и изменений и реагировать на них

Прогресс: реализуемость проекта, выполнение планов и дальнейшее развитие проекта

Семь процессов управления PRINCE 2

И наконец, PRINCE2 подразумевает следующие семь процессов управления проектом:

- запуск проекта
- руководство проектом
- инициация проекта
- контроль этапов
- управление созданием продукта
- управление границами этапов
- закрытие проекта

PRINCE2 позволяет стандартизировать процедуры управления проектами, улучшить координацию деятельности, а также помогает понять, каким образом следует планировать проект и осуществлять мониторинг его выполнения, что следует делать, если план проекта не выполняется. Однако методология PRINCE2 не является лучшим выбором для проектов небольшого масштаба или для проектов с большей степенью вероятности изменений объема работ и требований к ним.

Сильные стороны PRINCE2:

- Адаптируемость к особенностям организации;
- Наличие чёткого описания ролей и распределения ответственности;
- Акцент на продуктах проекта;
- Определённые уровни управления;
- Фокус на экономической целесообразности;
- Последовательность проектной работы;
- Акцент на фиксации опыта и постоянном совершенствовании.

Слабые стороны PRINCE2 – Отсутствие или нехватка отраслевых практик и отсутствие конкретных инструментов для работы.

Управление проектами на основе методологии «PROJECT MANAGEMENT INSTITUTE PMI»

Общие положения

Данная методология является представителем Процессно-ориентированное подходом у Управлению Проектами (Process-Based Project Management PBPM) и основывается на методологии традиционного, классического подхода к управлению проектами. Наиболее очевидный способ сделать свой проект более управляемым – это разбить процесс его исполнения на последовательные этапы. Именно на такой линейной структуре базируется традиционное проектное управление. Все процессы в руководстве PMBooK разделяются на следующие группы (фазы):

Группа процессов инициации проекта (Initiating)

На данном этапе часто проводятся совещания и «мозговые штурмы», на которых определяется что же должен представлять из себя продукт проекта. В некотором смысле самой важной частью проекта является его начало. Именно на этой стадии «бросается жребий»: либо проект хорошо разработан и может быть выполнен в сроки, установленные высшим руководством, и в рамках выделенных средств, либо он не имеет перспектив и обречен на провал с самого начала. Эта часть проекта – не место для опрометчивых обещаний в попытке предстать героем. Скорее это время для рационального и творческого подхода к формулировке начальных требований к проекту с целью избежать тупиковых ситуаций. Следует помнить, что после того как вы поставили цель проекта, руководство будет ожидать от вас ее достижение.

Группа процессов инициации состоит из процессов, способствующих формальной авторизации начала нового проекта.

- Разработка Устава проекта (Develop Project Charter)
- Определение заинтересованных сторон (Identify Stakeholders)

Группа процессов планирования проекта (Planning)

Определяет и уточняет цели и планирует действия, необходимые для достижения целей и содержания, ради которых был предпринят проект. Планирование – это определение ясных и точных задач, и как следствие рабочих заданий, служащих для достижения конечной поставленной цели. Цель может представлять собой решение какой-нибудь проблемы или достижение некоторого состояния или условия, отличного от существующего. На данном этапе команда решает, как она будет достигать цели, поставленной на предыдущем этапе. На данном этапе команда уточняет и детализует цели и результаты проекта, а также состав работ по нему. На основании данной информации формируется календарный план и бюджет, оценивает риски и уточняются заинтересованные стороны. Планирование проекта может потребовать значительных затрат времени, сил и ресурсов, в зависимости от его размера. Опыт показывает, что усилия и ресурсы могут быть потрачены впустую, если не осуществить надлежащего планирования конкретного проекта перед принятием решения о том, реализовывать ли его или нет. Организация проекта является первостепенной задачей по управлению проектом. Необходимо определение владельца проекта – человек или группа, обладает авторитетом по принятию решений, назначить менеджера проекта и сформировать рабочей группы, определить порядок управления и взаимодействия по проекту, а также выделить соответствую-

ющих полномочий руководителю проекта и рабочей группе. Управление проектом за пределами функциональных границ организации одна из задач руководителя проекта. Ему приходится общаться, подавать идеи, вести переговоры, решать проблемы и разрешать конфликты за пределами функциональных, а иногда и географических границ организации. Постановка задачи, обоснование необходимости в проекте и описание его возможностей позволяют сформулировать цель проекта. Такая формулировка может быть очень краткой, но точной. Постановка задач по проекту важна по двум причинам:

- Задача четко определяет, что должно быть сделано для достижения целей
- Задача является событием, срок окончания которого можно определить.

Метод «SMART» помогает сформулировать цели и задачи проекта как:

- Specific – Быть точным при постановке цели
- Measurable – Установить измеримые показатели состояния
- Assignable – Иметь возможность поручить выполнение задания кому-нибудь
- Realistic – Определить, могут ли быть реально выполнены в срок и в рамках выделенных ресурсов

ресурсов

- Time related – Определить временные рамки, т. е. продолжительность ее выполнения

Для достижения поставленной цели, необходимо выполнить несколько основных задач проекта. Эти задачи являются частными целями и представляют собой основные компоненты проекта (иногда используется термин «вехи» или milestone). Частные цели не являются фактическими рабочими заданиями, выполняемыми в рамках проекта, а представляют собой контрольные точки, задающие направление работ. Они точнее формулируются, чем основная цель, и также ориентированы на действия. Для достижения основной цели необходимо реализовать все частные цели.

Выделение ресурсов под проект. Не следует думать, что только деньги являются ресурсами. В качестве основных ресурсов можно выделить следующие:

- финансовые ресурсы (как прямые необходимые для реализации проекта, так и на управление проектом)
- людские ресурсы (кто, когда, как и на какой срок участвует в проекте)
- материальные ресурсы (имеющиеся в наличии, требуемые и т.п.)
- административные ресурсы (полномочия, организация)

Кроме этого в список необходимых ресурсов может быть включены изменения в организационной структуре компании, помещение под офис (для крупных проектов) и т.п.

Из практики участия в проектах, можно выделить два наиболее вероятных сценария выделения ресурсов:

- Руководитель проекта определяет необходимые ресурсы на основе предварительного плана, в котором дается первичная оценка количества ресурсов, необходимых для проекта. Руководитель проекта сможет сформулировать требования к ресурсам и обсудить их с уполномоченным руководителем. Данный сценарий наиболее предпочтительный.

- Необходимые ресурсы назначаются без участия руководителя проекта. Часто из практики, у руководителя проекта может и не быть выбора, независимо от того, достаточно ли выделено ресурсов на проект или нет. Не следует идти на поводу у руководства, давая согласие на уровень поддержки, который явно недостаточен для реализации проекта. Осторожность и здравый смысл должны быть определяющими на этой ранней стадии.

Документ, содержащий обзор проекта, составлен, проанализирован с участием экспертов, а затем подан на рассмотрение высшего руководства организации. Следующий важный

аспект управления проектом является процесс разбиения проекта на составные части, результатом которого является схема разбиения на *рабочие задания (СРРЗ)*. СРРЗ – это иерархическое представление проекта. С помощью этой схемы руководитель проекта определяет задания, которые требуется выполнить, чтобы начать и закончить проект. На этом этапе у руководителя есть цель и ряд задач, которые необходимо выразить через задания и работы, подлежащие выполнению. Четко определенное задание имеет следующие характеристики:

- Его состояние и срок завершения легко определить, оно имеет четко определенное начало и конец
- Понятно, поскольку возможно выполнялось ранее, а время, требуемое для его выполнения, и связанные с ним затраты легко оценить, используя опыт, приобретенный при реализации похожих заданий в прошлом.
- Включает в себя работы, которые поддаются контролю и не зависят от работ, составляющих другие задания
- Как правило, составляет одну непрерывную последовательность работ.

Оценка времени выполнения каждого из заданий, составляющих проект. Время, требуемое для выполнения задания, является случайной величиной. Это значит, что если данное задание выполняется много раз, то можно ожидать, что время его выполнения будет несколько меняться. Это будет справедливо даже для регулярных заданий. Такой разброс объясняется следующими факторами:

- уровнем квалификации сотрудников, выполняющих задание
- использованием различного оборудования
- доступностью материалов
- непредвиденными событиями (болезнями, стихийными бедствиями, авариями, текучестью кадров и т. п.)

Известно, что подобные события могут происходить, но нельзя точно предсказать их появление при реализации конкретного проекта или задания. Тем не менее, их нужно как-то учитывать. Для этого можно использовать метод оценки критичного пути.

Оценка затрат на задание. Существуют четыре основных категории затрат (хотя можно также использовать план бухгалтерских счетов организации), которые можно определить для каждого задания:

- Рабочая сила
- Материалы
- Другие прямые затраты (командировки, телефон, услуги по контрактам)
- Косвенные затраты (накладные расходы)

Далее необходимо определить последовательность выполнения заданий по проекту. Для простых проектов можно выполнять задания по одному по порядку. Другой путь – провести анализ всех заданий и определить, какие из них необходимо завершить до начала выполнения других. Такой анализ позволяет выявить порядок, в котором можно одновременно выполнять несколько заданий. Методом критического пути (МКП) определяет последовательность одновременно выполняемых заданий, который позволяет завершить проект в установленные сроки. Выявление критических заданий, определение критического пути и построение диаграмма приоритетов – представить проект в графическом виде. Для этого необходимо освоить несколько простых правил построения схем проектов. Задание является основной «единицей анализа» в упорядоченной схеме. Задания представляются на схеме в виде прямоугольников, называемых «узлами заданий». Символы, изображенные в прямоугольнике, описывают временные свойства задания. Некоторые из них описывают характеристики самого задания

(например, номер задания), тогда как другие представляют собой расчетные величины (ES, EF, LS, LF), связанные с заданием. Время выполнения проекта – это самый длинный временной путь на схеме. Последовательность заданий, составляющая самый длинный путь, называется критическим путем. До тех пор, пока задания, находящиеся на критическом пути, выполняются в срок, проект идет по графику. Определим теперь четыре расчетных параметра (ES, LS, EF, LF), связанные с каждым узлом задания. Следующие расчетные величины будут использованы для определения времени выполнения проекта и критического пути:

Самое раннее время начала (ES) выполнения задания – это самый ранний момент времени, когда все предшествующие ему задания уже завершены и можно приступить к выполнению данного задания. Время ES для задания, у которого нет предшествующих заданий, условно принимается равным нулю.

Самое раннее время окончания (EF) задания равно времени ES плюс предполагаемое время выполнения задания. Время ES для задания, которому предшествует одно задание, представляет собой время EF для этого предшествующего задания. Время ES для заданий, которым предшествуют два и больше заданий, является максимальным из времен EF для этих предшествующих заданий.

Самое позднее время начала (LS) и самое позднее время окончания (LF) задания – это самые поздние моменты времени, когда можно начать (LS) или закончить (LF) задание, не увеличивая время выполнения проекта в целом.

Чтобы рассчитать эти моменты, будем двигаться по схеме назад. Сначала примем время LF для последнего задания на схеме в качестве времени EF рассматриваемого задания. Время LS для данного задания равно его времени LF минус предполагаемое время выполнения этого задания. Время LF для всех непосредственно предшествующих заданий является минимальным из времен LS для всех заданий, для которых рассматриваемое задание является предшествующим. Необходимо рассчитать еще одну величину, называемую резервом времени для задания.

Резерв времени – это допустимая величина задержки начала или окончания задания, которая не приводит к задержке выполнения проекта в целом. Резерв времени математически представляет собой разность $LS - ES$ (или $LF - EF$, что-то же самое). По определению, последовательность заданий, имеющая нулевой резерв, является критический путь.

Техническое задание по проекту – представляет собой переход от этапа планирования (определение, составление плана) к этапам реализации (организация, контроль, завершение). Как будет показано далее, техническое задание представляет собой фундамент, на котором основана внутренняя согласованность проекта и который обеспечивает базис для принятия всех административных решений. Рассмотрим техническое задание, уделив особое внимание составляющим его частям и их использованию в качестве средств управления. Техническое задание проекта предназначено для того, чтобы получить:

- изложение возникшей проблемы, принимаемого общего подхода к ее решению и ожидаемых в результате выгод;
- полное описание заданий по проекту, необходимых затрат времени и ресурсов;
- такое подробное описание необходимо администрации, чтобы решить, следует ли перейти к этапам реализации проекта
- динамичный инструмент для руководителя проекта и рабочей группы, который будет использоваться для принятия решений в течение всего времени работы над проектом;
- справочную документацию для административного контроля;
- средство, с помощью которого можно ознакомить с проектом и подготовить к его выполнению новых членов рабочей группы по проекту;

• документ-сводку для тех представителей вашей организации, кто, не работая над проектом непосредственно, должны все-таки иметь о нем представление.

Очевидно, что задание – это ключевой документ в проекте. Техническое задание должно быть написано так, чтобы его понимали и использовали и вышестоящая администрация, и руководители проекта, и члены рабочей группы, и другие руководители, а также специалисты, которым необходима эта информация. Составляющие технического задания проекта:

- Название проекта
- Цели проекта
- Руководитель проекта
- Владелец проекта или заинтересованные лица
- Состав рабочей группы

• *Задания* – Этот раздел состоит из трех подразделов: номер задания, краткое, но смысловое имя, присвоенное заданию и описание задания. В нем должно содержаться в (точных выражениях) конкретное изложение содержания работы, которую необходимо выполнить.

- Оценочные даты начала и завершения работ.
- График выполнения проекта
- Смета проекта – Информация о смете в этом отчете обобщается на уровне задания.
- Метрики и критерии достижения целей
- Дополнительные условия

Все проекты имеющие финансовые аспекты получения прибыли, могут потребовать тщательной оценки влияния проекта на доходы и расходы, перед тем как утверждать проект,

План сроков и хода работ – это график, в котором для каждого рабочего пакета указывается срок начала и окончания, а также запланированная продолжительность. Как правило план сроков и хода работ изображается в виде диаграммы Гантта и сетевых графиков, в которых задается логическая последовательность выполнения рабочих пакетов.

В группу процессов планирования входят следующие процессы:

- Разработка плана управления проектом (Develop Project Management Plan)
- План управления содержанием (Plan Scope Management)
- Сбор требований (Collect Requirements)
- Определение содержания (Define Scope)
- Создание иерархической структуры работ – ИСР (Create Work Breakdown Structure – WBS)
- Разработка плана управления расписанием (Develop Schedule Management Plan)
- Определение операций (Define Activities)
- Определение последовательности операций (Sequence Activities)
- Оценка ресурсов операций (Estimate Activity Resources)
- Оценка длительности операций (Estimate Activity Durations)
- Разработка расписания (Develop Schedule)
- Разработка плана управления стоимостью (Develop Cost Management Plan)
- Оценка стоимости (Estimate Costs)
- Определение бюджета (Determine Budget)
- Планирование качества (Plan Quality)
- Разработка плана управления человеческими ресурсами (Develop Human Resource Plan)
- Планирование коммуникаций (Plan Communications)
- Планирование управления рисками (Plan Risk Management)
- Идентификация рисков (Identify Risks)
- Качественный анализ рисков (Perform Qualitative Risk Analysis)

- Количественный анализ рисков (Perform Quantitative Risk Analysis)
- Планирование реагирования на риски (Plan Risk Responses)
- Планирование закупок (Plan Procurements)
- Разработка плана управления заинтересованными сторонами (Develop Stakeholder Management Plan)

Группа процессов реализации проекта (Executing)

На этой фазе происходит собственно основная работа по проекту – написание кода, возведение здания и тому подобное. Следуя разработанным планам начинает создаваться содержание проекта, определённое ранее, проводится контроль по выбранным метрикам. Во второй части данной фазы происходит тестирование продукта, он проверяется на соответствие требованиям заказчика и заинтересованных сторон. В части тестирования выявляются и исправляются недостатки продукта.

Данная фаза включает в себя распределение обязанностей по выполнению рабочих заданий. Создание эффективно работающей группы – это в одинаковой степени и искусство, и наука. Чтобы создать эффективно работающую группу, необходимо принимать во внимание не только техническую квалификацию руководителя проекта и членов рабочей группы, но также их критические роли и взаимоотношения между ними. Выбор руководителя проекта и членов группы не будет совершенным – в любом решении по кадровым вопросам всегда присутствует риск. Главная цель при выборе руководителя проекта – назначить на эту должность человека опытного, компетентного и способного получить конечный результат при соблюдении требований проекта в установленные сроки с имеющимися ресурсами. С этой точки зрения все основные качества эффективно работающего руководителя проекта можно отнести к одной из следующих пяти категорий:

- Образование и опыт
- Лидерство и стратегическое мышление
- Техническая компетентность
- Умение работать с людьми
- Доказанные способности к управлению

Выбор рабочей группы зависит от ряда факторов:

- от задачи и целей проекта
- от характера работы, которая должна быть сделана
- от квалификации, необходимой для найма, назначения, получения полномочий, контроля, связи и выполнения требуемой работы на каждом из этапов проекта
- наличия соответствующих кадров в организации, где будет выполняться проект

Руководителю проекта и членам рабочей группы важно знать, что вновь образованные группы проходят через цикл становление и эволюция рабочей группы, имеющие следующие названия:

Этап формирования – члены рабочей группы знакомятся друг с другом, «ломается лёд» и налаживаются отношения.

Этап привыкания – На этапе привыкания конфликт естественен и неизбежен. Члены рабочей группы проверяют друг друга, появляется чувство границ и устанавливается доверие между ними.

Этап выработки норм поведения – выработки норм поведения разрабатываются приемлемые неписанные правила и нормы поведения, которых придерживаются все сотрудники. Члены рабочей группы знают, что можно ожидать друг от друга в процессе работы.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.