Национальный исследовательский университет
«Высшая школа экономики»

Е. П. Зараменских

Управление жизненным циклом информационных систем

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

ЦЕНТР РАЗВИТИЯ НАУЧНОГО СОТРУДНИЧЕСТВА

Е.П. ЗАРАМЕНСКИХ

УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ ИНФОРМАЦИОННЫХ СИСТЕМ

МОНОГРАФИЯ

НОВОСИБИРСК
2014

УДК 004
ББК 73.05
З 34

Рецензенты:

Ж.А. Мингалева – доктор экономических наук, профессор;
В.В. Вороной – кандидат технических наук.

Зараменских Е.П.
З 34 Управление жизненным циклом информационных систем: монография / Е.П. Зараменских. – Новосибирск: Издательство ЦРНС, 2014. – 270 с.

ISBN 978-5-00068-118-3

В монографии приведены результаты исследования, посвященного управлению жизненным циклом информационных систем, а также анализу стандартов, сводов знаний и корпоративных методик, использующихся в ИТ-проектах. Приведены характеристики фаз ЖЦИС из практики управления ИТ-проектами, а также практические рекомендации по управлению такими проектами. Книга предназначена для научных работников, сотрудников научно-технических предприятий и работников государственных органов управлений, а также студентов, аспирантов, слушателей бизнес-школ повышения квалификации и переподготовки кадров. Книга содержит практические рекомендации для руководителей ИТ-проектов, а также сотрудников компаний, занимающихся проектной деятельностью в области ИТ-проектов.

УДК 004
ББК 73.05

ISBN 978-5-00068-118-3

© Е.П. Зараменских, 2014

ОГЛАВЛЕНИЕ

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ 7

ВВЕДЕНИЕ 9

ГЛАВА 1. **ИНФОРМАЦИОННЫЕ СИСТЕМЫ В СОВРЕМЕННОМ МИРЕ** 11

1.1. История эволюции ИС 11

1.2. Жизненный цикл ИС и ПО 14

ГЛАВА 2. **ЖИЗНЕННЫЙ ЦИКЛ ИС** 16

2.1. Модели ЖЦИС 17

2.1.1. *Каскадная модель* 17

2.1.2. *Каскадная модель с промежуточным контролем* 18

2.1.3. *Спиральная модель* 19

2.1.4. *Модель разработки через тестирование (V-модель)* 21

2.2. Стандарты жизненного цикла ИС 22

2.2.1. *ГОСТ 34.601-90* 22

2.2.2. *ISO/IEC 12207:2008 (ГОСТ Р ИСО/МЭК 12207-2010)* 23

2.2.3. *ISO/IEC 15288 (ГОСТ Р ИСО/МЭК 15288-2005)* 25

2.3. Системный подход 27

2.4. Фазы ЖЦИС и специфика каждой из них 30

2.4.1. *Планирование проекта* 30

2.4.1.1. *Экспресс-обследование* 31

2.4.1.2. *Технико-экономическое обоснование* 33

2.4.1.3. *Оценка целесообразности проекта (TELOS)* 35

2.4.1.4. *Выбор программного решения* 36

2.4.2. *Анализ и постановка задачи* 38

2.4.2.1. *Информационное обследование предприятия* 39

2.4.2.2. *Описание бизнес-процессов* 42

2.4.2.2.1. *Основные нотации / методологии моделирования* 43

2.4.2.2.2. *Программные продукты моделирования деятельности организации* 51
2.4.2.3. *Сбор требований* 57
2.4.2.4. *Подготовка технического задания* 63
2.4.3. *Проектирование* 64
2.4.3.1. *Техническое проектирование* 65
2.4.3.2. *Рабочее проектирование / прототипирование при заказной разработке* 69
2.4.4. *Разработка* 71
2.4.4.1. *Закупка ПО* 71
2.4.4.2. *Настройка конфигураций* 72
2.4.4.3. *Создание ролей пользователей* 75
2.4.4.4. *Миграция данных* 77
2.4.4.5. *Разработка контрольного примера* 79
2.4.4.6. *Тестовая эксплуатация* 80
2.4.4.7. *Доработка по результатам тестирования* 83
2.4.4.8. *Прием результатов испытаний* 84
2.4.5. *Развертывание и внедрение* 85
2.4.5.1. *Закупка и настройка требуемой ИТ-инфраструктуры* 86
2.4.5.2. *Ввод начальных остатков* 89
2.4.5.3. *Обучение пользователей* 90
2.4.5.4. *Развертывание системы на рабочих местах* 92
2.4.5.5. *Основные виды тестирования* 94
2.4.5.6. *Опытно-промышленная эксплуатация* 96
2.4.5.7. *Приемо-сдаточные испытания* 98
2.4.6. *Эксплуатация* 99
2.4.7. *Сопровождение эксплуатации* 100
2.4.7.1. Авторский надзор 100
2.4.7.2. *Техническая поддержка* 101
2.4.7.3. *Постгарантийное сопровождение* 102
2.4.8. *Модернизация* 108

2.4.8.1. *Стратегии управления legacy-системами* 108
2.4.8.2. *Виртуализация как стратегия модернизации решений* 111
2.4.8.3. *Особенности проектов по модернизации* 113
2.4.9. *Утилизация* 115
2.4.9.1. *Технические аспекты* 115
2.4.9.2. *Организационные аспекты* 116
2.4.9.3. *Коммерческие аспекты* 117
2.4.9.4. *Юридические вопросы* 117
ГЛАВА 3. **ОСОБЕННОСТИ УПРАВЛЕНИЯ ПРОЕКТАМИ ПО ВНЕДРЕНИЮ КИС** 118
3.1. Создание ИС в соответствии с методологиями и стандартами 121
3.1.1. *Корпоративные методологии* 121
3.1.1.1. *IBM (Rational Unified Process, RUP)* 121
3.1.1.2. *Microsoft (Microsoft Solution Framework, MSF)* 123
3.1.1.3. *On Target* 126
3.1.1.4. *Microsoft Dynamics Sure Step и Microsoft Business Solutions Partner Methodology* 127
3.1.1.5. *SAP (Accelerated SAP)* 130
3.1.1.6. *Oracle (Oracle Unified Method, OUM)* 130
3.1.1.7. *Oracle / PeopleSoft One Methodology* 132
3.1.2. *Индустриальные стандарты и методологии* 133
3.1.2.1. *Agile* 133
3.1.2.2. *SCRUM* 135
3.1.2.3. *RAD* 136
3.1.2.4. *XP* 137
3.2. Сервисный подход к эксплуатации ИС 137
3.3. Элементы управления корпоративными ИТ 141
3.3.1. *CobiT* 141
3.3.2. *ITIL* 148
3.4. Проектное управление 151
3.4.1. *Российские и международные стандарты* 151

3.4.1.1. *SWEBOK* 151
3.4.1.2. *PMBOK* 153
3.4.1.3. *PRINCE2* 159
3.4.1.4. *ISO 21500:2012* 162
3.4.1.5. *ГОСТ Р 54869-2011* 164
3.4.1.6. *ISO 10006* 165
3.4.2. *Менеджмент ЖЦ ИС в контексте проектной деятельности* 166
3.4.2.1. *Управление стейкхолдерами* 166
3.4.2.2. *Управление человеческими ресурсами* 168
3.4.2.3. *Управление финансами* 175
3.4.2.4. *Управление коммуникациями* 180
3.4.2.5. *Управление качеством* 183
3.4.2.6. *Управление содержанием* 192
3.4.2.7. *Управление рисками* 193
3.4.3. *Управление программой проектов* 204
3.4.4. *Сбалансированная система показателей (BSC)* 208
ЗАКЛЮЧЕНИЕ 211
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 213
ПРИЛОЖЕНИЯ 216

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

ARIS (ARchitecture of Integrated Information Systems) – архитектура интегрированных программных систем.

BCP (Business continuity plan) – план обеспечения непрерывности бизнеса.

BPMN (Business Process Modeling Notation) – нотация моделирования бизнес-процессов.

BSC (Balanced Scorecard) – сбалансированная система показателей.

CobiT (Control Objectives for Information and Related Technology) – задачи информационных и смежных технологий.

DRP (Disaster recovery plan) – план аварийного восстановления.

eEPC (Extended Event-driven Process Chain) – расширенная событийная цепочка процессов.

ISACA (Information Systems Audit and Control Association) – международная ассоциация аудита и контроля за информационными системами.

IPMA (International Project Management Association) – некоммерческая профессиональная ассоциация по управлению проектами.

IT (Information Technology) – информационные технологии.

ITIL (IT Infrastructure Library) – библиотека инфраструктуры информационных технологий.

ITSM (IT Service Management) – управление IT-услугами.

NPV (Net Present Value) – чистая приведенная стоимость проекта.

PMBOK (Project Management Body of Knowledge) – свод знаний по управлению проектами.

PRINCE2 (PRojects IN Controlled Environments) – методология управления программами / проектами в организации.

ROI (Return On Investment) – возврат инвестиций.

SLA (Service Level Agreement) – соглашение об уровне предоставления услуг.

SWEBOK (Guide to the Software Engineering Body of Knowledge) – руководство к методическому справочнику по программной инженерии.

TCO (Total Cost of Ownership) – совокупная стоимость владения.

UML (Unified modeling language) – универсальный язык моделирования.

АРМ – автоматизированное рабочее место.

АС – автоматизированная система.

АСУ – автоматизированная система управления.

ГИС – географическая информационная система.

ЖЦ – жизненный цикл.

ЖЦИС – жизненный цикл информационной системы.

ИТ – информационные технологии.
КИС – корпоративная информационная система.
КСПД – корпоративные сети передачи данных.
КХД – корпоративное хранилище данных.
ЛВС – локальная вычислительная сеть.
ИС – информационная система.
ОПЭ – опытно-промышленная эксплуатация.
ОС – операционная система.
ПО – программное обеспечение.
ПР – программное решение.
ПС – программные средства.
СУБД – система управления базой данных.
СХД – система хранения данных.
ТЗ – техническое задание.
ТЭО – технико-экономическое обоснование.
ЦОД – центр обработки данных.
ЦП – центральный процессор.
ЭВМ – электронная вычислительная машина.
ЭЦП – электронная цифровая подпись.

ВВЕДЕНИЕ

Активное использование информационных технологий в бизнесе дает значительный результат для повышения эффективности деятельности. Сегодня без использования информационных технологий сложно представить как хозяйственную деятельность предприятий, так и процессы управления этими предприятиями. Но еще пока высокий процент неуспешных проектов по созданию, внедрению, эксплуатации и модернизации информационных систем заставляет искать методы и пути снижения таких результатов.

Гарантированное получение результата – это цель, к которой стремятся все команды, как с заказывающей стороны, так и со стороны исполнителей (подрядчиков). Понимание необходимых и достаточных шагов, соблюдение обязательной этапности в последовательности действий – это и есть понимание жизненного цикла информационных систем. Только получение запланированного результата с заданным качеством, в установленные сроки и в рамках согласованного бюджета есть результат, который устраивает обе стороны.

В первой главе рассмотрены основные вехи в истории развития информационных систем, их составные части и характеристики, знание которых необходимо для создания успешно выполняющих свои задачи систем.

Вторая глава рассматривает модели жизненного цикла информационных систем, стандарты, определяющие фазы ЖЦИС. В этой главе даны характеристики каждого из этих этапов в отдельности.

И, наконец, поговорив о содержании этапов жизненного цикла как проектах, необходимых для успешного создания и эксплуатации ИС, третья глава содержит основные аспекты управления ЖЦ ИС, каким образом организовать управление данными проектами как на операционном уровне, так и на более высоком уровне целой программы проектов (от формирования замысла ИС до передачи ее в эксплуатацию). В третьей главе приведены корпоративные методологии, а также индустриальные стандарты управления жизненным циклом информационных систем. В этой же главе приведен обзор российских и международных стандартов проектного управления.

В основе настоящего исследования практический опыт автора по управлению компанией «ИнтелГрупп», которая с 1994 года занималась созданием, внедрением, эксплуатацией корпоративной информационной системой «ТЕКТОН», а также опыт чтения лекций и ведения научных семинаров по теме «Управление жизненным циклом информационных систем» в магистерской программе «Бизнес-информатика» Национального исследовательского университета «Высшая школа экономики».

Автор выражает благодарность своим друзьям и коллегам, с кем вместе в течение 20 лет в компании «ИнтелГрупп» выполнял проекты для корпоративных заказчиков и формировал практический опыт управления жизненным циклом информационных систем: Д.Ф. Глямшину, Ю.Е. Лазареву, К.Е. Веригe, А.Г. Тараненко, Е.Ю. Кислицину, Г.Н. Денищенко, И.Е. Артемьеву, В.В. Ляпину, И.А. Лавриненко, Н.П. Нестеренко, Н.А. Филатову, О.Е. Андрейченко, Л.И. Петуховой, М.В. Подгузову, Р.А. Набиеву, А.В. Булдыгерову и другим.

ГЛАВА 1

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В СОВРЕМЕННОМ МИРЕ

1.1. История эволюции ИС

«Кто владеет информацией – владеет миром» – эта фраза, приписываемая иногда Черчиллю, иногда Ротшильду, как никогда подходит для описания современной действительности. Информация – ценнейший актив компании, и для получения благодаря ему выгоды и конкурентных преимуществ компании необходимы информационные системы соответствующего уровня.

Информационная система (ИС) – необходимая для сбора, обработки, поиска, хранения, передачи и предоставления данных совокупность программного, аппаратного и организационного обеспечения, а также методологических и информационных ресурсов.

На сегодняшний день существует огромное многообразие систем, и любая компания вне зависимости от своих масштабов, сферы деятельности, сложности и уровня зрелости собственных процессов может выбрать уже представленное на рынке решение либо создать систему по своим требованиям. Но в любом случае, любая информационная система должна удовлетворять требованиям и интересам пользователей и заказчика в рамках заранее определенной функциональности.

За счет чего это достигается? ИС имеет определенный набор свойств / характеристик, которые отличают ее от других методов сбора / обработки / хранения информации.

- Использование математических методов и алгоритмов.
- Автоматическая консолидация данных и возможность их представления в управленческой перспективе (к примеру, в виде отчетов и панелей мониторинга).
- Наличие инструментов получения, хранения и организации доступа к данным.
- Клиентский интерфейс с возможностью организации работы под конкретную роль пользователя.
- Организация профилей пользователей с разграничением доступа к информации.
- Широкие функциональные возможности, в том числе по регистрации, обработке, поиску информации.

Список этих свойств и характеристик регулярно расширяется по мере развития современных информационных систем и роста требований к ним. Ведь все, что мы видим, говорим, думаем, делаем – а также то, что происходит в мире является информацией, обрабатываемой нами и окружающими. Сейчас для ее хранения, интерпретации, обработки, передачи и выдачи результата используются информационные системы – но для понимания экономических, социальных, технологических предпосылок этого рассмотрим историю обработки и дальнейшей передачи данных в течение последнего столетия. Причем сфокусируемся на дистанционной передаче данных для более четкого формирования представления об историческом контексте.

	Основные вехи	Методы передачи информации
До конца XIX в.	-	**«Ручные»** методы. Инструменты: перо, чернильница, книга. Почтовая коммуникация с использованием пакетов, писем, депеш
1900-1940		**«Механические»** методы. Инструменты: печатная машинка, диктофон, телефон, технически модернизированная почта
1940-1950	Машины электромеханического типа	
1950-1964	Операционные системы и дисковая технология, развитие функциональности языков программирования. Электромеханические бухгалтерские машины, предназначенные для обработки бумажных документов	**«Электрические»** методы. Инструменты: первые ЭВМ, ПО, ксероксы, портативные средства аудиозаписи, электронные печатные машинки.
1965-1974	Первое ПО, ориентированное на менеджмент организации	
1975-1980	Первые микропроцессоры и интерактивные дисплеи, электронная почта, текстовая обработка	«**Электронные**» методы. Инструменты: обновленные ЭВМ, АСУ, информационно-поисковые системы.
1981-1985		
1986-1990	Развитие персональных / настольных компьютеров. Развитие электронных баз данных и локальных сетей	«**Компьютерные**» методы. Инструменты: персональный компьютер, ПО, персонализированные АСУ, системы поддержки принятия решений.
1991-1996	Повсеместное распространение Интернета, коммуникации в режиме реального времени.	
1997-2000		
2000-н.в.	Развитие инфраструктуры телекоммуникаций	

До второй половины XIX века основной целью используемых «информационных технологий» (то есть письменности и почтовых коммуникаций) в силу отсутствия вычислительных машин было лишь представление информации в необходимой форме. При этом подобную информацию потенциально в дальнейшем можно было бы передавать другим лицам (часто отделяемым расстоянием или временем, последующим поколениям).

В течение последующего века, вплоть **до 1940-х годов** (в некоторых странах **до 1960-х годов**) применялись механические методы, которые бы не просто представляли информацию в необходимой форме, но и делали бы это с использованием удобных и практичных средств, сокращающим время на создание или передачу данных (печатная машинка и телефон соответственно).

Начинали развиваться системы обработки потоков документов, но стал актуальным вопрос хранения этих данных и более эффективного / оперативного создания копий. О таком термине, как «программное обеспечение», речь еще не шла, и единственной возможностью для движения в сторону большей эффективности и производительности было развитие технических / аппаратных средств. В это время развивалась исключительно аппаратная составляющая информационных систем (если исходить из данного ранее определения).

Переход от механических к электрическим средствам, завершившийся в 1970-е годы, обусловил смещение фокуса технологий с представления информации на ее непосредственное формирование (вычисления при помощи ЭВМ, копирование при помощи ксерокса, запись звука без привязки к конкретному месту при помощи переносного диктофона). Однако можно сказать, что развитие программного обеспечения того времени отстает от развития аппаратных средств (к примеру, эпоха ЭВМ серии IBM/360), а для обработки данных используются вычислительные центры коллективного пользования, благодаря чему ряд рутинных действий оптимизируется. Тем временем, этих аппаратных возможностей по-прежнему недостаточно и остается актуальной проблема обработки больших объемов информации.

Критерием эффективности технологий **1960-х годов** являлась разница между затраченными на создание технологии средствами и полученными в процессе ее использования выгодами. Однако все операционные системы и ПО под них создается профессиональными программистами без участия пользователей. А подобный разрыв в коммуникациях не давал возможности разработчикам определить необходимости и собрать требования пользователей, которые в свою очередь (не будучи знакомы со спецификой предлагаемых систем и принципами их создания) не могли адекватно оценить функциональные возможности системы для полноценного использования и дать свои комментарии по совершенствованию, опираясь на свой опыт и знание процессов (например, производственных). Таким образом, организационное обеспечение систем было недостаточно развито, так как основное внимание уделялось поиску путей обеспечения соответствия между уровнями развития программных и аппаратных решений.

С середины 1970-х – начале 1980-х годов процесс создания информации и информационных систем значительно изменился благодаря появлению персональных компьютеров. Вне рамок какой-либо корпорации пользователи получили возможность самостоятельной работы с ПК, и на передний план вышла проблема удовлетворения стремительно растущих потребностей пользователей, использующих компьютер в личных целях. Информационные системы стали фокусироваться на нуждах индивидуальных пользователей (в отличие от предыдущего подхода с ориентацией на корпоративных пользователей, для которых создавались отчеты с целью поддержки принятия стратегических решений).

К тому времени основные рутинные функции (к примеру, ввода данных и их предоставления) уже были реализованы в системах и следующим шагом предстояло сфокусироваться на обработке всех этих данных. Создаются информационно-поисковые системы, совершенствуются системы формирования управленческой информации, и даже появляются первые системы бизнес-аналитики. Барьер между разработчиками и пользователями начинает стираться в силу важности участия последних в процессе создания ИС. С технической точки зрения данный период отличает еще то, что постепенно начинает применяться первая децентрализация, связанная с локальными базами на рабочем месте у специалиста (хотя в основном по-прежнему применяется централизованная обработка данных). Наступает время офисных, пользовательских информационных систем.

Со второй половины 1980-х годов локальные и глобальные компьютерные сети стали находить крайне широкое применения во всех областях жизни человека. Информационные системы стали не только решать различные стратегические задачи, но и применять в этих целях телекоммуникационные методы. Применяются системы поддержки принятия решений на разных уровнях производства применяются и развиваются по-прежнему, однако стал важен и фокус на получении конкурентного преимущества и взаимодействии с другими контрагентами на рынке, таким образом переходя от акцента на внутренних процессах организации к внешней кооперации.

Развитие в **1990-х годах** глобальной сети привело к переносу некоторыми компаниями многих операций и даже всего бизнеса в интернет-среду. Возникли системы электронной коммерции, интернет-магазины и «виртуальные» предприятия. Тем временем, нельзя не отметить ряд задач, появившихся с развитием сетевых технологий и требующих решения, а именно – проблемы доступа к стратегической информации, обеспечения информационной безопасности и разработки единых стандартов.

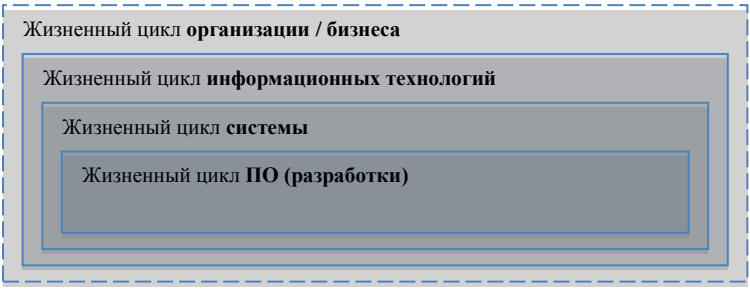
1.2. Жизненный цикл ИС и ПО

Показательно, что при рассмотрении основных вех истории информационных систем, в разные периоды фокус смещался с одного элемента ИС на другой. Так, первые десятилетия развивалось аппаратное, техническое обеспечение, затем со стремительным ростом масштабов разработки программных средств потребовалось срочно принимать необходимые организационные меры, чтобы обеспечить соответствие квалификации и компетенций специалистов минимальным требованиям к пользователям систем. Соответственно, различные элементы информационных систем развивались по разным сценариям, а значит, они имеют и свою специфику жизненных циклов?

Наиболее часто ключевые стороны, задействованные в создании и эксплуатации информационной системы, ассоциируют ее с программным обеспечением, лежащем в основе этой ИС. Тогда в чем основное отличие жизненного цикла информационной системы от жизненного цикла программного обеспечения?

Ответ начинается с самого определения информационной системы, охватывающего большее количество аспектов, нежели просто ПО. Программное обеспечение может быть модернизировано, обновлено до новых версий несколько раз в рамках одного жизненного цикла информационной системы. Однако это будет все та же информационная система, возможно, с теми же инфраструктурными и организационными ресурсами. Могут приходить и уходить сотрудники, обновляться системы хранения данных, но это лишь части информационной системы, которая все равно будет выполнять свои основные функции до момента вывода из эксплуатации.

Интересная для рассмотрения концепция уровней жизненного цикла была предложена в 2005 году Скоттом Амблером, автором практик гибкого моделирования Agile Modeling и концепции Enterprise Unified Process.



Логика рассуждений в данном случае следующая: жизненный цикл бизнеса включает всю деятельность ИТ-департамента, в том числе по разработке, развертыванию, поддержке и сопровождению информационных систем, частью которых является программное обеспечение. Но для того, чтобы проверить эту концепцию, необходимо рассмотреть основные аспекты жизненного цикла ИС подробнее, фокусируясь на основных ее стадиях и их содержании.

ГЛАВА 2

ЖИЗНЕННЫЙ ЦИКЛ ИС

Как уже было сказано, **жизненный цикл** информационных систем представляет собой непрерывный процесс из определенных этапов, начинающийся в момент принятия решения о необходимости создания ИС до отказа от ее использования.

Жизненный цикл информационной системы – непрерывный процесс, началом которого становится момент принятия решения о необходимости системы, а завершением – ее изъятие из эксплуатации. Этапы создания системы до момента ввода в эксплуатацию могут рассматриваться как самостоятельные проекты, каждый из которых имеет конкретный результат и ограничения.

А так как процесс создания и конфигурирования для различных информационных систем включает в себя один набор этапов, то можно говорить о **моделях жизненного цикла.**

Модель ЖЦ ИС – комбинация последовательности этапов жизненного цикла и переходов между ними, необходимых для гарантированного достижения поставленной для реализации проекта цели.

Сами фазы жизненного цикла фиксированы и для различных отраслей человеческой деятельности, по сути, одинаковы:

- Замысел (планирование проекта).
- Анализ и постановка задачи.
- Проектирование.
- Разработка.
- Развертывание и внедрение.
- Эксплуатация.
- Поддержка.
- Модернизация.
- Утилизация.

Можно говорить о том, что средняя продолжительность подобного цикла составляет порядка 15 лет, однако следует учитывать что в зависимости от огромного числа различных факторов, обусловленных спецификой предприятия, отрасли, самой информационной системы сроки физического и морального старения техники и ПО будут значительно отличаться. А значит, еще при проектировании ИС необходимо четко представлять себе возможности ее дальнейшей модернизации, в том числе факторы, которые могут вызвать эту необходимость.

Важно понимать, что переход на новые программные решения – это не утилизация старой и проектирование новой системы. Измениться может практически все: форматы данных, работающий с системой персонал, поддерживающая инфраструктура. Остается только информация, и именно она является связующим звеном, позволяющим говорить о работе с одной и той же информационной системой. Если в системе последние десять лет хранились и обрабатывались данные по незавершенному производству, то могут приниматься решения о переходе на другую платформу, выбор решения другого вендора, для чего проведена конвертация данных, но она все равно останется информационной системой работы с производственными данными.

Порядка трети ИС прекращают свое существование еще на этапе проектирования, причиной чего часто становится несоответствие методов управления проектами (в том числе, анализа) сложности самого проекта, которая постоянно возрастает в условиях рыночной экономики, где число стейкхолдеров постоянно увеличивается, ограничения по срокам и стоимости возрастают, появляются новые конкурирующие компании и продукты. Не всегда уровень выбираемых компаниями технологий анализа / проектирования систем, как и методик управления проектом внедрения соответствует специфике бизнеса и возрастающим требованиям к автоматизации процессов.

Рассмотрим основные модели жизненного цикла, позволяющие при определении правил и условий перехода на следующую стадию нивелировать риски и оптимизировать процесс создания и передачи системы в эксплуатацию.

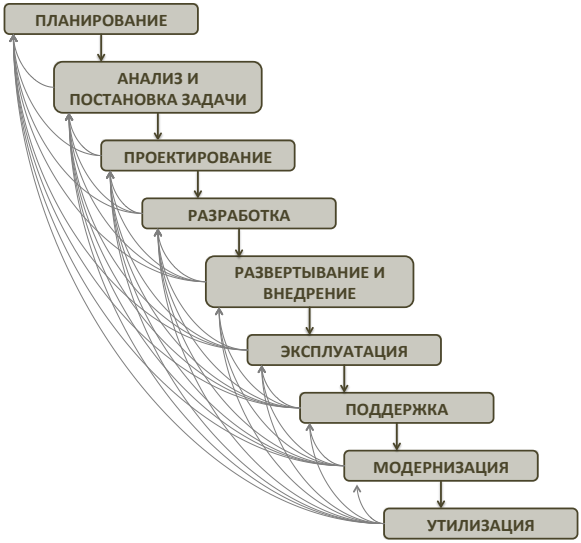
2.1. Модели ЖЦИС

2.1.1. Каскадная модель

Каскадная модель жизненного цикла, также называемая моделью «водопада» (waterfall), была разработана еще в 80-х годах, и на протяжении многих лет она считалась стандартом для разработки ПО. Данная модель характеризуется тем, что этапы строго последовательны и переход между ними невозвратный. Это означает, что в рамках каскадной модели переход к следующему этапу (например, от проектирования и сбора требований к разработке и развертыванию) может произойти только по завершении предыдущего этапа. Модель «водопада» была применена одной из первых и одно из ее основных достоинств в возможности планирования сроков и стоимости каждого этапа – однако, на практике разработка системы почти никогда не проходит строго в соответствии с жесткой заранее продуманной схемой. В частности, это касается сбора требований, так как реально при старте проекта требования бывают определены только частично и в дальнейшем уточняются, изменяются и дополняются. К тому же, если изначально требования были определены неточно, высока вероятность того, что система не будет удовлетворять потребностям заказчика.

2.1.2. Каскадная модель с промежуточным контролем

В качестве одной из вариаций каскадной модели для того, чтобы предусмотреть возможность возвращения к предыдущим этапам для внесения определенных изменений и пересмотра отдельных вопросов, была создана **каскадная модель с промежуточным контролем**.



Она предполагает увеличенное время, отведенное на разработку, за счет проведения промежуточных корректировок между фазами жизненного цикла. В свою очередь, это снижает риски получения некачественного продукта на выходе и повышает надежность системы в целом.

Важно отметить, что согласование результатов в двух описанных моделях происходит только по окончании внедрения – а значит, повышается вероятность получения программного продукта, который морально устареет либо не будет востребован рынком. Еще больше увеличивают риски возможные неточности в исходном техническом задании. В итоге можно говорить о проблеме определенной задержки в получении результата, которая не может быть решена в «каскадном» варианте разработки и внедрения системы.

2.1.3. Спиральная модель

Для нивелирования рисков, связанных с вышеописанной проблемой, была создана **спиральная модель** (еще называемая **итерационной**). Фазы жизненного цикла непоследовательны, то есть допустимо (но не обязательно!) начало работ над следующим этапом до завершения предыдущего. Таким образом, суть спиральной модели состоит в возможности прохождения всех этапов жизненного цикла системы в несколько итераций, каждый раз создавая новый прототип и проверяя актуальность требований, по которым он создавался, внося технические доработки в интерфейс и функциональность. Подобная гибкость позволяет использовать модель на предприятиях любого масштаба.

Прототип крайне важен в ситуациях, когда требуется разъяснить и уточнить требования, выбрать концептуальное решение или даже в целом определить целесообразность реализации проекта. При этом сам прототип может либо моделировать исключительно пользовательский интерфейс, либо архитектурную сторону системы (логика обработки и хранения данных).

Это означает, что процесс создания системы и само управление проектом будет более гибким и управляемым, с совершенствованием системы на каждом «витке спирали», то есть при выпуске каждой версии. Уменьшаются риски (в том числе, финансовые) для заказчика и спонсора системы, которые могут отказаться от проекта еще на этапе показа первого прототипа в случае абсолютного его несоответствия ожиданиям и потребностям (либо в случае изменения рыночной ситуации).

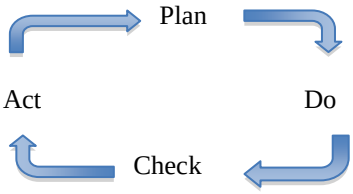
Как правило, подобная модель применяется при разработке нетиповых систем, предоставляя возможность оперативно создать прототип программного продукта для проверки его работоспособности с пользователями и соответственно, быстрого получения комментариев и замечаний к системе.

Цикл Шухарта-Деминга

Определенным прообразом спиральной модели ЖЦ стал классический цикл управления Деминга (PDCA-цикл).

PDCA (Plan – Do – Check – Act) – Цикл качества, еще называемый циклом Деминга. Этот алгоритм управления предполагает четыре основных шага «Планирование – Действие – Проверка – Корректировка» и помимо стандартной области применения в менеджменте предприятия возможно использовать его в управлении внедрением и развитием информационной системы. Таким образом, этап планирования может начинаться как при первичном проектировании системы, так и при старте ее модернизации, когда, как мы говорили, снова необходимо пройти все этапы жизненного цикла системы.

Графически цикл Деминга может быть представлен следующим образом:



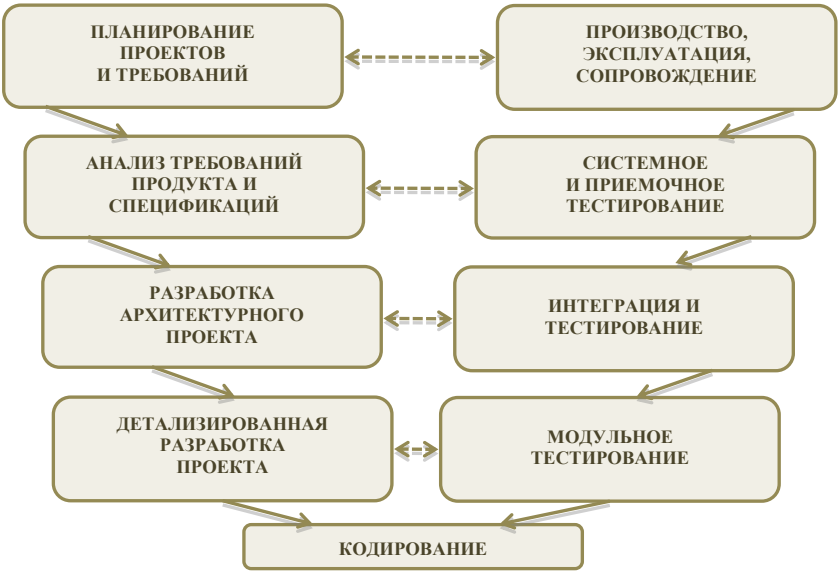
При планировании появляется возможность совершенствования процессов и операций, определения и выделения необходимых для них ресурсов. По окончании следующего этапа (выполнения запланированных работ) обязательно проводится мониторинг результатов и сбор данных для аналитики и определения соответствия плановым показателям для принятия дальнейших мер по корректировке проводимых активностей и принципов распределения ресурсов – что, как мы видели, и является основной идеей итерационной модели создания ИС.

Итерационную / спиральную модель невозможно применять для областей, в которых невозможно предварительное тестирование продукта, обладающего неполной функциональностью – в частности, военные разработки,

авиастроение, атомная энергетика, ведь любая ошибка стоит жизни. Однако при разработке бизнес-ориентированной информационной системы собственными силами ИТ-подразделения компании данная модель применяется достаточно часто, хотя важно принимать во внимание возможное сопротивление со стороны пользователей, которые хотят видеть требующуюся именно им функциональность в первом же релизе, а не втором или третьем. Соответственно, на первый план выходит необходимость грамотного управления ожиданиями пользователей.

2.1.4. Модель разработки через тестирование (V-модель)

Приближенная по своей сути к практикам PRINCE2, **V-модель разработки через тестирование** была разработана еще в конце 1980-х годов ведомствами Германии и США, и до сих пор является стандартом немецких правительственных и оборонных проектов. Основной ее принцип состоит в постепенном возрастании степени детализации проекта с течением времени и одновременном проведении «горизонтальных» итераций. Таким образом, результаты каждой из фаз левой стороны буквы V влияют на тестирование и компоновку проекта с правой стороны буквы V: приемо-сдаточные испытания основываются на проведенном анализе требований, интеграционное тестирование – на высокоуровневом описании архитектуры, модульное тестирование – на архитектуре, интерфейсах, алгоритмах и прочих элементах детализированных требованиях к системе.



Важна гибкость данной модели, так как по сути она адаптируема под любой тип организации. Промежуточные результаты проверяются на ранних стадиях и минимизация рисков достигается благодаря простому соотнесению фаз / итераций. V-модель не рассматривает непосредственно стадию обслуживания и утилизации, учитывая лишь активности по подготовке к ним.

Рассмотрев основные особенности моделей жизненного цикла, перейдем к фазам **проекта по созданию и внедрению** интегрированной **системы управления** как совокупности этапов по созданию, настройке, доработке и внедрению отдельных функциональных модулей системы, выполнение которых необходимо и достаточно для создания системы управления, соответствующей заданным требованиям.

2.2. Стандарты жизненного цикла ИС

2.2.1. ГОСТ 34.601-90

ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

Отличается высокой степенью формализации (как и ГОСТы 19-й серии) и по умолчанию, таким образом предполагает каскадный подход. На сегодняшний день ГОСТ многократно становился основой для доработок и частичного использования в других стандартах / методологиях и в целом в исходном виде не является исчерпывающим как единственный источник информации для выполнения проекта разработки / внедрения.

Стадии	Этапы
Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС 1.2. Формирование требований пользователя к АС 1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)
Разработка концепции АС	2.1. Изучение объекта 2.2. Проведение необходимых научно-исследовательских работ 2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя 2.4. Оформление отчета о выполненной работе
Техническое задание	3.1 Разработка и утверждение технического задания на создание АС
Эскизный проект	4.1. Разработка предварительных проектных решений по системе и ее частям 4.2. Разработка документации на АС и ее части
Технический проект	5.1. Разработка проектных решений по системе и ее частям 5.2. Разработка документации на АС и ее части 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации

Стадии	Этапы
Рабочая документация	6.1. Разработка рабочей документации на систему и ее части 6.2. Разработка или адаптация программ
Ввод в действие	7.1. Подготовка объекта автоматизации к вводу АС в действие 7.2. Подготовка персонала 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями) 7.4. Строительно-монтажные работы 7.5. Пусконаладочные работы 7.6. Проведение предварительных испытаний 7.7. Проведение опытной эксплуатации 7.8. Проведение приемочных испытаний
Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами 8.2. Послегарантийное обслуживание

Благодаря своей структурированности, ГОСТ 34.601-90 до сих пор служит самодостаточной базой для адаптации, которую можно адаптировать к конкретным условиям деятельности предприятия. Приложение к стандарту содержит детальное описание работ, включая списки формируемых по завершении этапа документов.

ПРИМЕР

На этапе 7.6 «Проведение предварительных испытаний» осуществляют:

а) Испытания АС на работоспособность и соответствие техническому заданию в соответствии с программой и методикой предварительных испытаний;

б) Устранение неисправностей и внесение изменений в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний;

в) Оформление акта о приемке АС в опытную эксплуатацию.

Также в стандарте приведен список основных типов организаций, участвующих в работах по созданию АС, что позволяет сформировать понимание сути процесса.

ПРИМЕР

Организация-заказчик (пользователь), разработчик, поставщик, генпроектировщик, …

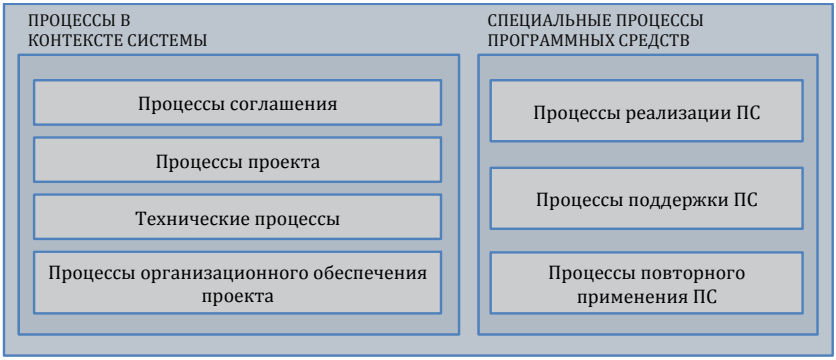
2.2.2. ISO/IEC 12207:2008 (ГОСТ Р ИСО/МЭК 12207-2010)

Международный стандарт: **ISO/IEC 12207:2008 Information technology – Software life cycle processes (Информационные технологии. Процессы жизненного цикла программного обеспечения).**

Российский аналог: **ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».**

Базируясь на процессном подходе, ISO 12207 определяет необходимость документирования основных результатов процесса, но не ограничивает их

содержание и тем более последовательность, а также не противоречит применению итераций в разработке. Данный стандарт стал основой для дальнейшей детализации в некоторых методологиях разработки ПО (в частности, Rational Unified Process), однако сам по себе лишь устанавливает структуру основных, вспомогательных и организационных процессов ЖЦ **программных средств**, определяя необходимые в их рамках работы и задачи. Таким образом формируется единое понимание жизненного цикла (и единая терминология) между заказчиком, разработчиком / подрядчиком и другими стейкхолдерами. С другой стороны, ISO 12207:2008 рассматривает лишь программные средства и соответствующие организационные процессы, не рассматривая аппаратную составляющую.



В РФ был разработан и принят идентичный ISO 12207 стандарт **ГОСТ Р ИСО/МЭК 12207-2010.** Основной идеей разработчиков ГОСТ 12207 являлось создание единого общекорпоративного стандарта, которым было бы возможно воспользоваться при возникновении любой задачи из тех, которые описаны в документе (будь это обучение пользователей, поставка ПО или любая другая активность в рамках ЖЦ).

Стандарт предполагает, что процессы состоят из работ, для которых определены задачи (а также цели и результаты). Тем не менее, допускается адаптация процессов к особенностям организации (например, при больших масштабах проекта изменять состав определенных задач или работ). Как правило, это возможно сделать в рамках существующих на предприятии процессов.

ПРИМЕР

Говоря о процессе поставки ПО (п. 6.1.2 стандарта), предполагаются следующие виды работ:

6.1.2.3.1 Идентификация возможностей

6.1.2.3.2 Предоставление заявки поставщикам

6.1.2.3.3 Согласование проекта

6.1.2.3.4 Выполнение контракта

6.1.2.3.5 Поставка и поддержка продукта (услуги)
6.1.2.3.6 Закрытие
Несомненно, что у организации – заказчика существуют свои корпоративные регламенты проведения закупок (например, согласования заявок, формирования годовой программы закупок по подразделению ИТ, определения критериев выбора поставщиков услуг и т.д.). Соответственно, компания будет осуществлять приведенные в ГОСТ активности, но на операционном уровне она будет это делать в соответствии со своими внутренними регламентами и процессами.

Приложения стандарта содержат (помимо эталонной модели процессов, их описаний и видов, а также истории разработки) отдельно выделенный процесс адаптации. Приведенные в нем рекомендации по переходу от стандарта к реалиям определенного предприятия в основном концентрируются на выборе из всего приведенного множества процессов тех работ, которые необходимы для реализации конкретного программного проекта. Однако практические рекомендации по организации внедрения ГОСТ 12207-2010 остаются за границами самого документа.

2.2.3. ISO/IEC 15288 (ГОСТ Р ИСО/МЭК 15288-2005)

Международный стандарт: **ISO/IEC 15288:2005 Systems engineering. System life cycle processes (Системотехника. Процессы жизненного цикла системы).**

Российский аналог: **ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.**

Достаточно «молодой» стандарт системной инженерии (впервые представленный в 2002 году), ISO/IEC 15288 фокусируется на вопросах жизненного цикла системного уровня, в особенности тейлоринге (tailoring) – по сути, настройке и адаптации ЖЦ к конкретным требованиям и ограничениям.

В отличие от рассмотренного ранее стандарта, ISO 15288 распространяется на системы в целом, охватывая такие их элементы, как: «технические средства, программные средства, люди, процессы (например, процесс оценки), процедуры (например, инструкции оператора), основные средства и природные ресурсы (например, вода, объекты живой природы, минералы)» [50, с. 4]. Согласно данному стандарту, любой процесс ЖЦ может быть начат в любой момент, без ограничения порядка использования и последовательности (в том числе, параллельном выполнении нескольких процессов).

Важно также отметить высокий уровень абстракции ISO 15288 в сравнении с ISO 12207, так как данный стандарт не приводит ролей, конечных результатов в виде списка выходных документов, либо же состава работ, лишь оставаясь на уровне концепции.

ПРИМЕР

В процессе управления ресурсами (п. 5.3.5 Стандарта) в качестве основных пунктов деятельности приводятся следующие:

a) определять и обеспечивать поддержку инфраструктуры ресурсов, необходимой для выполнения организацией требований настоящего стандарта и осуществления поддержки проекта;

b) получать ресурсы, за исключением персонала, необходимые для внедрения и осуществления проектов;

c) проявлять заботу о персонале, занятом в осуществлении текущих проектов;

d) стимулировать персонал, например, посредством предоставления возможности карьерного роста или при помощи системы поощрений;

e) контролировать области взаимодействия нескольких проектов для разрешения связанных с графиками их реализации конфликтов.

Очевидно, что формулировки «… обеспечивать поддержку инфраструктуры ресурсов…» или «проявлять заботу о персонале…» не являются очень конкретными и могут толковаться по-разному. В связи с этим, важно иметь поддержку высшего руководства для использования данного стандарта в проекте создания и эксплуатации системы. При этом особенность ISO 15288 в том, что использоваться он может как со стороны заказчика, так и со стороны исполнителя.

Стандарт содержит четыре основные группы процессов (предприятия, соглашения, проекта и технические), описывающие соответственно вспомогательные корпоративные процессы, взаимодействие с контрагентами, управление проектом и саму реализацию системы.

Важной положительной чертой стандарта является его связь с бизнес-стороной проекта создания системы за счет групп процессов предприятия и соглашения. Благодаря наличию подобных разделов стандарта появляется связь с соответствующими корпоративными функциями и для бизнеса становится более понятным место процессов ЖЦ в процессах организации в целом.

2.3. Системный подход

Вне зависимости от того, рассматривается ли проект управления внедрения ИС или речь идет об управлении всей организацией в целом, важно иметь общее системное представление о ее деятельности, раскрывая так называемый «черный ящик».

«**Системный подход** является отдельным направлением методологии научного познания и социальной практики, в основе которого лежит рассмотрение объектов как систем. Это ориентирует исследование на раскрытие целостности объекта, выявление многообразных типов связей в нем и сведение их в единую теоретическую картину».

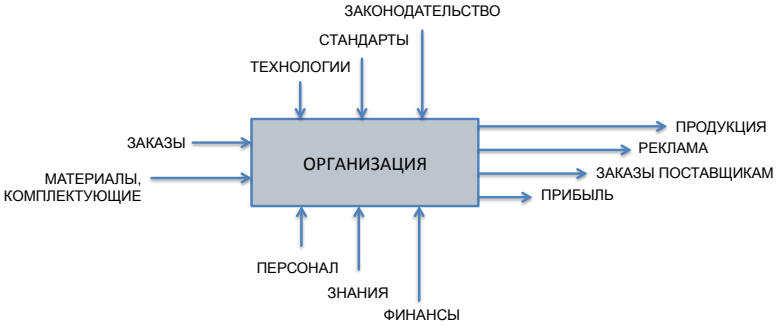
Важным свойством, характеризующим системный анализ, является эмерджентность, возрастание вероятности отличия свойств целого от свойств его компонентов по мере усложнения системы. И действительно, чем крупнее организация (которую мы рассматриваем как систему), тем больше зависимости между ее элементами влияют на общее направление ее развития.

ГОСТ Р ИСО/МЭК 15288-2005, подробно рассматриваемый в разделе стандартов и посвященный процессам ЖЦ систем, не только представляет из себя хороший пример единого системного подхода, рассматривая процессы соглашения, предприятия, проекта и технические процессы с точки зрения предприятия, а не разработчиков, но и сам в приложении содержит крайне наглядные представления понятий «системы» и «среды использования». Для успешного применения подобного системного подхода необходимо уметь рассматривать системы не только «изнутри» или «снаружи», но и «сбоку».

ПРИМЕР

Как мы уже говорили, информационная система может представляться с точки зрения структуры (набор программных модулей, аппаратного и организационного обеспечения) или, к примеру, с точки зрения функциональности (те функции и возможности, которые она предоставляет).

В качестве одного из результатов системного подхода к рассмотрению организации может являться моделирование ее связей с другими элементами внешней среды.

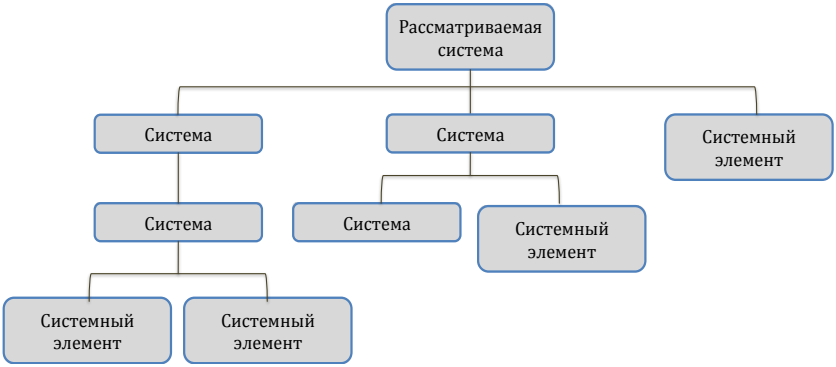


Подобная схема достаточно наглядно иллюстрирует взаимодействие компании с бизнес-системой и позволяет удостовериться, что ничего не ускользнуло из зоны внимания, и все ресурсы, ограничения, инструменты и результаты деятельности учитываются при дальнейшем структурном рассмотрении активностей предприятия. Такой анализ упрощается при моделировании и построении графических описаний, которые могут применяться как внутри организации, так и аналитиками консалтинговых / подрядных организаций для ознакомления с общей картиной бизнес-процессов.

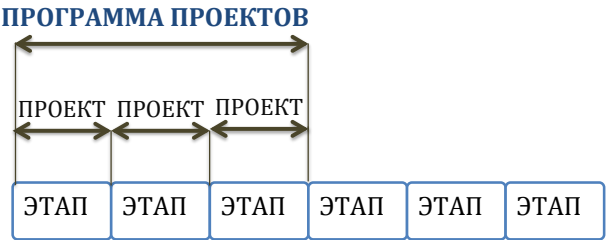
«**Структурный подход** представляет собой прохождение нескольких этапов детализации системы «сверху вниз», описывая таким образом все структурные единицы (подсистемы) компании».

Умение перемещаться по уровням абстракции и видеть иерархию, структуру для выделения общих свойств и зависимостей крайне важно для того, чтобы воспринимать систему как упорядоченный набор функций, взаимодействующий с окружающей средой. Имея высокоуровневое представление о структуре и связях, а также целях системы, можно переходить уже к дальнейшей детализации – а значит, структурному подходу.

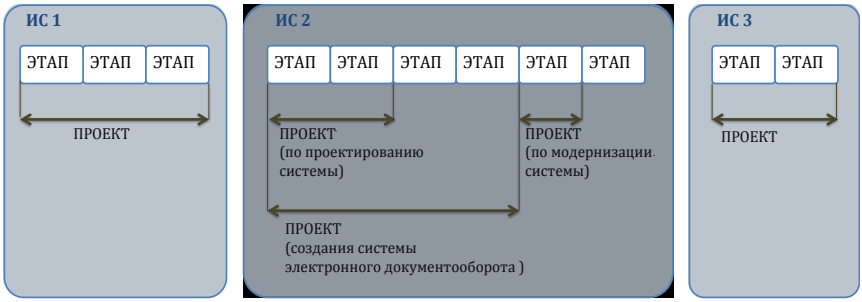
Структура системы формируется в иерархическом виде.



Так, объект любого уровня может рассматриваться в качестве системы. Именно поэтому представляется возможным в рамках создания системы рассматривать реализацию отдельных элементов (например, модулей) в рамках этапов жизненного цикла с точки зрения отдельных проектов, а создание единого решения (информационной системы) от концепции до ввода в эксплуатацию – как целую программу проектов. Планирование подобной программы, как правило, осуществляется «сверху вниз», в силу необходимости распределения временных, денежных и прочих ресурсов. Критическим фактором успеха является логическая связь этапов / проектов (определяемая выбранной моделью жизненного цикла). На схеме ниже приведен пример для каскадной модели.



Однако далее в тексте пособия термин «**проект**» используется и в другом значении – как ограниченная рамками договора с исполнителем деятельность, так как на протяжении ЖЦ ИС компания взаимодействует со множеством контрагентов.



Таким образом, совместные с подрядчиками / вендорами активности тоже могут рассматриваться как полноценные проекты, только уже не ограниченные рамками одного этапа ЖЦ, а охватывая либо несколько фаз, либо с реализацией нескольких проектов в течение одной фазы. Тогда, как правило, заключаются отдельные договора на разные проведение информационного обследования и формирование требований в рамках этапа анализа и постановки задачи.

2.4. Фазы ЖЦИС и специфика каждой из них

2.4.1. Планирование проекта

Обследование проводится в обязательном порядке на самом первом этапе для сбора информации о целях и задачах компании в целом и внедряемой системы в частности, а также получению данных об основных бизнес-процессах и покрывающем их ландшафте информационных систем (для определения в нем места внедряемого решения).

Важно верно определить:

- Организационный объем проекта (затрагиваемые реализацией проекта подразделения).
- Наличие зависимостей от других проектов.

ПРИМЕР

Если в этот же период внедряются другие системы и в обоих проектах задействованы одни и те же специалисты ИТ-департамента, одно оборудование, если они оба требуют незапланированных в исходном бюджете дополнительных расходов.

- Планируемые финансовые, инфраструктурные и человеческие ресурсы, бюджет проекта (включая необходимые закупки – например, лицензии от поставщика ПО – и планируемые затраты при необходимости оплаты услуг подрядчика по сопровождению, по обслуживанию серверов и т.д.).

Принято различать две категории обследования:

- Экспресс-обследование, имеющее своим результатом:
  - Краткое описание текущей бизнес-модели предприятия;
  - Коммерческое предложение со сформулированными проблемами, а также ориентировочными сроками и бюджетом работ по дальнейшему информационному обследованию и непосредственно внедрению системы.
- Информационное обследование, по результатам проведения которого должны быть подготовлены:
  - Полное описание текущей бизнес-модели предприятия заказчика;
  - Утвержденный план работ по разработке и внедрению (включая состав группы внедрения с обеих сторон).

Также по результатам информационного обследования со стороны заказчика должны быть приняты все организационные меры для обеспечения реализации проекта. В частности, это относится к внутренним приказам и документам предприятия заказчика (для назначения ответственных за предоставление информации и взаимодействие с проектной командой в рамках внедрения).

2.4.1.1. Экспресс-обследование

Основными целями экспресс-обследования являются:

- Выявление основных проблем и «узких мест» по участкам бизнес-процессов.
- Изучение и укрупненное описание бизнес-процессов в привязке к подразделениям предприятия с распределением ответственности между ними.
- Изучение и описание базы нормативно-справочной документации, используемых на предприятии.
- Изучение пожеланий по совершенствованию системы управления предприятием – и в свою очередь, формирование предложений по решению проблем, стоящих перед предприятием.
- Подготовка предложений по срокам и стоимости проведения полномасштабного информационного обследования требуемых бизнес-процессов предприятия.

Таким образом, экспресс-обследование – это своеобразное знакомство потенциального заказчика и исполнителя с работой друг друга, так как первый получает взгляд на свой бизнес со стороны и точную оценку объема и ресурсных требований к возможному проекту, а исполнитель – важную информацию для дальнейшего использования при подготовке коммерческого предложения и реализации проекта.

В качестве предусловий со стороны заказчика для старта этапа требуется подготовить и обеспечить издание приказа по предприятию о проведении экспресс-обследования существующей системы управления, а также создать / утвердить приказом по предприятию рабочую группу со стороны заказчика. Подробнее организационный аспект реализации проектов будет рассмотрен в главе 3, однако подчеркнем и здесь, что первоначально в нее могут входить следующие лица:

- Куратор проекта (представитель высшего руководства, иногда называемый спонсором проекта, т.е. отвечающим за финансирование проекта).
- Руководитель проекта (менеджер, ответственный за принятие решений в рамках согласованных в договоре бюджета и сроков).
- Руководитель финансового подразделения.
- Главный бухгалтер.
- Руководитель планового подразделения.
- Руководитель производственного подразделения.
- Руководитель сбытового (маркетингового) подразделения.
- Руководитель снабженческого подразделения.

На этом этапе, как правило, впервые происходит встреча бизнес-заказчика и спонсора проекта с проектной командой для обсуждения постановки

задачи и основных требований и ограничений (чаще высокоуровневых). Для сбора дополнительной / более детальной и оперативной информации применяются заранее разработанные специалистами команды исполнителя анкеты, заполняемые рабочей группой / ключевыми специалистами по функциональному признаку в течение одной недели.

Анкета экспресс-обследования предназначена для предварительного знакомства с предприятием, определения проблематики и потребностей развития компании. Она выдается руководителю проекта; возможно заполнение отдельных разделов анкеты руководителями подразделений, но в любом случае заполненная анкета должна быть заверена куратором проекта. По результатам анализа анкеты возможно определить потенциальный круг задач и варианты совместной работы заказчика и исполнителя для их решения. Среди информации, предоставляемой в анкете такого рода:

- Информация о предприятии.
- Система управления предприятием:
  - Организационная структура.
  - Процессы управления.
  - Информационное обеспечение процессов управления.
  - Документооборот.
  - Бизнес-процессы основной деятельности.
  - Автоматизация.
  - …
- Актуальные вопросы развития.
- Дополнительные пожелания заказчика, примечания.
- Данные о сотруднике, ответственном за заполнение анкеты.

В результате основного анкетирования какие-то моменты могут быть до конца не выяснены, где-то могут возникнуть противоречия. Поэтому возможно осуществление обратной связи по анкетам с целью уточнения ранее полученных данных. При этом крайне важное значение имеет то, что консультантам должны предоставляться исключительно точные данные в пределах предварительно оговоренных руководством предприятия информационных границ.

Продолжительность обратной связи по анкетам – 3-7 дней. Основным результатом проведения экспресс-обследования является отчет, формируемый консультантами в течение 1-2 рабочих недель.

В итоге отчет о результатах проведенного экспресс-обследования будет описывать следующие направления:

1. Краткая характеристика объекта исследования. Основные виды деятельности.
2. Схема и краткое описание организационной структуры предприятия.

3. Основные функции обследуемых структурных подразделений с группировкой по бизнес-процессам.
4. Краткое описание существующих на предприятии бизнес-процессов, принципов взаимодействия между подразделениями предприятия заказчика.
5. Описание информационного взаимодействия подразделениями.
6. Описание основных проблем и слабых мест на обследуемых участках бизнес-процессов.
7. Пожелания заказчика по совершенствованию системы управления предприятием.
8. Предложения о дальнейшем сотрудничестве, рекомендации по решению выявленных проблем и ориентировочная стоимость работ.

В результате экспресс-обследования консультанты получают знания о предприятии заказчика, включая представление о его бизнес-модели. В свою очередь, заказчик вместе с отчетом получает краткое описание бизнес-модели с ценной информацией о выявленных возможностях для улучшения – и конечно, коммерческое предложение с указанием ориентировочных сроков и стоимости работ по полному информационному обследованию и проведению работ по созданию и внедрению информационной системы на предприятии.

2.4.1.2. Технико-экономическое обоснование

ТЭО, технико-экономическое обоснование (в английской терминологии business case) – экономическое, бизнес-ориентированное обосновании выгод от ИТ-инвестиций (например, в информационную систему) и прогнозировании объема этих выгод.

Ключевая проблема, с которой сталкиваются как «обосновывающие» систему, так и принимающие решение – невозможность объективно предсказать и рассчитать совокупный эффект от самой системы и от ее взаимодействия с другими программными решениями, входящими в архитектуру предприятия. Именно поэтому ТЭО чаще всего превращается в декларативный документ, содержащий цели и задачи внедрения системы, ожидаемые результаты и возможные риски, по сути представляя собой краткую «бизнес-версию» технического задания.

Содержание ТЭО в разных компаниях различается, однако в классическом случае оно может состоять из следующих разделов:

1. Резюме для руководства.

Основные показатели эффективности проекта.

ПРИМЕР

– *Чистый денежный поток (млн. руб.).*
– *Суммарный денежный поток (млн. руб.).*

– *NPV, чистая приведенная стоимость (млн. руб.).*
– *IRR, внутренняя норма доходности, %.*
– *Срок окупаемости, лет.*

2. Описание проекта.

2.1 Предпосылки реализации проекта.

Документы, нормативно-правовые акты, события.

2.2. Мировой и отраслевой опыт реализации подобных проектов.

Аналогичные проекты, выполненные в схожих предметных областях / индустриях, их достижения и ошибки.

2.3. Ключевые заинтересованные стороны.

Заказчики проекта: функциональные подразделения различных уровней иерархии и конкретные сотрудники (в т.ч. акционеры и топ-менеджмент), в чьих интересах / с учетом чьих потребностей планируется реализация проекта

2.4. Организационный масштаб проекта.

Охватываемые проектом подразделения / сотрудники.

2.5. Проектные зависимости.

Другие проекты, результаты которых используются при реализации данного проекта. И напротив, проекты, реализация которых возможна только после окончания текущего.

2.6. Основные области достигаемых бизнес-выгод.

ПРИМЕР

– *Сокращение рисков несоблюдения установленных сроков и объемов продукции.*
– *Исключение рисков несоответствия местных нормативов требованиям корпоративных и отраслевых регламентирующих актов.*
– *Повышение качества производимой продукции.*
– *Сокращение расходов на создание, развитие и сопровождение ИТ инфраструктуры.*
– *Обеспечение качества и достоверности информации.*

3. Оценка коммерческой целесообразности реализации проекта.

3.1. Затраты и экономика проекта.

Покупка аппаратного и программного обеспечения, оплата услуг подрядчиков, операционные расходы, обслуживание оборудования и прочие статьи затрат.

3.2. Основные области выгод. Экономические выгоды.

Оценка чистого денежного потока, внутренней нормы доходности, срока окупаемости.

3.3. Структура финансирования проекта.

Определение источников финансирования (внешние инвесторы, бюджет подразделения, …).

4. Матрица основных проектных рисков.

Ключевые ассоциированные с проектом риски, сгруппированные по категориям (например, вероятность реализации риска и размер возможных убытков).

2.4.1.3. Оценка целесообразности проекта (TELOS)

Перед тем, как приступать к разработке, важно еще раз убедиться, что проведенные работы по анализу, обоснованию и проектированию системы достаточны для перехода к следующему этапу.

Напоминая по своей сути SWOT и PEST-анализы, акроним TELOS содержит основные аспекты, при рассмотрении которых можно удостовериться, что проект по реализации системы реалистичен и имеет значительный потенциал.

– **Технические** (*Technological*).

Проверка наличия инфраструктурных ресурсов организации, а также квалифицированного персонала и проектного опыта для создания и поддержки системы. Описание потенциальных сложностей и возможностей их разрешения.

– **Экономические** (*Economical*).

Определение ожидаемых экономических выгод от реализации информационной системы.

ПРИМЕР

Снижение времени обработки транзакции на 18 %.

– **Юридические** (*Legal*).

Определение, насколько все технологические спецификации и существующая функциональность обеспечивают требования к безопасности данных и их предоставлению в случае необходимости.

ПРИМЕР

Любые системы обработки персональных данных должны обеспечивать минимальный (закрепленный регламентами и законодательством) уровень защиты информации.

– **Операционные** (*Operatoinal*).

Покрытие сформированных на этапе анализа требований заинтересованных сторон подготовленным проектом. Определение, насколько система удовлетворяет возможностям и потребностям бизнеса с точки зрения сроков проекта, встраивания в бизнес-процессы в текущих рыночных условиях в моменту создания системы. Рассмотрение критериев надежности, возможности поддержки, удобства интерфейса, продуктивности, стабильности и прочих факторов создания системы.

– **Сроки реализации** (*Scheduling*).

Оценка сроков создания системы для определения возможных изменений во внешней среде за время реализации проекта. Оценка, насколько реалистичны поставленные крайние сроки сдачи системы в эксплуатацию, а также промежуточные вехи.

2.4.1.4. Выбор программного решения

Собрав требования и подходя к этапу проектирования, проектной команде предстоит принимать решение о том, каким образом подойти к выбору программного решения: необходимо ли его разрабатывать с нуля, либо приобрести уже существующий на рынке и зарекомендовавший себя продукт.

Заказные КИС являются уникальными, создаются исключительно для нужд и достижения бизнес-целей конкретной компании (чаще всего, конкретной узкой специализации или для автоматизации одной функции). Как следует из названия, такие системы не имеют аналогов и не подлежат дальнейшему тиражированию (а часто – и масштабированию). Обычно ИС такого класса не имеют прототипов. Иногда использование прототипов возможно, однако потребуется внесение серьезных изменений качественного характера.

Отдельно необходимо отметить, что к системам данного типа будем относить как самостоятельную разработку внутренними ИТ-подразделениями компании, так и действительно заказную разработку, осуществляемую силами подрядчиков / внешних компаний.

Среди основных причин, по которым компании рассматривают разработку бизнес-приложений с нуля как подходящий для себя вариант:

1. «Не существует двух одинаковых компаний, а значит, стандартные решения не смогут целиком отразить специфику компании».
2. «Стандартные системы содержат избыточный набор функций, за которые приходится платить».
3. «Стоимость самостоятельной разработки ниже стоимости лицензий и услуг по системной интеграции / консалтингу.

Разработках таких КИС сопряжена с дополнительными рисками в получении необходимых результатов.

В случае самостоятельной разработки важно помнить о том, что компетенции для работы с подобными системами (и / или их создания) часто являются уникальными, а значит риск ухода ключевых сотрудников, работавших с ними тоже нужно принимать во внимание. Также у разрабатываемых «для себя» приложений часто обнаруживаются ошибки, вызванные недостаточно тщательно проведенным тестированием. В силу того, что в большинстве компаний в штате ИТ-специалистов нет отдельных ответственных за версионность, бета-тестирование и другие аспекты процесса внедрения, основная часть разработчиков фокусируется исключительно на реализации функциональных возможностей и производительности системы, не уделяя должного внимания «чистоте» и безопасности кода.

В случае заказной разработки часто встречаются случаи неверного толкования технического задания (или ошибок при его составлении), высокая оценка стоимости реализации проекта, коммуникационные проблемы, неполное документирование системы.

По всем вышеописанным причинам, а в первую очередь, в силу развитости современной индустрии решений для автоматизации деятельности компании на сегодняшний момент заказная разработка уже далеко не столь распространена, как 15 лет назад. Ранее популярность подобных решений объяснялось тем, что адаптируемых или «коробочных» КИС для отраслей / бизнесов определенного масштаба почти не существовало. Сейчас почти любой проект разработки / внедрения предполагает лишь доработку стандартной или отраслевой конфигурации под согласованные с заказчиков требования. Указанный процесс и будет рассматриваться далее.

Противоположной категорией являются **тиражируемые КИС**. Иногда их еще называют «коробочными» или «шаблонными» решениями, что однако не совсем корректно в силу адаптации подобных систем к нуждам конкретного предприятия. Как правило, в основу подобных систем заложены общие процессы и свойства предприятий (примерно одного размера и работающих в одной индустрии). Производитель ИС такого типа при разработке использует опыт собственных проектов и «лучшие практики» (best practices) отрасли. Соответственно, одним из критериев при выборе ИС становится репутация и опыт производителя системы, а также общий масштаб системы и ее соответствие специфике предприятия, несмотря на то что способность к адаптации так или иначе предусмотрена в большинстве информационных систем.

Процесс адаптации тиражируемых КИС может быть как одноуровневым, так и двухуровневым. В первом случае настройку и изменение параметров исходной системы, купленной у предприятия-производителя, проводит непосредственно проектная команда по внедрению – либо со стороны исполнителя (консалтинговых компаний и системных интеграторов), либо собственными силами ИТ-подразделения компании.

ПРИМЕР

Закупка пакета лицензий SAP Business One с услугами по интеграции непосредственно у российского офиса SAP AG.

Во втором случае (особенно в случае зарубежных КИС) в исходное решение российскими компаниями–системными интеграторами на основе собственного опыта в национальной отраслевой специфике с учетом особенностей российского законодательства вносятся модификации и ими же производится дистрибуция созданного отраслевого решения (вместе с услугами по интеграции).

ПРИМЕР

Партнерство NaviCon Group и Microsoft, в рамках которого на базе решения Mictosoft Dynamics NAV разработан локализированный продукт NaviCon Trade, автоматизирующий управление сбытом, заказами и взаиморасчетами с клиентами в торговых предприятиях.

Резюмируя сказанное, самостоятельная (заказная) разработка предполагает равномерное распределение внимания между всеми фазами жизненного цикла (при написании кода с нуля), в то время как доработки и настройка тиражируемых информационных систем концентрируются на сборе требований, сравнении их с имеющимся решением и последующем развертывании ИС. Далее в пособие жизненный цикл информационных систем будет рассматриваться в контексте именно этого подхода в силу его распространенности и оптимальности в современных условиях жесткой конкуренции и стремления к сокращению времени и бюджета внедрения.

2.4.2. Анализ и постановка задачи

Существует определенный набор ключевых мероприятий, обязательное проведение которых может повысить вероятность успеха проекта. Уже на стадии постановки задачи должны быть определены цели этапов проекта и самого проекта, включая критерии их выполнения. При этом данные цели должны быть сформулированы предельно четко и ясно (по возможности, удовлетворяя критериям **SMART**: конкретность, измеримость, достижимость, актуальность, установка конкретного срока).

ПРИМЕР

«Установка бухгалтерской программы» (просто ради ее установки) или «разворачивание технических средств» не может быть самодостаточной целью внедрения системы, а будет являться лишь инструментом для достижения настоящей цели. Таковой, к примеру, может являться автоматизация бухгалтерского учета (для учета операций производственно-экономической деятельности).

Прежде, чем перейти к рассмотрению основных активностей, подчеркнем, что следующая часть документа основывается на опыте внедрения различных программных систем, и в частности – интегрированной системы управления, о которой уже было сказано ранее и рекомендациях стандартов и сводов знаний.

Под **интегрированной системой управления** понимается система управления, использующая совокупность современных программно-аппаратных средств, информационных технологий и экономико-математических методов для регулярного решения задач управления производственно-экономической деятельностью предприятий и коммерческих организаций.

Интегрированная система управления является сложной системой управления, включающей в свой состав отдельные модули (подсистемы).

Модуль – часть системы, выделенная по определенному признаку, отвечающему конкретным целям и задачам управления. В рамках этих задач он может рассматриваться как самостоятельная система.

Перейдем к основным этапам, составу, содержанию и порядку выполнения работ по внедрению программных продуктов в рамках реализуемых организациями ИТ-проектов.

2.4.2.1. Информационное обследование предприятия

Анкетирование

Основными целями информационного обследования являются детальное описание требуемых бизнес-процессов, подготовка предприятия заказчика к внедрению системы и уточнение графика выполнения работ по внедрению системы по срокам и видам работ. Информационное обследование осуществляется по определенной методике, предусматривающей три ключевых этапа:

- Сбор первичной информации (анкетирование, интервью и др.).
- Систематизация и анализ информации.
- Представление собранных данных (в формате отчета) для согласования результатов обследования с заказчиком проекта.

Важно отметить, что именно эти активности тесно пересекаются с получением информации на всех стадиях ЖЦИС, которое будет тем быстрее и эффективнее, чем более тщательная работа будет проведена на первом этапе.

Основные активности этапа:

- Определение даты начала работ, назначение руководителя работ и ответственных представителей заказчика по каждому подразделению предприятия, в которых проводятся работы по обследованию бизнес-процессов, издание приказа о начале работ.
- Определение основных характеристик предприятия командой исполнителя.
- Утверждение приказа об информационном обследовании бизнес-процессов предприятия с назначением руководителя проекта от заказчика.
- Формирование графика проведения собеседований по анкетам с указанием даты, места и участников встречи.
- Проведение интервьюирования для описания бизнес-процессов (включая сквозные) с указанием ответственных.
- Анкетирование основных подразделений.
- Подготовка отчета для заказчика о результатах обследования с его последующим согласованием и утверждением.

Среди предоставляемых заказчиком анкет информационного обследования:

- Опросники для высшего руководства.
- Опросники руководителей подразделений.

– Опросники специалистов: ведущих специалистов, сотрудников отделов, служб, главного бухгалтера, сотрудников бухгалтерии (в случае внедрения финансовых или интегрированных с ними систем).

Анкета для высшего руководства предназначена для определения проблемных зон в функциональном взаимодействии подразделений и информационном обмене, затрудняющих процесс принятия управленческих решений для руководства предприятия. Данная анкета может быть заполнена генеральным директором, финансовым директором, коммерческим директором, техническим директором и другими руководителями аналогичного ранга. Внимательное заполнение анкеты, а также предоставление полной и точной информации поможет консультантам лучше структурировать те проблемы на предприятии, которые могут быть решены с их помощью.

Анкеты руководителей бизнес-подразделений создаются в целях определения проблемных зон в информационном обмене и функциональном взаимодействии подразделений и с точки зрения определения текущей ситуации и выявления потенциальных областей для улучшения он является наиболее показательным.

ПРИМЕР

Руководители департаментов / функциональных направлений могут обозначить основные типы проблемных ситуаций, связанных с ***механизмом коммуникаций между подразделениями****: дублирование функций; перегруженность работой; не отрегулированный механизм обмена информацией, прохождения документов, заказов; недостаток информации для принятия оперативных решений по ряду вопросов; сложности при запросе дополнительных данных; отсутствие оперативной информации; задержка в предоставлении отчетов, различных данных; предоставление некачественной информации (неточности, произвольная форма) и т.п.*

Могут быть также определены некоторые ***проблемные участки работы****, например, составление отчета о работе подразделения за месяц. Если данная функция не автоматизирована, информация собираются вручную и часть информации поступает с задержками и неточностями. Это повышает напряженность во взаимоотношениях между сотрудниками, отвечающими за разные участки работы подразделения и требует много времени на обработку поступившей информации.*

Помимо анкет и интервью с ответственными от подразделений, используются в качестве источников информации также следующие передаваемые заказчиком документы:

– *Информационный буклет (рекламный проспект) об истории и основных направлениях деятельности предприятия.*
– *Схема организационной структуры.*
– *Положения о подразделениях, отделах.*
– *Учетная политика предприятия.*
– *Стандарты качества.*
– *Стратегия развития.*
– *Документация по используемому ПО.*

И наконец, анкеты для специалистов призваны выявлять зоны ответственности сотрудников, основные «узкие места» процессов, с которыми им приходится сталкиваться при работе. Определяется, каким образом специалисты организуют свою ежедневную деятельность, с какими системами / данными работают. Также вопросы анкеты помогают сформировать точку зрения сотрудников на потенциальные возможности улучшения процессов как в области их ответственности, так и в масштабах всей организации.

Прочие методы сбора информации

Как и в случае экспресс-обследования, помимо интервью с ответственными от подразделений и анкет на этом этапе используется значительный объем документальных источников информации:

- Все запрашиваемые первичные документы предприятия.
- Стандарты предприятия.
- Положения о подразделениях, должностные инструкции, штатное расписание, стандарты предприятия.
- Нормативно-справочная информация.
- Бизнес-план и / или стратегия развития предприятия.
- Имеющиеся на предприятия аналитические отчеты, результаты маркетинговых исследований и т.п.
- Финансовые отчеты, балансы и т.п.

В результате интервью должны быть:

- выявлены все внешние объекты, с которыми предприятие взаимодействует, технологии взаимодействия со стороны предприятия, а также информационные и материальные потоки, обеспечивающие эти взаимодействия;
- выявлены реальные технологии работы предприятия;
- определены реальные функции подразделений и их взаимосвязи и взаимозависимости, информационные потоки;
- идентифицированы и специфицированы все информационные хранилища, в т.ч. и бумажные (картотеки, архивы);
- оценены используемые на момент проведения обследования средства автоматизации и программное обеспечение как в структурных подразделениях, так и на предприятии в целом.

По результатам проведенных работ исполнителем должна быть подготовлена и передана заказчику бизнес-модель предприятия (с диаграммами сквозных бизнес-процессов), а также сформирован план мероприятий по внедрению системы. В типовом виде он содержит:

- Цели и задачи внедрения.
- Организационные мероприятия.
- Рекомендации по реорганизации необходимой организационной структуры для поддержки системы.

- Распределение функционально-рабочих мест системы у предприятия-заказчика.
- Требования к аппаратному и системному программному обеспечению локальной сети предприятия-заказчика.
- Пуско-наладочные работы (состав, сроки).
- Этапы ввода системы в эксплуатацию.
- Ориентировочный план-график проведения работ по внедрению системы на предприятии заказчика (в относительных датах).
- Ориентировочная стоимость поставки и внедрения системы.

И конечно, переход к следующему этапу невозможен, если не определены цели внедрения системы. Вне зависимости от того, приобретается ли «коробочное» решение или начинается разработка с нуля, очень важно на данном этапе уточнить с заказчиком и спонсором проекта список основных стейкхолдеров и достичь в переговорах с ними компромисса, так как у всех заинтересованных лиц могут быть очень разные представления о назначении и целевом использовании системы, как и о ее масштабах. В дальнейшем полученная информация уточняется в ходе фазы проектирования и становится основой для формирования технического задания, которое должно быть максимально проработанным со всеми участниками проекта, ведь именно на первых двух стадиях совершается большинство ошибок, которых можно избежать при применении более формальных методов анализа.

2.4.2.2. Описание бизнес-процессов

На этом этапе формируются описания законченных бизнес-процессов, участие в котором могут принимать одно или несколько подразделений.

ПРИМЕР
Определяются основные активности процесса.
Производство:
- *планирование выпуска и производственных мощностей;*
- *планирование обеспечения ресурсами (ТМЦ, услуги и работы, трудовые ресурсы, энергия, прочее);*
- *организация производственного процесса (подготовка документов для производства);*
- *учет затрат на производство (основное, вспомогательное)*
- *прямые производственные затраты, косвенные (общепроизводственные);*
- *учет выпуска готовой продукции и незавершенного производства;*
- *контроль над выполнением производственной программы (диспетчеризация производства);*
- *анализ производства (отчетность);*
- *основные проблемы на данном функциональном участке и рекомендации по их решению (проблема – причина – выход).*

Основная цель моделирования процессов – их документирование и последующее осуществление функционального анализа на предмет поиска «узких» мест процессов и возможностей для их совершенствования. Важно, что эта деятельность ни в коем случае не является «моделированием ради моделирования». В дальнейшем полученные результаты будут использоваться при формировании функциональных требований к системам, а также при реинжиниринге бизнес-процессов.

Далее отображаются также основные действующие лица, документооборот, поддерживающие информационные системы и прочие элементы процессов, уже в графическом виде (в том числе UML-диаграммы, нотации BPMN и EPC). Подобное моделирование отдельных предметных областей и процессов позволяет сформировать системное представление о деятельности организации.

2.4.2.2.1. Основные нотации / методологии моделирования

BPMN

Нотация моделирования бизнес-процессов BPMN (Business Process Modeling Notation) была впервые представлена публике еще в 2004 году и описана консорциумом Object Management Group. В основе управления процессами в BPM идея, что сама стратегия управления опирается на три основные методологии: моделирования бизнес-процессов, анализа и оптимизации. В свою очередь, их поддерживает ряд инструментальных средств, в частности:

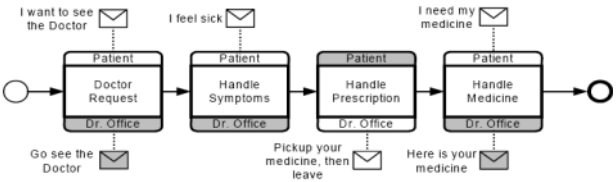
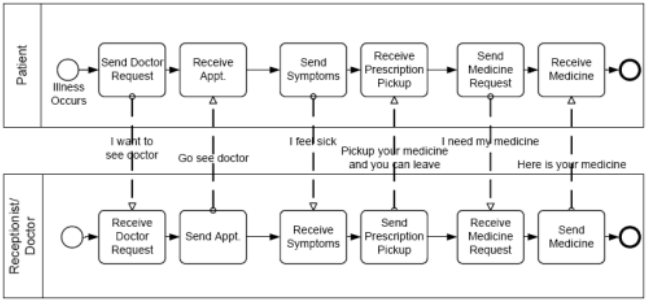
- Для разработки стратегии, описания, анализа, документирования.
- Для информационной поддержки бизнес-процессов.
- Для поддержки потока работ (Workflow management).

BPMN с самого начала создавалась как нотация, подходящая для применения любым пользователем – от бизнес-аналитиков и разработчиков до руководителей, менеджеров бизнес-процессов и просто рядовых сотрудников подразделений. Она объединяет различные точки зрения на бизнес-процесс, тем самым стандартизируя модель.

ПРИМЕР

В официальном полном описании нотации BPMN указывается, что для разработки первой версии модели были объединены концепции и некоторые объекты следующих диаграмм/нотаций:

- *Диаграммы активности UML (universal modeling language).*
- *Диаграмма потоков активностей и принятия решений ADF (activity decision flow).*
- *Диаграмма событийных цепочек процесса EPC (event-driven process chain),*
- *Нотация функционального моделирования IDEF (Icam DEFinition for functional modeling).*
- *Другие модели (UMLEDOC Business Processes, RosettaNet, LOVeM).*

Группы объектов	Описание
Действия	При помощи объектов **Действия** описываются задачи (бизнес-правила, сценарии, задачи-сервисы, задачи отправки сообщений и другие). Задачи затем детализируются, в том числе за счет маркеров действий (сценариев действий) и определения потоков (порядка и условий выполнения действий).
События	События в BPMN, как и в некоторых других нотациях, относятся к категориям начальных, промежуточных и завершающих. К событиям относятся: отправка и получение сообщений, таймеры, ошибки, сигналы, остановы и другие виды событий. По сути событие является индикатором требующего дальнейшего участия пользователя происшествия, что возможно организовать НЕ прерывая процесса (в отличие от предыдущих версий нотации).
Логические операторы	Логические операторы определяют порядок наступления событий процесса при ветвлении и слиянии / синхронизации.
Данные	Стандартные объекты данных (сообщения, хранилища, коллекции объектов), которые могут использоваться различными действиями
Хореография	Понятие, впервые появившееся именно во второй версии нотации BPMN. Его основная идея в отражении **задач взаимодействия** (обмена сообщения) между участниками (двумя и более). Хореография по сути представляет собой принципы поведения между различными пулами (а не внутри одного) без выделения единого центрального контроллера процесса. ***ПРИМЕР***[1] 
Диалоги / взаимодействия	Определение **характера информационных взаимодействий**: *один-к-одному* либо *один-ко-многим*. Отличие от хореографии – в выделении нескольких пулов действий (swimlanes) и детальном описании каждого из них ***ПРИМЕР*** 
Роли	Благодаря группе объектов **Роли** определяются: – Распределение обязанностей участников процесса. – Информационные потоки между ними. – Порядок обмена этими сообщениями.

[1] По материалам http://www.dorofey.su/_ld/0/69_BPMN_2_0.pdf.

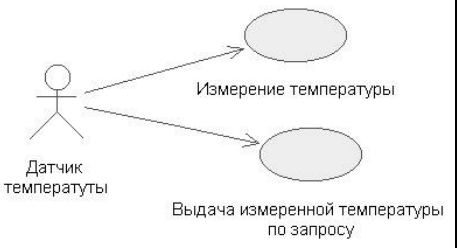
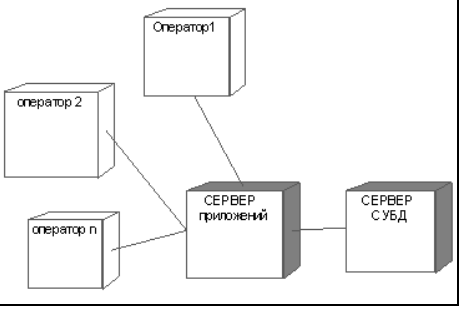
В 2010 году была опубликована версия BPMN 2.0, созданная при сотрудничестве многих исследовательских групп, в том числе консорциума Object Management Group, Института Хассо Платтнера (Hasso Plattner Institut – Потсдам, Германия), университета Гумбольдта (Берлин, Германия), университетской инициативы Signavio.

В 2013м версия BPMN 2.0.1 была принята как международный стандарт **ISO/IEC 19510:2013 Информационные технологии. Модель и нотация процесса менеджмента объекта в групповом бизнесе.**

Среди основных понятий и групп объектов в BPMN:

UML

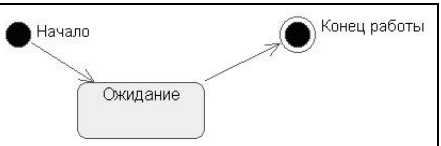
UML (Unified modeling language, универсальный язык моделирования)[1] является методологией объектно-ориентированного подхода, представляющей набор диаграмм для проектирования информационных систем.

Диаграмма вариантов использования / прецедентов (Use case diagram) Описание «вариантов использования» системы (по сути – выполняемого ей набора функций). Все варианты использования в подобных диаграммах должны относиться к тому или иному действующему лицу (актору), так как определяют для них варианты / сценарии поведения. Такие диаграммы применимы при описании определенных требований к системе, описывая как ее объекты, так и объекты предметной области. ***ПРИМЕР*** *Система бухгалтерского учета.* *Сотрудник бухгалтерии: «Сформировать счет за поставку товара для контрагента А».* *Главный бухгалтер: «Согласовать счет», «Сформировать сводный отчет по расчетам с контрагентом А за период 6 мес.»*	
Диаграмма развертывания (Deployment diagram) Диаграммы топологии предназначены для определения аппаратных средств, топологии проектируемой системы ***ПРИМЕР*** *Сервер приложений, СУБД, рабочие станции*	

[1] Источник изображений далее в таблице: http://www.interface.ru/home.asp?artId=4710.

Диаграмма состояний (statechart diagram)
Диаграмма, описывающая все возможные состояния в модели поведения системы

ПРИМЕР
Опрос Web-ресурса
Ожидание подключения к серверу



Диаграммы активности (activity diagram)
Определение основных бизнес-процессов объекта, последовательности их реализации, ветвлений при принятии решений, дальнейшей синхронизации процессов

ПРИМЕР
Получить заказ, Выполнить заказ, Сформировать счет, Отправить счет

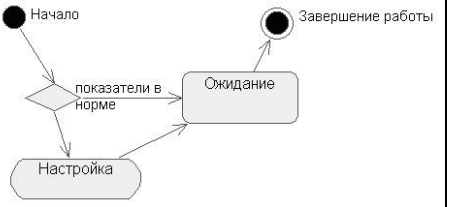


Диаграмма последовательности (sequence diagram)
Данная диаграмма представляет собой единый процесс взаимодействия между объектами, с фокусом на переносе активностей от одного объекта к другому и последовательности подобных действий.

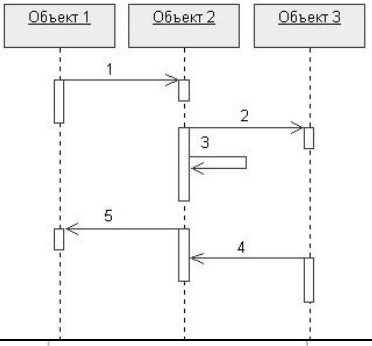


Диаграмма сотрудничества (collaboration diagram)
Данная диаграмма отражает взаимодействия между различными действующими лицами в процессе выполнения процесса.

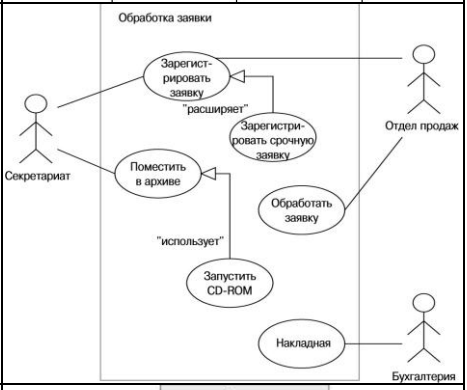


Диаграмма классов (class diagram)
Логическое представление системы через определение основных классов (например, Человек, Студент, Профессор) и отношений между ними (например, наследование)

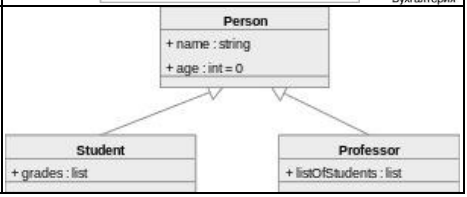
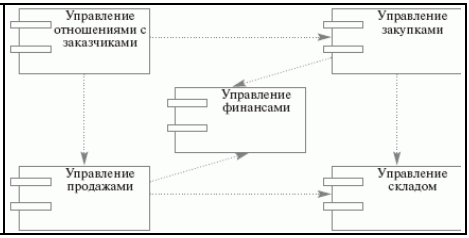


Диаграмма компонентов (component diagram) Определение основных модулей системы и связей между ними ***ПРИМЕР*** *Управление отношениями с потребителями, Управление продажами, Управление финансами*	

UML версии 2.4.1 был также принят в качестве международного стандарта моделирования. **ISO/IEC 19505:2012. Информационные технологии. Унифицированный язык моделирования группы по управлению объектами (OMG UML)**.

IDEF

IDEF (Icam (Integrated computer-aided manufacturing) DEFinition for functional modeling) представляет собой семейство стандартов описания и отображения бизнес-процессов.

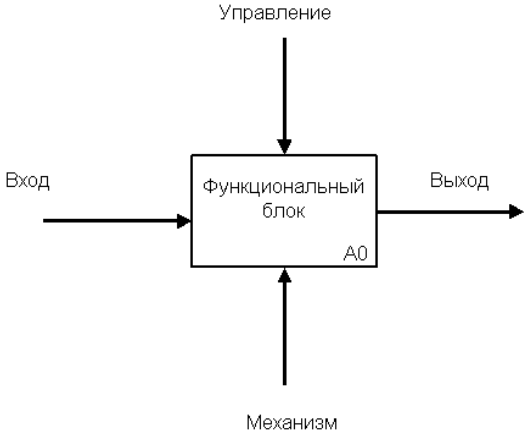
- IDEF0, отображающая процесс на уровне функций.
- IDEF1, фокусирующая на информационных потоках.
- IDEF1X для разработки реляционных баз данных.
- IDEF3, моделирующая технологические процессы как следующий уровень после IDEF0.
- Прочие (реже применяющиеся методологии IDEF):
  - IDEF2 для динамического моделирования систем.
  - IDEF4 для построения объектно-ориентированных систем.
  - IDEF5 для онтологического исследования сложных систем.
  - IDEF6, акцентирующая внимание на процессе создания модели (обстоятельствах и причинах выбора того или иного метода моделирования).
  - IDEF7 для аудита информационных систем.
  - IDEF8 для разработки пользовательских интерфейсов.
  - IDEF9 для определения бизнес-ограничений при сценарном проектировании информационных систем.
  - IDEF10 – IDEF14 – методы в области проектирования компьютерных сетей, архитектуры внедрения и прочих предметных областей, которые были определены как необходимые и востребованные, но разработка которых не была завершена.

Язык структурного анализа SADT (structured analysis and design technique).

Рассмотрим наиболее распространенные диаграммы (а именно: IDEF0 и IDEF3) подробнее.

IDEF0

IDEF0 представляет процесс в виде иерархической структуры функциональных блоков.



Функциональный блок представляет собой функцию (например, «Производить услуги», «Разработать план производства», «Выставить счет»), к которой относятся интерфейсные дуги четырех типов (Вход, Выход, Управление, Механизм). Входом и выходом функции могут являться как различные реальные объекты (детали, материалы), так и потоки информации (данные о заказах, аналитика продаж). Управление определяет все стандарты, методологии, распоряжения и прочие регламентирующие документы, а Механизм – поддерживающие выполнение функции информационные системы, сотрудники и ресурсы. IDEF0 в обязательном порядке требует указания как минимум одной управляющей и одной исходящей интерфейсной дуги, что объясняется с точки зрения необходимости для каждой функции иметь определенные правила выполнения и определенный результат, иначе смысла в функциональном блоке просто не будет.

IDEF0 используется не только для иллюстрации процессов предприятия, но и в целом для описания схемы его функционирования, с какими ресурсами компания работает, от чего зависит и какой итоговый результат производит. Однако в некоторой степени содержание функциональных блоков IDEF0 остается «черным ящиком», в связи с чем и проводится построение дополнительны схем процессов, уже в нотации IDEF3.

IDEF3

IDEF3 предназначена для документирования процессов системы, последовательности их операций, детализирующих функции, описанные в IDEF0. Данный подход более структурированный, чем IDEF0, и определяет

схемы последовательности процессов, их ветвления и слияния, что также полезно в случае использования модели для дальнейшей работы уже с методами имитационного анализа.

IDEF3 оперирует понятиями «единиц работы», которые объединяются по принципу временного предшествования либо использования результатов одной единицы работы в качестве входных потоков для другой.

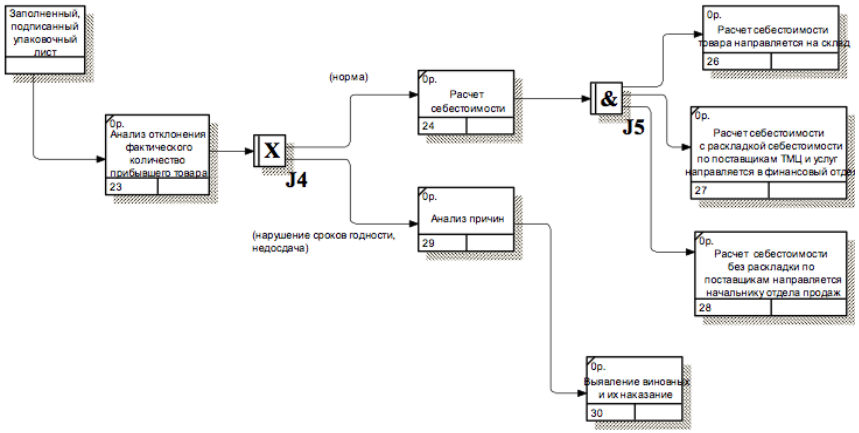
ПРИМЕР

Функция «Анализ сведений о приемке товара».

Текстовое описание.

Старший кладовщик направляет упаковочный лист в отдел закупок. В отделе закупок анализируются сведения о приемке товара. Если фактическое количество прибытия товара имеет отклонение в пределах нормы, то производится расчет себестоимости товара. Составляется 3 вида документов: расчет себестоимости товара направляется на склад; расчет себестоимости с раскладкой себестоимости по поставщикам ТМЦ и услуг направляется в финансовый отдел; расчет себестоимости без раскладки по поставщикам направляется начальнику отдела продаж. Если есть значительные ошибки (нарушение сроков годности, недостача), то проводится анализ причин, выявляются виновные, предъявляются претензии.

Графическое описание.



EEPC

Нотация **eEPC (Extended Event-driven Process Chain, расширенная событийная цепочка процессов)**, предполагает описание алгоритма действий, выполняемых отдельными организационными единицами, что позволяет сформировать общий сценарий процесса как последовательность отдельных шагов.

Основными объектами диаграммы eEPC являются следующие элементы:

Тип объекта	Описание	Графическое обозначение
Событие	Состояние внешней / внутренней среды, являющееся также обязательным условием начала и окончания выполнения функции. Конечные события могут также быть начальными событиями для другого процесса.	
Логическое правило	Правила выполнения функции (если наступает только одно из событий или обязательно оба события) и правила наступления событий при выполнении функций (по сути – правила слияния и разделения ветвей процесса)	
Функция	Описание элемента работы, который представляет собой один этап процесса	
Интерфейс процесса	Внешний (по отношению к текущей диаграмме) процесс или функцию. Используется для указания взаимосвязи процессов (как для логической последовательности: следующий / предыдущий процесс, так и для обозначения направления передачи объекта)	
Объекты организационной схемы	Обозначение организационных единиц (должностей, подразделений, ролей) – исполнителей, владельцев или участников функций. Бизнес-роли в данном случае рассматриваются как требующие полномочий для управления функциями.	

eEPC не случайна названа «расширенной» диаграммой – на практике в модели такого вида могут также включаться другие объекты, например:

- Товарно-материальные ценности.
- Бумажные и электронные документы.
- Продукты (используемые и производимые).
- Объекты информации.
- Информационные системы и их отдельные модули / функции.
- Базы данных.
- Цели (которые поддерживаются конкретной функцией).
- Места выполнения (например, производственный цех № 4).
- Другие элементы описания.

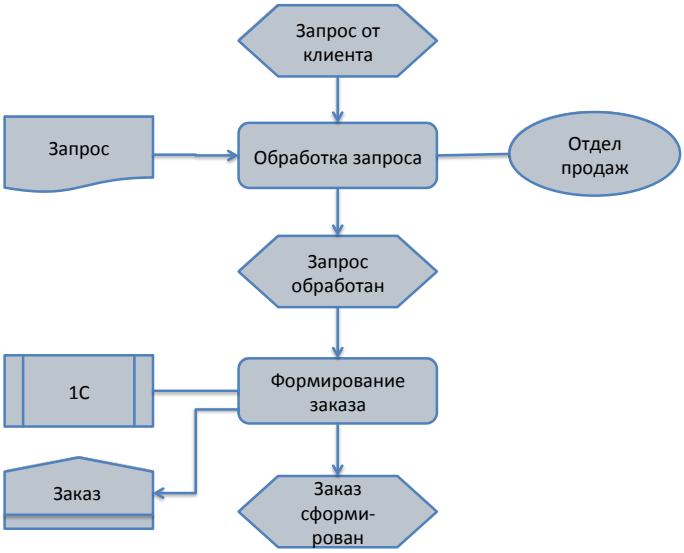
Между всеми объектами в обязательном порядке определяются связи, например, «создает» (документ), «распределяет» (задание между сотрудниками), «использует» (информационную систему 1С), «выполняет» (функцию выполняет менеджер), «принимает решение», «обеспечивает», «является владельцем» и многие другие.

ПРИМЕР

Текстовое описание.

На вход процесса поступает запрос от клиента, который должен быть обработан. Ответственность за выполнение данной функции возлагается на отдел продаж. По результатам обработки запроса будет сформирован заказ в системе 1С.

Графическое описание.



Таким образом, благодаря уровню процедур объединить на нем все объекты, задействованные в разных представлениях (документы, информационные системы, организационные единицы и должности, процессы), а значит составить интегрированную картину процесса. Она, в свою очередь, может в дальнейшем использоваться либо как основа для As-Is –анализа, либо для последующей доработки и автоматизации как самого процесса, так и управления им с точки зрения достижения целевых показателей эффективности.

2.4.2.2.2. Программные продукты моделирования деятельности организации

У использования программных продуктов, поддерживающих ту или иную методологию описания процессов, в масштабах всей компании есть несколько неоспоримых преимуществ. Во-первых, это помогает сузить спектр специализированных решений для моделирования, навыки работы с которыми необходимы сотрудникам компании, как и спектр необходимых для

детального изучения методологий. В свою очередь, это расширяет потенциальные границы использования на предприятии созданных моделей процессов. Во-вторых, программные решения позволяют использовать единый внутренний репозиторий объектов, который способствует связности моделей и проведению их верификации.

Для перевода информации о процессах организации в цифровой вид и моделировании в конкретной нотации описания процессов необходим следующий набор активностей:

- Организация рабочей группы и проведение ее заседания в целях анализа имеющихся данных по бизнес-процессам, их объектов и существующих взаимосвязях.
- Создание предварительных моделей существующих процессов (в состоянии «как есть»), их последующее обсуждение, доработка и согласование.
- Преобразование данных о процессах в вид моделей выбранной методологии описания в виде «как есть».
- Создание, доработка и утверждение моделей вида «как должно быть».
- Непрерывный анализ, мониторинг и оптимизация процессов.

Поддерживать системный и структурный уровни рассмотрения организации, а также модели бизнес-процессов позволяют CASE-технологии, реализованные в том числе в виде отдельного класса программного обеспечения (Computer-Aided Software / System Engineering). Данные системы: представленные CA ERWin Data Modeler (бывшие BPWin и Erwin), IBM Rational Rose и некоторыми другими продуктами, позволяют системным аналитикам и командам разработчиков после общения с бизнес-заказчиком сформировать проект описания деятельности компании и в дальнейшем перейти к ее автоматизации.

Завершая разговор о различных моделях описания процессов и инструментальных средствах, приведем сравнительную таблицу возможностей основных платформ моделирования.

CA ERWin Data Modeler

Комплекс CASE-средств, среда моделирования и коллективной работы с данными, проектирования и генерации баз данных. Интеграция процессов и данных (с IDEF0, DFD, IDEF3), Проверка и оптимизация дизайна баз данных, доступ к метаданным и их интеграция.

Software AG ARIS (ранее IDS Scheer ARIS)

Моделирование организации, процессов, данных, функций, продуктов и услуг. Поддержка создания ИС от анализа требований до разработки (через интерфейсы экспорта моделей в средства разработки). Дополнительные возможности работы с системой сбалансированных показателей, операционными рисками, стандартами качества

IBM Rational Rose

Проектирование приложений ИС с возможностью генерации баз данных и исходного кода. Реализация UML-диаграмм (состояний, классов, взаимодействия и других).

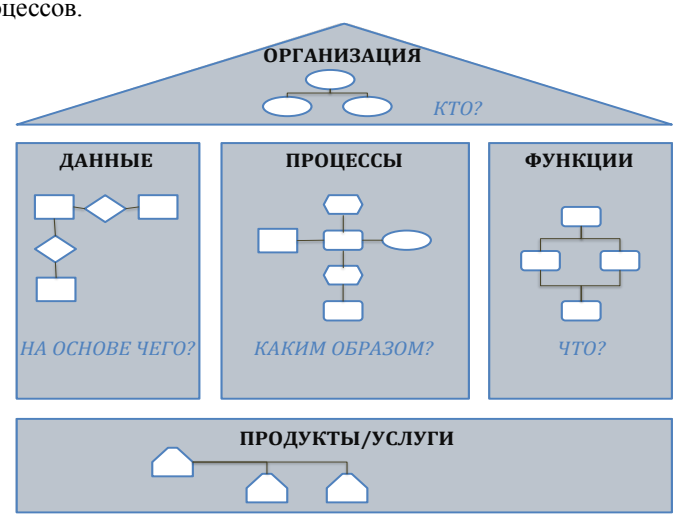
Microsoft Visio

Один из первых инструментов создания графических диаграмм. Со временем расширяя репозиторий доступных моделей и новых объектов, он основывается на отдельных наиболее распространенных видах диаграмм, хотя и не предоставляет автоматизированных инструментов их анализа.

ARIS EXPRESS & ARIS PLATFORM

Архитектура интегрированных программных систем ARIS (ARchitecture of Integrated Information Systems) – инструментальная система моделирования бизнеса, разработанная компанией IDS Scheer AG.

ARIS и как методология, и как платформа обеспечивают проектирование, внедрение и управление бизнес-процессами как в процессе повседневной работы, так и для целей анализа, оптимизации и реинжиниринга бизнес-процессов.



Модель организации в ARIS может быть определена следующим образом:

Модель организации – совокупность взаимосвязанных / взаимодополняющих графических моделей, которая адекватно описывает моделируемые предметные области деятельности организации.

Таким образом, ARIS как методология позволяет моделировать такие подсистемы предприятия, как организационная, функциональная, информационная, процессная и подсистема входов / выходов. Они формируются на трех уровнях описания, иерархически идущие от анализа проблем бизнеса к конкретной реализации при помощи ИТ:

- Требования (семантические модели).
- Спецификации (бизнес-описания с ориентацией на информационные технологии).
- Реализация (модели, еще более детализирующие спецификации на уровне информационных технологий).

Для каждого из представлений программный продукт ARIS предусматривает различные виды диаграмм.

ПРИМЕР

Модель организационной структуры будет содержаться в представлении Организация, а модель знаний персонала будет доступна в представлении Данные.

	Требования	***Спецификации***	***Реализация***
Организация	*Календарь смен*	*Топология сети*	*Диаграмма сети*
Данные	*Диаграмма категорий стоимости*	*Диаграмма реляционных таблиц*	*Диаграмма таблиц*
Процессы	*Диаграммы окружения функции, диаграммы eEPC*	*Диаграмма доступа*	*Модель физического доступа*
Функции	*Диаграмма приложений SAP*	*Типы прикладной системы*	*Диаграмма прикладной системы*

Для категории Продукты / Услуги существует уровень требований, примером которого может служить формируемое Дерево продуктов.

Все базовые возможности ARIS реализованы в программном продукте ARIS Express, полнофункциональная версия которого находится в свободном доступе. В ней пользователи могут создавать 9 базовых типов диаграмм:

- Организационная диаграмма.
- Диаграмма данных ERM.
- Диаграммы BPMN 2.0.
- Процессный ландшафт (цепочка добавленной стоимости VACD).
- Системный ландшафт (диаграмма компонентов).
- Иерархическая диаграмма активностей (whiteboard).
- Диаграмма бизнес-процессов EPC.
- Диаграмма ИТ-инфраструктуры (сети).
- Диаграмма общего вида.

Также представляются такие возможности, как печать моделей, экспорт в форматах PDF & EMF, создание собственных шаблонов диаграмм, панель инструментов моделирования, онлайн-поддержка, а также автоматическая генерация моделей из определенных форматов документов и импорт MS Visio.

Гораздо более расширенным с точки зрения функциональности является семейство программных продуктов ARIS Platform (в 2013 году – ARIS

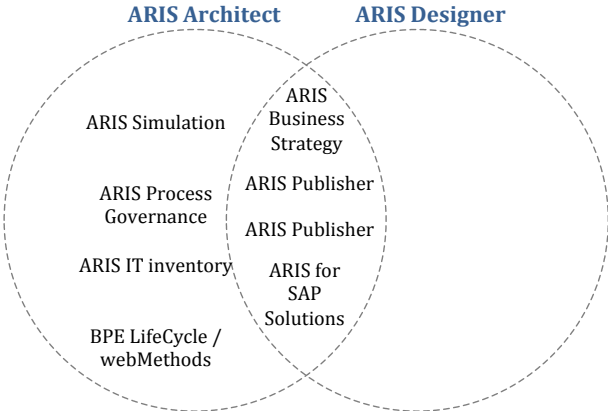
Architect & ARIS Designer). Список доступных для создания диаграмм гораздо больше и он исходит из идеи, что любая организация может быть описана с помощью целой иерархии моделей (от верхнего уровня цепочки добавленной стоимости VACD до диаграмм конкретных процедур и ресурсного окружения).

В нем доступны также центральный репозиторий моделей, версионность, поддержка множества языков, многопользовательская работа, версионность, конструктор отчетов WYSIWYG, конфигурируемые мета-модели и многие другие возможности.

Однако помимо стандартной функциональности формирования моделей платформа ARIS содержит дополнительные расширения:

Дополнительные инструменты / возможности	Предоставляемые преимущества
Процессная аналитика	Визуализация проблем производительности Получение отчетов / оповещений о достижении критических отметок показателей процессов Мониторинг данных, процессов и их ключевых индикаторов (например, функционально-стоимостной анализ затрат)
Управление рисками	Управление бизнес-операциями, рисками и требованиями, контроль исключений
Управление политиками и соответствием требованиям регуляторов	Формирование карт политик в бизнес-контексте с разграничением зон ответственности Сочетание политик управления требованиями, рисками и процессами Ведение журнала аудита с возможностью формирования сложных отчетов (в том числе, по контрольным панелям)
Управление взаимодействием: создание рабочей платформы коллаборации	Организация общих обсуждений процессов / приложений / данных Представление моделей процессов в формате web-страниц Публикация моделей процессов в интранет-сети компании Возможность для специалистов компании предложить свои улучшения в процессах
Симуляция	Возможность «прогона» (симуляции) моделей BPMN 2.0 и EPC для выявления различий моделей As-Is и To-Be Получение статистики и сводной информации по результатам симуляции моделей в режиме реального времени Возможность проведения сценарного анализа «Что если» для определения степени зависимости результата и показателей процессов от определенных факторов/групп факторов
Моделирование бизнес-правил	Возможность определения логики бизнес-правил и связывания сформированных с их помощью структурированных моделей с электронными таблицами (например, для анализа стоимости процесса)
Управление оптимизацией бизнеса	Интеграция информации о структуре и значениях ключевых показателей эффективности процессов, моделях бизнес-процессов, информации о центрах затрат для поддержки принятия стратегических решений

Все эти возможности представлены в виде дополнительных расширений к ARIS Architect&Designer:



Кроме того, для формирования специфических аналитических отчетов, регламентов процессов, создания базы моделей по спецификациям возможно использовать технологию ARIS Script и писать сценарии (подпрограммы) на языке SAX Basic.

CA ERWIN DATA MODELER

Среди ключевых функций:

- Визуализация структур данных.
- Создание проектов баз данных (по графическим моделям).
- Определение стандартов (например, шаблоны моделей).
- Сравнение моделей и баз данных.
- Отчетность и публикация (в том числе в HTML-виде с экспортом изображений графических моделей и метаданных).

Данный программный продукт интегрирует возможности BPWin и ERWin, обеспечивая также создание логических и физических моделей и трансформацию между ними.

IBM RATIONAL ROSE

Семейство продуктов IBM Rational Rose, предлагая средства разработки приложений, основывается на Unified Modeling Language:

- Rational Rose Enterprise.
- Rational Rose Modeler.
- Rational Rose Technical developer (Java/C/C++) & Developer for Java/UNIX/Visual Studio.

Одним из ключевых преимуществ IBM Rational Rose является возможность не просто моделирования, но и создания кода на основе UML, значительно упрощая и ускоряя процесс разработки ПО и формирования документации. В этом программном продукте доступно создание концептуаль-

ных и физических моделей структуры данных, а также диаграмм классов / состояний / сценариев и других моделей UML, что крайне важно в процессе разработки.

Среди основных компонент программного продукта:

- Репозиторий.
- Графический интерфейс пользователя.
- Средства просмотра проекта (для перехода между диаграммами).
- Средства контроля проекта (автоматический поиск / устранение ошибок логики диаграмм в ходе моделирования).
- Средства сбора статистики.
- Генератор документов.

Также отдельно доступен генератор кодов (для определенных языков программирования) и анализатор с возможностью обратного инжиниринга – формирования модели по исходному коду программы.

2.4.2.3. Сбор требований

Данная стадия подразумевает несколько аспектов – от анализа задачи и специфики деятельности предприятия до анализа требований к самой системе. Проектные спецификации и первые наброски технического задания на данном этапе составляются по результатам проведенных интервью и в дальнейшем уточняются для проведения оценки сроков, стоимости и содержания проекта в части функциональности системы.

Для того, чтобы правильно сформировать стоящие перед целевой информационной системой задачи и определить ее ключевые функции, критично проведение объективного и всестороннего сбора требований.

В IEEE Standard Glossary of Software Engineering Terminology (1990) понятие «требование» трактуется следующим образом.

«**Требование** – это:
- условия или возможности, необходимые пользователю для решения проблем или достижения целей;
- условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
- документированное представление условий или возможностей для пунктов 1 и 2»

По сути, требования представляют собой определенный набор входных данных, на основании которых проводится проектирование ИС. Существует множество источников подобных данных (ведь в конечном итоге необходимо создать систему для большого числа пользователей с различными задачами), в связи с чем исходная информация является достаточно противоре-

чивой, неструктурированной, изменяющейся во времени. Однако без этого этапа начать работу невозможно, так как именно согласие заказчика и разработчика-исполнителя по вопросам объема / содержания работ, временных и финансовых ограничений является критическим для выполнения проекта автоматизации.

FURPS+

Для охвата всех аспектов проектируемой ИС используется специальная классификация уровней требований (FURPS+, изначально предложена в начале 1990-х годов специалистом Hewlett Packard Робертом Грейди) [49].

Сокращение **FURPS** является аббревиатурой и в переводе на русский включает пять следующих аспектов: Функциональность – Удобство использования – Надежность – Производительность – Поддержка.

Рассмотрим каждый из уровней более детально.

- **Functionality, функциональность.**
  - Функциональные возможности системы (например, управление маркетинговыми компаниями, анализ клиентской базы, планирование контактов с клиентами, поиск по ключевым словам).
- **Usability, удобство использования** (характеристики пользовательского интерфейса).
  - Интуитивность.
  - Эстетичность.
  - Мультимедийность (использование анимации, графики, видео и других форматов контента).
  - Защита от ошибок (человеческого фактора).
  - Удобство обучения.
  - Полнота системной документации и руководств пользователя.
  - Легкость получения справочной информации.
- **Reliability, надежность.**
  - Доступность (минимально необходимое время бесперебойной работы системы – к примеру, 24 × 7).
  - Точность расчетов.
  - Отказоустойчивость.
  - Восстанавливаемость системы (частота резервного копирования данных).
- **Performance, производительность.**
  - Пропускная способность (число одновременно работающих пользователей).
  - Время отклика системы (время реакции системы на совершенное пользователем действие).
  - Время восстановления.

  - Масштабируемость (к примеру, при увеличении числа пользователей в случае внедрения в других филиалах).
  - Потребление ресурсов (к примеру, оперативной памяти).
- **Supportability, поддерживаемость.**
  - Ремонтопригодность.
  - Адаптируемость (скорость приспособления к условиям работы в конкретной среде).
  - Совместимость.
  - Легкость установки.
  - Конфигурируемость (гибкость и оперативность изменения параметров системы, добавления новых ролей, ограничения доступа, изменения функций).
  - Локализация (поддержка языков, валюты, настроек времени и дат выбранной страны).
  - Поддержка стандартов и требований регуляторов (например, поддержка SCORM – международного стандарта обмена электронными курсами как требование к системе дистанционного обучения).
  - …
- **+, дополнительные ограничения.**
  - Ограничения проектирования, design.

ПРИМЕР
Использование реляционной БД в качестве основной.

  - Ограничения разработки, implementation.

ПРИМЕР
Ориентация на выбранную методологию разработки (RUP/MSF), языки программирования (C#, Python, …).

  - Ограничения на интерфейсы, interface (накладываются пользователями или другими системами).

ПРИМЕР
Использование определенных форматов данныхю

  - Физические ограничения, physical (например, к аппаратным средствам и окружающей среде).

ПРИМЕР
Требования к уровню влажности / температурному режиму в комнате, где находится аппаратное обеспечение / окружение системы.

Требования по Карлу Вигерсу

Все вышеописанные требования (кроме первого, Functionality) по сути являются нефункциональными (URPS+). Эти две категории, в свою очередь,

легли в основу другой классификации требований (предложенной Карлом Вигерсом, известным лектором и консультантом в области сбора требований и совершенствования процесса разработки ПО). Среди **функциональных требований** он выделяет три аспекта:

– **Бизнес-требования,** формулируемые чаще всего спонсором / заказчиком проекта. Это цели, которые преследует создание системы, какие преимущества необходимо таким образом достичь и какие задачи решить. Таким образом, формируется общий образ и границы проекта, которые также могут быть закреплены в уставе проекта.

ПРИМЕР

На уровне технического задания могут быть сформированы требования к поддержанию бизнес-процессов. Так, в CRM-системе можно выделить коммуникационные процессы (информирование и регистрация и обработка обращений клиентов), отчетные и управленческие процессы.

– **Пользовательские требования.**

Решаемые системой задачи для различных категорий пользователей. В данном случае требования могут быть представлены как в виде сценариев (user journey), алгоритмов (для учета всех возможных вариантов действий), таблиц «событие-отклик».

ПРИМЕР

Могут также описываться ключевые роли, такие как: «Оператор в банке», «Инвестор», «Партнер», «Клиент – физическое лицо», «Клиент – юридическое лицо». В зависимости от выделяемых групп пользователей будут различаться и их возможности в системе.

– **Функциональные требования.**

Определяемая функциональность системы, описываемая затем детально в техническом задании для реализации разработчиком.

Выделяемые Вигерсом нефункциональные требования являются дополнительными аспектами, которые необходимо принимать во внимание при разработке.

– **Бизнес-правила.**

Требования регуляторов (например, экологические нормы для производственных компаний), промышленные стандарты, корпоративные стандарты и политики как налагаемые внешней средой ограничения.

– **Атрибуты качества.**

Дополнительные требования к программному продукту, не относящиеся к задаваемой классом систем функциональности (к примеру, для систем документооборота это приём, хранение, пересылка, контроль исполнения документов), а являющиеся необходимыми для эффективности системы характеристиками (целостность, интероперабельность с другими системами, интегрируемость).

– **Ограничения.**

Преимущественно технические ограничения, накладываемые условиями, в которых система должна быть реализована

ПРИМЕР

Диктуемые целевыми параметрами производительности протоколы, серверы приложений.

При этом источниками информации для функциональных требований могут служить описания деловых (бизнес-) объектов (подразделений, должностей, ролей, процессов, систем, носителей информации) с указанием их взаимосвязей, а вид моделей и требования к ним определяются целями моделирования. В то же время, нефункциональные требования создаются по результатам обследований предприятия (интервью, анкетирование, наблюдение за типовым рабочим днем сотрудника).

Для описания необходимых шагов корректного формирования требований обратимся к мировым практикам, описанным в сборнике **SWEBOK**. Данный документ объединяет знания в области разработки ПО и будучи разработанным комитетом сообщества IEEE Computer Society, призван определить набор знаний и рекомендуемые практики по пяти основным стадиям ЖЦ, а также управлению конфигурацией, качеством, ИТ-проектом, в том числе с описанием процесса разработки ПО и используемых при этом методов и инструментов.

Этапы формирования требований по SWEBOK

Согласно SWEBOK, работа над требованиями предполагает 3 основные активности: их сбор (по результатам интервью с представителями заказчика и обследования предприятия), анализ (в т.ч. на полноту, непротиворечивость) и документирование (в виде сценариев использования, user journeys, спецификаций процессов).

Как правило, в сборе требований со стороны заказчика принимают участие ключевые специалисты, наиболее четко и ясно представляющих бизнес-деятельность своего подразделения и грамотно формулируя требования для дальнейшей корректной их интерпретации группой внедрения. Важно отметить, что практически в любой организации можно встретить пользователей – экспертов, которые не просто готовы пользоваться системой или даже принимать участие в ее улучшении, но хотят полностью подстроить ее под свои нужды – а значит, важно соблюдать баланс и реалистично представлять целесообразность добавления той или иной функции.

ПРИМЕР

В организации АВС с 300 сотрудников до внедрения единой системы учет всех проектов вели вручную. Каждый проектный менеджер отвечал за 10-15 видов активностей и либо заносил их в Excel в свободном формате, либо пользовался

онлайн-сервисами, либо вообще просто все важные вещи записывал в свой ежедневник. После того, как руководством было принято решение об интеграции всех данных в единую систему и подключении к работе с ними филиала предприятия, эта идея была встречена достаточно негативно. Помимо того, что сотрудники чувствовали, что их хотят контролировать, мало кто представлял, что делать с теми данными, которых будет не хватать для корректной работы с системой – или с той информацией, которую никто кроме них не собирал (например, даты промежуточных проектных встреч). Более того, одно и то же поле (например, «перспективность проекта») не всегда заполнялось. К примеру, 60 % пользователей заполняли без единого формата ввода:

- *Менеджер A ставил цифры от 1 до 5 (5 – высшая оценка, самый перспективный).*
- *Менеджер B ставил цифры от 1 до 5 (**но** 5 – самый низкий приоритет, 1 – высший приоритет).*
- *Менеджер C выбирал из вариантов «Перспективен для рынка Европы», «Перспективен для рынка США и Китая», «Перспективен в России», «Под вопросом», «Не перспективен», вариант «Другое»).*
- *Менеджер D же просто писал текстовые комментарии, когда по его мнению конечный продукт, разрабатываемый в ходе проекта, может выйти на рынок.*

Однако из 50 сотрудников 5 основных подразделений двое специалистов в свое время разработали по сравнению с остальными сложную систему ведения проектов, включающую макросы для объединения данных нескольких Excel-файлов, и естественно, хотели чтобы именно их принцип организации данных был взят за основу. Так, если они ввели собственные развернутые классификации зрелости и перспективности проектов с drop-down меню и 15 варинтами ответа, именно ее они предлагали взять за основу при определении формата соответствующих полей. Их мнение, безусловно, важно – и если они начинают пользоваться системой (а не бойкотировать ее), «рядовые» пользователи их подразделений, несомненно, присоединятся. Но насколько такая система будет удобна для них, насколько уровень детализации и классификации, «навязанные» коллегами, соответствуют потребностям массовых пользователей, в особенности других подразделений и филиалов?

Важно помнить, что несмотря на важность получения поддержки ключевых пользователей, в конечном итоге система внедряется именно в интересах всех пользователей: именно их регулярная работа в системе и импортированные ими данные являются базой для достижения эффективности системы.

Если, как уже было сказано, компания-заказчик пришла к решению о самостоятельном сопровождении системы на этапе эксплуатации, именно эти ключевые сотрудники будут принимать систему, для чего необходимо отчетливо понимать, как те или иные действия пользователей отразятся на работе системы и каким образом происходит движение информации в ней. И говоря о движении информации, вновь вернемся к этапу анализа и постановки задачи, на котором проводится описание автоматизируемых бизнес-процессов и актуальных для них потоков данных.

2.4.2.4. Подготовка технического задания

Техническое задание является достаточно регламентированным документом и рекомендации по его формированию приведены в российском ГОСТ 19.201-78 «Техническое задание, требования к содержанию и оформлению» и ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» (ТЗ на АС, которое подробнее рассмотрено ниже).

Техническое задание (**проектное решение**) на выполнение работ является основным документом, определяющим порядок создания, настройки, доработки и внедрения модуля системы (системы в целом).

- общие сведения о модуле / системе;
- назначение и цели настройки (создания, доработки) модуля / системы (вид автоматизируемой деятельности (управление, проектирование и т.п.) и перечень объектов автоматизации);
- характеристики объекта автоматизации (наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть выполнены в результате создания системы, критерии оценки достижения целей ее создания);
- требования:
  - к модулю / системе:
    - к структуре и функционированию;
    - к численности и квалификации персонала заказчика для работы с системой;
    - к надежность и безопасности / отказоустойчивости;
    - к эксплуатации и техническому обслуживанию;
    - к безопасности информации;
    - к непрерывности бизнеса / сохранности информации при сбоях или авариях;
    - к функциональности:
      - перечень функций и задач, подлежащих автоматизации;
      - сроки реализации каждой функции (задачи);
      - форма представления выходной информации (и ее характеристики);
  - к проекту: информационное, программное, техническое, организационное обеспечение, включая рекомендации по техническим характеристикам стенда предприятия и рабочим местам пользователей;
  - к составу и содержанию работ:
    - по созданию и настройке системы (включая порядок и критерии ее приемки в эксплуатацию);

  - по подготовке объекта автоматизации к вводу системы в опытно-промышленную и промышленную эксплуатацию;
- к проектной документации.

Техническое задание систематизирует все функциональные и нефункциональные требования, тем самым обеспечивая базу для дальнейших активностей по созданию и внедрению системы.

2.4.3. Проектирование

Проектирование включает в себя два основных аспекта: техническое и рабочее проектирование. В первом случае проводится описание технических деталей (к примеру, модели данных и интерфейса), а также моделирование ситуаций, в которых система будет использоваться – для дальнейшей разработки сценариев использования системы и ее тестирования на их основе. При рабочем же проектировании могут составляться блок-схемы и алгоритмы для программных модулей.

В целом же техническое проектирование ИТ-системы предполагает детальную подготовку к этапу настройки / развертывания модулей системы, для чего необходимо решить несколько задач:

- Сформировать список модулей и функций системы, необходимых для поддержки определенных на этапе анализе автоматизируемых бизнес-процессов.
- Сформировать список справочников систем (будущей и если применимо, текущей) для дальнейшего переноса и обновления данных.
- Определить примерный сценарий работы системы по категориям пользователей для формирования необходимого набора диалогов и процедур проектируемой системы (включая реакции на все возможные и даже очень маловероятные действия со стороны пользователя).
- Определить элементы интерфейса пользователей (в случае гибкого или разрабатываемого «с нуля» решения) для достижения удобства работы с системой.
- Сформировать список отчетов / панелей мониторинга (включая их формы и обязательные для реализации элементы). Эти отчеты в дальнейшем будут использоваться как для учетных целей, так и для целей мониторинга системы ее администратором (в части формирования и сбора статистики по нагрузке на ее отдельные модули, свободных ресурсах, активности пользователей и т.п.).
- Определить перечень настроек функциональных компонентов системы в соответствии со выделенными на предыдущем этапе требованиями.

– Определить необходимость, возможности и пути интеграции с существующими и планируемыми к реализации системами на предприятии заказчика, чтобы своевременно предусмотреть технические средства для интеграции.

На этапе детального проектирования важно крайне четко определить все функциональные возможности системы и определить ее место в общей программной архитектуре предприятия.

ПРИМЕР

Если предполагается, что система взаимодействия с клиентами CRM будет получать данные из личных кабинетов клиентов на портале компании, необходимо предусмотреть подобную интеграцию. Если в компании уже внедрена и работает ERP-система, CRM должна получать и предоставлять ей данные.

С организационной перспективы к этому моменту важна подготовка иерархической структуры работ / базового календарного плана / дорожной карты, которые бы позволили управлять проектом на протяжении всей его реализации.

ПРИМЕР

Календарный ресурсный план.

Роль	Модуль	Проектная команда	ФИО	Итого, дней	Июл-09	Авг-09	Сен-09	Окт-09	Ноя-09	Дек-09	Янв-10	Фев-10
Консультант												
		Управление программой и проектный офис										
Партнер проекта	PMO	Партнер программы	TBD*	13	2	1	2	2	2	2	1	1
Руководитель проекта	PMO	Директор программы	TBD*	135	20	10	20	20	25	20	10	10
Администратор	PMO	Администратор проектного офиса	TBD*	135	20	10	20	20	25	20	10	10
			Итого:	**283**								
		Контроль качества										
Контроль качества	QA	Партнер по контролю качества	TBD*	6	-	2	-	2	-	-	2	-
Контроль качества	QA	Индустриальный эксперт	TBD*	3	2	-	-	1	-	-	-	-
			Итого:	**9**								
		Процессы и приложения										
		Финансы и контроллинг										
Руководитель команды	FI/CO	Руководитель команды	TBD*	135	20	10	20	20	25	20	10	10
Консультант	FI/CO	Основные средства	TBD*	145	20	10	20	20	25	20	10	20
Консультант	FI/CO	Дебиторы\Кредиторы, НДС	TBD*	135	20	10	20	20	25	20	10	10
Консультант	FI/CO	Дебиторы\Кредиторы	TBD*	145	20	10	20	20	25	20	10	20
Консультант	FI/CO	Главная книга	TBD*	135	20	10	20	20	25	20	10	10
Консультант	FI/CO	Казначейство	TBD*	145	20	10	20	20	25	20	10	20
Консультант	FI/CO	Калькулирование себестоимости	TBD*	135	20	10	20	20	25	20	10	10
Консультант	FI/CO	Накладные расходы	TBD*	145	20	10	20	20	25	20	10	20
Консультант	FI/CO	Бизнес аналитик	TBD*	145	20	10	20	20	25	20	10	20
Консультант	FI/CO	Бизнес аналитик	TBD*	145	20	10	20	20	25	20	10	20
		Управление цепочкой поставок										
Руководитель команды	MM	Руководитель команды	TBD*	135	20	10	20	20	25	20	10	10
Консультант	MM	Управление запасами	TBD*	135	20	10	20	20	25	20	10	10
Консультант	MM	Закупки	TBD*	145	20	10	20	20	25	20	10	20

2.4.3.1. Техническое проектирование

Цель данного этапа – подготовка к этапу настройки модулей системы в целом. Данный этап предусматривает разработку организационной и функциональной структуры модуля системы, определение уровня автоматизации информационного процесса предприятия, определение структуры информационного и технического обеспечения и требований к их элементам, постановку и алгоритмизацию задач обработки данных, разработку системы ведения нормативно-справочной базы.

Результаты работ оформляются в виде технического проекта (проектного решения на автоматизацию модуля системы) и представляют собой задание на программирование. Техническое проектирование, создание, настройка, доработка и внедрение модуля системы осуществляются только на основании соответствующих утвержденных технических заданий (проектных решений).

Технический проект (проектное решение по автоматизации модуля) – документ, содержащий описание функциональных компонентов системы, необходимых для реализации бизнес-процессов, описанных на этапе подготовки проектного решения, а также необходимых доработок системы и (при необходимости) процедур интеграции с модулями (или внешними системами), внедренными ранее.

Перечень документов, создаваемых на стадии «Технический проект», определяется документом ГОСТ 34.201-89. Основной задачей является разработка архитектуры системы и технических решений по ее реализации. Перечень документов, служащих базой технического проекта, определяется условиями договора / технического задания, но чаще всего в него включаются следующие документы:

1. Пояснительная записка.

– Общие положения.

ПРИМЕР

Стадии и сроки, цели и задачи, используемые при разработке системы объекты ИС.

– Описание процесса деятельности.
– Основные технические решения.

ПРИМЕР

Структура системы, основные функции, режимы функционирования, требования к персоналу.

– Мероприятия по подготовке объекта автоматизации к вводу системы в действие.

ПРИМЕР

Подготовка информации к импорту, создание рабочих мест.

2. Входные и выходные данные системы.

ПРИМЕР

Импортируемые и вводимые вручную пользователями данные (названия и источники документов), формируемые системой документы.

3. Схема функциональной структуры.

– Описание функций подсистем.

ПРИМЕР
Описание функций по отдельным модулям (например, модуль расчета заработной платы, модуль учета труда).

– Информационные связи между элементами и с внешней средой.

ПРИМЕР
Условия обмена информацией с внешними системами.

4. Описание автоматизируемых функций.
– Исходные данные.
– Цели АС и автоматизируемые функции.
– Характеристика функциональной структуры.

ПРИМЕР
Список подсистем, описание процесса реализации функций, требования (к надежности, защите информации от несанкционированного доступа, сохранности информации).

– Типовые решения.

5. Постановка задач и алгоритмы решения.
– Характеристики комплекса задач.
– Используемая информация.
– Результаты решения.

ПРИМЕР
Информация для выдачи выходных сообщений и данные исключительно для использования внутри системы.

– Алгоритм решения.

6. Программное обеспечение.
– Структура ПО.
– Функции частей ПО.
– Методы и средства разработки ПО.

ПРИМЕР
Инструментальные средства проектирования модели предметной области, структур баз данных, разработки.

ПРИМЕР
Применяемые методологии разработки, проектирования системы / интерфейсов.

– Операционная система.

ПРИМЕР
Совместимые / предпочтительные операционные системы для клиентской части ПО и серверов (баз данных).

– Средства, расширяющие возможности ОС.

7. Информационное обеспечение.

– Состав информационного обеспечения.
  - **Внутримашинная** информационная база – набор электронных таблиц и других объектов баз данных для обработки и хранения данных.
  - **Внемашинная** информационная база – информация, поступающая в бумажном виде.

ПРИМЕР
Нормативно-справочная информация, годовые бюджеты.

– Организация информационного обеспечения.
  - Принципы организации информационного обеспечения.

ПРИМЕР
База данных должна представлять собой взаимосвязанные реляционные таблицы.

  - Обоснование выбора носителей данных.

ПРИМЕР
Расчет числа запросов, которые необходимо обрабатывать, на основе статистики предыдущих периодов по числу пользователей, объему информации, частоте обращения к ней.

  - Принципы и методы контроля в маршрутах обработки данных.

ПРИМЕР
Контроль форматов данных, контроль заполнения обязательных полей.

  - Описание решений, обеспечивающих информационную совместимость АС с другими системами.

– Организация сбора / передачи информации.
  - Основные источники информации.

ПРИМЕР
Информация из внешних систем, бумажные носители информации, передаваемые по электронной почте файлы.

  - Общие требования к организации сбора, передачи, контроля и корректировки информации.

ПРИМЕР
Импорт данных в формате XML.

– Построение системы классификации и кодирования.
  - Классификации объектов, принятые для применения в АС.

ПРИМЕР
Общероссийский классификатор валют (ОК 014-94), Коды представления названий стран (ИСО 3166-93), внутриорганизационный справочник сотрудников.

- Классификации объектов, принятые для применения в АС.

ПРИМЕР

Типы документов обозначаются целыми положительными числами от 1 в возрастающей последовательности.

– Организация внутримашинной информационной базы.
  - Принципы построения.
  - Структура.

ПРИМЕР

Физическая структура базы данных системы (с полями, типами данных).

– Организация внемашинной информационной базы (информации в бумажном виде от внешних организаций).

8. Комплекс технических средств системы.

- Структура комплекса технических средств.

ПРИМЕР

В состав подобного комплекса могут включаться: серверы баз данных, приложений, веб-сервер, персональные компьютеры. Принципы построения кластеров.

– Вычислительный комплекс.
– Абонентские пункты.
– Аппаратура передачи данных.

9. Ведомость документов.

В результате подготовки технического проекта должен быть определен перечень компонентов модуля системы, необходимых доработок, процедур интеграции с существующей системой (при ее наличии и необходимости), необходимый для реализации бизнес-модели предприятия To-Be. Также однозначно утверждается перечень работ по настройке и доработке модуля системы и перечень / формы отчетов и первичных документов, получаемых из модуля системы. Помимо отчетов, должны быть определены процедуры для справочников системы (при ее наличии), которые необходимо использовать для импорта, и тех справочников, для которых необходимо разработать механизмы двустороннего обновления информации.

Важно отметить, что после детального определения и согласования всех вышеописанных параметров необходимо к моменту начала следующего этапа выбрать и подготовить также среду разработки и тестирования для проведения развертывания системы.

2.4.3.2. Рабочее проектирование / прототипирование при заказной разработке

В целях снижения риска создания не соответствующей требованиям и техническому проекту системы перед разработкой возможно создать ее про-

тотип (макет) для проверки предлагаемых архитектурных / функциональных / интерфейсных решений на практике с будущими пользователями.

Прототип – «черновая» реализация интерфейса и базовой функциональности системы для анализа принципов ее работы и тестирования совместно с будущими пользователями в целях дальнейшей доработки и совершенствования.

Обратная связь в рамках прототипирования, организованная на первых этапах жизненного цикла, очень важна для устранения замечаний по итогам проектирования и внесении доработок в интересах пользователей и заказчиков системы. В первую очередь создается макет, а уже потом проводятся тестирование / обсуждение и внесение корректировок в спецификации и прототип. Рассмотрим основные виды / типы прототипов.

В качестве первой классификации можно выделить «горизонтальное» и «вертикальное» прототипирование. Первый случай («**горизонтальное**» прототипирование) предполагает моделирование исключительно пользовательского интерфейса и форм без фокусирования на логике обработки информации. Имитируются результаты действий / запросов при нажатии на элементы управления и осуществляются переходы между формами системы для формирования представления, каким образом система реагирует на те или иные действия пользователя (и какие не предусмотренные в текущей реализации функции / действия потенциальный пользователь может совершить). Таким образом, возможно проверить и устранить противоречия между сформированными на этапе анализа требованиями и их реализацией в техническом проекте. А в рамках «**вертикального**» прототипа, в отличие от предыдущего варианта, фокус переходит на саму структуру системы, используемые среды / языки программирования для проверки корректности и работоспособности сформированного архитектурного решения.

Второй (и более известной) классификацией является «быстрое» и «эволюционное» прототипирование. В первом случае (как и следует из названия) по технологии RAD, более подробно рассматриваемой в главе методологий разработки, создается макет определенных компонентов системы для более предметного и эффективного диалога между разработчиками кода / интерфейсов / пользователями. Важно, что подобный прототип не призван в дальнейшем дорабатываться и становиться частью системы, соответственно, достигается значительная экономия времени и ресурсов, так как нет необходимости фокусироваться на технических деталях быстродействия или энергоэффективности системы.

Напротив, эволюционное прототипирование, по сути являясь итерационной разработкой, подразумевает создание в несколько итераций кода и тестирование полноценной рабочей системы, с реализованными алгоритмами и целевым уровнем производительности.

2.4.4. Разработка

Ко времени старта реализации системы, как уже было сказано, должна быть выбрана и подготовлена среда разработки, а также выбрана методология, в соответствии с которой будет осуществляться управление разработкой. От нее зависит очень многое: принцип взаимодействия команды разработки / внедрения (между собой и с основными стейкхолдерами проекта), следование одной из моделей жизненного цикла ИС (каскадной, итерационной и пр), длительность самого процесса разработки и тестирования и прочие важные аспекты процесса реализации системы. Подробнее различные методологии как управления проектами, так и непосредственно разработки будут рассмотрены в следующей главе.

2.4.4.1. Закупка ПО

Как уже было сказано, в рамках данной монографии будем фокусироваться исключительно на ситуации, когда организуется закупка стандартного («тиражируемого») решения и его последующая настройка под бизнес-процессы компании. Случаи заказной разработки системы «с нуля» не рассматриваются.

Каждая организация выбирает собственные, оптимальные именно с учетом ее специфики критерии выбора между рядом платформ/решений, среди наиболее распространенных из которых:

- Функциональность системы.
- Надежность / стабильность работы решения (устойчивость к различным режимам работы и степени активности пользователей).
- Наличие встроенных средств администрирования / управления.
- Стоимость (внедрения, лицензий, поддержки).
- Удобство организации услуг по поддержке и сопровождению системы.
- Наличие отраслевого решения.
- Наличие успешного опыта реализации проектов на базе данной платформы (надежность системы).
- Опыт и компетенции организаций-подрядчиков.
- Учет отраслевой / корпоративной специфики / соответствие корпоративным стандартам.

Список критериев очень индивидуален в зависимости от специфики проекта и как правило, формируются из разработанных на этапе проектирования требований к системе.

Следующим же шагом будет организация непосредственно процедуры закупки (например, путем открытого запроса предложений), выбор поставщика и заключение договора на поставку ПО.

Типовая структура подобного договора приведена ниже.

1. Предмет договора.

Например, поставка, установка ПО, монтаж оборудования, обучение персонала заказчика.

2. Стоимость договора и порядок оплаты.

Сумма оплаты, сроки и размер платежей, форма оплаты (наличный / безналичный платеж).

3. Права и обязанности сторон.

Порядок поставки / установки / ввода в эксплуатацию ПО, предоставления информации со стороны поставщика. Порядок назначения ответственных лиц, предоставления информации со стороны покупателя.

4. Ответственность сторон.

Условия выплаты неустойки / расторжения контракта в случае невыполнения или ненадлежащего выполнения обязательств (в т.ч. несвоевременная поставка или оплата). Порядок разрешения споров, в т.ч. в случае действия обстоятельств непреодолимой силы.

5. Гарантийное сопровождение / авторский надзор.

6. Условия конфиденциальности.

2.4.4.2. Настройка конфигураций

К моменту инсталляции системы на стенде предприятия, заказчиком должна быть обеспечена устойчивость работы ЛВС стенда, проведена закупка программно-технического обеспечения в виде основного внедряемого программного продукта (самой системы) и ключей электронной защиты, а также завершены все работы предшествующих фаз жизненного цикла. Для инсталляции / тестирования системы программно-техническое обеспечение, в том числе клиентские лицензии предоставляет заказчик (из числа закупленных ранее либо через осуществление их закупки на этом этапе).

В процессе инсталляции модули системы устанавливаются на стендовый сервер с проведением основных необходимых настроек под подлежащие автоматизации бизнес-процессы предприятия заказчика. В зависимости от типа и особенностей системы в ней могут быть реализованы разные механизмы дополнения / изменения объектов в системе.

Создание кода программы

Единого языка программирования, использующегося для всех приложений, просто на бывает. Сейчас для многих стандартных пользовательских приложений используется комбинация SQL и C# для серверной программной части и HTML/CSS/JavaScript для создания интерфейса пользователей.

ПРИМЕР

Современные ERP-системы наиболее часто используют собственные языки программирования, в связи с чем поддерживающие их специалисты должны обладать знаниями не только по работе с формами/ интерфейсом пользователей, но и в первую очередь по структуре системы и быть способными безопасно внести необходимые изменения в код.

- *SAP – ABAP.*
- *Oracle – PL/SQL, Java.*
- *Microsoft Dynamics AX – среда разработки MorphX с языком программирования X+.*
- *Microsoft Dynamics NAV – графическая среда разработки C/SIDE с языком программирования C/AL.*

Код программы может являться самодостаточным способом настройки системы либо же дополняться т.н. «конфигуратором» или «конструктором» системы. Разница между ними в возможностях внесения изменений (бизнес-логика и правила работы приложения, как правило, требуют большего и более сложного объема работы, нежели просто добавление нового поля). Для более простых же функций используются, как правило, стандартные конфигураторы системы, не требующие знания языка программирования.

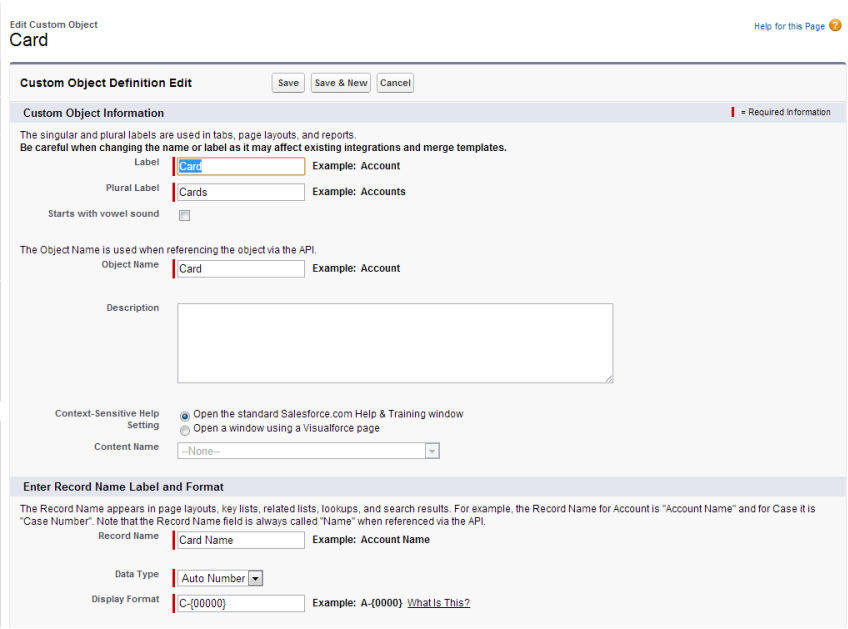
Таким образом, может быть выстроена следующая иерархия возрастания сложности и функциональных возможностей внесения изменений:

Интерфейс пользователя → Конфигуратор → Код программы

Разумеется, крайне важно, чтобы конфигурация была тщательным образом документирована, особенно в части «надстроек» над стандартными функциональными возможностями, описываемыми в том числе в поставляемой вендором документации. Также необходимо обеспечить такую доработку кода, которая бы не содержала в себе элементы т.н. «закрытого кода», недоступного для изменения. В противном случае если в ходе разработки была создана определенная схема расчета или логика процесса, то в случае необходимости внесения изменений ИТ-специалистам придется с нуля создавать требуемый код / функциональность. Именно поэтому специалисты должны быть обучены не только технике конфигурирования продукта, но и самим особенностям конкретной конфигурации.

Настройка параметров системы в режиме «конфигуратора»

Примером системы, работающей в подобном режиме конструктора, является Salesforce.com. В ней разработчик имеет возможность определять множество параметров, таких как: формат данных создаваемых полей, их взаимосвязи, формат отображения, источники данных (ручной ввод, вычисление по формуле или получение информации из других источников). Пример экрана создания подобных полей приведен ниже.



В том или ином виде конфигураторы присутствуют практически во всех системах, но объем их возможностей (и требования к квалификации работающих с ними специалистов) значительно различаются.

1. Настройка полей и создание библиотек системы.

Наиболее простой из типов конфигурируемой информации. По сути таким образом предоставляется возможность расширить информацию об объектах системы и расширить возможности их взаимодействия. Дополнительные прописываемые поля, типы данных и прочие элементы позволяют кастомизировать каждый объект, а также описать возможные для него действия.

ПРИМЕР

Возможность дополнить карточку сотрудника в системе набором новых полей с выбором типа каждого поля и установки для них ограничений. Так, если в стандартной конфигурации существуют только три поля: Идентификатор сотрудника, Фамилия и Имя, то можно добавить любые дополнительные поля: Дата рождения, Должность, Дата принятия в штат. Для системы в дальнейшем не будет различий, были ли созданы поля автоматически или вручную, и они все в полной мере могут быть использованы в функциях поиска, фильтрации, создания отчетов.

2. Настройка логики бизнес-процессов.

Вторым видом наиболее часто используемых объектов конфигурации являются сами процессы. Например, в системе могут прописываться пути

движения документов, когда после утверждения документа первым согласующим он автоматически направляется второму согласующему с отправкой уведомления о статусе «владельцу» документа в системе.

ПРИМЕР

Приведем пример настройки логики бизнес-процесса выплаты бонусов сотрудникам филиала компании на основе результатов продаж. Бонусы планируются к выплате на периодической основе.

На примере конфигурации в системе SAP в таком случае объектами настройки будут:

- *Создание вида выручки к счету.*
- *Определение схемы перерасчета, схемы учета результатов для заказа.*
- *Определение вида заказа для отражения бонуса.*
- *Ведение профиля расчета, присвоение профиля расчета виду заказа.*

3. Интерфейс пользователя, расположение элементов на вкладках.

В конфигураторах большинства систем предоставляется также возможность изменять расположение объектов, скрывать некоторые из них, создавать персонализированные фильтры / настройка для групп пользователей. Данный шаг подробнее рассматривается в следующем разделе пособия – «*Создание ролей пользователей*».

По результатам конфигурирования системы подрядчиком подготавливаются описание оптимальных настроек сервера баз данных и рекомендации по развертыванию рабочих мест, которое и проводится следующим шагом. Это также позволяет выявить и устранить ошибки и неточности в программном продукте, затем подготовив описание **тестового сценария** (контрольного примера действий пользователя) для проверки корректности и актуальности осуществленных настроек.

Результатом инсталляции и настройки модули системы являются развернутые и готовые к переносу в них справочники и данные, формы ввода данных адаптированы, подготовлен акт сдачи-приемки работ.

2.4.4.3. Создание ролей пользователей

Права доступа (и ролей пользователей) являются именно тем инструментом, который определяет:

- **Какой именно функциональностью системы будет пользоваться** тот или иной сотрудник (группа сотрудников).
- **Какие данные будут ему доступны**.
- **Какие он будет иметь права** на чтение, редактирование информации, ее экспорт (!) и удаление.

Отдельно отметим экспорт информации как предоставляемую системой функциональность, тесно взаимосвязанную с информационной безо-

пасностью – а значит, возможность выгрузки конфиденциальных данных должна быть предоставлена только ограниченному кругу лиц.

С технической точки зрения подобный доступ позволяет одному пользователю иметь несколько ролей по отношению к разным объектам системы.

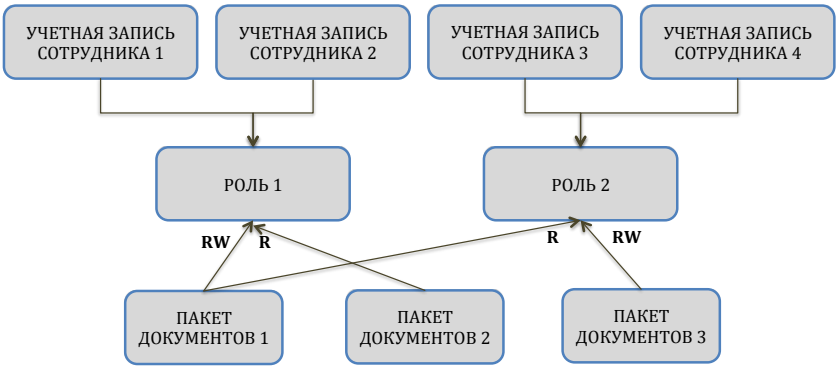
Роли пользователей значительно зависят от профиля компании и специфики системы, однако в целом они различаются по правам доступа к каждой из категорий данных. В списках доступа чаще всего задаются следующие возможности (по возрастанию объема прав):

R (read) – Чтение	Возможность видеть элементы страницы и просматривать прикрепленные файлы
W (write) – Запись	Возможность изменять / редактировать значения полей страницы, добавлять файлы (возможность чтения, разумеется, сохраняется)
D (delete) – Удаление	Возможность удалять страницу вместе с прикрепленными файлами (доступ на чтение / запись при этом сохраняется)
A (admin) – Администрирование	Возможность добавлять / удалять дополнительные поля, менять формат представления, предоставлять и отзывать доступ к элементам (при сохранении возможности чтения, записи и удаления).

Получается логическая связка следующего вида:

Учетная запись пользователя – Роль пользователя – Права доступа (комбинация прав и элементов, к которым осуществляется доступ)

ПРИМЕР



Как видно из данного примера, несколько сотрудников (например, работающие на равных позициях одного подразделения) могут обладать одной ролью в системе (например, аккаунт-менеджера). В таком случае все права доступа будут прописываться не для каждого отдельного пользователя, а для ролей (что гораздо эффективнее с точки зрения трудозатрат, в особенности при внесении изменений).

Далее на примере приведены пакеты документов, к которым определен доступ, однако тот же самый принцип действует и для полей (групп полей) и страниц в системе. Так, в системе CRM аккаунт-менеджеру будет доступна возмож-

ность просмотра, редактирования и удаления мобильных телефонов его клиентов, но может быть отказано в доступе на внесение изменений в финансовые детали осуществления платежей. И напротив, сотрудники бухгалтерии с ролью «специалист по расчетам» смогут редактировать любые финансовые данные, но не будут видеть мобильные телефоны и историю сообщений руководства компании-клиента, которые доступны аккаунт-менеджеру.

2.4.4.4. Миграция данных

Данные на протяжении своего жизненного цикла претерпевают огромное число действий над ними: создание, обработка / редактирование, резервирование, копирование, перемещение в новую виртуальную среду, новую систему. Миграция данных является очень важной составляющей процесса внедрения системы – по сути, на этом этапе создается механизм, позволяющий автоматически переносить основные справочники и другие данные в модули системы.

В первую очередь, важнейшим шагом является непосредственно перенос (первичная загрузка) и настройка **основных справочников**. По сути, это все те данные, которые являются едиными для различных модулей и систем предприятия и используются ими совместно и одновременно: справочники поставщиков, клиентов, складов, материалов, продукции (номенклатуры) и другие. Именно благодаря наличию подобных основных данных не происходит дублирования записей, а в при внедрении систем не приходится заниматься ручным поиском, сбором и вводом этой информации. В целях тестирования основные справочники новой системы заполняются тестовыми данными, достаточными для тестирования модуля и принципов поступления и передачи из него информации, и затем переносятся в систему. Если по результатам выверки тестовых данных не наблюдается ошибок, то можно говорить об успешной миграции справочников и на первый план выходит уже другой вопрос, а именно: поддержание всех этих данных в актуальном состоянии.

Другой категорией данных для передачи является информация из функционирующих на предприятии программных рещений / legacy-систем, на смену которым внедряется новый программный продукт. При корректной подготовке на стадии проектирования передача данных в новую систему организовывается автоматизированно, через специальные коннекторы / шины данных. В противном случае миграция данных становится «ночным кошмаром», когда они должны быть в ручном режиме выгружены из существующих систем, приведены к единому формату, определено соответствие полей и данные загружены в новую систему. Но ведь данные в существующих системах могут обновляться ежесекундно. Насколько допустим этот простой во время миграции и как избежать всех ассоциированных с распределенным онлайн-

вводом? Не говоря уже о проблеме ресурсоемкости и неимоверной трудоемкости данного процесса, связанного с огромным риском потери информации.

В таких случаях миграция организуется в онлайн-режиме. Такой способ предполагает перенос данных при непрерывной работе всех серверов и приложений. Это особенно важно для критических для бизнеса приложений с высокими требованиями к доступности: Oracle, Exchange, биллинговых систем, систем электронной коммерции. Новые данные внедряются в уже существующий поток информации, объединяются с ними на логическом уровне через отдельный адрес диска, который после успешной миграции удаляется или архивируется.

К наиболее распространенным рискам миграции данных и причинам их возникновения можно отнести:

- **Недостаточные компетенции / опыт** команды проекта в миграции данных (в том числе, в части формирования документации об уже осуществленных проектах).
- **Отсутствие единого ответственного за данные**, что сказывается на невозможности определить более корректные и актуальные данные.
- **Низкое качество переносимых данных** (дублирование, некорректные форматы данных).
- **Нестабильность целевой системы для переноса данных.**

В случае, если разработка еще не завершена и миграция данных выполнена преждевременно, могут возникнуть ошибки / потеря информации.

- **Несвоевременность организации синхронизации данных.** Актуализация данных должны производиться в новой системе сразу после миграции.
- **Низкая степень документированности процесса миграция.** Данный риск относится не только к невозможности использовать полученный опыт для выполнения дальнейших проектов. В большей степени связан этот риск связан с невозможностью в случае возникновения необходимости продемонстрировать аудиторам/ компетентным контролирующим органам отчеты по датам и масштабам перемещения данных. А значит, потенциально может возникнуть огромное количество проблем, связанных с доказательством достоверности предоставленной отчетности. Особенно это актуально, если данные для отчетности выгружались в период параллельной работы старой и новой систем, при наличии различий в объемах данных, хранившихся в них в конкретный момент.

Таким образом, миграция данных – длительная, кропотливая, требующая большого внимания работа, связанная с высоким риском потери данных. В девяти из десяти случаев основной сложностью, с которыми сталкиваются команды проекта при миграции является возникновение дублирования данных. Среди основных причин появления подобных данных-«двойников»:

1. Различия в написании объектов:

ПРИМЕР

– *Организационно-правовые формы / формы собственности («ЗАО» Вымпел, оао «Вымпел», ВЫМПЕЛ). Это приводит к путанице: идет ли речь о разных предприятиях либо об одной и той же компании?*
– *Использование латиницы и сокращений: Ivanov Ivan, Иванов И.*

2. Несовпадение информации из-за устаревания данных.

ПРИМЕР

В одной системе в качестве адреса ООО «Вымпел» указывается г.Воронеж, а в другой – г. Москва. Разница может быть обусловлена переездом компании либо же разной трактовкой понятия «адрес» сотрудниками, вводившими информацию в систему: в первом случае – юридический адрес компании, во втором – фактический.

3. Несовпадение форматов данных

ПРИМЕР

– *Неиспользование выпадающих списков для полей с ограниченным числом значений: г. Балашиха в одной системе и «Московская область» в другой.*
– *Использование текстовых полей для написания цифр (например, в случае миграции Excel-таблиц): 84951234567 добав 1111.*
– *Стиль написания цифр: +7 (495) 123-45-67, 84951234567.*

Во всех этих случаях необходима тщательная выверка всех данных, проводящаяся частично в автоматизированном режиме (например, для замены «+7» в телефонных номерах на «8», удалении лишних пробелов / символов), частично в ручном режиме пользователями.

2.4.4.5. Разработка контрольного примера

Контрольный пример предназначен для проверки корректности работы алгоритма, правильности заполнения и реакции форм на ввод определенных форматов данных, корректности обработки исключений. В конечном итоге (в идеальном случае) результаты реализации контрольного примера должны целиком совпадать с ожидаемым результатом. В противном случае все полученные расхождения (в случае их наличия) должны быть объяснимы (например, большей точностью вычислений новой системы).

Контрольный пример составляется по каждому функциональному модулю и используется для выявления и устранения недостатков / замечаний силами сначала проектной команды, а затем – с привлечением ключевых пользователей. Затем этот же пример может использоваться в процессе обучения рядовых пользователей. Подготовка контрольного примера должна проводиться очень тщательно, по разработанному плану тестирования (регламент и сценарий тес-

тирования, о котором уже говорилось при описании инсталляции системы, а также с обязательным наличием начальных данных для примера).

Контрольный пример должен следовать определенной логике, покрывать ситуации типичного использования приложения и наиболее важные для пользователя аспекты работы с системой, проверять распределение функций по модулям, быть удобным и понятным для тестирования. Должна проверяться также реализация изначально собранных требований пользователей. Среди других характеристик тестового сценария:

- Проверка наиболее «слабых» мест системы для повышения вероятности обнаружения ошибок.
- Выполнение необходимого минимума действий для проверки (неизбыточность).
- Явно выявляет ошибки / недоработки.
- Понятность и логичность описания примера для пользователей.

Тестовые сценарии также могут относиться к конкретной проверяемой области: конфигурации системы, юзабилити / удобство использования, безопасности, взаимодействия / интеграции компонентов. Однако может создаваться и общий пример, с единым описанием и дальнейшей детализацией в зависимости от специфики проверяемой области / задач тестирования.

Ко времени старта тестовой эксплуатации таким образом должны быть подготовлены такие элементы модели тестирования, как:

- **Набор исходных данных** / условий тестирования / планируемых результатов.
- **Методика испытаний**: технический документ, описывающий настройку системы для проведения теста и принцип оценки итоговых результатов.
- **Контрольный пример** / сценарий тестирования.
- **Тестовый скрипт** (опционально: в случае использования автоматизированных инструментов тестирования).

Формальным основанием для старта тестовой эксплуатации могут служить:

- Протокол совещания группы внедрения о передаче модуля системы в тестовую эксплуатацию.
- Регламент тестирования контрольного примера.
- Перечень замечаний ключевых пользователей, собранных при предварительном тестировании контрольного примера (по каждому модулю системы), с описанием решений по их устранению, отраженный в Протоколе совещания группы внедрения.

2.4.4.6. Тестовая эксплуатация

Между первоначальной настройкой системы на стенде по сформированным и утвержденным требованиям и ее полномасштабным вводом в экс-

плуатацию требуется один из самых сложных с точки зрения согласования интересов всех сторон этап тестовой эксплуатации. Как правило, прием системы в эксплуатацию и обучение пользователей проводятся исключительно после подтверждения своего согласия с «итоговой версией». Разумеется, для проведения опытной эксплуатации составляется отдельный регламент / методика – и затем, все замечания, получаемые в ходе работы с системой среди выбранной группы ключевых пользователей, заносятся в отдельный журнал запросов на изменение с расстановкой приоритетов исполнения.

Среди наиболее больших блоков / классов проводимых тестов можно выделить следующие:

- **Компонентное / модульное тестирование** (component / unit testing). Его основная цель состоит в идентификации ошибок реализации модулей в ходе проверки работоспособности (например, при различных поступающих на вход данных). Проверяются отдельные объекты, функции, классы, модули.
- **Интеграционное тестирование.** Данный тип тестирования проводится для всей системы в целом. Его основная задача – проверка корректности передачи информации и взаимодействия между всеми модулями / компонентами системы и слоями архитектуры (в частности, прикладным программным обеспечением, инфраструктурными сервисами / операционной системой). Особенно данный тип тестирования актуален для клиент-серверных / распределенных систем.
- **Системное тестирование (в т.ч. функциональное тестирование).** Определение степени соответствия созданной ИС исходным требованиям (функциональным и нефункциональным). Проводится проверка корректности настройки внешних интерфейсов системы. Оцениваются различные характеристики надежности, производительности, отказоустойчивости системы, в дальнейшем также проверяемые в ходе нагрузочного тестирования.

В качестве дополнительных проверок могут создаваться тестовые окружения для ИС при эмуляции работы отдельных компонентов систем / внешнего окружения / активностей пользователей. В частности, при **эмуляции** возможно смоделировать работу того или иного компонента системы (модуля, сервиса), которые еще не были созданы, для более раннего проведения тестирования.

Тестирование в зависимости от специфики системы / предметной области может проводиться в ручном / автоматизированном или полуавтоматизированном режиме.

- **В ручном режиме** специалисты по тестированию самостоятельно проходят весь контрольный пример / сценарий тестирования. Этот способ применяется для крайне динамической разработки в условиях постоянных изменений. По результатам составляется протокол

тестирования, в который заносятся все замечания, комментарии и предложения.

- **Автоматизированное тестирование** значительно упрощает и ускоряет процесс тестирования, предоставляет возможность провести большее число проверок, в том числе до и после внесения доработок в тестовую среду.
- **Полуавтоматизированное тестирование** выполняется по заданному сценарию под постоянным контролем тестировщика. Таким образом создаются преимущества автоматизированного тестирования, но с возможностью в случае необходимости изменить сценарий его выполнения и организовать дополнительные проверки / проверки при новых условиях.

Тестирование, как наиболее важный элемент обеспечения / контроля качества может и должно быть организовано по специальной методике / в соответствии со стандартами. Среди наиболее часто применяющихся стандартов:

- **ГОСТ 12119** ИТ. Пакеты программ. Требования к качеству и тестирование.
- **ГОСТ 12207** ИТ. Процессы жизненного цикла программных средств.
- **ГОСТ 14764** ИТ. Сопровождение программных средств.
- **ГОСТ 15288** ИТ. Системная инженерия. Процессы жизненного цикла систем.
- **ГОСТ 15504** ИТ. Оценка процессов.
- **ГОСТ 28195** ИТ. Оценка качества программных средств. Общие положения.
- **ГОСТ 9126** ИТ. Оценка программной продукции. Характеристики качества и руководства по их применению.
- **ГОСТ серии 19** Единая система программной документации (ЕСПД).
- **ГОСТ серии 34** Разработка автоматизированной системы управления.
- **IEEE 829** Стандарт документирования тестирования программного обеспечения и систем.

Отдельно можно отметить унифицированный процесс RUP, который по своей сути исходит из идеи постоянного регулярного тестирования на каждой итерации.

Срок тестовой эксплуатации, как правило, не превышает одного месяца и не меняется при отсутствии или несвоевременном предоставлении замечаний. Среди выполняемых для организации тестовой эксплуатации мероприятий (все они проводятся по каждому функциональному модулю в отдельности):

- Подготовка перечня замечаний ключевых пользователей, собранных в процессе окончательного тестирования контрольного примера (по каждому модулю системы), с описанием решений по их устранению, отраженный в Протоколе совещания Группы внедрения.

– Предварительное тестирование командой внедрения контрольного примера (по отдельным модулям!) с занесением замечаний в журнал тестирования и последующим их устранением (в случае необходимости с дополнительными настройками / доработками).
– Финальное тестирование с привлечением ключевых пользователей (здесь помимо функциональности крайне важно также проверить степень удобства и интуитивности интерфейса системы для пользователей).
– Разработка общего руководства по работе с системой и инструкций пользователей по модулям системы (включая назначение и условия работы с модулем, описание операций по подготовке и самой работе).
– Обучение пользователей базовым принципам работы с системой на контрольном примере (в отдельности для каждого модуля системы). В процессе обучения пользователь знакомится с интерфейсом системы, получает информацию по основным принципам работы и основным возможностям системы. В случае наличия разработанных методических руководств пользователю дополнительно предоставляется необходимая документация. Подробнее обучение пользователей рассматривается в соответствующем разделе пособия.

2.4.4.7. Доработка по результатам тестирования

Основная цель данной активности – устранить замечания, возникшие в результате тестирования на контрольном примере и, при необходимости, осуществить дополнительные настройки и доработки модулей системы. Как уже было сказано, на данном этапе также важно разделять критичные для успешной работы с системой и дополнительные, nice-to-have требования пользователей-экспертов, которые можно будет включить в следующие итерации работы и версии системы.

Что касается порядка организации данных типов проверки, то для первых сборок проводится поверхностное **smoke-тестирование**, направленное на критическую функциональность. В данном случае организуется короткий цикл тестов для подтверждения того, что после сборки кода (вне зависимости, нового или исправленного) приложение будет успешно выполнять свои основные функции. Код проверяется на предмет наличия быстронаходимых критических дефектов, которые необходимо срочно исправить, отправив на доработку. В случае их отсутствия и успешного прохождения тестирования и приложение передается для проведения полного цикла тестирования.

Так как при smoke-тестировании в максимально короткие сроки покрывается максимально широкий объем функциональных возможностей, то существует и другой вид тестов, дополняющих первый (функции проверяются скорее «вглубь», максимально тщательно тестируя каждый аспект).

Это обусловлено тем, что при проведении любого рода изменений необходимо обеспечивать их корректность и отсутствие негативного влияния на работоспособность и эффективность приложения.

С этой целью проводится **регрессионное тестирование (regression testing)**, один из наиболее известных видов которого получил название **sanity-тестирования.** Это тестирование определенных участков кода, проводимое по результатам внесенных изменений. Его основная задача – удостовериться в том, что внесенные изменения не затронули работоспособность системы и не послужили причиной новых ошибок. К примеру, проверяется, действительно ли исправленные ранее ошибки исправлены, не приводят ли сделанные изменения к нейтрализации исправлений старых ошибок (**регрессия багов, bug regression**) и есть ли новые ошибки.

В результате должны быть:

- Проведены тестирование, выявление и устранение замечаний по контрольным примерам на каждый модуль системы (замечания обработаны и, либо сделаны соответствующие изменения в модуле, либо они отклонены с указанием причин).
- Проведены дополнительные настройки и доработки модуля системы (в срок до 4 недель с момента передачи модуля системы в тестовую эксплуатацию и получения перечня окончательных доработок и настроек модуля системы).
- Составлен протокол выполненных работ по устранению задокументированных замечаний ключевых пользователей Заказчика по результатам тестирования контрольного примера по каждому модулю системы.
- Ключевые пользователи обучены базовым основам работы с модулем системы на данных контрольного примера и готовы к его практической эксплуатации на своих рабочих местах, а также развертыванию на основном рабочем сервере и проведению опытно-промышленной эксплуатации (ОПЭ).
- Сформирован протокол выполненных работ по обучению ключевых пользователей Заказчика базовым основам работы с модулем системы на данных контрольного примера.
- Подготовлен перечень окончательных доработок и настроек каждого модуля системы.

2.4.4.8. Прием результатов испытаний

Результаты тестовых испытаний заносятся в единый отчет для рассмотрения заказчиком. При этом организуется **приемочное тестирование (acceptance testing)**, представляющее собой комплексную проверку компонентов системы на предмет их работы с плановыми параметрами производи-

тельности / быстродействия и выполнением всех функциональных требований, в том числе на различных конфигурациях.

Прием результатов испытаний проводится при выполнении следующих условий:

- Специалисты группы внедрения со стороны заказчика владеют функциональностью модуля системы и готовы к его развертыванию на рабочем сервере и опытно-промышленной эксплуатации.
- Все замечания, возникшие в ходе первоначального тестирования контрольных примеров, задокументированы и устранены.
- Ключевые пользователи готовы к работе с модулем на своих рабочих местах.
- Все замечания ключевых пользователей, собранные во время окончательного тестирования контрольного примера, обработаны и, либо сделаны соответствующие изменения в модуле, либо замечания отклонены с указанием причин отклонения.
- Заказчиком утверждены итоговые документы.

2.4.5. Развертывание и внедрение

Этап развертывания системы является одним из наиболее важных с точки зрения распределения ответственности за работы между заказчиком и подрядчиком. Исполнитель выполняет основную задачу – инсталляцию модуля на рабочий сервер с перенесением на него итоговую конфигурацию с тестового сервера. Однако при верном последовательном обучении и вовлечении специалистов заказчика в процесс создания / внедрения системы остальные операции могут быть осуществлены ими:

- Предшествующее работам обеспечение охвата всех автоматизируемых рабочих мест пользователей функционирующей ЛВС.
- Настройка рабочего сервера.
- Определение перечня / количества рабочих мест, которые необходимо развернуть для ОПЭ.
- Развертывание рабочих мест пользователей с определением прав доступа.
- Подготовка наиболее актуальной информации по каждому модулю для ввода в систему (включая подготовку, дополнение и выверку справочников, которые могли потерять актуальность за время между тестовой эксплуатацией и развертыванием системы).
- Организация поддержания данных системы в актуальном состоянии (в том числе при интеграции с другими системами).

Целью данного этапа является подготовка модуля системы к опытно-промышленной эксплуатации – подготовка конечных пользователей, ввод начальных данных, подготовка приказов по предприятию заказчика о передаче модуля системы в опытно-промышленную эксплуатацию.

Среди основных работ этапа:

- Обеспечить безусловное выполнение условий готовности модулей системы к сдаче в опытно-промышленную эксплуатацию (закрепленных в отдельном документе).
- Ввести и выверить начальные данные (например, начальные остатки) по каждому модулю для ввода информации в систему.
- Отработать на рабочих местах пользователей модуля системы практические действия пользователей в параллельном режиме с имеющимися приложениями.
- Разработать, согласовать с подрядчиком и утвердить регламент взаимодействия подразделений заказчика, участвующих в опытно-промышленной эксплуатации модуля системы.
- Осуществить интеграцию модуля с другими модулями или внешними системами, внедренными ранее.
- Подготовить и издать приказ по предприятию заказчика о передаче модуля системы в опытно-промышленную эксплуатацию, который должен содержать:
  - наименование модуля системы, проходящего опытно-промышленную эксплуатацию;
  - наименование компании-исполнителя;
  - сроки проведения опытно-промышленной эксплуатации;
  - список должностных лиц со стороны заказчика и исполнителя, ответственных за проведение опытно-промышленной эксплуатации модуля;
  - перечень подразделений предприятия заказчика, участвующих в проведении опытно-промышленной эксплуатации;
  - порядок и сроки перевода персонала заказчика на работу в условиях функционирования модуля системы.

К моменту ОПЭ уже должны быть проведены следующие работы:

- Модули системы успешно перенесены и функционируют на рабочем сервере и автоматизированных рабочих местах пользователей (с разделением прав доступа).
- Подготовлены, дополнены и введены недостающие справочники по каждому модулю системы, проведена выверка введенных в систему недостающих справочников.
- Заказчиком утверждены итоговые документы.

2.4.5.1. Закупка и настройка требуемой ИТ-инфраструктуры

Важно учитывать, что всех предприятий (в особенности – энергетической, производственной, финансовой, телекоммуникационной и прочих отраслей)

чрезвычайно высока ценность хранимой и обрабатываемой информации, а также стоимость времени простоя бизнеса. Поэтому к техническому обеспечению ИТ-систем компании должны предъявляться жесткие требования:

- **по производительности** – обеспечение приемлемого времени реакции с точки зрения пользователя, использующего данную систему;
- **по надёжности и доступности** – пользователи должны продолжать работу при выходе из строя единичных устройств и система не должна простаивать и не должно быть потерь данных;
- **по катастрофоустойчивости** – простой системы должен быть минимальным, но все данные должны быть сохранены. Допускается некоторое ухудшение производительности и увеличение времени реакции системы;
- **по безопасности** – система работает с данными составляющими коммерческую тайну и должна обеспечивать должный уровень защиты вплоть до аттестации и сертификации решений и оборудования;
- **по масштабируемости** – система должна обладать способностью приспосабливаться к увеличивающейся нагрузке со сравнительно малыми архитектурными изменениями или совсем без них.

Все данные требования к характеристикам решения в процессе подготовки ИТ-инфраструктуры детализируются на гораздо более подробном уровне, и ниже рассмотрим пример описания конкретных действий по обеспечению масштабируемости решения.

ПРИМЕР

Основной объем обработки проводится при помощи рабочих процессов сервера приложений (диалог, фон, обновление, очередь или подкачка данных). Эти процессы запускаются на инсталляциях сервера приложений (первичная и дополнительная инсталляции дополнительного сервера приложений, за исключением процессов постановки в очередь, которые запускаются на инсталляции центральных сервисов). Вся обработка, связанная с базой данных, контролируется инсталляцией базы данных. Это значит, что критически важно обеспечить высокую масштабируемость инсталляций сервера приложений и базы данных. Масштабируемость этих инсталляций обеспечивается следующим образом:

- *Сервер приложений масштабируется по горизонтали путем добавления дополнительных серверных хост-узлов, в которых выполняются дополнительные инсталляции приложений. Отправка запросов на инсталляции сервера приложений выполняется при помощи сервиса отправки сообщений, работающего на инсталляции центральных сервисов.*
- *Инсталляция базы данных масштабируется по вертикали путем адаптации необходимых аппаратных ресурсов (увеличения мощности процессора, расширение памяти и так далее).*

К ИТ-инфраструктуре в таком случае можно отнести следующие компоненты:

– Вычислительная инфраструктура:

– **Базовые инфраструктурные сервисы.**

Описание и процесс организации таких инфраструктурных сервисов, как IP телефония, электронная почта, резервное копирование, антивирусная защита и антиспам, служба каталогов AD, служба удаленных рабочих столов, сетевая печать и пр.

Данные приложения автоматизируют вспомогательные задачи конечных пользователей, поэтому к этому классу в первую очередь относятся приложения, связанные с пользовательскими коммуникациями.

– **Серверное и пользовательское оборудование и системы хранения данных (СХД).**

Определение перечня пользовательского и серверного оборудования и систем хранения данных.

– **Серверные площадки и центры обработки данных (ЦОД).**

Планирование типовых центров обработки данных и серверных комнат, вплоть до составления некой усредненной модели серверной комнаты/ЦОД.

– **Системное ПО.**

К этой категории отнесем описание операционных систем и программных платформ, как для серверов, так и для оборудования, доступного конечному пользователю. Помимо этого, в этом разделе рассматриваются службы виртуализации.

– Сетевая инфраструктура:

– **Инфраструктура локальных вычислительных сетей (ЛВС).**

Данный компонент должен представлять собой типовое решение по организации внутренней локальной вычислительной сети. Именно этот элемент является ключевым для обеспечения связи между оборудованием в офисах, филиалах, ЦОДах и других типах помещений. Важно, что в случае производственной или любой другой компании со значительной долей технологических систем проводится физическое / логическое разделение локальной сети на два сегмента, каждый из которых эксплуатирует только собственное сетевое оборудование. Таким образом, в корпоративной сети работают все бизнес-пользователи, а в технологической располагаются основные сервисы (оборудование, пользователи), поддерживающие технологическую составляющую / процессы компании.

– **Корпоративные сети передачи данных (КСПД).**

– **Телефония, ВКС, подключение к сети Интернет.**

– Инженерная инфраструктура:

Устройства бесперебойного питания, электропитания, охлаждения, кабельная инфраструктура.

Также возможно упомянуть, что среди современных тенденций построения ИТ-инфраструктуры – использование нескольких уровней абстракции для вычислительных комплексов, сетей передачи данных, с целью повыше-

ния гибкости и обеспечения регулирования потребления ресурсов (что снижает в конечном итоге суммарную стоимость оборудования).

Среди непосредственных активностей этапа построения необходимой для системы ИТ-инфраструктуры:

1. **Закупка оборудования** – данный шаг обычно значительно увеличивает время подготовки технической платформы, так как закупка может быть связана с проведением конкурсов, заключением договоров, ожиданием поставки оборудования. Все эти шаги могут приводить к потерям времени от нескольких недель до нескольких месяцев.
2. **Подготовка инфраструктуры** – расчет возможности установки оборудования в серверное помещение, выделение и конфигурирование сетевых портов, планирование мощностей компонентов инженерной инфраструктуры серверных и пр.
3. **Аппаратное конфигурирование** – получение оборудования, его конфигурирование и установка (в т.ч. тестовый сервер, рабочие станции, вспомогательное оборудование)
4. **Установка системного программного обеспечения –** система управления базами данных, операционная система, генераторы отчетов (либо установка базового программного обеспечения, реализующего слой виртуализации)
5. **Конфигурирование пользовательских рабочих мест –** рекомендуется определить несколько стандартных конфигураций пользовательских рабочих мест, которые бы позволяли обеспечивать приемлемое качество работы пользователей. При наличии ограниченного количества конфигураций появится возможность не только упростить поддержку клиентских рабочих мест, но и, например, настроить систему мониторинга для проверок ключевых параметров работы приложений на всех конфигурациях пользовательских рабочих мест, используемых в компании.

В результате команда внедрения со стороны подрядчика получает развернутую функционирующую стендовую локальную вычислительную сеть, удовлетворяющую требованиям ТЗ, и программно-аппаратная платформа уже готова к последующей инсталляции системы.

2.4.5.2. Ввод начальных остатков

Данный этап, актуальный для систем финансового управления, является логическим продолжением проведенной ранее миграции данных (со вводом справочников и первичной информации). Формирование начальных остатков проводится уже после настройки параметров приложения и на их основе составляется начальный баланс, с помощью которого проверяется, насколько корректно были введены данные. В зависимости от системы при

вводе и выверке данных может проводиться либо редактирование сальдо счета / субсчета, либо формирование фиктивных проводок. Делаются записи по дополнительным регистрам. При этом проводится тщательный контроль реакции программы на все операции, проверки сбалансированности сальдо, проверки каждого счета (в том числе аналитических).

2.4.5.3. Обучение пользователей

Для минимизации сопротивления, которое может встречаться со стороны специалистов организации при внедрении «очередной» системы обучение проводится в три основных этапа:

1. Обучение ключевых пользователей на этапе внедрения перед организацией тестирования.
2. Обучение ключевых пользователей перед инсталляцией на стендовый сервер.
3. Обучение рядовых пользователей перед стартом опытно-промышленной эксплуатации.

Как можно заметить, подобное разделение этапов обусловлено тремя основными вехами: в первую очередь, необходимо объяснение принципов работы уже настроенной системы перед первым тестированием с ключевыми пользователями.

Следующая веха – инсталляция на стендовый сервер – важна как промежуточный этап, на котором снова будут привлекаться ключевые пользователи, только уже для описания работы / интерфейсов в гораздо более детализированном виде для идентификации и устранения всех остающихся критических замечаний перед ОПЭ. Разумеется, к этому моменту все специалисты ИТ, которые будут заниматься поддержкой системы, уже должны в деталях знать внутреннюю архитектуру системы и быть способными проводить ее доработку / формировать требования на доработку в случае необходимости.

И наконец, наиболее масштабное обучение организуется уже для всех остальных сотрудников, которые будут работать с системой. В данном случае система описывается не с точки зрения внутренней архитектуры, а с пользовательской перспективы принципов ввода, обработки и получения данных.

Вне зависимости от специфики системы / предметной области / этапа обучения для его организации необходимы следующие активности:

Шаг 1. В первую очередь, **выделяются «ключевые пользователи» –** один-два специалиста от каждого структурного подразделения (деление также может быть не с организационной, а с логической точки зрения). Их поддержка будет крайне важна на следующем этапе при проведении тренингов для остальных пользователей, так как помимо предоставления комментариев по пользованию системой и о необходимости доработок, их опыт

может помочь остальным пользователям разобраться с работой в системе и не обращаться каждый раз к команде сопровождения со стороны подрядчика.

Важно, что в зависимости от масштаба и специфики системы, эти специалисты могут работать не только с пользовательскими аспектами системы, но и самостоятельно разворачивать функциональные компоненты / модули, конфигурировать их по требованиям пользователей. Именно поэтому их обучение, аттестация и вовлечение в работу крайне важно гораздо раньше ОПЭ, ведь основное число сотрудников увидят систему и ознакомятся с принципами работы в ней лишь во время опытно-промышленной и промышленной эксплуатации, когда команда исполнителя и ключевые пользователи будут проводить обучение.

Подобное прохождение учебных курсов по основам работы с системой для сформированной группы ключевых пользователей (по функциональным направлениям) организуется специалистами исполнителя (включая аттестацию). В качестве формы организации обучения может быть выбрана очная (преподавателем от команды исполнителя в офисе компании заказчика или подрядчика), либо дистанционная (когда личное присутствие преподавателя не целесообразно либо невозможно). Все популярнее становится проведение вебинаров и телеконференций, позволяющих экономить временные и денежные ресурсы.

Шаг 2. Обучение рядовых пользователей.

К этому моменту необходимо, чтобы достаточное количество специалистов группы внедрения от заказчика были способны самостоятельно разворачивать функциональные компоненты системы / модуля на рабочих местах пользователей, проводить обучение конечных пользователей, оказывать консультации по стандартной конфигурации системы, а также сопровождать ее в оперативном режиме.

Принципы проведения тренингов в зависимости от масштабов компании, проекта / системы и пожеланий заказчика могут значительно различаться, однако на практике чаще всего они проводятся несколько раз, не более чем для 10-15 человек единовременно, при этом группы обычно формируются под каждую отдельную роль пользователя. Основные темы включают общий обзор задач системы, рассмотрение ее интерфейса, основных возможностей, затем специфической для конкретной роли пользователя функциональности и секции вопрос / ответ. Иногда после этого проводится настройка необходимых параметров (например, интеграции с учетными записями Outlook, настройка синхронизации) уже на местах пользователей, однако необходимость этого зависит от конкретной системы.

Шаг 3. Повышение квалификации / переобучение.

Следует учитывать также необходимость последующего периодического обучения с целью повышения квалификации сотрудников, в котором условно возможно выделить три основные категории:

- Плановое обучение после изменений в системе (при добавлении дополнительных возможностей, внедрения нового модуля, и т.п.).
- Плановое повышение квалификации: предназначено для совершения знаний работников, уже обладающих определенными профессиональными навыками работы с системой.
- Обучение по запросу – относится к сфере поддержки пользователей на этапе эксплуатации системы и проводится в случае, если пользователь в своей работе столкнулся с конкретными проблемами / нуждается в проведении методического занятия.

Отметим, что для работы с некоторыми классами систем (как правило, если их деятельность связана с управлением объектами, представляющих в случае выхода из-под контроля риск для жизни / здоровья людей) просто проведения обучения недостаточно, и организуются дополнительные тестирования знаний, успешное прохождение которых дает допуск к пользованию системой при предоставлении персонального сертификата. Примером могут являться системы, автоматизирующие управление безопасностью железнодорожного движения и авиаперевозок, запуск космических объектов, химическое производство и другие виды деятельности.

И разумеется, при планировании обучения следует учитывать такие аспекты, как: опыт сотрудников, предполагаемые и видимые знания, а также навыки руководства и управления, организационные способности.

2.4.5.4. Развертывание системы на рабочих местах

Следующим шагом организуется развертывание модуля для дальнейшей его опытно-промышленной эксплуатации. Успешность данного этапа целиком зависит от проведенной ранее работы. Так, если уже был реализован пилотный проект в другом подразделении / филиале или в рамках другой функциональной области, этот опыт поможет компании воспользоваться рекомендациями по развертыванию системы и выполнить оптимальные настройки. Также, инсталляция проводится еще до тестирования, а значит результаты используются исключительно в целях совершенствования системы в так называемой «песочнице» (sandbox).

«Песочница» (sandbox) – закрытое для доступа извне виртуальное пространство, в котором можно работать с ПО без изменения системных файлов, в связи с чем они используются для запуска кода, еще не прошедшего тестирование.

Настройки «песочницы» и ее элементы дублируют основные элементы/возможности реальной системы для первоначальной отладки и проверки кода в виртуальной среде перед его переносом в продукт. Это помогает избежать значительного числа рисков, сбоя настроек сложной конфигурации системы.

Набор ресурсов системы, выделяемый для «песочницы», жестко ограничен в части доступа к объему памяти, к сети, обмену информацией с основной системой. Это обусловлено в первую очередь нестабильностью работы и значительной нагрузкой на систему, почему «песочницы» часто и используются исключительно для проверки отдельных критических частей кода.

Сделанные в среде sandbox настройки переносятся при полноценном развертывании модуля системы после тестирования перед ОПЭ. Модули системы устанавливаются на стендовый сервер с проведением основных необходимых настроек под подлежащие автоматизации бизнес-процессы предприятия заказчика, по результатам чего подрядчиком подготавливаются описание оптимальных настроек сервера баз данных и рекомендации по развертыванию рабочих мест, которое и проводится следующим шагом.

Перед развертыванием системы необходимо формирование командой внедрения следующей информации:

- Выделение основных блоков функциональности, их запусков, взаимозависимостей.
- Выделение основных блоков работ, требуемых для запуска функциональности системы, и их взаимозависимости.
- Расстановка контрольных точек для верхнеуровневого мониторинга процесса.
- Информация по исполнителям работ (зонам ответственности).

При переходе в продуктив должны быть осуществлены такие активности, как:

- Проверка необходимых предпосылок для перехода в продуктивную систему.
- Описание мероприятий по переходу в продуктивную систему.
- Описание критериев, на основании которых будет приниматься решение о начале работы пользователей.
- Описание порядка действий в случае возникновения ошибок / сбоев / нарушения работы системы.

При развертывании системы на рабочих местах проводятся следующие активности:

- установка / подключение технических средств;
- установка / конфигурация ПО;
- развертывание баз данных, служебных программ;
- доработка интерфейса / процессов / кастомизация профилей пользователей / настройка отчетов;
- установка прав доступа;
- пуско-наладочные работы и окончательная отладка конфигурации.

Потенциальные сложности могут возникнуть также, если несколько смежных подпроектов (например, в части разных модулей системы) реализуются несколькими подрядчиками. Тогда следует уделять пристальное

внимание управлению интеграцией всех проектов на протяжении времени их реализации и при подготовке к ОПЭ в особенности.

2.4.5.5. Основные виды тестирования

Прежде, чем приступать к промышленной эксплуатации системы важно проверить ее работоспособность во всех предусмотренных спецификацией режимах, а также «экстремальных условиях». Как минимум, если планируется одновременный распределенный ввод данных 100 пользователями, то необходимо проверить правильность обработки системой одновременной активности в системе именно не менее сотни пользователей. Подобная идея лежит в основе **нагрузочного тестирования.**

Нагрузочное тестирование необходимо для предсказания поведения системы в реальных и экстремальных условиях, выявления ошибок, отслеживать производительность и доступность при изменении различных параметров работы с системой. Среди его других задач:

- Оценка производительности и работоспособности приложения на этапах:
  - Разработки.
  - Опытно-промышленной эксплуатации.
  - Сопровождения и эксплуатации (выпуск новых релизов, патчей).
- Оптимизация приложений.
- Подбор оптимальной программно-аппаратной платформы и конфигурации сервера.

Помимо собственно нагрузочного тестирования чаще всего выделяют еще несколько ключевых категорий тестирования:

- **Тестирование производительности** (load testing).

Моделирование ожидаемой интенсивности использования путем распределенной работы большого числа пользователей с различными модулями системы.

Среди возможных активностей подобного типа тестирования:

- Определение времени выполнения операций с заданной интенсивностью (на разных нагрузках).
- Определение максимально достижимой производительности системы.

- **Стрессовое тестирование** (stress testing).

Определение стабильности системы при интенсивности работы, превышающей плановые / стандартные значения. Стресс-тестирование по своей сути проверяет, возвращается ли (и насколько быстро) система после запредельной нагрузки к нормальному режиму работы, тестируется способность системы к регенерации.

Особенно стресс-тестирование важно для компаний, в которых стоимость отказа системы в экстремальных условиях может быть очень велика (а стандартного тестирования разработчиками недостаточно для эмуляции тех условий, в которых произойдет отказ системы).

Среди основных активностей подобного тестирования:

- Оценка динамики производительности при нестандартных ситуациях: аварийное изменение серверной конфигурации либо резкое повышение числа пользователей и выполняемых одновременно операций.
- Определение условий и скорости возвращения системы к нормальному режиму работы после окончания стрессового тестирования.

– **Тестирование надежности** (reliability testing).

- Определение длительности бесперебойной и безошибочной работы системы.
- Оценка стабильности системы при многочасовом тестировании со стандартным средним уровнем нагрузки для определения:
  - Утечек памяти.
  - Некорректных конфигурационных настроек.
  - Случаев перезагрузки сервера и других требующих устранения проблем.

– **Конфигурационное тестирование** (configuration testing).

Оценка степени влияния на производительность изменений в конфигурации / различной балансировки нагрузок. Конфигурационное тестирование относится как к серверному уровню (совместимость с окружением), так и к клиентскому уровню (например, кросс-платформенное или кросс-браузерное тестирование).

- Оценка обработки ошибок и исключительных ситуаций, а также «узких мест» отдельных модулей и компонент системы при непропорциональных нагрузках.
- Определение оптимальной конфигурации оборудования, которая бы обеспечивала требуемые характеристики производительности и отклика системы.
- Проверка определенных компонент системы на предмет совместимости с указанным в спецификации оборудованием, операционными системами и программными продуктами внешних поставщиков.

С точки зрения последовательности действий для проведения данного вида тестирования:

1. Создается матрица покрытия с описанием всех возможных конфигураций системы.
2. Приоритезация конфигураций.

3. И, наконец, в ходе тестирования в соответствии с имеющимися приоритетами организуется проверка всех основных конфигураций.

– **Объемное тестирование** (volume testing).

Тестирование программного обеспечения системы на предмет стабильности обработки определенного объема данных.

- Измерение динамики выполнения операций при увеличении объемов данных в базе данных, постепенном росте числа запросов пользователей.
- Определение максимального числа пользователей, которые могут работать с приложением без снижения производительности.

Отсутствие или ошибки проведения подобных видов тестирования значительно увеличивают риски и могут даже приводить к ситуациям, когда после внедрения срок эксплуатации систем не превышал даже года в связи с невозможностью конфигурации системы справляться с необходимыми задачами и нагрузками. Именно для нивелирования подобных рисков необходимо своевременно организовывать тестирование еще на этапе внедрения (хотя его проведение возможно и часто проводится и для уже эксплуатируемых систем).

2.4.5.6. Опытно-промышленная эксплуатация

Опытно-промышленная эксплуатация представляет собой тестирование в полной функциональности и полной нагрузке для определенного количества пользователей. Ее основной целью является апробирование работы пользователей в системе в реальных производственных условиях. Это означает, что если по спецификациям системы предполагается, что она будет обрабатывать 500,000 записей в день, необходимо проверить корректность обработки именно этого количества записей. При этом важные задачи:

– тестирование модуля системы в условиях, максимально приближенных к реальным условиям промышленной эксплуатации (в том числе, при необходимости, в интеграции с другими модулями или внешними системами) – нагрузочное тестирование, рассмотренное ранее;
– проведение множественных расчетов по фактической производственной информации с применением соответствующих программно-аппаратных средств, предусмотренных техническим проектом (проектным решением);
– достижение наиболее полного охвата бизнес-процессов автоматизируемых подразделений предприятия.

При условии соблюдения ранее определенных требований к предыдущим этапам и их успешном завершении в первую очередь с этого момента необходимо провести:

- Ввод наиболее актуальной информации по каждому модулю для ввода в систему (подготовка которой проведена на предыдущем этапе).
- Отработку действий пользователя в параллельном режиме с уже имеющимися приложениями (это проводится непосредственно на рабочих местах пользователей и включает дополнительную, но необходимую нагрузку для пользователей, которые в короткий период времени должны адаптироваться к работе в другой системе (а также помочь адаптировать ее для наиболее оптимальной и эффективной работы).
- Интеграцию внедряемого модуля с уже существующими модулями и / или внешними системами, которые были внедрены ранее.
- Разработку и согласование регламента взаимодействия внутренних подразделений предприятия-заказчика в рамках ОПЭ и дальнейшей промышленной эксплуатации (помимо схемы и порядка передачи функций с обязательным определением зон ответственности за актуальность и корректность данных).

ПРИМЕР

Несмотря на то, что официально «владельцем» системы будет являться одно подразделение (или даже одно ответственное лицо), ввод и редактирование информации будут осуществляться распределенно и часто одновременно различными категориями пользователей. Соответственно, существует практика определения зон ответственности за различными подразделениями, которые вне зависимости от источника ввода будут обладать исключительными правами на удаление и изменение информации. Этот механизм может быть организован многими способами, самым «простым» из которых является четкое определение алгоритма взаимодействия подразделений в рамках работы с системой (сотрудники будут сами знать, какую информацию в какой момент они будут вводить и обрабатывать). Во многом подобный механизм исходит из принципов «презумпции невиновности», предполагая что каждый специалист достаточно компетентен и отвечает за каждое совершенное им действие (как в целом в работе, так и в системе).

Другим, менее распространенным способом управления взаимодействием подразделений (и главное – контроля данных) является распределение прав доступа таким образом, что у наиболее важной информации будет только один источник ввода / проверки. И хотя другие пользователи могут изменять поле (например, вносить данные об инвестициях и договорах), в системе эта информация будет «висеть» в режиме ожидания до момента ее одобрения/авторизации действия ответственным за поле пользователем. Однако несмотря на значительную гарантию достоверности данных, подобная схема крайне неэффективна с точки зрения использования ресурсов, особенно в условиях большого числа и масштаба операций.

Основная часть (как по объему, так и по трудозатратам) проводимых в ОПЭ мероприятий предусматривает сопровождение заказчиком системы в ходе ОПЭ на рабочих местах пользователей с формированием перечня заме-

чаний ключевых пользователей (при их «переводе» из бизнес-терминологии в более техническую область) и устранением этих возникших замечаний.

ПРИМЕР

Замечание пользователей:
В список партнеров нельзя добавить университеты (нет пункта в списке).
Требование к реализации подрядчиком:
Создать в типе организаций «Прочие» наследующий его свойства подтип «Университет» с дополнительными полями «Ректор», «Факультет», «Кафедра». Настроить общедоступный отчет по всем университетам.

Устранением замечаний и внесением изменений в систему на этом этапе (длящимся, как и в случае тестовой эксплуатации, не менее одного месяца) занимается команда внедрения со стороны исполнителя, которая помимо простого соответствия системы всем требованиям и бизнес-процессам должна удостовериться в устойчивости работы системе при нагрузке, максимально приближенной к реальной промышленной нагрузке этапа эксплуатации (с исключением риска отказа системы в момент её пика). Система также должна безошибочно функционировать в параллельном режиме и при интеграции с внешними приложениями в течение всего срока ОПЭ для того, чтобы перейти к полноценной промышленной эксплуатации. При наличии подобных legacy-приложений об успешном внедрении можно говорить, если результаты обработки данных в них и в новой системе совпадают и / или различия объяснимы и некритичны.

ПРИМЕР

Новая система предоставляет более точный результат либо использует другой формат выдачи данных.

Для окончания ОПЭ необходимо подтверждение того, что модуль системы функционирует в параллельном режиме с существующими внешними приложениями (при их наличии) в течение срока опытно-промышленной эксплуатации. Вторым условием является подтверждение, что любые отклонения (в случае их обнаружения) признаны некритичными и объяснимыми.

2.4.5.7. Приемо-сдаточные испытания

Именно передача системы из ОПЭ в промышленную эксплуатацию (в частности, при проведении приемо-сдаточных испытаний) и является целью комплексного проекта по автоматизации деятельности компании. Данный тип приемо-сдаточных испытаний получил также название **интеграционного тестирования**. Путем его проведения обеспечивается комплексная проверка реализованных проектных решений системы.

Цели интеграционного тестирования следующие:

– проверка корректности функционирования бизнес-процессов и функций;

- уточнение настроек бизнес-процессов в системе;
- уточнение и доработка настроек пользовательского интерфейса;
- проверка функций начальной загрузки в систему основных и переменных данных;
- проверка полноты переносов настроек посредством транспортной системы;
- проверка работоспособности интерфейсов с внешними системами;
- проверка корректности проектной и эксплуатационной документации;
- определение правильности функционирования системы на реальном объеме данных в реальном времени.

Тестирование проводится на продуктивном наборе данных для проверки работоспособности всей системы и правильности настройки интерфейсов взаимодействия с внешними системами.

Как уже было сказано, именно этот этап является переходным между этапом ОПЭ и реальной эксплуатацией. С организационной точки зрения перед началом промышленной эксплуатации необходимо удостовериться в наличии следующего набора документов:

- Рабочий журнал опытно-промышленной эксплуатации с перечнем замечаний ключевых пользователей и результатами их устранения (заполняемый совместно командой внедрения и согласовываемый затем представителями).
- Протокол об окончании ОПЭ и готовности к промышленной эксплуатации совместно с Актом приемки-передачи работ.
- Приказ по предприятию заказчика о приеме системы в промышленную эксплуатацию (с составом функций, описанием программно-аппаратной платформы, списком ответственных за работу модуля и списком подразделений-пользователей, а также порядком и срокам перехода персонала на работу в системе, включая аспект разработки и принятия новых форм документов в случае необходимости).

После этого непосредственно использование системы уже не имеет признаков проектной деятельности и ограничений по срокам / содержанию / стоимости. Однако такие последующие активности, как сопровождение эксплуатации, модернизация и утилизация системы будут по своей сути являться проектами и могут также выполняться не только за счет внутренних ресурсов компании, но и (что случается гораздо чаще) с привлечением внешних подрядчиков.

2.4.6. Эксплуатация

Этап эксплуатации системы является наиболее ожидаемым для бизнес-заказчиков, которые только сейчас начинают получать ожидаемый возврат инвестиций и видеть реальный результат внедрения. Однако это и самый

опасный этап, так как полученный результат может (и скорее всего, не будет) не соответствовать первоначальным ожиданиям и представлениям как о функциональности, так и даже о внешнем виде / интерфейсе системы. Технические специалисты и подрядчик фокусируются на решении задач бесперебойной работы, обслуживания баз данных, а в случае необходимости – на донастройке системы (что означает одновременное наступление стадии модернизации, продолжающееся параллельно с эксплуатацией). Но бизнес-заказчик и спонсор системы, как и все участники процесса сбора требований часто имеют собственное представление о целевом состоянии системы, именно поэтому критически важно с самого начала вовлекать в проект специалистов, переводящих пожелания бизнеса в плоскость разработчиков (данная роль в английской терминологии получила название IT/Business Relationship Manager).

Стадия сопровождения, приводящая к изменениям к системе в процессе эксплуатации (по окончании приемки работ) охватывает несколько аспектов:

- Устранение замечаний, не затрагивающее изменение ТЗ.
- Обновления (по сути – новые версии системы), выпускаемые при накоплении критического объема доработок.
- Увеличение производительности системы.

Важно отметить, что на сегодняшний день основной задачей компании-подрядчика является не просто настройка системы, но передача накопленных и базирующихся на опыте множества проекта знаний и методик работы с внедренным программным решением. По окончании проекта специалисты компании-заказчика должны быть способны самостоятельно осуществлять полноценную поддержку и развитие системы. В первую очередь, это относится к развитию уже созданных модулей, совершенствованию функциональности, реализации новых задач для новых категорий пользователей, формирование аналитических отчетов и панелей мониторинга. Разумеется, часто заказчик принимает решение о продолжении сотрудничества с подрядчиком, внедрявшим систему, в части ее сопровождения. Это относится и к базовым функциям (построению дополнительных типов отчеты или изменению параметров настроек), и к технологической и методической поддержке.

2.4.7. Сопровождение эксплуатации

2.4.7.1. Авторский надзор

В условиях промышленной эксплуатации в течение определенного договором времени после сдачи модуля системы со стороны исполнителя-подрядчика проводится наблюдение за его функционированием и по мере необходимости оказание помощи специалистам заказчика по устранению

замечаний. В свою очередь, уже обученные ответственные за систему сотрудники заказчика в тесном взаимодействии с ключевыми пользователями формируют перечень замечаний, осуществляют технологические обновления модуля / системы и ведут документирование эталонных версий модулей и их программной документации. При этом предприятие-подрядчик включается в работу только при необходимости значительных доработок либо если таковое определено заключенным контрактом на поддержку. Это объясняется тем, что ключевая необходимая для успешной и бесперебойной работы системы функциональность и программно-аппаратная база уже была сформирована на предыдущих этапах и специалисты заказчика уже в состоянии осуществлять стандартную техническую поддержку на своем предприятии самостоятельно.

Именно в это время устраняются основные возникающие замечания / ошибки / неточности первоначальных расчетов нагрузки на систему. Как правило, длительность этапа авторского надзора составляет не менее года, и в отличие от фактора масштаба и сложности проекта, отсутствие или несвоевременное представление замечаний не влияют на продолжительность этапа авторского надзора за промышленной эксплуатацией модуля.

Для этой и последующих стадий сопровождения могут применяться специальные **метрики оценки работ** со стороны предприятия-подрядчика, отвечающего за сопровождение. Четыре основные метрики (актуальные для всего жизненного цикла) были выделены Институтом Программной Инженерии университета Карнеги-Меллон (SEI CMU): **размер, усилия, расписание и качество**.

Однако далее на базе классификации эксплуатационных характеристик программного обеспечения они могут дополняться и другими параметрами:

- **Анализируемость**: оценка не предусмотренных изначально усилий или ресурсов, необходимых для диагностики недостатков или причин сбоев, а также для идентификации тех фрагментов программной системы, которые должны быть модифицированы.
- **Изменяемость**: оценка усилий, необходимых для проведения заданных модификаций.
- **Стабильность**: оценка случаев непредусмотренного поведения системы, включая ситуации, обнаруженные в процессе тестирования.
- **Тестируемость**: оценка усилий персонала сопровождения и пользователей по тестированию модифицированного программного обеспечения.

2.4.7.2. Техническая поддержка

Техническая поддержка системы начинается после приема в промышленную эксплуатацию и может продолжаться до снятия с эксплуатации и

утилизации системы. Как правило, варианты осуществляемой поддержки варьируются по процентам от стоимости приобретенного ПО, в зависимости от набора оказываемых услуг, а объем и периодичность выполнения работ на данном этапе определяются отдельным договором.

В качестве поддержки могут предоставляться следующие услуги в различных комбинациях в зависимости от прописанных в договоре условий:

- Консультирование по «горячей линии» (телефон, e-mail, skype) по определенному в договоре графику.
- Консультирование по вопросам программно-аппаратной платформы системы.
- Создание новых учетных записей пользователей системы.
- Настройка форм отчетов и панелей мониторинга для определенных групп пользователей заказчика.
- Диагностика и устранение неисправностей (с учетом гарантии на сами программные решения).
- Уведомление о выпуске обновлений и их загрузка через Интернет.
- Замена ключей электронной защиты, переустановка ПО в случае необходимости (например, при замене компьютерной базы заказчика).
- Миграция данных и настройка интеграции с новыми системами предприятия заказчика.
- Помощь в формализации требований к новым программным решениям предприятия заказчика в части интеграции с внедренной системой – либо в постановке и формализации задач (на уровне ТЗ) по включению новых бизнес-процессов в систему.

2.4.7.3. Постгарантийное сопровождение

Постгарантийное сопровождение предполагает заказываемые у производителя работы, не предусмотренные изначальным контрактом на автоматизацию процессов / системную интеграцию. Сопровождение программного обеспечения определяется стандартом IEEE Standard for Software Maintenance (IEEE 1219) как «модификация программного продукта после передачи в эксплуатацию для устранения сбоев, улучшения показателей производительности и / или других характеристик (атрибутов) продукта, или адаптации продукта для использования в модифицированном окружении». Функционирование программного продукта таким образом поддерживается на протяжении всего периода его эксплуатации.

В рамках заключаемого на этом этапе договора исполнителем оказывается помощь по устранению замечаний и привлечение в случае необходимости специалистов для выполнения доработок (например, в части новой функциональности или отчетов). Однако в силу того, что если в течение всего жизненного цикла исполнителем проводились мероприятия по обуче-

нию заказчика и вовлечению его в проект, в результате на фазе постгарантийного сопровождения специалисты заказчика будут способны:

- Сопровождать модуль системы в ходе промышленной эксплуатации на участках.
- Формировать перечень замечаний ключевых пользователей.
- Выполнять работы по устранению возникших замечаний (оказывать помощь по устранению замечаний).
- Осуществлять своевременные обновления модуля системы в компании исполнителя (прежде всего – технологические).
- Хранить и вести эталонные версии модуля системы и программной документации (Постгарантийное сопровождение старых версий может продолжаться и после выпуска новых версий программных продуктов, однако чаще поддерживаются исключительно новые версии).

Со стороны компании-исполнителя в данном случае осуществляются постановка задачи, анализ проблем в предметной области, планирование и реализация необходимых доработок и прочие активности.

В условиях договоров на постгарантийное сопровождение, как правило, прописываются также условия, касающиеся: возможности приобретения дополнительных клиентских лицензий; возможности поддержки устаревших версий программных продуктов, на которых основана система; оплаты командировочных расходов компании-подрядчика при необходимости выезда в филиалы других регионов и государств; скидках и особых условиях при продлении контракта.

Для рассмотрения работ по сопровождению системы будем использовать руководство SWEBOK в области знаний «Поддержка ПО» (Software maintenance) как содержащее достаточно полный набор материалов на эту тему.

SWEBOK приводит ряд процессов (работ, практик), которые для деятельности по сопровождению являются уникальными:

– **Передача** (Transition).

Внутренние коммуникации разработчиков и группы поддержки для грамотной координации процесса передачи программного решения на сопровождение;

– **Принятие / отклонение запросов на модификацию** (Modification Request Acceptance / Rejection).

Рассмотрение таких характеристик, как: объем и/или сложность требуемых изменений, необходимые для их реализации активности. Решения по принятию или отклонению запросов на модификацию могут также основываться на анализе приоритетности, оценке обоснованности, отсутствии ресурсов (в том числе, отсутствия возможности привлечения разработчиков к решению задач по модификации, при реальном наличии такой потребности), внесению в план к реализации следующих релизов.

– **Средства извещения персонала сопровождения и отслеживания статуса запросов на модификацию и отчетов об ошибках** (Modification Request and Problem Report Help Desk).

Поддержка конечных пользователей, в т.ч. анализ приоритетности и стоимости модификаций, связанных с поступившим запросом или сообщенной проблемой.

– **Анализ влияния** (Impact Analysis).

Анализ возможных последствий изменений, вносимых в существующую систему, при идентификации всех связанных с ней систем / программных продуктов, на которых эти изменения могут потенциально отразиться.

– **Поддержка программного обеспечения** (Software Support).

Консультирование пользователей, проводимое по их информационным запросам (request for information), например, в отношении бизнес-правил, проверки, содержания данных и получаемых сообщений о проблемах (ошибках, сбоях, непредусмотренном поведении, непонимании принципов функционирования системы).

– **Контракты и обязательства.**

Закрепленные в договорной форме обязательства по определенным параметрам оказания услуг сопровождения (в т.ч. классическое соглашение об уровне предоставляемого сервиса, SLA).

Обновление и релизы

Неотъемлемой частью сопровождения является выпуск обновлений и релизов программ / конфигураций. В них, в частности, могут быть реализо-

ваны новые правила обмена данными, настроены новые формы регламентированной отчетности, усовершенствованы интерфейсы, адаптированы функциональные возможности и произведены другие доработки.

Подобного вида изменения вносятся в программное обеспечение либо подрядчиков в рамках постгарантийного сопровождения при накоплении критического числа замечаний, либо вендором программного продукта в рамках планового выпуска его очередной версии. Это может проводиться к рамках подготовки нового технического дистрибутива, в рамках выпуска локализованной под конкретную страну / регион конфигурации, в рамках отраслевого решения, в рамках обновления определенного модуля и прочее.

Если речь идет о **выпуске версий при накоплении критического числа замечаний**, то сопровождение подобного рода, как правило, складывается из следующих этапов:

1. Формирование заявок для специалистов, работающих в рамках сопровождения.
2. Планирование работ (выявление проблем / требований, объема и содержания задач).
3. Оказание услуг (согласование / уточнение требований, реализация работ, консультации, обучение, доработки)
4. Сдача / приемка работ.

Говоря о **плановых обновлениях версий программного обеспечения вендорами**, приведем пример компании 1С.

ПРИМЕР

В 2012 году 1С выпустила новую конфигурацию «Бухгалтерия предприятия» (редакция 3.0) и в пресс-релизе были отмечены следующие внесенные изменения:

1. Повышение удобства работы.

Новые пиктограммы в интерфейсе пользователя, кастомизация панелей действий и навигации. Появление внутренних ссылок для каждого из объектов информационной базы.

2. Развитие функциональности учета заработной платы.

Начисление заработной платы, НДФЛ и страховых взносов стало проводиться одним документом автоматически при начислении заработной платы. Разделы «Зарплата» и «Кадры» объединены в один раздел, за счет чего все кадровые документы стали доступны на карточке сотрудника.

3. Развитие прав доступа.

Добавлена функция доступа к информационной базе в режиме просмотра без прав на внесение изменений.

4. Работа через Интернет по модели SaaS.

Для работы, начиная с версии 3.0 более не требуется установка платформы / информационных баз на компьютере пользователя. Вместо этого доступен запуск программы через веб-браузер с сайта компании-поставщика «облачного» сервиса.

5. Повышение удобства работы при выполнении длительных операций.

Возможность выполнения операций в фоновом режиме.

6. Новые средства защиты персональных данных.
Изменения внесены в связи с требованиями Федерального закона.

Таким образом, важность уделения должного внимания выпускаемым обновлениям несомненна и активности по их реализации также несут в себе все признаки проектной деятельности.

Увеличение производительности системы

Во время проектирования / внедрения системы для определения ее возможностей определяется, как быстро приложение должно работать, какой объем памяти / загрузку процессора использовать, и какие другие характеристики системы можно считать приемлемыми.

В течение срока эксплуатации / сопровождения системы ее окружение меняется, и прежние параметры производительности становятся уже недостаточными для успешной работы. В частности, это может относиться к таким аспектам, как:

– **Время отклика.**

Время, необходимое серверу для получения ответа на произведенный запрос.

– **Пропускная способность.**

Число запросов, которые могут быть обработаны за единицу времени. Часто при измерении данного параметра определяется число запросов/ логических транзакций в секунду.

– **Использование ресурсов.**

Потребление приложением серверных / сетевых ресурсов (ЦП, память, дисковое пространство).

– **Рабочая загрузка.**

Общее количество пользователей (либо только активных пользователей), объем данных и транзакций.

Соответственно, рано или поздно владельцы системы и бизнес в целом сталкиваются с необходимостью совершенствования этих характеристик. При этом иногда подобный подход является проактивным, когда принимаемые меры направлены на оптимизацию, снижение операционных расходов. Однако чаще эти действия являются уже реактивными и проводятся, когда стоимость владения системой и затраты на нее уже становятся неприемлемыми и поднимается вопрос о целесообразности ее эксплуатации.

Проводимое в рамках сопровождения системное улучшение параметров получило название **инженерии производительности (performance engineering)**. Изначально данная область разрабатывалась в рамках направления системотехники как включающая в себя единый набор ролей, знаний, практик, инструментов и результатов каждого этапа ЖЦ системы. Основная цель состояла в том, чтобы гарантировать, что создаваемое программно-аппаратное решение соответствует нефункциональным требованиям к его производительности.

В настоящее время область применения инженерии (и ре-инжиниринга) производительности в значительной части сфокусирована на проектах сопровождения / модернизации системы, когда требования уже проверены временем в ходе эксплуатации и могут быть определены более конкретно.

Среди некоторых из задач инженерии производительности:

- Повышение окупаемости бизнеса через обеспечение своевременной обработки необходимых объемов транзакций.
- Своевременное выявление потенциальных проблем масштабирования / производительности и их устранение.
- Снижение стоимости поддержки ПО и внеплановых затрат через своевременное выявление проблем производительности.
- Предотвращение задержек, вызванных проблемами с производительностью, при развертывании систем.

При планировании производительности необходимо вначале понимать, какими ресурсы системы доступны в текущий момент. Для этого оцениваются:

- **Сетевое обеспечение** (в том числе пропускная способность).
- **Аппаратное обеспечение** (характеристики серверов, рабочих станций).
- **Зависимости ресурсов** (число доступных соединений баз данных, веб-сервисов).
- **Совместно используемые ресурсы** (в т.ч. принципы распределения ресурсов между разными элементами системы).
- **Проектные ресурсы** (финансовые и человеческие ресурсы) для поддержки системы и осуществления проекта по повышению ее производительности.

При этом в процессе могут быть задействованы администратор системы, системный архитектор, разработчики, тестировщики и другие члены проектной команды. Именно они производят моделирование/ прототипирование желаемого результата, проектирование новых параметров системы, доработку программного кода, мониторинг, создание надежных тестов производительности, различного рода тестирование и настройку системы до необходимых значений.

ПРИМЕР

Вполне конкретное требование к определению планового значения отклика системы на запрос может выглядеть следующим образом:

«Для сценария использования А отклик системы на корректный запрос пользователя не должен превышать:

- *5 секунд для нагрузки в 250 активно использующих систему пользователей (200 онлайн пользователей в 95 % случаев);*
- *10 секунд для пиковой нагрузки в 500 активных пользователей (400 онлайн пользователей в 90 % случаев)».*

В общем случае для подобных требований определяется типичный бизнес-день разбитый на часы, для формирования представления о том, какова

динамика нагрузки на систему в течение дня. В результате можно получить конкретные предложения по улучшению (например, «распараллелить работу модуля и запустив ее на четырех процессорах вместо одного») и конкретные результаты «повысить таким образом эффективность загрузки ресурсов модуля, получив прирост производительности в 5 раз».

2.4.8. Модернизация

2.4.8.1. Стратегии управления legacy-системами

В течение периода эксплуатации системы происходят неизбежные изменения как вне организации (в том числе, изменение рынка, контрагентов), так и внутри организации (к примеру, смещение бизнес-приоритетов в другую индустрию, на другой рынок и пр.). Растет объем обрабатываемой информации, объем файлов, снижается пропускная способность, оборудование выходит из строя и появляется новое – все эти факторы неизбежно влияют на актуальность самой системы и решаемых ей задач. В то время, когда системы уже не способны с должной степенью эффективности решать задачи бизнеса, говорят об «унаследованных» (legacy) системах.

Legacy systems (унаследованные системы) – системы, более не соответствующие текущим потребностям бизнеса, но по-прежнему эксплуатирующиеся компаниями. Как правило, в их основе лежат устаревшие со временем технологии, платформы и не представляется возможным далее и совершенствовать и видоизменять для соответствия требованиям безопасности, новому аппаратному обеспечению, обновленным операционным системам.

Чаще всего модификация подобных приложений требует значительных затрат времени и финансовых средств, в то время как их замена может требовать даже реинжиниринга бизнес-процессов организации. В таком случае требуется доработка системы, для чего необходимо заново пройти все фазы жизненного цикла – от сбора / формирования требований до сопровождения и эксплуатации. Соответственно, определяются сроки, ресурсы и содержание, результат подобной доработки будет являться уникальным, и потому можно говорить об отдельном проекте по модернизации. Этим проектом может заниматься как команда внедрения, так и абсолютно другие компании-подрядчики (особенно если речь идет о модернизации системы, созданной несколько лет назад, когда компания-подрядчик может уже даже прекратить свое существование).

Модернизация системы подобного рода позволяет продлить срок его эксплуатации, снизить совокупную стоимость владения (TCO), расширить функциональные возможности, совершенствовать программно-аппаратную

платформу минимальными доступными средствами БЕЗ остановки процесса эксплуатации системы. В ходе модернизации устраняются многие возникавшие при эксплуатации проблемы. Однако необходимость реализации определенных доработок, число которых уже достигло критического значения, является лишь одной из возможных причин старта модернизации системы.

Другим важным фактором / триггером является аудит информационных систем, проводимый самостоятельно организацией или при привлечении внешних экспертов / консультантов. Результаты аудита могут сочетаться с реинжинирингом бизнес-процессов компании, разработкой новой целевой прикладной архитектуры или же обновлением ИТ-стратегии в целом. Принятию решения о модернизации в обязательном порядке должно предшествовать определение сроков, стоимости и состава работ по доработке, так как возможна даже ситуация, когда списание системы и внедрение новой в конечном итоге окажется более оптимальным вариантом, чем изменение существующей. Ведь в целом задачи модификации, будучи на первый взгляд простыми (хотя и трудоемкими), представляются компаниям, как правило, несвоевременными либо неперспективными. Поколения сотрудников (как заказчика, так и компании-подрядчика) меняются, работавшие при внедрении специалисты уходят, а в случае некорректного документирования системы при ее создании результаты попыток совершенствования ПО могут быть достаточно плачевными.

Таким образом, среди внешних и внутренних факторов, которые могут служить своеобразными индикаторами необходимости модификации систем, можно перечислить следующие (на примере экономической ИС):

- Изменения в системе документооборота, формате / структуре формируемых документов.
- Изменения в перечне задач, принадлежащих к основной функциональности системы, или в методах их решения.
- Мнения и оценки пользователей о работе с системой, качестве обработки данных и результирующей информации.
- Информация специализированного ПО (например, в составе СУБД и операционных систем) по статистике работы системы, ее быстродействию и отказоустойчивости.
- Характеристики организации и внешней среды (выпуск новых изделий, появление новых технологических ограничений).
- Изменение законодательства и требований регуляторов (например, в части стандартов учета).
- Уход из компании и недостаток на рынке специалистов со знаниями, необходимыми для поддержки конкретной legacy-системы;
- Слишком высокая стоимость поддержки системы / совокупная стоимость владения.
- Отсутствие технических возможностей (высокая сложность) интеграции с новыми, внедряемыми на предприятии системами.

- Рыночная конкуренция и бизнес-приоритеты (например, интернет-магазин, который фокусируется на удовлетворении потребностей пользователей, может принять решение о модернизации пользовательского интерфейса и системы принятия / обработки заказов клиентов).
- Осуществляемые процессы слияний и поглощений M&A (так, в случае объединения информационных систем с другой компанией требуется полный пересмотр существующего ландшафта систем и частичная / полная модернизация).

Сами активности по модификации и совершенствованию системы могут представлять собой: трансформацию исходного кода вплоть до смены среды или языка программирования, совершенствование архитектуры приложения и принципов взаимодействия его модулей на основе анализа качества кода. Чаще всего выделяются следующие варианты проведения модернизации:

- **Миграция (migration).**

Переход на более новую версию платформы / операционной системы, базы данных, языка программирования. Обеспечивается достаточно эффективный с точки зрения соотношения стоимости и качества способ трансформации устаревающих систем.

ПРИМЕР

Среди основных категорий миграции:

*– Миграция **платформ.***

Миграция с IBM Mainframe COBOL/CICS/DB2/VSAM на Intel-платформу Windows Micro Focus COBOL (с БД Oracle).

*– Миграция **языков программирования.***

Конвертация кода на C#

*– Миграция **пользовательского интерфейса.***

Переход от ПК-версии к веб-интерфейсу

*– Миграция **аппаратного обеспечения.***

Переход с DEC VAX-обеспечения на Intel PC сервера и рабочие станции (что в свою очередь, требует перевода ПО с VAX OpenVMS на MS Windows.

*– Миграция **баз данных.***

Переход с IDMS на Oracle.

- **Реинжиниринг (re-engineering).**

Чаще всего применяется при адаптации сервисно-ориентированной архитектуры SOA, при построении прежних приложений на основе новых технологий / платформ, с прежней или же расширенной функциональностью.

- **Рефакторинг кода.**

Преобразования / реорганизация кода. Рефакторинг может быть обоснован необходимостью реализации новой функции, которая не соответствует текущему архитектурному решению. Однако в особенности подобные активности необходимы, когда логика программы слишком сложна для понимания и требует структуризации и совершенствования в целом.

ПРИМЕР

Дублирование кода, слишком длинный список параметров метода или код самого метода, «мертвый код», лишние переменные, несгруппированные данные и пр.
Для устранения этих и других типов несовершенств используются дополнительно методы инкапсуляции, делегирования, полиморфизма, выделение классов / интерфейсов, замена типа кода подклассами и пр.

Следует отметить, что несмотря на то, что рефакторинг относится к одному из вариантов модернизации, он должен проводиться на протяжении всего цикла разработки (к примеру, именно такая логика лежит в основе концепции экстремального программирования и ряда других методологий).

– **Смена хостинга (re-hosting).**

Чаще всего проводится в качестве промежуточного шага при предстоящей замене аппаратного обеспечения (например, на UNIX/Wintel платформах).

– **Внедрение «коробочного» решения (package implementation).**

Замена устаревшего ПО целиком или частично на целые семейства решений (SAP, Oracle Apps).

2.4.8.2. Виртуализация как стратегия модернизации решений

Виртуализация – предоставление вычислительных ресурсов, абстрагированное от аппаратной реализации и изолирующее вычислительные процессы конкретных физических ресурсов. Таким образом, при виртуализации создаются различные уровни абстракции.

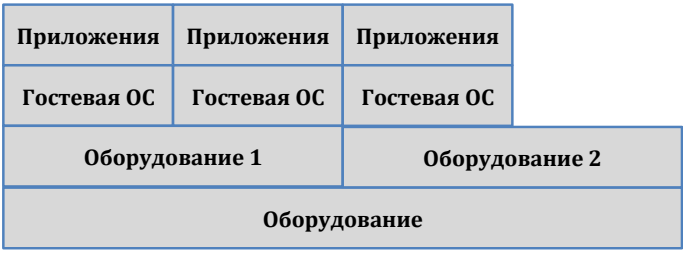
Сама концепция виртуализации появилась еще в 1970х годах, когда IBM виртуализировала интерфейсы своего оборудования. VMS запускается непосредственно на основном оборудовании, позволяющем создавать множество виртуальных машин (VM), каждая из которых может обладать своей собственной операционной системой. Другим вариантом виртуализации в то время была симуляция процессора (выполнение псевдо-кода на виртуальной машине вместо реального оборудования).

В настоящее время виртуализация не теряет своей актуальности и становится все более и более популярным решением в силу следующих причин:

– Происходит оптимизация расходов на ИТ.
– Увеличивается степень контроля над ИТ-инфраструктурой.
– Сокращаются плановые / внеплановые простои.

Среди вычислительных ресурсов, для которых доступна виртуализация, операционные системы и сети передачи данных, программно-аппаратные платформы, сервера / системы хранения данных. Один из наиболее сложных видов виртуализации обеспечивается эмуляцией аппаратных средств (когда виртуальные машины аппаратных средств создаются на хост-системе, чтобы эмулировать интересующее оборудование).

ПРИМЕР



– **Виртуализация приложений** относится ко всем программным технологиям, повышающим мобильность, совместимость и управляемость приложений за счет инкапсулирования от операционной системы, в которой они выполняются.

При этом создается возможность выполнять приложение в виртуальном пространстве, которым возможно управлять как отдельной единицей. Приложения могут выполняться в потоковом режиме на сервере приложений или на клиентском устройстве. Поэтому виртуализация отлично подходит для компаний, обладающих широким спектром приложений с относительно небольшим потреблением ресурсов центра данных, но используемых большим количеством сотрудников.

– **Виртуализация серверов** маскирует от рядовых пользователей детали использования ресурсов (количество и основные данные серверов, процессоров, операционных систем), с которыми ведется работа. Она призвана обеспечить централизованное управление всеми серверными мощностями с максимально возможной гибкостью и высокой степенью масштабирования, а также быстрое развертывание виртуальных машин из готовых шаблонов, оперативное восстановление работоспособности серверов, мониторинг текущей загрузки физических и виртуальных серверов.

Одновременно с этим снижается зависимость эффективности сервиса от старения оборудования, так как появляется возможность осуществлять модернизацию практически без прерывания сервиса, что критично для крупных территориально распределенных компаний.

Крайне ярким примером виртуализации серверов является услуга **VPS-хостинга** (**виртуальный выделенный сервер** – virtual private server, VPS). В подобных случаях в сети могут создаваться несколько независимых друг от друга виртуальных серверов с собственным служб и уникальными характеристиками, которые должны существовать как независимые узлы сети.

– **Виртуализация систем хранения данных** с точки зрения пользователя переносит данные с разрозненных сетевых устройств хранения на единое, управляемое централизованно.

Основной задачей в данном случае является повышение эффективности и надежности, гибкости хранения данных и их мобильности. Дополнительное отделение систем хранения от серверного оборудования уровнем виртуализации позволяет прозрачно перемещать данные между системами хранения и упростить процессы обслуживания и поддержки.

– **Виртуализация представлений** – предоставление вычислительных ресурсов терминальным сервером и выполнение всех операций клиентских приложений на нем. Несколько компьютеров могут быть представлены как один отдельный компьютер (серверный кластер / grid computing). Пользователь при этом лишь видит собственное представление, что значительно снижает сложность администрирования, так как к подобной терминальной сессии всегда можно подключиться, избегая установки дополнительным программных средств. К тому же, несмотря на высокую стоимость подобных мощных серверов, итоговая стоимость проекта может оказаться более низкой, нежели при установке многочисленных локальных рабочих станций.

2.4.8.3. Особенности проектов по модернизации

Большая часть задач по модернизации решения могут проводиться параллельно с эксплуатацией (не дожидаясь ее прекращения), однако это будет требовать повышенного контроля ко всем параметрам и данным системы, вероятность ошибок в которых резко повышается. С другой стороны, успешное применение подобного способа позволяет избежать необходимости мигрировать данные после модернизации и искать «промежуточное» решение, на время модификации ИС заменяющее по функциональности и удобству основную систему.

Прежде, чем начать модернизацию, следует рассмотреть следующие аспекты:

– **Целесообразность модернизации.**

Иногда лучшим выбором может быть продолжение использования системы, фокусируясь на снижении затрат на поддержку. Однако любая legacy-система со временем будет требовать все больших вложений за счет необходимости ее сопровождения сотрудниками с необходимым набором компетенций и опытом работы с подобными системами.

– **Аутсорсинг систем и процессов.**

Все большее число компаний выбирают вариант рехостинга legacy-систем (особенно в условиях наличия многочисленных вариантов планов аутсорсинга, предлагаемых поставщиками). При этом модификации самой системы можно избежать.

– **Целесообразность перехода на другую платформу.**

Для компаний, активно участвующих в процессах слияний и поглощений (M&A), эффективной альтернативой модернизации может стать консолидация и перевод legacy-систем на единую корпоративную платформу (в идеале – с хорошей масштабируемостью). Подобные активности могут значительно повысить затраты в краткосрочном периоде, но долгосрочные выгоды оказываются очень существенными.

– **Минимизация числа изменений.**

В некоторых случаях предприятия могут вносить изменения во внутренний дизайн приложения, сохраняя его функциональные возможности. Так, рефакторинг кода программного решения иногда позволяет модифицировать приложение гораздо более эффективно, нежели масштабная модернизация.

– **Минимальная кастомизация.**

Многие вендоры положительно относятся к проектам по кастомизации / локализации их программных решений, когда это означает более высокую выручку и выход на новые рынки. Так, в большинстве случаев компании приобретают бизнес-платформы и адаптируют исходный код под собственную специфику (либо выпускают патч для более старой версии, например, с учетом изменений налогового законодательства).

Однако в тот самый момент, когда предприятие примет решение о выпуске новых релизов, ей придется столкнуться с множеством кастомизированных доработок и отклонений от исходной версии, что увеличит время- и трудозатраты на модификацию.

– **Длительные проекты внедрения значительно повышают риски.**

В современных условиях, когда жизненный цикл программных решений постоянно сокращается, а новые технологии и стандарты появляются с завидной регулярностью, система, разработанная в сложном длительном проекте внедрения с многочисленными итерациями тестирования и доработок, может оказаться неактуальной еще до окончания проекта.

– **Интеграция систем на основе SOA не является панацеей от всех проблем.**

Все чаще вендоры рекомендуют настройку SOA-интерфейсов legacy-систем, когда «монолитные» приложения разбиваются на определенные компоненты, которые затем по отдельности реализуются по SOA-принципам. Адаптеры инкапсулируют различные технологии для реализации интерфейса между приложениями / для подключения используется единая сервисная шина.

Однако следует отметить, что данный подход никоим образом не снижает сложность приложения / затраты на его сопровождение, а лишь обеспечивает его более эффективное взаимодействие с окружающей средой.

– **Возможности обновлений.**

Компании, заключающие контракт на сопровождение, чаще всего ограничиваются обновлением приложений до новой версии в случае необходи-

мости. Однако возможна ситуация, когда приобретенное приложение больше не выпускается и не поддерживается, и организациям приходится сталкиваться с масштабной заменой программного решения. Поэтому необходимо взвешенно принимать решение об обновлении ПО при выпуске каждой новой версии, либо же с определенными промежутками, только после выпуска самых значительных обновлений.

– **Планирование архитектуры приложений.**

Замена системы может решить только небольшую часть проблем, важно в целом эффективно планировать ландшафт приложений, формировать принципы принятия решения о внедрении/ модернизации/ утилизации систем, согласовывать действия со стратегическими целями ИТ и предприятия в целом.

2.4.9. Утилизация

Не существует системы, которая бы использовалась всегда – а значит, необходимо предусмотреть условия, при которых понадобится отказаться от системы и вывести ее из эксплуатации. Эта деятельность, в свою очередь, также является проектом, ключевая цель которого – снятие системы (или ее отдельных модулей) с эксплуатации, что чаще всего проводится уже после внедрения новой системы. Рассмотрим классификацию необходимых в рамках данного проекта активностей [51].

2.4.9.1. Технические аспекты

– **Аппаратное обеспечение.**

Существующую платформу (компьютерное и серверное оборудование, сетевые и телекоммуникационные устройства, параметры сети и безопасности, например, выделенные IP-адреса и настройки сетевого экрана) возможно использовать для развертывания других систем (например, изменив некоторые параметры / конфигурацию), продать, расторгнуть контракт лизинга.

– **Программное обеспечение.**

Основные программное решение, компиляторы и среды разработки, коннекторы, внутренние библиотеки и прочие относящиеся к системе компоненты ПО возможно сохранить в качестве резервной копии.

– **Данные.**

Для данных (как необходимых для работы с системой, так и созданные в процессе ее эксплуатации) необходимо создать резервные копии как в «родном» формате системы, так и в читаемых на других платформах форматах (pdf, xml, …).

– **Документация.**

Бизнес- и техническая документация системы в качестве детального описания алгоритмов и бизнес-правил, заложенных в систему, также должны быть сохранены в резервных копиях внутреннего формата системы и общедоступных (pdf, docx, …).

– **Замещающие системы.**

К моменту вывода системы из эксплуатации должны быть предусмотрены все средства плавного перехода к эксплуатации другого решения (включая аспекты управления изменениями).

ПРИМЕР

При смене системы «Учет кадров», которая ранее эксплуатировалась в системе «Парус» компании «Парус» на систему 1С решение о снятии с эксплуатации первой системы (утилизации этой подсистемы) принимается только после полной миграции данных, одновременного сопровождения двух систем и полномасштабного ввода соответствующего модуля 1С в эксплуатацию.

– **Зависимые / интегрированные системы.**

Этот пункт предполагает два основных аспекта. Первый – вопрос прекращения передачи и получения данных в / из интегрированных систем и необходимость предусмотреть другие источники данных для них. Особенно это касается внешних приложений и интерфейсов работы с системами контрагентов. Второй – вопрос определения стратегии действий для вспомогательных систем (базы данных, средства отчетности, приложения автоматизации бизнес-процессов и прочие), необходимость в которых со снятием системы с эксплуатации может исчезнуть / потерять актуальность.

2.4.9.2. Организационные аспекты

– **Сотрудники ИТ-департамента компании.**

К моменту прекращения необходимости поддержки эксплуатации и модернизации системы ИТ-специалистами необходимо предусмотреть их занятость в других проектах с учетом наиболее оптимального применения приобретенных ими за время проекта знаний и умений.

– **Подрядчики / специалисты, нанятые по контракту.**

Внешние специалисты, на постоянной или временной основе занятые в проекте внедрения / поддержки системы, могут остаться в компании (продлив контракт или перейдя в штат), либо покинуть ее по окончании контракта. Соответственно, в случае необходимости использовать их компетенции на других проектах необходимо своевременно предусмотреть им замену и организовать передачу критически важных знаний новым сотрудникам.

– **Конечные пользователи.**

Если на смену утилизируемому ПО внедряется альтернативное решение, пользователи перед этим в обязательном порядке должны пройти необходимое обучение.

2.4.9.3. Коммерческие аспекты

– **Контрактные вопросы.**

В процессе эксплуатации системы могли заключаться договоры с подрядчиками по поддержке, сервисами веб-хостинга, дата-центрами, и прочими организациями – а значит, каждый из контрагентов должен быть своевременно оповещен о планируемых изменениях и принято решение о продолжении / прекращении совместной работы.

2.4.9.4. Юридические вопросы

– **Лицензирование.**

Закупленные лицензии на ПО могут быть дезактивированы либо «заморожены» (в этом случае право на владение лицензиями сохраняется, но право пользования ими временно утрачивается до заключения дальнейших соглашений с поставщиком (например, об обмене лицензий на другую версию или другой программный продукт).

– **Отчетность.**

Любые обязательства компании по раскрытию информации, ранее осуществлявшиеся при помощи выводимой из эксплуатации системы, должны быть предусмотрены в заменяющей ИС (включая передачу исторических данных).

ПРИМЕР

В качестве еще одного напоминания, каких ситуаций следует избегать, приведем следующий пример.

В одной из организаций используемое решение (по учету транспорта) было разработано 5-10 лет назад и пара разработчиков давно перешли на работу в другую компанию. Тем временем, по мере роста масштабов бизнеса понадобилось внедрение приложения с расширенной функциональностью, отвечающее новым потребностям бизнеса. Однако миграция данных оказалась невозможной в силу отсутствия открытой для пользователя функции экспорта и недоступности аккаунта администратора системы, как и документации, включая исходный код со спецификациями, единственные версии которых были только у разработчиков системы.

В качестве общих замечаний к жизненному циклу ИС можно добавить, что цена ошибки экспоненциально увеличивается на протяжении этого времени – а значит в течение всего жизненного цикла необходимо применять специальные методы поиска ошибок, почти всегда предусматриваемые различным и методологиями внедрения систем.

ГЛАВА 3

ОСОБЕННОСТИ УПРАВЛЕНИЯ ПРОЕКТАМИ ПО ВНЕДРЕНИЮ КИС

Управление жизненным циклом информационной системы – это управление в условиях постоянно изменяющихся требований со стороны бизнеса, в условиях развивающейся функциональности системы, в условиях постоянного развития новых информационных технологий.

Работа КИС касается деятельности различных служб и должностных лиц предприятия. Однако отношения между ними могут быть более чем непростыми. Информатизация всегда вызывает слухи о сокращении штатов и повышении норм по выработке. Более того, многие специалисты стремятся сохранить свой объем деятельности от постороннего вмешательства, а потому возможны попытки прямого (или скрытого) саботажа нововведений на предприятии. В редких случаях может потребоваться привлечение психологов для работы с коллективом.

К тому же, к сожалению, наблюдается тенденция к «внедрению ради внедрения», когда важен даже не результат, а сам факт работы с ПО определенного масштаба или определенного производителя. В таком случае «заниженная планка» качества / возможностей может устраивать заинтересованные стороны, но важно донести выбранный подход до всех ключевых участников проекта и грамотно подойти к решению вопроса о целях проекта, наличии ограничений, возможностях экономии времени / ресурсов, снижении объема проекта.

Невозможно не отметить, что создание КИС вызывает ряд сложных организационных проблем. Для создание КИС ИТ-службе выгоднее всего привлечь какую-либо стороннюю организацию. Однако ИТ-службе в любом случае придется отвечать за их работу. Попытка обойтись своими силами чревата срывом сроков, а также острым дефицитом ресурсов при обилии текущих задач.

Зачастую профессиональные консультанты, нанятые «на стороне», могут беспристрастно и объективно оценить используемые на предприятии информационные технологии. Консультанты могут разграничить свою сферу деятельности и сферу деятельности ИТ-службы компании-нанимателя. В штате предприятия, тем не менее, должны находиться сотрудники, способные квалифицированно понять и проанализировать действия нанятых консультантов, поэтому предприятию не обойтись без собственных специалистов. ИТ-службы должны контролировать и координировать работы по созданию КИС, начиная с самого первого этапа планирования и обследования –

так как только они являются представителями интересов бизнес-заказчика на предприятии и могут обеспечить непредвзятость расчетов финансовых и временных ресурсов для проекта.

Процедура стратегического планирования должна стать основанием построения КИС на предприятии. Следом за этим необходимо определить основные бизнес-процессы предприятия и выделить информационные структуры, обеспечивающие их. Только в этом случае созданная информационная система сможет стать базой для работы и развития предприятия.

Следует учитывать и изменяющиеся условия бизнеса, иначе КИС может устареть прежде, чем будет завершена. Безусловно, управлять глобальным проектом вроде создания КИС следует с точки зрения финансового менеджмента, однако традиционное планирование становится неадекватным в условиях современного динамичного рынка. Однако полный отказ от планирования также рискует привести к серьезным проблемам. Более того, создавать КИС по частям можно только при наличии единого системного проекта, иначе потери могут превысить ожидаемую полезность. Именно поэтому в настоящее время все больше компаний в сотрудничестве с консалтинговыми компаниями (такими, как Accenture, IBM и BearingPoint) организуют обновление или же создание с нуля стратегии развития ИС. Как правило, подобные проекты достаточно стандартны в плане фаз и включают в себя три основные стадии:

I. Анализ текущего состояния ИТ на предприятии (анализ As-Is).

II. Разработка целевого плана ИТ (формирование целевой модели To-Be).

III. Разработка плана перехода от текущего состояния к целевому, еще называемого «дорожной картой» (roadmap).

ИТ-стратегия – план управления ИТ стратегического уровня, определяющий основные направления развития информационных технологий для оптимального их использования и достижения поставленных компанией бизнес-целей.

Причины, по которым компании принимают решение разрабатывать ИТ-стратегию, могут быть самыми разными.

ПРИМЕР

- *Изменение стратегии бизнеса требует пересмотра ИТ стратегии.*
- *Изменения организационной структуры (новое руководство ИТ-департамента, слияние нескольких компаний и их ИТ-функций).*
- *Предыдущая концепция развития ИТ требует пересмотра в силу своей неэффективности, превышения запланированных затрат, нарушения графика реализации.*
- *Необходимо сформировать механизм взаимодействия бизнеса и ИТ в соответствии со стратегическими задачами компании.*
- *Необходимо обосновать эффективность долгосрочных ИТ-инвестиций.*

Соответственно, определенную роль играют и внутренние факторы (операционный менеджмент, оргструктура), и внешние (например, M&A). В зависимости от целей формирования стратегии определяется, в течение какого периода она будет использоваться (кратко-, средне-, долгосрочный).

В итоге подобных проектов по формированию стратегии могут создаваться следующие результирующие документы:

- Целевые мероприятия и портфель проектов ИТ (в том числе планирование инвестиций в ИТ, паспорта проектов ИТ, балансировка портфеля ИТ, оценка бюджетов проектов ИТ).
- Система управления проектной деятельностью (в том числе организационно-методологические принципы и стандарты управления проектами).
- Каталог услуг ИТ.
- Модель сорсинга ИТ.
- Организационная структура ИТ.
- Модель ИТ-процессов.

А в ходе непосредственно разработки также создаются как целевое видение: ландшафт бизнес-процессов, приложений, данных и ИТ-инфраструктуры, которые могут использоваться при дальнейшем построении систем на предприятии.

Другим значимым типом проектов, способствующих эффективному построению ИС на предприятии, является ИТ-аудит.

ИТ-аудит – системный процесс получения, оценки и предоставлению руководству компании объективных данных о текущем состоянии ИТ-инфраструктуры, информационных систем, бизнес-процессов организации, действиях и событиях, происходящих в сфере ИТ, устанавливающий уровень их соответствия определенным критериям и предоставляющий результаты заказчику.

Его результаты могут использоваться для планирования дальнейшей деятельности компании в сфере ИТ и ее оптимизации. Решения могут затем приниматься с учетом того, насколько эффективно ИТ-подразделения используют имеющиеся ресурсы для решения задач компании; насколько ИТ-деятельность соответствует предъявляющимся требованиям как со стороны организации, так и со стороны внешних контрагентов и регуляторов; насколько надежно организована работа ИТ-систем и обеспечение безопасности данных.

В ходе ИТ-аудита руководству компании предоставляются объективные данные о текущем состоянии ИТ-инфраструктуры, информационных систем, бизнес-процессов организации, действиях и событиях, происходящих в сфере ИТ. В частности, может проводиться аудит систем в процессе разработки, в ходе которого определяется соответствие процессов проектирования и разработки систем утвержденным регламентам и стандартам уровня

организации / отрасли / мировых практик. Оценка эффективности управления проектом, анализа выгод и рисков, сбора требований, анализа запросов на изменение (в том числе при использовании данных CMDB), процедур отката изменений и управления исходным кодом, процессов разработки системы контролей.

В процессе внедрения также важен аудит инфраструктуры ИТ, проводящий оценку размещения, состава и состояния инфраструктурных активов организации, в том числе инженерных систем (бесперебойное электропитание, кондиционирование), сетевого обеспечения (LAN), вычислительных мощностей (программно-аппаратные платформы, хранение данных и резервное копирование).

Внедрение на предприятии информационной системы требует значительных (и не только финансовых) ресурсов. Эффективное внедрение КИС может стать только результатов командной работы, причем в команду должны входить представители разработчика и заказчика. Необходима сторонняя критика людей аналитического и синтетического складов ума. Подробнее подход к формированию проектной команды будет рассмотрен в соответствующем разделе «Управление человеческими ресурсами».

Но не следует также забывать, что далеко не последнюю (и даже главную) роль в процессе внедрения информационной системы играет пользователь. Пусть он не участвует в активностях по внедрению на регулярной основе, но именно для него создается система, именно он будет ее использовать в своей деятельности. А значит, именно на пользователей должен быть направлен фокус в проекте, именно их требования необходимо учитывать для достижения максимального эффекта от внедрения системы и получении ожидаемых результатов при ее эксплуатации.

3.1. Создание ИС в соответствии с методологиями и стандартами

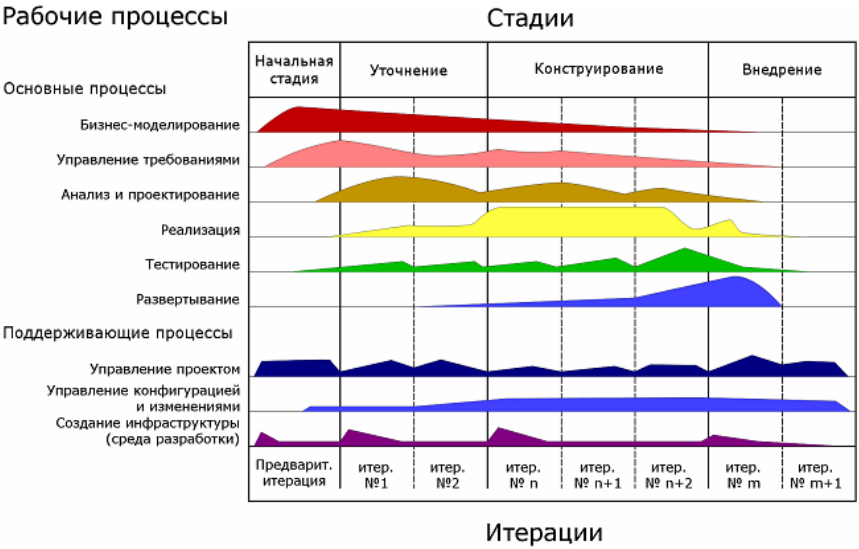
3.1.1. Корпоративные методологии

3.1.1.1. IBM (Rational Unified Process, RUP)

Один из наиболее ориентированных на пользователя подходов, RUP (Rational Unified Process) предполагает итеративную разработку, большое внимание к архитектуре, к потенциальным рискам, и главное – к управлению на основе пользовательских «юзкейсов» (use-cases).

«**Юзкейсы** определяют соглашение между стейкхолдерами системы относительно ее поведения. Юзкейс описывает поведение системы при различных условиях в ответ на запрос со стороны одного из стейкхолдеров» (Коберн).

По сути юзкейсы описывают варианты использования системы и ее действия по достижению поставленных целей. Стейкхолдерами могут быть: заказчики (определяющие конечную цель), менеджеры (ответственные за оперативное управление), аналитики (документирующие требования), разработчики (понимающие реальные потребности заказчика), а также специалисты по тестированию, техническому описанию и разработке интерфейсов пользователя, понимающие поведение системы. Для каждого из них определяются необходимые действия и варианты поведения, но во избежание «тяжеловесности» методологии она предполагает возможность отказаться от определенных работ и задач, в которых нет необходимости для конкретного проекта. Все эти работы и задачи сформированы в рамках четырех основных этапов и сочетания шести основных и трех вспомогательных рабочих процессов.



Важно, что RUP, представляя собой в конечном счете набор решений оптимизации разработки ПО, автоматизируем. В частности, компания Rational Software (в 2003-м году ставшая подразделением IBM), описав возможные роли участников проектов, все возможные виды проектных документов для каждого из них, как и формы распределения ответственности, создала программное решение для поддержки всего процесса разработки. И его использование по сути гарантирует достижение третьей стадии зрелости модели CMM, что для многих процессов многих организаций является достаточным уровнем развития. Данный инструмент, IBM Rational Method Composer, позволяет создавать, конфигурировать и публиковать определенные процессы разработки.

И наконец, RUP содержит шесть основных «лучших практик», описанных в RUP и направленных на повышение продуктивности и снижение вероятности появления ошибок (снижение рисков):

– **Итеративная разработка.**

Несмотря на рациональность единовременного сбора всех требований и создания на их основе системы, значительным преимуществом итеративной разработки является минимизация стоимости каждой из ее фаз за счет снижения вероятности появления ошибок.

– **Управление требованиями.**

– **Компонентный подход.**

Разбиение масштабных проектов на несколько этапов / областей неизбежно, и важно обеспечивать качество каждого из компонентов перед его интеграцией в итоговую систему. При этом RUP предлагает повторное использование кода через объектно-ориентированное программирование как эффективный элемент компонентного подхода.

– **Визуальное моделирование.**

Использование диаграмм, в частности, UML, позволяет обеспечить большую прозрачность процесса как для представления бизнес-заказчику по результатам проектирования, так и для обсуждения внутри проектной команды и последующей реализации на стадии настройки / внедрения решения.

– **Поддержание уровня качества.**

Фокус на тестировании, поддерживать который становится все сложнее по мере развития проекта, всегда должен оставаться одним из главных элементов обеспечения качества.

– **Мониторинг изменений.**

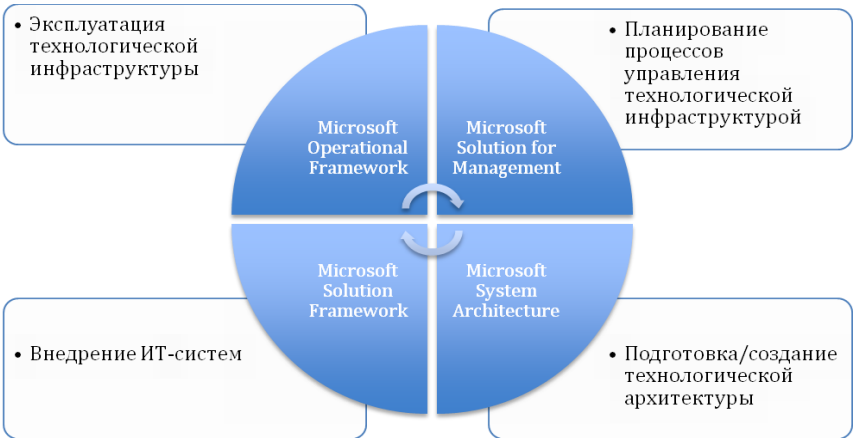
Актуальный в условиях разработки продукта несколькими командами, в особенности географически распределенными, мониторинг сделанных изменений позволяет обеспечить плавную интеграцию и консолидацию отдельных элементов системы.

Таким образом, широкий диапазон возможностей и инструментов, предлагаемый методологией, является общей моделью «конструктора», необходимые элементы которого можно выбрать – и таким образом, значительно снизить стоимость / сроки внедрения. Однако крайне важно принимать во внимание, что подобные решения должны приниматься исключительно профессионалами, уже имеющими значительный опыт подобных проектов для манипулирования со степенью формализма разработки и грамотного применения итерационных возможностей.

3.1.1.2. Microsoft (Microsoft Solution Framework, MSF)

Компанией Microsoft, будучи мировым гигантом в сфере разработки программного обеспечения, подготовлены и применяются несколько мето-

дик для покрытия не только ЖЦ информационных систем, но и технологической инфраструктуры, их поддерживающей.



Однако в контексте рассмотрения ЖЦ ПО нас интересует именно методология разработки: как являющийся одним из основных аспект управления и взаимодействия участников процесса, так и другие области знаний (управление рисками, планирование). В целом охватываемые MSF дисциплины описаны в пяти частях (т.н. «белых книгах»), однако интересно, что командами консультантов Microsoft применяется на практике не этот ресурс, а методика MSF for Agile Software Development, являясь прикладным вариантом MSF и разумеется, отражая общий методологический подход к итеративной разработке.

Если обратиться непосредственно к процессу разработки / внедрения, то его характеризуют:

– Итеративность.
– Формирование в качестве результата **ИТ-решения**.

ИТ-решение – скоординированная поставка набора элементов (таких как программно-технические средства, документация, обучение, сопровождение и внешние коммуникации), необходимых для удовлетворения некоторой бизнес потребности конкретного заказчика.

Именно ИТ-решения (а не просто программные продукты, поставляемые в виде дистрибутивов) являются основным продуктом и результатом процесса разработки и внедрения по MSF (несмотря на то, что аналогичная философия лежит в основе многих методологий, все равно термин «программный продукт» является очень распространенным).

– Базируемость на вехах (основных и вспомогательных) и артефактах как результатах их достижения.

ПРИМЕР

На схеме концепции MSF основными вехами являются

– *«Концепция проекта утверждена».*
– *«Разработка завершена».*

В то же время, промежуточными вехами между ними могут быть:

– *«Верификация технологий осуществлена».*
– *«Базовая версия функциональной спецификации создана».*
– *«Базовая версия сводного плана создана».*
  • *«Базовая версия сводного календарного графика проекта создана».*
– *Среды разработки и тестирования развернуты».*

В целом, важным преимуществом MSF является ее практическая направленность и простота. Для MSF очень важно взаимодействие внутри проектной команды, и несмотря на то, что минимальное число ее участников официально в методологии ограничено всего тремя, выделяются шесть основных ролей внутри команды, и связь между участником и ролью имеет тип «многие-ко-многим». Среди шести упомянутых кластеров:

– управление программой (архитектурным решением);
– разработка (программной и технической архитектуры);
– тестирование (планирование, разработка, отчетность);
– управление релизами;
– управление требованиями заказчика (в части интерфейса решения, а также, конечно, обучения и технической поддержки);
– управление продуктом (по сути, управление требованиями бизнес-заказчика и бизнес-приоритетами).

Именно на этих областях концентрируется методология, каждый из кластеров отвечает за достижение определенной цели, однако за итоговый результат отвечает вся команда.

Для более крупных проектов с большими командами внедрения могут дополнительно создаваться группы направлений и функциональные группы, причем MSF даже предлагает таблицу совместимости ролей: какие из

них допустимо, нежелательно или совсем нельзя совмещать. Допустимо совмещать, например:

– Управление продуктом, Тестирование, Управление требованиями заказчика.
– Управление релизами и Тестирование.

Однако возложение ответственности по Управлению проектом и Управлению продуктом на одного человека может привести к конфликту интересов, так как менеджер проекта должен контролировать основные процессы в области стоимости, сроков, взаимодействия команды, рисков и т.д., а не определять приоритеты бизнеса. Есть и роль, которую вообще не рекомендуется совмещать с другими – роль Разработчика, что сделано исходя из предположения, что его активности являются наиболее критичными в проекте, и любое наделение разработчиков дополнительными обязанностями приведет к срыву план-графика проекта.

Данное распределение становится более понятным, если рассмотреть значение каждой роли и основные активности:

Таким образом, в отличие от многих других корпоративных методологий, определенные в MSF этапы / вехи, состав проектной группы, ролевая модель и другие элементы подходят не только для решений Microsoft. А значит, MSF представляет собой более гибкий и универсальный подход для внедрения других систем / программных продуктов.

3.1.1.3. On Target

Методология внедрения решений On Target была разработана компанией Navision для внедрения своих программных продуктов. После приобре-

тения Navision и вхождения ее в состав Microsoft было принято решение доработать On Target, к тому моменту содержавшую шаблоны описаний бизнес-процессов, документации, организационных структур ИТ и квалификационных требований к специалистам.

Рассмотрим основные этапы внедрения в соответствии с On Target:

– Подготовка проекта.

Формирование проектной команды и подготовка документации.

– Анализ.

Формирование функциональных требований к системе по результатам анализа бизнес-процессов. Обучение ключевых пользователей, разработчиков и системных администраторов со стороны компании заказчика.

– Дизайн.

Разработка и согласование технического задания и эскизного проекта. Финализация плана проекта и бюджета.

– Разработка и тестирование.

Написание программного кода, разработка интерфейсов, настройка и тестирование.

– Развертывание.

Установка системы на стендовом сервере и рабочих местах заказчика, настройка прав доступа, миграция данных, подготовка системной документации и инструкций пользователей, их обучение.

– Опытная эксплуатация.

Приемка в ОПЭ, окончательная верификация данных и процедур после миграции, опытная эксплуатация, передача в промышленную эксплуатацию.

В силу того, что к моменту приобретения Navision у Microsoft уже применялись свои проверенные корпоративные методологии MSF и MOF, в дальнейшем On Target была дополнена и к моменту выведения на рынок Microsoft Dynamics превратилась в результате доработок в MS Dynamics Sure Step / Microsoft Business Solutions Partner Methodology.

3.1.1.4. Microsoft Dynamics Sure Step и Microsoft Business Solutions Partner Methodology

Sure Step была разработана для партнерской сети Misrosoft с целью обеспечения единого подхода к внедрению решений класса MS Dynamics (почему и получила название MS Business Solutions Partner Methodology, в частности используемое для программного решения Modeling Tool). Она основывается на лучших практиках проектов внедрения и для соответствия различным сценариям предлагает несколько компонентов:

– **Этапы.**

С предшествующей методологии On Target отличия незначительные: **Подготовка проекта** заменена на **Диагностику,** а **Опытная эксплуатация** на **Начальное сопровождение.** Соответственно, охватывается уже не только внедрение, но и поддержка заказчика по различным каналам коммуникации, периодические обновления, а также планирование новых проектов.

– **Процессы.**

ПРИМЕР

На стадии дизайна схема процессов выглядит следующим образом:

– **Предложения.**

Sure Step предоставляет возможность не обязательно проходить все этапы модели методологии, но и выбирать необходимые в соответствии со спецификой бизнеса и задачами проекта, что и отражается в компоненте Предложения.

ПРИМЕР

Диагностика, подробный анализ; внедрение (полное либо быстрое); оптимизация.

– **Отчетные материалы.**

Методология предоставляет информацию по входным и итоговым документам каждого этапа, а в отношении всего проекта в качестве стандартных конечных результатов она предполагает создание следующих документов:

– оценка инфраструктуры;
– план проекта;

– функциональные требования;
– план и контрольный список реализации.

Отчетные материалы являются прямыми результатами соответствующих процессов этапов.

– **Межэтапные (постоянные) процессы.**

Как уже было сказано, в зависимости от сценария методология внедрения может меняться и для ее поддержки существует группа постоянных процессов в виде взаимосвязанных операций.

– анализ бизнес-процесса;
– конфигурация;
– перенос данных;
– инфраструктура;
– инсталляция;
– интеграция;
– тестирование;
– обучение.

Выделяются те операции, которые относятся к конкретным процессам и охватывают несколько этапов.

– **Процедуры управления проектами.**

А в свою очередь, для охвата всех процессов управления проектами выделены и такие процедуры, как Управление временем и затратами, коммуникациями, качеством, снабжением и прочие стандартные области проектного менеджмента.

– **Роли консультантов и клиентов.**

Sure Step определяет основные роли специалистов со стороны заказчика внедрения и подрядчиков и охватывает рекомендации для каждой из них.

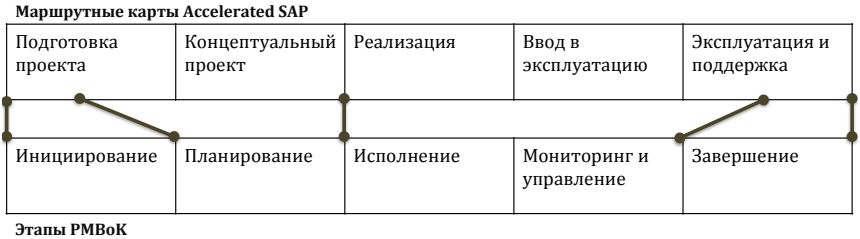
Со стороны заказчика	Со стороны исполнителя
– Руководитель, принимающий решения – Руководитель проекта – Руководитель отдела ИТ – Ключевой пользователь – Конечный пользователь	– Руководитель проекта – Менеджер по контактам – Архитектор решений – Консультант по приложениям – Консультант по разработке – Технический консультант

Сочетание одним специалистом нескольких ролей, в особенности в проектах небольшого масштаба соответствует методологии в полной мере.

Таким образом, MS Dynamics Sure Step представляет собой комплексную методологию внедрения, оптимальную для программных продуктов Dynamics CRM, AX, NAV за счет подробных шаблонов и схем процессов, а также средств редактирования для адаптации методологии к специфике предприятия и отрасли. Методология определяет необходимые шаги каждого этапа внедрения продуктов MS Dynamics как в виде иерархической структуры работ, так и в виде графических схем процессов MS Visio.

3.1.1.5. SAP (Accelerated SAP)

Призванная снижать риски, стоимость и время внедрения программных продуктов, методология ASAP основывается на собственном опыте SAP в области разработки технических решений и бизнес-консалтинга, со значительным фокусом на уже проверенных методиках управления проектами и инфраструктуре PMBoK. Так, пять ключевых этапов маршрутной карты SAP достаточно четко соотносятся с этапами проектного менеджмента PMBoK [53, сл. 6].



В целом же корпоративная методология Accelerated SAP (ASAP) предлагает комплексный инструментарий, сочетающий технологическую и управленческую части в виде трех основных компонентов:

- Синхронизация бизнес-приоритетов клиента (в зависимости от индустрии) с конкретными решениями SAP через отраслевые карты бизнес-ценностей (value maps) инструмента Solution Explorer.
- Карты управления проектами (маршрутные карты) с основными этапами и активностями по ним.
- Проектирование, настройка, тестирование и эксплуатация решения при помощи менеджера решений SAP Solution Manager.

Таким образом, за счет выстроенной методологии управления проектами внедрения своих программных продуктов, сочетающейся с программным инструментом для ее практического применения в виде Solution Explorer, SAP способствует структурированию большого количества задач, которые необходимо исполнять на каждом этапе.

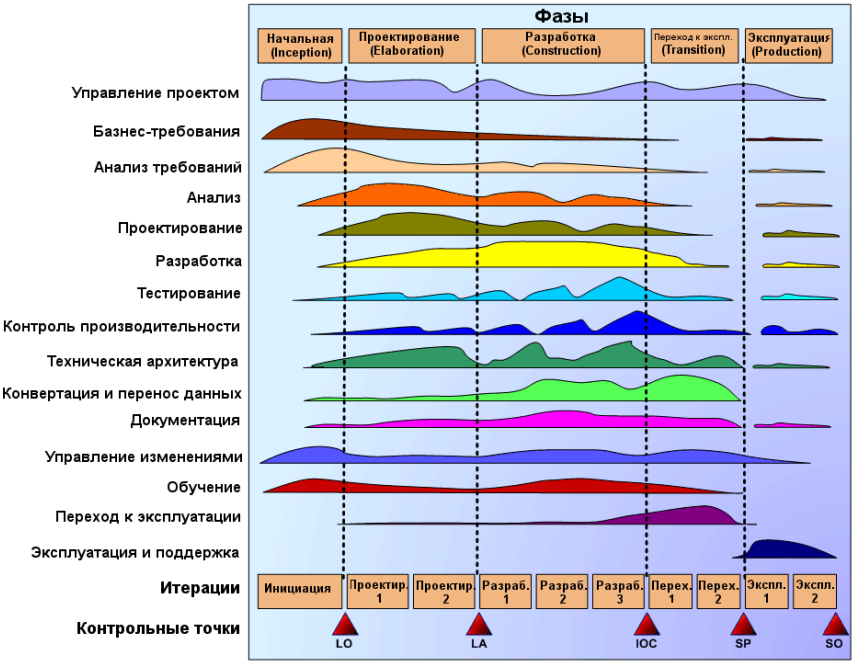
Среди других преимуществ данной методологии, уменьшается время между инсталляцией и запуском системы в продуктив в режиме реального времени, а также создает общую модель управления проектом, которая может впоследствии применяться и для других внедряемых компаний продуктов / обновлений системы.

3.1.1.6. Oracle (Oracle Unified Method, OUM)

Oracle Unified Method (OUM) – корпоративная методология поддержки ЖЦ информационных технологий на предприятии, содержащая общее ру-

ководство по настройке и типовую структуру работ, а также набор шаблонов, комплекты материалов по конкретным продуктам Oracle и необходимые программные инструменты. Изначально основой методологии Oracle стала модель унифицированного процесса разработки (Unified Process), которая была адаптирована также IBM для формирования собственной корпоративной методологии Rational Unified Process.

OUM охватывает три основных аспекта: управление программами и проектами, поддержку ИТ-архитектуры и подход к реализации проектов. Выделяя аналогичные ASAP пять стадий, Oracle изначально предполагает возможность итераций внутри каждой из них (две итерации проектирования, три итерации разработки) с наглядным представлением, на каком именно из 15ти процессов разработки следует фокусироваться в каждый конкретный момент [52].



Вполне резонно методология от Oracle предполагает регулярные активности процессов управления изменениями и документацией, множество раундов тестирования и контроля производительности, отдельное внимание технической архитектуре, обучение (в особенности на начальной стадии и в ходе разработки при подготовки к эксплуатации). А релизы поддерживают также сервис-ориентированную архитектуру, управление учетными запися-

ми, идеологию Entreprise 2.0 и другие актуальные проектные технологии, что способствует популярности данного метода разработки ПО среди корпоративных клиентов.

3.1.1.7. Oracle / PeopleSoft One Methodology

Разработанная компанией PeopleSoft, вошедшей в состав Oracle данная методология предназначена для описания внедрения ИС американской компании JD Edwards (в свою очередь, тоже поглощенной PeopleSoft). Первоначально разрабатывая финансовое ПО под аппаратное обеспечение IBM, JD Edwards создала методологию, в равной степени применимую к разным категориям систем: управления ресурсами, цепочками поставок, взаимоотношениями с клиентами, интеллектуального анализа данных и сотрудничества в Интернет.

Среди некоторых целей One Methodology – создание единой системы из иерархии целей проекта, его ограничений и планируемых результатов, формирование требований к команде внедрения, определение политики в области рисков. Содержание этапов данной методологии значительно отличается от рассмотренных ранее[1].

Этап	Описание работ
Границы внедрения (основные задачи и цели в рамках проекта)	– Определение функциональных целей – Разработка технологической архитектуры – Конвертация данных – Интерфейсы с внешними программами
Модель (проект системы)	– Общий обзор и планирование – Моделирование бизнес-процессов – Анализ недостающей функциональности – Планирование доработок ПО
Конфигурирование (реализация пилотного проекта и инсталляция / развертывание системы)	– Обучение проектной группы – Прогонка по системе – Ввод исходных данных – Конфигурирование ПО – Разработка пользовательской документации – Формирование прав доступа – Интеграция
Ввод в эксплуатацию (старт ОПЭ)	– Тестирование рабочей конфигурации – Обучение конечных пользователей – Настройка производительности системы – Запуск системы в ОПЭ
Развитие (оптимизация системы)	– Оценка работоспособности недостающей функциональности – Оптимизация бизнес-процессов – Передача системы

[1] По материалам [54].

Однако в результате последовательного поглощения JD Edwards → PeopleSoft → Oracle и значительной направленности JD Edwards на узкую линию программных продуктов, данная методология потеряла свою актуальность и уступила место другим методологиям, в частности, разработанной Oracle AIM в составе Oracle Unified Method.

3.1.2. Индустриальные стандарты и методологии

3.1.2.1. Agile

В переводе с английского Agile означает «гибкий» – именно поэтому такое название получила методология разработки, которая не содержит четких инструкций и «лучших практик», однако представляет собой сборник основных принципов и ценностей в предметной области разработки ПО. К тому же, сам процесс гибкой разработки является адаптивным по отношению к постоянно изменяющимся условиям, что достигается разработкой за короткие итерации, после каждой из которых происходит пересмотр требований и в случае необходимости – изменение практик коммуникаций и работы команды. Тем временем, есть еще несколько не менее важных и требующих рассмотрения идей Agile:

- Приоритет взаимодействия людей над процессами и традиционными инструментами управления.
- Приоритет получения работающего продукта над исчерпывающей всеобъемлющей документацией.
- Приоритет сотрудничества с потребителями (заказчиком) над формальными вопросами контрактов.
- Приоритет быстрого реагирования на изменения над неотступным следованием плану.

Исходя из этих ценностей, среди команд, использующих Agile-методологию, почти не встречается одинаковых практик документирования или координации, так как «гибкая методология» предполагает самоорганизацию, самостоятельное определение объема, содержания и ограничений каждого элемента управления разработкой. Со времени своего появления в 1990-х Agile-практики получили свое распространение во множестве компаний, в то время как первоначально они позиционировались для управления ИТ-проектами.

В качестве одного из важнейших элементов Agile-подхода к разработке выступают пользовательские истории (user stories). Это описанные одним или несколькими предложениями основные аспекты функциональности системы, необходимые для выполнения пользователем своей работы. В достаточно сжатом виде, они должны отвечать на вопросы: «Кто?», «Что?» и «По-

чему?» и могут создаваться как менеджером проекта по результатам обследования как один из форматов фиксирования требований, так и разработчиками для выражения нефункциональных требований (например, к безопасности, производительности, качеству решения).

ПРИМЕР

Отдельные требования, представленные в виде user stories:

Как ответственный за проведение рассылок, я должен вести мониторинг адресатов по разным категориям.

Как ответственный за проведение рассылок, я бы хотел давать права администратора моим подчиненным в случае необходимости.

Как ответственный за проведение рассылок, я хочу видеть знать, сколько процентов пользователей проходят по ссылке в письме.

Как ответственный за проведение рассылок, я должен иметь возможность выделить адресатам, не открывшим письмо в течение двух недель.

Как ответственный за проведение рассылок, я бы хотел создавать их план в перспективе на 2 недели вперед.

Как ответственный за проведение рассылок, мне необходим удобный формат визуального представления и выгрузки статистических данных.

Как ответственный за проведение рассылок, я должен знать источник получения контактных данных каждого адресата.

Как ответственный за проведение рассылок, я должен получать недублированные данные, то есть одному пользователю может соответствовать только один текущий электронный адрес.

Итоговая user story на «техническом» языке:

Как ответственному за проведение рассылок, пользователю необходимо обеспечить следующую функциональность:

- *Иметь конфигурацию статусов каждого адресата: адрес электронной почты, реакция на каждую из проведенных рассылок (красная иконка для открывших письмо и не прореагировавших на него, желтая – для неоткрытых сообщений, зеленая – для прочитавших и выполнивших требовавшееся в рассылке действие).*
- *Иметь несколько уровней прав доступа к рассылкам (на просмотр статистики, на выгрузку e-mail адресов, на отправку рассылок, …).*
- *Создать персональные ссылки в отправляемых сообщениях, через которые будет осуществлен подсчет числа пользователей, пришедших на страничку портала по результатам рассылки.*
- *…*

В то же время, agile почти не ограничивает команду проекта: так, если необходим больший уровень детализации технического задания / спецификаций / другой документации – можно его написать, если необходимы дополнительные приемочные критерии – можно их составить, если нужны прототипы – можно их сформировать или в виде макетов (мокапов, mockups), или в виде блок-схем, или в виде реальной модели прототипа. Уровень детализации и формализации всегда зависит от специфики системы и проектной команды, от их компетенций и опыта реализации подобных проектов.

3.1.2.2. SCRUM

В переводе с английского языка «SCRUM» означает «схватка». Впервые подобная методология была введена в употребление профессорами японского Hitotsubashi University в их статье «Новая игра в производстве продукта» («The New Product Development Game»), написанной в 1986 году. В ней разработка продукта была сравнена со схваткой вокруг мяча в регби – приемом, называемым «скрам». Тем не менее, авторами концепции считаются разработчики из США, Кен Швабер и Джефф Сазерленд, впервые определившие и описавшие основополагающие принципы SCRUM – «гибкой» методологии управления проектами (чаще всего применяющейся в области разработки ПО).

По своей сути SCRUM представляет собой набор принципов разработки, которые позволяет представлять конечному пользователю (и заказчику) действующий продукт / прототип, обладающий новыми функциями и возможностями с наивысшим приоритетом. Работа в этом случае проводится в четко фиксированные недлительные (в среднем от 1 до 6 недель, чаще от 2 до 4) итерации (спринты). Ожидаемые результаты спринта определяются командой под руководством SCRUM-мастера в самом его начале и НЕ могут изменяться до завершения. К ним под руководством SCRUM-мастера (по сути руководителя проекта) и product-мастера (заказчика, владельца проекта) создается список работ на период спринта и на весь проект (sprint-backlog и product-backlog соответственно).

С одной стороны, в силу того, что релизы выпускаются часто, минимизируется вероятность ошибок за счет постоянной обратной связи с потребителями при демонстрации новой функциональности, а также ежедневных (!) встреч команды проекта. С другой стороны, увеличиваются временные (и денежные) затраты на проведение презентаций, а также на исправление выявленных проблем и осуществляемый ретроспективный анализ.

Однако при более глубоком рассмотрении SCRUM имеет очень важные основы: методология предполагает циклический и очень активный процесс, минимизирующий неопределенность требований за счет коротких итераций и предоставляя возможность детального прототипирования. Долгие, нестандартные, динамические проекты, при расплывчатом представлении о конечном ожидаемом результате и постоянной смене приоритетов не являются проблемой для SCRUM, и именно поэтому он часто применяется на первых этапах, затем в целях экономии средств / ресурсов переходя к использованию другой методологии. Соответственно, SCRUM может применяться практически любым типом организаций – от стартапов с их высокой неопределенностью до госзаказчиков с частым изменением требований и частой необходимостью демонстрации текущей версии для отчета о результатах.

При принятии решения о переходе на SCRUM необходимо принимать во внимание возможное сопротивление со стороны некоторых специалистов, не привыкших к подобным подходам или предпочитающими индиви-

дуальную работу, так как именно в этой методологии крайне важна сплоченность команды, ее уровень ответственности и кросс-функциональность для ежедневного поиска оптимальных путей совершенствования продукта. Также высока зависимость успеха SCRUM-проектов от мастерства и компетенций SCRUM-мастера, который должен обладать достаточным авторитетом и умением управлять командой для успешной организации выполнения работ и контроля проекта на всех этапах.

3.1.2.3. RAD

Методология быстрой разработки приложений[1] RAD (rapid application development) предполагает наличие трех основных элементов: небольшой команды до 10 человек, непродолжительный период внедрения в 2-6 месяцев и итеративный цикл, в основе которого спиральная модель жизненного цикла ИС. Основу разработки RAD составляет использование инструментальных средств для каждой фазы жизненного цикла системы – анализа требований, проектирования, генерации кода и разработки интерфейса. Часто используются CASE-средства и инструменты прототипирования интерфейса, которые позволяют продемонстрировать саму концепцию планируемого результата уже в самом начале проекта. Именно поэтому итерационный принцип настолько соответствует методологии и он удобен при нечетко определенных и изменяющихся требованиях к итоговой системе. Итоговый результат, работоспособная система, актуальная на текущий момент (не обязательно в перспективе нескольких лет!) может быть получен более оперативно именно при помощи RAD.

Однако есть и некоторые отличия с традиционной спиральной моделью. В силу этого же итеративного подход, как и другие подобные методологии, RAD не может применяться для разработки сложных систем. (например, от которых зависит безопасность людей, таких как системы управления судами или самолетами), так как первые версии не будут полностью работоспособны. Не следует удивляться также и предложению подрядчика использовать принципы RAD на первых этапах, затем переписывая часть кода на «традиционном» языке программирования – таком, как C# или C++.

Преимуществом RAD среди методологий разработки является совместная работа с заказчиком / бизнес-пользователями, которые активно участвуют в прототипировании, написании сценариев тестирования и его проведении. Другой особенностью является совместное принятие решений команды, и подобный децентрализованный стиль управления может привести к периодической смене курса проекта или возможному излишне длительному принятию решения из-за поиска консенсуса.

[1] На основе материалов портала edu.dvgups.ru.

3.1.2.4. XP

Основная особенность[1] методологии экстремального программирования (**eXtreme Programming**) состоит в ее эффективности в условиях неопределенных или нечетких требований. Популярность данного подхода увеличивалась по мере роста числа разработчиков, недовольных традиционным подходом со множеством формальных требований и постоянно готовивших документацию, собиравшими информацию о показателях проекта. Напротив, новая методология предлагала простой дизайн, переработку (рефакторинг) кода для контроля затрат, постоянное присутствие заказчика, разработку через тесты и другие аспекты, выгодно отличавшие ее.

Обозначим другие ключевые аспекты данной методологии.

Для работы в подобных условиях необходима **постоянная связь с заказчиком**, которая обеспечивается полноценным участием в работе проектной команды квалифицированного, находящегося в курсе проекта представителя заказчика.

При разработке в большинстве случаев выбираются наиболее простые методы, исходя из принципа того, что легче в будущем вносить дополнения в базовую версию, чем перестраивать усложненную (хотя, разумеется, бывают исключения, когда изначально учет многих факторов в системе может помочь избежать в дальнейшем многих сложностей). **Принцип простоты** также важен и в интерфейсном решении, когда более интуитивный пользовательский дизайн интерфейса обладает исключительно необходимыми (но не излишними!) функциями. Однако предварительное детальное проектирование интерфейса согласно исходной версии методологии не осуществляется, он создается лишь по мере работы команды в течение проекта.

Коллективная работа по написанию кода / внесению изменений в настройки. Этот принцип относится к двум аспектам. В первую очередь, это «**коллективное владение кодом**», когда любой участник команды разработчиков может внести изменения благодаря единым правилам оформления когда и использованию стандартов для ПО. С другой стороны, применяется метод «**парного программирования**», в соответствии с которым двое разработчиков используют один компьютер, чередуя написание кода и доработку настроек, реализацию требований и прочие вопросы.

3.2. Сервисный подход к эксплуатации ИС

На этапе эксплуатации информационной системы любые сбои в ее работе будут иметь гораздо большее негативное влияние и последствия, нежели на стадии разработки и внедрения. Для того, чтобы обеспечить постоян-

[1] На основе материалов портала edu.dvgups.ru.

ное качество работы с системой, необходимо с самого начала ориентироваться на бизнес-пользователей и их потребности. Эффективность внедряемой системы во многом от технической составляющей, а именно: доступности и надежности / безошибочности работы. Эти факторы часто упоминаются пользователями при определении требований к системе еще на этапе аудита бизнес-процессов наряду с функциональностью системы, но редко пользователи осознают, что говоря о своем целевом представлении системы на самом деле они дают формулировку той ИТ-услуге, которую при помощи системы будут получать. Именно эта идея работы ИТ-подразделений как равноценного партнера бизнеса (**поставщика ИТ-услуг**, принимающего на себя внутренние процессы по ежедневной поддержке программных и инфраструктурных ресурсов для непрерывного и качественного функционирования систем), является основой **сервисного подхода.**

Сервис (услуга) – способ предоставления ценности заказчикам посредством содействия им в получении требуемых результатов без овладения специфическими расходами и рисками (ITIL v3).

Наиболее важные и заманчивые для руководителей компаний положения сервисного подхода направлены на улучшение коммуникаций и более бизнес-ориентированную трактовку деятельности ИТ-подразделений.

– Все активы ИТ-подразделений (лицензии, сервера, системы хранения) являются не отдельными объектами, а составляющими, необходимыми для предоставления той или иной услуги. Пользователи систем являются потребителями данных сервисов.

ПРИМЕР

Система документооборота – не просто комбинация необходимых программных и аппаратных компонентов, а также поддерживающих ИТ-специалистов, а отдельный сервис «Предоставление услуги документооборота», со своими требованиями к доступности, предоставлению консультационной поддержки и планами на расширение.

– Наиболее оптимальными для Бизнеса и ИТ являются отношения «покупатель-продавец» услуг (вместо «владелец-пользователь» бюджета и приложений / инфраструктуры соответственно).
– Качество (а также объем и время предоставления) услуг оговариваются заранее в виде соглашений об уровне предоставления услуг SLA – и именно под эти соглашения / условия выделяется финансирование.

SLA (Service Level Agreement, Соглашение об уровне обслуживания) – заключенный в соответствии с общепринятыми мировыми практиками договор, включающий описание предоставляемой ИТ-услуги, требования к ее доступности, порядок действий исполнителя при различных условиях предоставления сервиса, критерии удовлетворительного предоставления услуг и порядок их оплаты.

ПРИМЕР

Приведем пример некоторых метрик соглашений SLA:

- ***Название сервиса:*** *Услуги удостоверяющего центра.*
- ***Краткое описание:*** *Обеспечение использования сервисов защиты информации на основе сертификатов открытых ключей (с применением шифрования и ЭЦП).*
- ***Доступность сервиса****: 24 · 7 · 365.*
- ***Контрольное время реакции:*** *60 минут с момента регистрации инцидента.*
- ***Контрольное время устранения инцидента****: 1 сутки с момента регистрации заявки.*
- ***Режим работы службы поддержки пользователей****: 09:00-18:00 понедельник-пятница, за исключением праздничных дней.*
- ***Тип предоставляемой отчетности****: Еженедельный отчет обо все существенных инцидентах, которые привели к недоступности сервиса более 4 часов.*
- ***Ответственный за сервис****: ФИО, контактные данные.*
- ***Стоимость услуг:***
  - *Сервисная годовая поддержка: ___ руб.*
    - *Аутентификация домена в AD и на ЭТП компании: ___ руб.*
    - *Внутренний почтовый документооборот: ___ руб.*
  - *Подключение сервиса у оператора: ___ руб.*
  - *Дополнительные услуги:*
    - *Сервисная поддержка с выездом специалиста: ___ руб./час (за выезд)*
    - *SSL-сертификат (одно- и мульти-доменный): ___ руб. / 3 года*
    - *Восстановление сервиса при утрате носителя: ___ руб.*
    - *Подключение к VPN: ___ руб. (при подключении).*

Говоря о качестве услуг, невозможно не упомянуть столь важный его критерий, как доступность информационной системы. Чем реже происходят отказы (в идеале – никогда!) и чем к меньшему числу часов вынужденных простоев они приводят (чем быстрее устраняются), тем выше доступность ИС. Основные методы достижения максимальной надежности системы направлены на управление тремя взаимозависимыми факторами:

- Снижением вероятности отказов системы.
- Уменьшение влияния этих отказов на работоспособность системы – и как следствие.
- Сокращение времени простоев при отказе [56].

Среди факторов, влияющих на обеспечение доступности / **снижение вероятности отказов системы**:

- Соответствие выбранного программно-аппаратного решения специфике бизнес-процессов предприятия.
- Низкая степень сложности архитектуры системы.
- Надежность (отсутствие ошибок) программного продукта и проведенных в дальнейшем настроек.
- Минимизация риска некорректного использования системы и внесения нежелательных изменений со стороны пользователя.

– Организация регламентно-профилактического обслуживания во время эксплуатации системы для обнаружения и предотвращения ошибок.

Для **уменьшения влияния возможных отказов на работоспособность системы** необходимо обеспечить восстановление деятельности систем в случае нарушения их непрерывности. В настоящее время распространены несколько типовых концепций восстановления. Они могут различаться по принципу построения (территориально распределенная, с использованием мобильной платформы, локальная), по принципу резервного копирования («холодный» резерв, удаленное резервное копирование, отсутствие резервных ресурсов).

ПРИМЕР

Для компаний с филиалами в различных районах региона и страны может быть актуально рассмотрение конфигурации, основанной на мобильной платформе с вычислительным резервом средней производительности. Таким образом, если один из вычислительных центров подвергся природным катаклизмам / катастрофе / вышел из строя, подобная платформа будет обладать минимальным необходимым для поддержания деятельности набором систем хранения данных, набором ИТ-сервисов и резервными сервисами. А значит, она может быть в оперативном режиме перемещена в другую локацию и подключена к каналам передачи данных и другим системам для временного развертывания на ней необходимых приложений (до восстановления основной площадки).

Таким образом, выбор стратегии непрерывности зависит от допустимого времени восстановления (от нескольких часов до нескольких дней), территориальной распределенности, необходимого режима восстановления и стоимости / сложности реализации.

Не последнюю роль в стратегии непрерывности играет решение по хранению и обработке данных, в частности, информационная, телекоммуникационная и инженерная инфраструктуры дата-центра. Как правило, при выборе площадки для дата центра в первую очередь обращают внимание на его стоимость эксплуатации и отказоустойчивость, однако с точки зрения ИТ важны также безопасность и энерговооруженность, так как возможности выбранной площадки дата-центра должны соответствовать задачам и потребностям самого бизнеса. Для размещения ИТ-инфраструктуры критически важных бизнес-сервисов (например, системы онлайн-процессинга операций по карточкам для банка) возможно выбрать более дорогой и надежный ЦОД уровня **TIER IV** (отказоустойчивость 99.995 % – простои не более 24 минут в год), ведь банк несет ответственность как перед международными системами типа Visa и Mastercard, так и своими клиентами за бесперебойное качественное предоставление сервиса обработки транзакций. В это же время для инфраструктуры остальных сервисов (например, call-центра) довольствоваться более дешевым решением **TIER II** с допускаемыми перерывами в работе до 22 часов в год.

TIER (англ. уровень) – классификация ЦОД по четырем уровням надежности. Первый уровень TIER I является наименее отказоустойчивым, на время проведения регламентных работ допускается временное прекращение работы ЦОД. С технической точки зрения в дата-центрах TIER I отсутствуют фальшполы, резервирование инженерной инфраструктуры и источников электроснабжения. С увеличением уровней TIER растет степень организации резервирования, и уже для TIER IV любые работы могут проводиться без остановки дата-центра и все инженерные / информационные / телекоммуникационные системы продублированы. Таким образом, проводится максимально качественный мониторинг параметров функционирования оборудования, состояния сети, гораздо более высокие требования к персоналу.
Первым российским дата-центром, получившим сертификат подтверждения высшего уровня надежности TIER IV от Uptime Institute, стал ЦОД «Технопарк-Мордовия», открытый в ноябре 2013 года.
Существуют два основных документа, регламентирующие требования к уровням TIER:
1. стандарт рекомендательного характера TIA 942 с пошаговыми инструкциями и схемами, на соответствие которым проверяется проектная документация.
2. методология нормирования отказоустойчивости Uptime Institute, строгое соответствие требованиями которой как во время создания ЦОД, так и в процессе его эксплуатации необходимо для получения сертификации площадки.

И наконец, переходя к третьему показателю надежности, **снижению времени простоев при отказе**, отметим важность реализации возможностей быстрого и эффективного осуществления ремонта. Для этого еще на этапе проектирования необходимо предусмотреть инструменты восстановления / коррекции данных, принудительного завершения не отвечающих на запросы процессов, и прочих возможностей реагирования на возникшие ошибки.

3.3. Элементы управления корпоративными ИТ

3.3.1. CobiT

CobiT (сокращение от Control Objectives for Information and Related Technology, «Задачи информационных и смежных технологий») представляет собой пакет открытых документов, около 40 международных и национальных стандартов и руководств в области управления IT, аудита и IT-безопасности.

Создателем CobiT является Международная ассоциация аудита и контроля за информационными системами (ISACA), которая провела анализ и оценку практик управления ИС, объединив лучшее из международных технических стандартов, стандартов управления качеством, аудиторской деятельности, а также из практических требований и опыта – все то, что так или иначе имело отношение к целям управления. Впервые будучи опубли-

кованным в апреле 1996 года, CobiT с тех пор претерпел немало изменений как в части структуры, так и по содержанию. На сегодняшний день для наиболее полного описания руководства и управления ИТ на предприятии создано целое семейство публикаций CobiT 5, среди основных областей внимания которых:

- Информационная безопасность.
- Управление рисками.
- Управление непрерывностью бизнеса.
- Управление качеством.
- Обеспечение соответствия требованиям регуляторов.
- Защита интеллектуальной собственности.

COBIT, будучи единым сборником рекомендаций, включает некоторые элементы таких концепций и методологий, как Закон Сарбейнса-Оксли (Sarbanes-Oxley), Базельское соглашение (Basel III), COSO, ITIL, PRINCE2, TOGAF, PMBOK, PCI DSS. Среди данного списка отсутствуют такие популярные руководства, как Val IT и Risk IT, так как пятая версия интегрирует их содержание.

Далее рассмотрим основные элементы стандарта для формирования представления о его структуре.

Модель связи целей бизнеса и ИТ

Пятая версия COBIT акцентирует внимание на тесной взаимосвязи целей бизнеса и ИТ, связывая их при помощи четырех измерений Системы сбалансированных показателей BSC. Каждому измерению BSC ставятся в соответствие определенные Цели предприятия, которые в свою очередь реализуют потребности заинтересованных сторон (в реализации выгод, оптимизации рисков или оптимизации ресурсов). В дальнейшем Цели предприятия переводятся в плоскость ИТ-целей.

ПРИМЕР

Допустим, бизнес-целью является «Обеспечить корпоративную репутацию и лидерство». Эта цель может быть измерена числом инцидентов, вызывающих публичные проблемы. В таком случае ИТ-целью будет «Обеспечить защиту и восстановление ИТ-сервисов от атак», индикатором для которой является «Число ИТ-инцидентов с последствиями для бизнеса».

Руководящие принципы CobiT

- Соответствие потребностям заинтересованных сторон.
- Комплексный взгляд на предприятие.
- Применение единой интегрированной методологии.
- Обеспечение целостности подхода.
- Разделение руководства и управления.

7 факторов влияния CobiT (CobiT Enablers)

Авторы COBIT выделяют семь ключевых групп факторов влияния (т.н. **enablers**), на которых и основываются все публикации семейства COBIT.

- **Принципы, политики и концепции –** основные инструменты «трансляции» предлагаемой линии поведения в области управления ИТ.
- **Процессы –** организованный набор практик и активностей для достижения определенного результата для обеспечения достижения относящихся к ИТ целей компании.
- **Организационные структуры –** основные сущности в области принятия решений на предприятии.
- **Культура, этика и поведение –** важный фактор успеха при управлении.
- **Информация –** необходимый элемент поддержки деятельности организации, который также на операционном уровне может являться основным продуктом, производимым компанией.
- **Сервисы, инфраструктура и приложения –** включают в себя инфраструктуру, технологии и приложения, которые обеспечивают обработку информации и предоставление сервисов в компании.
- **Сотрудники, навыки и компетенции –** необходимые составляющие деятельности компании.

Общая модель факторов влияния представлена ниже.

Среди атрибутов факторов влияния:

- **Заинтересованные стороны**. Этот атрибут важен, так как потребности выделенных заинтересованных сторон транслируются в цели компании, а те, в свою очередь, в ИТ-цели.
- **Цели факторов влияния**. Они могут относиться к точности / объективности / надежности результатов их работы, доступности и безопасности, актуальности / понятности / удобству применения факторов.
- **Жизненный цикл**. Он подобен практически любому жизненному циклу систем, организационных структур и политик, и включает планирование – проектирование – построение / приобретение / создание / внедрение – использование / эксплуатацию – оценку / мониторинг – обновление / утилизацию.
- **Передовые практики**. Примеры наиболее эффективной реализации факторов влияния, их входов и выходов.

Процессы CobiT

CobiT делит всю деятельность ИТ на 5 доменов (группы или сферы деятельности), которые включают 37 процесса (в версии CobiT 5).

> В терминах **CobiT процесс** – «набор задач (действий), объединенных естественным образом и содержащих детальные цели контроля».
> Цели контроля описывают требования по эффективному контролю и представляют собой:
> – Формулировки управленческих действий,
> – Политики, процедуры, практики, организационные структуры.

Процессы таким образом объединяются в следующие домены (группы):

- Координация, планирование и организация, APO (**процесс управления** Plan and Organise в CobiT 4.1).

Согласованность ИТ-стратегии с бизнесом и корпоративной стратегией, обеспечение соответствия ИТ потребностям бизнеса, вопросы управления инвестициями в ИТ и оптимального использования имеющихся ИТ-ресурсов.

– Разработка, приобретение и внедрение, BAI (**процесс управления** Acquire and Implement в CobiT 4.1).

Выбор платформ и инфраструктурных ресурсов для соответствия потребностям бизнеса, вопросы интеграции систем и управления изменениями.

– Предоставление, обслуживание и поддержка, DSS (**процесс управления** Deliver and Support в CobiT 4.1).

Управление уровнем обслуживания и обеспечение непрерывности бизнеса, управление информационной безопасностью (процессы данного домена во многом соотносятся с процессами библиотеки ITIL).

– Мониторинг, оценка и анализ, MEA (**процесс управления** Monitor and Evalutate в CobiT 4.1).

Обеспечение обратной связи: «эффективность ИТ» – «цели бизнеса», корпоративное управление ИТ и оценка эффективности ИТ.

– Оценка, задание направления и мониторинг, EDM (**процесс руководства**, отсутствующий в CobiT 4.1).

COBIT также приводит процессы на более детализированном уровне, полная схема приведена в приложении.

ПРИМЕР

– *DSS (Предоставление, обслуживание и поддержка):*
– *DSS01 – Управление эксплуатацией.*
– *DSS02 – Управление запросами на обслуживание и инцидентами.*
– *DSS03 – Управление проблемами.*
– *DSS04 – Управление непрерывностью.*
– *DSS05 – Управление услугами безопасности.*
– *DSS06 – Управление контролями бизнес-процессов.*

Именно в подобной модели процессов и становится более очевидной связь с ITIL и TOGAF в силу использования соответствующей терминологии и сходства упоминаемых процессов.

Также для каждого процесса определены:

1. **Ключевые индикаторы достижения цели** (KGI) – метрики, которые показывают руководству, достигнуты ли цели бизнеса с помощью ИТ-процесса, и в какой степени.
2. **Ключевые показатели эффективности** (KPI) – метрики, которые показывают, насколько хорошо работает ИТ-процесс.
3. **Степень зрелости процесса** – определяется по шкале от 0 до 5 (5 – высшая оценка, когда процессы развиты до уровня передовой практики) и помогает организациям понять, где они сейчас находятся и куда могут (или хотят) прийти в будущем (включая пути совершенствования процесса).

Цели и метрики процессов

Важной особенностью пятой версии документа является новое понятие **модели возможностей процессов.** Она создана как способ измерения производительности процессов доменов руководства и управления ИТ (EDM и PBRM соответственно). Отдельно основные отличия данного подхода рассмотрены в публикации **ISACA COBIT Process Assessment Model (PAM).**

На первый взгляд, может показаться что дополнение модели новым элементом – **атрибутами способностей процессов** – только усложняет ее. Однако их роль состоит в том, чтобы являться своего рода признаками, необходимыми для того, чтобы утверждать о достижении определенного уровня зрелости.

ПРИМЕР

Процесс является «установленным» только в случае, когда реализованы все атрибуты текущего (3-го) и предыдущих (1-го и 2-го уровней), а именно – налажено управление производительностью, рабочими продуктами, внедрением процесса и пр.

Уровень 0. Неполный процесс.

Процесс еще не внедрен либо неспособен систематически достигать своих целей.

Уровень 1. Осуществленный процесс.

Внедренный процесс соответствует своему назначению.

Уровень 2. Управляемый процесс.

Осуществленный процесс планируется, отслеживается и корректируется.

Уровень 3. Установленный процесс.

Управляемый процесс достигает ожидаемых результатов.

Уровень 4. Предсказуемый процесс.

Установленный процесс получает результатов даже в условиях задаваемых ограничений.

Уровень 5. Оптимизируемый процесс.

Предсказуемый процесс непрерывно улучшается.

Матрица ответственности RACI

Также в помощь проектным менеджерам CobiT предоставляет RACI-матрицы распределения ответственности и делегирования задач, определенные для процессов ресурсы и результаты (показывающие, что именно владелец процесса должен получить от других лиц для успешного выполнения своего процесса – и что, в свою очередь, предоставить в конечном счете потребителям). RACI проводит соответствие между видами деятельности процессов и участниками процессов, определяя для них одну из четырех ролей:

- ответственный (Responsible);
- утверждающий (Accountable);
- консультирующий (Consulting);
- информируемый (Informed).

ПРИМЕР

Ключевые практики	*Совет директоров*	*CEO*	*CFO*	*COO*	*Руководители бизнес-подразделений*	*Владельцы бизнес-процессов*	*Комитет по стратегии*	*Управляющий комитет программ / проектов*	*Проектный офис PMO*
EDM01.01 *Оценка системы корпоративного управления*	*A*	*R*	*C*	*C*	*R*		*R*		
EDM01.02 *Руководство системы корпоративного управления*	*A*	*R*	*C*	*C*	*R*	*I*	*R*	*I*	*I*
EDM01.03 *Мониторинг системы корпоративного управления*	*A*	*R*	*C*	*C*	*R*	*I*	*R*	*I*	*I*

3.3.2. ITIL

Несмотря на разницу в областях применения, ITIL часто упоминается в одном ряду с COBIT как популярные инструменты управления ИТ. Однако в то время, как COBIT является более высокоуровневым руководством, концентрирующемся в основном на корпоративном управлении ИТ и ИТ-аудите, ITIL представляет собой сборник лучших практик построения / организации ИТ-процессов.

ITIL (IT Infrastructure Library) – библиотека, описывающая лучшие из применяемых на практике способов организации работы подразделений или компаний, занимающихся предоставлением услуг в области информационных технологий.

Библиотека ITIL появилась около 25 лет назад по заказу британского правительства. В семи томах библиотеки описан весь набор процессов, необходимых для того, чтобы обеспечить постоянное высокое качество ИТ-сервисов и повысить степень удовлетворенности пользователей. Официально данные книги носят название **OGC Service Books** (OGC – правительственное агентство Великобритании, выпустившее данные издания).

ПРИМЕР
OGC Service Strategy Book. TSO (The Stationery Office), 2007.

Третья, выпущенная в 2007 году, редакция ITIL (с последующим обновлением в 2011) включила в себя пять основных книг, поддерживающих концепцию «жизненного цикла услуг»:

- Стратегия услуг (Service Strategy).
- Проектирование услуг (Service Design).
- Преобразование услуг (Service Transition).
- Эксплуатация услуг (Service Operation).
- Постоянное улучшение услуг (Continual Service Improvement).

Основные положения ITIL

ITIL исходит из того, что деятельность ИТ-службы фокусируется на обеспечении основной бизнес-деятельности компании полным набором информационных сервисов. Качество сервиса при этом является величиной измеряемой и фиксируется в Соглашениях об уровне предоставления услуг SLA, в которых также указываются все параметры поставляемых услуг.

Приведем ниже списки процессов каждой из книг ITIL и отдельно выделим 10 процессов, которые выделялись в качестве основных в предыдущей версии ITIL (v2) и до сих пор наиболее часто упоминаемые многими источниками по всему миру.

Книга ITIL	Основные процессы
Стратегия услуг Бизнес-ориентация в управлении ИТ	– Формирование стратегии – **Управление финансами** – Управление портфелем услуг – Управление требованиями
Проектирование услуг Построение поддерживающей стр=атегию системы управления	– Управление каталогом услуг – **Управление уровнем сервиса** – **Управление доступностью** – **Управление мощностями** – **Управление непрерывностью** – Управление безопасностью – Управление поставщиками
Преобразование / внедрение услуг Трансформация системы управления за счет сочетания человеческих ресурсов, процессов и технологий	– Планирование и поддержка развития – **Управление изменениями** – **Управление активами сервисов и конфигурациями** – **Управление релизами и развертыванием** – Отладка и тестирование сервисов – Оценка – Управление знаниями
Эксплуатация услуг Операционная деятельность	– Управление событиями – **Управление инцидентами** – Выполнение запросов – **Управление проблемами** – Управление доступом
Постоянное улучшение услуг Совершенствование ИТ-процессов	– Совершенствование сервисов – Измерение сервисов – Подготовка отчетности о сервисах

Подробнее опишем десять вышеприведенных процессов.

Управление инцидентами (Incident management).

Регистрация и реагирование на любые события и обращения пользователей, которые требуют реагирования. Как правило, эта функция возлагается на службу поддержки, которая должна разрешать основные вопросы в сфере своих компетенций и в случае необходимости привлекать смежные службы и другие группы / линии поддержки для скорейшего закрытия инцидентов.

Управление проблемами (Problem management).

Поиск, анализ, мониторинг проблем, вызывающих инциденты, с целью минимизации их негативного влияния на предоставление услуг, а также предотвращения последующих инцидентов. В качестве таких проактивных мер проводятся анализ трендов, анализ инцидентов в рамках определенной тематики или происходящих в определенной организационной единице. Важно перейти от множества проблем (как неизвестных причин появления инцидентов) к множеству «известных ошибок» (как причин появления инцидентов, которые уже определены и пути решения которых уже найдены).

Управление конфигурациями (Configuration management).

Хранение информации о логической модели ИТ-инфраструктуры компании: существующих конфигурационных компонентах и их взаимосвязях.

Именно этим процессом поддерживается создание базы конфигурационных элементов CMDB, ее планирование и поддержка.

Управление изменениями (Change management).

Координация производимых изменений ИТ-сервисов и поддерживающих ее ресурсов и минимизация рисков возникновения вызванных изменениями инцидентов. Изменением признается событие, приводящее к смене статуса одного или более конфигурационных элементов. Важно, что изменение должно быть санкционировано руководством, быть эффективным с точки зрения стоимости и влияния на бизнес-процессы. В ITIL используется распространенный термин **запроса на изменение (request for change, RFC)**, которым обозначается документ с описанием деталей необходимых изменений, широко использующийся и в сфере управления проектами.

Управление релизами (Release and deployment management).

Планирование и мониторинг жизненного цикла релизов от проектирования до тестирования и эксплуатации при сохранении работоспособности всех систем / инфраструктуры во время изменений. Данный процесс в ITIL отвечает за ввод в эксплуатацию программно-аппаратного обеспечения, которое бы полностью удовлетворяло общей ИТ-архитектуре компании. Сюда относятся и крупные релизы с большим объемом новой функциональности (возможно, даже избыточной), и небольшие изменения (содержащие определенные улучшения / исправления), и срочные доработки для устранения найденных проблем.

Важными элементами управления релизами является контроль версионности, лицензий, проведение тестирования, управление ожиданиями заказчика и другие аспекты.

Управление уровнем сервиса (Service level management).

Написание и заключение соглашений об уровне услуг SLA для формирования единого представления о выполнении услуг и его закрепления в виде договорных обязательств. Одним из признаков успешного управления уровнем сервиса является возможность определения метрик качества обеспечения сервиса и их сравнения с контрольными / плановыми значениями либо практиками других подразделений / компаний / отраслей в процессе бенчмаркинга.

Управление финансами (Financial management).

Обеспечение проведения всех необходимых финансовых операций и получения средств для поддержки ИТ-активов (в частности, задействованных в предоставлении сервисов).

Управление мощностями (Capacity management).

Поддержка оптимальных по ресурсным и стоимостным характеристикам структуры и объема ресурсов, необходимых для предоставления услуг. При этом в управлении мощностями важно рассчитывать и общие стратегические потребности, в том числе количество персонала (их распределение

на проекты и потребности в обучении), мощности систем, мощности компонентов инфраструктуры. В данном процессе очень важно найти компромиссное решение в условиях риска задействовать недостаточный либо же избыточный объем ресурсов.

Среди активностей этого процесса: сайзинг приложений, управление спросом, управление рабочей нагрузкой, ресурсами, эффективностью.

Управление непрерывностью (Continuity management).

Обеспечение возможности восстановления нормального режима работы сервисов и проведения бизнес-операций в случае реализации рисков/ наступления чрезвычайной ситуации. К процессу также относится снижение вероятности реализации подобных рисков нарушения непрерывности бизнес-операций.

Для этого в рамках процесса организуются планирование и приоритезация активностей для проведения **анализа бизнес-влияния (business impact analysis)**, оценка рисков и разработка альтернативных путей восстановления, включение их в план обеспечения непрерывности, корректируемый на регулярной основе.

Управление доступностью (Availability management).

Обеспечение возможности представления услуг за счет контроля и управления всех ресурсов, их поддерживающих. Доступность сервисов часто включается в SLA-соглашения как один из основных критериев оценки уровня предоставления услуг (и обоснования стоимости сервиса).

В рамках процесса проводятся анализ требований к доступности, составление плана, мониторинг его исполнения и принятие корректирующих мер. В частности, оценивается стабильность ИТ-компонентов, поддерживаемость, безопасность, гибкость (зависимость от сбоев аппаратных платформ).

3.4. Проектное управление

3.4.1. Российские и международные стандарты

3.4.1.1. SWEBOK

Одним из самых полных руководств по системной инженерии на сегодняшний день является **Руководство к методическому справочнику по программной инженерии** (Guide to the Software Engineering Body of Knowledge, SWEBOK). Формально данный документ не является стандартом или методологией и представляет собой использующийся тысячами компаний систематизированный справочник по основным десяти областям знаний по разработке программного обеспечения.

Ключевые 10 процессов (глав) SWEBOK:

- Программные требования (*Software requirements*).
- Дизайн / архитектура (*Software design*).
- Конструирование программного обеспечения (*Software construction*).
- Тестирование (*Software testing*).
- Эксплуатация / поддержка программного обеспечения *(Software maintenance).*
- Конфигурационное управление (*Software configuration management*).
- Управление в программной инженерии (*Software engineering management*).
- Процессы программной инженерии (*Software engineering process*).
- Инструменты и методы (*Software engineering models and methods*).
- Качество программного обеспечения (*Software quality*).

При этом отдельно SWEBOK рассматривает также несколько смежных дисциплин, понимание и знание которых с точки зрения авторов крайне важно для успешного управления процессом создания системы.

- Системная инженерия (*Software engineering professional practice*).
- Экономика систем (*Software engineering economics*).
- Основы информатики (*Computing foundations*).
- Основы математики (*Mathematical foundations*).
- Основы инжиниринга систем (*Engineering foundations*).

Еще одной отличительной чертой SWEBOK является охват данным руководством множества внешних источников (для всех активностей по каждой области знаний в конце главы приводятся источники дополнительной информации), а также международных стандартов и методологий (обзор которых содержится в приложении). Ведь несмотря на все существующее многообразие моделей процессов и вариантов активностей по созданию и эксплуатации систем, все они являются в определенной степени взаимодополняющими. А изучение руководства SWEBOK дает необходимые знания об инженерии и способствует формированию понимания, что именно может быть сделано, для дальнейшего выбора одной / нескольких методологий разработки / внедрения / проектного менеджмента.

ПРИМЕР

В качестве примера рассмотрим четвертую главу SWEBOK «Тестирование ПО». В ней рассматриваются не просто процессы или активности по формированию технического проекта, но и все смежные вопросы: от основной терминологии до деталей существующих техник и методик.

Тестирование программного обеспечения (Software Testing).

1. Основы тестирования ПО (Software Testing Fundamentals).

1.1. Терминология тестирования (Testing-Related Terminology).

1.2. Ключевые аспекты (Key Issues).

1.3. Связь тестирования с другими активностями (Relationshops of testing with other activities).

2. Уровни тестирования (Test Levels).

2.1. Объекты тестирования (The target of the test).
2.2. Цели тестирования (Objectivies of Testing).
3. Техники тестирования (Test Techniques).
3.1. Техники, базирующиеся на интуиции и опыте разработчика (Based on the software engineer's intuition and experience).
3.2. Техники, базирующиеся на исходных данных (Input-based techniques).
3.3. Техники, базирующиеся на программном коде (Code-based techniques).
3.4. Тестирование, базирующееся на выявлении дефектов (Fault-based techniques).
3.5. Техники, базирующиеся на условиях использования (Usage-based techniques).
3.6. Техники, базирующиеся на моделях (Model-based techniques).
3.7. Техники, базирующиеся на структуре приложения (Techniques based on the nature of the application).
3.8. Выбор и комбинация различных техник (Selecting and combining techniques).
4. Измерение результатов тестов (Test-related measures).
4.1. Оценка программы в процессе тестирования (Evaluation of the program under test, IEEE 982.1-98).
4.2. Оценка проведенных тестов (Evaluation of the tests performed).
5. Процесс тестирования (Test Process).
5.1. Практические соображения (Practical considerations).
5.2. Активности в рамках тестирования (Test Activities).

Таким образом, SWEBOK, изначально разработанный в рамках совместного проекта IEEE и ассоциации ACM при поддержке крупнейших компаний (SAP, Boeing, Rational Software), остается важнейшим источником систематизированной из многих первоисточников информации об активностях программной инженерии.

3.4.1.2. PMBOK

Вряд ли есть менеджеры, не слышавшие о PMBoK. Свод знаний по управлению проектами (Project Management Body of Knowledge) интегрирует в себе множество аспектов, связанных с проектным управлением, основываясь на лучших мировых практиках.

Различные стандарты и методологии предлагают разные определения проекта:

> **Проект** – предпринимаемое усилие, организующее человеческие, материальные и финансовые ресурсы в неизвестный путь в рамках уникального предмета работы, заданной спецификации, с ограничениями на затраты и время, с тем, чтобы следование стандартному жизненному циклу проекта приводило к осуществлению успешных изменений, определенных посредством количественных и качественных целей и задач.
> *IPMA Competence Baseline. Version 2.0.*

или

Проект – уникальный процесс, состоящий из набора взаимоувязанных и контролируемых работ с датами начала и окончания и предпринятый, чтобы достичь цели соответствия конкретным требованиям, включая ограничения по времени, затратам и ресурсам.
ISO/TR 10006 Guidelines to quality in Project Management.

Именно данные идеи и реализует PMBoK, рассматривая аспекты времени, затрат и сроков особенно детально. Важно, что данные ограничения взаимозависимы, а значит, сжатые сроки приводят к увеличению стоимости проекта, увеличение состава работ – также к увеличению стоимости и/или времени выполнения проекта. И грамотное управление данными тремя рычагами дает возможность менеджеру принимать проактивные действия.

Начинается PMBoK с описания общего жизненного цикла управления проектом и его основных групп процессов:

- Инициация.
- Планирование.
- Исполнение.

ПРИМЕР

PMBoK рассматривает следующие процессы исполнения:

- *Руководство и управление исполнением процесса.*
- *Подтверждение качества.*
- *Набор команды проекта.*
- *Развитие команды проекта.*
- *Управление команды проекта.*
- *Распространение информации.*
- *Управление ожиданиями заинтересованных сторон.*
- *Осуществление закупок*

- Мониторинг и управление.
- Завершение проекта.

При дальнейшей детализации могут становиться основой любых отраслевых стандартов и в целом являются достаточно универсальными.

Более интересны выделяемые PMBoK области знаний:

- управление интеграцией;
- управление содержанием;
- управление сроками;
- управление стоимостью;
- управление качеством;
- управление человеческими ресурсами;
- управление коммуникациями;
- управление рисками проекта;
- управление поставками;
- управление стейкхолдерами (область знаний, появившаяся только в 5-й версии PMBoK, опубликованной в 2013 году).

Этим областям знаний посвящены соответствующие разделы и для каждой из них предлагается ряд активностей, методик, инструментов и других видов знаний, зарекомендовавших себя на практике.

Рассмотрим пример структуры и содержания PMBoK на примере области знаний «Управление содержанием». По структуре остальные главы PMBoK аналогичны.

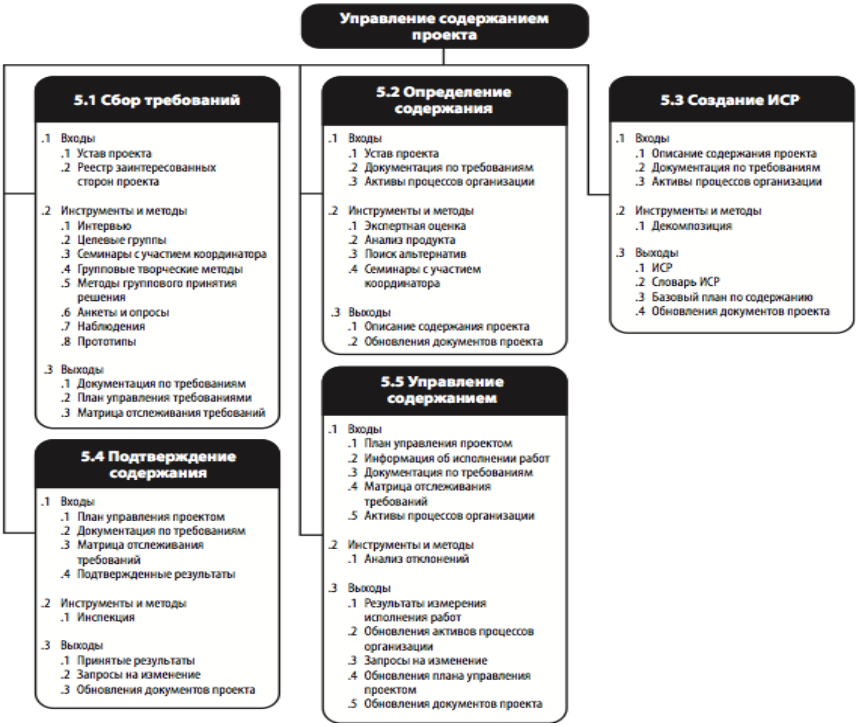
ПРИМЕР

1. В первую очередь, PMBoK дает ***определение области знаний****. «Управление содержанием проекта» включает в себя процессы, обеспечивающие включение в проект тех и только тех работ, которые необходимы для успешного завершения проекта. Управление содержанием проекта непосредственно связано с определением и контролем того, что включено и что не включено в проект» (PMBoK v4).*

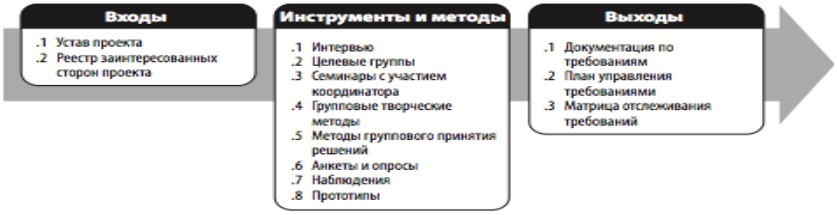
2. Приводится ***общая схема области знаний****.*

*На ней указываются все основные процессы, а также относящиеся к каждому из них «****Входы****», «****Инструменты и методы****», а также «****Выходы****».*

«Входы» обозначают необходимую для исполнения процесса информацию (например, «Реестр заинтересованных сторон», на основании которого определяются основные лица, для которых организуется сбор требований).



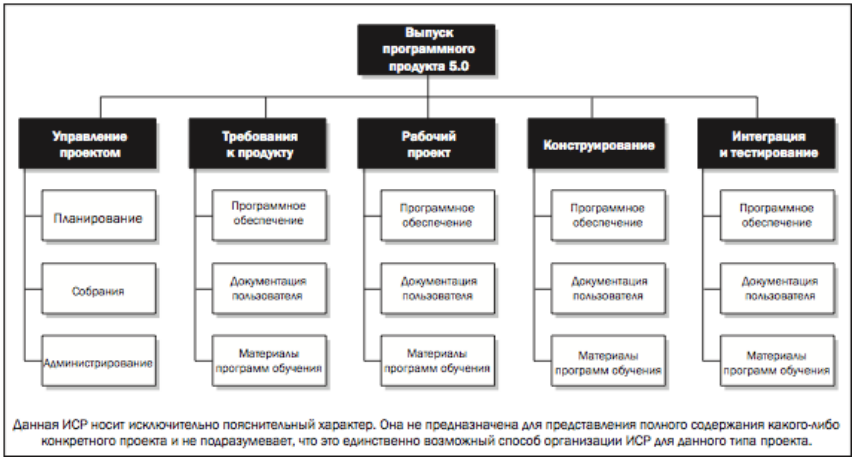
«Инструменты и методы» – основные активности и методы их осуществления (семинары, анкеты, групповые творческие методы», при помощи которых определяются содержание и иерархическая структура работ).
«Выходы» – ключевые результаты процесса (например, «документация по требованиям» и «матрица отслеживания требований», сформированная на основании осуществленных в ходе процесса активностей).
Всего в данной области знаний выделяется пять групп активностей: «Сбор требований», «Определение содержания», «Содержание иерархической структуры работ», «Подтверждение содержания», «Управление содержанием».
3. Рассмотрим один из ***процессов*** *выбранной области знаний подробнее.*



На этапе сбора требований в числе прочих предлагаются ***групповые творческие методы*** *(пп. 4): мозговой штурм, метод номинальных групп (мозговой штурм голосование), метод Дельфи (ответы на анкету группы экспертов в течение нескольких раундов), интеллект-карты (выявление сходств и различий в понимании среди идей мозгового штурма), диаграммы сходства (группировка идей). Для каждого из этих методов приводятся формулы и инструкции по применению.*

PMBoK приводит также подробные описания процессов и формируемых в процессе работы с ними документов / результатов.

ПРИМЕР
Образец организованной по фазам иерархической структуры работ.



Каждая вновь издаваемая версия PMBoK учитывает наиболее важные тенденции опыта управления проектами. Пятая версия (изданная в 2013 году) также содержит несколько нововведений: например, к ним относятся четыре новых процесса планирования (содержанием, расписанием, стоимостью и заинтересованными сторонами), которые развивают концепцию формирования плана управления проектом из нескольких вспомогательных планов. Также в ней произведены изменения содержания некоторых групп процессов и даже добавлена одна новая область знаний («управление заинтересованными лицами»), по сути частично основанная на управлении коммуникациями (концентрирующейся теперь исключительно на взаимодействии внутри проектной команды).

Однако несмотря на всю свою распространенность, PMBoK не является единственной доступной методологией и для предметной области ИТ-проектов применяются также PRINCE2, MSF, RUP и многие другие. Они могут оперировать такими параметрами, как число стадий и областей знаний, роли и процедуры управления в проектах, варьируя их в различной степени.

ПРИМЕР

В PMBoK критериями успешности можно считать соответствие треугольнику ограничений – составу работ, срокам, стоимости, а также качеству. PRINCE2 предлагает для оценки проекта фокусироваться на аспектах финансовой успешности, положительном мнении конечного пользователя и косвенной пользе для исполнителя (три перспективы: заказчик, исполнитель и пользователь). В то же время SCRUM, будучи еще более ориентированным на заказчика, рассматривает единственный фактор: удовлетворенность клиента полученными результатами (проект успешен, если заказчик готов продолжать сотрудничество с исполнителем и / или рекомендовать его другим).

Переходя от эталонных рекомендаций PMBoK к более практической стороне управления проектами внедрения, отметим ряд особенностей/ принципов.

– **Этапность.**

Масштабные проекты разбивается на этапы (подэтапы), продолжительность выполнения которых не должна превышать 2-3 месяцев и суть которых не обязательно должно однозначно соответствовать фазам жизненного цикла ИС. Более важно, что каждый этап работ должен иметь законченный результат, полезный для предприятия, независимо от хода и результата выполнения всего проекта. На каждом этапе проводятся работы, необходимые, в том числе, для реализации последующих этапов, однако некоторые из них должны выполняться строго последовательно, в то время как другие возможно реализовывать параллельно. Процесс планирования возможно осуществлять встроенными средствами MS Project либо любыми другими инструментами автоматизированного управления проектами и потоками работ. Это важно для поддержки таких активностей, как оперативный контроль

хода выполнения проекта, оценка потребных ресурсов (человеческих, материальных, временных и т.д.).

– **Результативность.**

Каждый этап проекта характеризуется наличием ясно определяемых и оцениваемых результатов. Критерии оценки результатов определяются при планировании проекта. Если общий процесс внедрения занимает более полутора лет, проект планируется таким образом, чтобы первые результаты были получены уже через 5-7 месяцев после его начала. Это дает дополнительную мотивацию сотрудникам предприятия, которые видят конкретные результаты проекта. Внедрение выполняется помодульно и начинается с тех модулей, функционирование которых необходимо для внедрения других модулей (либо с тех, которые способны достаточно быстро принести реальную отдачу).

– **Постоянный контроль.**

Необходимо постоянно держать под контролем как сам процесс выполнения работ, так и достижение целей их выполнения. По окончании каждого этапа или подэтапа осуществляется анализ степени изменения внешних и внутренних факторов проекта и определяется необходимость корректировки дальнейших планов. Контроль проводится посредством составления и анализа Графика выполнения работ по внедрению системы, а также периодических отчетов должностных лиц проекта на каждом его этапе.

– **Сквозное документирование проекта.**

Все результаты выполняемых работ, а также возникающие в процессе работы дополнения, замечания, разногласия, предложения в обязательном порядке оформляются соответствующими документами, причем помимо их согласования / утверждения даже получение пакета документов должно фиксироваться подписью ответственного лица.

– **Активное вовлечение заказчика в проект внедрения.**

Как уже обсуждалось в предыдущем разделе, группа внедрения от заказчика должна формироваться из руководителей среднего звена и ключевых специалистов предприятия. Функции специалистов группы внедрения от заказчика и команды подрядчика определяются при планировании проекта и в дальнейшем могут корректироваться в начале работ по каждому этапу проекта.

Без активного участия специалистов заказчика в процессе внедрения успешная реализация проекта внедрения практически невозможна. Работа консультантов подрядчика и сотрудников заказчика в одной команде значительно повышает эффективность проекта внедрения системы. Для успеха проекта в целом группе внедрения необходимы общее видение проблемы, нацеленность на один и тот же положительный результат, заинтересованность каждого и группы в целом в достижении этого результата.

3.4.1.3. PRINCE2

***Методология управления программами / проектами в организации PRINCE2 (PR**ojects **IN C**ontrolled **E**nvironments) – методология, концентрирующаяся на организации, менеджменте и мониторинге хода ИТ-проектов с точки зрения продукта.*

Окружение методологии PRINCE2 состоит из трех основных типов элементов: 7 принципов, 7 тем (далее – элементов) и 7 процессов (для каждого из которых определяются входные и выходные объекты, цели и ключевые активности).

Принципы

PRINCE2, будучи процессно-ориентированной методологией проектного управления, базируется на семи основных принципах.

Постоянное бизнес-обоснование	Каждый проект должен быть целесообразен и обоснован с точки зрения соотношения затрат / выгод его реализации
Обучение на собственном опыте	Проектные команды должны непрерывно обучаться, анализировать опыт предыдущих проектов (основные проблемы, риски, преимущества) и делиться выводами
Определение ролей и ответственности	Организационная структура должна учитывать закрепление ответственности за членами проектной команды в части учета интересов всех сторон: бизнес-заказчиков, пользователей, поставщиков / потребителей
Управление по стадиям	Планирование, мониторинг и контроль проектов на каждом этапе
Управление исключениями	Для каждого проекта определены конкретные критические значения показателей и уровни / механизмы эскалации возникающих проблем
Фокус на продуктах	Определение требований к качеству результатов проекта и их характеристик
Адаптация к условиям конкретного проекта	Адаптация методологии в зависимости от масштабов / сложности / важности / окружения конкретного проекта

Роль проектной команды в данном случае может быть описана пятью ключевыми активностями:

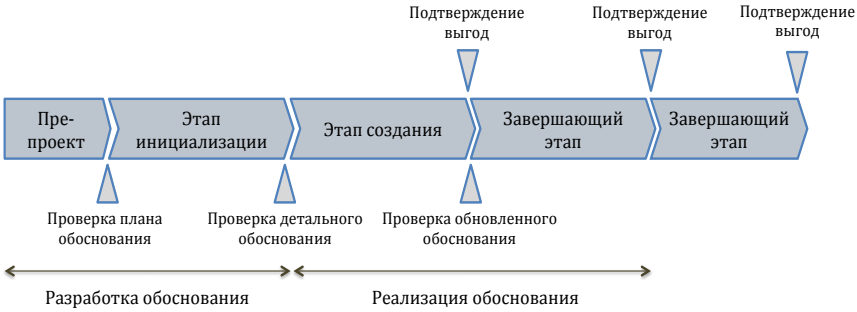
– Адаптация семи элементов PRINCE2 к условиям проекта.
– Применение / адаптация специфической терминологии.
– Адаптация описаний продукта к условиям проекта.
– Адаптация описаний ролей в соответствии с PRINCE2.
– Адаптация процессов для соответствия вышеописанным принципам.

Все эти активности и их особенности подробно описываются в тексте методологии.

Элементы

1. Бизнес-кейс (Обоснование проекта) = «Почему?».

Основная цель создания подобного бизнес-кейса состоит в создании механизмов оценки целесообразности (и возможности) реализации проекта.



2. Организация = «Кто?».

Определение и создание проектной структуры ответственности.

3. Качество = «Что?».

Определение методов создания качественного продукта (включая аспект контроля качества).

4. Риск = «Что если?».

Определение, оценка и контроль всех рисков.

5. Планы = «Как? Сколько? Когда?».

Определение способов создания / распространения продуктов, прогнозирование динамики спроса.

6. Изменения = «Какое влияние?».

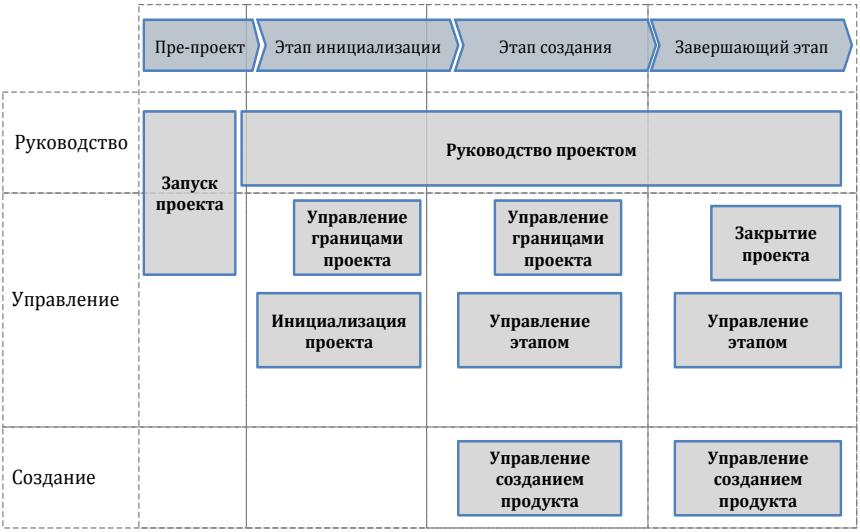
Определение эффекта от внедрения одобренных изменений проекта.

7. Прогресс = «Где мы сейчас? Куда направляемся?».

Создание механизмов мониторинга и оценка реальных достижений для планирования целей проекта, в том числе ответ на вопрос «Стоит ли продолжать двигаться в выбранном направлении?».

Процессы

И наконец, PRINCE2 предлагает собственную карту процессов управления проектом.



Рассмотрим каждый из процессов подробнее.

Запуск проекта	Обоснование основных причин осуществления проекта, его задач и ожидаемого результата, при этом запуск проекта должен осуществляться в очень короткие сроки
Навигация проекта	Координация проекта, определение ролей команды проекта, создание плана этапов
Инициация проекта	Обоснование необходимости проекта (например, технико-экономическое обоснование), планирование ресурсов
Управление границами этапа	Мониторинг выполнения планами/ результатами, контроль ключевых показателей проекта, предоставление информации спонсорам / заказчикам проекта (управление коммуникациями)
Контроль этапа	Мониторинг и контроль, осуществляющийся лидером проекта (определение подлежащих решению задач, сбора информации, предоставление отчетов, принятие необходимых корректирующих действий.
Управление созданием продукта	Обеспечение создания требуемых в ходе проекта продуктов
Завершение проекта	Обеспечение контролируемости стадии завершения проекта, контроль соответствия результатов документированным положениям при инициации проекта, подготовка финальных отчетов, обеспечение последующей поддержки и обучения

Таким образом, PRINCE2 является достаточно гибким руководством проектного управления, которое обеспечивает четкую структуру, предос-

тавляет общие методы и терминологию проектного управления. Методология PRINCE2 может быть адаптирована для любого типа проектов, индустрий. И наконец, возможность дополнительно получить сертификацию своих знаний по PRINCE2 делает ее популярной среди консультантов и компаний-подрядчиков.

3.4.1.4. ISO 21500:2012

Международный стандарт:

ISO 21500 Guidance on project management.

Российский аналог:

Международный стандарт по Управлению Проектами.

ISO 21500:2012 изначально создавался как стандарт, регламентирующий проектное управление на международном уровне. Разработка стандарта велась силами трех рабочих групп, ответственных за различные направления:

- Разработка терминологии.
- Регламентация процессов управления проектами.
- Определение и описание в приложениях ключевых концепций проектного менеджмента.

Один из самых «молодых» стандартов, он во многом схож с PMBOK, но уже за первые же годы использования стал достаточно популярен.

В ISO 21500:2012 приведены описания понятий и процессов, формирующих управление проектами, однако детального руководства не приводится. Сам документ предназначен в первую очередь для топ-менеджеров и спонсоров проекта, а также менеджеру проекта и проектной команды. Благодаря ISO 21500 они получают знания о принципах и практиках управления проектами, а также некое эталонное представление об актуальных практиках управления.

Структура ISO 21500 приведена ниже:

1. **Общие положения.**
2. **Термины и определения** (в т.ч. «операция», «запрос на изменения», «критический путь», «лаг», «лид», «кривая обучения», «словарь ИСР» и пр.).
3. **Концепция управления проектами.**

Например, общий обзор, среда проекта (в т.ч. программы и портфели проектов), внешнее руководство проекта, заинтересованные стороны, компетенции участников проекта, ограничения проекта.

ПРИМЕР

Схема заинтересованных сторон проекта.

4. **Процессы управления проектами.** Процессы управления проектами рассматриваются с двух ракурсов: с точки зрения управления проектом как групп процессов (применимых к конкретным фазам или проекту в целом), а также с точки зрения группировки процессов по предметным областям. **Предметные области** состоят из процессов, которые могут относиться к любой фазе проекта и не зависят от отрасли / специфики.

Соответствие процессов управления проектами группам процессов и предметным группам приведено в Приложении.

Схема взаимодействия группа процессов с основными входами и выходами приведена в Приложении.

ПРИМЕР

Рассмотрим описание одного из процессов на примере «Управления содержанием» для формирования представления о разнице стандарта с PMBOK. Одним из его подпроцессов является 4.3.13. «Определение работ». Приводится ***цель*** *(«Выявление, определение и документирование всех операций, которые должны быть запланированы и осуществлены в целях достижения целей проекта»), а также* ***таблица входов / выходов.***

Основные входы	*Основные выходы*
– *Иерархическая структура работ* – *Словарь иерархической структуры работ* – *Планы проекта* – *Утвержденные изменения*	– *Перечень работ*

5. Приложения. **Блок-схемы взаимодействия отдельных процессов** в каждой группе процессов.

3.4.1.5. ГОСТ Р 54869-2011

ГОСТ Р 54869-2011 Национальный стандарт «Проектный менеджмент. Требования к управлению проектом».

Стандарт ГОСТ 54869 принадлежит к серии стандартов, в которую также входят:

– ГОСТ Р 54871-2011 «Требования к управлению программой».
– ГОСТ Р 54870-2011 «Требования к управлению портфелем проектов».

Стандартом определяются следующие процессы управления проектом:

– процесс инициации проекта;
– процесс планирования управления изменениями в проекте;
– процесс планирования содержания проекта;

ПРИМЕР

Цель процесса: определение требований проекта и состава работ проекта.
Выходы процесса:

а) требования к проекту со стороны заказчика, других заинтересованных сторон проекта, а также законодательства и нормативных актов определены, проанализированы на предмет возможности их выполнения, согласованы с заказчиком проекта и документированы;

б) определены, согласованы с заказчиком и документированы ключевые данные по продукту проекта, а именно:

1. назначение, свойства и характеристики продукта;
2. критерии и методы приемки продукта проекта и его составных частей;
3. допущения и исключения, касающиеся продукта проекта;

в) определены, согласованы с заказчиком и документированы работы проекта, а также допущения и исключения, касающиеся работ проекта.

– процесс планирования расписания проекта;
– процесс планирования бюджета проекта;
– процесс планирования персонала проекта;
– процесс планирования закупок в проекте;
– процесс планирования реагирования на риски;
– процесс планирования обмена информацией в проекте;
– процесс организации исполнения проекта;
– процесс контроля исполнения;
– процесс завершения проекта.

В стандарте также приводится взаимосвязь основных элементов проектного управления.

ПРИМЕР

Соответствие процессов управления проектами группам процессов и предметным группам приведено в Приложении.

3.4.1.6. ISO 10006

Международный стандарт:

ISO 10006:2003 Quality management systems – Guidelines for quality management in projects.

Российский аналог:

ГОСТ Р ИСО 10006-2005 Системы менеджмента качества. Руководство по менеджменту качества при проектировании.

Данный стандарт применим к проектам любой сложности, масштаба, сроков, независимо от вида проектируемых к выпуску изделий (включая технические средства, программное обеспечение, услуги).

Структура международного стандарта (ISO 10006):

1. Область действия.
2. Нормативные ссылки.
3. Определения.
4. Характеристики проекта.
   - 4.1. Общая характеристика.
   - 4.2. Руководство проектом.
   - 4.3. Организация.
   - 4.4. Этапы проектирования и процессы проектирования.
5. Качество в процессе руководства проектом.
   - 5.1. Общие положения.
   - 5.2. Процесс стратегии проекта.
   - 5.3. Процессы управления взаимосвязью.
   - 5.4. Процессы, связанные с объектом деятельности.
   - 5.5. Процессы, связанные со временем.
   - 5.6. Процессы, связанные со стоимостью.
   - 5.7. Процессы, связанные с ресурсами.
   - 5.8. Процессы, связанные с персоналом.
   - 5.9. Процессы, определяющие связи.
   - 5.10. Процессы, связанные с риском.
   - 5.11. Процессы, связанные с закупками.
6. Обучение на основе разработанного проекта.
7. Приложения (в т.ч. «Использование оценки развития проекта для повышения качества»).

Далее для каждого из процессов приводятся описания и рекомендации, наиболее важные для компаний, являющихся поставщиками или потребителями услуг и продукции проектных организаций.

ПРИМЕР
5.11.4. Заключение субподрядных контрактов.
…
Оценка предложений должна быть основана не только на оценке стоимости продукции субподрядчиков, но она должна также учитывать другие затраты как то стоимость эксплуатации, обслуживания, лицензионные сборы, транспортные расходы, страхование, таможенные пошлины, изменения валютного курса, стоимость контроля, аудита качества и стоимость, связанную с отклонениями от требований.
Перед заключением контракта на поставку продукции должно быть надлежащим образом, рассмотрено влияние на качество каких-либо отклонений в условиях продажи от первоначальных требований проектной организации. Документы контракта должны быть проверены, чтобы гарантировать, что они содержат результаты всех предконтрактных согласований с субподрядчиком.

3.4.2. Менеджмент ЖЦ ИС в контексте проектной деятельности

3.4.2.1. Управление стейкхолдерами

Конфликт интересов бизнеса и ИТ

Как уже неоднократно отмечалось, важным аспектом для внедрения и существования систем в частности – и деятельности организации в целом является согласованность действий ИТ-департамента с интересами бизнес-пользователей и конечно, руководства компании. Формирование бизнес-архитектуры, сбор требований, процесс внедрения – любые действия, относящиеся к ИТ, так или иначе затрагивают интересы других сторон. Ценность ИТ-проекта должна быть так или иначе «донесена» до каждой из них для понимания смысла сотрудничества и конечно, выделения необходимых инвестиций.

Посвятившие значительно количество времени и ресурсов исследованию подходов и практик определения ценности ИТ для бизнеса, создатели проекта it-value.ru выделили в ходе работы несколько основных стейкхолдеров и основные вопросы, актуальные для них [48].

Бизнес-заказчики	Представления о ценности ИТ и основные актуальные вопросы
Предприниматели	Как ИТ поможет мне создать новые конкурентные преимущества?
Инвесторы	Как и с каким процентом вернутся мои инвестиции. Сравнение эффекта от ИТ с другими инвестиционными проектами.
Топ-менеджеры	Как корпоративные ИТ помогают реализовать стратегические цели и планы бизнеса и снизить бизнес-риски?
Менеджеры среднего уровня	Ценность ИТ в сокращении затрат и уменьшении рисков операционной деятельности. Как обеспечить заданный уровень качества дешевле и с меньшими рисками?
Консультанты	Ценность ИТ не обсуждается – она очевидна. Ценность ИТ подтверждается практикой лидеров. Копируйте их – и ценность придет сама.

Таким образом, важен не просто учет наличия возможного конфликта интереса, но и понимание основных приоритетов и движущих сил каждой из сторон, что в дальнейшем можно использовать с выгодной стороны. Как уже неоднократно подчеркивалось, любое внедрение системы сопряжено с внесениями изменений в процессы текущей деятельности как минимум задействованных в проекте подразделений (как максимум-предприятия и принципов ее взаимодействия с контрагентами). Соответственно, среди всех сотрудников обязательно будут как активные сторонники, так и противники подобных перемен, и крайне важна корректная оценка потенциального влияния проекта на всех субъектов, а также их влияния непосредственно на проект (как положительного, так и отрицательного).

Организационные условия осуществления проекта

Необходимые условия отражают потребности проекта, отсутствие которых непосредственно влияет на сроки, стоимость и даже принципиальную возможность достижения результата проекта. Этими условиями являются:

- Сильная поддержка проекта со стороны Генерального директора (и других топ-менеджеров) предприятия заказчика.
- Осознание руководством предприятия крайней необходимости внедрения системы.
- Готовность руководства предприятия к четкой организации проекта обследования предприятия и реализации системы.
- Готовность руководства предприятия к выделению квалифицированных сотрудников для осуществления совместной работы по проекту со специалистами подрядчика.
- Готовность предприятия и его руководства к внедрению и проведению неизбежных изменений в управленческих процессах, корпоративных стандартах учета и отчетности, созданию у сотрудников предприятия твердого убеждения неизбежности нововведений, поддержанию высокого статуса проекта и закреплению всех проектных распоряжений соответствующими приказами руководства.
- Формирование совместной группы внедрения со стороны заказчика и подрядчика.
- Создание органа корпоративного управления заказчика и проведение его заседаний по утвержденному регламенту, но не реже одного раза в месяц.
- Назначение куратора проекта в ранге не ниже заместителя генерального директора компании заказчика, а также ответственных лиц по подразделениям предприятия для участия в проекте.
- Создание для группы внедрения необходимых условий для работы на предприятии заказчика (помещение, компьютеры для каждого, сеть, связь, доступ к ксероксу и принтеру), в том числе для работы в условиях вахтенного метода, т.е. возможно без выходных дней и в нерабочее время (после окончания рабочего дня в организации заказчика).

3.4.2.2. Управление человеческими ресурсами

При привлечении подрядчика к внедрению проектная команда будет состоять из специалистов с обеих сторон: как заказчика, так и исполнителя.

От заказчика	От исполнителя
Координатор проекта	Куратор проекта
Менеджер (руководитель) проекта	Менеджер (руководитель) проекта
	Администратор проекта
– Специалисты АСУ – Системный администратор – Системный архитектор – Консультанты-программисты	– Бизнес-аналитики – Консультанты по функциональности – Консультанты по постановке задач – Технические консультанты – Программисты-консультанты
	– Преподаватель курсов (проведение обучения по системе)

И кураторы, и проектные менеджеры, принимая самое непосредственное участие в принятии высокоуровневых решений по управлению проектом внедрения, как правило, являются членами Координационного совета или другого органа корпоративного управления ИТ со стороны Заказчика для определения стратегических, финансовых и организационных вопросов, а также согласования и утверждения результатов работы команды внедрения по проекту. Заседания подобного органа управления проводятся по мере необходимости, в зависимости от масштаба и сложности проекта (чаще всего на ежемесячной основе).

Далее опишем последовательно все основные проектные роли с обоих сторон.

Со стороны исполнителя

Куратор проекта – как правило, представитель топ-менеджмента компании, участвует в активностях по оформлению договора на выполнение проекта, осуществляет общее руководство проектом и определяет стратегические аспекты взаимодействия специалистов и компаний, занятых в нем.

Среди его основных задач в проекте:

- Участие в работе Координационного совета.
- Принятие оперативных административных решений, касающихся компании – исполнителя, контроль исполнения принятых решений.
- Согласование с Координатором проекта изменение рамок проекта, добиваясь при необходимости дополнительного финансирования (согласование изменения содержания / сроков / стоимости проекта).
- Принятие результатов этапов проекта и проекта в целом.
- Контроль выполнения принятых постановлений в проекте.
- Назначение ответственных исполнителей со стороны компании исполнителя.
- Приведение в жизнь административных решений Координационного совета, касающихся компании исполнителя.

ПРИМЕР

В части эскалации вопросов необходимо еще в момент старта проекта решить, в каких случаях группа внедрения сама может принимать решение (например, по поводу конфликта требований разных пользователей), и в каких необходима эскалация вопроса руководителем проекта для вынесения на заседанием Координационного совета (к примере, в части определения прав доступа различных подразделений или внешних / дочерних предприятий).

Руководитель проекта участвует во всех этапах проекта по внедрению, осуществляет руководство работами, определяет операционные аспекты взаимодействия специалистов компаний, занятых в нем. Как правило, занятость в проекте от 80 % до 100 % (неполная занятость и совмещение нескольких ролей и / или нескольких проектов одновременно существенно повышает риск неудачи проекта).

Среди его основных задач в проекте:

- управление проектом;
- ведение проекта, составление и исполнение бюджета проекта;
- участие на уровне Координационного совета в принятии стратегических решений по ходу выполнения проекта;
- представление Координационному совету результатов работы группы внедрения;
- мониторинг хода проекта / ведение отчетности и в случае необходимости вынесение на рассмотрение Куратором проекта (Координационный совет) решения об изменении объема / сроков / стоимости работ;
- разработка сценариев и планов проекта, процедур принятия решений;
- поддержание мотивации команды;
- осуществление контроля качества ведения проекта;
- обеспечения соответствия всех аспектов реализации проекта условиям заключенного с заказчиком договора;
- идентификация рисков и проблем процесса внедрения.

Руководителем группы внедрения является руководитель проекта от исполнителя, и на время выполнения работ по проекту все консультанты-программисты заказчика, входящие в группу внедрения, также работают под функциональным руководством руководителя проекта от исполнителя. Любые вопросы планирования работ, отчетности и представления к премированию решаются в первую очередь им. Однако данные полномочия могут быть в зависимости от ситуации им делегированы менеджеру проекта со стороны заказчика.

Перейдем к рассмотрению ролей проектной команды со стороны **исполнителя** (напомним, один человек может сочетать в себе несколько функций: и консультанта по функциональности, и консультанта по постановке задач):

Основной задачей **консультантов** со стороны исполнителя является внедрение системы и обеспечение ее сопровождаемости (для участия в дан-

ном процессе специалистов со стороны заказчика). Обладая знаниями в области функциональности системы и ее отдельных модулей, а также навыками проведения обучения пользователей, консультанты подрядчика участвуют во всех видах проектных активностей:

- Собирают требования и проектируют систему.
- Инсталлируют и настраивают программное решение на тестовом сервере.
- Консультируют и обучают сотрудников заказчика.
- Тестируют и совершенствуют систему совместно с заказчиков при развертывании и проведении опытно-промышленной эксплуатации.

Бизнес-аналитик выполняет работы по проведению обследования предприятия, формирует в случае необходимости рекомендации по оптимизации и реинжинирингу бизнес-процессов Заказчика и участвует в их исполнении, формирует рекомендации по информационному обеспечению бизнес-процессов и взаимодействию, а также принимает участие в формировании требований на создание системы (совместно с консультантом по постановке задач, если подобная роль отдельно выделяется).

Консультант по функциональности выполняет работы по настройке и доработке функциональных возможностей системы с учетом специфики бизнес-процессов предприятия Заказчика (на основе данных, предоставленных бизнес-аналитиком), а также принимает участие в процессе внедрения системы.

Технический консультант осуществляет техническую и технологическую поддержку проекта, совместно с **программистом-консультантом**, ответственным за программное обеспечение.

Преподаватель курсов по функциональности осуществляет обучение специалистов со стороны заказчиков, в дальнейшем играющих роль пользователей, описывая им интерфейс системы, ее ключевые возможности и настройки. Другой группой обучаемых сотрудников являются ИТ-специалисты заказчика, после внедрения осуществляющих текущую техническую и консультационную поддержку пользователей.

Со стороны заказчика

На стороне заказчика среди руководящих ролей выделяется роль **Координатора проекта**. Координатор проекта, будучи как и куратор от исполнителя, представителем топ-менеджмента компании (в ранге не ниже заместителя генерального директора предприятия заказчика), участвует в активностях по принятию оперативных административных решений и контролю их выполнения.

Иногда эта роль называется куратор проекта от заказчика.

Также он:

- руководит работой Координационного совета;

- созывает заседания Координационного совета и определяет их повестку;
- утверждает изменение рамок проекта, обеспечивая при необходимости дополнительное финансирование;
- принимает результаты этапов проекта и проекта в целом;
- назначает ответственных исполнителей со стороны заказчика;
- отвечает за согласование действий различных подразделений и специалистов участвующих в проекте сторон;
- проводит в жизнь административные решения Координационного совета, касающиеся подразделений заказчика, принимает решение об их премировании.

В случае, когда система внедряется для предприятия, имеющего несколько филиалов, либо на нескольких предприятиях (одновременно или последовательно), то рекомендуется для каждого предприятия / филиала создавать свою группу внедрения со своим собственным руководителем. При этом желательно, чтобы Координационный совет был общим, и именно координатор проекта будет в таком случае осуществлять руководство всем проектном внедрения.

Руководитель проекта со стороны заказчика выполняет те же функции, что и руководитель проекта от исполнителя и фактически является его заместителем (во время отсутствия последнего на предприятии).

Оба руководителя проекта – как от заказчика, так и от исполнителя – имеют достаточно четко определенные зоны ответственности. Однако весь процесс до проведения тестовой эксплуатации наиболее критична роль руководителя проекта от исполнителя как координатора всех ключевых активностей. На время тестовой эксплуатации данная роль может переходить к руководителю проекта от заказчика как лицу более беспристрастному, у которого нет подобной непосредственной заинтересованности в положительных результатах тестирования. Тем не менее, на практике координирует работы и управляет процессом создания и внедрения системы тот менеджер проекта, который более деятелен, активен и вовлечен в проект.

Кроме того, руководитель проекта со стороны заказчика несет ответственность за:

- Распоряжается выделенными на проект фондами (в рамках бюджета проекта!).
- Обеспечение обязательного присутствия требуемых специалистов предприятия заказчика на рабочих местах во время реализации проекта.
- Создание сплоченной, работоспособной группы внедрения и укрепление у нее уверенности в конечном успехе проекта (привлечение всех членов группы к обсуждению текущих вопросов, периодическая замена людей, которые возглавляют обсуждение).
- Устранение препятствий на пути к повышению эффективности функционирования группы (выделение помещения под обучение и об-

суждение вопросов по проекту или привлечение дополнительных ресурсов для облегчения работы членов группы внедрения).
- Выносит возникшие проблемы на Координационный совет и контролирует выполнение принятых соответствующих постановлений.
- Своевременное представление проблем группы внедрения перед вышестоящим руководством и руководителями других подразделений.
- Оказание помощи в идентификации требований и проблем процесса внедрения.
- Обеспечение наличия и работоспособности компьютеров, принтеров, сети и другого необходимого на проекте оборудования.
- Установку и настройку требуемого общесистемного и специального программного обеспечения на рабочих местах специалистов группы внедрения и пользователей заказчика.

В качестве руководителя проекта от заказчика должен выступать всесторонне развитый человек. Он должен обладать достаточными знаниями о методах ведения бизнеса на предприятии, способностью организовывать, обучать людей и преподносить им целостное видение функционирования предприятия.

В идеальной ситуации руководитель проекта от заказчика как функциональный руководитель всех работ должен 100 % своего рабочего времени посвящать проекту. Совмещение этой деятельности с выполнением других обязанностей в значительной степени повышает риск неудачи проекта.

Руководитель проекта со стороны заказчика также должен обеспечить команде исполнителя выделение необходимого оборудования и инфраструктурных условий (включая установку требуемого общесистемного и специального ПО на рабочих местах группы внедрения). Как правило, ему в этом помогает **Администратор проекта** (выделяющийся либо от одной из сторон, занятых в проекте, либо от нескольких или всех участников). Он осуществляет корректное своевременное документирование всех этапов проекта (включая написание протоколов встреч), согласование организационно-финансовых и договорных вопросов, а также организационное сопровождение проекта (заказ переговорных комнат, необходимого оборудования).

Группа внедрения со стороны заказчика включает как ИТ-специалистов, так и сотрудников задействованных во внедрении и работе с системой бизнес-подразделений. **ИТ-специалисты** заказчика являются сотрудниками отдела автоматизации / ИТ-департамента (чаще всего программистами) и принимают активное участие во всех работах по внедрению системы (особенно на этапах создания проектного решения, доводки модулей системы в ходе тестирования и опытно-промышленной эксплуатации, формирования системной документации).

При планировании проекта необходимо оценить усилия, которые необходимо приложить для того, чтобы все члены группы внедрения имели дос-

таточно времени на работу в рамках проекта, а также определить сферы ответственности. Это может потребовать выделения дополнительных полномочий, ресурсов, временной перемены сферы ответственности.

Функции ИТ-специалистов (консультантов-программистов) заказчика:

- содействие консультантам исполнителя по вопросам внедрения и сопровождения системы;
- сотрудничество с командой от исполнителя в части предоставления информации о системах заказчика и специфике предприятия (сбора недостающей информации);
- обучение специфике работы с системой и ее настройке с дальнейшей передачей необходимых знаний ключевым сотрудникам предприятия и консультированием пользователей;
- формирование инструкций пользователей и администратора системы (баз данных) с последующим их обновлением по мере необходимости;
- обеспечение наполнения системы необходимыми выверенными данными (заполнение справочников, подготовка и ввод первичной информации, начальных остатков и т.п.) через личное участие (обучение) и через осуществление контроля над наполнением системы силами других ответственных специалистов от заказчика.

Консультантам-программистам от заказчика крайне важно перенимать навыки и знания о системе от консультантов исполнителя, так как после окончания контрактов на внедрение / сопровождение именно они будут заниматься поддержкой и развитием данной системы.

ICB IBMA – требования к компетенциям проектных команд

Основные компетенции и рекомендации по формированию проектных команд описываются в Руководстве, опубликованном некоммерческой профессиональной ассоциацией по управлению проектами IPMA (International Project Management Association). Будучи созданной еще в 1965 году, сейчас IPMA объединяет 50 национальных ассоциаций по управлению проектами, и в России ее представляет национальная ассоциация СОВНЕТ. СОВНЕТ разработала также соответствующий стандарт для сертификации российских специалистов – «Основы профессиональных знаний и Национальные требования к компетентности специалистов по управлению проектами» (НТК).

Ключевым стандартом IPMA по управлению проектами является ICB – IPMA Competence Baseline – руководство, описывающее требования к компетенциям, необходимым менеджерам проектов и членам проектных команд для управления проектами / программами / портфелем проектов. Система оценки компетенций состоит из четырех уровней сертификации IPMA:

1. уровень A – Сертифицированный директор проектов;
2. уровень B – Сертифицированный старший менеджер проектов;
3. уровень C – Сертифицированный менеджер проектов;

4. уровень D – Сертифицированный специалист по управлению проектами.

Для каждого из уровней определяются:

– Входные требования.
– Основная компетенция.
– Дополнительные требования.

ПРИМЕР

Уровень B – Сертифицированный старший менеджер проектов.

– *Входные требования.*

Минимум 5 лет опыта управления проектами, из которых 3 года в статусе руководителя сложных комплексных проектов.

– *Основная компетенция.*

Способность руководить комплексными проектами.

– *Дополнительные требования.*

Ответственность за все элементы управления комплексным проектом.
Использование релевантных процессов, методов, техник и инструментов управления проектами.
Статус руководителя крупной проектной команды.

Структура документа ICB IPMA выглядит следующим образом:

– Глава 1. Введение (Introduction).
– Глава 2. Ключевые концепции (Key concepts).
– Глава 3. Сертификация (Certification).
– Глава 4. Описания элементов (Element descriptions).
  - Технические компетенции (Technical competence elements).
  - Поведенческие компетенции (Behavioural competence elements).
  - Контекстуальные компетенции (Contextual competence elements).
– Глава 5. Ссылки (References).
– Приложения (в т.ч. система оценок компетенций, taxonomy).

Технические компетенции	Поведенческие компетенции	Контекстуальные компетенции
1.1. Успешность управления проектом	2.1. Лидерство	3.1. Ориентация на проект
1.2. Заинтересованные стороны	2.2. Вовлечение и мотивация	3.2. Ориентация на программу
1.3. Требования и цели проекта	2.3. Самоконтроль	3.3. Ориентация на портфель проектов
1.4. Риски и возможности	2.4. Уверенность в себе	3.4. Осуществление проектов, программ и портфелей проектов
1.5. Качество	2.5. Разрядка	
1.6. Организационная структура проекта	2.6. Открытость	3.5. Родительская компания
1.7. Работа команды	2.7. Творчество	3.6. Бизнес и предпринимательская деятельность
1.8. Разрешение проблем	2.8. Ориентация на результат	
1.9. Структура проекта	2.9. Продуктивность	3.7. Системы, продукты и технологии
1.10. Замысел и итоговый продукт проекта	2.10. Согласование	3.8. Управление персоналом
1.11. Время и фазы жизненного цикла проекта	2.11. Переговоры	3.9. Здоровье, безопасность, охрана труда и окружающей среды
1.12. Ресурсы	2.12. Конфликты и кризисы	
1.13. Стоимость и финансы	2.13. Надежность	3.10. Финансы
1.14. Закупки и контракты	2.14. Понимание ценностей	3.11. Юридические аспекты проекта
1.15. Изменения	2.15. Этика	
1.16. Контроль и отчетность		
1.17. Информация и документация		
1.18. Коммуникации		
1.19. Инициация проекта		
1.20. Закрытие проекта		

Основной отличительной особенностью ICB является выделение 46 различных систематизированных «элементов компетентности» со взаимодействиями между ними. Для каждого из элементов определяются: название, описание содержания, основные шаги и критерии оценки опыта для сертификации.

3.4.2.3. Управление финансами

Совокупная стоимость владения (TCO)

Совокупная стоимость владения (Total Cost of Ownership, TCO) – суммарная величина прямых и косвенных затрат владельца системы на протяжении всего ее жизненного цикла.

Идея TCO была предложена еще в 1987 году компанией Gartner Group, и с тех пор создано множество различных инструментов / методов определения значения стоимости владения системой.

Однако единой методики или формулы расчета до сих пор не существует, так как структура осуществляемых затрат существенно различается в зависимости от специфики предприятия, проекта в целом и объектов расчета (оборудование, ПО, консалтинговые услуги, …) в частности.

TCO учитывает три основных типа затрат:

– Прямые капитальные (CAPEX).
– Прямые операционные (OPEX).
– Косвенные затраты (потенциальные потери от различных плановых и внештатных ситуаций).

Отметим, что по сути косвенные затраты отражают то, насколько эффективно осуществляется управление информационными системами. Так, чем более эффективно ИТ-служба обеспечивает управление ИТ-инфраструктурой и системами, тем меньше косвенные затраты от простоя систем, потери времени конечных пользователей на самостоятельный поиск решения и ожидание помощи.

Однако один расчет совокупной стоимости владения без сравнения с прочими параметрами не дает полного представления о целесообразности эксплуатации системы. Логическая цепочка в данном случае выглядит следующим образом: чем большее число пользователей работают в единой системе и чем сложнее ее бизнес-процессы, тем выше как совокупная стоимость владения, так и результативность использования системы.

ПРИМЕР

При внедрении различных технологий могут учитываться следующие типы затрат:

– *Капитальные затраты.*

- *Сетевое программное и аппаратное обеспечение.*
- *Серверное программное и аппаратное обеспечение.*
- *Клиентское программное и аппаратное обеспечение.*
- *Стоимость исследований и разработки технологий.*
- *Стоимость лицензий (первоначальные инвестиции).*
- *Стоимость продления лицензий.*
- *Затраты на миграцию.*
- *Амортизация оборудования.*
- *…*

– *Операционные затраты.*
  - *Аренда серверов / ЦОД.*
  - *Затраты на тестирование.*
  - *Стоимость резервирования / копирования данных.*
  - *Обучение.*
  - *Внутренний и внешний аудит / консалтинговые услуги.*
  - *Оплата труда персонала.*
  - *Администрирование инфраструктуры.*
  - *Ввод оборудования в эксплуатацию.*
  - *…*

– *Косвенные затраты.*
  - *Потенциальные потери от плановых и внеплановых сбоев программно-аппаратных платформ.*
  - *Потери при нарушении конфиденциальности данных/при инцидентах информационной безопасности, а также затраты на восстановление деловой репутации.*
  - *Потери времени сотрудников на помощь коллегам в решении вопросов поддержки информационных систем и пр.*
  - *Снижение производительности (пользователи вынуждены ждать реакции системы, например, печать отчетов, что снижает эффективность их работы и приводит к денежным потерям организации).*
  - *…*

Возможно учитывать также затраты на модернизацию и утилизацию систем (например, масштабирование, замену программно-аппаратной платформы, увеличение производительности).

Выгода от реализации проекта таким образом может рассчитываться как разница между затратами до проекта и после его завершения.

ПРИМЕР

*Для проекта **виртуализации ИТ-инфраструктуры** до и после предполагаемой его реализации могут быть рассчитаны значения таких параметров, как:*

– *Администрирование серверов.*
– *Восстановление серверов в случае чрезвычайной ситуации.*
– *Затраты на незапланированный простой сервера.*
– *Охлаждение и электропитание.*
– *Подключение к сети.*

– *Лицензии на VMware.*
– *Проектирование / развертывание виртуальной ИТ-инфраструктуры.*
– *Консалтинговые услуги по внедрению и обучению специалистов.*
– *…*

Часть данных затрат будет более высока при реализации проекта (например, при необходимости закупки лицензий), но значительная часть снизится в случае его реализации (стоимость аппаратного обеспечения, администрирования и пр.).

В целях количественного анализа используются и специальные программные средства, которые в основном реализуются на уровне электронных таблиц. При этом пользователи получают возможность сравнивать итоговые результаты с референтными значениями других компаний в отрасли и сходных масштабов проектов / систем для проведения анализа. Некоторые программные решения также позволяют осуществлять What-If анализ для моделирования динамики изменения ТСО при принятии того или иного решения в области ИТ (например, при консолидации серверов).

Чистая приведенная стоимость проекта (NPV)

Чистая приведенная стоимость проекта (Net Present Value, NPV) – Текущая стоимость будущих денежных потоков инвестиционного проекта.

Смысл данного показателя состоит в вычислении текущей стоимости потоков денежных средств по периодам с последующим суммированием результатов. Риски проекта учитываются при использовании различных ставок дисконтирования (чем выше риск, тем выше будет выбранная для него ставка дисконтирования, которая может учитывать также и инфляцию / прогнозную норму прибыли).

Всего для вычисления NPV применяются три основных шага.

1. Определение текущей стоимости затрат (объема инвестиций), через суммирование различных типов затрат.
2. Расчет текущей стоимости денежных потоков (PV) к текущей дате:

$$PV = \sum_{t=0}^{n} \frac{CFt}{(1+r)^t}$$

$$NPV = \sum_{t=0}^{n} \frac{CFt}{(1+r)^t} - I$$

где CF – денежный поток;

r – ставка дисконтирования;

I_t – инвестиции периода.

Либо в случае осуществления инвестиций в несколько этапов применяется следующая формула:

$$NPV = \sum_{t=1}^{n} CFt - \sum_{t=0}^{n} \frac{It}{(1+r)^t}$$

где *CF* – денежный поток;

r – ставка дисконтирования;

I_t – инвестиции периода *t*;

n – общее число периодов.

3. Сравнение текущей стоимости инвестиций в проект с текущей стоимостью доходов. Полученная разница и представляет собой значение чистого дисконтированного дохода.

Формула:

$$NPV = PV - I_0$$

NPV > 0	Реализация предложенного проекта экономически эффективна, так как потенциально он принесет больше, чем требуемый процент возврата инвестиций.
NPV = 0	Проект выгоден, так как принесет прибыль, эквивалентную требуемому проценту возврата инвестиций.
NPV < 0	Реализация проекта экономически нецелесообразна.

ПРИМЕР

Рассчитаем чистую приведенную стоимость проекта по консолидации серверов.

Год (t)	***Коэффициент дисконтирования (при ставке 10 %) = 1 / (1 + 0.1)^t***	***Поступления денежных средств (PV)***	***Инвестиции (I_t)***	***Чистый денежный поток (CF_t)***	***NPV_t***
0	*1.000*	*0*	*500 000*	*-500 000*	*-500 000*
1	*0.909*	*1 000 000*	*0*	*+1 000 000*	*909 091*
2	*0.826*	*750 000*	*0*	*750 000*	*619 835*
3	*0.751*	*500 000*	*0*	*500 000*	*375 657*
Итого		***2 250 000***	***500 000***		***1 404 583***

Как можно видеть из вышеприведенных вычислений, несмотря на то, что денежные поступления при реализации проекта будут превышать инвестиции на 1,75 млн., чистая приведенная стоимость будет меньше, порядка 1,4 млн. с учетом текущей ставки дисконтирования, применяемой для каждого периода.

В MS Excel для расчета NPV используется функция: =ЧПС(). В английской версии: =NPV()

Синтаксис: =NPV(rate;value1;[value2]...)

rate – ставка дисконтирования, выраженная в единицах за период. Так, если представлены годовые потоки денежных средств, то ставка должна быть годовая.

value1;[value2]... – представленные потоки денежных средств по периодам фиксированной длины, которые могут быть представлены как диапазоны, конкретные значения через точку запятой или несколько диапазонов.

Возврат инвестиций (ROI)

Возврат инвестиций (Return On Investment, ROI) – отношение полученных (сэкономленных) денег от реализации проекта к вложенным в него средствам. ROI = 180 % означает, что на каждый вложенный рубль вернулась (или была сэкономлена) сумма в 1 рубль 80 копеек.

Смысл показателя ROI в определении чистой прибыли от инвестиций, которая необходима для получения прибыли. Он также может говорить о том, насколько эффективно используются активы.

В MS Excel для расчета IRR используется функция: =ВСД(). В английской версии: =IRR()

Синтаксис: =IRR(values;[guess])

values – диапазон с потоками денежных средств.

guess – необязательное поле, используемое для определения ставки методом подбора. Предположение, о том, какая будет величина ставки, необходимо для начала отсчета именно от этого значения.

ПРИМЕР

Для приведенного ранее примера расчета NPV и значений денежного потока сумма CF по всем периодам составит 1,75 млн, а значит при исходном значении инвестиций в 500 тыс. коэффициент возврата инвестиций будет равен 250 %.

Чистый денежный поток (CF)
-500 000
+1 000 000
750 000
500 000

Период окупаемости инвестиций (Payback Period)

Период окупаемости инвестиций (Payback Period, PP) – Продолжительность времени, в течение которого обеспечивается возврат первоначальных инвестиций.

Как правило, инвесторы склонны более благоприятно относиться к проектам с более быстрым оборотом средств, нежели преследовать долгосрочные выгоды, которые ассоциированы с большими рисками.

Для определения, в какой момент проект достигнет точки окупаемости, необходимо последовательно рассчитывать показатель NPV для каждого периода проекта, пока значение NPV

Этот показатель определяется последовательным расчетом NPV для каждого периода проекта, точка, в которой NPV станет положительным, будет являться точкой окупаемости.

ПРИМЕР

Год (t)	***NPV***
0	*-500 000*
1	*909 091*
2	*619 835*
3	*375 657*
Итого	***1 404 583***

Так, для рассмотренного при расчете NPV примера точка окупаемости будет достигнута уже к первому году, кода NPV станет больше 0.

3.4.2.4. Управление коммуникациями

Управление реализацией проекта
и решение спорных вопросов / проблем

Правильное и непрерывное управление проектом является залогом его успешного осуществления и завершения. Появление промежуточных результатов в ходе реализации проекта позволяет начать их использование еще до завершения проекта в целом. Например, завершение этапа по автоматизации учета основных средств предприятия позволяет, не дожидаясь окончания проекта, осуществлять расчет амортизации.

Для решения возникающих в ходе проекта спорных вопросов и проблем применяется процедура управления решением спорных вопросов, призванная обеспечить качественное выполнение задач проекта в срок и снизить степень влияния связанных с этими вопросами факторов риска.

Решение каждого вопроса (проблемы) проходит четыре стадии:

- идентификация проблемы;
- принятие к дальнейшему рассмотрению либо отказ от такового;
- одобрение предложенного решения;
- закрытие проблемы.

Проблема может быть идентифицирована членами команды внедрения, пользователями или управленческим персоналом. В момент регистрации каждый спорный вопрос должен быть задокументирован с помощью Формы регистрации проблем или аналогичного документа. Для принятия решения о целесообразности документирования проблемы используется один из приведенных ниже критериев:

- На сколько существенно повлияет отказ от решения этой проблемы на проект (например, вызвать задержку, изменение курса проекта, снижение качества или увеличение стоимости проекта)?
- Выходит ли проблема за рамки одной задачи или требует внимания более чем одного консультанта?
- Требует ли проблема решения со стороны куратора или руководителя проекта?

Проблемы должны оцениваться по степени своего влияния на проект по следующим категориям важности:

1. Высокая (такие проблемы должны быть рассмотрены руководителем проекта и утверждены Координационным советом).

ПРИМЕР

Изменение порядка лимитирования накладных расходов относительно порядка, описанного в проектном решении.

2. Средняя (должны быть рассмотрены и решены руководителем проекта совместно с кем-либо из заместителей директоров предприятия заказчика. Обо всех принятых таким образом решениях руководитель проекта отчитывается на очередном заседании Координационного совета).

ПРИМЕР

Необходимость перераспределения должностных обязанностей между двумя или несколькими подразделениями, работающими в системе.

3. Низкая (рассмотрение и решение таких проблем осуществляются руководителем проекта самостоятельно. Обычно проблемы такого рода не имеют альтернативных решений. Суть такой проблемы обычно заключается в необходимости принять конкретное заранее известное решение).

ПРИМЕР

Необходимо организовать своевременность ввода первичных документов на складе N.

Каждая проблема должна быть рассмотрена руководителем проекта со стороны исполнителя, в результате чего ей должен быть назначен один из следующих статусов:

- Отклонена – проблема отклоняется, если она не препятствует нормальному ходу выполнения проекта и не влияет на его конечный результат. Перед тем как отклонить проблему, необходимо обсудить это решение с лицом, идентифицировавшим эту проблему. Это позволит инициировавшему лицу еще раз сформулировать суть проблемы, и, возможно, более удачно передать ее важные аспекты.
- Отложена – проблема откладывается, если решение по этой проблеме не может быть принято в момент рассмотрения. В этом случае обязательно должна быть указана дата, до которой откладывается решение проблемы.
- Объединена – проблемы, связанные тесно между собой, должны быть объединены. При объединении одна из проблем должна быть закрыта, а описание другой проблемы должно быть изменено с учетом произошедшего объединения.
- Принята – для принятых проблем производится рассмотрение и оценка вариантов решения, выбор варианта и назначение исполнителя для его реализации.

Как только по какой-либо проблеме принято решение, оно должно быть преобразовано в последовательность шагов по его реализации, в числе которых должно быть определено:

- Какие задачи должны быть выполнены.
- Какие документы, отчеты или другие рабочие продукты должны быть подготовлены.
- Кто будет отвечать за выполнение задач и за техническую подготовку.
- Критерии качества и завершения.

Как правило, списком проблем управления проектом занимается проектный менеджер и затем последовательно передается на рассмотрение куратору и Координационному совету (с занесением записей в проектную документацию – как правило, Протокол выполненных работ и Журнал регистрации проблем).

Способы предотвращения конфликтных ситуаций

Можно представить множество причин возникновения конфликтных ситуаций в процессе развития проекта. В данном разделе выделены основные источники и причины возникновения таких проблем, а также возможные пути их предотвращения.

1. **Цель проекта** в контексте решения критических вопросов предприятия должна быть поставлена перед группой внедрения достаточно четко. Высшее руководство предприятия должно заявить о полной поддержке людей, задействованных в проекте. В противном случае это приведет к снижению мотивации и личной заинтересованности и, в конечном итоге, существенно снизит вероятность успешного и эффективного завершения проекта.

2. Представители проектной команды должны **регулярно встречаться** друг с другом и с членами Координационного совета с целью обсуждения текущих вопросов, анализа состояния проекта и обмена мнениями. В противном случае может возникнуть ситуация, когда проект не будет расцениваться как самая приоритетная задача, а цели и задачи станут несколько размытыми. Созыв Координационного совета также позволяет решать спорные вопросы и подводить промежуточные итоги по выполненному объему работ.

Обсуждение способов решения конфликтных ситуаций должно проводиться в максимально полном составе проектной команды. Каждый ее представитель представляет отдельное подразделение и, следовательно, имеет свое видение проблем и методов их решения. Различия в подходах должны быть учтены и, как следствие, должны быть выработаны единые порядок и нормы взаимодействия.

3. Другой потенциальный источник конфликтных ситуаций – это **приоритеты управления и оценка производительности**. Типичная ситуация, когда член проектной команды напрямую отчитывается перед своим непо-

средственным руководителем и косвенно – перед руководителем команды внедрения. Требования, высказанные обоими руководителями, вступают в противоречие, и рядовой член команды внедрения находится в состоянии растерянности. Ситуация, когда человек задействован на двух работах, потенциально может вылиться в две проблемы:

- члены проектной команды не поощряются в материальном плане за совмещение двух работ, и
- их вклад в общее дело остается недооцененным.

4. Исторически сложилось, что **оценка действий сотрудника** в основном зависит от того, насколько успешно он работает на своем непосредственном рабочем месте. Построение системы учета вклада каждого члена проектной команды в проект гарантирует, что он будет лучше осознавать собственную значимость, а так же предоставит ему дополнительные стимулы к повышению эффективности выполняемых работ.

Моральное и материальное поощрение членов проектной команды по результатам выполненных работ следует рассматривать, с одной стороны, как возможный источник мотивации, а с другой стороны – как источник конфликтных ситуаций, как в пределах группы внедрения, так и при взаимодействии с другими подразделениями предприятия. Существует множество систем поощрений, которые можно использовать в целях повышения мотивации. Любая из них должна быть тщательно взвешена с позиции ее влияния на работу команды.

3.4.2.5. Управление качеством

На сегодняшний день существует бесчисленное множество определений понятия «качество», в частности, для различных отраслей бизнеса / промышленности. Одно из самых употребляемых определений было сформулировано в 1994 в Международном стандарте ISO 8402:94 «Управление качеством и обеспечение качества».

«**Качество** – совокупность характеристик объекта, относящихся к его способности удовлетворять установленные и предполагаемые потребности».

Управление качеством системы является комплексной задачей, для успешной реализации которой возможно, во-первых, фокусироваться на производстве качественного продукта изначально, а во-вторых, проводить своевременную оценку и корректировку уровня качества.

В первом случае, мероприятия направлены на обеспечение соответствия проекта и ИС как таковой ранее определенным требованиям.

- **Использование стандартов (управления проектами, разработкой программного обеспечения, управления ЖЦ и т.п.).**

Разумеется, опыт аналогичных проектов многих компаний и «лучшие мировые практики» дают общие руководства по управлению качеством и предлагают различные методики и инструменты, использование которых разумно и целесообразно. Однако необходимо понимать, что для их эффективного применения необходимо обладать соответствующим опытом и грамотно комбинировать разные стандарты во избежание их конфликтов между собой и несовместимости со спецификой предприятия или проекта.

– **Обучение, повышение квалификации персонала, использование внешних консультантов.**

Повышение уровня компетенций проектной команды является широко применяемым и даже обязательным условием получения качественных результатов.

– **Четкие критерии качества и стабильная внешняя среда проекта.**

Вероятность повышения качественного продукта повышается, если критерии оценки информационной системы заранее четко сформулированы и прозрачны, а рамки проекта остаются неизменными, без уменьшения сроков или стоимости.

В случае же оценки и корректировки качества, очень важны:

– **Прототипирование и тестирование с пользователями.**

Уже рассмотренное ранее в разделе ЖЦ ИС, прототипирование существенно снижает риски получения некачественного или несоответствующего требованиям продукта и своевременная проверка функциональности с пользователями при тестировании также способствует выявлению неверных архитектурных / функциональных / интерфейсных решений и их последующей корректировке.

– **Регулярный контроль и анализ.**

Контрольные карты, диаграмма Ишикавы для причинно-следственного анализа, план / факт анализ, планы совершенствования качества и процессов и другие методы управления проектами в зависимости от ситуации оказываются достаточно полезными в целях мониторинга качества.

Среди прочих стандартов, содержащих рекомендации по управлению качеством систем:

- ISO 9000:2005 «Системы менеджмента качества. Основные положения и словарь».
- ISO 9001:2008 «Системы менеджмента качества. Требования».
- ISO 9004:2009 «Менеджмент с целью достижения устойчивого успеха организации. Подход с позиции менеджмента качества».
- ISO 10001:2007 «Менеджмент качества. Удовлетворенность потребителя. Руководящие указания по кодексу поведения для организаций».
- ISO 10005:2005 «Системы менеджмента качества. Руководящие указания по планам качества».
- ISO 10006:2005 «Системы менеджмента качества. Руководящие указания по менеджменту качества проектов».
- ISO 10007:2003 «Системы менеджмента качества. Руководящие указания по менеджменту конфигурации».
- ISO/TR 10013:2001 «Рекомендации по документированию систем менеджмента качества».
- ISO/TR 10014:2006 «Менеджмент качества. Руководящие указания по реализации финансовых и экономических выгод».
- ISO 10015:1999 «Управление качеством. Руководящие указания по обучению».
- ISO 19011:2011 «Руководящие указания по аудиту систем менеджмента».

ISO 10006:2003 (ГОСТ Р ИСО 10006-2005)

Международный стандарт: **ISO 10006:2003 «Управление качеством. Руководящие указания по менеджменту качества проектов».**

Российский аналог: **ГОСТ Р ИСО 10006-2005 «Системы менеджмента качества. Руководство по менеджменту качества при проектировании».**

Стандарт акцентирует внимание своей аудитории на двух основных аспектах качества в управлении проектами: качестве процессов проекта и качестве продукции, так как недостаточное внимание / реализация требований хотя бы к одному из них может оказать негативное воздействие на ход проекта, его результаты, саму организацию.

Помимо собственных определений многим проектным терминам, ISO 10006:2003 описывает организационный аспект управления качества в части ответственности руководства, затем переходя к управлению ресурсами,

изготовлением продукта, и непосредственно к качеству (измерениям, анализу и улучшениям). Еще один раздел стандарта (глава «Системы менеджмента качества при проектировании», со ссылкой на ISO 9000), приводит восемь основных принципов: ориентацию на потребителя, лидерство руководителя, вовлечение работников, процессный подход, системный подход к менеджменту, постоянное улучшение, принятие решений на основании фактов и взаимовыгодные отношения с поставщиками [57, с. 8].

Рассмотрим основные выделяемые стандартом процессы подробнее:

Раздел	Процессы	Комментарии
Ответственность руководства	– Стратегический процесс управления	Обзор этапов и принципов менеджмента качества
Управление ресурсами	– Процессы, связанные с ресурсами	Принципы планирования ресурсов и их контроля (сравнение план / факт, примеры корректирующих мер)
	– Процессы, связанные с персоналом	Установление организационной структуры проекта, выбор персонала с необходимыми компетенциями и его последующее развитие
Изготовление продукции	– Взаимозависимые процессы	Работа с требованиями стейкхолдеров проекта, управление взаимодействием и получением обратной связи
	– Процессы, связанные с областью применения	Определение концепции и результатов применения разрабатываемого продукта (системы), документирование и последующий контроль необходимых для достижения целей проекта действий
	– Процессы, связанные со временем	Определение зависимостей между проектными активностями и ограничений задач типа «Start no later than», «Must finish on», оценка их продолжительности и на основе полученных данных формирование графика проекта с контролем выполнения
	– Процессы, связанные со стоимостью	Оценка стоимости проекта с составлением, согласованием и контролем бюджета
	– Процессы, связанные с обменом информацией	Формирование системы информирования по проекту его участников
	– Процессы, связанные с риском	Идентификация потенциальных рисков, оценка возможных последствий, разработка планов реагирования и контроль
	– Процессы, связанные с закупкой	Составление плана закупаемой продукции, технических требований, формирование критериев выбора поставщиков, а также процессы контрактной работы
Измерение, анализ и улучшение	– Процессы, связанные с улучшением	Определение «слабых мест» в части управления проектом и принятие корректирующих мер
	– Измерение и анализ	Сбор и оценка корректности данных
	– Постоянное улучшение	Распределение ответственности заказчика и подрядчика в части совершенствования процессов проекта

Непосредственно же текст стандарта содержит основные моменты, которым следует уделять внимание для успешной реализации проектов.

ПРИМЕР

7.8.3 Документирование требований к закупкам.

В документах на закупку должны быть идентифицированы продукция, ее характеристики, соответствующие требования системы менеджмента качества и связанная с ними документация. Документы должны также преду-

сматривать ответственность покупателя, определять стоимость и дату поставки продукции, требования аудита (при необходимости) и права доступа в помещения поставщика. Требования заказчика должны быть отражены в документах на закупку.

Документы (например, «Запрос расценок») должны быть построены так, чтобы упростить сопоставимые и полные ответы потенциальных поставщиков.

Документы на закупку должны быть рассмотрены до выполнения закупок с целью проверить, что все связанные с изделием требования и другие аспекты (такие как ответственность покупателя) полностью определены.

Рассмотрим еще один стандарт, концентрирующийся на одном из двух рассмотренных в ISO 10006:2003 аспектов: качестве процессов. Важным фактором обеспечения качества является повышение степени развития соответствующих процессах внутри самой организации-заказчика и ее подрядчиков. Чем раньше будут идентифицированы пути повышения зрелости данных процессов и предприняты необходимые меры, тем больше вероятность реализации проектов с высоким уровнем качества результатов. Для целей оценки уровня зрелости процессов управления компании в целом и процессов управления разработкой программных решений в частности применяются описанные ниже стандарт ISO 15504 и модель CMMI.

ISO 15504/SPICE. Модели уровней зрелости CMM/CMMI

Международный стандарт: **ISO 15504:2004 «Information Techology. Process Assessment» / SPICE (Software Process Improvement and Capability Determination).**

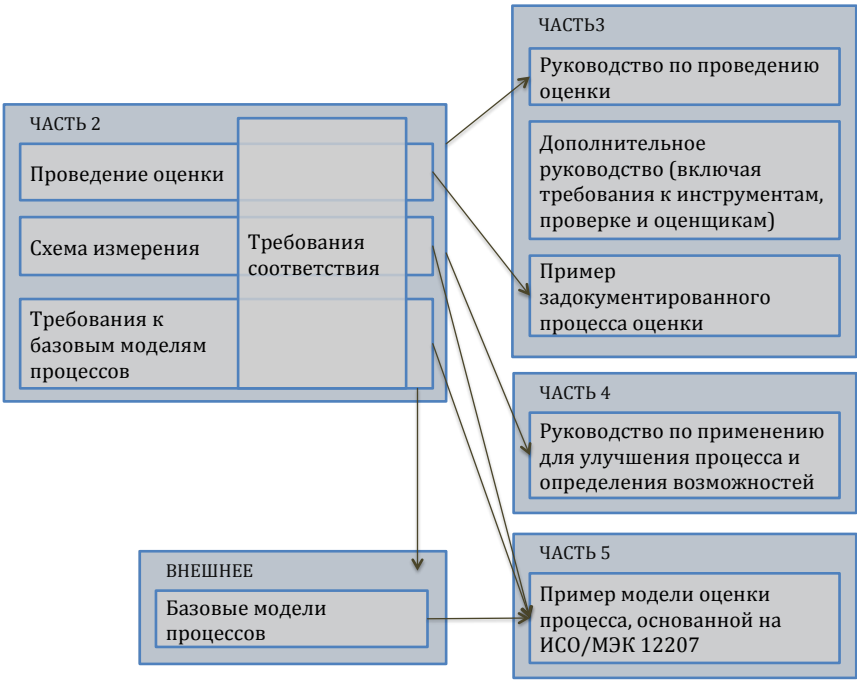
Российский аналог: **ГОСТ Р ИСО/МЭК 15504 «Информационные технологии. Оценка процессов».**

Стандарт, впервые представленный в 90-х годах, отличает фокус на модели улучшения процессов, а именно: эффективности работы и возможностей процесса. Под «эффективностью работы» понимаются продуктивность, учет потребностей пользователей / бизнес-заказчика, а «возможности процесса» определяются, как правило, организацией – поставщиком услуг по разработке и сопровождению системы для формулирования целевых потенциальных уровней совершенства этих процессов.

Всего в стандарте детализируются и рассматриваются пять основных групп процессов:

- Процессы взаимодействия «потребитель-поставщик» (Supplier-customer).
- Разработка (Engineering).
- Поддержка (Supporting).
- Управление проектом (Management).
- Организационные процессы (Organization).

Сама же структура стандарта приведена в схематичном виде на рисунке ниже.



Изначально данный стандарт исходит из концепции, согласно которой существуют определенные критерии / признаки уровня развития процессов организации. В некотором роде он опирается на методологию зрелости систем управления CMM и ее модификацию для оценки процессов разработки ПО – CMMI.

CMM (Capability Maturity Model) – методология, предложенная в конце 1980-х группой специалистов Университета Карнеги-Меллон и призванная сформировать систему критериев выбора организаций-подрядчиков для цели проектов разработки ПО. В связи с этим она называлась SE-CMM (Software Engineering Capability Maturity Model). А ее переработанная версия, посвященная целиком зрелости процессов разработки программного обеспечения, получила название **CMMI** (Integrated CMM).

Модели CMM/CMMI базируется на трех основных предположениях, которые применимы также и к стандарту ISO 15504.

1. Существуют несколько уровней зрелости проектной организации.

Эти уровни зависят от ряда критериев и применимы также и к разрабатывающим и внедряющим системы организациям. Всего данных уровней выделяется пять.

2. Любая организация заинтересована в переходе на более высокий уровень зрелости.

Это важно не только для выживания и сохранения конкурентоспособности на рынке, но и для повышения внутренней эффективности и качества выполнения проектов.

3. Переход возможен только на следующий уровень.

Нельзя «перескакивать» через уровни, так как одновременное внедрение большого числа мер для повышения зрелости резко повысит риски и может привести к непредсказуемым последствиям в случае неготовности организации к переходу.

Рассмотрим описание уровней зрелости SE-CMM в их первоначальном виде:

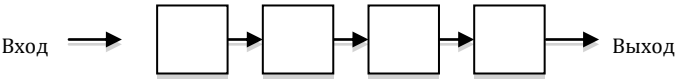
Уровень 1 – Начальный (initial).

Процесс разработки ПО представляет собой «черный ящик», не существует стандартов планирования и контроля проектов. Успех подобных проектов зависит целиком от компетентности, мотивации и героизма сотрудников, а не от применения отработанных процессов.

Схематично данный уровень зрелости может быть представлен следующим образом:

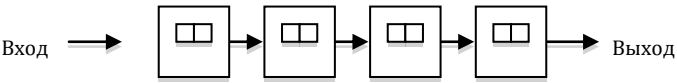
Уровень 2 – Повторяемый (repeatable).

Созданы базовые способы контроля проектов: планирование и мониторинг времени, стоимости, содержания и качества.



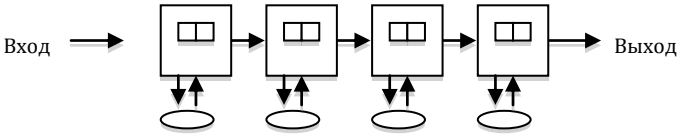
Уровень 3 – Определенный (defined).

Стандартные процессы документированы и внедрены в организации. Внутренняя структура «черного ящика» становится видимой. Менеджеры и команда проектов хорошо понимают свои роли и обязанности в процессах разработки.



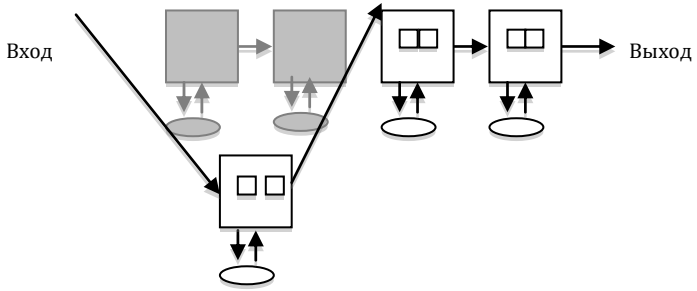
Уровень 4 – Управляемый (managed).

На данном уровне организацией определяются количественно измеряемые цели по достижению качества как программных продуктов, так и процессов. На основании получаемых при измерении результатов принимаются решения.



Уровень 5 – Оптимизирующий (optimizing).

Проактивное устранение существующих недостатков / «узких мест» процессов и их постоянный поиск и реализация возможностей по совершенствованию этих процессов.



Рассмотрев основные условия и особенности моделей зрелости в целом, перейдем к конкретным описаниям уровней по CMMI (ISO 15504) в отношении процессов разработки. CMMI отличают более детализированные описания и использование не только терминологии пяти уровней зрелости, но и концепции «непрерывной модели» оценки процессов (дополнительные введенные на каждом уровне метрики: например, PA 2.1 «Управление производительностью»). Для большей наглядности и демонстрации применимости модели CMMI как части стандарта ISO 15504 к процессам создания и эксплуатации системы приведем по уровням соответствующие комментарии.

Уровень 1 – Выполняемый (performed).

PA 1.1 Исполнение процесса.

Комментарии в отношении процессов разработки. При разработке ПО используются типовые инструменты, основные процессы ЖЦ не регламентированы и исполняются в соответствии с квалификациями и опытом специалистов. Сроки неопределенны в связи с сильной зависимостью от конкретной ситуации и отсутствием методик / регламентов в качестве базы.

Уровень 2 –Управляемый (managed).

PA 2.1 Управление производительностью.

PA 2.2 Управление продуктом.

Комментарии в отношении процессов разработки. С ростом сложности проектов по разработке и внедрению систем, ростом требований к квалификации специалистов появляются процессы управления процессом и самим продуктом. Проводится документирование проводимых работ и предпри-

нимаются первые шаги по регламентированию и формированию методик разработки, оценки и сопровождения систем.

Уровень 3 –Устоявшийся (established).

РА 3.1 Определение процесса.

РА 3.2 Становление процесса.

Комментарии в отношении процессов разработки. Третий уровень зрелости процессов жизненного цикла программных средств предполагает их стандартизацию, возможность доработки с учетом текущих особенностей конкретного проекта. К этому моменту процессы уже описаны и для них сформированы четкие входные и выходные параметры / результаты, требования и условия. Определены рекомендуемые к использованию с учетом конкретной ситуации программные средства и инструменты.

Уровень 4 – Предсказуемый (predictable).

РА 4.1 Измерение / оценка процесса.

РА 4.2 Контроль / мониторинг процесса.

Комментарии в отношении процессов разработки. На данном уровне зрелости процессов ЖЦ организации уже способны реализовывать крупномасштабные проекты внедрения даже с учетом жестких ограничений сроков / стоимости / заданного уровня качества. Соответственно, управление на четвертом уровне модели предполагает сформированные количественные показатели и характеристики процессов, для которых проводится мониторинг с последующим использованием полученной статистики и результатов для улучшения управления проектами и во избежание ошибок.

Уровень 5 –Совершенствуемый (optimizing).

РА 5.1 Инновации процесса.

РА 5.2 Непрерывная оптимизация.

Комментарии в отношении процессов разработки. Наконец, на «Совершенствуемом» уровне зрелости возможно снижать совокупную стоимость создания / владения системой за счет своевременного планирования, проведения ретроспективного анализа и использования новых, более эффективных инфраструктурных ресурсов, методик организации управления проектом, выбора оптимальных с точки зрения качества / стоимости / соответствия требованиям программных решений.

Для каждого пункта вышеприведенного списка проводится оценка степени выполнения того или иного процесса. Итоговая шкала состоит из всего четырех значений:

- Не реализован (0-15 %).
- Частично реализован (> 1-50 %).
- В значительной степени реализован (> 50-85 %).
- Полностью реализован (> 85-100 %).

Таким образом, рассмотренные методологии и стандарты способствуют развитию взаимопонимания между всеми сторонами, задействованными в

проектах разработки и внедрения информационных систем, предоставляя бизнес-ориентированным специалистам информацию о технической стороне внедрения, а ИТ-специалистам, в свою очередь, перспективу бизнеса и проектных менеджеров.

3.4.2.6. Управление содержанием

Проектная документация

Важность документации очень просто недооценить, однако именно благодаря ей любой человек сможет уяснить как общую картину и суть проекта, так и узнать самые незначительные детали его реализации. Сквозное документирование проекта (включающее все результаты проведенных работ, замечаний, разногласий, предложений, проведенных встреч) должно осуществляться в обязательном порядке. Основным элементом управления проектом является регулярный отчет руководителя проекта о состоянии проекта, в котором отражаются результаты выполнения работ по внедрению на момент формирования отчета. На протяжении всего проекта в отчете отражаются работы, выполненные в настоящее время, находящиеся на стадии выполнения (с указанием степени завершенности) и предстоящие в ближайшем будущем.

Появление промежуточных результатов в ходе реализации проекта позволяет начать их использование еще до завершения проекта в целом. Например, завершение этапа по автоматизации учета основных средств предприятия позволяет, не дожидаясь окончания проекта, осуществлять расчет амортизации.

Название документа	Автор / ответственный	Частота	Адресат документа
Регулярная отчетность по проекту			
1. Индивидуальный отчет о проработанном времени	Консультант-программист от Исполнителя	По указанию Руководителя проекта от Исполнителя	Руководитель проекта от Исполнителя
2. Отчет Руководителя проекта	Руководитель проекта от исполнителя (при его отсутствии – Руководитель проекта от Заказчика)	Ежемесячно	Куратор проекта, Координатор проекта
3. Протокол совещания Координационного Совета	Секретарь совещания	При проведении совещания	
Отчеты Руководителя проекта			
1. Регулярный отчет о состоянии проекта	Руководитель проекта от исполнителя	Ежемесячно	Куратор проекта, Координатор проекта
2. Отчет о результатах этапа	Руководитель проекта от исполнителя	При завершении этапа	
3. Протокол выполненных работ	Руководитель проекта от исполнителя	При выполнении работ	Руководитель проекта от исполнителя
Процесс решения проблем			
1. Форма регистрации проблемы	Консультант со стороны исполнителя		Руководитель проекта от исполнителя
2. Журнал регистрации проблем	Руководитель проекта от исполнителя		Куратор проекта, Координатор проекта
3. Изменения статусов проблем	Руководитель проекта от исполнителя		

Для эффективного управления проектом важными являются все без исключения составляющие. К примеру, даже такая, незначительная на первый взгляд, информация, как сведения о плане отсутствия специалистов (отпуска,

командировки и пр.) является необходимой для планирования конкретных работ проекта и соответствующего задействования специалистов. Именно из совокупности таких мелких элементов складывается качественное выполнение проекта внедрения и обеспечивается гарантия успеха проекта в целом.

Системная документация

Пристальное внимание следует уделить формированию системной документации. Для формирования документации можно использовать диаграммную технику представления управленческих процессов. Примером такой техники могут служить IDEF-диаграммы. При помощи функций описываются основные процессы, которые преобразуют входящую информацию и формируют выходную. Количественные функции (длительность, стоимость и т.п.) позволяют моделировать управленческие процессы и анализировать варианты их улучшения.

Документация системы должна включать в себя следующие элементы: информационная архитектура, программная архитектура, программно-техническая реализация. Базисом документации по поддержке и дальнейшему развитию КИС должны стать технологические, организационные и методологические составляющие. Эти составляющие представляют собой, по сути, правила изменения системы и могут охватывать все стадии проекта.

Документация должна обладать логической полнотой и удобством представления. Предприятия, включая АСУ-подразделения, нередко страдают отсутствием или невыполнением регламентов, слабой дисциплиной в области эксплуатации и замедленном восприятии изменений системы. Язык спецификаций – важная деталь, обеспечивающая возможность «говорить на одном языке» для заказчика и подрядчика. Создание любой составной части КИС должна сопровождаться созданием структурных схем и спецификаций, заранее согласованных с заказчиком. Логическая полнота спецификаций является обязательным условием, как и понимание их формулировок.

Для того, чтобы все вышеперечисленные особенности были учтены, разработаны специальные методики, позволяющие применять лучшие практики управления каждой фазой проекта. Однако более того, их понимание со стороны бизнеса важно для соблюдения интересов предприятия (и корректного планирования ресурсов и масштабов проекта) как во время заключения с подрядчиком контракта на системную интеграцию, так и в дальнейшем при прохождении всех этапов жизненного цикла этой информационной системы.

3.4.2.7. Управление рисками

Выбор и реализация стратегии управления рисками

Рассматривая важнейшие моменты управления проектами, невозможно обойти стороной риски, грамотное управление которыми способно спасти

проект, в то время как неверные действия по отношению к которым могут его «потопить».

Риск – неопределенное событие (условие), реализация которого влияет на ход проекта и достижение его целей.

В любом ИТ-проекте необходимо учитывать несколько крайне важных категорий рисков, ассоциированных как со внедрением, так и с поддержкой, технологическим совершенствованием и в целом управлением системы:

- Организационные: ассоциированные с поддержкой руководства компании и корректностью проведенного на первом этапе обследования бизнес-архитектуры компании.
- Ресурсные: риски проектного управления в части «треугольника ограничений»: времени, ресурсов, качества.
- Проектные: остальные риски проектного управления (в части контроля показателей проекта, управления коммуникациями, мотивацией, интеграцией).
- Технологические: риски возникновения ошибок, вызванных недостаточно точным анализом и проектированием системы на уровне архитектуры приложений, данных и технологий.

Не менее важно понимать не только, какие потенциальные ошибки можно совершить, но и когда именно следует проявлять к каждой особое внимание. Рассмотрим некоторые примеры в привязке к пяти этапам управления проектами по PMBoK (за основу берутся стадии проектного менеджмента, а не жизненного цикла ИС в силу значительной зависимости рисков именно от ошибок управления).

Инициирование

При старте проекта крайне важна поддержка руководства компании и то, насколько высокий уровень приоритета проекта (в сравнении с другими реализуемыми в компании активностями) будет им определен. Именно здесь на первый план выходит возможный конфликт интересов Бизнеса и ИТ, в случае появления которого может быть выделено недостаточно ресурсов, что повышает потенциальные «шансы» срыва сроков и / или снижения качества проекта в дальнейшем. Также значительные риски предполагает этап обследования, на котором неверная постановка требований (в частности, из-за неэффективных коммуникаций) приведет к несоответствию итогового продукта ожиданиям / потребностям заказчика.

Планирование

На этапе планирования появляются также новые риски, в частности, нехватка необходимых компетенций у проектной команды для корректно-

го проектирования системы и подготовки к ее реализации. Другой риск – неверная оценка сроков / объема проекта может привести к значительным задержкам проекта, к примеру, по причине опоздания в согласовании контракта на закупку лицензий. Никогда не следует забывать также об ассоциированных со всеми контрагентами: поставщиками, подрядчиками рисках, так как их банкротство, повышение тарифов, невыполнение договорных обязательств может существенно сказаться на ходе реализации проекта.

Исполнение

Уже во время реализации системы вероятно обнаружение невозможности реинжиниринга бизнес-процессов в текущих условиях, изменение которых потребует значительного увеличения сроков реализации либо бюджета проектов. Отсутствие мотивации у проектной команды, ошибки управления коммуникациями, рисками, интеграцией – все те риски, которые потенциального могут на этом этапе негативно сказаться на реализации проекта. Всегда существуют программные риски при приобретении ПО третьих фирм, однако еще более критичны здесь технологические риски. Ведь любые ошибки проектирования и настройки / реализации системы могут привести к потере данных, невыполнению контрактных обязательств, остановке системы (не говоря уже о последствиях ошибок для систем, выход из строя которых способен повлиять на безопасность жизни и здоровья человека).

Мониторинг и управление

На этапе мониторинга высокую степень важности имеет сравнение первоначальных целей / задач / содержания проекта и полученного результата для максимально оперативного выявления и устранения замечаний. Именно поэтому столь необходимо участие грамотных специалистов в тестировании системы – и обеспечение всех необходимых для процесса условий и ресурсов. Другим аспектом является вновь управление коммуникации, только уже при опытной эксплуатации и обучении пользователей, так как их участие и содействие крайне важно для последующего успеха проекта.

Завершение проекта

Основным риском по завершении проекта является изменение как внешних рыночных условий, так и внутренних процессов компании. А значит, решение, создаваемое в ходе проекта, может потерять актуальность, и даже более того – его изменение (в силу сложности и низкой гибкости архитектуры) может оказаться нерентабельным или просто невозможным.

Перед началом проекта внедрения необходимо четко уяснить, что может помешать достижению результата, и разработать меры по предотвращению возможных негативных последствий. Например, риск заключается в следующем: на предприятии идет подготовка к внедрению корпоративной системы управления, однако многие сотрудники не обладают основами компьютерной грамотности, что может привести к существенному снижению эффективности процесса внедрения. В этом случае мерой по предотвращению риска является предварительное обучение сотрудников предприятия основам работы с базовыми компьютерными программами и проведение последующей аттестации. Иногда в ходе анализа рисков выявляются десятки факторов, которые могут повлиять на процесс внедрения, однако суть остается прежней – выявление рисков, разработка и выполнение мероприятий по их устранению, контроль над исполнением.

В первую очередь, проводится **идентификация рисков**. Их список может быть сформирован руководителем проекта с командой внедрения при дальнейшем согласовании результата с куратором проекта. Этот список может быть представлен в форме таблицы по этапам ЖЦ системы, по этапам управления проектом внедрения, по основным классам рисков и другим видам группировки. Часто карта рисков составляется на основе проведенного бизнес-моделирования (карты бизнес-процессов).

Следующим шагом проводится **оценка рисков**, которые возможно представить в виде таблицы / матрицы, с обязательным указанием описания риска, его возможных последствий в качественном и количественном выражении (либо денежный эквивалент ущерба от реализации, либо оценка по шкале от 1 до 5). Вычисление данного значения чаще всего проводится на основании предварительного анализа вероятных потерь (от временных и денежных ресурсов до ущерба репутации компании, потери рыночной доли и т.п.). На этой стадии возможно использование анализа чувствительности, сценарного анализа, имитационного моделирования Монте-Карло, «bow-tie» анализа причин / событий / последствий реализации рисков и прочих специальных методов.

Затем указывается **вероятность реализации риска** (в %), а также возможные меры по снижению этой вероятности либо уменьшению потенциального ущерба. В самом простейшем варианте для получения итоговой оценки для каждого риска необходимо умножить значение *Ущерба от реализации* на *Вероятность осуществления риска* (переведя % в значение от 0 до 1). После ранжирования результатов и выделения рисков с наиболее высоким приоритетом назначаются ответственные за управление данным риском и принятие необходимых мер.

Определив самые критичные для проекта и бизнеса в целом угрозы, существует несколько **стратегий их минимизации.**

Сценарий действий	Необходимые меры
Снижение риска	Уменьшение как вероятности его наступления, так и возможных негативных последствий. ***ПРИМЕР*** *Так, для снижения риска недостоверности предоставленных на этапе обследования данных возможно фиксировать каждый факт предоставления информации, в формате аудио- / видеозаписей встреч, сохранения полученных / переданных сообщений, подписания протоколов совещаний со списком принятых решений, и пр. Еще в данном случае возможно привлечение квалифицированных специалистов со значительным и релевантным (!) опытом реализации подобных проектов (в предметной области либо индустрии), а также подготовка шаблонов заполнения / загрузки информации для минимизации вероятности ошибки.*
Перенос риска	Перенос риска на другую сторону при заключении отдельных соглашений. ***ПРИМЕР*** *Заключение договоров страхования, аутсорсинга и прочие связанные с третьими сторонами активности. Так, потери от срыва работ в рамках контракта с подрядчиком / поставщиком могут быть (по крайней мере, частично) покрыты соответствующим пунктом договора об ответственности за неисполнение его условий. Убытки от пожара в серверной (или какая-то их часть) могут быть покрыты страховой компанией.* Чаще всего исключением (риском, перенос которого почти невозможен) является ущерб деловой репутации компании и ее бренду).
Избежание риска	В определенной степени наиболее «практичный» сценарий, предусматривающий реализацию необходимых проектных активностей (возможно, чуть в большем объеме или в более сжатые сроки) во избежание многих проектных / организационных / технологических рисков. ***ПРИМЕР*** *Ежедневная отчетность для риска слабой организованности участников проекта, фиксация числа итераций и составление графика согласования / утверждения решений и результатов во избежание риска слишком длительного процесса оформления документов.*
Принятие риска	Занятие «позиции боевой готовности», если риск выявлен слишком поздно или уже реализован, определение способов покрытия ожидаемого ущерба и избежания подобных ситуаций в будущем. ***ПРИМЕР*** *Потеря данных при проблеме с дата-центром в качестве реализовавшегося (возможно, не идентифицированного своевременно) риска. В этом случае в качестве реактивных мир возможно активировать последние сделанные резервные копии, восстановить работоспособность системы, а в дальнейшем даже пересмотреть прежнюю политику компании в отношении достаточного уровня надежности используемого ЦОД, а также обновить (создать) стратегию обеспечения непрерывности бизнеса.*

И наконец, заканчивая разговор о рисках, упомянем столь важный аспект, как **«положительные» риски**. Часто сосредотачиваясь на негативной стороне наступления определенных возможных событий / условий, участники проекта забывают о другой его стороне. А ведь, к примеру, в то время как «обычным» риском будет являться уход из компании ключевых сотрудников, «позитивным» возможным риском будет участие в проекте признанного в своей предметной области эксперта, у которого неожиданно освободится время. Рассмотрим ту же матрицу сценариев действий, только уже для позитивных рисков.

Сценарий действий	Необходимые меры
Усиление (vs снижение) риска	***ПРИМЕР*** *Новость о возможном (при совпадении графиков) участии в качестве консультанта известного эксперта является позитивным риском, однако для усиления вероятности его реализации необходимо пересмотреть график проведения семинаров и встреч, чтобы поменять их расписание в соответствии со свободными временными слотами эксперта.*
Перенос риска	***ПРИМЕР*** *Если компания – системный интегратор, работающий в России с программными продуктами Microsoft, создаст на их базе отраслевое решение для определенного локального рынка, которое будет пользоваться популярностью, Microsoft может предложить дальнейшее сотрудничество на льготных условиях.*
Использование (vs избежание) риска	***ПРИМЕР*** *При наличии возможности получения скидки вендора при закупке определенного объема лицензий (который ранее предполагалось приобрести двумя этапами), компания может принять меры для изменения первоначального плана закупок и согласования нового контракта на больший объем лицензий.*
Принятие риска	***ПРИМЕР*** *Положительным риском может быть расширение числа пользователей внедренной системы. Если изначально планировалось использовать автоматизированную систему управления проектами только в головном офисе, но другие филиалы / дочерние компании, увидев все ее преимущества, могут также принять решение адаптировать данный программный продукт для своих нужд. Таким образом, сама компания будет иметь возможность реализовывать проекты в совместной работе нескольких филиалов, а компания – системный интегратор (либо внутреннее ИТ-подразделение) получает новый проект для выполнения.*

Риск всегда является неотъемлемой частью любого сложного и ответственного процесса. Более того, совершение рискованных действий необходимо для прогресса, а ошибки, как известно, являются основой приобретения опыта. Несмотря на то, что некоторые риски в ходе реализации проекта внедрения системы неизбежны, это не означает, что попытки определить их и управлять ими вредят творческой работе.

Обеспечение непрерывности бизнеса, планы DRP и BCP

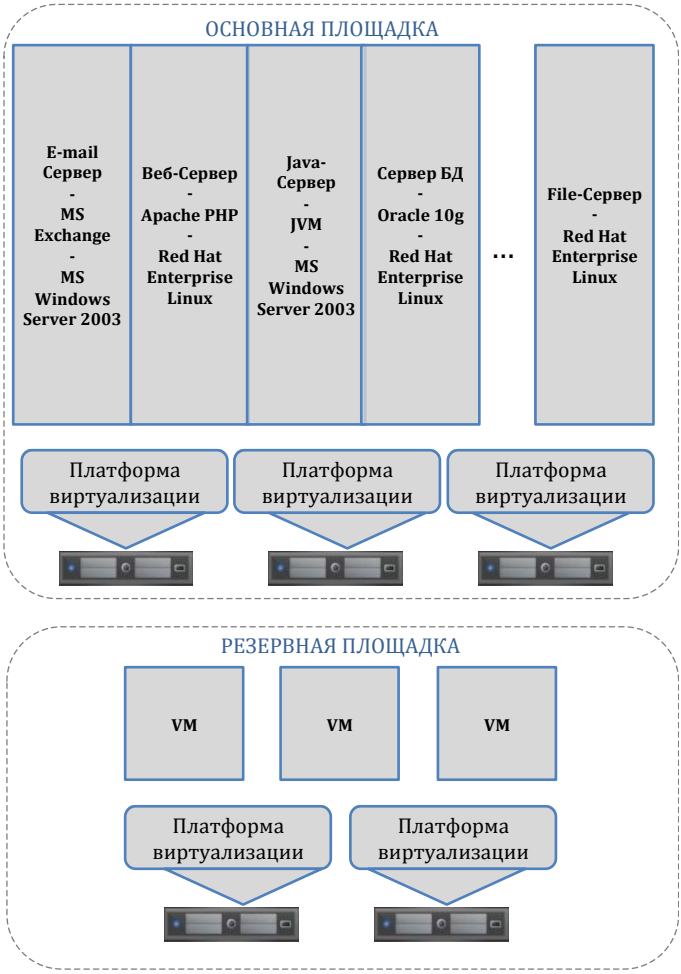
Для того, чтобы ИТ-инфраструктура не просто соответствовала потребностям бизнеса, но и обеспечивала непрерывность доступа к системе/ предоставляемым сервисам необходимо также предусмотреть концепцию / стратегию непрерывности. В этих целях возможно использовать серверные технологии в различных конфигурациях. В зависимости от ситуации технологические площадки:

- Могут быть **территориально распределены или локализированы**.
- Могут иметь **полный или частичный резерв оборудования** или не иметь его вообще.
  - В случае полного резерва при катастрофе на технологической площадке предоставление ИТ-услуги (приложения) обеспечивается за счет оборудования на второй площадке.

- В случае частичного резерва основное вычислительное оборудование установлено на одной площадке.
- Если резервирование не осуществляется, то восстановление на новом оборудовании производится с резервных внешних носителей информации.

ПРИМЕР

Возможен и другой вариант: построение резервной площадки «в облаке» (основанной на SRM). Таким образом, на основной площадке обеспечивается локальная доступность, реализуются решения по защите данных, а на резервной площадке организуется аварийное восстановление.



– Могут иметь **кластерную конфигурацию** (распределение нагрузки между равнозначными активными серверами приложений).

ПРИМЕР

Территориально распределенная конфигурация с подготовкой вычислительных ресурсов в виде «холодного» резерва.

Данная концепция предполагает подготовку резервной серверной площадки, включая всю необходимую серверную инфраструктуру, а также предварительную подготовку и размещение на ней набора оборудования, идентичного оборудованию основного ЦОД.

Работа основных и резервных вычислительных ресурсов в режиме взаимного резервирования, при котором часть оборудования на каждой из серверных площадок имеет запас вычислительной мощности. Резервные вычислительные мощности находятся в режиме простоя, либо используются для решения второстепенных задач.

Установка на резервной площадке дополнительных систем хранения данных и прочих технических средств резервного копирования данных, с сеансами репликации данных с основным ЦОД периодичностью 1 раз в сутки.

При выборе конкретной стратегии непрерывности и восстановления принимаются во внимание следующие аспекты:

– Уровень критичности приложения (системы), исходя из которого определяется допустимое время восстановления.
– Режим восстановления (ручной / автоматический).
– Стоимость реализации.
– Сложность / время реализации.

Подход по реализации стратегии непрерывности бизнеса, как правило, закрепляется в плане обеспечения непрерывности (BCP).

План обеспечения непрерывности бизнеса (Business continuity plan, BCP) – документ, предназначенный для выработки и соблюдения мер по минимизации рисков системных сбоев информационных систем компании и снижения возможных их последствий для бизнеса.

BCP-план направлен на обеспечение поддержки бизнес-процессов компании во время и после чрезвычайной ситуации и может охватывать либо один отдельный процесс, либо совокупность нескольких. ИТ системы рассматриваются в BCP исключительно с точки зрения поддержки бизнес-процессов и может даже не рассматривать вопрос восстановления процессов, фокусируясь только на временных требованиях непрерывности бизнеса. Основная цель подобного документа – своевременное определение вариантов действий в различных критических ситуациях. В случае подобных происшествий снижается объем финансового ущерба, обеспечивает более быстрое восстановление, а также удовлетворяет требованиям всех заинтересованных сторон (клиентов, акционеров, регуляторов, руководителей) к непрерывности работы компании.

Однако список задач по формированию плана обеспечения непрерывности бизнеса охватывает большее количество аспектов:

1. **Определение всех рисков для бизнеса** с формированием плана работ по каждому из данных рисков (в т.ч. ответственные сотрудники, меры предотвращения реализации рисков и нивелированию их негативного воздействия).

Так как план учитывает множество бизнес-факторов (например, репутационные риски), то как сам список рисков, так и список вариантов их снижения должен постоянно актуализироваться. Любые изменения в бизнес-процессах и бизнес-сервисах после их реализации должны быть пересмотрены в BCP-плане.

2. **Определение критических сервисов /** приложений, параметров целевого времени восстановления (recovery time objective, RTO) и целевой точки восстановления (recovery point objective, RPO), стоимости каждого часа простоя.

ПРИМЕР

Как правило, подавляющее большинство приложений требуют значение RTO, не превышающее 4 часов, и значение RPO менее 2-х часов.

3. **Принципы переноса данных** между основной и резервной площадками, выбор оптимального решения (в т.ч. с рассмотрением удаленности резервной площадки, пропускной способности каналов связи).

4. **Формирование процесса / процедур восстановления** сервиса в случае наступления критической ситуации – разработка плана.

План аварийного восстановления (Disaster recovery plan, DRP) – план действий по восстановлению полной работоспособности информационной системы (в т.ч. на альтернативных площадках)

План аварийного восстановления, как правило, фокусируется на тех ситуациях, когда доступ к информационным системам отсутствует / ограничен в течение длительного периода. В отличие от BCP, план DRP существует на более операционном уровне, описывая порядок действий (организации технических и организационных решений) по построению системы защиты инфраструктуры ИТ. Так, среди наиболее популярных решений по защите от чрезвычайных ситуаций – построение резервной площадки / ЦОД (disaster recovery site).

В качестве основных методологических материалов разработки планов обеспечения непрерывности / аварийного восстановления используются следующие международные стандарты и руководства:

- ISO/IEC 27001:2005 (ISO/IEC 17799:2005).
- Сборник лучших практик BCI (The Business Continuity Institute).
- Руководство CobiT.
- Contingency Planning Guide for Information Technology Systems (NIST).

Для планирования непрерывности бизнеса, как правило, проводятся следующие активности:

1. **Разработка политики непрерывности бизнеса** – определение требований и основных принципов.
2. **Анализ влияния на бизнес BIA (business impact anaylsis) –** определение критических ИТ-ресурсов, приоритетов восстановления.
3. **Разработка / внедрение превентивных мер** – внедрение и управление активностями.
4. **Определение стратегии восстановления.**
5. **Создание плана обеспечения непрерывности бизнеса BCP.**
6. **Тестирование** и обучение персонала.
7. **Эксплуатация плана BCP** – в т.ч. мониторинг реализации и обновление плана, взаимодействие с внешними контрагентами по вопросам непрерывности бизнеса.

Обеспечение информационной безопасности на предприятии

Цель информационной безопасности – обеспечить бесперебойную работу организации и свести к минимуму ущерб от событий, таящих угрозу безопасности, посредством их предотвращения и сведения последствий к минимуму. С точки зрения безопасности все виды информации, включая бумажную документацию, базы данных, информационные системы, съемные носители, разговоры и другие способы, используемые для передачи знаний и идей, требуют надлежащей защиты. Информация и поддерживающие её информационные системы и сети являются ценными производственными ресурсами организации. Их доступность, целостность и конфиденциальность является приоритетными направлениями программы развития информационной безопасности.

В основе развития информационной безопасности на предприятии часто находится риск-ориентированный подход, который подразумевает оптимальный набор средств защиты в соответствии с уровнем угрозы и возможными последствиями, определенными в ходе анализа рисков. Контроль и обеспечение защиты осуществляется на всех уровнях системы/предприятия в зависимости от потенциальных угроз и включает такие механизмы, как:

– **Сетевая безопасность.**

На этом уровне реализуются задачи защиты внешнего периметра и локальной сети:

- Потоковое сканирование входящего трафика.
- IPSEC/SSL VPN сервера, в т.ч. поддержка мобильных устройств, удаленного доступа, токенов, RSA-сертификатов.
- Выявление / блокирование атак, вирусов, вредоносного ПО.
- Предотвращение вторжений.
- Веб-фильтрация по предустановленным категориям трафика между объектами ЛВС и глобальной сетью.

При выборе конкретных программных средств для защиты внешнего периметра в первую очередь важны критерии пропускной способности, высокой доступности сервиса и устойчивости к атакам на отказ в обслуживании.

Пользователи подсети более низкого уровня безопасности не должны иметь доступ в сеть более высокого уровня, пользователи сетей одинакового уровня безопасности не должны иметь доступ в сети друг друга без явного разрешения в политике безопасности, пользователи сетей более высокого уровня безопасности должны иметь возможность взаимодействия с остальными частями сети.

При построении беспроводных сетей должны использоваться протоколы, с поддержкой механизмов аутентификации пользователей, взаимной аутентификации между беспроводным устройством и точкой доступа, шифрования передаваемых данных, а также обязательное физическое разделение беспроводных и локальных сегментов, с направлением трафика беспроводного сегмента на выделенный интерфейс межсетевого экрана. При организации гостевого сегмента, его пользователям доступ может быть разрешен только в публичные сети.

При подключении в локальную сеть каждый пользователь должен проходить обязательную проверку на наличие работающего антивируса, актуальности базы антивируса, установки критичных обновлений операционной системы и т.д.

– **Защита серверов и рабочих станций.**

Защита от несанкционированного доступа и воздействия вредоносного ПО на пользовательские и системные файлы

Защита серверного оборудования, рабочих станций и мобильных устройств:

- Антивирусная защита.
- Межсетевое экранирование трафика на конечном устройстве.
- Контроль целостности файлов.

– **Контроль и управление доступом.**

Создание централизованной системы идентификации и управления доступом пользователей к корпоративным информационным ресурсам:

- Централизованное ведение учетных записей пользователей.
- Поддержка ролевой модели предоставления доступа пользователям.
- Поддержка механизмов согласования со средствами локального администрирования систем.
- Поддержка Single sign-on, многофакторной аутентификации и пр.

– **Обеспечение конфиденциальности данных.**

Обеспечение конфиденциальности данных, обрабатываемой и хранящейся в информационно-телекоммуникационных системах (в частности, в области коммерческой тайны и персональных данных сотрудников, клиентов, контрагентов):

- Использование меток данных.
- Шифрование данных.
- Управление сертификатами ЭЦП.
- Мониторинг доступа пользователей к данным.
- Контроль печати конфиденциальных документов и записи информации на внешние носители данных.

– **Защита от утечки данных.**

Функции контентной фильтрации трафика, URL фильтрации, e-mail фильтрации, а так же контроль съемных носителей и печати конфиденциальных документов.

– **Построение централизованной системы управления и мониторинга средств информационной безопасности.**

Внедрение совокупности решений в централизованную систему мониторинга для обеспечения контроля событий ИБ и оценки защищенности инфраструктурных компонентов:

1. устройств сетевой безопасности;
2. конечных устройств;
3. системы централизованного управления доступом;
4. dlp компонент;
5. системы сканирования уязвимостей;
6. прочие средства безопасности.

Проводятся также следующие активности:

– Инвентаризация информационных активов.
– Ведение / анализ журналов / логов сетевого оборудования, межсетевых экранов, операционных систем, СУБД, приложений.
– Аудит управления обновлениями.
– Тестирование на проникновение.

При реализации инициатив необходимо не только провести интеграцию отдельных технических решений, но и принять ряд организационных мер:

– Разработать соответствующее методологическое обоснование (детальная политика безопасности, роли и матрица доступа пользователей всех приложений, классификация инцидентов информационной безопасности и т.д.).
– Подготовить организационно-распорядительные документы по управлению и взаимодействию в области информационной безопасности.
– Зафиксировать зоны ответственности подразделений.

3.4.3. Управление программой проектов

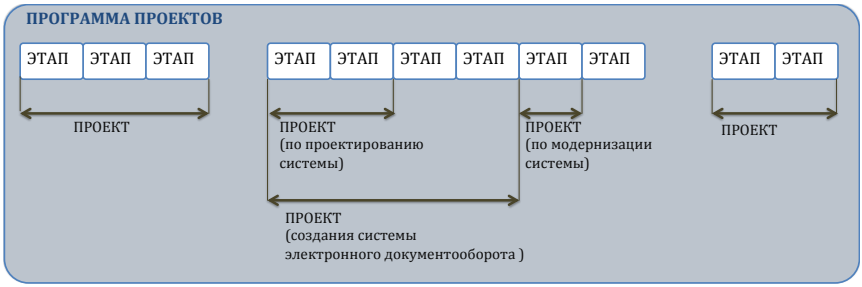
Когда в организации реализуются несколько взаимосвязанных проектов, либо объединенных временными / ресурсными ограничениями, либо имею-

щих перед собой одну цель, следует говорить о **программе / портфеле проектов.** Два эти понятия разные по своей сути, но часто используются в качестве синонимов – и поэтому важно в самом начале прояснить существующие различия.

Выделим все различия в табличной форме (на основании Стандарта по управлению программами, разработанному в 2006 году составителем PMBoK – институтом PMI и Руководства ICB IPMA).

	Проект	Программа	Портфель проектов
Основная цель	Проекты уникальны по сути и имеют четкие результаты	При реализации программы изменяются многие условия и элементы для соответствия ожидаемым выгодам	Портфели проектов концентрируются на бизнес-активностях, меняющихся с изменением стратегических целей компании
Видение и стратегия	Определяются в ТЭО проекта	Реализуются в рамках программы	Проводится мониторинг портфеля на соответствие стратегии
Бизнес-выгоды	Не рассматриваются в рамках проекта	Являются основным элементом программы	Часто не рассматриваются
Организационные изменения	Часто не рассматриваются	Рассматриваются	Не рассматриваются
Ограничения по времени и стоимости	Определяются в начале проекта и постоянно осуществляется их мониторинг	Относятся к конкретным проектам внутри программы	Базируются на приоритетах и стратегических целей конкретного портфеля проектов
Управление изменениями	Менеджер проектов старается минимизировать необходимость изменений	Менеджеры программ принимают происходящие изменения	Менеджеры портфеля проектов проводят регулярный мониторинг изменений
Измерение успеха	Успех измеряется соответствием плану по бюджету, срокам и содержанию/качеству работ	Успех измеряется в достижении выгод, реализации стратегических преимуществ, положительном ROI	Успех измеряется в терминах интегрированного успеха компонентов портфеля проектов
Стиль лидерства	Стиль лидерства основывается на выполнении задач для достижения целевых показателей	Стиль лидерства основывается на управлении взаимоотношениями сторон и разрешении конфликтов при соблюдении интересов стейкхолдеров	Стиль лидерства основывается на добавочной ценности, привносимой в процесс принятия решений
Принцип руководства	Менеджеры проекта руководят специалистами в команде	Менеджеры программ руководят менеджерами проектов	Менеджеры портфеля проектов координируют своих подчиненных (в офисе управления проектами)
Принципы планирования	Детальное планирование для обеспечения выполнения работ	Высокоуровневое планирование всех проектов	Акцент не на планировании, а на создании и поддержке необходимых процессов и коммуникаций в портфеле проектов
Принципы мониторинга	Менеджеры проектов проводят мониторинг задач и работ для выполнения проекта	Менеджеры программ организуют мониторинг проектов и текущую работу через структуры корпоративного управления	Менеджеры портфеля проектов контролируют интегрированные показатели эффективности проектов

Дальнейшие шаги будем рассматривать на примере управления программой проектов, как более сложной структуры, фокусирующейся не только на формальных ограничениях и методах балансировки проектов, но и на стратегическом аспекте их выбора. **Программой** проектов в данном контексте и терминологии будет являться объединенный общей целью и условиями выполнения набор проектов, реализация которого призвана обеспечить соответствие ожидаемым выгодам. В таком случае важен именно совокупный эффект от выполнения всей программы, а не отдельные результаты каждого из проектов.



ПРИМЕР

Рассмотрим программу проектов, цель которой – «Организация обеспечения информационной поддержкой управления сетью распределенных мультифункциональных придорожных комплексов». При этом среди предоставляемых на комплексе услуг: терминалы оплаты, вендинговые автоматы, кафе с wifi – и разумеется, сама заправка (газом / нефтепродуктами/стеклоомывающей жидкостью). Соответственно, среди систем автоматизации будут как минимум:

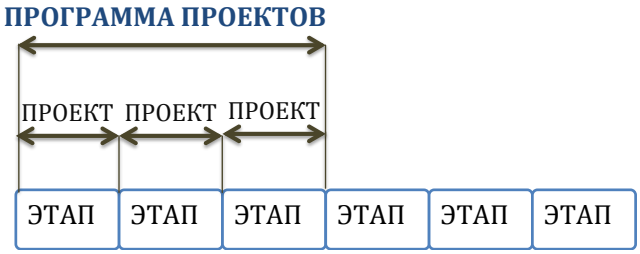
- *Система управления автозаправочным комплексом;*
- *Система управления гостиничным комплексом;*
- *Система управления розничной торговлей в региональной сети АЗК,*
- *Развитие мультисервисной корпоративной сети;*
- *Развитие ИТ-инфраструктуры в центральном офисе;*
- *Внедрение систем безопасности и охраны:*
  - *Система видеонаблюдения внутренней территории комплекса;*
  - *Автоматическая паркинг-система;*
  - *Система регистрации автомобильных номеров (въезды / выезды на территории комплекса);*
  - *Система видеонаблюдения касс;*
  - *Другие системы.*

Соответственно, подобная автоматизация бизнес-процессов и построение комплекса (в значительной части, с инфраструктурной точки зрения) не могут происходить одновременно – и важно корректно расставить приоритеты и определить те системы, создание которых является необходимым условием для реализации следующих.

Так, в построении систем безопасности и охраны с большой долей вероятности будут задействованы другие специалисты, нежели в автоматизации процессов розничной торговли, а также масштаб и длительность второго проекта превысят аналогичные показатели первого. Это необходимо также учитывать при планировании персонала на проекты и заключении контрактов на их реализацию.

Однако для успешного прохождения всего жизненного цикла крайне важен учет тесной взаимосвязи ее этапов как отдельных проектов. К тому же, неизбежно и их взаимодействие с прочими реализующимися компанией проектами, возможно, в рамках борьбы за совместно используемые ресурсы. Соответственно, удобнее и целесообразнее рассматривать **жизненный**

цикл ИС до этапа эксплуатации как **единую программу проектов**, планирование которой, как правило, осуществляется «сверху вниз», в силу необходимости распределения временных, денежных и прочих ресурсов. Критическим фактором успеха является логическая связь этапов/проектов (определяемая выбранной моделью жизненного цикла). На схеме ниже приведен пример для каскадной модели.



ПРИМЕР

Если, как уже было сказано, жизненный цикл информационной системы может рассматриваться как целая программа проектов, разберем отдельно проект по модернизации на примере системы бухгалтерского учета, проведенный российской компанией – системным интегратором.

Допустим, компания (головной офис которой находится в одной из европейских стран) во всех российских подразделениях использовала Microsoft Dynamics Nav версии 3.01. При анализе и сборе требований, предшествующем модернизации, было определено что с ростом масштабов бизнеса, числа операций, количества пользователей возникла необходимость расширения ее функциональных возможностей и поддержки вдвое увеличившегося числа одновременно работающих в системе сотрудников.

В результате было принято решение модернизировать ИС для обеспечения полноценного учета финансовой деятельности (по двум стандартам: РСБУ и US GAAP); получения информации из систем других филиалов компании (для чего понадобится разработка интеграционного решения и интерфейсов), увеличения скорости обработки данных.

При подготовке технического задания на модернизацию был проведен анализ всех финансовых операций, сформирована схема работы сотрудников в системе (с проверкой вводимых данных, оптимизацией многих процедур), а также выбрано программное решение в качестве платформы для модернизации (Microsoft Dynamics Nav, только уже версии 4.0).

По итогам разработки / настройки и тестирования получена модернизированная информационная система с расширенной функциональностью (например, выгрузка данных по контрактам во вновь интегрированную систему оперативного учета), увеличенным быстродействием и более эффективным использованием аппаратных ресурсов (за счет обновленной версии ПО).

А значит, действительно, этап модернизации (как и остальные стадии, за исключением эксплуатации) имеет признаки проектной деятельности, и

каждым из этапов можно управлять через определение конечного результата проекта, а также временных и стоимостных ограничений.

3.4.4. Сбалансированная система показателей (BSC)

Концепция Сбалансированной системы показателей BSC впервые была представлена в работах Каплана и Нортона, определивших набор показателей, позволяющих руководителям компаний получить полную высокоуровневую картину состояния бизнеса и адекватно оценить степень реализации своего видения / стратегии, их внешние и внутренние перспективы. Представляя модель как инструмент стратегического управления, авторы предложили, исходя из видения развития компании определять цели, показатели эффективности (финансовые и нефинансовые), их целевые значения и инициативы по достижению.

Всего в BSC выделяется четыре основные точки зрения / **перспективы**, по сути являющиеся основными направлениями развития любой компании, для которых и определяются в дальнейшем стратегические цели:

– **Финансовая.**

Бизнес-выгоды от реализации проектов.

– **Клиентская.**

Повышение уровня обслуживания и удовлетворенности пользователей.

– **Процессная.**

Совершенствование процессов, обеспечение качества и внутренней эффективности.

– **Обучение и развитие.**

Эффективность управления персоналом, мотивацией, знаниями. Управление инновациями.

Эти четыре перспективы находятся в иерархической взаимосвязи между собой.

ПРИМЕР

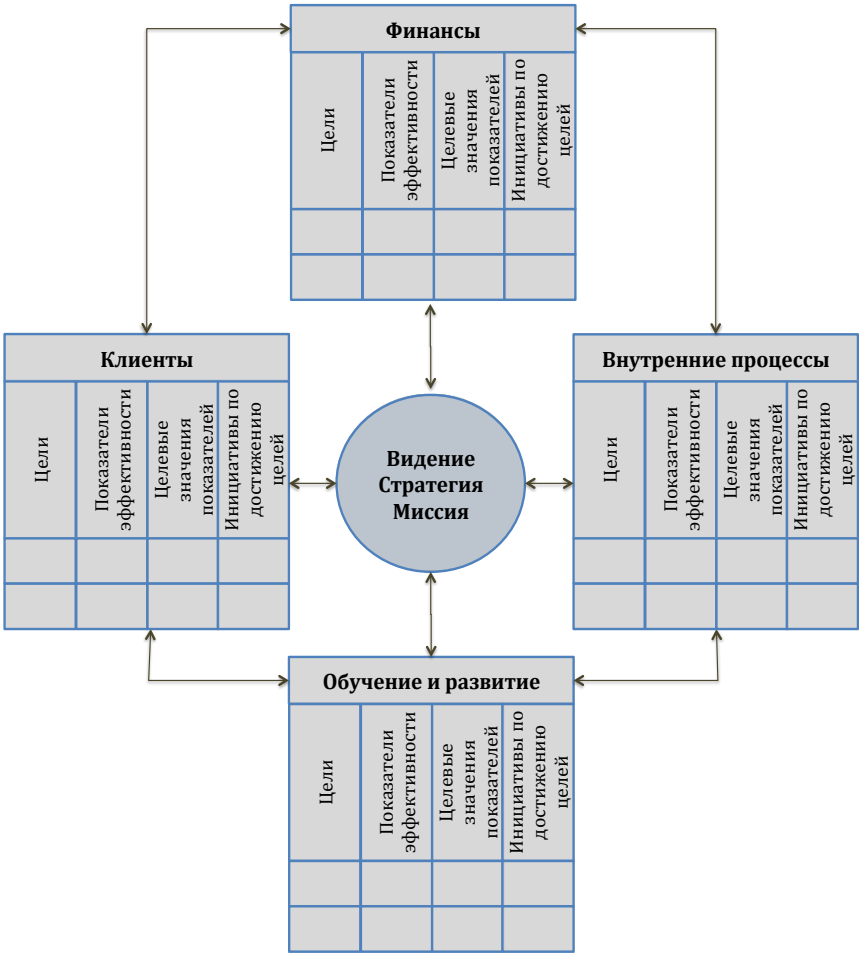
Эффективность обучения персонала и инновационных активностей оказывает непосредственное влияние на внутренние бизнес-процессы, эффективность деятельности (производства). Правильно выстроенные внутри компании бизнес-процессы способствуют повышению удовлетворенности клиентов, усиливают рыночную позицию. Высокий уровень обслуживания и устойчивое положение на рынке повышают финансовые показатели, что в целом говорит об успехе компании.

Для всех стратегических целей в соответствии с BSC определяются основные **ключевые показатели эффективности** (факторы, влияющие на достижение этих целей и оценивающие прогресс компании в этом направлении) и **критические факторы** (по сути – основные риски).

Для определения КПЭ необходимо проведение следующих действий:

1. Документирование бизнес-процессов.
2. Определение целей бизнес-процессов.
3. Формирование индикаторов их достижения в количественных и качественных терминах.
4. Определение критических значений для созданного списка показателей эффективности (для дальнейшей оценки степени отклонения от эталонных значений).

Общая схема карты показателей эффективности в четырех основных перспективах BSC представлена на схеме ниже.



ПРИМЕР

Приведем несколько примеров показателей эффективности:

- ***Финансовая перспектива.***
  - *Объем рынка (в денежном выражении) / рыночная доля.*
  - *Степень лояльности акционеров.*
  - *Размер выручки / прибыли.*
- ***Клиентская перспектива.***
  - *Процент клиентов, остающихся с компанией уже более 2 лет.*
- ***Процессная перспектива.***
  - *Среднее время разрешения инцидентов.*
  - *Финансовые потери клиентов, понесенные по вине компании.*
  - *Процент невыполненных запросов, поступивших за период времени*
- ***Перспектива «Обучение и развитие».***
  - *Уровень удовлетворенности сотрудников обучением в компании (по шкале от 1 до 10).*
  - *Количество часов обучения на одного сотрудника в год.*
  - *Разница в уровне прибыли до и после внедрения инноваций.*
  - *Размер прибыли после вывода на рынок нового продукта / услуги.*

ЗАКЛЮЧЕНИЕ

В монографии представлен материал по жизненному циклу информационных систем, по основным моделям ЖЦИС, по этапам, а также по проектному управлению, по особенностям ИТ-проектов. Дана краткая характеристика системы проектного управления, принципы организации управления проектом, программой. Приведена информация о российских и международных стандартах и корпоративных методиках проектного управления.

Автор дает свое определение информационной системы. Это определение позволяет более точно определить результат, к которому стремятся проектные команды, когда они выполняют проекты по созданию, внедрению, эксплуатации, модернизации ИС.

Приведенные стандарты и корпоративные методики дают общее представление, каким путем можно придти к запланированному результату, и опираясь на эти стандарты разработать свой план реализации.

Автор дает характеристику этапов жизненного цикла.

1. Планирование проекта.
   - Экспресс-обследование.
   - Технико-экономическое обоснование.
   - Оценка целесообразности проекта (Telos).
   - Выбор программного решения.
2. Анализ и постановка задачи.
   - Информационное обследование предприятия.
   - Описание бизнес-процессов.
   - Сбор требований.
   - Подготовка технического задания.
3. Проектирование.
   - Техническое проектирование.
   - Рабочее проектирование / Прототипирование при заказной разработке.
4. Разработка.
   - Закупка ПО.
   - Настройка конфигураций.
   - Создание ролей пользователей.
   - Миграция данных.
   - Разработка контрольного примера.
   - Тестовая эксплуатация.
   - Доработка по результатам тестирования.
   - Прием результатов испытаний.

5. Развертывание и внедрение.
   – Закупка и настройка требуемой ИТ-Инфраструктуры.
   – Ввод начальных остатков.
   – Обучение пользователей.
   – Развертывание системы на рабочих местах.
   – Основные виды тестирования.
   – Опытно-промышленная эксплуатация.
   – Приемо-сдаточные испытания.
6. Эксплуатация.
7. Сопровождение эксплуатации.
   – Авторский надзор.
   – Техническая поддержка.
   – Постгарантийное сопровождение.
   – Обновление и релизы.
   – Увеличение производительности системы.
8. Модернизация.
   – Стратегия управления LEGACY-Системами.
   – Виртуализация как стратегия как стратегия модернизации решений.
   – Особенности проектов по модернизации.
9. Утилизация.

Успех работы над проектом зависит от точности определенной цели работы, от согласованной работы проектных команд как от заказчика, так и от исполнителя (подрядчика). В приложениях приведены образцы документы, которые используются на определенных фазах жизненного цикла информационных систем для организации процесса управления работой таких команд. Приведенные формы документов не претендуют на полноту. Они являются образцами документов. Каждая проектная команда может разрабатывать свои проектные документы, состав и полнота которых определяется целями и задачами проекта.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

2. ISO/IEC 12207:2008 Information technology – Software life cycle processes (Информационные технологии. Процессы жизненного цикла программного обеспечения).

3. ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».

4. ISO/IEC 15288 Systems engineering. System life cycle processes (Системотехника. Процессы жизненного цикла системы).

5. ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.

6. ISO/IEC 19510:2013 Информационные технологии. Модель и нотация процесса менеджмента объекта в групповом бизнесе.

7. ISO/IEC 19505:2012. Информационные технологии. Унифицированный язык моделирования группы по управлению объектами (OMG UML).

8. ГОСТ 19.201-78 Техническое задание, требования к содержанию и оформлению.

9. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.

10. ГОСТ 34.201-89 Виды, комплектность и обозначение документов при создании автоматизированных систем.

11. ГОСТ 12119 ИТ. Пакеты программ. Требования к качеству и тестирование.

12. ГОСТ 12207 ИТ. Процессы жизненного цикла программных средств.

13. ГОСТ 14764 ИТ. Сопровождение программных средств.

14. ГОСТ 15504 ИТ. Оценка процессов.

15. ГОСТ 9126 ИТ. Оценка программной продукции. Характеристики качества и руководства по их применению.

16. ГОСТ серии 19 Единая система программной документации (ЕСПД).

17. IEEE 829 Стандарт документирования тестирования программного обеспечения и систем.

18. IEEE 1219 Standard for Software Maintenance.

19. CobiT (Control Objectives for Information and Related Technology).

20. ITIL (IT Infrastructure Library).

21. SWEBOK (Guide to the Software Engineering Body of Knowledge).

22. PMBOK (Project Management Body of Knowledge).

23. PRINCE2 (PRojects IN Controlled Environments).

24. ISO 21500:2012 Guidance on project management.

25. ГОСТ Р ИСО 21500:2012 Международный стандарт по Управлению Проектами.

26. ГОСТ Р 54869-2011 Проектный менеджмент. Требования к управлению проектом.

27. ГОСТ Р 54871-2011 Требования к управлению программой.

28. ГОСТ Р 54870-2011 Требования к управлению портфелем проектов.

29. ISO 10006:2003 Quality management systems – Guidelines for quality management in projects.

30. ГОСТ Р ИСО 10006-2005 Системы менеджмента качества. Руководство по менеджменту качества при проектировании.

31. ISO 8402:94 Управление качеством и обеспечение качества.

32. ISO 9000:2005 Системы менеджмента качества. Основные положения и словарь.

33. ISO 9001:2008 Системы менеджмента качества. Требования.

34. ISO 9004:2009 Менеджмент с целью достижения устойчивого успеха организации. Подход с позиции менеджмента качества.

35. ISO 10001:2007 Менеджмент качества. Удовлетворенность потребителя. Руководящие указания по кодексу поведения для организаций.

36. ISO 10005:2005 Системы менеджмента качества. Руководящие указания по планам качества.

37. ISO 10006:2005 Системы менеджмента качества. Руководящие указания по менеджменту качества проектов.

38. ISO 10007:2003 Системы менеджмента качества. Руководящие указания по менеджменту конфигурации.

39. ISO/TR 10013:2001 Рекомендации по документированию систем менеджмента качества.

40. ISO/TR 10014:2006 Менеджмент качества. Руководящие указания по реализации финансовых и экономических выгод.

41. ISO 10015:1999 Управление качеством. Руководящие указания по обучению.

42. ISO 19011:2011 Руководящие указания по аудиту систем менеджмента.

43. ISO 10006:2003 Управление качеством. Руководящие указания по менеджменту качества проектов.

44. ГОСТ Р ИСО 10006-2005 Системы менеджмента качества. Руководство по менеджменту качества при проектировании.

45. ISO 15504:2004 Information Techology. Process Assessment / SPICE (Software Process Improvement and Capability Determination).

46. ГОСТ Р ИСО/МЭК 15504 Информационные технологии. Оценка процессов.

47. Contingency Planning Guide for Information Technology Systems (NIST).

48. Ананьин В., Зимин К. Ценность ИТ. Теория и практика. CIO congress Kirov. 28.04.2012. Режим доступа от 18.02.2013. [http://bit.ly/139QkWJ].

49. Блинов Д. Требования к системе: классификация FURPS+. Портал BeamTeam.ru, 2010г. [http://beamteam.ru/2010/09/furps/].

50. http://www.novsu.ru/file/977849.

51. http://www.swell-services.be/cmsms/index.php?page=it-system-decommissioning.

52. http://2.bp.blogspot.com/-6w5M_Pf3Gq8/TskqA8pxBFI/AAAAAAAAF9Y/YQPTqN6wy9A/s1600/image00.png.

53. http://www.cnews.ru/promo/sap/forum/usp_project/Serikov.pdf.

54. http://www.intuit.ru/studies/courses/2196/267/print_lecture/6796.

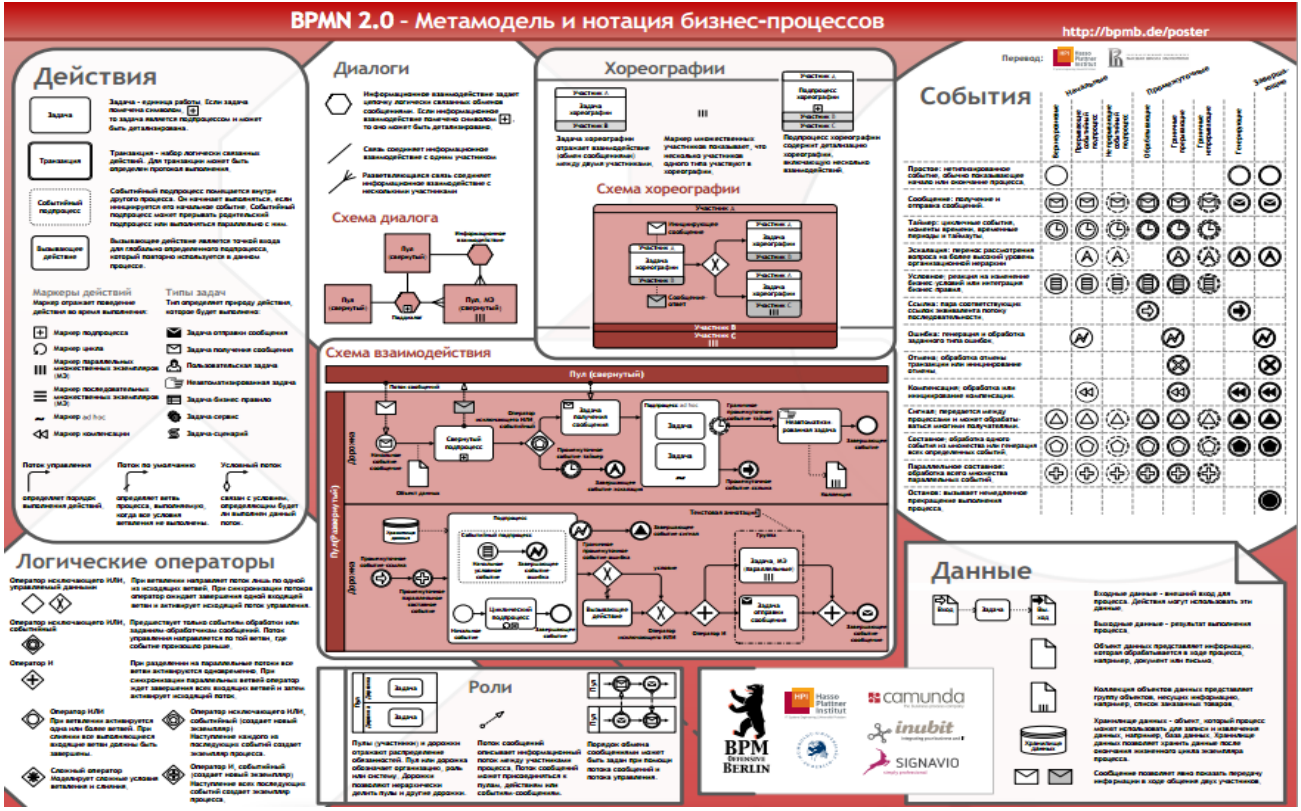
55. http://www.edu.dvgups.ru.

56. http://tsconsulting.ru/upload/iblock/37b/37b27c616e326793c95eb98b325a0748.pdf.

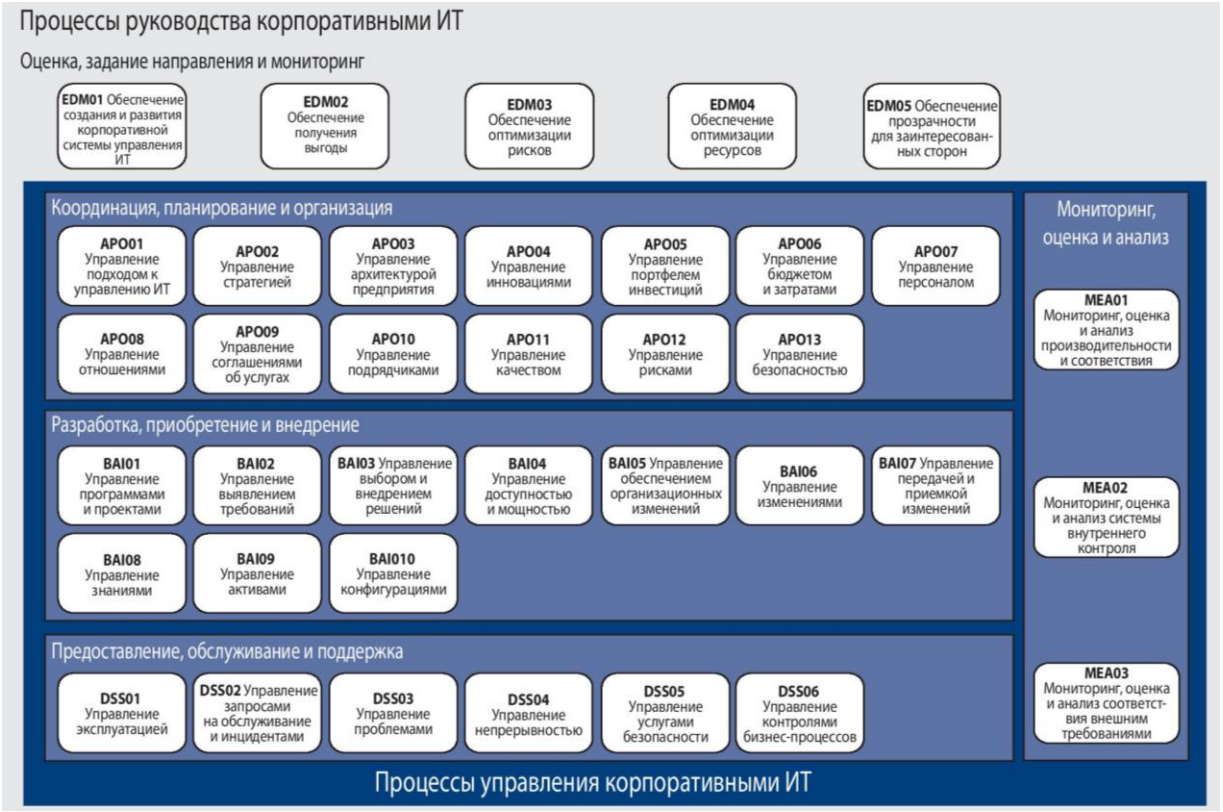
57. http://www.pqm-online.com/assets/files/standards/gost_r_iso_10006-2005.pdf.

ПРИЛОЖЕНИЯ

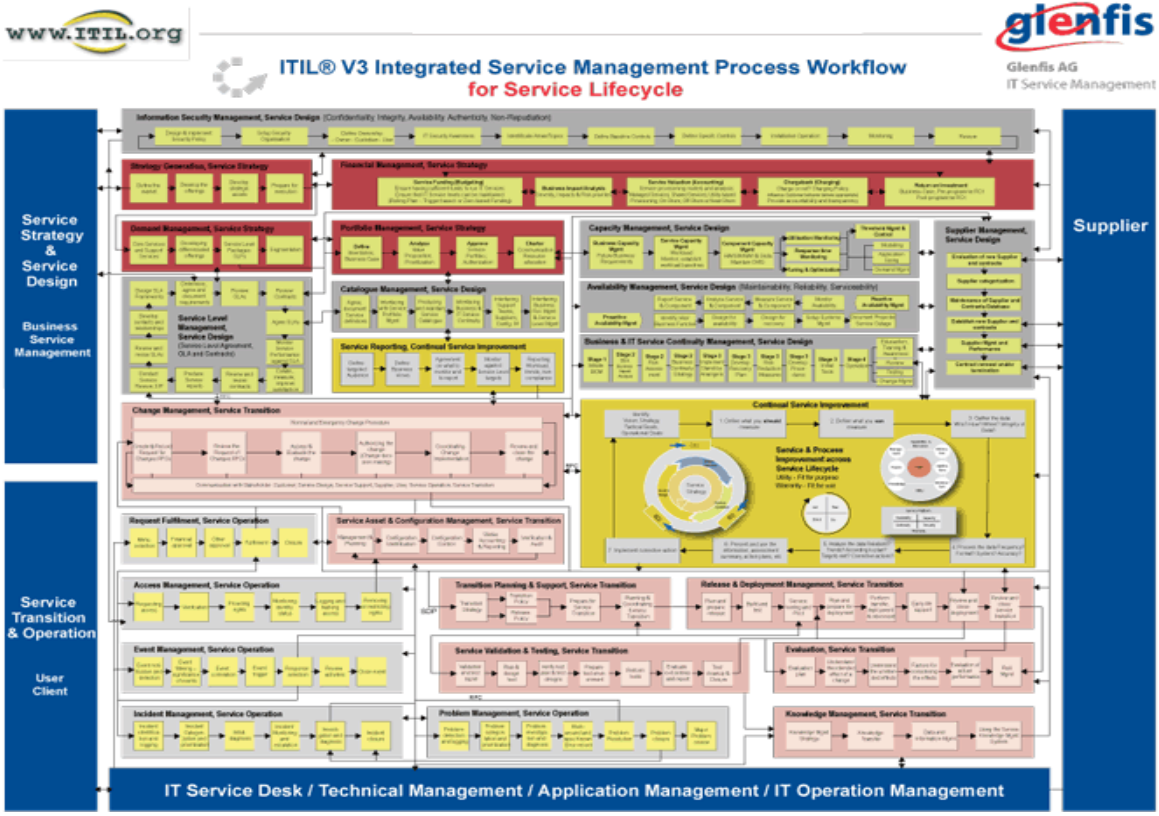
ПРИЛОЖЕНИЕ 1. СХЕМА НОТАЦИИ BPMN 2.0



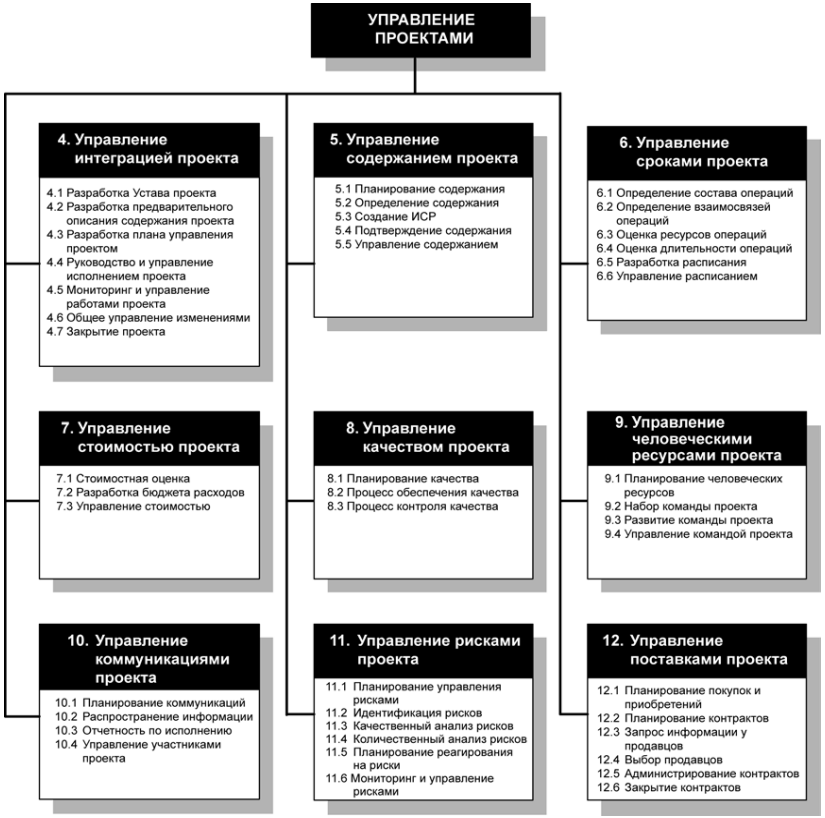
ПРИЛОЖЕНИЕ 2. **МОДЕЛЬ COBIT 5**



ПРИЛОЖЕНИЕ 3. **МОДЕЛЬ ITIL V3**



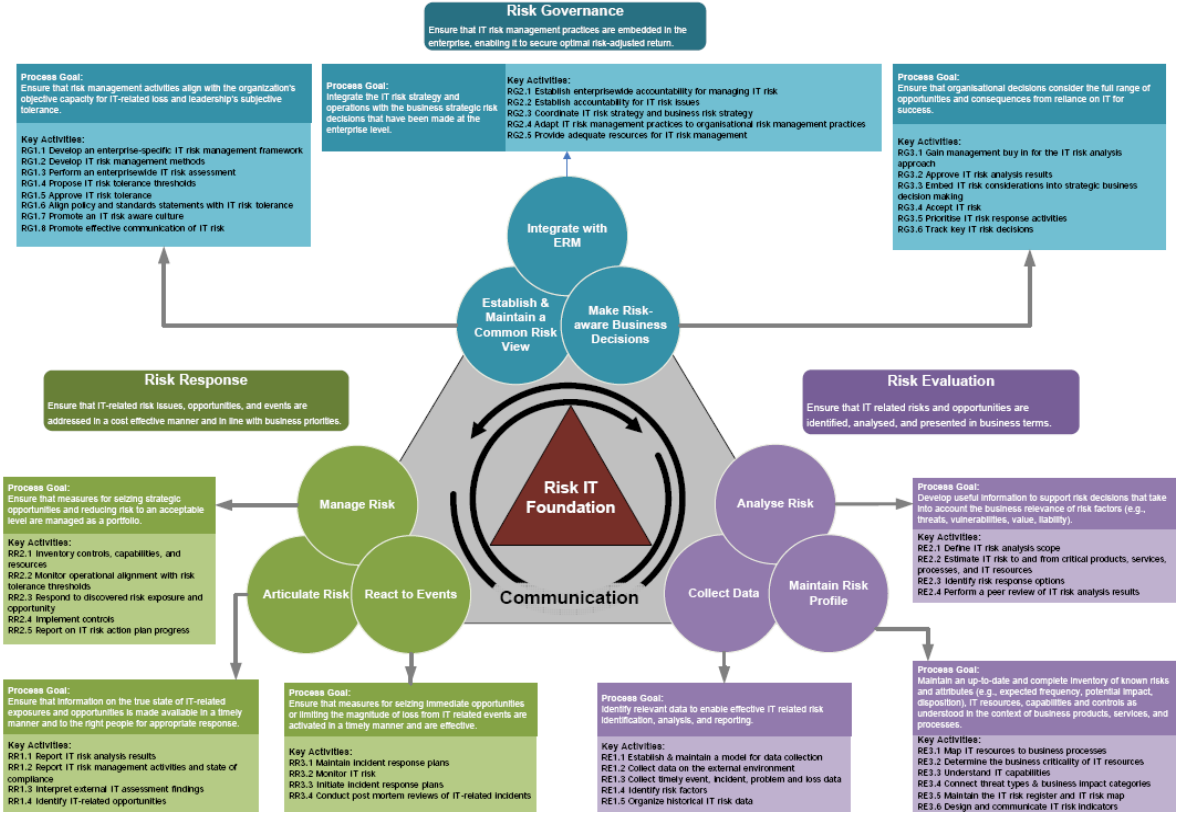
ПРИЛОЖЕНИЕ 4. ОБЛАСТИ ЗНАНИЙ PMBOK V4



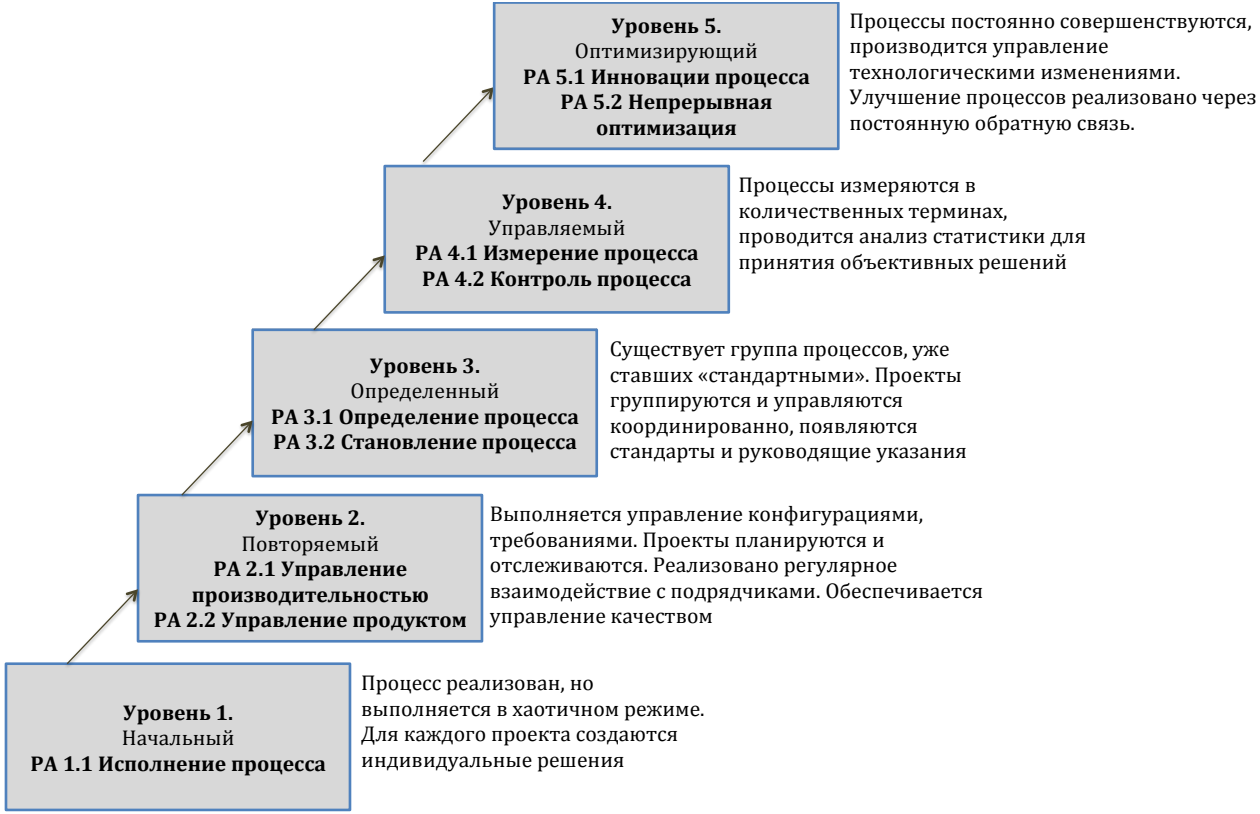
ПРИЛОЖЕНИЕ 5. SWEBOK. ОСНОВНЫЕ РАЗДЕЛЫ

ПРОГРАММНАЯ ИНЖЕНЕРИЯ. **ОСНОВНЫЕ РАЗДЕЛЫ**				
Инженерия требований	**Проектирование ПО**	**Конструирование ПО**	**Тестирование ПО**	**Сопровождение ПО**
Инженерия требований Выявление требований Анализ требований Спецификация требований Проверка требований Управление требованиями	Базовые концепции Ключевые вопросы Архитектура ПС Анализ качества Нотации Стратегия и методы проектирования	Снижение сложности Отклонения от стиля Проверки Использование внешних стандартов	Концепции Уровни тестирования Техники Метрики Управление тестированием	Концепции Сопровождение Ключевые вопросы Спецификация требований Процесс сопровождения
ПРОГРАММНАЯ ИНЖЕНЕРИЯ. **ОРГАНИЗАЦИОННЫЕ РАЗДЕЛЫ**				
Управление конфигурацией	**Управление проектами**	**Процесс инженерии ПС**	**Методы и средства инженерии**	**Инженерия качества ПС**
Управление: Процесс Идентификация Контроль Учет статуса Аудит Управление версиями	Организация управления Управление процессами Управление проектом Инженерия планирования Инженерия измерения Инженерия стоимости	Концепции Инфраструктура Определение Оценка Количественный анализ Процесс изменений	Методы: Объектные Структурные Компонентные Инструменты: Инженерии Oracle, Rational Rose,…	Концепции качества Деятельности и техники Гарантии качества ПС Планирование качества Проверка и контроль качества Измерение качества

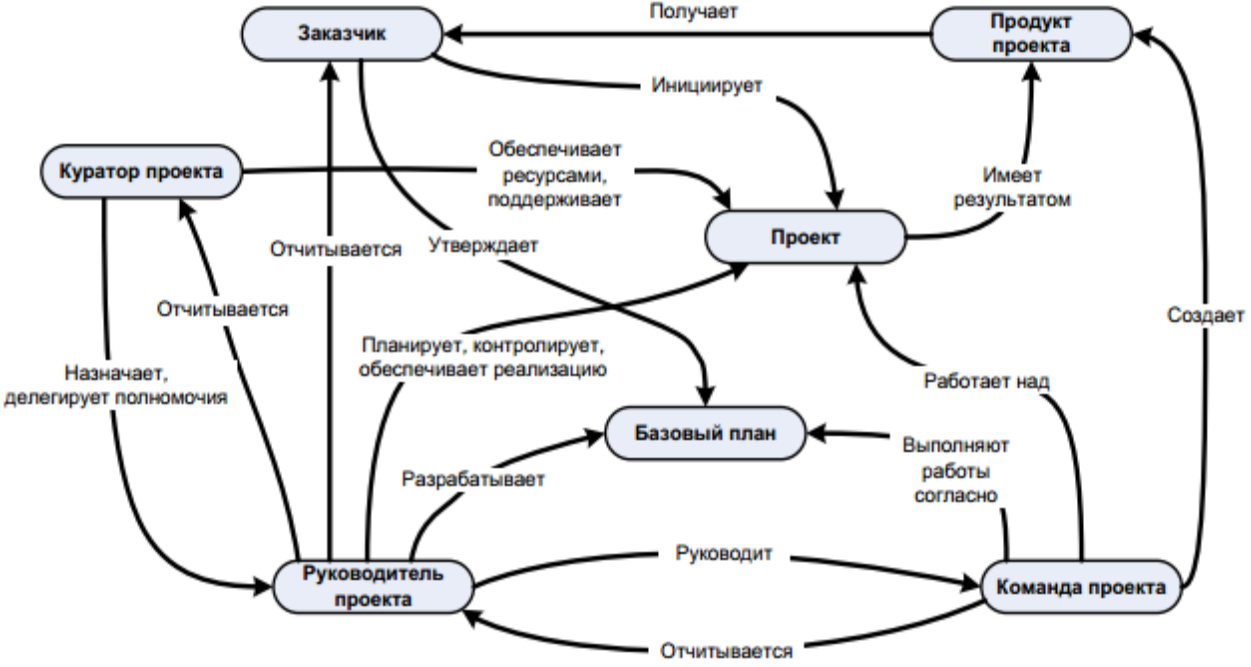
ПРИЛОЖЕНИЕ 6. МОДЕЛЬ RISK IT



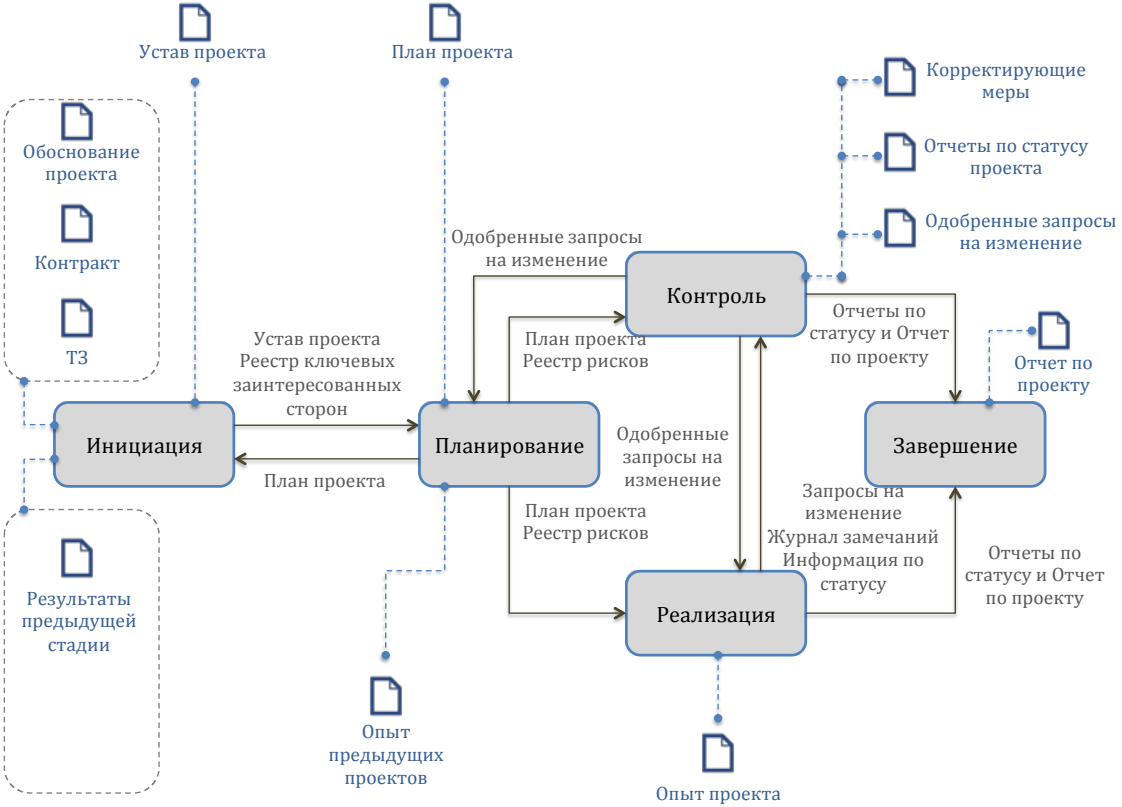
ПРИЛОЖЕНИЕ 7. МОДЕЛЬ УРОВНЕЙ ЗРЕЛОСТИ CMMI



ПРИЛОЖЕНИЕ 8. **ГОСТ 54869 ОСНОВНЫЕ ПОНЯТИЯ ПРОЕКТНОГО МЕНЕДЖМЕНТА И ИХ ВЗАИМОСВЯЗЬ**



ПРИЛОЖЕНИЕ 9. ISO 21500 ВЗАИМОДЕЙСТВИЕ ГРУПП ПРОЦЕССОВ, ИХ ВХОДЫ И ВЫХОДЫ



ПРИЛОЖЕНИЕ 10. ISO 21500 ПРОЦЕССЫ УПРАВЛЕНИЯ ПРОЕКТАМИ ПО ПРЕДМЕТНЫМ ГРУППАМ

Предметная группа	Группы процессов				
	Инициирование	**Планирование**	**Исполнение**	**Управление**	**Завершение**
Интеграция	4.3.2 Разработка устава проекта	4.3.3 Разработка планов проектов	4.3.4 Непосредственная работа по проекту	4.3.5 Управление проектными работами	4.3.7 Закрытие отдельной фазы или проекта
				4.3.6 Управление изменениями	4.3.8 Извлеченные уроки
Заинтересованные стороны	4.3.9 Определение заинтересованных сторон		4.3.10 Управление заинтересованными сторонами		
Содержание		4.3.11 Определение содержания проекта		4.3.14 Управление содержанием проекта	
		4.3.12 Создание структуры декомпозиции работ			
		4.3.13 Определение состава работ			
Ресурсы	4.3.15 Создание команды проекта	4.3.16 Оценка ресурсов	4.3.18 Развитие команды проекта	4.3.19 Управление ресурсами	
		4.3.17 Определение организационной структуры проекта		4.3.20 Управление командой проекта	
Время		4.3.21 Последовательность работ		4.3.24 Управление расписанием	
		4.3.22 Оценка длительности работ			
		4.3.23 Разработать расписания			
Стоимость		4.3.25 Оценка затрат		4.3.27 Управление затратами	
		4.3.26 Разработка бюджета			
Риски		4.3.28 Определение рисков	4.3.30 Отношение к рискам	4.3.31 Управление рисками	
		4.3.29 Оценка рисков			
Качество		4.3.32 План по качеству	4.3.33 Обеспечение требований качества	4.3.34 Управление качеством	
Поставки		4.3.35 План поставок	4.3.36 Выбор поставщиков	4.3.37 Администрирование контрактов	
Коммуникации		4.3.38 План коммуникаций	4.3.39 Распространение информации	4.3.40 Управление коммуникациями	

ПРИЛОЖЕНИЕ 11. ТИПОВЫЕ ШАБЛОНЫ ИТОГОВЫХ ДОКУМЕНТОВ

ОТЧЕТ МЕНЕДЖЕРА ПРОЕКТА

<table>
<tr><td>Проект:</td><td colspan="2">Менеджер проекта:</td><td colspan="2">Неделя №
Отчетный период
с_____ по_____</td></tr>
<tr><td colspan="3">Ключевые вопросы:
1. Изменился ли состав задач?
2. Будут ли планируемые сроки нарушены?
3. Существуют ли проблемы в планировании?
4. Существуют ли технические проблемы?
5. Существуют ли проблемы рецензий и утверждений?
6. Существуют ли проблемы с оплатой по договору?</td><td>Да
----</td><td>Нет
-------</td></tr>
<tr><td colspan="5">Объяснения по вопросам, отмеченным «Да»:</td></tr>
<tr><td colspan="5">Достижения прошлой недели:</td></tr>
<tr><td colspan="5">Планируемые достижения следующей недели:</td></tr>
<tr><td colspan="5">Проблемы и рекомендуемые / или планируемые решения:</td></tr>
<tr><td>Составлено:</td><td>Дата:</td><td>Принял:</td><td colspan="2">Дата:</td></tr>
</table>

ПРОТОКОЛ ЗАСЕДАНИЯ КООРДИНАЦИОННОГО СОВЕТА

Протокол № ___ от _____

совещания Координационного совета

Председатель (Ф.И.О.):		Секретарь (Ф.И.О.):	
Дата/время/место проведения совещания:			
Присутствовали (Ф.И.О.):	От Заказчика: От Исполнителя:		
Отсутствовали (Ф.И.О.):	От Заказчика: От Исполнителя:		

Цели совещания:

1.
2.
3.

№	Решение/следующий шаг	Кому поручено	Дата выполнения
1			
2			

№	Повестка следующего совещания	Участники	Время
1			
2			

Председатель Координационного совета
_____ (должность, Ф.И.О.)
(подпись)

Секретарь совещания
_____ (должность, Ф.И.О.)
(подпись)

РЕГУЛЯРНЫЙ ОТЧЕТ О СТАТУСЕ ПРОЕКТА

Проект:	**Консультант: (Ф. И. О.)**		**Отчетный период: с _____ по _____**
Статус	**Да**	**Нет**	**Объяснение**
Рамки проекта			
Изменились ли задачи проекта (этапа, подэтапа)?			
Изменились ли выходные результаты проекта (этапа, подэтапа)?			
Изменились ли логические рамки проекта (этапа, подэтапа)?			
План работ			
Выйдут ли работы за рамки ожидаемого окончания проекта (этапа, подэтапа)?			
Отличается ли оценка времени завершения (ОВЗ) более чем на 10% от согласованного плана работ?			
Работа / Люди			
Есть ли какие-нибудь значительные проблемы?			
Есть ли внешние факторы, влияющие на проект?			
Есть ли проблемы с проверкой и утверждением рабочих продуктов?			
Есть ли какие-нибудь проблемы, мешающие проектной команде успешно работать?			
Есть ли какие-нибудь проблемы с персоналом?			
Сделано в этом периоде:			
Будет сделано в следующем периоде:			

Сводка по команде:	Текущий период:		С начала проекта:	
	Часы по Плану:	Факт. Часы:	Часы по Плану:	Факт. Часы:

Выходные результаты:	Должны быть подготовлены	Фактически подготовлены	Планируется подготовить

Оценка для завершения проекта (этапа, подэтапа):

_____ (Ф. И. О.)
(подпись)

ПРОТОКОЛ ВЫПОЛНЕННЫХ РАБОТ ПО МОДУЛЮ СИСТЕМЫ

Протокол № __ выполненных работ по модулю «_____» системы «__» _____ 20__ г.

1. Результаты выполнения работ по предыдущему протоколу № __ от «__» _____ 20_ _ года:

№ п/п	Перечень работ к выполнению	Отметка о выполнении
	Выполнено специалистами Заказчика:	
1.		
2.		
…		
	Выполнено специалистами Исполнителя:	
1.		
2.		
…		

2. Дополнительно выполнено в ходе текущей встречи:
   1.
   2.
3. Необходимо выполнить:

№ п/п	Перечень работ к выполнению	Срок выполнения
Специалистам Заказчика:		
1.		
2.		
…		
Специалистам Исполнителя:		
1.		
2.		
…		

От Заказчика:
_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:
_____ (должность, Ф.И.О.)
(подпись)

АКТ ПРИЕМА-СДАЧИ ВЫПОЛНЕННЫХ РАБОТ
Акт приема-сдачи выполненных работ по _____ (Вид работ)

г. Москва «__» _____ 20_ г.

Мы, нижеподписавшиеся, от лица **ИСПОЛНИТЕЛЯ** (Должность, Ф.И.О.) и от лица **ЗАКАЗЧИКА** (Должность, Ф.И.О.), составили настоящий Акт в том, что в соответствии с п. ___ Договора Исполнитель сдает, а Заказчик принимает работы по _____
_____

(информационному обследованию предприятия Заказчика).
(инсталляции системы «_____» на стенд предприятия Заказчика).
(доработке и внедрению программного модуля «_____»).
(постгарантийному сопровождению модуля «_____»).
(оказанию информационно-консультационных услуг сотрудникам Заказчика по обучению основам работы с системой «_____»).

Услуги оказаны в полном объеме, программный модуль «_____» полностью соответствует требованиям технического задания. Работы по п. ___ Договора в части _____ (вид работ) считаются выполненными и принятыми Заказчиком. Стороны претензий друг к другу не имеют.
Сумма по п. ___ Договора в части доработки и внедрения программного модуля «_____» составляет _____ руб. (_____ рублей _____ копеек), в том числе НДС 18% в размере _____ руб. (_____ рублей ___ копеек).
Настоящий Акт является основанием для проведения расчетов по Договору № _____ от «___» _____ 20__ года.

Печати и подписи сторон:

ЗАКАЗЧИК:	ИСПОЛНИТЕЛЬ:
(Должность)	(Должность)
_____ (Ф.И.О.)	_____ (Ф.И.О.)

ФОРМА РЕГИСТРАЦИИ ПРОБЛЕМЫ

<table>
<tr><td colspan="5">Проект: Внедрение системы «_____» в подразделении _____ Заказчика</td><td colspan="4">№ Проблемы:</td></tr>
<tr><td colspan="5" rowspan="2">Проблема:</td><td colspan="4">Тяжесть:</td></tr>
<tr><td>Высокая</td><td colspan="2">Средняя</td><td>Низкая</td></tr>
<tr><td colspan="2">Кем обнаружено:</td><td colspan="3">Кому поручено:</td><td colspan="4">Дата:</td></tr>
<tr><td colspan="9">Описание проблемы:</td></tr>
<tr><td colspan="9">Решение / план действий:</td></tr>
<tr><td colspan="2">Кто принимает решение:</td><td colspan="2">Утверждено:</td><td colspan="2">Ожидаемая дата исполнения решения:</td><td colspan="3">Дата исполнения:</td></tr>
<tr><td>Статус</td><td colspan="5">Открыта</td><td colspan="3">Закрыта</td></tr>
<tr><td>Подчеркнуть</td><td>Поставлена</td><td>Принята</td><td>Отложена</td><td>Оценена</td><td>Одобрена</td><td>Решена</td><td>Отклонена</td><td>Объединена</td></tr>
</table>

От Заказчика:

_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:

_____ (должность, Ф.И.О.)
(подпись)

ЖУРНАЛ РЕГИСТРАЦИИ ПРОБЛЕМ

Проект: Внедрение системы «_____» в подразделении _____ Заказчика		Кем подготовлено:	Дата:	Стр.1 Всего стр. 1	
Код	Описание проблемы	Важность	Дата открытия	Статус	Дата закрытия

От Заказчика:

_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:

_____ (должность, Ф.И.О.)
(подпись)

ГРАФИК ПРОВЕДЕНИЯ РАБОТ ПО ВНЕДРЕНИЮ СИСТЕМЫ

График проведения работ

по внедрению системы «_____» на _____

(наименование предприятия Заказчика)

«УТВЕРЖДАЮ»
Генеральный директор
(наименование предприятия Заказчика)

_____ Ф.И.О.
«___» _____ 20_ года

№ п/п	Наименование модуля /перечень работ по внедрению	Количество дней	Начало работ	Окончание работ	Состав группы от Заказчика	Состав группы от Подрядчика	Примечание
1.	**Модуль № 1:**						
1.1	– работа № 1;						
1.2	– работа № 2;						
…	…						
2.	**Модуль № 2:**						
2.1	– работа № 1;						
2.2	– работа № 2;						
…	…						

На весь период внедрения модулей системы «_____» в состав Группы внедрения от Подрядчика могут дополнительно привлекаться специалисты аудиторских и консалтинговых компаний.

От Заказчика:

_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:

_____ (должность, Ф.И.О.)
(подпись)

ИНДИВИДУАЛЬНЫЙ ОТЧЕТ О ПРОРАБОТАННОМ ВРЕМЕНИ
Индивидуальный отчет о проработанном времени

Проект:	Консультант-программист:	Неделя №: Отчетный период с_____по_____

№ п/п	Наименование задачи	Время	Что было сделано
1.			
	Итого за период:		

Подготовил:	Дата:	Утвердил:	Дата:

ОТЧЕТ О РЕЗУЛЬТАТАХ ЭТАПА
Отчет о результатах этапа

Проект: Внедрение системы «_____» в подразделении _____ Заказчика	Менеджер проекта:	Дата:
Оценка выполнения этапа:		
Дополнительные комментарии:		
Визы: Менеджер проекта от Заказчика: Координатор проекта предприятия Заказчика:		

Отчет о результатах этапа – отклонения

Проект: Внедрение системы «_____» в подразделении Заказчика	
Дата:	Подготовлено менеджером проекта:
Перечисляются отклонения от стандартных задач и действий и прилагается описание не стандартных задач и действий, возникших на данном этапе:	
Отклонения от плана (обсуждаются причины значительных отклонений плана от факта):	
Предложения по улучшению стандартной методологии:	

ПРИКАЗ О ПРОВЕДЕНИИ ПРОЕКТА
ПО ИНФОРМАЦИОННОМУ ОБСЛЕДОВАНИЮ
БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ ЗАКАЗЧИКА

Приказ о проведении проекта по информационному обследованию бизнес-процессов предприятия заказчика

«__» _____ 20_ года г. Москва № ___

В целях обеспечения эффективного выполнения проекта по информационному обследованию бизнес-процессов предприятия Заказчика

ПРИКАЗЫВАЮ:

1. Совместно с компанией ООО «_____» (наименование компании-подрядчика) с «__» _____ 20__ года приступить к осуществлению проекта по информационному обследованию бизнес-процессов предприятия.
2. Создать Координационный совет проекта в составе следующих руководителей подразделений Заказчика _____:

Задачи Координационного совета проекта:

- утверждает результаты работы Рабочей группы и все основные документы проекта;
- утверждает решения проблем, возникающих в ходе проекта, или решает эти проблемы;
- утверждает изменение рамок проекта.

3. Назначить Координатором проекта от Заказчика (должность, Ф.И.О.) _____. Его права и обязанности:

- принимает оперативные административные решения, а также добивается исполнения административных решений Координационного совета;
- распоряжается выделенными на проект фондами;
- утверждает изменение рамок проекта, обеспечивая при необходимости дополнительное финансирование;
- принимает результаты проекта;
- выносит возникшие проблемы на Координационный совет и контролирует выполнение принятых соответствующих постановлений;
- назначает ответственных исполнителей со стороны Заказчика;
- отвечает за согласование действий различных организаций и групп, участвующих в проекте, отслеживает график работ и своевременность реализации принятых решений;
- проводит в жизнь административные решения Координационного совета, касающиеся подразделений Заказчика, руководит оперативным планированием загрузки сотрудников подразделений Заказчика и принимает решение об их премировании.

4. Назначить Руководителем проекта от Заказчика (Должность, Ф.И.О.) _____. Руководитель проекта от Заказчика является заместителем Руководителя проекта от компании _____ (наименование компании-подрядчика), руководит работой Рабочей группы и обязан:

- разрабатывать сценарии и планы проекта, а также процедуры принятия решений и стандарты оценки достигнутых результатов;
- участвовать на уровне Координационного совета в принятии стратегических решений по ходу выполнения проекта;
- представлять все результаты работы Рабочей группы;
- осуществлять контроль качества ведения проекта;
- контролировать точное исполнение условий договора со стороны Заказчика;
- принимать решение о приостановке и возобновлении работ по проекту;
- регулярно отчитываться перед Координатором проекта о состоянии дел на проекте, при необходимости – представлять корректировки бюджета и графика внедрения Координационному совету и Рабочей группе.

Руководитель проекта от Заказчика несет ответственность за:

- материально-техническое и организационное обеспечение работ по проекту;
- обеспечение обязательного присутствия требуемых специалистов предприятия Заказчика на рабочих местах во время реализации проекта;
- создание сплоченной, работоспособной Рабочей группы и укрепление у нее уверенности в конечном успехе проекта (привлечение всех членов Группы к обсуждению текущих вопросов, периодическая замена людей, которые возглавляют обсуждение);
- своевременное представление проблем Рабочей группы перед вышестоящим руководством и руководителями других подразделений;
- оказание помощи в идентификации требований и проблем проекта;
- обеспечение наличия и работоспособности компьютеров, принтеров, сети и другого необходимого на проекте оборудования;
- установку и настройку необходимого общего и специального программного обеспечения на рабочих местах специалистов Рабочей группы.

5. Сформировать Рабочую группу от Заказчика в составе:

Задачи Рабочей группы:

- участие в работах по сбору информации;

- обеспечение подготовки и корректировки всей документации;
- контактировать с консультантами и разработчиками ___ _____ (наименование компании-подрядчика), по вопросам сбора информации.

6. Рабочей документацией проекта являются:
- Протокол заседания Рабочей группы. Составляется секретарем заседания, определяемым в рабочем порядке;
- Листы передачи, на основании которых Исполнитель передает Заказчику документы, являющиеся результатами работы по этапам проекта;
- Листы замечаний, заполняемые сотрудниками Заказчика в ходе ознакомления с итоговыми документами по проекту, представленными Исполнителем, и являющиеся основанием для внесения в них корректировок и исправлений;
- Протокол совещания Координационного совета. Составляется секретарем совещания, определяемым в рабочем порядке;
- Акт (рабочий протокол) приема-сдачи выполненных работ, отражающий состояние дел по проекту и подписывается Координатором проекта от Заказчика. Составляется Руководителем проекта от _____ (наименование компании-подрядчика) по выполнении какого-либо объема работ.

7. Руководителям структурных подразделений, служб, отделов Заказчика все поручения Руководителя проекта, связанные с исполнением плана по проведению информационного обследования бизнес-процессов предприятия, считать обязательными к исполнению.

8. Предоставить консультантам от _____ (наименование компании-подрядчика) рабочее помещение, требуемое количество компьютеров, оргтехнику, средства связи на время проведения работ по информационному обследованию.

Генеральный директор _____
(наименование предприятия Заказчика)

_____ (Ф.И.О.)

АНКЕТА ЗНАКОМСТВА С ПРЕДПРИЯТИЕМ
Инструкция по заполнению анкеты

Анкета предназначена для предварительного определения проблематики и потребностей развития Вашей компании. По результатам анализа анкеты мы сможем понять общий круг задач, которые Вы хотели бы решить и предложить Вам несколько вариантов совместной работы для их решения, из которых Вы сможете выбрать наиболее выгодный для Вас.

Внимательное заполнение данной анкеты поможет Вам лучше структурировать те проблемы, которые могут быть решены с нашей помощью.

Вы можете не заполнять анкету, если считаете, что ответы на вопросы анкеты затрагивают конфиденциальную информацию о Вашей компании.

Вы можете ответить лишь на те пункты, которые устраивают Вас. Кроме того, некоторые вопросы или варианты ответов могут быть не применимы к Вашей фирме. Пропускайте их при заполнении.

Если какие-то пункты анкеты не до конца понятно сформулированы или выбор варианта ответа не однозначен, укажите номера вопросов в примечании в конце анкеты, и мы обсудим их при личной встрече.

Пожалуйста, приложите к анкете схему и Положение об организационной структуре Вашего предприятия.

Благодарим за заполненную анкету!

1. Информация о Предприятии.

1.1. Сколько лет на рынке существует Предприятие?
_____.

1.2. Укажите основные направления деятельности Предприятия (если несколько, то указать все):

☐ торговля ☐ строительство
☐ производство ☐ оказание услуг
☐ финансы ☐ другое _____

1.3. Сколько продуктов – товаров и / или услуг – производит / реализует Предприятие в настоящее время?
_____

1.4. Какие виды деятельности в настоящее время являются прибыльными?
1
2
3

1.5. Есть ли перспективные направления деятельности, которые не приносят прибыль в настоящее время, но она запланирована в будущем? Если «да», то перечислить их.
1
2
3

1.6. Есть ли бизнес-план развития перспективных направлений?
☐ да
☐ нет
☐ в процессе разработки

1.7. Есть ли «убыточные» направления деятельности, которые не приносят прибыль в настоящее время, и Вы сомневаетесь, в их перспективе? Если «да», то перечислить их.
1
2
3

1.8. Укажите причины не закрытия «убыточных» направлений?

1

2

3

1.9. Какой уровень управления характерен для компании?
□ одно юридическое лицо с единоличным руководителем
□ одно юридическое лицо с коллегиальным руководством
□ несколько юридических лиц с управляющей компанией во главе
□ группа юридических лиц без управляющей компании

1.10. Какая структура собственности на Предприятии?
□ вся собственность (или контрольный пакет) принадлежит одному человеку
□ есть несколько человек, владеющих контрольным пакетом совместно
□ контрольный пакет принадлежит другой организации
□ открытое акционерное общество без контрольного пакета

1.11. Есть ли обособленные подразделения Предприятия (филиалы, представительства и т.п.) в других городах? Если «да», то указать города.
1

2

3

2. Система управления Предприятием.
Организационная структура.

2.1. Есть ли на Предприятии утвержденная и документально закрепленная организационная структура?
□ да
□ нет

2.2. Уровень распределения ответственности среди высшего руководства:
□ высокий (четко оговорены и зафиксированы в документах зоны ответственности)
□ средний (оговорены, но не зафиксированы, есть дублирование или зоны «безответственности»)
□ низкий (ответственность не распределена)

2.3. Функциональная нагрузка, зоны ответственности высшего и среднего руководства закреплены:

□ в Положениях об отделах / службах
□ в должностных инструкциях
□ в приказах
другое (указать)
_____

2.4. Положения о подразделениях / службах / отделах и должностные инструкции являются для сотрудников:
□ руководством к действию
□ формальным документом
□ справочным материалом, требующих существенных изменений
□ не используются вообще

2.5. Уровень нагрузки высшего руководства:
□ высокий (работают без выходных по 10-12 часов)
□ средний (работают нормальный рабочий день с редкими авралами)
□ низкий (авралы бывают крайне редко)

2.6. Схема взаимодействия между подразделениями / отделами / службами:
□ определена и закреплена в Положении об оргструктуре / Положениях о подразделениях
□ определена и действует в силу сложившихся традиций, но не закреплена документально
□ не определена и не закреплена документально

2.7. Проблемные зоны при взаимодействии подразделений между собой:
1
2
3

2.8. Контроль со стороны руководства за исполнением подразделениями своих функций:
□ подразделения регулярно готовят для руководства письменные отчеты о проделанной работе
□ отчеты о проделанной работе носят устный характер
□ контроль не регулярный, отчеты готовятся по требованию
□ контроля нет

2.9. Организационная структура Предприятия:

□ обновляется регулярно в зависимости от изменений структуры бизнеса или требований рынка
□ структура устоявшаяся, пересматривается крайне редко
□ не обновляется вообще

2.10. При появлении новых функций:
□ создается специально новое подразделение
□ функции распределяются между существующими подразделениями
другое (указать)
_____

2.11. Основные функции управления персоналом, существующие сейчас на Предприятии:
□ планирование потребностей в персонале
□ определение квалификационных требований
□ определение уровня и типа оплаты
□ сбор и анализ информации по кадрам
□ поиск и подбор персонала
□ заключение контракта
□ постановка задач и управление сотрудниками
□ адаптация новых сотрудников
□ первичное обучение
□ формирование мотивации
□ оценка результатов работы сотрудника
□ регулярная аттестация персонала
□ формирование компенсационного пакета

Планирование и контроль.

2.12. Есть ли Бизнес-план и / или стратегия развития Предприятия?
□ да
□ нет

2.13. Периоды стратегического планирования:
□ на три-пять лет
□ на 1-2 года
□ на полгода-год

2.14. Стратегический план развития закрепляется:
□ бизнес-планом
□ протоколом собрания совета директоров (акционеров)

□ устным обсуждением стратегического плана развития без документального оформления

2.15. Системность и четкость целей предприятия:
□ высокая (взаимосвязаны и просчитаны)
□ средняя (взаимосвязаны, могут быть проконтролированы по косвенным параметрам)
□ низкая (противоречат друг другу, не конкретны и не контролируемы)

2.16. Использование стратегических целей в оперативном управлении:
□ высокое (используются при постановке своих оперативных задач)
□ среднее (используются при разработке планов работ)
□ низкое (практически не используются в работе)

2.17. Понимание целей Предприятия персоналом:
□ высокое (понимают согласованно)
□ среднее (понимают, но каждый по-своему)
□ низкое (не понимают)

2.18. Период оперативного планирования:
□ на год
□ на полугодие-квартал
□ на месяц

2.19. Результаты оперативного планирования закрепляются:
□ бюджетом предприятия
□ прогнозами доходов и расходов, поступающими из разных подразделений
□ документально не оформляется

2.20. Бюджет предприятия состоит из:
□ бюджета продаж
□ производственной программы
□ бюджета закупки и использования материалов
□ бюджета трудовых затрат
□ бюджета общепроизводственных расходов
□ бюджета коммерческих расходов
□ бюджета общехозяйственных расходов
□ бюджета доходов и расходов
□ бюджета движения денежных средств
другое (указать)
_____

2.21. Результаты планирования в оперативном управлении используются:
□ как документ, регламентирующий оперативную деятельность
□ как справочная информация
□ практически не используются

2.22. Выполнение планов на Предприятии:
□ контролируется выделенным подразделением, осуществляющим контроль над выполнением планов как по всему Предприятию в целом, так и в разрезе конкретных подразделений
□ руководителями подразделений контролируется исполнение планов, относящихся непосредственно к их подразделениям
□ контролируется высшим руководством по мере необходимости
□ никем не контролируется

2.23. Контроль над выполнением планов осуществляется:
□ на стадии оперативного управления (ежедневно, еженедельно)
□ стадии получения фактических данных о деятельности за период
□ контроль не осуществляется

2.24. Методика планирования и контроля над выполнением планов:
□ существует формализованная методика составления планов и контроля над их выполнением
□ планы составляются и периодически контролируются, но методика не формализована
□ планы составляются по требованию руководства без применения определенной методики

2.25. Соответствие методики формирования планов методикам формирования отчетов по фактическим данным:
□ методика формирования плановых и фактических показателей идентична
□ приходится постоянно перегруппировывать фактические данные в соответствии с методиками планирования
□ методики совершенно разные и не стыкуются между собой

2.26. Источником информации об исполнении планов являются:
□ учетные регистры бухгалтерского учета
□ формы внутренней управленческой отчетности
□ устные отчеты ответственных сотрудников

Учет.

2.27. Учетная политика на Предприятии отражает:

□ формальные требования законодательства
□ требования законодательства и потребности руководства Предприятия

2.28. Есть ли на Предприятии система управленческого учета?
□ да
□ нет

2.29. Система управленческого учета:
□ разработаны и документально закреплены регламенты управленческого учета
□ управленческая информация предоставляется по требованию и в запрашиваемых разрезах
□ информация берется только в существующих разрезах бухучета

2.30. Управленческая информация передается:
□ в утвержденных формах
□ в произвольных формах
□ в устном виде

2.31. При передаче данных руководству в устной форме:
□ данные потом всегда проверяются
□ данные проверяются в случаях исключительной важности
□ не проверяются вообще

2.32. Учет и оценка результатов деятельности Предприятия:
□ ежедневно или еженедельно
□ ежемесячно или ежеквартально
□ реже чем раз в квартал

2.33. Детальность и достоверность учетных данных:
□ высокая (по всем параметрам и на основе автоматизированного учета)
□ средняя (только по основным параметрам, на основе документов)
□ низкая (по тем параметрам, которые легче подсчитать или на основе устной информации)

2.34. Оформление, передача и хранение управленческой информации осуществляется:
□ регламентировано и систематично проводится
□ не регламентировано, но проводится почти систематично
□ проводится в случайном порядке

2.35. Оперативность и контролируемость управляющих воздействий
□ распоряжения выполняются точно в срок, качество выполнения контролируется
□ сроки не редко нарушаются, контроль проводится не систематически
□ сроки не соблюдаются, контроль практически не проводится

2.36. Существуют ли меры ответственности работников за своевременное предоставление информации?
□ да
□ нет

Документооборот.

2.37. Регламент документооборота на Предприятии:
□ утвержден и закреплен документально
□ устоявшийся
□ регламента нет, документы ходят хаотично

2.38. Составляющие документооборота, охваченные регламентом:
□ бухгалтерский учет
□ управленческий учет
□ производство
□ снабжение / сбыт
□ планирование
другое (указать)
_____

2.39. Контроль над выполнением регламента осуществляется:
□ генеральным директором
□ зам. директора по общим вопросам
□ каждым руководителем непосредственно в своем подразделении
□ никем
другое (указать)
_____

2.40. Существуют ли меры ответственности за неисполнение сотрудниками регламента документооборота?
□ да
□ нет

2.41. Существующий на предприятии регламент документооборота является для сотрудников:

□ руководством к действию
□ справочной информацией
□ не используется в работе

Снабжение, сбыт.

2.42. Охарактеризуйте свои каналы сбыта:
□ прямой сбыт
□ филиалы
□ торговые представители
□ дилерская сеть
□ выставки-ярмарки
другое (укажите)
_____

2.43. Изменения объема продаж за последние 3 года:
20____г.
_____
20____г.
_____
20____г.
_____

2.44. События, повлиявшие на изменение объема продаж с января 20__г. по ____ 20__г.:
чрезвычайные (укажите какие)
_____
непредвиденные (укажите какие)
_____
сезонные (укажите какие)
_____
другое (укажите что)
_____

2.45. Структура портфеля продукции Предприятия:
1
2
3
4

2.46. Опишите ситуацию с заказами на Предприятии в настоящий момент:
_____

_____
_____
_____
_____
_____

2.47. Размеры групп клиентов Предприятия и их относительная доля в объеме продаж:

1
2
3
4

2.48. Сбор и анализ данных о рынке:

□ собираются в обязательном порядке все данные о потребителях, поставщиках, конкурентах
□ собираются обязательно только данные о конкурентах
□ собираются обязательно только данные о потребителях
□ собираются обязательно только данные о поставщиках
□ не собираются вообще никакие данные
□ данные собираются во фрагментарном виде (в каком конкретно указать)
_____

2.49. Информация о рынке:

□ централизованно собирается и хранится в ответственном подразделении
□ собирается и хранится в разных подразделениях
□ не хранится вообще

2.50. Полученная информация о рынке используется руководством Предприятия:

□ всегда при принятии оперативных и стратегических решений
□ только при разработке бизнес-плана или стратегии развития предприятия
□ только как справочная информация
□ не используется вообще

2.51. План продаж составляется:

□ с учетом использования данных о рынке и предшествующей деятельности
□ интуитивно
□ спускают сверху
□ не составляется вообще

2.52. Поставщики основного сырья:
□ сформирован круг постоянных поставщиков, работаем только с ними, в результате чего имеем выгодные условия закупки
□ закупаем как у постоянных, так и сторонних поставщиков, в зависимости от того, где дешевле
□ постоянного круга поставщиков нет, закупаем там, где дешевле

2.53. План мероприятий по продвижению продукции и стимулированию продаж:
□ разрабатывается регулярно на основе анализа рынка и является руководством к действию
□ разрабатывается в случаях крайней необходимости (напр., при резком падении объема продаж)
□ разрабатывается по требованию руководства и является чисто формальным документом
□ документально оформленного плана нет, действуем интуитивно

2.54. Система скидок:
□ разработана, официально утверждена и используется гибкая система скидок
□ скидки действуют только для постоянных клиентов
□ скидки предоставляются только по просьбе клиента
□ скидок нет

2.55. Практика ценообразования на Предприятии:
□ цены рассчитываются с учетом анализа рынка и конкурентов
□ цены спускают сверху
□ цены рассчитываются интуитивно
другое (указать)
_____

2.56. Отгрузка продукции:
□ схема отгрузки утверждена, четко функционирует и удобна для клиентов
□ из-за удаленности торговых и складских помещений друг от друга клиент тратит много времени на выписку и получение товара
□ схема отгрузки не отрегулирована до конца, есть зоны «безответственности» и дублирования операций подразделениями, клиент тратит много времени на оформление документов и получение товара
другое (указать)
_____

2.57. Информация о движении товарно-материальных ценностей:

□ разработан и утвержден регламент обмена оперативной информацией о движении товара, остатках на складах между подразделениями
□ устоявшийся порядок обмена информацией на складах не отражает потребности подразделений в оперативном получении данных о получении оплаты, движении товара, остатках на складах
□ нет четкого и утвержденного регламента обмена информацией о движении товара, остатках на складах, информация запрашивается по мере необходимости
другое (указать)
_____

Автоматизация.

2.58. Автоматизированные участки на предприятии:

□ ТМЦ
□ бухгалтерия
□ заработная плата
□ договорная работа
□ финансы и расчеты
□ бюджетирование
□ касса, банк
□ кадры
□ производство
□ планирование
□ управленческий учет
другое (указать)
_____

2.59. Информационная система на Предприятии представляет:
□ несвязанные блоки, возможен повторный ввод информации в необходимые блоки для получения реальной картины деятельности
□ несколько участков с разрозненными базами, связанных между собой посредством сеансов экспорта-импорта
□ единое информационное пространство деятельности предприятия в реальном времени

2.60. Существующее программное обеспечение на Предприятии:
□ свои разработки
□ среда разработки
_____
с какого времени в эксплуатации
_____
приобретенное (какое указать)
_____

2.61. Поддержка информационной системы, ее развитие и изменение (при необходимости) осуществляется:
□ силами отдела АСУ Предприятия
□ сторонними консультантами
□ организацией-разработчиком программного обеспечения

2.62. Компьютеры объединены в локальную сеть:
□ да
□ нет

2.63. Операционная система:
□ Windows
□ UNIX (Linux)
другое (указать)
_____

2.64. Используемые форматы баз данных:
□ DBF
□ Oracle
□ Access
SQL (какой)
_____
другое (указать)
_____

2.65. Рабочие характеристики:
количество пользователей (всего)_____, активных _____

2.66. Оснащенность ПК:
сервер (количество, характеристики) _____
клиентские машины (число): _____
характеристики сети: _____

2.67. Основные недостатки существующей информационной системы на Предприятии и причины их возникновения:
_____
_____
_____
_____
_____

2.68. Ваши требования к «идеальной» информационной системе для Вашего Предприятия:
_____
_____
_____
_____

3.1. Укажите в порядке значимости цели, которые Вы хотите достичь в ближайшее время:

1
2
3
4

3.2. Укажите в порядке важности проблемы, сдерживающие динамичное развитие Предприятия:
1
2
3
4

4. Дополнительные пожелания Заказчика.

4.1. Дополнительная информация о Предприятии, которая не была освещена в анкете, но является важной, на Ваш взгляд:
_____
4.2. Первоочередные и дополнительные задачи по консультированию:
1
2
3
4

5. Примечание.

5.1. Укажите направления, по которым Вам требуется помощь консультантов:
1
2
3
4

5.2. Укажите те пункты анкеты, заполнение которых вызвало сомнения или затруднения, и их следует обсудить с консультантами:

6. Приложение.

Данные о сотрудниках, участвующих в заполнении анкеты.

Ф. И. О. _____
Подразделение: _____
Должность: _____
Телефон: _____
Пункты, на которые были даны ответы (указать): _____

АНКЕТА ДЛЯ ВЫСШЕГО РУКОВОДСТВА ПРЕДПРИЯТИЯ
Инструкция по заполнению анкеты

Настоящая анкета предназначена для определения проблемных зон в функциональном взаимодействии подразделений и информационном обмене, затрудняющих процесс принятия управленческих решений для руководства Предприятия.

Настоящая анкета заполняется генеральным директором, финансовым директором, коммерческим директором, техническим директором.

Внимательное заполнение анкеты, а также предоставление полной и точной информации поможет консультантам лучше структурировать те проблемы на Предприятии, которые могут быть решены с их помощью. Ниже предлагаются несколько рекомендаций по заполнению анкеты.

П. 1. Необходимо полностью указать фамилию, имя, отчество.

П. 2. В этом пункте необходимо указать должность, которую вы занимаете в настоящее время согласно штатному расписанию Предприятия.

П. 3. Укажите Ваш контактный телефон.

П. 4. Здесь необходимо указать структурные подразделения и функциональных руководителей, которые в настоящий момент находятся непосредственно в Вашем прямом подчинении.

П. 5. Здесь необходимо перечислить подразделения, сотрудников (с указанием занимаемой должности), при взаимодействии с которыми у Вас возникают проблемные ситуации. Опишите подробно проблемы и причины их возникновения.

Например, проблемы могут быть следующего характера: дублирование функций; перегруженность работой; напряженность; не отрегулированный механизм обмена информацией, прохождения документов, заказов; недостаток информации для принятия оперативных решений по ряду вопросов (указать каким конкретно); сложности при запросе дополнительных данных; отсутствие оперативной информации; задержка в предоставлении отчетов, различных данных; предоставление некачественной информации (неточности, ошибки, произвольная форма) и т.п.

П. 6. В этом пункте необходимо указать всю информацию (название документов), которые Вы используете в настоящее время для принятия управленческих решений в области планирования, организации исполнения, контроля и анализа по каждому из бизнес-процессов, происходящих на предприятии. Укажите также периодичность и способ предоставления данных (устное сообщение на совещании, письменный отчет в произвольной форме, документ утвержденной формы с установленным регламентом предоставления). Оцените, пожалуйста, степень достоверности и оперативности получаемой Вами информации.

П. 7. В данном пункте укажите информацию, которую Вы бы хотели иметь при принятии решений по каждому из бизнес-процессов для планирования,

организации исполнения, контроля и анализа. Укажите желаемую периодичность, способ получения информации и причины недостатка данной информации в настоящий момент.

П. 8. В данном пункте укажите направления совершенствования системы управления на Вашем предприятии и расставьте их по степени приоритетности.

П. 9. Сформулируйте Ваши пожелания по совершенствованию системы управления и решению проблемных ситуаций.

Благодарим за заполненную анкету!

1) Ф. И. О. _____
2) Должность (согласно штатному расписанию) _____
3) Контактный телефон: _____
4) Какие структурные подразделения находятся в Вашем подчинении?
_____
5) С какими из подразделений (или функциональных руководителей) возникают проблемные ситуации (дублирование функций, отсутствие оперативной информации, напряженность и т.п.)? Опишите эти ситуации.
_____
_____
_____
6) Какая информация в настоящее время Вами используется для принятия управленческих решений?

Бизнес-процесс в разрезе функций управления	*Используемая информация*	*Периодичность предоставления*	*Способ предоставления (устная информация на совещаниях, письменные отчеты в произвольной форме, утвержденные формы документов)*	*Степень достоверности оперативности данных*
Продажи:				
Планирование				
Организация				
Контроль				
Анализ				
Производство:				
Планирование				
Организация				
Контроль				
Анализ				
Снабжение:				
Планирование				
Организация				
Контроль				
Анализ				
Финансы:				

Бизнес-процесс в разрезе функций управления	*Используемая информация*	*Периодичность предоставления*	*Способ предоставления (устная информация на совещаниях, письменные отчеты в произвольной форме, утвержденные формы документов)*	*Степень достоверности оперативности данных*
Планирование				
Организация				
Контроль				
Анализ				

7) Какой информации в настоящее время, по Вашему мнению, не хватает для повышения качества и обоснованности принимаемых решений?

Бизнес-процесс в разрезе функций управления	*Недостающая информация*	*Желаемая периодичность предоставления*	*Желаемый способ предоставления*	*Причина недостатка данной информации в настоящий момент*
Продажи:				
Планирование				
Организация				
Контроль				
Анализ				
Производство:				
Планирование				
Организация				
Контроль				
Анализ				
Снабжение:				
Планирование				
Организация				
Контроль				
Анализ				
Финансы:				

Бизнес-процесс в разрезе функций управления	*Недостающая информация*	*Желаемая периодичность предоставления*	*Желаемый способ предоставления*	*Причина недостатка данной информации в настоящий момент*
Планирование				
Организация				
Контроль				
Анализ				

8) Какие направления совершенствования системы управления, по Вашему мнению, являются наиболее важными, приоритетными (расставить по степени уменьшения приоритетности)?

1) _____
2) _____
3) _____
4) _____
5) _____

9) Ваши пожелания по совершенствованию системы управления и решению проблемных ситуации на предприятии.

_____
_____

АНКЕТА ДЛЯ РУКОВОДИТЕЛЯ ПОДРАЗДЕЛЕНИЯ
Инструкция по заполнению анкеты

Настоящая анкета предназначена для определения проблемных зон в информационном обмене и функциональном взаимодействии подразделений Предприятия.

Данная анкета заполняется руководителями функциональных подразделений Предприятия.

Внимательное заполнение анкеты, а также предоставление полной и точной информации поможет консультантам лучше структурировать те проблемы на Предприятии, которые могут быть решены с их помощью. Ниже предлагаются несколько рекомендаций по заполнению анкеты.

П. 1. Необходимо полностью указать имя, фамилию, отчество.

П. 2. Укажите Ваш контактный телефон.

П. 3. В этом пункте необходимо указать должность, которую вы занимаете в настоящее время согласно штатному расписанию Предприятия.

П. 4. Здесь необходимо указать структурное подразделение, которое Вы возглавляете в настоящий момент.

П. 5. В этом пункте укажите ***все*** функции возглавляемого Вами подразделения, вне зависимости от того закреплены ли они официально в Положении о подразделении или возложены на него в силу сложившихся традиций, возникших обстоятельств и т.п.

Функции подразделения необходимо расписать в разрезе функций управления: планирование, организация мероприятий, учет, контроль исполнения, анализ.

П. 6. Здесь указывается должность и Ф. И. О. Вашего непосредственного вышестоящего руководителя.

П. 7. Здесь необходимо указать имеющиеся в Вашем непосредственно подчинении участки, группы, бюро и т.п.

П. 8. Здесь необходимо перечислить подразделения, при взаимодействии с которыми у Вас возникают проблемные ситуации. Опишите подробно проблемы и причины их возникновения.

Например, проблемы могут быть следующего характера: дублирование функций; перегруженность работой; напряженность; не отрегулированный механизм обмена информацией, прохождения документов, заказов; недостаток информации для принятия оперативных решений по ряду вопросов (указать каким конкретно); сложности при запросе дополнительных данных; отсутствие оперативной информации; задержка в предоставлении отчетов, различных данных; предоставление некачественной информации (неточности, ошибки, произвольная форма) и т.п.

П. 9. В этом пункте укажите, пожалуйста, проблемы, которые препятствуют эффективности выполнения подразделением своих функций. Указанные проблемы необходимо разделить на три группы:

– *организационные* (например, дублирование функций, отсутствие регламентов обмена информацией, нечеткое распределение функциональной нагрузки и т.п.);
– *методические* (например, отсутствие методики распределения затрат, методики формирования себестоимости, методики планирования и т.п.);
– *технические (*например, отсутствие необходимых бах данных, программного обеспечения, разрозненность информации и т.п.).

П. 10. В данном пункте необходимо сформулировать три группы предложений по совершенствованию работы Вашего подразделения: организационные, методические и технические.

Благодарим за заполненную анкету!

1) Ф. И. О.: _____
2) Телефон: _____
3) Должность: _____
4) Структурное подразделение: _____
5) Функции структурного подразделения в разрезе функций управления:

Планирование	*Организация*	*Учет*	*Контроль*	*Анализ*
1.	1.	1.	1.	1.
2.	2.	2.	2.	2.
3.	3.	3.	3.	3.
4.	4.	4.	4.	4.
5.	5.	5.	5.	5.

6) Кому из вышестоящих руководителей Вы подчинены в общей структуре предприятия?

_____

7) Какие структурные подразделения находятся в Вашем подчинении?

_____

8) С какими из подразделений возникают проблемные ситуации (дублирование функций, отсутствие оперативной информации, напряженность и т.п.)? Опишите эти ситуации.

_____
_____
_____
_____

9. Перечислите, пожалуйста, основные проблемы, мешающие своевременности и эффективности выполнения функций подразделения?
Организационные:

_____

Методические:

_____

Технические:

_____

9. Предложения по совершенствованию работы Вашего подразделения.
Организационные:

_____

Методические:

_____

Технические:

_____

ОТЧЕТ О РЕЗУЛЬТАТАХ ИНФОРМАЦИОННОГО ОБСЛЕДОВАНИЯ
Отчет о результатах информационного обследования

1. Общие положения.
Информация о датах проведения обследования, договоре, целях работ, участниках проекта, источниках информации, о методах сбора информации и схеме анализа.
2. Результаты обследования.
2.1. Описание структуры управления предприятия:
2.1.1. Краткая характеристика объекта.
Информация об истории организации предприятия, формирования системы управления, основных функциях, среднемесячных объемах производства и реализации товаров и услуг.
2.1.2. Организационная структура предприятия.
Организационная схема и краткое описания принципов организации управления.
2.1.3. Функции основных структурных подразделений с группировкой по бизнес-процессам.
Информация может быть представлена в виде текстового описания, таблицы в разрезе подразделений либо моделей бизнес-процессов в общепринятой нотации.
2.2. Описание информационных потоков и документооборота предприятия.
2.2.1. Список документов предприятия.
Входящие, внутренние и исходящие документы по подразделениям. Возможно также разделение на следующие категории: управленческие, первичные (бухгалтерские, небухгалтерские), отчетные.
2.2.2. Описание процесса формирования документа.
- Наименование документа;
- Категория (входящий, исходящий, внутренний);
- Источник и адресат документа;
- Периодичность получения / обработки документа;
- Цель получения (заполнения) документа;
- Информация, необходимая для заполнения документа;
- Количество документов в месяц;
- Способ передачи документа.

2.2.3. Описание информационных потоков и документооборота по бизнес-процессам:
Каждый раздел содержит описание законченного бизнес-процесса, участие в котором могут принимать одно или несколько подразделений. Описание сопровождается схемой, отображающей информационные потоки и документооборот внутри этого процесса. Процесс описывается в разрезе функ-

ций управления (таких, как: планирование; организация исполнения; учет; контроль; анализ).

3. Требования и пожелания компании-заказчика по автоматизации основных бизнес-процессов.

4. Существующее положение по автоматизации.

4.1. Обзор программных средств.

Основные информационные системы в разрезе поддерживаемых ими бизнес-процессов.

4.2. Обзор технических средств.

Техническая архитектура (базовые инфраструктурные сервисы, системное ПО, серверное и пользовательское оборудование и системы хранения данных, инфраструктура ЛВС, серверные площадки и ЦОД).

5. Обзор типовых функциональных и технологических возможностей системы.

Базовая концепция организации документооборота при работе с системой и обзор функциональных возможностей системы по бизнес-процессам предприятия.

6. Анализ соответствия функциональных и технологических возможностей системы требованиям и пожеланием компании-заказчика.

Анализ проводится в разрезе бизнес-процессов или отдельных функций (например, финансы, бухгалтерский учет, кадры, планирование).

7. Выводы о применимости системы для автоматизации бизнес-процессов предприятия.

8. Рекомендуемый план мероприятий по внедрению системы на предприятии заказчика.

ПРОТОКОЛ ПРИЕМА-ПЕРЕДАЧИ В ОПЫТНО-ПРОМЫШЛЕННУЮ ЭКСПЛУАТАЦИЮ МОДУЛЯ СИСТЕМЫ

Протокол приема-передачи в опытно-промышленную эксплуатацию модуля «_____» контура «_____» системы «_____», разработанного по Договору № _____ от __ _____ 20_ года

г. Москва «__» _____ 20 _ г.

Модуль «_____» контура «_____» системы «_____» разработан на основании договора № _____ от «__» _____ 20_ г. и технического задания (проектного решения).

Разработчиком модуля «_____» контура «_____» системы «_____» является «_____» (Наименование компании-исполнителя).

Заказчику предъявлены:

1. Программный модуль «_____» контура «_____» системы «_____»;
2. «Рабочий журнал опытно-промышленной эксплуатации модуля «_____» контура «_____» системы «_____».

Срок опытно-промышленной эксплуатации модуля «_____» контура «_____» системы «_____» устанавливается на период с «__» _____ 20__ г. по «__» _____ 20__ г.

Программный модуль «_____» контура «_____» системы «_____» соответствует требованиям, изложенным в утвержденном Техническом задании, и принят в опытно-промышленную эксплуатацию.

С «__» _____ 20_ года по «__» _____ 20_ года Исполнителем осуществляется авторский надзор за промышленной эксплуатацией модуля «_____» контура «_____» системы «_____».

Ошибки и замечания, выявленные в ходе опытно-промышленной эксплуатации, Заказчик обязуется заносить в прилагаемый к настоящему Акту «Рабочий журнал опытно-промышленной эксплуатации модуля «_____» контура «_____» системы «_____» с подробным описанием ошибок и замечаний, датой внесения и подписью сотрудника, внесшего замечания. Исполнитель обязуется на основании данного «Рабочего журнала…» по согласованию с Заказчиком произвести необходимые исправления в выполненной работе без дополнительной оплаты, при условии того, что исправления не выходят за пределы утвержденного Технического задания.

От Заказчика: **От Исполнителя:**

_____ (должность, Ф.И.О.) _____ (должность, Ф.И.О.)

(подпись) (подпись)

РАБОЧИЙ ЖУРНАЛ ОПЫТНО-ПРОМЫШЛЕННОЙ ЭКСПЛУАТАЦИИ

Рабочий журнал опытно-промышленной эксплуатации (авторского надзора за промышленной эксплуатацией) модуля «_____» системы «_____», разработанного в рамках выполнения работ по Договору № ___ от _____

Дата начала: «____» _____ 20 _ г.
Дата окончания: «____» _____ 20 _ г.

№	Дата	Описание ошибки, замечания, предложения	Подпись ответственного от Заказчика	Подпись ответственного от Исполнителя	Дата исправления ошибки, замечания	Подпись ответственного от Заказчика	Подпись ответственного от Исполнителя	Примечание

От Заказчика:

_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:

_____ (должность, Ф.И.О.)
(подпись)

ПРОТОКОЛ ПРИЕМА-ПЕРЕДАЧИ В ПРОМЫШЛЕННУЮ ЭКСПЛУАТАЦИЮ МОДУЛЯ СИСТЕМЫ

Протокол приема-передачи в промышленную эксплуатацию модуля «_____» контура «_____» системы «_____», разработанного по Договору № _____ от __ _____ 20_ года

г. Москва «__» _____ 20 __ г.

Модуль «_____» контура «_____» системы «_____» разработан на основании договора № _____ от «__» _____ 200_ г. и технического задания (проектного решения).

Разработчиком модуля «_____» контура «_____» системы «_____» является «_____» (Наименование компании-исполнителя).

Заказчику предъявлены:

1. Программный модуль «_____» контура «_____» системы «_____»;
2. «Рабочий журнал опытно-промышленной эксплуатации модуля «_____» контура «_____» системы «_____»;
3. «Рабочий журнал авторского надзора за промышленной эксплуатацией модуля «_____» контура «_____» системы «_____».

Опытно-промышленная эксплуатация модуля «_____» контура «_____» системы «_____» проводилась в период с «__» _____ 20__ г. по «__» _ __ 20_ г. Выявленные в процессе опытно-промышленной эксплуатации ошибки и замечания устранены, стороны друг к другу претензий не имеют.

Модуль «_____» контура «_____» системы «_____» соответствует требованиям, изложенным в утвержденном Техническом задании, и принят в промышленную эксплуатацию.

С «__» _____ 20__ года по «__» _____ 20__ года Исполнителем осуществляется авторский надзор за промышленной эксплуатацией модуля «_____» контура «_____» системы «_____».

Ошибки и замечания, выявленные в течение авторского надзора, Заказчик обязуется заносить в прилагаемый к настоящему Акту «Рабочий журнал авторского надзора за промышленной эксплуатацией модуля «_____» контура «_____» системы «_____» с подробным описанием ошибок и замечаний, датой внесения и подписью сотрудника, внесшего замечания. Исполнитель обязуется на основании данного «Рабочего журнала…» по согласованию с Заказчиком произвести необходимые исправления в выполненной работе без дополнительной оплаты, при условии того, что исправления не выходят за пределы утвержденного Технического задания.

От Заказчика: **От Исполнителя:**

_____ (должность, Ф.И.О.) _____ (должность, Ф.И.О.)

(подпись) (подпись)

РАБОЧИЙ ЖУРНАЛ ОПЫТНО-ПРОМЫШЛЕННОЙ ЭКСПЛУАТАЦИИ

Рабочий журнал опытно-промышленной эксплуатации (авторского надзора за промышленной эксплуатацией) модуля «_____» системы «_____», разработанного в рамках выполнения работ по Договору № ___ от _____

Дата начала: «____» _____ 20 __ г.
Дата окончания: «____» _____ 20 __ г.

№	Дата	Описание ошибки, замечания, предложения	Подпись ответственного от Заказчика	Подпись ответственного от Исполнителя	Дата исправления ошибки, замечания	Подпись ответственного от Заказчика	Подпись ответственного от Исполнителя	Примечание

От Заказчика:

_____ (должность, Ф.И.О.)
(подпись)

От Исполнителя:

_____ (должность, Ф.И.О.)
(подпись)

ВЕДОМОСТЬ ПЕРЕДАЧИ ДОКУМЕНТОВ

Исходящий № от г.
Входящий № от г.

В секретариат Заказчика

Ведомость передачи документов

Сегодня, «_____» _____ 20__ года, компания _____ (наименование компании-подрядчика) в лице _____ (должность, фамилия, имя, отчество представителя компании-подрядчика) передала _____ (наименование предприятия Заказчика) на рассмотрение и утверждение следующие документы:

1. _____
2. _____
3. _____
4. _____
5. _____

Ведомость составлена в 2-х экземплярах – по одному для каждой стороны.

Документы переданы _____
(подпись, Ф. И. О. лица, передавшего документы)

Документы приняты _____
(подпись, Ф. И. О. лица, принявшего документы)

НАУЧНОЕ ИЗДАНИЕ

Евгений Петрович Зараменских

УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ ИНФОРМАЦИОННЫХ СИСТЕМ

Монография

Подписано в печать 02.08.2014. Формат 84×60X1/16. Бумага офсетная.
Тираж 500 экз. Уч.-изд. л. Печ л. Заказ

Отпечатано в типографии
ООО Издательство «СИБПРИНТ»
630099, г. Новосибирск, ул. Максима Горького, 39

Зараменских Евгений Петрович — *профессор кафедры корпоративных информационных систем факультета бизнес-информатики Национального исследовательского университета «Высшая школа экономики», Doctor of Philosophy (Ph.D.), кандидат технических наук, руководитель магистерской программы «Бизнес-информатика» НИУ ВШЭ, председатель профессиональной коллегии УМС по бизнес-информатике НИУ ВШЭ, генеральный директор «ИнтелГрупп».*