

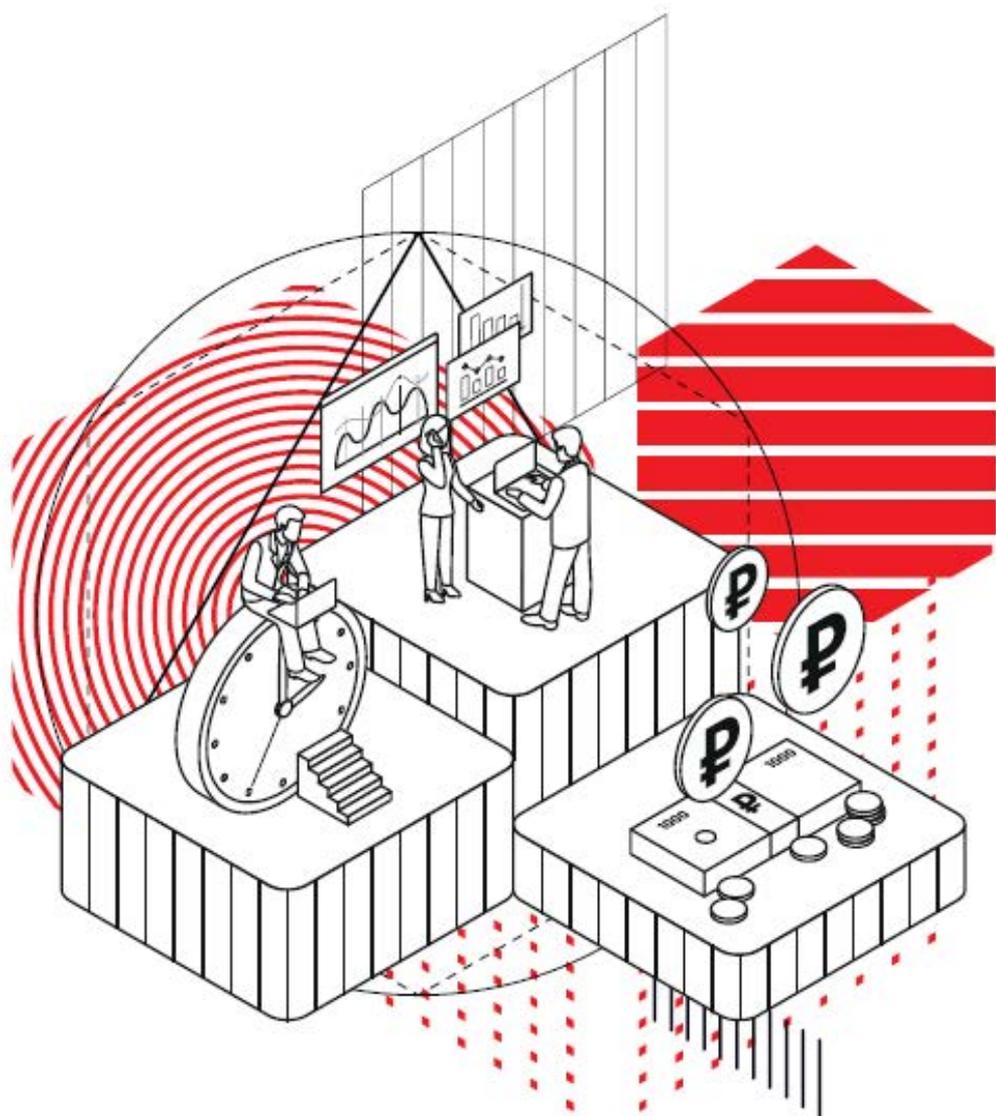
ERP

для начинающих

А.Э. Бобровников

ВВЕДЕНИЕ В УПРАВЛЕНИЕ ПРОЕКТАМИ ВНЕДРЕНИЯ ERP-СИСТЕМ

Москва
«ИС-Публишинг»



А. Э. Бобровников

Введение в управление проектами внедрения ERP-систем

Электронная книга в формате pdf; ISBN 978-5-9677-3018-4.

Электронный аналог издания "Введение в управление проектами внедрения ERP-систем" (ISBN 978-978-5-9677-2941-6, М.: ООО "1С-Паблишинг", 2021; артикул печатной книги по прайс-листу фирмы "1С": 4601546142726; по вопросам приобретения печатных изданий издательства "1С-Паблишинг" обращайтесь к партнеру "1С", обслуживающему вашу организацию, или к другим партнерам фирмы "1С").

Современные ERP-системы позволяют автоматизировать бизнес-процессы компании за счет разнообразия функциональности программных модулей, решать определенные задачи в зависимости от потребностей организации. Успешный запуск такой системы в промышленную эксплуатацию сопряжен с проектом внедрения, идущим несколько месяцев или лет (в случае комплексных проектов из разных этапов). Как пройти этот путь от начала (выбора самой ERP-системы) до конца (перехода к работе в системе и перевода ее в режим сопровождения)? Книга дает начальное представление об этом. Вопросы внедрения системы автоматизации иллюстрируются на примере прикладного решения «1С:ERP Управление предприятием».

Книга адресована начинающим специалистам, участвующим в проектах внедрения ERP-системы, как на стороне исполнителя, так и заказчика проекта. Специалистам исполнителя книга будет полезна для выработки единого представления внутри команды о процессах внедрения, выбора своей методологии и сплочения вокруг нее всех участников. Специалистам заказчика книга будет полезна для выработки понимания, как будет происходить процесс выбора и внедрения в компании ERP-системы. Студенты и аспиранты могут использовать данное издание для расширения и закрепления полученных знаний в области автоматизации компаний.

Оглавление

Введение	7
Принятые термины и сокращения	9
Глава 1. Что такое ERP-система и зачем она компании	13
Понятие ERP-системы.....	13
Отличие ERP-системы от учетной бухгалтерской системы.....	18
Зачем ERP-система компании	22
Затраты на владение системой (стоимость эксплуатации).....	24
Облачные сервисы как путь снижения затрат	29
Глава 2. Как выбрать ERP-систему	37
Функциональные требования от ключевых сотрудников	37
Тендер и участие в нем	41
Со стороны заказчика	41
Со стороны исполнителя	45
Fit-gap анализ разных систем	50
ИТ-ландшафт текущий и перспективный	55
Нагрузочные тесты и выбор «железа».....	60
Окупаемость и обоснование затрат на автоматизацию	63

Глава 3. Как внедрять ERP-систему	67
Технологии управления проектами.....	67
Agile-методологии	69
Серия стандартов ISO	72
ГОСТ 34	75
РМВОК.....	77
Технологии управления проектами фирмы «1С».....	85
Введение в терминологию.....	87
Проектный треугольник.....	90
Реинжиниринг бизнес-процессов	94
Какие подсистемы в какую очередь внедрять.....	96
Глава 4. Кто участвует в проекте внедрения.....	105
Организационная структура проекта.....	105
Управляющий комитет.....	109
Проектная команда исполнителя	113
Рабочая группа со стороны заказчика	117
Мотивация участников	
проекта внедрения.....	122
Зачем это все руководителю проекта	122
Модели мотивации	124
Коммуникация и конфликты	131
Удаленное взаимодействие	133
Командообразование	136
Управление конфликтами.....	137
Глава 5. Этапы и документация проекта.....	143
Жизненный цикл проекта внедрения ERP-системы	143
Документация проекта	150
Фазы проекта	154
Инициация проекта.....	154
Анализ и концептуальное проектирование	159
Дизайн архитектуры системы	161
Разработка	163
Опытная эксплуатация.....	165
Ввод в промышленную эксплуатацию.....	167
Закрытие проекта.....	169

Глава 6. Что анализировать и как настроить прототип системы	171
С чего начать анализ.....	171
Готовим отчет.....	176
Приоритизация требований	182
Описание бизнес-процессов.....	184
Функциональные блок-схемы.....	188
IDEFO	192
EPC	195
BPMN	199
Особенности описания бизнес-процессов.....	200
Прототип и его демонстрация	205
Глава 7. Как оценить срок и бюджет проекта	213
Сроки и бюджет проекта	213
Методы оценки трудозатрат.....	219
Оценка по аналогам	219
Анализ предложений исполнителей	220
Оценка «снизу вверх».....	220
Оценка «сверху вниз».....	220
Экспертная оценка	221
Параметрическая оценка	222
Оценка по трем точкам (метод PERT).....	222
Мифический человеко-час	223
Расчет себестоимости человеко-часа	228
Как получить итоговый срок проекта	235
Глава 8. Риски проекта и управление ими	243
Что такое риски проекта	243
Какие риски бывают в проекте внедрения ERP-системы	248
Профилактика и как бороться с проявлениями рисков.....	253
Когда риски можно игнорировать.....	266
Процедура управления рисками в проекте внедрения ERP-системы	267

Глава 9. Как пережить фазы разработки и опытной эксплуатации.....	271
Сколько и каких баз нужно в проекте на фазе разработки.....	271
Стандарты разработки.....	278
Тестирование.....	280
Готовимся к обучению пользователей.....	287
Опытная эксплуатация.....	294
Ведем список запросов на изменение.....	297
Глава 10. Запуск системы и что дальше	301
Ввод в промышленную эксплуатацию и закрытие проекта.....	301
Система продолжает работать – сопровождение корпоративной системы.....	305
Заключение.....	317
Список использованной литературы	319

Введение

Данная книга – это компромисс между существенным объемом материала по теме проектного управления и желанием сделать обзор в простом изложении важных для успеха любого внедрения системы управления предприятием тем. Существует довольно много методической литературы и курсов по тематике проектного управления вообще (абстрактно, без отраслевой специфики) и отдельно – применительно к информационным технологиям и конкретно к внедрению корпоративных информационных систем (КИС) класса ERP.

В основе всего зачастую лежит свод знаний от PMI – PMBOK. Особенностью данного издания является упор на практическую сторону вопроса управления проектом внедрения ERP-системы. По опыту общения на семинарах, проводимых фирмой «1С», автором неоднократно замечалась потребность в таком кратком пособии для начинающих ERP-специалистов, с которого можно начать путь в мир проектного управления.

Понятно, что в хорошей компании, занимающейся внедрениями, есть проектный менеджер (project manager), который все прекрасно знает и применяет на практике, натаскивает команду со стороны исполнителя проекта и рабочую группу со стороны заказчика, управляет и минимизирует риски. С чего-то нужно начать, чтобы стать хорошим специалистом и участником такой «правильной», применяющей проектные методики,

команды? Данная книга должна в этом помочь! Она будет интересна студентам и начинающим специалистам, которым в будущем предстоит столкнуться с реальной практикой работы на проекте (со стороны пользователей или специалистов по внедрению КИС).

По применению ERP-систем на предприятиях существует дополнительно рекомендуемая к изучению отдельная серия книг «Академия ERP», информация из которых будет полезна для понимания процессов, протекающих в компаниях, и их взаимосвязи. В изданиях серии теория иллюстрируется описаниями практических возможностей программных продуктов «1С».

Отзывы и предложения по улучшению этой книги и серии книг «Академия ERP» можно присылать на электронную почту publishing@1c.ru.

Принятые термины и сокращения

- **1С:ERP** – система управления предприятием «1С:ERP Управление предприятием» на базе платформы «1С:Предприятие 8».
- **CRM** – управление взаимоотношениями с клиентами (англ. Customer Relationship Management).
- **ERP** – планирование ресурсов предприятия (англ. Enterprise Resource Planning).
- **FIT-GAP анализ** – анализ «подходит – не подходит» (от англ. слов «подходит» и «разрыв»).
- **IoT** – Интернет вещей (англ. Internet of Things).
- **ISO** – международная организация по стандартизации (англ. International Organization for Standardization).
- **ITIL** – библиотека инфраструктуры информационных технологий (англ. IT Infrastructure Library).
- **KPI** – ключевые показатели эффективности (англ. Key Performance Indicators).
- **MRP** – планирование потребности в материалах (англ. Material Requirements Planning).

- **PERT** – техника оценки и анализа программ и проектов (англ. Program Evaluation Review Technique).
- **PM** – он же «ПиЭм» или «РП», руководитель проекта (англ. Project Manager).
- **PMБOK** – свод знаний по управлению проектами (англ. Project Management Body of Knowledge).
- **PMI** – институт по управлению проектами (англ. Project Management Institute).
- **PMO** – офис управления проектами (англ. Project Management Office).
- **RDP** – протокол удаленного рабочего стола (англ. Remote Desktop Protocol).
- **RFI** – запрос информации (англ. Request For Information).
- **RFP** – запрос на предложение (англ. Request For Proposal).
- **RFQ** – запрос о цене (разрабатываемой системы) (англ. Request For Quotation).
- **RLS** – разграничение доступа на уровне записей (англ. Record Level Security).
- **ROI** – окупаемость инвестиций (англ. Return On Investment).
- **SaaS** – программное обеспечение как услуга (англ. Software as a Service).
- **SLA** – соглашение об уровне предоставления услуги (англ. Service Level Agreement).
- **T&M** – оплата по факту (англ. Time & Material).
- **TCO** – совокупная стоимость владения (англ. Total Cost of Ownership).

- **VPN** – виртуальная частная сеть (англ. Virtual Private Network).
- **ИСР** или **WBS** – иерархическая структура работ (англ. Work Breakdown Structure).
- **ИТ/ИТ** – информационные технологии (англ. Information Technology).
- **КИС** – корпоративная информационная система.
- **МСФО** – международные стандарты финансовой отчетности.
- **ПО** – программное обеспечение.
- **РСБУ** – российские стандарты бухгалтерского учета.
- **СМИ** – средства массовой информации.
- **ФОТ** – фонд оплаты труда.

Глава 1. Что такое ERP-система и зачем она компании

Понятие ERP-системы

Сам термин ERP – это аббревиатура из английских слов Enterprise Resource Planning, дословно «планирование ресурсов предприятия». Такие системы (и сам термин) были наследием систем MRP и MRP II класса (англ. Material Requirements Planning – планирование потребности в материалах). Но за счет включения в себя множества других модулей или подсистем (тоже имеющих устоявшиеся двух-трехбуквенные обозначения) ERP стало собирательным наименованием систем планирования, учета, управления и принятия решений.

Методология MRP берет начало в 1950-х годах в США, но, формализованная на бумаге, до середины 1970-х отклика в реальном бизнесе не находит, пока компьютерная техника достаточно не эволюционирует. В СССР в середине 1960-х тоже был интересный проект автоматизации народного хозяйства в единой (сетевой по всей стране) компьютерной системе, но техническое воплощение не соответствовало этим опережающим время идеям. Какие-то наработки в 1980-х на Западе пошли в дело и вернулись к нам обратно в виде готовых систем автоматизации в конце 1990-х годов.

ERP-система – это корпоративная информационная система, предназначенная для автоматизации основных бизнес-процессов компании, учета и управления (планирование, контроль и анализ) ресурсами.

Определение термина ERP, которое дает APICS (American Production & Inventory Control Society, Американское общество управления производством и распределением материалов): **Планирование ресурсов предприятия (ERP)** – это рамки, основа (framework) для организации определения и стандартизации бизнес-процессов, необходимых для эффективного планирования и контроля организации таким образом, чтобы организация могла использовать внутренние знания для поиска внешнего преимущества.

Интересно отметить, что до 2005 года определение от APICS было совсем другим: **ERP-система** – это финансово-ориентированная информационная система, служащая для определения и планирования ресурсов всего предприятия, требуемых для получения, изготовления, отгрузки и учета заказов потребителей.

Ниже представлен неисчерпывающий перечень систем (и методологий), которые могут входить в состав комплексной ERP-системы как платформы или быть отдельными системами, использующими/предоставляющими данные из/в ERP. Забегая вперед, можно отметить,

что на описываемые ниже методики управления проектами внедрения это не сильно влияет – системы внедряются, по сути, по общим методологиям, хотя функциональность и покрываемые бизнес-процессы в них, конечно, разные, есть своя специфика, но все это класс ИТ-проектов внедрения систем автоматизации.

Общепринятые обозначения некоторых систем и понятий:

- **MES** (Manufacturing Execution System) – производственные управляющие системы;
- **WMS** (Warehouse Management System) – системы управления складами;
- **CRM** (Customer Relationship Management) – системы управления взаимоотношениями с клиентами;
- **SCM** (Supply Chain Management) – системы управления цепочками поставок;
- **TMS** (Transportation Management System) – системы управления транспортировками;
- **PDM** (Product Data Management) – системы управления данными о продуктах (изделиях);
- **KPI** (Key Performance Indicators) – ключевые показатели эффективности. Входят во многие системы показателей, например, **BSC** (Balanced Scorecard) – сбалансированная система показателей. Объединяются в отдельный блок (подсистему) визуализации показателей (монитор/витрина целевых показателей);
- **EAM** (Enterprise Asset Management) – управление основными фондами предприятия (процессами эксплуатации и ремонта). **ТОиР** – техническое обслуживание и ремонт;
- **HRM** (Human Resources Management) – управление персоналом;

- **CPM** (Corporate Performance Management) – система управления эффективностью предприятия. Синонимы для такого класса систем: **BPM** (Business Performance Management), **ЕРМ** (Enterprise Performance Management), **SEM** (Strategic Enterprise Management);
- **MDM** (Master Data Management) – системы управления мастер-данными (сведение и управление общей **НСИ** – нормативно-справочной информацией);
- **BPMS** (Business Process Management System) – системы управления бизнес-процессами.

Схема подсистем 1С:ERP представлена ниже на рисунке.



Рис. 1.1. Схема подсистем 1С:ERP

Подсистемы на схеме соответствуют разделам или подразделам внутри интерфейса системы и доступны при включении соответствующих опций. В процессе подготовки к внедрению определяется перечень внедряемых блоков – можно использовать только некоторые нужные блоки, а остальные подключать по мере готовности компании к их использованию.

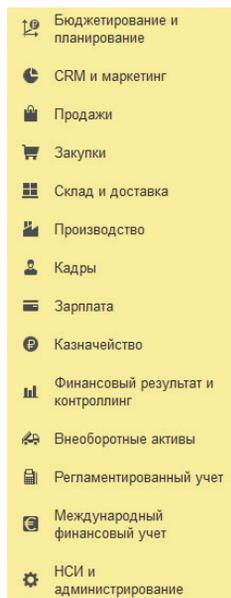


Рис. 1.2. Панель разделов 1С:ERP

В ERP-системах от других производителей состав подсистем примерно такой же, но они могут быть как единой системой, так и отдельными системами, интегрированными друг с другом.

Состав и особенности функциональности той или иной подсистемы ERP-системы в данной книге рассматриваться не будет. В рамках серии книг «Академия ERP» можно ознакомиться с основными функциональными блоками ERP-систем, теорией и практикой работы с ними в компаниях. Здесь же рассматриваются вопросы именно внедрения, потому система/модуль не так существенна, хотя, конечно, примеры будут приводиться на базе 1С:ERP конфигурации для наглядности, так как иначе описание

внедрения абстрактного «нечто» превратит весь замысел практической книги в очередную методичку по управлению проектами (неважно какими).

Отличие ERP-системы от учетной бухгалтерской системы

Самое важное и определяющее все остальные различия в системах – это целевая аудитория: кто и зачем работает в системе.

Учетная бухгалтерская система используется бухгалтерами предприятия для составления финансовой отчетности, часто в первую очередь для проверяющих фискальных органов и расчета налогов, во вторую очередь уже для руководства и принятия управленческих решений.

Современные бухгалтерские системы имеют функциональность для организации контура оперативного учета в компании, но исторически в них часто работают только бухгалтеры. А задачи планирования и оперативного учета остаются «за бортом», менеджеры работают в «эксельках» и «вордах», наклеивают стикеры на монитор с напоминками или используют для этого календарь и заметки в смартфоне, все бизнес-процессы идут через корпоративную (или бесплатную от известных брендов) почту либо в мессенджерах. Реальные отчеты для управления, планирования, анализа целевых показателей и их достижения по периодам делаются вручную по таким менеджерским базам + выгрузкам отчетов из бухгалтерской системы.

В самой бухгалтерской программе – красота и порядок, наведенный главным бухгалтером компании. Все бухгалтеры работают с понятными учетными функциями, которые проходят в профильных институтах и на курсах бухгалтеров, для всего есть положения по бухгалтерскому учету (ПБУ) и налоговый кодекс (НК) с набором писем Минфина по уточнению трактовок (для уточнения нововведений законодательства).

В свою очередь, в ERP-системе на первое место выходят оперативный учет и автоматизация бизнес-процессов внутри компании, автоматизация документооборота, согласования, подготовки пакета документов, маркетинг и отношения с контрагентами, контроль и планирование денежных потоков, управление производством и прочее.

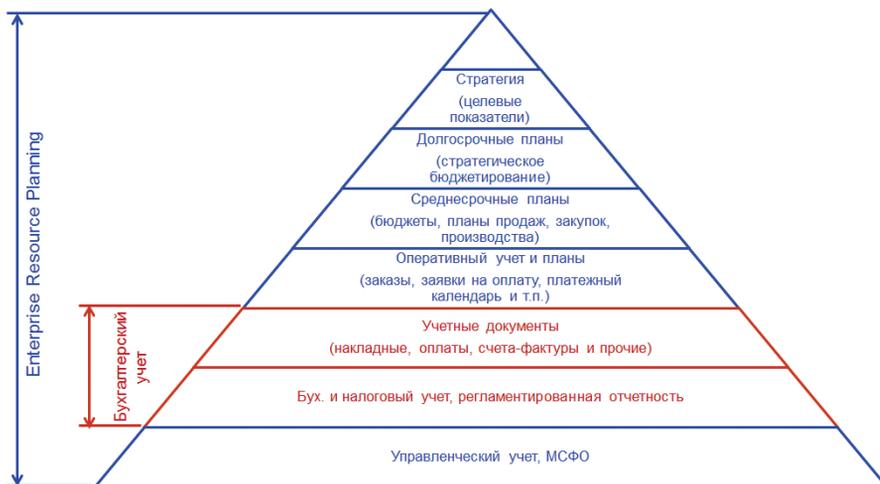


Рис. 1.3. Уровни данных в ERP-системе

И уже как следствие оперативного учета по этим данным формируются бухгалтерские записи (либо данные из оперативных документов передаются в отдельную бухгалтерскую систему, и проводки формируются там). Бухгалтеры в такой системе – не на главных ролях как монопольные пользователи, где все для них, а специалисты по подготовке регламентированной отчетности на базе первичных данных, формируемых оперативным учетом, который ведут специалисты по основной деятельности компании (продажи, закупки, производство).

Ситуация потери монополии на работу в системе (и ее непринятие бухгалтерией), кстати, одно из конфликтных мест при внедрении. Об этом поговорим подробнее в других главах.

В такой ERP-системе можно оперировать уже понятиями реального мира, а не синтетическим планом счетов с его аналитикой. Например, закупая товар, достаточно отметить в нормативно-справочной информации (далее НСИ) для номенклатуры, к какой группе он относится или на какой склад приходит. Тогда для системы будет понятно, что это товары для продажи или материалы, и счет учета, когда придет время формировать бухгалтерские проводки, система подставит сама.

В такой ERP-системе можно оперировать уже понятиями реального мира, а не синтетическим планом счетов с его аналитикой. Например, закупая товар, достаточно отметить в нормативно-справочной информации (далее НСИ) для номенклатуры, к какой группе он относится или на какой склад приходит. Тогда для системы будет понятно, что это товары для продажи или материалы, и счет учета, когда придет время формировать бухгалтерские проводки, система подставит сама.

Нужно договариваться, сводить все к единой НСИ. Иначе будет «лебедь, рак и щука» – кто куда, по своим интересам.

А еще могут свое влияние и требования привнести специалисты, составляющие отчетность по международным стандартам (МСФО). Там есть свои стандарты, накладывающие требования по детализации аналитики учета для составления комплекта отчетности и примечаний к нему. Ведение управленческого учета по МСФО очень полезно для менеджмента компании, так как дает им отчетность по принятым мировым методикам именно для целей оценки бизнеса по единым правилам (сопоставимость отчетности с другими компаниями отрасли и конкурентами). Современное образование и особенно MBA (Master of Business Administration) привили новому поколению руководителей компаний навыки работы с показателями отчетности на базе единых мировых стандартов, потому для управленческого учета в качестве корпоративных стандартов часто выбирают МСФО (а не изобретают свой «велосипед» для правил составления корпоративной отчетности).

Некоторые операции в национальном бухгалтерском (и налоговом) учете и в МСФО трактуются сильно по-разному: относятся на разные даты (и даже периоды), разными суммами, по разным статьям. В итоге финансовый результат по периоду может отличаться в разных учетах. Например, по РСБУ компания в прибыли и платит налоги, тогда как по МСФО у компании многомиллионные убытки. И пора бить тревогу, принимая своевременные управленческие решения.

Нужно отметить, что отчетность по МСФО может формироваться по данным бухгалтерской учетной системы, именно ERP для этого не требуется. Но такая отчетность строится обычно перекладкой РСБУ-отчетности (трансформацией) в МСФО-отчетность.

Процедура получения МСФО-отчетности трудоемкая, делается после сведения РСБУ-отчетности и для анализа на стол менеджмента, заинтересованных инвесторов и учредителей ложится сильно позже, чем хотелось бы. Например, в апреле – мае публикуется отчетность за прошедший год, до процесса аудита отчетность для внутренних нужд готова где-то в марте. По такой отчетности повлиять на финансовый результат прошедшего года уже нельзя, это «посмертная отчетность», показывающая свершившийся факт. Принимать решения по ней можно только для исправления будущих периодов. Сокращение сроков подготовки отчетности с годовой до квартальной или месячной требует уже автоматизации и формирования МСФО-отчетов не только по данным РСБУ-отчетности, но и по оперативному учету. Возможно, вообще независимо от РСБУ, со своими сроками закрытия периодов. Тогда отчетность становится оперативной, позволяющей управлять компанией в реальном времени.

Альтернатива МСФО, когда международные стандарты не используются, – оперативный учет, и управленческие отчеты по нему ведутся по корпоративным стандартам с достаточной аналитикой и сроками

предоставления отчетности, комфортными для принятия управленческих решений в компании. Это тоже требует данных, которых просто нет в обычной учетной бухгалтерской системе. А вот в ERP такие данные есть!

В общем случае в ERP-системе больше разнообразных «небухгалтерских» данных, нефинансовых в том числе, но очень важных для управления компанией. Можно процитировать Генри Форда: «Когда я изучаю свой баланс, я не могу увидеть там двух вещей: цены моей марки и стоимости моих кадров. Однако именно они и делают меня богачом». В современных бухгалтерских программах есть и цены, и ставки зарплаты, но вот в классической бухгалтерской отчетности этого действительно нет.

Зачем ERP-система компании

Выше рассматривались отличия ERP-системы от бухгалтерской учетной системы. Из этого следуют основные причины потребности компании в ERP-системе. Система может быть необходима:

- для реального понимания бизнеса и управления им;
- прозрачности данных для аудиторов;
- предоставления требующейся оперативной отчетности по МСФО для родительской компании или инвесторов;
- повышения ценности компании, деловой репутации (гудвил в терминах МСФО), т.к. брендовая система автоматизации, по мнению аудиторов, один из необходимых критериев;
- как конкурентное преимущество: снижение затрат (ФОТ, себестоимость продукции, оборачиваемость склада, ассортиментное планирование) – инструмент для захвата рынка;
- для внедрения бюджетирования – тактического и стратегического;
- введения системы КРІ и контроля за ними;

- поддержки «цифровой экономики», как модель предприятия и продукции;
- переход к «Интернету вещей» – новый уровень контроля оборудования и/или конечной продукции.

Список этот не исчерпывающий, могут быть и другие причины. Если обратить внимание, то пункты сильно не однородны, условно их можно разделить на типы:

- для внутренних нужд (конкретные бизнес-преимущества для работы компании);
- для внешнего мира (аудиторы, информация для роста акций на бирже и для инвесторов за счет уверенности, что в компании с такой замечательной системой «все будет хорошо»).

Первая группа причин внедрения ERP представляет интерес для бизнеса, так как дает реальный (монетизируемый) эффект: замедляет рост расходов на ФОТ персонала (замедляет рост числа сотрудников, т. к. они справляются с большим потоком за счет автоматизации), минимизирует запасы на складе (и аренду площадей), управляет кассовыми разрывами (минимизация привлечения кредитов и займов за счет управления дебиторской и кредиторской задолженностью), уменьшает время на подготовку пакета документов по сделке, увеличивает количество продаж при том же штате отдела продаж и многое другое.

Еще важный момент сегодняшнего дня – это вхождение в нашу жизнь и работу современных ИТ-технологий: на рабочем месте сотрудника уже может не быть персонального компьютера, вся работа идет через смартфон или планшет, «на ногах» (когда сотрудник в разъездах, а не сидит в офисе). Система должна обеспечивать доступ к корпоративным ресурсам и комфортную работу с таких мобильных устройств. Современные ERP-системы, и IC:ERP в частности, такое позволяют.

Иногда даже один какой-то пункт и его экономический эффект окупает многолетнее внедрение ERP-системы. А если пользы несколько – совсем хорошо!

Затраты на владение системой (стоимость эксплуатации)

Чтобы закрыть вопрос о понятии «ERP» и перейти наконец-то к основной теме проектов внедрения, нужно рассмотреть еще немаловажные финансовые вопросы затрат на владение системой такого класса.

ТСО (англ. Total Cost of Ownership) – это совокупная стоимость владения, общая величина целевых затрат, которые вынужден нести владелец с момента начала владения до момента выхода из состояния владения.

Для начала нужно убрать заблуждение, которое реально встречается в тех компаниях, что хотят перейти на ERP-систему: «Вот мы сейчас все это внедрим (пусть и дорого) – и будет нам хорошо, одна сплошная прибыль и окупаемость затрат на внедрение».

Если понимание, что сама система имеет цену, а работы по ее пуско-наладке как услуги тоже что-то стоят, есть, то вот что с этой системой происходит дальше – понимания часто нет.

Но бывает и такое, совсем парадоксальное: автор лично был свидетелем ситуации, когда компания покупала систему для автоматизации бизнес-процессов документооборота и управления финансами и эта система лежала в красивой коробке почти год. При этом по договору о продаже системы были какие-то часы на демонстрацию ее работы, но на компьютере, где это показывали, переустановили операционную систему, а сотрудник, кому показывали, уволился. И на балансе бухучета числился этот «актив», по сути являясь балластом. Бизнес-процессы так и шли через бумажки, стикеры (кому отдать, что сделать) и электронную почту.

Важно понимать, что делать с приобретаемой КИС (при внедрении) и потом (при использовании). Собственно, об этом в книге речь и идет: цель – убрать такие пробелы в понимании, в том числе у тех, кому внедряют ERP-систему.

Как и из чего формируется стоимость проекта внедрения, будет подробно рассматриваться в главе «Как оценить срок и бюджет проекта». Но там рассмотрение будет со стороны управления проектами, то есть для внедренцев. А для потребителя что?

Прикинем состав затрат на приобретение и запуск системы в работу:

- стоимость «коробочной» версии ERP;
- стоимость лицензий на одно рабочее место;
- стоимость лицензий на сервер;
- стоимость сервера системы управления базами данных (СУБД). Это ПО от сторонних поставщиков. Там своя лицензионная политика: на пользователей, на ядра процессора;
- стоимость аппаратного сервера (нескольких серверов), куда все будет устанавливаться. Возможно, нужен не один, а несколько серверов с объединением в кластер, либо настройки идут через виртуальные машины на очень мощном едином сервере;
- серверные операционные системы для серверов;
- системы резервного копирования;
- услуги по пусконаладке серверов и встраиванию их в сетевую инфраструктуру компании;
- квадратные метры помещения для размещения аппаратуры, вентиляция, аккумуляторные блоки для бесперебойной работы серверов в режиме 24/7;
- затраты на организацию системы удаленного доступа к системе;

- консалтинговые услуги проекта внедрения ERP-системы;
- зарплата сотрудников, задействованных частично или полностью на проекте внедрения со стороны заказчика (включая сотрудников, отвлекаемых от основной работы на время обучения новой системе);
- обновление или приобретение новых компьютеров и/или мониторов для рабочих мест сотрудников – будущих пользователей. Это часто не учитывается, а между тем текущий парк персональных компьютеров может не позволить комфортно работать в новой системе (скорость работы, разрешение экрана). А если пользователей много, то и бюджет обновления будет значительным.

Сегодня ERP-системы поставляются в электронном формате, документация также доступна через Интернет, что позволяет поставщику поддерживать документацию и систему всегда в актуальном состоянии, регулярно публиковать обновления для системы.

Но некоторые покупатели, которые ожидают что-то материальное после приобретения ERP-системы, что можно подержать в руках, могут восклицать: «Где все? Мы же заплатили деньги, где система? А нам какую-то ссылку на сайт прислали!»



Рис. 1.4. Коробка с системой и документацией 1C:ERP

Для таких клиентов фирма «1С» дополнительно к электронной поставке может предоставить физическую картонную коробку – там диски и документация. Коробка с 1С:ERP весит очень прилично и, прямо скажем, «внушает». Это снимает дискомфорт от нематериальной сути приобретаемой ERP-системы. Рассмотрим отдельно важную статью расходов – на системы резервного копирования.

Очень хочется прокомментировать ситуацию с резервным копированием известной в ИТ-кругах поговоркой: «Есть те, кто **еще** не делает резервные копии, и есть те, кто **уже** делает!» Совет: не нужно дожидаться этого «уже»!

Это важный пункт в перечне дел для внедрения, и нужно сразу настраивать создание резервных копий базы данных, т. к. ценность ее перекроет все затраты. Частота бэкапов должна быть разумной, как и глубина хранения версий. Даже откат базы на один день назад – это потеря сотен человеко-часов работы сотрудников и потерянные полдня-день на восстановление. А неделя, а вся база за годы?

Расходы на покупку сервера: как правило, имеющиеся на предприятии серверные ресурсы уже распределены между существующими задачами, и дополнительная нагрузка способна негативно сказаться на новых и старых задачах. Оборудование просто не потянет. А серверы стоят прилично. На этапе проектирования следует предусмотреть модернизацию или приобретение серверного и сетевого оборудования под планируемые объемы данных и нагрузку.

Как видно из перечня затрат на приобретение и внедрение ERP, это далеко не одна-две строки. Аппаратное обеспечение для серьезного объема данных в такой системе тоже будет иметь стоимость. Все это должно работать и после запуска в промышленную эксплуатацию.

Вот мы и переходим к стоимости владения. Что в нее входит?

- Техническое обслуживание серверов (затраты на ИТ-специалиста в штате или приглашаемого на время).

- Подписка на сопровождение и обновления для КИС (для западных ERP-систем это значительная статья расходов, т.к. стоимость подписки может быть 20–30 % от стоимости всех лицензий на КИС в год).
- Затраты на службу внутренней поддержки пользователей или внешних специалистов, привлекаемых для решения возникающих в работе вопросов и небольших доработок (включая затраты на командировки, если это требуется, например, для обучения новых сотрудников в филиалах).
- Регулярные обновления системы на новые версии, содержащие новый функционал, исправление ошибок и, главное, поддержка изменений законодательства (для систем автоматизации бухгалтерского и налогового учета). Нужно учесть, что чем больше доработок и изменений было внесено в «коробочную» версию ERP при внедрении, тем дороже и сложнее происходит такое обновление.
- Обновление (upgrade) аппаратуры серверов для масштабирования при росте нагрузки или поломках/устаревании/износе.
- Дополнительные регулярные расходы на возникшую из-за появления ERP инфраструктуру в компании.

Если все эти затраты оцифровать и составить бюджет проекта внедрения и месячной (или годовой) стоимости эксплуатации, то станет понятно еще одно отличие ERP от учетных бухгалтерских систем – это еще и прилично дороже. Ведь бухгалтерская программа для небольшой компании и одного бухгалтера может работать на обычном ПК, без серверов и не требует дорогого консалтинга на внедрение. Курсы или самостоятельное изучение по книжкам вполне реалистичны для запуска бухгалтерского учета.

Можно ли как-то оптимизировать затраты на ERP-систему? Можно, если посмотреть в сторону облачных решений и SaaS (Software as a Service) предложений аренды аппаратного и программного обеспечения.

Облачные сервисы как путь снижения затрат

Рассмотрим аспект использования ERP-системы в облаках.

Облака, cloud-сервисы, облачные технологии – модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых аппаратных и/или программных ресурсов.

В такой облачной модели нет затрат на ERP-систему:

- Разовых:
 - приобретение аппаратных серверов;
 - услуги по пусконаладке аппаратуры серверов и серверной комнаты;
 - приобретение лицензий на все ПО (сервер приложений, само ERP-решение, лицензии на пользователей, серверные операционные системы);
 - резервное копирование;
 - обновление (upgrade) серверов по необходимости (поломки, устаревание) или для масштабирования (рост компании и числа пользователей).
- Регулярных:
 - оплата ИТ-специалистов по обслуживанию серверов;
 - содержание серверной комнаты;
 - годовая подписка поддержки на обновление ПО;

- процесс обновления ERP-системы на актуальную версию (тут не все просто, об этом ниже).

В свою очередь, появляются новые регулярные траты – месячная подписка на обслуживание по каждому пользователю или пакетно.

Экономия на стоимости разовых затрат будет существенной. Причем временная экономия тоже: не нужно ждать поставки и настройки оборудования. Оплата месячного обслуживания, даже если будет в итоге больше, чем затраты на поддержку работающей системы своими силами, все же распределена по времени, и бизнесу не нужно вынимать из оборота часть средств в момент запуска системы. Но, главное, не нужно организовывать внутри компании непрофильную для ее деятельности ИТ-службу по поддержке всего серверного аппаратного и программного обеспечения.

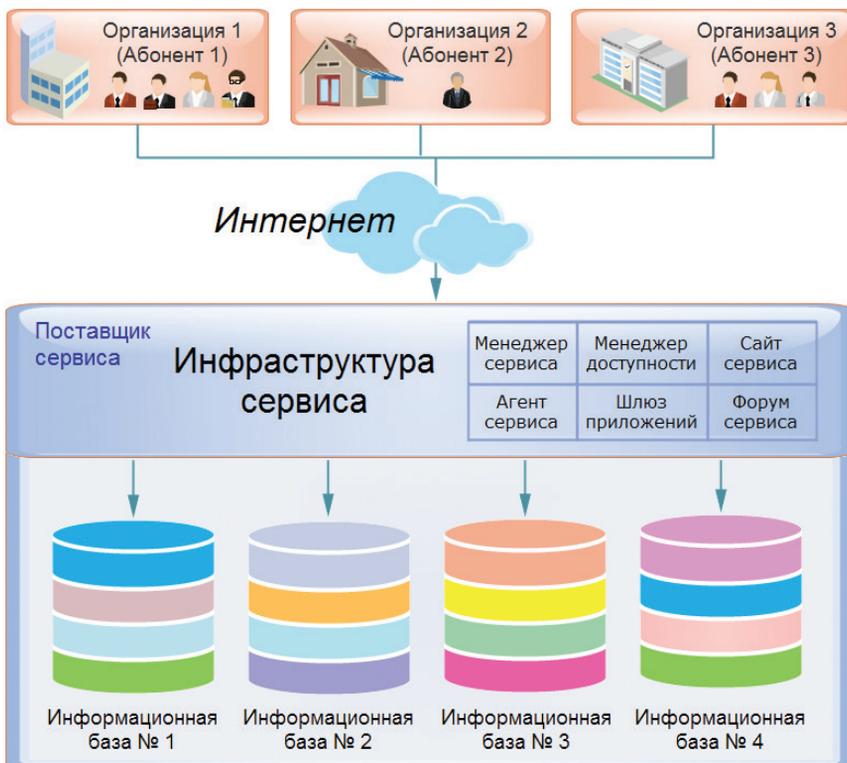


Рис. 1.5. Схема облачного сервиса

В случае проекта внедрения можно на первых этапах без приобретения дорогостоящего оборудования и самой ERP-системы «попробовать», как это будет на практике: готова ли компания к проекту внедрения вообще и подходит эта конкретная система бизнесу. Нужно отметить, что облачные сервисы возможны нескольких вариантов (и все они могут быть сопряжены с использованием ERP-системы):

- **IaaS** (инфраструктура как сервис) – клиент использует физические или виртуальные серверы, дисковые хранилища и прочее оборудование оператора облака для конфигурирования своей ИТ-инфраструктуры.
 - Клиент не может определять, как указанные блоки физически соединены между собой, но может строить из них логические конфигурации в рамках ограничений, определенных оператором.
 - Оператор несет ответственность за функционирование аппаратных блоков и их связь между собой.
 - Обслуживанием ОС и программного обеспечения (включая ERP) занимается клиент.
 - Резервное копирование СУБД и приложений настраивает и обеспечивает клиент.
- **PaaS** (платформа как сервис) – позволяет пользователю установить в облачной инфраструктуре необходимое приложение, использующее операционные системы, языки программирования, библиотеки, СУБД и другие сервисы, предоставленные оператором.
 - Оператор обеспечивает функционирование платформы и ее масштабирование при необходимости.
 - Клиент занимается только поддержкой и работой приложения.
 - Резервное копирование настраивает и обеспечивает оператор.

- **SaaS** (программное обеспечение как сервис) – клиент использует приложение, запущенное внутри cloud-среды оператора.
 - Пользователи подключаются к нему с помощью различных протоколов (HTTP, RDP или VPN) и разных устройств, состав которых определен оператором, возможностями приложения (например, вся работа через веб-браузер).
 - Поддержка приложения и аппаратуры полностью осуществляется оператором.
 - Клиент может влиять только на ограниченное число настроек (заведение пользователей, назначение прав, включение/выключение дополнительных опций).
 - Резервное копирование настраивает и обеспечивает оператор.

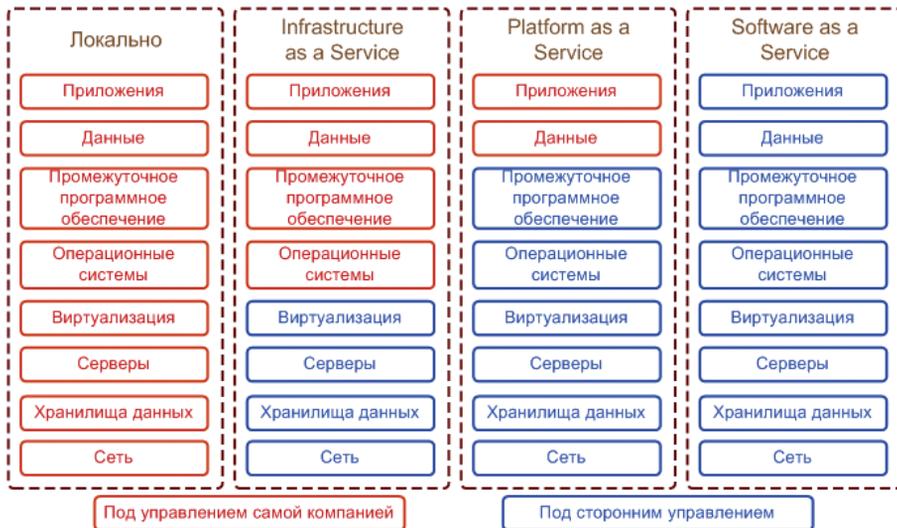


Рис. 1.6. Отличие технологий

Особый момент с обновлениями. Как было сказано выше, чем больше модификаций в ERP-системе в процессе внедрения, тем сложнее ее обновлять. Это накладывает ограничения на доступный вид облачного сервиса:

- Если компанию устраивает штатный функционал «из коробки», бизнес-процессы компании покрываются возможностями системы (или подстраиваются под них), то хорошо подходит модель **SaaS**.
- Если предполагается серьезная модификация системы в процессе внедрения, то потребуется модель **PaaS**.
- Если же ERP-автоматизация будет из нескольких интегрируемых систем от разных поставщиков, под которую потребуется специальная настройка окружения, то, возможно, хорошим вариантом будет **IaaS**.

При серьезных доработках ERP-системы ее обновление на актуальную версию будет отдельным мини-проектом и происходить автоматически, как обновления в облачном сервисе, не сможет. Нужно иметь своих специалистов или договор на осуществление этих работ со сторонней организацией.

Понятно, что чем больше компании придется настраивать и делать самостоятельно, тем выше будет стоимость эксплуатации системы, но тем меньше может стоить услуга по самому облачному сервису.

Тут нужно решить для себя:

- Хотим ли мы всем управлять и менять систему гибко, как нужно нам для наших устоявшихся бизнес-процессов.
- Хотим ли мы не думать о поддержке, а работать в готовом решении, принимая все его методики и ограничения как есть, подстраивая бизнес под них.

И в зависимости от этого выбрать тот или иной способ и определить (заранее) стоимость владения ERP-системой.

Подробнее об облачных технологиях компании «1С» можно почитать здесь:

http://v8.1c.ru/overview/Term_000000803.htm



Рис. 1.7. Облачный сервис компании «1С»

Система 1С:ERP присутствует в облачном сервисе «1С:Предприятие 8 через Интернет» по модели SaaS. Подробнее об этом – по ссылке: <https://1cfresh.com/solutions/erp>.

Нужно отметить момент резервного копирования информации и выбора поставщика услуги облачного сервиса. Все поставщики услуг обычно пишут на своих сайтах общие слова о гарантии работы 24/7, резервных копиях и о том, что «вам ни о чем не нужно думать, кроме самих данных и работы с ними». Но на практике это могут оказаться рекламные слова. Базу можно потерять безвозвратно. Нужно уточнить правила создания резервных копий, с какой глубиной они хранятся, сколько времени нужно на восстановление по запросу.

Автор знает такой пример недобросовестного предоставления облачной услуги, когда предположительно бэкап базы делался с недостаточной глубиной резервных копий, и при аппаратном сбое в бэкап стали писаться уже испорченные базы, а версии, где проблемы еще не было, уже не оказалось для восстановления.

В серьезных дата-центрах все многократно дублируется (рейд-массивы дисков с горячей заменой), используется полноценное резервное копирование, что гарантирует заявленный сервис и качество услуги. Так, например, в дата-центре «1С» для облачного сервиса стоит соответствующая аппаратура и обеспечивается бесперебойный уровень работы для конечного пользователя.

Глава 2. Как выбрать ERP-систему

Функциональные требования от ключевых сотрудников

Вопросам выбора конкретной ERP-системы предшествуют определенные процессы внутри компании: кто-то почему-то хочет ERP-систему. В компании есть группа лиц, заинтересованных в автоматизации своих процессов. Пусть даже это только итоговая финансовая отчетность и вычисление KPI по ней (цели финансового директора), а остальным сотрудникам «ничего от этой ERP не нужно, и так все хорошо». Либо причины иные – зачем ERP-система компании, рассматривалось в первой главе.

Кому-то система нужна, кому-то предстоит с ней работать. Это могут быть разные группы сотрудников, либо они пересекаются. Плохо, когда те, кому предстоит работать в системе, потребности что-то менять и автоматизировать не имеют. Возможны конфликты интересов внутри компании. В любом случае, если не рассматривать вопросы явного саботажа, а, наоборот, компанию рассматривать как единое целое, когда все сотрудники дружным коллективом работают на ее благо и в едином порыве, единогласно приветствуют процессы перехода на новую ERP-систему, тогда остаются практические вопросы: что и как должна уметь система, чем она поможет в работе за счет автоматизации. И как ее выбрать из представленных на рынке вариантов.

Проблемы конфликтов в коллективе будут рассмотрены ниже в главе про риски.

С системой предстоит работать разным подразделениям, от которых выделяются ключевые сотрудники (руководители подразделений или продвинутые исполнители бизнес-процессов). Эти специалисты формируют списки своих требований к системе, которые объединяются в общий сводный список требований к функциональности системы или, по-другому, **список функциональных требований**.

Ключевой сотрудник – это работник, от результатов труда которого зависит эффективная работа компании. Этот специалист четко понимает и выполняет бизнес-процессы внутри своего отдела (или функцию, за которую отвечает или является ее основным исполнителем).

Ключевые требования к ERP-системам:

- функциональность, соответствие уникальным особенностям бизнеса (отраслевая специфика);
- быстродействие, отказоустойчивость, масштабируемость системы;
- инструментарий для мониторинга системы, сбор статистики для анализа факторов, влияющих на быстродействие, аудит работы пользователей;
- возможность и инструментарий доработки функционала;
- разграничение прав доступа к данным и функциям;
- интеграция с существующими системами;

- наличие материалов для обучения пользователей и понятный интерфейс для их работы, предъявление низких требований к уровню квалификации пользователей;
- стоимость приобретения и владения ERP-системой;
- наличие проверенной методики внедрения (тиражирование готового решения).

Требования разделяются на функциональные и нефункциональные. Разница между ними простая: нефункциональные требования – это требования к характеристикам системы, а не к ее функциям. Функциональные требования отвечают на вопрос: «Что должна делать система?»

Критерии, которых нужно придерживаться при формулировании требований к КИС:

- полнота описания;
- правильность, точность и недвусмысленность формулировок;
- осуществимость (можно сделать в принципе);
- необходимость (только нужное, без лишних фантазий: «а давайте еще попросим...»);
- приоритет;
- проверяемость.

Формальное описание требований должно исключать использование «туманных» формулировок: «прозрачный», «гибкий», «быстрый», «приемлемый» и т.п. Проверяемость таких требований потребует указания четких критериев ожидаемых метрик. Иначе всегда можно не принять такую систему: «Получилось не прозрачно, не гибко и с неприемлемым временем работы». И нужно пойти и все переделать. Если уже понятно, что ожидается от системы, это нужно четко формулировать: «Отчет должен выводить такие-то аналитические разрезы и строиться не более 3 минут за месячный период по всем организациям». Впрочем, на начальном этапе сбора списка требований от заинтересованных сотрудников подойдут любые формулировки, они впоследствии

будут уточняться и войдут в технические задания уже конкретными и формальными, иначе их просто нельзя будет сделать (программисту и настройщику системы сложно оперировать абстрактными понятиями).

Можно разделить требования к системе на три уровня:

- Есть глобальные цели автоматизации, которые, как правило, озвучивают спонсоры проекта (собственники бизнеса). Это что-то типа: повысить производительность/оборачиваемость, сократить запасы/издержки и т. д.
- Есть задачи линейных руководителей (снабжение, производство и т. д.), как правило, это какие-то функциональные блоки, глобальные консолидированные отчеты, доп. аналитика.
- Требования рядовых пользователей: интерфейсы, кнопки, отчеты, печатные формы, конкретные рабочие места.

При выборе системы крайне важно понимать цели и задачи руководителей, а детальные требования будут сформулированы уже на уровне ТЗ. К сожалению, часто встречаются предпроектные обследования или ТЗ, где большое внимание уделено мелочам («в системе необходима возможность вести справочник номенклатуры») и нет ключевых целей и задач самого проекта и системы автоматизации.

Сводный список требований должны составлять понимающие в вопросах автоматизации и бизнес-процессах компаний специалисты. Такого ответственного нужно еще найти. Возможно, что среди сотрудников компании его нет вовсе и нужно нанимать или привлекать внешних специалистов.

На практике встречается: поручить просто секретарю собрать все требования к системе по электронной почте со всех отделов в единый файл, без дополнительного анализа и формализации специалистом. Такие требования будут носить разрозненный характер, кто-то опишет свои потребности в автоматизации подробно, кто-то только «больные места» сегодняшнего дня. В любом случае в таких письмах будет не формализованное описание для передачи третьей стороне (участникам тендера по выбору ERP-системы), а описание для внутренних нужд, исходя из понимания реалий бизнеса и бизнес-процессов отделов в частности.

То есть еще не факт, что такие описания даже в соседнем отделе поймут, если отделы достаточно независимы и, например, в холдинге это разные бизнес-единицы со своими юридическими лицами и удаленными офисами. А ожидается, что список требований поймут потенциальные исполнители будущего проекта внедрения. Собранную информацию (пусть и с помощью секретаря, которая всех знает и у всех все запросит) нужно переработать и формально описать в сводный список для передачи третьим лицам для организации тендера. При этом нужно сохранить информацию об изначальном авторе требования (не для передаче третьей стороне, а для внутренних нужд): это понадобится для последующих уточнений по требованию от исполнителей или согласования отмены либо переформулировки требования в процессе его анализа.

Полноценный и формально описанный список функциональных требований, если самостоятельно получить его до начала тендера сложно, можно составить уже в процессе, привлекая для этого внешних консультантов, участников этого тендера. При этом хоть в каком-то сводном виде для начала работ по уточнению и анализу его получить нужно.

Тендер и участие в нем

Со стороны заказчика

Ну что же, решение о том, что компании нужна ERP-система, принято, начинаем выбирать саму систему и исполнителей для проекта внедрения.

В первой главе рассматривалось заблуждение, что можно ограничиться только самой системой и она запустится «как-то так, сама». Исходим из того, что понимание возможных вариантов ERP-систем, бюджетов и стоимости владения у компании уже есть (какие-то сотрудники представляют, что это такое). Либо такого понимания нет, и его как раз нужно выработать в процессе выбора системы.

Второй случай (нет понимания) нужно приводить к первому. Для этого нанять в штат специалиста, который понимает в автоматизации компаний, – это будущий руководитель проекта внедрения со стороны заказчика, а впоследствии руководитель группы поддержки и развития системы в компании. Вся группа, впрочем, может быть из одного этого специалиста или разрастись до целого отдела. Можно выбрать специалиста с целевыми компетенциями из текущих

сотрудников и предложить сменить вид деятельности, карьеру и занять такую интересную вакантную должность. Но нужно учесть, что этот специалист должен иметь опыт или профильное образование для соответствия задачам внедрения проекта автоматизации. Итак, с помощью Интернета, знакомых коллег из других компаний отрасли получено первичное представление о мире ERP-систем: какие системы бывают, кто основные игроки на рынке (кто их производит, кто внедряет). Например, по 1С:ERP много информации предоставлено в открытом доступе на официальном сайте продукта <http://v8.1c.ru/erp/>. Составляется шорт-лист систем и их потенциальных поставщиков. Компании, туда попавшие, могут быть как крупными, известными на рынке, так и мелкими, но по рекомендации знакомых.

Забегая вперед, отметим, что размер компании на проект внедрения влияет слабо: внедряют конкретные специалисты. Команда небольшой компании может оказаться сильнее, чем команда крупной компании, где проекты поставлены на поток и для комплектации команд в бой могут пойти стажеры с горящими глазами, но без соответствующей квалификации.

Далее нужно составить документ (несколько документов) из серии RFx:

- **RFI** – запрос информации (англ. Request For Information);
- **RFP** – запрос на предложение (англ. Request For Proposal);
- **RFQ** – запрос о цене (разрабатываемой системы) (англ. Request For Quotation).

Можно свести все в единый документ или один документ и таблицу приложения к нему. Получаем на выходе документ или письмо, который публикуется на сайте или рассылается по заинтересованным компаниям – поставщикам услуг.

Цель документа – собрать письменную информацию о возможностях различных поставщиков. Как правило, запрос предполагает заполнение таблички опросника в определенном формате, благодаря чему ответы

от разных поставщиков по шаблону могут использоваться для сравнения информации. RFI редко является заключительным этапом, часто используется в комбинации с запросом предложения (RFP) и запросом цены (RFQ).

Компании, проявившие интерес, и составят круг участников тендера на проект автоматизации.

Если уже ясно, какая ERP-система нужна (а такое бывает, т.к. определяется, например, стоимостью самой системы или поддержкой отраслевой специфики), то выбор самой системы уже ясен, и проводится выбор исполнителя для проекта автоматизации.

Нужно отметить, что можно принять решение об автоматизации своими силами. В компании уже есть или планируется ИТ-служба с командой, которая справится с задачами внедрения. С одной стороны, это может быть дешевле оплаты почасовых ставок проекта внедрения, с другой – нужно оперативно найти такой сработанный коллектив и нести риски конфликтов в нем, непрофессионализма конкретных сотрудников, возможной ротации (ухода из компании). А это повлияет на сроки и итоговый бюджет проекта.

Как управлять рисками и бюджетом проекта, в этой книге описано для стороны исполнителя. Можно эти знания применить и для отдела внутри компании.

Если сама система еще не выбрана или есть различные отраслевые решения внутри одной ERP-системы, то среди них нужно выбрать подходящее решение. Для этого понадобится список функциональных требований оформить в виде RFP, по которому будет проводиться fit-gap анализ.

О **fit-gap** анализе речь пойдет в следующем разделе этой главы.

На данном этапе считаем, что такой анализ проведен, результаты получены. Что дальше?

Внутри компании-заказчика нужно сформировать рабочую группу из тех самых ключевых сотрудников, что генерировали требования к системе. Им будет интересно узнать возможности и ограничения той или иной системы. Но только по таблице с fit-gap анализом, сухим цифрам из ответов в опроснике о возможностях исполнителя понять что-то сложно, нужна очная фаза – презентации ERP-систем и компаний исполнителей.

Тут возможны варианты: можно пригласить потенциальных исполнителей к себе (позволяет собрать расширенный состав своих участников), а можно приехать в гости в саму компанию, занимающуюся внедрением (тогда состав участников может быть сужен, зато будет видно, как компания выглядит изнутри).

После первичного знакомства с системой и компанией-внедренцем может быть серия референс-визитов по компаниям-клиентам, использующим эту систему и прошедшим через проект внедрения. Такая живая практика использования системы на конкретных пользователях очень полезна: покажет пригодность системы для реальных бизнес-процессов, вскроет ее минусы и возможность их преодолеть, раскроет со стороны заказчика комфортность работы со специалистами внедренца и соблюдение сроков и бюджета проекта.

До момента выбора системы такую информацию «из первых рук» можно получать на профильных мероприятиях, где доклады о ходе внедрения и интересных практических кейсах делают сами пользователи ERP-системы. Например, у «1С» такое мероприятие проходит ежегодно: «Бизнес-форум 1С:ERP» <http://1c.ru/bf/>. После докладов можно подойти к докладчикам и узнать подробности, договориться о визите в гости с показом системы.

Со стороны исполнителя

В какой-то момент приходит запрос об автоматизации от потенциального клиента. Запрос может прийти:

- на корпоративный e-mail, указанный на сайте;
- через форму обратной связи сайта;
- через знакомых;
- на личных встречах на профильных форумах и других мероприятиях (да хоть в бане у общих знакомых) вместе с вручением визитки с контактами.

Если запрос представляет собой детальное описание с ожидаемым формуляром ответа – все проще, нужно просто его заполнить. Обычно он краткий. Некоторые компании, у которых используется оценка контрагентов по разным критериям, могут запросить большой пакет документации, включая финансовые показатели компании за несколько лет.

Но чаще запрос будет неформальным письмом: «Нам нужна автоматизация, что вы можете предложить и что ваша система может?»

В предыдущем разделе книги «Со стороны заказчика» исходим из правильного подхода: RFx-документы и все такое. В этом разделе (со стороны исполнителя) будем реалистами: запрос будет не «каноническим» с высокой вероятностью. Пойдем по пессимистичному (реальному) сценарию.

Тогда нужно что-то формулировать в ответ. Обычно в компании-внедренце есть отдел или специалист, отвечающий за продажи и привлечение новых клиентов, либо этим занимаются сами руководители проектов или сам директор (для маленьких компаний, где многие роли завязаны на одном человеке).

У них уже должен быть готовый шаблон коммерческого предложения, который включает описание компании, программного продукта и перечня возможных услуг. Дополнительно может быть презентация, или ссылка на сайт с презентацией, или ролики по программному продукту и о компании.

На сайте компании полезно иметь раздел с отзывами довольных клиентов по другим проектам, краткую информацию о команде и ее специалистах, их проектном опыте и образовании. Это все равно понадобится, т. к. заказчики могут запросить резюме команды и референсы (ссылки) на завершенные проекты.

Формулировать конкретное коммерческое предложение с суммами и сроками всего проекта пока рано. Речь идет скорее о часовых ставках и примерных «вилках» типовых проектов такого плана «от» и «до», плюс информация о ценообразовании на саму систему ERP. В описании системы может присутствовать информация о требовании к аппаратной части (серверам) для работы с пометкой, что возможны варианты в зависимости от объема данных и количества пользователей. Можно сразу предложить разные варианты поставки системы, включая облачные сервисы.

Если шаблона коммерческого предложения нет – нужно его сделать! Для этого можно воспользоваться готовыми шаблонами из различных проектных методологий. О них речь пойдет в следующих главах.

Ответ на RFI лучше формулировать с учетом специфики работы компании, изучив доступную информацию на сайте заказчика или по иным доступным источникам. Клиент должен увидеть понятные для себя термины и знакомые (свои родные) бизнес-процессы.

Если сразу прислали RFP со спецификацией функциональных требований, то нужно заполнить эту таблицу, проведя быстрый fit-gap анализ (привлечь для этого методиста по системе и, если нужно, ведущего разработчика).

Даже если у ERP-системы есть демоверсия со свободным доступом, то предоставлять ее заказчику и оставлять его один на один с незнакомой системой – не лучший вариант ее презентации. Нужно, чтобы систему показывал специалист. ERP – система не того уровня, чтобы без подготовки в ней свободно ориентироваться. Эффект от такого деморежима может быть обратный – отрицательный.

Далее нужно связаться (не только по электронной почте) с представителем заказчика, проводящим тендер, и договориться об очной презентации (на своей или их территории). Возможна дистанционная встреча (при географической удаленности заказчика и исполнителя), современные технологии это позволяют. Возможен промежуточный вариант – совмещение дистанционного и очного участия. Дело в том, что в презентации со стороны исполнителя могут участвовать несколько специалистов:

- менеджер, который ведет переговоры;
- руководитель проектов;
- (опционально) куратор со стороны исполнителя (обычно сам директор, участие зависит от состава участников с той стороны, может быть, не на первой встрече, а на продолжении);
- ведущий консультант по системе (он будет показывать ERP-систему и делать по ней доклад);
- (опционально) ведущий разработчик (для оценок на лету возможности реализации того или иного требования в процессе разговора);
- если идет показ нескольких блоков системы, то консультанты и разработчики по блоками могут быть свои (в ERP-системах практикуется узкая специализация специалистов). Это еще плюс несколько участников.

В минималистическом варианте все эти роли собраны внутри одного человека. Он и швец, и жнец, и на дуде игрец – руководитель проектов, продажник, способный показать систему и знающий, как ее доработать и внедрить. Вернемся к экономии: в командировку может отправиться один специалист с хорошими навыками коммуникации и умением сглаживать конфликты в стрессовых ситуациях (разговоры в процессе показа могут принимать совершенно разный эмоциональный окрас). Обычно это или продажник, или руководитель проектов. Остальные участники встречи подключаются удаленно, показывают систему и отвечают на вопросы.

При удаленном показе презентации консультант не видит лиц и реакции слушателей, ему сложно подстроиться под ситуацию в зале. Представитель исполнителя, который находится в этот момент в зале заказчика, должен помочь коллеге, вовремя сориентировать – например, дать дополнительные пояснения или передать докладчику возникающие вопросы.

Специалисты заказчика могут организовать запись презентации системы. Такая запись может потом использоваться для подключения к проекту сотрудников заказчика, которые не смогли принять участие в показе. А переданная исполнителю запись может использоваться им для составления протокола встречи и как материалы для обучения консультантов проведению презентаций.

Как проводить презентацию, что и кому стоит показывать, зависит от участвующих ключевых сотрудников: главный бухгалтер, финдиректор, ИТ-директор, гендиректор. Каждому нужны свои акценты в тезисах и показе системы. Эти особенности требуют отдельного рассмотрения, которое подразумевает наличие знаний в предметных областях и в данную книгу не входит.

Если совсем кратко, на что обратить внимание при проведении презентации:

- вежливость и внешний вид, впечатление профессионализма;
- интерес к проблематике заказчика, а не решению;

- показ возможностей системы в терминологии заказчика (их продуктовая линейка, отраслевая специфика);
- быть на одной стороне (в идеале – физически за столом тоже, иначе это сразу позиция противостояния);
- рассказать о процессе внедрения: не только показ системы и ее возможностей, а как именно все это будет происходить, что потребуется от сотрудников заказчика по этапам проекта;
- конспектировать вопросы и решения по ним;
- предоставить протокол встречи всем участникам в течение нескольких дней после презентации.

После первой встречи выдать финальное коммерческое предложение с конкретной суммой и сроками все еще затруднительно. Если интерес к продолжению сотрудничества есть, нужно переходить в этап предпроекта.

Об этапах проекта и его жизненном цикле речь пойдет в главе «Этапы и документация проекта».

В таком состоянии могут оказаться и другие участники тендера, то есть успешная презентация и продолжение работ – это еще не окончательный выбор исполнителя и начало проекта, а шанс продолжить конкурсную борьбу за клиента.

Может сложиться ситуация, когда потребуется фиксировать суммы бюджета и сроки проекта уже на этом этапе и нужно давать финальное коммерческое предложение, без выделения предпроекта с обследованием и уточнением работ, сроков и бюджета в отдельный договор. Такая ситуация для оценки не очень приятная, придется «угадывать» трудозатраты, сроки и закладывать риски в оценку. Тут можно предложить заказчику провести блиц-опросы (по заранее подготовленным анкетам) и интервью ключевых сотрудников, хотя бы в течение нескольких дней. Это дополнительные затраты на работы со стороны исполнителя,

но и минимизация рисков и более точные оценки будущего (возможного) проекта. Если же требований с должной детализацией у заказчика вообще нет, то и готовности к проекту нет, нужно настаивать на отдельных работах на предпроект, предлагать коммерческое предложение именно на него и дать «вилки» стоимости на основной проект с уточнением по результатам обследования. Конечно, всегда остается метод «угадывания» бюджета и сроков на все этапы проекта и по существующей информации, но это явные проектные риски для последующих этапов и значительные допуски в оценках.

Fit-gap анализ разных систем

Рассмотрим более детально вопросы анализа сближений (fit) и разрывов (gap) пригодности ERP-системы для компании согласно ее требованиям. Как провести такой анализ (исполнителю) по полученной спецификации функциональных требований и как потом эти результаты трактовать (заказчику)?

Если исполнителями являются не внешние специалисты, а команда сотрудников в штате компании, то подход тот же самый. Здесь и далее «исполнитель» – это специалисты, отвечающие за автоматизацию и хорошо знающие рассматриваемую ERP-систему, а «заказчик» – заинтересованная в проекте сторона, пусть даже внутри одной и той же компании.

При fit-gap анализе за основу берется список требований от заказчика, он передается в составе RFP, разбит по разделам (модулям) системы: продажи, закупки, бухгалтерский и налоговый учет, производство, казначейство и прочие. Этот список переводится в таблицу, если изначально был не в виде таблицы, и к нему добавляются столбцы для оценок анализа.

Типовой шаблон таблицы для проведения fit-gap анализа содержит столбцы:

- **Наименование требования** – краткое описание критерия для оценки;
- **Приоритет** – важность и очередность для автоматизации: 1 – высший приоритет, 10 – низший;

- **Обязательность** – критичность реализации на первом этапе для запуска в эксплуатацию (так называемое must have – должно быть сразу);
- **Поддержка** – поддерживается штатно «из коробки»;
- **Настройка** – поддерживается настройкой системы (не требует программирования);
- **Модификация** – поддерживается доработкой (требует программирования);
- **3-я сторона** – поддержка решениями третьих сторон;
- **Будет в следующих версиях** – поддержка анонсирована в следующих версиях решения;
- **Не поддерживается** – нельзя сделать, технические и функциональные ограничения;
- **Комментарий к требованию** – пояснения к требованию, ссылки на другие документы с детальным описанием;
- **Комментарий к оценке fit-gap** – пояснения выбранной отметки при анализе.

Варианты оформления таблицы для анализа разные, зависят от применяемой методологии (могут входить как готовые шаблоны в приложение к методологии). Шаблон может требоваться самим заказчиком, что логично: потом все сводить к единому виду.

Например, можно не разделять столбцы **Модификация** и **Настройка**, считая их доработкой, но разница между ними есть. Нужно привлекать разработчика или нет, потом такие доработки в коде нужно аккуратно проверять на совместимость при обновлениях системы на новые версии. Еще может быть разная трудоемкость работ по доработке модификацией системы или настройками опций, прав доступа, внешнего вида форм и т. п.

Разница между **Поддерживается** и **Настройка** тоже есть, хотя можно утверждать, что если что-то настраивается без доработок – то это уже

поддерживается. Но на практике это могут быть разные трудозатраты: есть сразу по умолчанию или нужно настроить, описать правила работы, помоделировать варианты.

Выделение поддержки за счет сторонних решений тоже интересно – эти места потребуют интеграции и дополнительных затрат на приобретение самих решений.

Так или иначе, список критериев оценки может быть короче, если разделение не критично. В частных случаях это может быть два столбца в таблице: *fit* (поддерживается) и *gap* (не поддерживается). Собственно, потому и анализ *fit-gap*. Но нюансы существенны. Их тогда придется описывать в комментариях. Например, «не поддерживается» – да, но можно доработать. Или наоборот – «поддерживается» – да, но не из «коробки», а после модификаций или установки сторонних решений.

Потому выше рассмотрена более общая структура таблицы *fit-gap* анализа, где такие тонкости вынесены в отдельные столбцы.

Еще вариант этой таблицы: один столбец «*fit-gap*», в который ставится числовая оценка сближения или разрыва. Например, по шкале от 10 до 0, где 10 – подходит идеально, 1 – не подходит, 0 – нельзя доработать. Тогда промежуточные оценки показывают степень и сложность доработок (см. рис. 2.1).

Как, собственно, заполнить этот файл на стороне исполнителя? Привлекаются все эксперты (коллеги по компании-исполнителю) по системе:

- рассылкой по электронной почте для оценок каждым своего блока и первоначального знакомства с требованиями;
- в виде мозгового штурма в переговорной комнате, где все собираются и обсуждают, что имеется в виду под каждым пунктом требований, как это соотносится с возможностями системы и можем ли мы это доработать и настроить.

Если что-то неясно, то эти моменты уточняются у заказчика: потребуется еще итерация дооценок либо уточнения сразу по телефону у ответственного с той стороны. Ставится оценка, исходя из понимания, и пишется комментарий к оценке, что под этим понималось, если возможна неоднозначность трактовки.

№ пункта	Наименование требования	Приоритет (1-10)	Обязательность	Поддержка	Настройка	Модификация	3-я сторона	Бюджет	Не поддерживается	Комментарий к требованию	Комментарий к оценке fit-гар
1	Модуль 1	1	X								
1.1	Категория модуля 1	5	X								
1.1.1	Подкатегория категории 1	1	X								
1.1.1.1	Требование 1	1	X	X							
1.1.1.2	Требование 2	5							X		
1.1.1.3	Требование 3	5					X				
1.1.1.4	Требование 4	10						X			
1.1.1.5	Требование 5	7									
1.1.1.6	Требование 6	2	X		X						

Рис. 2.1. Пример таблицы fit-гар анализа требований

На данном этапе по названию требования и несколькими фразами комментария детально понять потребность в доработке или то, как именно нужно все настроить, сложно. Скорее, это быстрый анализ: есть ли такая возможность в ERP-системе как заявленная в спецификации к программному продукту или это ограничение, которое известно (неизвестно), как преодолеть. Тут нужны реальный экспертный опыт проектной команды и хорошая интуиция, чтобы отметки были как можно точнее.

Для исполнителя такой беглый анализ покажет места, которые нужно будет прорабатывать детальнее для большего понимания и уточнения оценок бюджета внедрения уже в процессе полноценного проекта, в фазе анализа и дизайна.

Заказчик, собирая такие fit-gap таблицы по своим требованиям, может их соотнести между собой, это позволит определить применимость той или иной ERP-системы, если идет выбор между разными системами. Поэтому столбцы для fit-gap анализа желательно добавить сразу на стороне заказчика, чтобы все участники тендера работали с одним шаблоном RFP и потом можно было соотносить результаты.

Если таблица объемная и зрительно сравнить затруднительно, вводится дополнительный столбец с оценками по каждой отметке, где проставляются баллы: поддержка «из коробки» – максимум, «не поддерживается» – ноль. Далее по сумме баллов определяется победитель.

Может получиться, что по каким-то разделам одна система лучше другой, а по другим разделам все наоборот. Выбор непростой, когда явного лидера нет. Нужно смотреть системы вживую на следующем этапе тендера, в его очной части, и учитывать уже факторы нефункциональных требований (стоимость, производительность и т. п.).

Зато если система не подходит совсем, то это сразу будет видно по сплошным «разрывам» в анализе. На практике fit-gap анализ скорее редкость или его детализация довольно поверхностная, не сотни строк в таблице, а десятки. Связано это с тем, что детальные функциональные требования заказчику собрать самостоятельно на начальном этапе сложно – это задача для специалистов на проектное обследование. И это самостоятельный мини-проект, целью которого является получить документ, содержащий требования и сразу результат анализа применительно к ERP-системе.

ИТ-ландшафт текущий и перспективный

Важный аспект при выборе ERP-системы – это окружение (программных средств и аппаратных комплексов), в котором ей предстоит работать.

ИТ-платформа, ИТ-инфраструктура, ИТ-архитектура – понятия похожие, но можно вкладывать в них разный смысл. Большой упор на аппаратную часть: серверы, сетевое оборудование, каналы удаленного доступа – или на программную часть: операционные системы, настройки безопасности и корпоративной сети, учетные системы.

Нас интересует состав систем управления и автоматизации бизнеса. Обычно для этого используется термин **ИТ-ландшафт**, так как часто систем используется больше, чем одна, и они как-то между собой сочетаются. Получается такой вот ИТ-ландшафт, похожий на географический ландшафт местности, на которой работают все системы: со всеми низинами, возвышенностями, дорогами и постройками. Сами КИС и более простые системы управления работают в некоей аппаратной среде информационных технологий. Детальное рассмотрение аппаратной инфраструктуры мы оставим за границами данной книги, при этом нужно обозначить несомненную важность вопроса аппаратного обеспечения для бесперебойной работы программных комплексов. Но это совсем иная история, по которой достаточно своих книг. В следующем разделе кратко рассмотрим только вопросы производительности, зависимые от «железа».

В части ИТ-ландшафта нужно составить схему текущего (as is) состояния и взаимосвязи систем и иных программ, используемых в бизнес-процессах компании.

Эта схема потребуется для лучшего понимания задач предстоящей автоматизации и выработки и выбора вариантов перспективных состояний, возможно, с какими-то промежуточными этапами на моменты перехода из текущего состояния в целевое (to be).

Некоторые системы могут быть вообще не связаны между собой, и данные в них перебиваются вручную, двойным вводом или полуавтоматически, экспортом и импортом из отчетных форм или промежуточных файлов.

Типичный пример такой связи – это обмен системы «клиент-банк» с учетной системой через файл специального текстового формата, содержащий информацию о платежных поручениях. Файл при этом выгружает и загружает оператор. Либо настроен прямой обмен и промежуточные файлы не используются, либо никакого обмена нет и все платежи перебиваются вручную в обе системы. На схеме это можно рисовать разным типом стрелок или подписывать, как именно связаны системы.

Все используемые программные средства и учетные системы размещаются на каких-то серверах, собранных в сеть, возможно, территориально распределенную (в случае холдинга или филиальной структуры организации). Либо все серверы существуют в виртуальном пространстве облачных сервисов. На схеме можно отметить, где именно размещаются конкретные программы.

Итак, с чего начать составление карты ИТ-ландшафта «как есть»?

- С составления списка используемых систем и программных средств.
- Часто для каких-то задач может использоваться база в виде Excel-файла, такие базы тоже нужно учитывать.
- На схеме каждый объект, содержащий данные, рисуется отдельным элементом.
- Элементы объединяются в смысловые группы, функциональные разделы.
- Между элементами указываются стрелки потока данных (одно- или двусторонних обменов).

Рассмотрим пример такой схемы «как есть». В холдинге, состоящем из нескольких юридических лиц, есть множество бухгалтерских систем. По каждому юрлицу – своя система, так сложилось исторически, велись разными бухгалтерами разрозненно. Для ведения зарплаты и кадров используется единая система. Используются разные банки со своими системами «клиент-банк». Управленческий учет на предприятии первичный, ведется в отдельной системе и базах в Excel-файлах, где по результатам деятельности собирается управленческая отчетность

по холдингу. Присутствует связь с бухгалтерскими системами по первичным документам, а также с внешней системой заказов у основного поставщика с интеграцией через web-сервис.

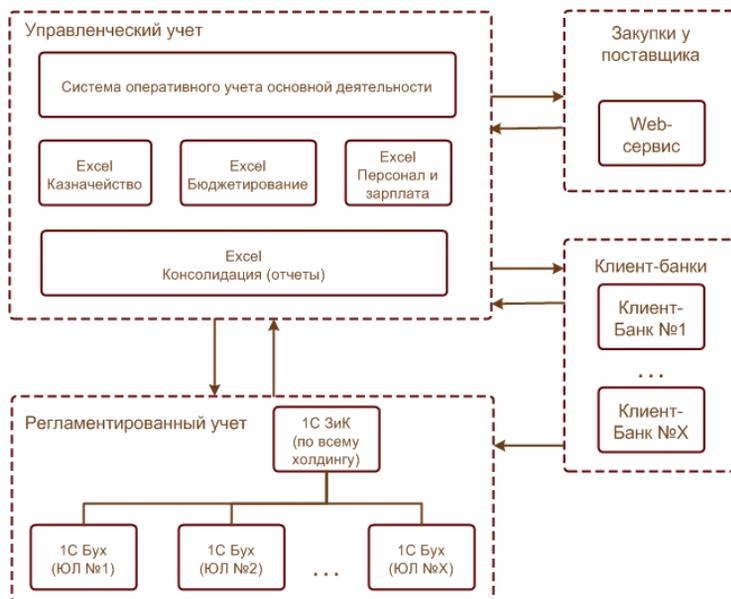


Рис. 2.2. Пример схемы. ИТ-ландшафт систем «как есть»

Целевая задача – это отказ от разрозненных систем и файлов, перевод всех бизнес-единиц холдинга в единое информационное пространство ERP-системы.

Как рисовать перспективную схему «как будет»?

- Выявить элементы, которые будут заменяться, объединяться или как-то меняться в процессе проекта автоматизации.
- Представить, что проект уже завершился и все элементы, что заменяются на ERP-систему, убираются из схемы и вместо них ставится элемент ERP-системы.
- Нарисовать связи, которые остаются, но уже применительно к ERP-системе.

В нашем примере получается такая целевая схема «как будет».



Рис. 2.3. Пример схемы. ИТ-ландшафт систем «как будет»

Так как у компании есть удаленные подразделения, то часть пользователей работают не в локальной сети компании, а через интернет-доступ. Схема серверов в компании из примера может выглядеть так.

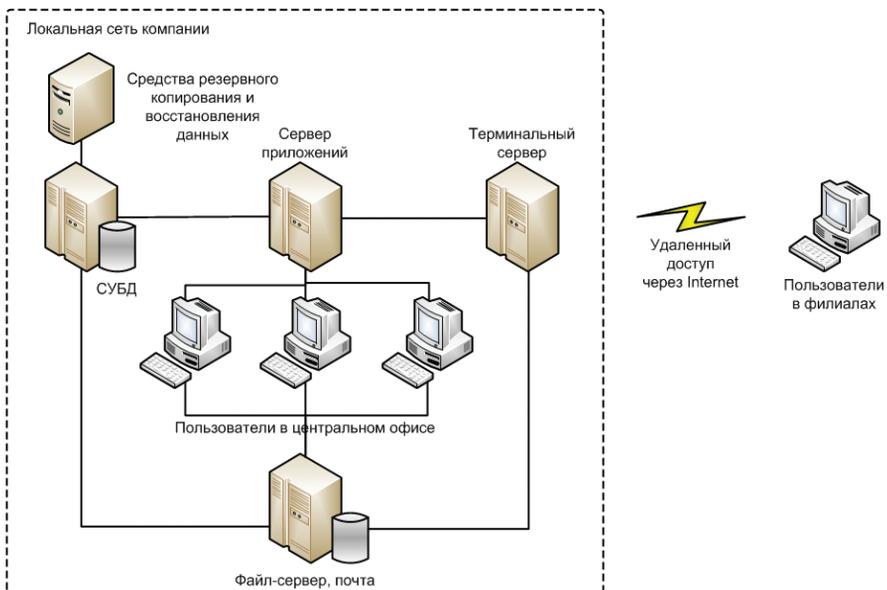


Рис. 2.4. Пример общей схемы аппаратных компонентов сети компании

На практике получить сразу целевое состояние информационных систем затруднительно. Проект автоматизации идет какое-то время (порой значительное – месяцы, годы) параллельно с непрерывающейся деятельностью компании. Нужно сдавать отчетность, взаимодействовать с другими системами. Принимается решение об очередности (этапах) внедрения функциональности. Обычно хочется начать внедрение с «горящих» мест, автоматизация которых дает быстрый эффект для бизнеса и обеспечивает окупаемость проекта внедрения на ранних стадиях. Работающие и решающие свои задачи прочие блоки могут отойти на вторую очередь и остаться временно на старых механизмах.

Так, в промежуточном состоянии наш пример может иметь следующую схему целевого состояния первой очереди автоматизации.

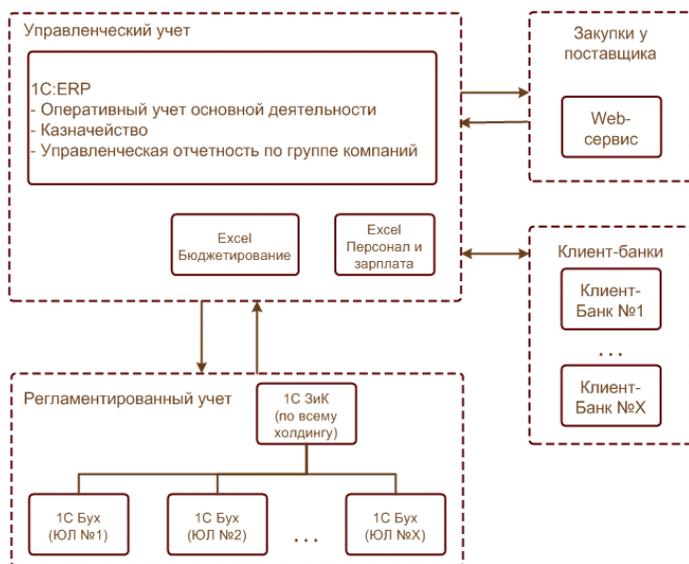


Рис. 2.5. Пример схемы. ИТ-ландшафт систем после первой очереди автоматизации

Отрисовка разных вариантов схем ИТ-ландшафта компании позволяет моделировать разные состояния и очередность внедрения подсистем ERP-системы. Это позволяет находить компромисс между желанием «все и сразу» и реальными возможностями: ресурсами исполнителя, бюджетом проекта, готовностью и восприятием нововведений сотрудниками на местах.

Такие схемы позволяют говорить на одном языке специалистам заказчика и исполнителя, зафиксировать текущие и целевые состояния КИС по этапам проекта и начать работу по достижению этих состояний на практике.

Нагрузочные тесты и выбор «железа»

С одной стороны, на этапе выбора ERP-системы употреблять понятие «нагрузочные тесты» еще преждевременно, т.к. ничего же не выбрано и «нагружать» вроде бы нечего. С другой стороны – об этом нужно помнить и учитывать изначально, т.к. выбираемая система должна справляться с ожидаемым объемом данных. И тут мы приходим к пониманию, что на момент выбора системы нужно знать этот самый «ожидаемый объем данных» и еще его динамику изменения (прироста) по годам использования системы. А также нужен ориентир на потребное для работы оборудование (и его стоимость или плату за сервисы предоставления вычислительных мощностей в аренду).

Это одно из важных нефункциональных требований – производительности системы (в данном случае аппаратной и программной части) должно хватать на сегодня, завтра и пару лет перспективы.

Поэтому этот раздел помещен тут, в самом начале, в главу о выборе ERP-системы. Сами нагрузочные тесты конкретного функционала будут интересны на реальных данных и уже готовой к опытной эксплуатации системе (или ее блоках по мере готовности). На этапе выбора системы можно воспользоваться информацией от поставщика системы по проводимым замерам, референс-проектам с аналогичной бизнес-структурой и процессами. Хватает ли им производительности, какое «железо» они для этого используют? Можно оценить для себя по аналогии.

Дополнительно нужно оценить ожидаемый объем данных и динамику его прироста в месяц/год:

- число организаций и подразделений, какой прирост ожидается. Если ожидается подключение новых организаций/подразделений, то нужно вводить типовые расчеты по типам подключаемых единиц по списку ниже;
- количество контрагентов и их прибавление в месяц, в год;

- количество заключаемых договоров, заказов в месяц, в год;
- количество документов по основной деятельности (закупки, продажи, производство) в день, месяц, год;
- количество оплат и поступлений денежных средств в день, месяц, год;
- количество касс, ККМ, р/с, клиент-банков, эквайринговых терминалов;
- количество внеоборотных активов и динамика изменений;
- прочие документы и договоры по неосновной деятельности;
- число сотрудников, динамика изменений и многое другое.

На непосредственную нагрузку на серверы (количество процессорных ядер и ОЗУ) влияет:

- количество одновременно работающих пользователей;
- что именно они делают, типовой сценарий работы;
- режим работы (с учетом часовых поясов, смен и дней недели);
- объем данных, используемых в запросах для пересчетов регламентными операциями (например, процедурой закрытия месяца).

Детальные рекомендации по подбору оборудования даны на сайте «1С» в разделе для технических специалистов:

<https://its.1c.ru/db/metod8dev#content:5810:hdoc>

У фирмы «1С» для целей нагрузочного тестирования есть специальный инструментарий – Тест-центр. **Тест-центр** – инструмент автоматизации многопользовательских нагрузочных испытаний информационных систем на платформе «1С:Предприятие 8». С его помощью можно моделировать работу предприятия без участия реальных пользователей, что позволяет оценивать применимость, производительность и масштабируемость информационной системы в реальных условиях.

Подробнее можно почитать на сайте: http://v8.1c.ru/expert/tc/tc_overview.htm.

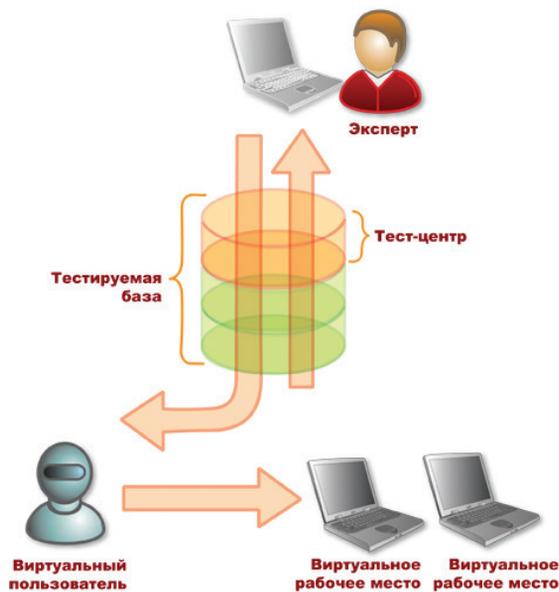


Рис. 2.6. Общая схема работы Тест-центра

В общем случае сценарий нагрузочного теста ERP-системы для выбора «железа» такой:

- Подготовить нагрузочный тест для Тест-центра.
- Запустить нагрузочный тест на одном сервере.
- По количеству пользователей, которые будут работать с системой, пересчитать полученные результаты в требуемые параметры оборудования.
- Реализовать инфраструктуру так, чтобы она была горизонтально масштабируема (мало ресурсов – докупили сервер, ресурсы увеличились).

Горизонтальное масштабирование для СУБД и серверов приложений поддерживается, то есть можно начинать работу, исходя из расчетного объема данных и количества пользователей, а по мере роста бизнеса,

количества сотрудников и данных можно увеличивать ресурсы оборудования, подключая новые серверные единицы в кластер серверов или дисковые массивы в существующие сервера.

В случае использования облачных технологий масштабирование происходит за счет выделения дополнительных вычислительных мощностей на стороне поставщика услуги. Это, очевидно, потребует дополнительных расходов (рост абонентской платы), но может оказаться дешевле (и быстрее) приобретения и настройки оборудования в локальной сети компании.

Окупаемость и обоснование затрат на автоматизацию

В завершение главы по выбору системы рассмотрим важный вопрос обоснования затрат на автоматизацию и ожидаемый срок окупаемости.

В первой главе мы рассмотрели, зачем ERP-система компании, из чего состоят затраты на приобретение и запуск ERP-системы и последующую стоимость владения такой системой, рассмотрели варианты оптимизации расходов за счет использования облачных сервисов.

Понимание самих сумм затрат и бюджета проекта, выявленных по результатам тендера и записанных в договор с будущим исполнителем, требует дополнительного обоснования предстоящих затрат перед учредителями как заинтересованными сторонами. Это внутренний документ заказчика и готовится его же специалистами для внутренних нужд. Исполнитель может даже не увидеть этот документ или, наоборот, помочь составить его – это зависит от содержания информации (целей автоматизации) и открытости приводимых сумм и критериев.

Нужно отметить, что прямой экономии и даже окупаемости может не быть, но автоматизация необходима из-за требований законодательства, и несоответствие этим требованиям грозит штрафами или вообще закрытием бизнеса. Тогда обоснование в этом и состоит: автоматизация – это необходимое условие продолжения непрерывной деятельности компании.

Другой простой расчет – это экономия на фонде оплаты труда (ФОТ) от фиксации роста штата сотрудников при росте бизнеса (числа сделок, товаропотока и прочего). Берутся число ненанимаемых сотрудников, их оклады и налоги с них, аренда помещений и их рабочие места – получается месячная сумма расходов, возникающих без автоматизации. Далее приводим сумму к году или паре лет, сопоставляем с бюджетом автоматизации и вычисляем срок окупаемости.

Более сложные обоснования – увеличение доли рынка, прозрачность бизнеса, оптимизация в нем, аудируемость третьей стороной, оптимизация оборотов запасов, снижение затрат на хранение запасов. Такие понятия оцифровать сложно, но как текстовое обоснование для заинтересованных в экономическом процветании компании лиц они тоже годятся.

Можно совместить все вышеперечисленные виды выгод от автоматизации. По факту так и получается – автоматизация может дать сразу несколько позитивных достижений, что в совокупности дает синергетический эффект для всего бизнеса.

Дополнительно можно добавить в обоснование информацию по бизнес-преимуществам от внедрения аналогичной системы в компаниях с похожим бизнесом. Компании, внедряющие ERP-системы, участвуют в тематических конференциях и часто делают доклады о ходе внедрения и полученных компанией бизнес-преимуществах от работающей некоторое время после запуска системы, а также о преследуемых изначально целях автоматизации. Такие или похожие цели могут быть и у нашей компании. На сайте фирмы «1С» приводится средняя статистика экономического эффекта по проектам внедрения ERP-систем – http://v8.1c.ru/erp/economic_effect.htm. В таблице приводятся данные по 136 опубликованным проектам внедрения с экономическими показателями, подтвержденными клиентами.

Окупаемость и обоснование затрат на автоматизацию

Раздел	Показатель эффективности	Среднее значение
Запасы и производство	Снижение объемов материальных запасов	24 %
	Сокращение расходов на материальные ресурсы	17 %
	Снижение производственных издержек	16 %
	Снижение себестоимости выпускаемой продукции	9 %
	Увеличение объема выпускаемой продукции	36 %
	Рост производительности труда в производстве	33 %
Оборотные средства	Рост оборачиваемости складских запасов	28 %
	Сокращение дебиторской задолженности	22 %
Эффективность и оперативность	Ускорение обработки заказов	75 %
	Сокращение сроков исполнения заказов	26 %
	Сокращение операционных и административных расходов	17 %
	Рост прибыли	14 %
Трудозатраты и отчетность	Сокращение трудозатрат в различных подразделениях	29 %
	Ускорение получения управленческой отчетности	В 2,9 раза
	Ускорение подготовки регламентированной отчетности	В 2,8 раза

Защитив и обосновав бюджет на проект внедрения перед учредителями и иными заинтересованными сторонами, мы переходим к вопросам непосредственно проекта внедрения и управления им.

Глава 3. Как внедрять ERP-систему

Технологии управления проектами

Внедрение корпоративных информационных систем класса ERP требует обязательного соблюдения методологии управления проектом, выбранной и утвержденной перед началом основных работ.

Другое дело, какую методологию выбрать, как все распланировать, оценить, соблюдать и контролировать выполнение работ, минимизировать различные риски и проблемы?

В этой главе рассмотрим, какие вообще бывают технологии управления проектами, отличие их от методологий и основные понятия и определения. В следующих главах будут рассматриваться практические вопросы внедрения.

Технологии и стандарты управления проектами можно разделить на три условные категории:

- **Гибкие** (Agile), такие как Scrum, Kanban, Extreme programming. Представляют собой:
 - Семейство «гибких» подходов к разработке программного обеспечения. Такие подходы иногда называют фреймворками или agile-методологиями.

- Манифест: *«Люди и взаимодействие важнее процессов и инструментов. Работающий продукт важнее исчерпывающей документации. Сотрудничество с заказчиком важнее согласования условий контракта. Готовность к изменениям важнее следования первоначальному плану».*
- Бюджет всего проекта заранее неизвестен, есть примерные оценки, работа по факту (time in material).
- **Классические**, такие как ISO 21500, ГОСТ 34, PMBOK. Представляют собой:
 - Каскадные, или «водопадные», модели (waterfall model) – поток из последовательности этапов/фаз работ над проектом.
 - Подробное планирование и проектирование до начала работ (самой разработки), подробное документирование всех стадий проекта и взаимодействия участников.
 - Бюджет заранее известен и часто фиксирован (fix price).
- Когда нет методологии:
 - Основная идея: *«Зачем какие-то методологии и управление, просто сделайте/сделаем, что просим/просите».*
 - Может трактоваться как подвид гибких технологий, но это не так.
 - Сама методика не формализована либо имеет атрибуты, например, классических подходов. Все равно используется какой-то план дел в голове, так как нужно же как-то к графику оплат по договору привязывать какие-то работы.
 - На практике встречается в небольших командах из 1–2 специалистов и в краткосрочных работах.
 - На выходе нет никаких проектных документов (т.к. и проекта же нет, просто какие-то услуги за какое-то время).

Все на вербальной коммуникации, квалификации конкретного специалиста и проверке результата его работ.

- Управления рисками нет (а сами риски есть), сроки и объем работ проконтролировать сложно. Типичная итерация: «Готово?» – «Да, почти» – «Когда будет?» – «На следующей неделе», – и так каждую неделю. Понять, где мы сегодня, практически нельзя – точнее, можно, но для этого нужно на этот «непроект» применить какую-то методологию и формально описать, что было нужно сделать, что уже сделано, что осталось, какие трудозатраты и ресурсы.
- ERP-системы так внедрять настоятельно не рекомендуется. Почему – станет понятно из последующих глав, из перечня того, что нужно учитывать. Эта категория выделена не просто так, а исходя из суровой реальности: такой подход встречается на практике. Рассматривать детально эту категорию, очевидно, не будем. Такого просто не должно быть.

Agile-методологии

Для начала нужно сказать, что даже PMI перестал считать гибкие методологии лженаукой и на сегодня в PMBOK входят описания гибких и гибридных проектных технологий.

В случае разработки софта в целом и определенной функциональности при развитии ERP-системы гибкие методологии хорошо помогают добиваться успеха быстро и качественно. Далеко не всегда можно полноценно заранее спланировать и зафиксировать в требованиях и дизайне подробный план работ. Иногда проще развивать готовый прототип или уже запущенную систему итерациями, небольшими модификациями, дающими сразу готовый результат для пользователей (и обратную связь от них) и сохраняющими работоспособную систему.

Для внедрения ERP-системы с нуля такой способ может не подойти, тут потребуются первично все спроектировать, спланировать и двигаться по этапам. Хотя на практике для ситуаций, когда система ERP ставится на новый бизнес (стартап) и компания готова подстраиваться под возможности системы «из коробки», гибкие методологии применимы и дают быстрый результат. Например, быстрое внедрение 1С:ERP за три месяца

в объеме, достаточном для работы компании: производство, закупки, продажи, регламентированный учет. А уже следующими итерациями, в рамках проекта развития, довнедрение прочих подсистем: казначейство, бюджетирование.

Мнение практиков в поддержку гибких подходов для ERP-проектов из круглого стола семинара по 1С:ERP в апреле 2019 г.: «ERP-система замечательно внедряется частями. Главное – грамотно спланировать этапы внедрения. А функционал системы настолько велик, что спроектировать все от и до качественно заранее очень сложно, поэтому можно использовать гибкие подходы с короткими итерациями выпуска рабочей версии».

Гибкие методологии хорошо подходят для частых выпусков работоспособных версий. И в связи с этим с успехом применяются при разработке и развитии некоторых программных продуктов с частой доставкой новой функциональности до конечного пользователя.

Обычная практика и мнение против гибких методик для ERP: «Внедрение ERP-системы часто проходит по классическим проектным методологиям в силу объемности и сложности внедряемой функциональности, когда нужно все четко спроектировать, спланировать и оценить».

На сегодняшний день грань между методологиями, в принципе, размывается и хорошо сочетаются гибкий подход с классическим – так называемые **гибридные подходы**, например: высокоуровневое планирование и этапность проекта, а внутри этапа разработки – гибкие итерации «разработка – тестирование» и частый выпуск релизов в эксплуатацию.

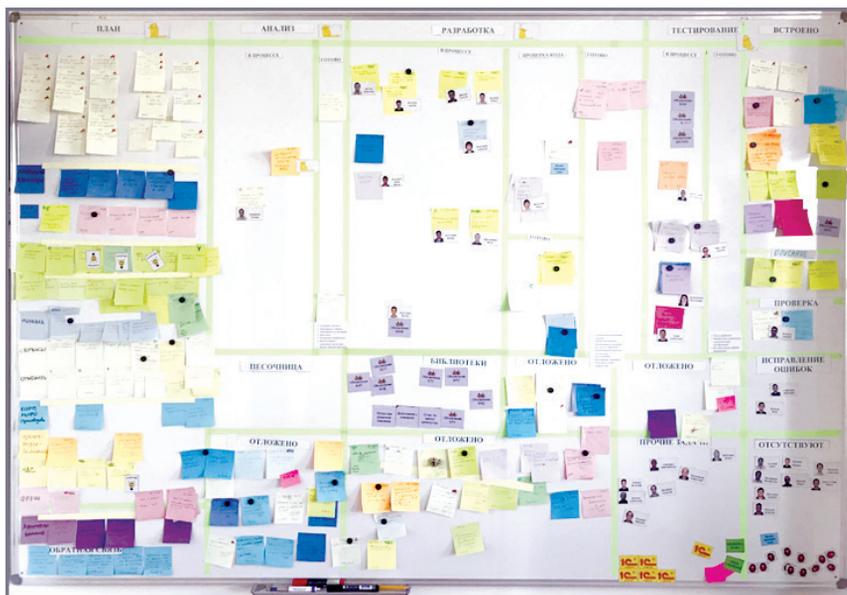


Рис. 3.1. Фото с примером доски задач по agile-методологии

Гибкие методологии возникли в ИТ-сфере в начале 2000-х годов, потом стали применяться в других сферах деятельности.

В гибких методологиях есть своя философия, и для того, чтобы привить полноценно методологию компании и сотрудникам, нужно приложить определенные усилия. На рынке присутствуют компании – тренеры agile, которые проводят обучение команд и помогают выстроить весь процесс с нуля.

Будет заблуждением считать, что «гибкость» – это «когда нет методологии». Нет! Во всех методиках есть свои правила, которые нужно соблюдать, какими бы странными они ни казались поначалу.

На практике в конкретной компании происходит трансформация гибких методологий из оригинальной системы в удобную для бизнес-процессов компании, так как начинаются отступления от оригинальной системы: «Мы только разик сделаем как обычно, а потом снова все как положено по методологии». В итоге получается адаптация и получение какой-то гибридной методологии. Это нормально, главное, чтобы методология вообще была и давала положительный результат.

В данной книге не будет подробного рассмотрения гибких методологий и отличий Scrum от Kanban или как организуется парное программирование в Extreme programming. Это все очень интересно и можно изучить самостоятельно.

Если есть желание применять гибкие методики, лучше опираться на более классические проектные технологии, чтобы было полное понимание, от чего отказываться в пользу «гибкости» и к чему это может привести. А не по принципу: «У нас все гибко, и всякие там непонятные вещи из управления проектами нам совсем не нужны».

При этом совместить гибкие методологии с классическим подходом именно в разработке (а не консалтинге, анализе бизнес-процессов и дизайне системы) вполне можно: после декомпозиции (пусть высокоуровневой) на задачи саму разработку можно гибко вести «спринтами» и итерациями на быстрый результат с вовлечением заказчика и минимумом проектной документации.

Из опыта практиков гибких методик для внедрения ERP: «Спринты и итеративность при моделировании тоже крайне эффективны, а не только на этапе разработки. Также сам этап запуска системы спринтами, разбитыми по разной функциональности и пользователям, позволяет сильно сократить градус экстрима на опытной эксплуатации».

Так что внедрение ERP по гибким методикам в ряде случаев вполне возможно – выбор тут за проектной командой.

Серия стандартов ISO

ISO 9000 – серия международных стандартов (англ. International Standard Organization), содержащих термины и определения, основные принципы менеджмента качества, требования к системе менеджмента качества организаций и предприятий, а также руководство по достижению устойчивого результата.

Стандарты серии ISO 9000, принятые более чем 190 странами мира в качестве национальных, применимы к любым предприятиям независимо от их размера, форм собственности и сферы деятельности.

Появился стандарт в 1987 году, обновляется периодически, иногда с полным пересмотром. Так, в 2015 г. он был серьезно пересмотрен в очередной раз.

В России аналогами ISO 9000-2015 и ISO 9001-2015 являются ГОСТ Р ИСО 9000-2015 и ГОСТ Р ИСО 9001-2015, они введены в действие 1 ноября 2015 года, утверждены в Росстандарте 28 сентября 2015 года.

Это стандарт по управлению качеством. Стоит отметить, что это не требования к определенному продукту или услуге, не гарантия их качества. Это требования к системе управления, но результат, конечно, сказывается на конечном продукте.

Основной цикл процесса контроля качества – «**Планируй – Делай – Проверьй – Действуй**», или **PDCA** (англ. Plan – Do – Check – Act). Иначе его называют циклом Шухарта – Деминга (Уолтер Шухарт и Эдвард Деминг – американские ученые, консультанты по менеджменту качества).

Цикл PDCA может быть применен ко всем процессам и к системе менеджмента качества в целом:

- **планируй** – разработка целей системы и ее процессов, а также определение ресурсов, необходимых для достижения результатов в соответствии с требованиями потребителей и политикой организации, определение и рассмотрение рисков и возможностей;
- **делай** – выполнение того, что было запланировано;
- **проверьй** – мониторинг и измерение процессов, продукции и услуг в сравнении с политикой, целями, требованиями и запланированными действиями и сообщением о результатах;
- **действуй** – принятие мер по улучшению результатов деятельности в той степени, насколько это необходимо.



Рис. 3.2. Схема цикла PDCA

Что дает компании сертификация по ISO 9001:

- преимущество при участии как в коммерческих тендерах, так и в госзаказах;
- возможность выхода не только на общероссийский, но также и зарубежный рынок;
- подтверждение выхода компании на высокий уровень организации бизнеса;
- подтверждение безупречной репутации, достижений и перспектив компании в глазах как клиентов, так и партнеров.

Можно рекомендовать самостоятельное изучение документации по ISO 9001 (есть русский перевод в виде ГОСТ Р ИСО 9001) для полноты понимания, что предполагается и ожидается от компаний, которые борются за повышение качества своих продуктов и услуг.

Непосредственно к проектам внедрения ERP эти стандарты не относятся, они позволяют выстроить взаимодействие всех сторон проекта и повысить качество оказываемых услуг (в нашем случае по внедрению). Это не методология внедрения КИС, а внеотраслевая технология борьбы за качество.

Есть стандарты ISO серии 10000 – технологии поддержки и удовлетворенность потребителей (например, ГОСТ Р ИСО 10003-2009).

Есть стандарт **ISO 21500:2012** (и его аналог ГОСТ Р ИСО 21500-2014) «Руководство по проектному менеджменту» – вот этот стандарт особо интересен для внедрения ERP-систем. В основе лежит РМВОК, поэтому это не какая-то альтернатива, а формальная фиксация наработок от PMI.

ГОСТ 34

ГОСТ 34.xxx «Единый комплекс стандартов и руководящих документов на автоматизированные системы» – комплекс государственных стандартов СССР, межотраслевых руководящих документов и методических материалов, задуманный в конце 1980-х годов.

Преимущества:

- лаконичность и доступность для непрофильных специалистов;
- краткость описания (включает несколько ГОСТов по 15–25 стр., что относительно 750+ страниц того же РМВОК очень облегчает изучение);
- самодостаточность;
- широкая распространенность на территории РФ и ряда стран СНГ, привычность терминологии и базовых понятий;
- минимальный набор жестких требований и возможность адаптации требований стандартов под конкретные условия тех или иных проектов.

Недостатки:

- избыточные и несовременные требования по оформлению документов (недостаток существенный: документы делать долго и дорого, а потом их нужно читать и согласовывать, а документы перегруженные получаются);
- стандарты не являются процессно-ориентированными, что лишает их перспектив развития и затрудняет использование совместно с более современными стандартами;

- отсутствует целый ряд ключевых понятий современного управления проектами, таких как риски, программы проектов и портфели проектов, как следствие, их проработка стандартом не предусмотрена;
- незавершенность комплекса стандартов. Имеющиеся стандарты не образуют целостную систему и больше не актуализируются;
- очень поверхностно рассматриваются аспекты обслуживания внедренной автоматизированной системы: обучение персонала, выполнение регламентных процедур и т. п.

Изучить состав и терминологию ГОСТов 34-й серии рекомендуется самостоятельно. Объем этих документов в страницах вполне краткий для самостоятельного изучения. Это, конечно, не методика управления проектами, но полезные и формальные вещи в описании присутствуют. Столкнуться с ГОСТ 34 можно на внедрениях в государственных компаниях или в ситуациях с привлечением целевого финансирования из бюджета, когда контроль расходования средств будет со стороны государства и требования к ведению проектной документации могут быть как раз по ГОСТ 34. Также эти стандарты могут требоваться в автоматизации коммерческих компаний, руководство которых имеет успешную практику применения этого ГОСТ в других организациях (а о каких-то других методологиях и слышать не хочет).

Нужно отметить, что сейчас многие ведущие предприятия требуют сертификацию PMI или ISO 21500. Так что все уже в курсе современных методик. Поэтому адаптировать и оптимизировать документацию по ГОСТ 34 под современное ведение проектов на конкретных внедрениях, скорее всего, получится.

Если стоит выбор между отсутствием проектной методологии вообще или применением стандартов и шаблонов документов от ГОСТ 34 – выбор очевиден. Применяя совместно с ГОСТ Р ИСО 21500-2014, можно получить отличную систему проектного управления для автоматизированных систем.

PMBOK

PMBOK (англ. Project Management Body of Knowledge) – свод знаний по управлению проектами, разрабатываемый и поддерживаемый институтом по управлению проектами **PMI** (англ. Project Management Institute).

На момент написания данной книги актуальным является 6-е издание PMBOK от 2017 года.

Давайте разберемся, что это такое и чем поможет конкретно в проектах внедрения ERP-систем.

Стандарт управления проектом изначально был разработан как стандарт Американского национального института стандартов (American National Standards Institute, **ANSI**). Стандарт лежит в основе книги и представлен во второй части. Нужно понимать, что более 80 % книги – это уже не стандарт, а именно свод знаний по теме.

PMBOK – это фреймворк (свод, система, концепция) из знаний по теме управления проектами. Таблица ниже наглядно показывает динамику развития проектных технологий в виде их формального описания в своде знаний: за 20 лет в 4 раза вырос объем информации (знаний) об управлении проектами.

Следующий свод знаний, с учетом 4–5-летних квантов, ожидается в 2021–2022 году (см. таблицу ниже).

Важно отметить, что сам PMBOK из-за универсальности своих рассматриваемых проектных технологий и применимости их к любой сфере деятельности, по сути, проектной методологией не является. Это следует из определения методологии в самом своде знаний (в следующем разделе будут рассматриваться определения, вводимые в PMBOK).

Используя все эти знания, нужно сформировать свою проектную методологию для конкретной компании и проекта внедрения ERP-системы. Либо использовать готовые методологии, например 1С:ТКВ (что это, будет описано ниже).

Издание PMBOK	Характеристики
№ 1 1996 год	<ul style="list-style-type: none"> ● 176 стр. ● 9 областей знаний ● 37 процессов
№ 2 2000 год	<ul style="list-style-type: none"> ● 211 стр. ● 9 областей знаний ● 39 процессов
№ 3 2004 год	<ul style="list-style-type: none"> ● 390 стр. ● 9 областей знаний ● 44 процесса
№ 4 2009 год	<ul style="list-style-type: none"> ● 467 стр. ● 9 областей знаний ● 42 процесса
№ 5 2013 год	<ul style="list-style-type: none"> ● 589 стр. ● 10 областей знаний ● 47 процессов
№ 6 2017 год	<ul style="list-style-type: none"> ● 756 стр. ● 10 областей знаний ● 49 процессов ● + Agile Practice Guide 167 стр.

Также отрезвляющим является и уведомление (соглашение) по предоставленным материалам в самом своде знаний: *PMI не несет ответственность за какие-либо травмы, повреждения, нанесенные собственности, или какие-либо другие убытки, будь то реальные, косвенные или компенсаторные, произошедшие непосредственно или косвенно вследствие издания, применения или использования данного документа. PMI не несет ответственность и не предоставляет гарантию, прямую или предполагаемую, относительно точности или полноты любого материала, содержащегося в данном документе, а также не несет ответственность и не предоставляет гарантию того, что содержащаяся в данном документе информация отвечает каким-либо вашим целям или нуждам. PMI не предоставляет гарантию относительно качества каких-либо продуктов или услуг отдельного производителя или продавца, проистекающего из использования данного стандарта или руководства.*

Тут все честно: свод знаний сам по себе не может повлиять на применение этих знаний, которые помогают большинству добиваться успеха. Конкретно в чьих-то руках (головах) все может пойти совсем не так. Нужно думать своей головой и принимать ответственность на себя, а свод знаний – это лишь помогающий в работе инструментарий (фреймворк).

В РМВОК выделяются 49 главных процессов, происходящих при управлении проектами. Эти процессы разделены на пять основных групп:

- Процессы **инициации** – необходимы для определения и авторизации проекта или его фазы.
- Процессы **планирования** – необходимы для определения и уточнения целей, планирования действий по достижению этих целей.
- Процессы **исполнения** – необходимы для объединения человеческих и прочих ресурсов для выполнения плана;
- Процессы **мониторинга и контроля** – необходимы для регулярной оценки прогресса проекта, обнаружения отклонений и корректировки действий.
- Процессы **закрытия** – необходимы для формализации приемки результата проекта, подведения проекта или его фазы к завершению.

Каждый из этих процессов относится к одной из десяти областей знаний, определяемых РМВОК:

1. **Управление интеграцией.** Под интеграцией понимается принятие решений на тему концентрации ресурсов; попытки предугадать потенциальные проблемы и разрешить их до перехода в критическое состояние; координирование работы над проектом. С помощью интеграции можно находить компромиссы между пересекающимися целями и альтернативными вариантами.
2. **Управление содержанием.** Сюда относятся такие процессы, как создание иерархической структуры работ по проекту, определение, планирование, подтверждение и управление содержанием.

3. **Управление расписанием (сроками).** Здесь определяется состав операций и взаимосвязи между ними, оцениваются ресурсы и длительность операций, разрабатывается расписание и производится управлением им.
4. **Управление стоимостью.** Имеется в виду разработка бюджета и контроль затрат. Для успешной реализации проекта осуществляется стоимостная оценка, разрабатывается бюджет расходов и производится управление стоимостью.
5. **Управление качеством.** Эта область включает все процессы, связанные с выполнением целей. К ним относятся планирование, обеспечение и контроль качества.
6. **Управление ресурсами.** Направление действий – организация проектной команды и управление ей. Планируются человеческие ресурсы, набирается и развивается коллектив, предпринимаются меры по управлению командой.
7. **Управление коммуникациями.** Планируются коммуникации, распространяется информация, создается отчетность по исполнению, происходит управление участниками проекта.
8. **Управление рисками.** Процессы, относящиеся к данной области, – это планирование управления рисками, идентификация, качественный и количественный анализ рисков, планирование реагирования, мониторинг и управление рисками.
9. **Управление поставками.** К этой области относится планирование покупок и контрактов, запрос информации у поставщиков, подбор поставщиков, администрирование и закрытие контрактов.
10. **Управление заинтересованными сторонами.** Процессы, необходимые для выявления людей, групп и организаций, которые могут оказывать воздействие на проект или на которых проект может оказывать воздействие.

Таблица сопоставления групп процессов управления проектом и областей знаний с процессами на пересечениях.

	Инициация	Планирование	Исполнение	Контроль	Закрытие
Интеграция	<ul style="list-style-type: none"> Разработка устава проекта 	<ul style="list-style-type: none"> Разработка плана управления проектом 	<ul style="list-style-type: none"> Руководство и управление работами проекта Управление знаниями в проекте 	<ul style="list-style-type: none"> Мониторинг и контроль работ проекта Интегрированный контроль изменений 	<ul style="list-style-type: none"> Закрытие проекта или фазы
Содержание		<ul style="list-style-type: none"> Планирование управления содержанием Сбор требований Определение содержания Создание иерархической структуры работ 		<ul style="list-style-type: none"> Подтверждение содержания Контроль содержания 	
Расписание		<ul style="list-style-type: none"> Планирование управления расписанием Определение операций Определение последовательности операций Оценка ресурсов операций Оценка длительности операций Разработка расписания 		<ul style="list-style-type: none"> Контроль расписания 	
Стоимость		<ul style="list-style-type: none"> Планирование управления стоимостью Оценка стоимости Определение бюджета 		<ul style="list-style-type: none"> Контроль стоимости 	

	Инициация	Планирование	Исполнение	Контроль	Закрытие
Качество		<ul style="list-style-type: none"> Планирование управления качеством 	<ul style="list-style-type: none"> Управление качеством 	<ul style="list-style-type: none"> Контроль качества 	
Ресурсы		<ul style="list-style-type: none"> Планирование управления ресурсами 	<ul style="list-style-type: none"> Набор команды проекта Развитие команды проекта Управление командой проекта 	<ul style="list-style-type: none"> Контроль ресурсов 	
Коммуникации		<ul style="list-style-type: none"> Планирование управления коммуникациями 	<ul style="list-style-type: none"> Управление коммуникациями 	<ul style="list-style-type: none"> Мониторинг коммуникаций 	
Риски		<ul style="list-style-type: none"> Планирование управления рисками Идентификация рисков Качественный анализ рисков Количественный анализ рисков Планирование реагирования на риски 	<ul style="list-style-type: none"> Применение реагирования на риски 	<ul style="list-style-type: none"> Мониторинг рисков 	
Покупки		<ul style="list-style-type: none"> Планирование управления закупками 	<ul style="list-style-type: none"> Проведение закупок 	<ul style="list-style-type: none"> Контроль закупок 	
Заинтересованные стороны	<ul style="list-style-type: none"> Определение заинтересованных сторон 	<ul style="list-style-type: none"> Планирование вовлечения заинтересованных сторон 	<ul style="list-style-type: none"> Управление вовлечением заинтересованных сторон 	<ul style="list-style-type: none"> Мониторинг вовлечения заинтересованных сторон 	

Каждое описание процессов в PMBOK состоит из трех основных частей:

- Входы.
- Выходы.
- Инструменты и методы.

Процессы управления проектом логически связаны друг с другом посредством выходов, которые они производят. Процессы могут содержать накладывающиеся друг на друга действия, которые выполняются на протяжении реализации проекта. Результатом выхода процесса является либо вход в другой процесс, либо поставляемый результат проекта или фазы проекта.



Рис. 3.3. Пример входов/выходов процесса «Разработка устава проекта»

Повторение процессов и их взаимодействие варьируются в зависимости от потребностей проекта. В целом процессы попадают в одну из трех категорий:

- **Применяют единожды или в predetermined моменты в ходе реализации проекта.**
 - Примерами могут служить разработка устава проекта и закрытие проекта или фазы.
- **Выполняются периодически, по мере необходимости.**
 - Процесс приобретения ресурсов осуществляется тогда, когда в ресурсах возникает необходимость.

- Процесс проведения закупок осуществляется до возникновения необходимости в закупаемом продукте.
- **Реализуются постоянно на всем протяжении проекта.**
 - Процесс определения операций может происходить на протяжении всего жизненного цикла проекта, особенно в тех случаях, когда в проекте применяется планирование методом набегающей волны или методом адаптивного подхода к разработке.
 - Большая часть процессов мониторинга и контроля реализуются постоянно с момента начала проекта до его закрытия.

Управление проектом осуществляется, таким образом, через применение и интеграцию между собой логически сгруппированных процессов.

Институт PMI запустил первую программу сертификации с присвоением статуса **Профессионал в управлении проектами** (Project Management Professional, **PMР**) с середины 1980-х годов. Для получения сертификата PMI кандидаты должны документально подтвердить соответствие требованиям к образованию и опыту. Далее необходимо пройти тестирование, состоящее из вопросов с несколькими вариантами ответов: 200 вопросов за 4 часа с 67 % правильных ответов для успеха.

У PMI есть и другие сертификаты. Большинство сертификатов необходимо продлять каждые три года. Для этого необходимо накапливать **Единицы профессионального развития** (Professional Development Units, **PDU**). Существуют различные способы получения PDU, такие как участие в обучении, посещение международных конгрессов PMI, внесение вклада в профессиональные исследования или написание и публикация статей, посвященных управлению проектами.

Технологии управления проектами фирмы «1С»

Технологии управления проектами внедрения, предлагаемые фирмой «1С», разработаны в соответствии с рекомендациями и требованиями мировых и отечественных стандартов, таких как:

- «Руководство к своду знаний по управлению проектами PMI PMBOK»;
- рекомендации по использованию agile-подходов к организации разработки программного обеспечения, такие как Extreme programming и SCRUM;
- других рекомендаций выдающихся практиков в области разработки и внедрения информационных систем.

Подробное описание приведено на сайте: <https://consulting.1c.ru/technologies/>.

В настоящее время разработаны и применяются для внедрения и сопровождения ERP-решений следующие проектные технологии:

- **«1С:Технология быстрого результата» («1С:ТБР»)** – базовая технология, основанная на agile-подходе, подходит для проектов внедрения типовых и отраслевых решений на платформе «1С:Предприятие 8» (<https://consulting.1c.ru/technologies/tbr/>);
- **«1С:Технология корпоративного внедрения» («1С:ТКВ»)** – «классическая» проектная технология, ориентированная на проекты разработки и масштабные проекты внедрения решений на платформе «1С:Предприятие 8» (<https://consulting.1c.ru/technologies/tkv/>);
- **«1С:Технология корпоративного сопровождения» («1С:ТКС»)** – технология поддержки и сопровождения запущенных в промышленную эксплуатацию систем (<https://consulting.1c.ru/technologies/tks/>).

Перечисленные выше проектные технологии являются прикладными методологиями ведения проектов внедрения и сопровождения КИС, в отличие от универсального и теоретического PMBOK. Они содержат в себе шаблоны проектной документации для внедрения решений фирмы

«1С». Технологии описаны в виде цикла статей в электронном журнале фирмы «1С» «Управляем предприятием» (сайт <http://upr.ru/>). А также входят в свод знаний «1С:ПрофКейс», доступный для партнеров «1С».

Доступна для дистанционного прохождения серия курсов:

- «Ключевые вопросы управления требованиями в проектах»;
- «Эффективное планирование работ в проекте: разрабатываем план-график проекта»;
- «Правильное инициирование проекта и разработка устава проекта»;
- «Управление рисками проекта с пользой для проекта»;
- «Управление изменениями в проекте»;
- «Распределение ролей в проекте внедрения программных продуктов фирмы «1С»»;
- «Как правильно завершить проект»;
- «Теоретические основы управления проектами по стандарту РМІ РМВОК».

Фирма «1С» проводит оценку и сертификацию экспертов по двум уровням:

- **«1С:Руководитель проектов»** – это эксперт в области управления проектами. Для получения сертификата необходимо документально подтвердить теоретические знания в области общего менеджмента и управления проектами, а также продемонстрировать практический опыт и навыки управления проектами внедрения программных продуктов фирмы «1С»;
- **«1С:Руководитель корпоративных проектов»** – это эксперт в области управления крупными проектами. Основное отличие сертификата «1С:Руководитель корпоративных проектов» заключается в опыте управления масштабными и сложными проектами, которые продемонстрировал обладатель сертификата.

Подробнее о курсах и сертификации см. <https://consulting.1c.ru/services/pm-edu/>.

Введение в терминологию

Для того чтобы двигаться дальше, необходимо ввести и рассмотреть ряд понятий и определений согласно общепринятому своду знаний об управлении проектами РМВОК и стандарту по управлению проектами ISO 21500.

- **Проект** – это временное предприятие, направленное на создание уникального продукта, услуги или результата.
- **Проект** (альтернативное определение из ISO 21500) – это уникальный набор процессов, состоящих из скоординированных и управляемых задач с начальной и конечной датами, предпринятых для достижения цели. Достижение цели проекта требует получения результатов, соответствующих определенным заранее требованиям, в том числе ограничениям на получения результатов, таким как время, деньги и ресурсы.
- **Фаза проекта** – совокупность логически связанных операций проекта, завершающихся достижением одного или ряда поставляемых результатов.
- **Управление проектом** – это приложение знаний, навыков, инструментов и методов к работам проекта для удовлетворения требований, предъявляемых к проекту. Управление проектом – это не просто работа с цифрами, шаблонами, схемами, графиками и компьютерными системами.
 - Общим знаменателем всех проектов являются люди. Людей можно сосчитать, но они не сводятся к числам.
- **Методология** – это система практик, методов, процедур и правил, используемых в определенной сфере деятельности.
 - Тут важно отметить, что сам РМВОК, исходя из этого определения, методологией не является.

- **Адаптация управления проектом к конкретному проекту** – для осуществления управления проектом необходимо выбрать соответствующие процессы, входы, инструменты, методы, выходы, а также фазы жизненного цикла.
 - В процессе адаптации руководитель проекта взаимодействует с командой проекта, спонсором, руководством организации или с некоторыми из них в определенном сочетании. В некоторых случаях организация может требовать применения конкретных методологий управления проектом.
- **Устав проекта** – документ, выпущенный инициатором или спонсором проекта, который формально авторизует существование проекта и предоставляет руководителю проекта полномочия использовать ресурсы организации в операциях проекта.
- **Описание содержания проекта** – документ, в который включены описания всех планируемых работ и результаты поставки, необходимые к выполнению.
- **План управления проектом** – это документ, описывающий, как проект будет исполняться, как будут происходить его мониторинг и контроль.
- **Запрос на изменение** – документ, который определяет предлагаемые изменения в проекте.
- **Жизненный цикл проекта** – определенная последовательность фаз, продолжающаяся от начала до окончания проекта.
- **Заинтересованное лицо/сторона (stakeholder)** – лицо, группа или организация, которая может влиять, на которую могут повлиять или которая может воспринимать себя подвергнутой влиянию решения, операции или результата проекта, программы или портфеля.
- **Руководство проектом** – это структура, функции и процессы, которые регламентируют операции по управлению проектом с целью создания уникального продукта, услуги или результата для достижения организационных, стратегических и операционных целей.

- Не существует единственной модели руководства, одинаково результативной для всех организаций. Модель руководства, чтобы добиться ее результативности, требуется адаптировать с учетом культуры организации, типов проектов и потребностей организации.
- **Руководитель проекта** – лицо, назначенное исполняющей организацией руководить командой и отвечающее за достижение целей проекта.
- **Заказчик** – человек или компания, планирующая использовать продукт (уровней заказчиков может быть несколько).
- **Исполнитель** – компания, обязавшаяся реализовать проект.
- **Проектная команда** – группа лиц, которая поддерживает руководителя проекта в исполнении работ проекта для достижения целей проекта.
- **Команда управления проектом** – члены команды проекта, непосредственно занятые в операциях по управлению проектом.
- **Спонсор** – лицо (или группа лиц), предоставляющее ресурсы и поддержку для проекта, программы или портфеля и ответственное за достижение успеха.
- **Источники влияния** – люди или организации, не связанные напрямую с получением или применением продукта, но способные оказать положительное или отрицательное воздействие на реализацию проекта.
- **Прототипирование** – метод получения предварительных отзывов относительно требований путем предоставления рабочей модели ожидаемого продукта, прежде чем создавать продукт в действительности.
- **Офис управления проектами (ОУП)** – это организационная структура, стандартизирующая процессы руководства проектами и способствующая обмену ресурсами, методологиями, инструментами и методами.

- Сфера ответственности ОУП может варьироваться от функций оказания поддержки в управлении проектами до прямого управления одним или более проектами.

Терминология в PMBOK и ISO приводится значительно полнее, но нам для оперирования в последующих разделах и главах этого введения в терминологию будет достаточно. Рекомендуется самостоятельное изучение глоссария из PMBOK.

Проектный треугольник

Рассмотрим отдельно и подробнее такое важное понятие, как проектный треугольник, или «треугольник управления проектом».

Условно его идею можно сформулировать в одну строку:

«Быстро – Качественно – Дешево: выберите любые два пункта».

Из этого получается:

- хочется быстро и качественно – будет дорого;
- нужно дешево и качественно – будет долго;
- нужно быстро и дешево – будет, очевидно, некачественно.

Как следует из определения проекта – нужно сделать определенный объем работ (закрыть требования к автоматизации) за определенное время (у проекта есть дата начала и окончания) и за определенный бюджет (зависит от привлекаемых к работе ресурсов, нормы прибыли компании-исполнителя и т.п.). Получаем проектный треугольник со сторонами: сроки, ресурсы (стоимость), объем работ.

Все эти величины имеют верхние пределы:

- **разумный срок**, далее которого проект автоматизации теряет смысл: система устареет, бизнесу жить годы без автоматизации тяжело;

- **конечное количество денег** у компании, которое она готова заплатить за проект;
- **конечный список требований**, который нужно автоматизировать (список со стороны заказчика может быть изначально большим, но все равно у него где-то есть конец).



Рис. 3.4. Проектный треугольник

Качество, как четвертый параметр, находится внутри треугольника и напрямую зависит от всех трех граней – как результат того, что делается с объемом работ за отведенное время и выделенные средства.

Внесение изменений в одну из сторон треугольника меняет как минимум еще одну связанную сторону.

Нематериальность выходной продукции в проекте автоматизации создает иллюзию того, что для внесения изменения в список требований или в уже реализованные требования нужны незначительные усилия. Это заблуждение! Для понимания можно приводить аналогию со стройкой и сносом и переделкой кирпичной стены, если ее нужно сдвинуть, например, на 15 см. «Кирпичики» кода системы тоже нужно разобрать и перенести – пусть это не физические, а умственные усилия исполнителей, но время это занимает, как и перекладка кирпичей.

Далее в таком треугольнике проявляется конфликт интересов заказчика и исполнителя:

- Цели заказчика проекта: сделать побольше (увеличить объем работ), побыстрее (сроки сократить), подешевле (меньше ресурсов потребуется).
- Цели исполнителя: заработать максимум денег, закрыть потребности заказчика в приемлемом качестве. А это: сделать поменьше (минимизировать объем работ), подольше (увеличить срок, чтобы все успеть без авралов), подороже (увеличить бюджет).

Как мы видим, цели заказчика и исполнителя прямо противоположны, поэтому в процессе заключения договора на проект автоматизации нужно искать компромисс и договариваться. Проектный треугольник при этом должен устраивать обе стороны.

Для того чтобы понизить расходы на проект, понадобится уменьшить объем работ, убрав некоторые задачи, или снизить качество реализации задач, что позволит сделать их быстрее (явная экономия на оплате исполнителей).

Если выделить дополнительное время на работы, то можно увеличить объем работ, добавив дополнительные задачи (например, на большее количество итераций тестирования и опытной эксплуатации), увеличить тем самым общее качество результата, но это потребует оплаты привлеченных ресурсов, а значит, и увеличения бюджета проекта.

Если мы хотим сделать запланированный объем работ без сокращений, но быстрее, то это потребует привлечения дополнительных ресурсов (распараллелить работы на большее количество исполнителей), а значит, и оплаты этих специалистов, что увеличит бюджет проекта.

Как ни крути, но изменить только одну сторону треугольника не получается.

В общем случае получается четыре подхода:

- Фиксируем объем работ: меняем сроки и бюджет проекта.
- Фиксируем бюджет проекта: меняем сроки и объем работ.
- Фиксируем сроки: меняем объем работ и бюджет проекта.
- Меняем все стороны треугольника.

В гибких agile-методиках, когда оценить объем работ заранее сложно, работы ведутся небольшими временными интервалами (фиксированный интервал), за понятную стоимость (оплата участников). В этом случае в регулярную сборку входит какой-то созданный за отведенное время функционал, который сразу передается пользователям (управляем объемом работ, который можно сделать за это время, и бюджетом).

В классических «водопадных» методиках объем работ заранее утвержден, а оцениваются и управляются сроки и стоимость реализации этого объема работ.

Графически это можно представить на схеме.

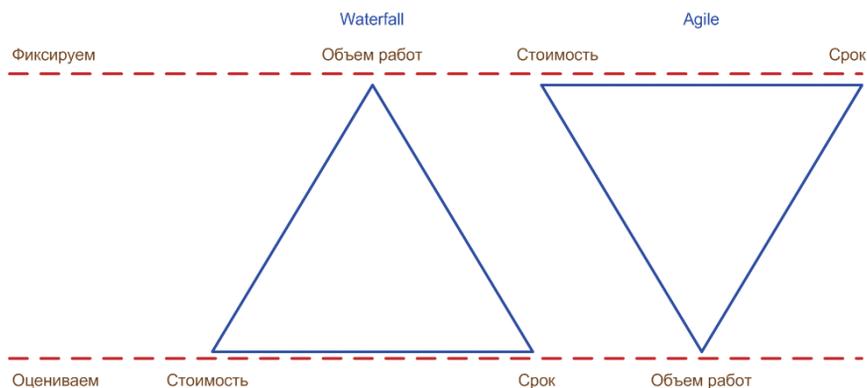


Рис. 3.5. Разница между гибким и «водопадным» подходами

Удержание проекта в границах проектного треугольника – одна из важнейших задач руководителя проекта. В ходе работ над проектом могут возникать (и обычно возникают) различные ситуации: вскрываются новые требования, без которых нельзя обойтись, затягиваются сроки, так как неточно были даны изначальные оценки, от этого раздувается бюджет, что никак не может радовать заказчика. Руководитель проекта должен своевременно выявлять такие ситуации и управлять ими: увеличивать бюджет и список требований дополнительными соглашениями к договору, следить за сроками и эффективной работой над задачами в проектной команде.

Реинжиниринг бизнес-процессов

Один из важных моментов процесса внедрения ERP-системы – это возможность взглянуть свежим взором на привычные, устоявшиеся за годы (а может, и десятилетия) бизнес-процессы.

Велик соблазн считать, что бизнес постоянен, люди, которые занимаются основной оперативной деятельностью, сработаны и в целом все в компании хорошо, кроме, пожалуй, учета, управленческой отчетности, а от этого – оперативности управленческих решений; ну, может, еще проблема с регламентированной отчетностью и ее аудитом, из-за чего иногда компания попадает на штрафы при фискальных проверках. Знакомо?

Логично предположить, что раз компания созрела для внедрения ERP-системы (так как ее не было вообще или существующая не соответствует потребностям), то на это были причины. В главе 1 в разделе «Зачем ERP-система компании» такие причины рассматривались.

Если цель – получить прозрачность в бизнесе, его учете и масштабируемости, то нужно трезво оценить текущее состояние бизнес-процессов «как есть» и выбрать состояние «как будет» с учетом возможностей новой ERP-системы. В случае отраслевой специфики предприятия имеет смысл рассматривать отраслевые решения на базе ERP-системы, где заложена отраслевая логика бизнес-процессов и адаптирована под нее функциональность.

Внедрение ERP-системы – это отличный шанс провести реинжиниринг бизнес-процессов под современное состояние рынка, отрасли, технологий, политического момента и прочего.

Реинжиниринг бизнес-процессов (англ. business process reengineering) – это фундаментальное переосмысление и, возможно, радикальное перепроектирование бизнес-процессов компании для достижения максимального эффекта производственно-хозяйственной и финансово-экономической деятельности. Результат пересмотра процессов оформляется соответствующими организационно-распорядительными и нормативными документами – изменением регламентов по предприятию.

Реинжиниринг использует специфические средства представления проблемной информации (схемы в нотации описания бизнес-процессов), понятные как менеджерам, так и разработчикам информационных систем. В процессе реинжиниринга выделяются проблемные процессы, которые можно соотнести с типовыми процессами в ERP-системе в процессе настройки прототипа, и далее определяется оптимальный вид бизнес-процессов и способы перевода текущего состояния в оптимальный вид (по используемым для этого средствам, времени, ресурсам и т.п.). И именно это оптимальное целевое состояние переносят во внедряемую систему, фиксируют в процедурах работы для пользователей. Затем измененному бизнес-процессу (как будем жить с новой системой) обучают пользователей в процессе обучения работе в системе.

Поэтому рекомендуется не держаться за вековые устои и традиции ведения бизнеса, а использовать процесс внедрения ERP-системы для их переосмысления. К сожалению, тут инициатива должна исходить со стороны команды заказчика, так как исполнитель сам может не предлагать варианты – либо отметить какие-то моменты, но не настаивать на них, а, наоборот, занять позицию «сделаем все, что скажете, за ваши же деньги». Хотя именно тут от специалистов исполнителя как профессионалов требуется оказать консалтинговые услуги: дать советы и провести консультации по замеченным со стороны проблемным местам, требующим реинжиниринга.

Поэтому рекомендация специалистам заказчика – сразу планировать возможность реинжиниринга бизнес-процессов, проявить инициативу, задавать вопросы: «А как это оптимально в ERP-системе сделать?» – и быть готовыми к изменениям.

Рекомендация исполнителям – не молчать, если видно, что что-то можно сделать иначе, лучше, чем в исходных требованиях, выявлять и доносить до заказчика такие места, предлагать варианты, исходя из возможностей ERP-системы.

Какие подсистемы в какую очередь внедрять

Еще один важный аспект темы «как внедрять ERP-систему» – это какие именно подсистемы из ERP-системы внедрять и в какой очередности.

Очевидно, хочется получить «все и сразу» и еще желательно мгновенно. Но обычно переход из текущего состояния в целевое занимает время, а компания продолжает вести свою деятельность без паузы на время проекта внедрения. Как быть?

Составив схему текущего и перспективного ИТ-ландшафта (см. главу 2, раздел «ИТ-ландшафт текущий и перспективный»), можно рассмотреть возможные варианты перехода от одного состояния к другому:

- Полная замена текущих систем на новую ERP («все и сразу»).
- Частями, через цепочку промежуточных состояний. Потребуется интеграция с другими системами, которые остаются в ИТ-ландшафте, на длительный срок или временно, на время последующих этапов проекта внедрения.

Текущие учетные системы или их аналоги, до того неавтоматизированные блоки (например, в файлах Excel) при полной замене и переходе в единую ERP-систему просто перестают использоваться, и идет переключение (одномоментно) сотрудников на работу в новую систему. До переключения на новую систему, разумеется, прошли фазы анализа, настройки, переноса начальных данных и, главное, обучения пользователей (подробнее о фазах проекта будет рассказано в следующих главах). До момента переключения все работает по-старому, после переключения – по-новому. Тут вроде все понятно. Новая система обеспечивает полный «замкнутый» цикл учета по всем основным бизнес-процессам, что планировались к автоматизации.

А что делать, когда так поступить невозможно (сроки/ресурсы/объем работ не позволяют, см. раздел «Проектный треугольник») и принимается решение о запуске частями? Как эти «части» определить?

Если посмотреть на схемы и пример ИТ-ландшафта из главы 2, то видно, что промежуточное состояние для ИТ-ландшафта – это автоматизация горячих мест по основной деятельности, а участки, которые уже имеют автоматизацию или где трудоемкость ведения учета без специального инструментария приемлемая, остаются вне границ первой очереди автоматизации. В примере на рис. 2.5 остались без автоматизации: регламентированный учет (закрит текущими учетными системами), бюджетирование и расчет бонусных частей в зарплате персонала по управленческой отчетности (остались без изменений в ручных базах Excel).

ERP-система состоит из связанных между собой блоков (см. рис. 1.1), запуск которых в работу возможен не всех сразу, а в некотором сочетании между собой. Так, в 1С:ERP настройками функциональных опций некоторые разделы (подсистемы) можно вообще скрывать или отключать функциональность частично – например, не использовать платежный календарь и заявки на расходы денежных средств в «Казначействе», а пользоваться только учетном денежных средств и интеграцией с клиент-банком.

Какие именно участки считаются «горящими» в компании – это ясно самой компании, выявляется и фиксируется после анализа и составления схемы ИТ-ландшафта. Собственно, раз речь зашла о внедрении ERP, то цели автоматизации (и зачем ERP-система компании) уже сформулированы на этот момент, и из них следуют приоритеты автоматизации по внедряемым подсистемам.

Выбирая частичную автоматизацию, нужно следовать следующим критериям:

- Подсистемы, выбранные для автоматизации, должны обеспечивать замкнутый контур учета.
- Например, чтобы продавать товары, их нужно закупать; чтобы оплачивать закупки, деньги должны поступать на расчетные счета; чтобы получать регламентированную отчетность и вести баланс – должны отражаться все учетные операции.

- Если какие-то блоки не автоматизируются, но результат их работы важен для целостности учета (товарные и денежные потоки, формирование бухгалтерских проводок), то необходимо предусмотреть интеграцию с существующими системами, где это все учитывается на приемлемом уровне.
- Если таких существующих систем нет или они не поддерживают интеграцию, нужно признать, что эти места требуют автоматизации и не могут быть вынесены на вторую очередь.
- Для прочих, не массовых, операций можно ограничиться документами ввода финансовой и бухгалтерской учетной информации, достаточной для целостного учета без полноценной автоматизации этих бизнес-процессов в оперативном контуре системы.
- За бортом автоматизации первой очереди можно оставить блоки, которых вообще никогда не было в бизнес-процессах компании.
 - Например, подсистему «Бюджетирование» хочется начать использовать, но до внедрения ERP в компании не было процессов бюджетирования, значит, этот блок можно отложить на вторую очередь. Если, конечно, это не основная цель автоматизации.

Существует еще вариант развития компании и получения быстрого эффекта от внедрения новых подсистем ERP, которых не было до этого в текущей учетной системе, но которые уже нужны сегодня (как цель автоматизации), – это внедрение только этих новых подсистем, сохраняя остальные блоки без изменений. Это подвид частичной автоматизации, но тут основная часть ИТ-ландшафта остается без изменений, добавляются новые блоки без замены существующих.

Предположим, что в компании есть учетная система, решающая задачи оперативного и регламентированного учета на должном уровне. И цели заменить эту систему на другую непосредственно ради этих блоков нет. Все хорошо, все вопросы и бизнес-процессы закрыты.

Но бизнес развивается, и ему требуются, например, два новых блока: отчетность по международным стандартам и бюджетирование.

Текущая учетная система не имеет этих блоков, либо внедрение их неперспективно (блоки имеют устаревший функционал и уже не поддерживаются). В целом, в перспективе, компания хотела бы перевести все свои процессы в 1С:ERP, получив единое информационное пространство и масштабируемое решение, но сегодня нужны только пара новых блоков и проект внедрения по ним, без потрясений для всего остального бизнеса.

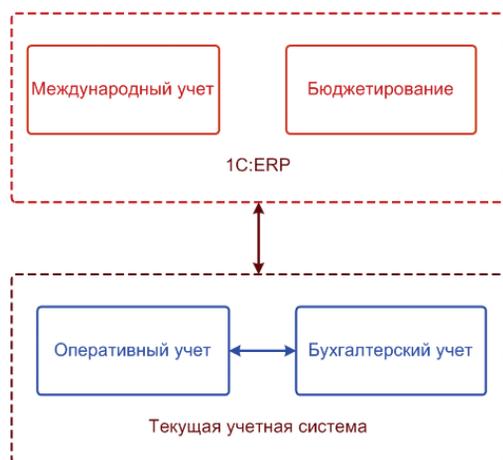


Рис. 3.6. Подсистемы из новой ERP-системы дополнительно к текущей учетной системе

В этом случае:

- оставляем текущую учетную систему,
- рядом ставим 1С:ERP,
- настраиваем одно- или двустороннюю интеграцию систем,
- внедряем новые подсистемы из 1С:ERP в компании,
- получаем ожидаемый эффект от нововведений.

Этот подход требует анализа возможностей текущей системы. В ней может быть недостаточно данных либо какой-то аналитики для выгрузки в ERP-систему. Тогда надо продумать, как и куда доносить недостающую информацию. Плюс нужно предупреждать заказчика, что качество учета в ERP-системе будет напрямую зависеть от качества данных в текущей учетной системе. Нужно быть осторожнее с формулировками критериев успешности проекта, иначе для завершения проекта придется налаживать (дорабатывать) учет, в том числе в текущей системе.

Важно отметить, что не любые подсистемы ERP так можно запускать: где-то связи настолько тесные, что настройка обмена данными с другими системами по трудоемкости сопоставима с переводом работы в новую систему целиком. Но какие-то блоки (как, например, МСФО и бюджетирование) вполне автономны: получение фактических данных в бюджетировании и данных для трансляции проводок в МСФО можно организовать передачей бухгалтерских проводок из текущей учетной системы в новую систему.

Для выстраивания очередности внедрения подсистем рассмотрим вопрос проектирования НСИ. Необходимо уделять должное внимание на внедрении выбору и настройкам нормативно-справочной информации, особенно при частичной автоматизации. Например, на предприятии в первой очереди автоматизации стоит бухгалтерский учет, на потом оставлены оперативный контур и автоматизация производства. В этом случае проектирование НСИ нужно проводить в целом, а не только для задач бухгалтерского учета.

Спроектированная только для целей бухгалтерского учета номенклатура не будет учитывать требования для учета производства, и, когда дойдет до автоматизации производства, придется перепроектировать НСИ, а исторические данные могут оказаться несовместимы, и их нужно будет конвертировать. Либо придется запускать систему переносом остатков с нового периода, а это, по сути, уже перевнедрение – лишнее потрясение для компании и ее сотрудников.

Правильным подходом является выделение проектирования общей НСИ в отдельную задачу на фазе дизайна архитектуры системы. Проектирование НСИ должно выйти за рамки первой внедряемой подсистемы и учесть перспективные требования других, стоящих в очереди автоматизации, подсистем. Причем самые требовательные к детализации НСИ подсистемы (такие как производство) должны иметь сформулиро-

ванные требования, а для этого придется рассматривать и анализировать комплексную автоматизацию и перспективный ИТ-ландшафт из всех блоков автоматизации, хотя исходно ожидается на первом этапе только одна бухгалтерская подсистема.

Еще одним важным аспектом выстраивания очередности внедрения для производственного контура является выбор уровня автоматизации: производственный учет или управление производством. И то и другое декларируется как автоматизация контура производства на предприятии. Но задача именно управления производством достаточно сложная, тогда как организация производственного учета проще. В системе 1С:ERP для этого есть отдельные инструменты. Система позволяет вести производственный учет при помощи инструментов управления производством. Популярная ошибка на внедрениях – это, реально не запуская управление производством, использовать сложные инструменты управления только для целей учета. Все данные в систему еще не вводятся, т.к. контур управления полноценно не запущен, от этого будет страдать учет.

В заключение раздела рассмотрим на схемах несколько различных вариантов очередности запуска подсистем в эксплуатацию в процессе проекта внедрения ERP-системы.

Вариант 1. Запуск от оперативного контура.

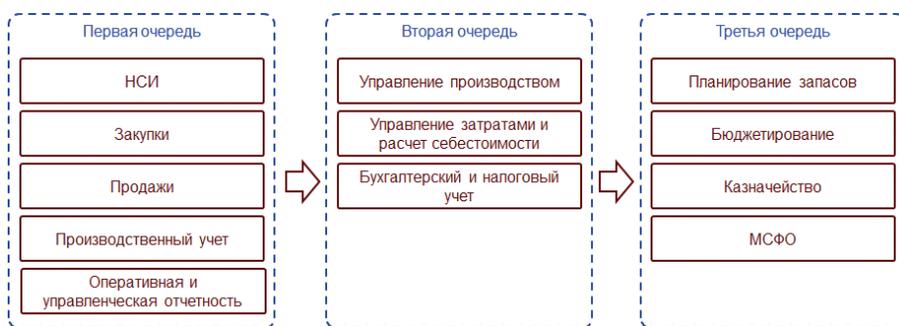


Рис. 3.7. Запуск от оперативного контура

В первую очередь автоматизации подлежат требования оперативного учета и управленческой отчетности, предполагается, что текущий бухгалтерский и налоговый учет покрыт отдельной используемой для этого учетной системой. Запуск в эксплуатацию первой очереди даст предприятию ожидаемые бизнес-преимущества подключения деятельности менеджеров и прочих сотрудников оперативного контура в ERP-систему. Бухгалтерия не будет затрагиваться и переживет этот этап проекта без потрясений. На втором этапе будет переход к управлению производством и полноценному учету в системе с отказом от внешних бухгалтерских систем. Третий этап будет связан с развитием финансового и товарного планирования, управления денежными потоками и управленческой отчетностью по стандартам МСФО.

Вариант 2. Запуск от бухгалтерского учета.

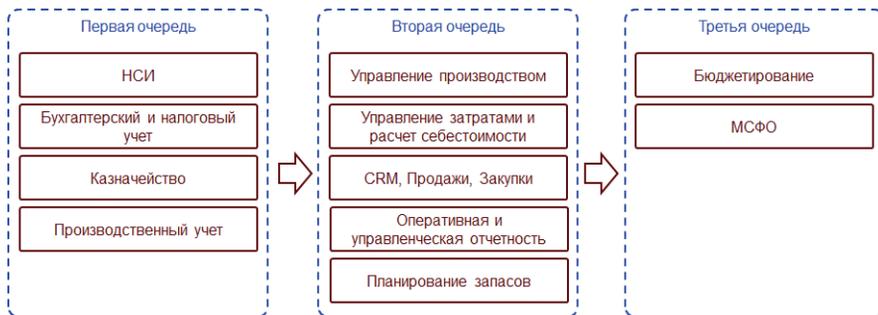


Рис. 3.8. Запуск от бухгалтерского учета

В этом сценарии первым внедряется регламентированный учет, нет иных внешних систем для бухгалтерского учета, учет сразу выставляется в ERP-системе. К первой же очереди относится необходимый для составления регламентированный отчетности минимум производственного учета, и задействуются возможности подсистемы казначейства для управления денежными потоками через заявки на расходование денежных средств и формирование платежного календаря. Вовлечение в бизнес-процесс формирования и согласования заявок казначейства затронет всех сотрудников затратных подразделений компании, повысит соблюдение платежной дисциплины. Следующим этапом будет наращивание оперативного контура и управленческой отчетности. Третьей очередью пойдут бюджетирование и отчетность по международным стандартам.

Вариант 3. Запуск от финансового планирования и МСФО.

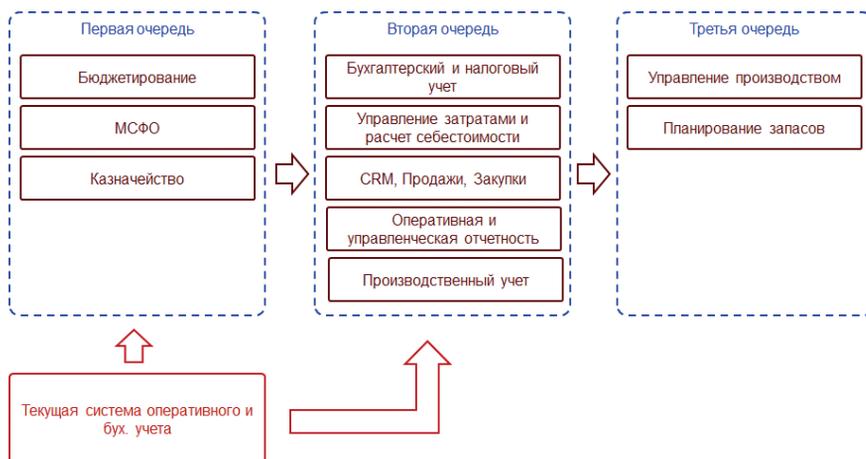


Рис. 3.9. Запуск от финансового планирования и МСФО

Этот подход к внедрению уже рассматривался выше. У компании уже есть текущая система автоматизации, оперативный контур и бухгалтерский учет покрыты в ней на должном уровне, но компании нужно перейти к финансовому планированию, прогнозированию финансового состояния, управлению денежными потоками и получать отчетность по международным стандартам. Внедрение начинается с этих блоков, которые не были охвачены в текущей системе. Результат после первого этапа принесет компании существенные бизнес-преимущества от внедрения новых подсистем, которых не было ранее в существующей системе автоматизации, что даст результаты от их работы для бизнеса. На втором этапе проекта будет отказ от текущей системы автоматизации с переходом в единую систему. На третьем этапе будет развитие – переход к полноценному управлению производством и планированию запасов, чего до этого не было в текущей системе.

Как мы видим из приведенных примеров, сочетание различных подсистем и их очередности (этапов может быть не обязательно три, но меньше или больше) зависит от разных факторов: что уже есть у компании, какие насущные потребности нужно закрыть, выстраивается ли логическая последовательность из подсистем, дающая целостный учет в переходном периоде.

Глава 4. Кто участвует в проекте внедрения

Организационная структура проекта

Проект внедрения ERP-системы осуществляют конкретные люди, и именно от их коммуникации, сработанности и квалификации зависит успех или неуспех проекта.

Рассмотрим всех участников проекта подробнее.

Для начала рассмотрим стороны, которые могут участвовать в проекте:

- заказчик;
- исполнитель;
- поставщик (разработчик) ERP-системы;
- спонсор проекта (может быть отдельной организацией относительно заказчика или внутри заказчика).

Основные участники, конечно, это заказчик и исполнитель. Но в общем случае все 4 стороны могут быть задействованы, особенно на ранних стадиях проекта.

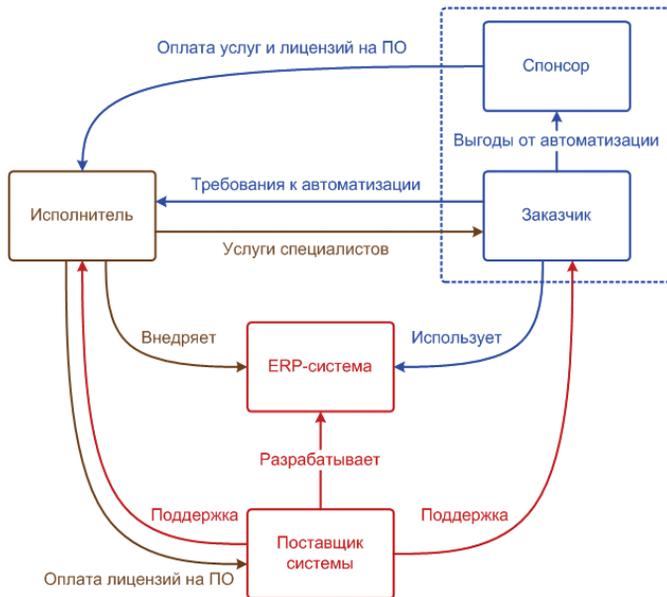


Рис. 4.1. Стороны – участники проекта

От поставщика системы исполнитель (или заказчик) может попросить помощи в части экспертной поддержки оценок производительности или определенных функциональных требований, которые критичны для бизнеса заказчика, но еще не поддерживаются в системе или поддерживаются иначе, чем требуется. Возможно, что у поставщика в планах развития ERP-системы уже содержится такая функциональность и она появится к моменту перехода к промышленной эксплуатации, либо он даст рекомендации по использованию отраслевых решений под специфику бизнеса заказчика, либо такая специфика должна рассматриваться в проекте через призму реинжиниринга бизнес-процессов и дорабатываться по месту исполнителем проекта внедрения.

Спонсор проекта может быть как самим заказчиком, так и отдельной организацией. Условно это то лицо или группа лиц, которые за все платят, но хотят получить от проекта максимум пользы, то есть сделать работу бизнес-процессов компании заказчика автоматизации оптимальнее с целью увеличить прибыль и свою (как заинтересованной стороны) выгоду от этого. Прямой экономической заинтересованности у спонсора может и не быть – в случае, например, целевого финансирования из государственного бюджета цели проекта могут быть не только экономические, но и социальные или иные. Система, например,

нужна для организации самих бизнес-процессов, т. к. без автоматизации осуществление деятельности заказчика просто невозможно.

Спонсор может быть внутри компании, когда, например, заказчиком системы автоматизации выступает финансовый директор, а спонсором является директор – владелец бизнеса. Альтернативная ситуация: заказчик – это директор компании, но у бизнеса есть внешние учредители (возможно, организации, а не только физические лица), которые будут спонсорами проекта.

Заказчик обеспечивает формулировку целей и требований автоматизации, осуществляет приемку результатов проекта. Специалисты заказчика могут выполнять отдельные работы, находящиеся в компетенции заказчика, вследствие:

- исключительного доступа к данным/регламентам/процедурам принятия решений;
- решения о самостоятельном выполнении части работ по модификации и настройке системы;
- решения о самостоятельном обучении существующих и новых пользователей компетентными сотрудниками заказчика, прошедшими обучение у исполнителя.

Исполнитель обеспечивает выполнение работ и формирование результатов проекта в соответствии с требованиями, установленными заказчиком. Исполнитель может быть генеральным подрядчиком, а в проекте могут участвовать несколько компаний-внедренцев на подряде под управлением генерального подрядчика или как независимые исполнители на разные блоки работ под управлением руководителя проекта со стороны заказчика (такое практикуется, если параллельно идет внедрение разных систем, которые либо интегрируются, либо нет между собой).

Организационная структура типового проекта внедрения состоит из следующих лиц/групп лиц:

- Управляющий комитет:
 - куратор проекта со стороны заказчика;

- руководитель проекта со стороны заказчика;
 - куратор проекта со стороны исполнителя;
 - руководитель проекта со стороны исполнителя.
- Рабочая группа со стороны заказчика:
 - центр компетенции;
 - ключевые сотрудники заказчика.
 - Проектная команда со стороны исполнителя:
 - специалисты исполнителя.

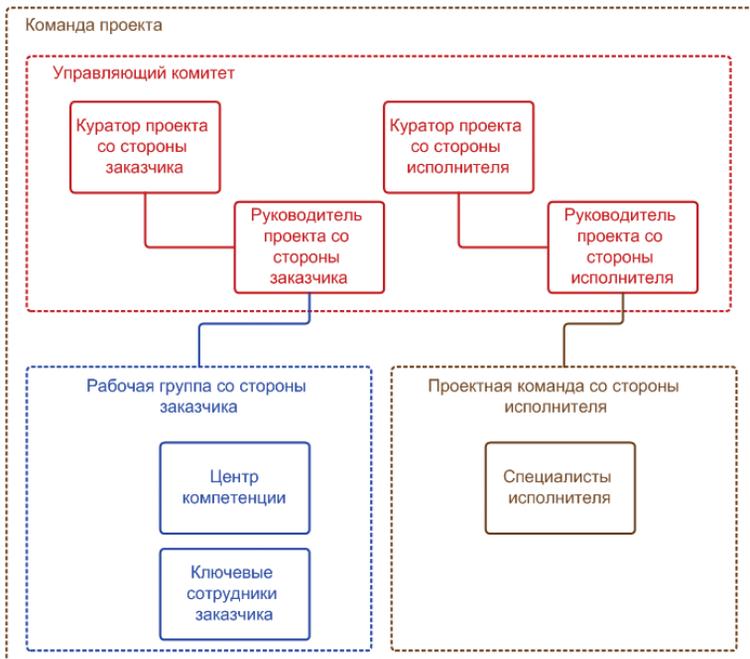


Рис. 4.2. Организационная структура проекта

Управляющий комитет

Управляющий комитет проекта создается для решения организационных и технических вопросов, управления ходом работ по проекту в целом, принятия решений по составу и объему работ, оценки качества полученных результатов и утверждения отчетных документов. На управляющий комитет также выносятся вопросы, решение которых не было найдено на уровне руководителей проекта.

Функциональные обязанности:

- утверждает устав проекта и согласовывает изменения в нем;
- согласовывает бюджет дополнительных работ и изменение плана проекта;
- определяет ожидания и показатели успеха проекта;
- принимает стратегические решения по проекту;
- контролирует статус проекта.

Куратор проекта – лицо, заинтересованное в успехе проекта в целом и имеющее полномочия для решения ресурсных и других проблем, эскалированных руководителем проекта.

Куратор проекта со стороны заказчика входит в состав руководства организации-заказчика (обычно сам директор) и осуществляет:

- общий контроль исполнения договора, соблюдения условий, целей, задач, объемов и границ проекта;
- назначение руководителя проекта со стороны заказчика;
- принятие окончательных решений со стороны заказчика.

Куратор проекта со стороны исполнителя назначается руководством организации-исполнителя (это может быть сам директор или руководитель направления) и осуществляет:

- общий контроль исполнения договора, соблюдения условий, целей, задач, объемов и границ проекта;
- назначение руководителя проекта со стороны исполнителя;
- принятие окончательных решений со стороны исполнителя.

Руководитель проекта (РП, РМ – project manager) – менеджер, ответственный за организацию и успешную реализацию проекта.

Руководитель проекта со стороны заказчика назначается куратором проекта со стороны заказчика и отвечает:

- за взаимодействие с исполнителем;
- управление проектом в части, выполняемой рабочей группой со стороны заказчика;
- участвует в подготовке отчетности по проекту;
- оперативный контроль исполнения условий договора, соблюдения условий, целей, задач, объемов и границ проекта;
- участие в оперативном планировании проектных задач, выработке решений о внесении изменений или дополнений в планы работ;
- организацию и контроль сбора информации, необходимой исполнителю для реализации проекта;
- оперативное планирование работ сотрудников, задействованных со стороны заказчика;
- контроль исполнения поставленных задач сотрудниками заказчика, задействованными на проекте;

- обеспечение взаимодействия и коммуникации сотрудников заказчика, задействованных на проекте, и специалистов исполнителя;
- обеспечение материально-технической базы для выполнения сотрудниками заказчика их задач по проекту и для проведения работ специалистами исполнителя по месту нахождения заказчика;
- обеспечение устойчивой телекоммуникационной связи с исполнителем со своей стороны для выполнения задач, согласованно решаемых в заочном режиме, но требующих взаимодействия в режиме онлайн;
- организацию обучения персонала заказчика;
- контроль и утверждение результатов проекта в рамках существующих планов, предоставление в документальной форме дополнений и замечаний.

Как мы видим, список дел у руководителя проекта со стороны заказчика довольно значительный, поэтому будет ошибочно считать, что такой выделенный руководитель не нужен или эти задачи может решить руководитель со стороны исполнителя.

Руководитель проекта со стороны исполнителя назначается куратором проекта со стороны исполнителя и отвечает:

- за управление проектом (сроки, бюджет, объем работ);
- взаимодействие с заказчиком, субподрядчиками и подразделениями исполнителя. Управление проектом в части, выполняемой рабочей группой со стороны заказчика, осуществляется через взаимодействие с руководителем проекта со стороны заказчика;
- подготовку и предоставление отчетности по проекту (план-график, статус-отчеты, журнал изменений);
- формирование проектной команды;

- управление проектной командой;
- управление проектными рисками и их минимизация;
- общее и оперативное планирование и контроль выполняемых задач на всем протяжении проекта;
- формирование запросов заказчику о предоставлении информации, необходимой для реализации проекта;
- оперативный контроль исполнения условий договора, соблюдения условий, целей, задач, объемов и границ проекта;
- оценка необходимости дополнительных работ, выработка предложений о внесении изменений или дополнений в планы работ;
- подбор субисполнителей (при необходимости, совместно с куратором) и управление ими;
- обеспечение устойчивой телекоммуникационной связи с заказчиком со своей стороны;
- организация проведения обучения персонала заказчика;
- определение критериев оценки заказчиком готовности системы к промышленной эксплуатации;
- согласование с заказчиком результатов проекта.

Со стороны исполнителя у руководителя проекта еще больше задач, поэтому важно понимать: при всем желании поручить роль руководителя проекта ведущему разработчику или старшему консультанту в дополнительную нагрузку нужно четко отдавать себе отчет, что этот специалист должен обладать соответствующими навыками коммуникации и лидерства и будет занят некоторыми бюрократическими (в хорошем смысле этого слова) задачами по поддержанию документарного оформления проекта и управленческих решений в нем. И на основную работу (проектирование, разработка, консультирование) времени может не остаться, либо задачи управления пойдут по остаточному принципу. Тут все зависит от объема проекта и команды участников в нем.

Если проект небольшой и сводная команда проекта (исполнителя и заказчика) до 10 специалистов, то совмещение еще как-то возможно, если же проект значительный, то внимания он потребует 100 %.

Проектная команда исполнителя

Проектная команда исполнителя состоит из специалистов, занимающихся внедрениями ERP-систем. В нее входят:

- методисты (консультанты и аналитики),
- системные архитекторы,
- разработчики,
- тестировщики,
- технические писатели.

Обычно роли методистов, тестировщиков и технических писателей совмещают одни и те же специалисты с общим названием «консультанты». От слова «консультировать», поэтому и «консалтинговые услуги», и «консалтинг» (англ. consulting) как вид деятельности.

Системный архитектор формирует общую архитектуру будущей системы и принимает решение о способах реализации требований к ней.

Функциональные обязанности:

- анализ функциональных требований к системе и выявление противоречий;
- принятие решений о выборе способа реализации функциональных требований в системе;
- разработка концепции проектного решения, подготовка технического проекта;
- участие в разработке плана проекта и оценках бюджета;

- проработка границ прототипа системы;
- формирование спецификаций по возникающей в ходе проекта дополнительной разработке;
- поддержка и контроль качества работы разработчиков и консультантов.

Консультант выполняет основные работы в ходе проекта по коммуникации со специалистами заказчика, настройке прототипа учета в системе, подготовке технических заданий и инструкций, тестированию и обучению пользователей.

Функциональные обязанности:

- изучает требования заказчика к системе автоматизации, помогает их сформулировать и уточнить, если изначально требования не формализованы;
- определение, документирование и анализ бизнес-процессов заказчика;
- сбор требований к системе, анализ требований и их непротиворечивости;
- описание технических заданий (ТЗ) на модификации стандартной функциональности системы;
- согласование и утверждение технических заданий с ключевыми сотрудниками заказчика;
- настройка прототипа системы;
- тестирование и приемка модификаций от разработчиков;
- настройка системы в соответствии с требованиями учета заказчика;
- настройка ролей и прав доступа пользователей к системе;

- поддержка интеграционного тестирования системы с участием членов рабочей группы со стороны заказчика;
- подготовка документации для пользователей и администраторов системы;
- презентации функциональности системы для наглядного показа принимаемых проектных решений и достигнутых результатов;
- обучение членов рабочей группы со стороны заказчика и ключевых пользователей системы;
- аттестация сотрудников, прошедших обучение, на доступ к работе в системе;
- планирование и обеспечение переноса исторических данных из унаследованных систем в систему;
- начальная поддержка пользователей после запуска системы в промышленную эксплуатацию.

Разработчик проводит работы, необходимые для модификации стандартной функциональности системы по утвержденным техническим заданиям проекта.

Функциональные обязанности:

- проведение работ по программированию в системе;
- первичное тестирование и отладка модифицированных модулей;
- исправление ошибок в разработанном функционале системы;
- исправление критических ошибок штатной функциональности, по которым нет приемлемого способа обхода и срок исправления поставщиком системы не определен.

Внутри команды исполнителя у консультантов и разработчиков может быть дополнительная градация в зависимости от квалификации

специалиста, например: стажер, специалист, ведущий специалист. Или возможна иная система позиционирования должностей (система грейдов, англ. grading) и ставок от этого. Главное – это поставить на задачи проекта специалистов с достаточной квалификацией для их решения.

Бывают ситуации, когда несколько ролей сосредоточены в одном человеке. Например:

- руководитель проектов + системный архитектор;
- руководитель проектов + консультант;
- системный архитектор + разработчик;
- системный архитектор + консультант;
- консультант + разработчик.

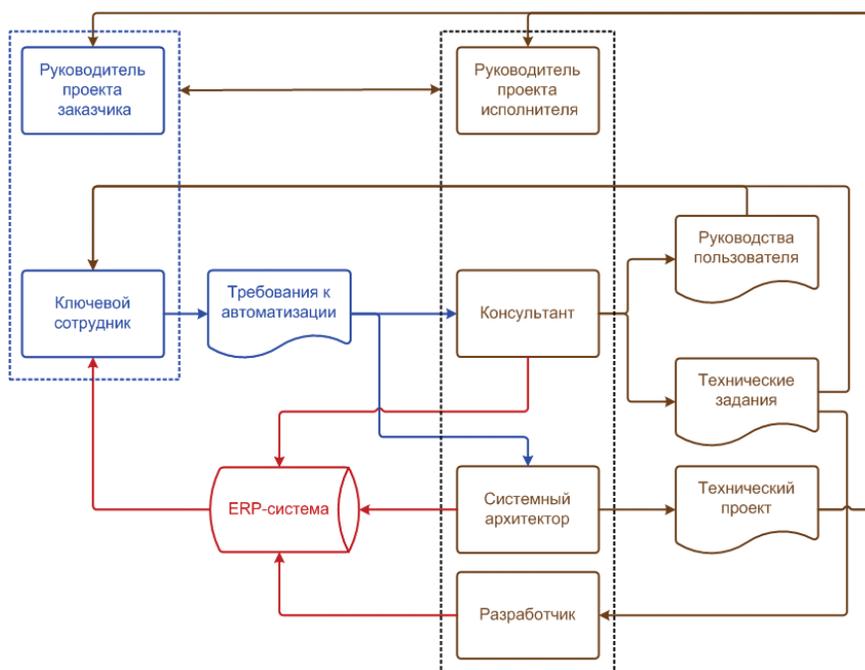


Рис. 4.3. Схема взаимодействия команды проекта

В частных случаях и в небольших проектах (реже бывает, что и в больших) это может быть небольшая группа таких «универсальных солдат», которые творят чудеса при внедрении системы. Но на практике для проектов внедрения систем класса ERP есть опасность возникновения ситуации, когда один человек – это и консультант, и разработчик: сам себе написал ТЗ, сам по нему все сделал, сам все протестировал (в процессе отладки, как разработчик и не более). Все вроде хорошо? Не совсем: ТЗ нужно формально описать и согласовать с ключевыми сотрудниками заказчика, как бы ни хотелось оптимизировать работу и самому себе ТЗ не делать. Протестировать модификацию необходимо. Руководство пользователя написать тоже нужно, обучение пользователей провести. А есть еще задачи коммуникации со сложными в общении пользователями и навыки формулирования понятных непрофессионалам текстовых описаний. И вот это часто совместить в одном человеке сложно: консультант ориентирован на коммуникацию с пользователями заказчика, разработчик – на стройную логику в коде. Но если у вас в команде такие универсальные люди есть (а они бывают, автор таких встречал), то их компетенциями необходимо умело управлять, соблюдая проектные методологии!

Рабочая группа со стороны заказчика

Рабочая группа со стороны заказчика создается из числа ключевых сотрудников заказчика или других подчиненных структур для работы над проектом. Участники рабочей группы подчиняются руководителю проекта со стороны заказчика.

В состав рабочей группы входят ключевые специалисты различных областей, осуществляющие постановку задач по разделу, составляющему сферу их профессиональной ответственности, а также оказывающие содействие во время выполнения работ по проекту, относящихся к их подразделению или сфере деятельности.

Функциональные обязанности:

- постановка задач по разделам учета;
- экспертиза и согласование проектных решений;

- оказание содействия в написании функциональных спецификаций по разделам учета;
- рассмотрение и утверждение документов по требованиям к системе и ее настройке по разделам учета;
- оказание содействия при тестировании и опытной эксплуатации системы по разделам учета;
- формирование критериев оценки готовности системы к промышленной эксплуатации;
- рассмотрение и подтверждение материалов для проведения обучения;
- обучение конечных пользователей (тиражирование знаний) и обеспечение их соответствующими инструкциями и информацией, необходимыми для эффективного ввода в эксплуатацию и использования результатов проекта;
- подготовка данных для ввода начальных остатков в систему, при необходимости – ввод или корректировка автоматически загруженных исходных данных.

Внутри рабочей группы начинает формироваться центр компетенции по системе.

Центр компетенции представляет собой подразделение в структуре заказчика (может не совпадать с подразделениями согласно организационной структуре заказчика) и создается для участия в проектных работах по внедрению системы, обеспечения поддержки и развития системы после запуска в промышленную эксплуатацию.

В это подразделение входят специалисты заказчика с квалификацией уровня консультантов и разработчиков по системе автоматизации, то есть способных фактически на проведение самостоятельного внедрения (или части работ) и расширенную поддержку после запуска. Впрочем, таких сотрудников может быть мало, либо изначально у них нет экспертных знаний по системе, и потому основную работу делают привлеченные специалисты исполнителя, а сотрудники центра компетенции перени-

мают знания и растят компетенции внутри компании заказчика в ходе проекта внедрения.

Центр компетенции заказчика является важной частью успешного внедрения и сопровождения системы на всем ее жизненном пути, особенно по окончании проекта внедрения. Процесс создания центра компетенции подробно рассматривает рекомендованная для корпоративных проектов «1С:Технология корпоративного сопровождения» (<https://consulting.1c.ru/technologies/tks>).

Функциональные обязанности:

- участие в проектных работах (совместно с проектной командой со стороны исполнителя), в том числе:
 - проектирование и разработка отдельных модификаций функциональности системы;
 - участие в интеграционном тестировании;
 - подготовка отдельных разделов документации для пользователей и администраторов системы;
 - подготовка технической инфраструктуры для разворачивания системы;
 - подготовка к переносу и участие в переносе исторических данных из унаследованных систем;
 - участие в обучении пользователей;
 - участие в настройке ролей и прав доступа пользователей к системе;
 - участие в начальной поддержке пользователей в ходе опытной и промышленной эксплуатации, сбор сообщений

об ошибках в системе, обеспечение исправления ошибок (собственными силами или силами исполнителя);

- поддержка промышленной эксплуатации системы, в том числе:
 - обучение новых пользователей;
 - тиражирование работы с системой на филиалы (для заказчика с филиальной структурой);
 - настройка ролей и прав доступа новых пользователей;
 - администрирование технической инфраструктуры;
 - поддержка работы пользователей;
 - сбор и систематизация замечаний пользователей и сообщений об ошибках в системе, обеспечение исправления ошибок;
- развитие системы в ходе промышленной эксплуатации:
 - планирование доработок в соответствии с замечаниями и пожеланиями пользователей и руководства заказчика, изменениями бизнес-процессов заказчика, нормативных актов и т. п.;
 - участие в доработке системы, пользовательской документации, дополнительном обучении пользователей и т. п.

Центр компетенции должен быть создан после подписания договора и начала работ по проекту. Центр компетенции является частью рабочей группы со стороны заказчика, но он не распускается по завершении проекта, а остается поддерживать работоспособность системы и пользователей на местах.

Руководителем центра компетенции является руководитель проекта со стороны заказчика. Руководитель проекта со стороны заказчика организует деятельность центра компетенции, в том числе:

- ведет подбор специалистов в центр (из числа персонала заказчика на момент начала проекта или принятых в штат в ходе проекта);
- обеспечивает обучение специалистов центра (при необходимости);
- распределяет ответственность и работы между специалистами центра.

Степень участия специалистов центра компетенции в проектных работах определяется в оперативном режиме по согласованию между руководителями проекта со стороны заказчика и исполнителя. При этом:

- Руководитель проекта со стороны исполнителя определяет задачи, передаваемые на выполнение специалистам центра, и критерии их выполнения.
- Руководитель проекта со стороны заказчика распределяет ответственность и работы между специалистами центра и контролирует их выполнение.
- Руководитель проекта со стороны исполнителя организует при необходимости консультации для специалистов центра по вопросам выполнения проектных работ (например, по принципам работы системы или проектирования и разработки модификаций функциональности).
- Руководитель проекта со стороны исполнителя обеспечивает контроль качества выполнения проектных работ специалистами центра и несет ответственность за использование результатов указанных работ в проекте.

Мотивация участников проекта внедрения

Зачем это все руководителю проекта

Как мы интуитивно знаем и неоднократно слышали, «все люди разные» – и это не просто слова, люди разные по своим поведенческим моделям, своей мотивации и подходу к делу. К каждому нужен свой подход, и успешность команды, зависящая от сработанности и коммуникации, определяется во многом сочетанием психотипов участников. Каждый член команды будет эффективен в решении задач, которые ему комфортны и интересны. Понятно, что на практике у нас есть ограниченный выбор состава участников и работать нужно с теми, кто есть – как со стороны заказчика (тут со стороны исполнителя влияния нет вообще, зато есть со стороны заказчика), так и со стороны исполнителя (задача исполнителя – вырастить и сберечь сработанную команду, заказчика – выбрать правильную команду исполнителя).

Руководителю проекта для успешной работы пригодятся познания из психологии и теории мотивационных систем, потому что в проекте работают конкретные люди. Команду можно (и нужно) формировать заранее, проводить интервью сотрудников. Нужен индивидуальный подход к каждому, стимулирование мотивации, сработанность внутри команды и с ключевыми сотрудниками заказчика.

Управление коммуникацией – это сглаживание конфликтов и мотивация на ожидаемый результат (убедить специалиста решать неинтересные ему задачи в том числе).

Хироши Танака (профессор, аналитик в области глобального проектного менеджмента, соавтор РМВОК) предложил формулу успеха для руководителя проектов (формула трех «Н»): Head + Heart + Hands (голова + сердце + руки).

PMI, в свою очередь, вводит треугольник талантов руководителя проектов, который описывает три ключевые группы навыков для РП:

- **Техническое управление проектами.** Знания, навыки и типы поведения, относящиеся к конкретным областям управления

проектом, программой и портфелем. Технические аспекты исполнения порученной роли.

- **Лидерство.** Знания, навыки и типы поведения, необходимые для управления, мотивации и руководства командой с целью помочь организации в достижении ее бизнес-целей.
- **Стратегическое управление и управление бизнесом.** Знания, профессиональная квалификация и опыт работы в отрасли и организации, которые улучшают исполнение и дают более высокие бизнес-результаты.



Рис. 4.5. Треугольник талантов PMI

У многих специалистов заказчика (и даже самого внедренца) бытует мнение, что РП «ничего не делает»: он же не проектирует, не пишет код, даже не тестирует. Зачем он? Да еще по повышенным ставкам в час, как какой-то архитектор-гуру для ERP-системы. Как следствие: «Давайте поручим роль РП ведущему разработчику или старшему консультанту – они ребята проверенные, справятся».

Это сразу сработанный риск, если не повезло в конкретных чудо-людях, так называемых «человеко-пароходах», специалистах, способных сочетать в себе разные роли и на все и всех находить время.

Нужно понимать, что оценки трудозатрат и сроков, контроль выполнения работ, вся бюрократия по проекту, коммуникация и решение конфликтов – это отдельный вид работ. Хорошему РП, как и хорошему системному администратору (отвечающему за бесперебойную работу сети и оборудования), действительно может показаться со стороны,

что ему нечего делать. Все управление проектом (а у сисадмина – работа оборудования) делается как бы само, не видны прилагаемые усилия, сотрудник не зашивается, потом и кровью не истекает – но это же его заслуга, а не вина. И стоит это дорого! Это как раз пример отличного специалиста.

Модели мотивации

Помимо личностных разнообразий участников команды проекта, у любого работника существуют мотивы, побуждающие его делать работу в должном качестве и в разумные сроки (или, наоборот, не побуждают, всякое бывает). Рассмотрим кратко некоторые теории систем мотивации как точку введения в эту тематику для дальнейшего углубленного изучения самостоятельно по профильным методическим пособиям, а также для структуризации личного опыта (если он уже есть, конечно), хотя бы относительно себя самого в разных ролях в процессе работы (пусть и в разных командах).

Согласно рекомендациям РМВОК в части мотивации персонала: *Руководитель проекта также должен учитывать т. н. «синдром студента» (или прокрастинацию), когда человек начинает работать с полной отдачей лишь в самый последний момент при приближении крайнего срока, а также закон Паркинсона, согласно которому объем работы расширяется так, чтобы заполнить все отведенное на ее выполнение время.*

Нам же нужно, чтобы работа не занимала весь выделенный на нее объем времени и еще чуть-чуть, а эффективная работа была не в последнюю неделю перед сдачей проекта ночами и без выходных, а в рабочем ритме на протяжении всего проекта с первого его дня. Эти важные моменты нужно будет учитывать при планировании сроков (рассмотрим оценку сроков детально дальше, в главе 7).

Мотивация – это причины (мотивы) действий людей. Стремление является *внутренним* и проявляется только тогда, когда оно до конца понято субъектом. В основе мотивации лежит определенная потребность (физиологическая, духовная, финансовая), после удовлетворения которой импульс к действию существенно снижается.

Мотивация команд состоит в создании возможностей для участия в принятии решений и поощрении их самостоятельной работы.

Стимулирование – это мера *внешнего* воздействия (поддержки), благодаря которой осуществляется увеличение активности исполнителя (увеличение его мотивации). Стимулирование может быть как позитивным (вознаграждение – «пряник»), так и негативным (санкции – «кнул»). Стимулирование может быть краткосрочным, а эффект от него длительным. Например, пообещать всем дополнительные бонусы в конце проекта при соблюдении сроков – и несколько месяцев/лет все будут замотивированы.

Это интересно: «стимул» в Древнем Риме – это тонкая длинная острая палка для понукания волов в телеге, чтобы они шли быстрее (стимулирование = «тыкать острой палкой»). Поэтому «стимулирование экономики» или «стимулирование сотрудников» в оригинале – это потыкать их острой палкой, чтобы расшевелить.

Мотивация сотрудников зависит от чувства, что их ценят в организации. Вариант с запугиванием (страх – сильный стимул) тоже возможен, но его не рассматриваем как непродуктивный в длительной перспективе и конкретной проектной деятельности внедрения системы ERP, тем более когда есть возможность уйти из этого проекта в соседний, пусть и в другой компании. Ценность специалиста для организации можно, очевидно, продемонстрировать через вознаграждение. Как правило, деньги рассматриваются как материальный аспект любой системы вознаграждения, но и нематериальные награды могут быть столь же или даже более результативными. Большинство членов команды проекта мотивируются благоприятной возможностью развиваться, совершенствоваться, получить высокую оценку их вклада, применять свои профессиональные навыки для решения новых сложных и, главное, интересных задач. Высококвалифицированным специалистам в области внедрения ERP должно нравиться то, что они делают (и это обычно так, иначе бы они не стали сильными специалистами).

Поэтому важно признавать заслуги команды на всем протяжении жизненного цикла проекта, а не только после его завершения. Достижение контрольных точек проекта, соблюдение сроков по ним, положительная (а иногда и отрицательная) обратная связь от пользователей – также являются хорошим мотивирующим стимулом работать дальше (успеть больше, нагнать отставание, сделать лучше).

Уместно вспомнить старую притчу о трех каменщиках, которые строили здание. Когда спросили первого: «Что ты делаешь?» – он ответил: «Я выкладываю камни в ряд». Второй ответил: «Я строю стену, слежу, чтобы она была ровной». А третий ответил: «Я строю храм!»

Как мы видим, у этих людей совершенно разная мотивация в работе и результат в скорости и качестве работы, и сам подход к делу тоже будет разным.

Когда что-то получается (пусть даже у коллег) – это всегда позитивный момент (радость), стимул делать дальше, больше. Когда же есть затыки и нет понимания, как из них выйти, то нужна внешняя поддержка – организация мозгового штурма, просто выслушать эмоции, как и что не получается.

Простое общение с руководителем проекта (вариант – с системным архитектором или кем-то из команды) с озвучиванием проблемы и формальным ее описанием, переводом из несвязных мыслей в голове специалиста: «Все плохо, ничего не выходит» – в четкие формулировки (пусть и не сразу, но сформулировать, чтобы поняли, придется) позволит выявить проблему и найти варианты ее решения или зафиксировать в качестве ограничения проекта. Такая поддержка даст новый виток самомотивации этого специалиста, а там, может, и решение найдется. Причем по факту его находит сам «страдалец», пока объясняет свою проблему, через несколько уточняющих вопросов или вброс идеи: «А если так?..»

На практике РП нужно уметь быть «полковым капелланом» – выслушать, можно вообще при этом молчать. Диалог (монолог специалиста) может быть таким: «У меня проблема... Я уперся в... Пробовал... и... Но не пробовал еще... Точно! Это и проверю. Спасибо за помощь!» РП молча кивает, улыбается и с чувством выполненного долга переключается на другую задачу.

Рассмотрим несколько теорий мотивации:

- теория мотивации Портера-Лоулера;
- теория справедливости Адамса;
- теория Х и Y МакГрегора.

Теория мотивации Портера-Лоулера

Теория Портера-Лоулера была представлена в 1968 г. и основана на утверждении, что на мотивацию человека влияют целый ряд факторов, важнейшими из которых являются: затраченные усилия, полученный результат, вознаграждение, его восприятие и степень удовлетворенности. В модели Портера-Лоулера на мотивацию работника влияют не только ожидание вознаграждения или чувство справедливости, но все эти факторы и иные в совокупности. Рассмотрим эти факторы модели Портера-Лоулера подробнее:

- **Затраченные усилия** – уровень приложенных работником усилий зависит от ожидаемого вознаграждения и уверенности в том, что вознаграждение будет адекватно затраченным усилиям.
- **Полученный результат** – эффективность работы зависит не только от приложенных работником усилий, но и от его способностей, особенностей личности, осознания им своей роли в общем деле и т. д.
- **Вознаграждение и его восприятие** – работник сравнивает полученное вознаграждение с затраченными усилиями и решает, справедливое оно или несправедливое. Если

вознаграждение воспринимается как справедливое, это повышает мотивацию работника, и наоборот.

- **Степень удовлетворенности** – как результат вознаграждения внешний (получена премия, похвала) и внутренний (чувство собственной значимости, самовыражение) и как мерилло его ценности.

Важный вывод из теории Портера-Лоулера – именно результативный эффективный труд ведет к удовлетворению работника, а не наоборот.



Рис. 4.6. Модель мотивации Портера-Лоулера

Практической стороной теории является разделение оплаты труда сотрудника на три части:

- первая часть – оплата за выполнение сотрудниками своих прямых должностных обязанностей, ставка от занимаемой должности;
- вторая часть определяется выслугой лет и факторами стоимости жизни – ее получают все сотрудники организации, но размер выплаты должен автоматически регулироваться;
- третья часть – переменная для каждого работника, зависит от достигнутых им результатов за период. Для плохого работника

третья часть должна быть минимальной. Для хорошего сотрудника – максимальной, примерно как первые две части вместе взятые.

Оклад (первые две части) может увеличиваться только в связи с изменением должности и ответственности, выслугой лет и индексацией на повышение стоимости жизни. Реально заслуженная и заработанная человеком часть заработной платы (третья) может изменяться, и весьма резко, в обе стороны в зависимости от достигнутых результатов. Поэтому производительность труда влечет за собой изменения в суммарной оплате.

Такой подход часто применяется в проектах внедрения со стороны исполнителя, когда у специалистов есть фиксированная часть оплаты и переменная, зависящая от эффективности работы (оплачиваемая по ставкам в человеко-часах). Поэтому специалисты замотивированы делать все как можно быстрее в должном качестве (без лишних затрат на переделки и доделки), укладываться в предварительные и зафиксированные в бюджете проекта оценки задач. Насколько эти оценки честные и реализуемые – отдельный вопрос, рассмотрим его в следующих главах.

Теория справедливости Адамса

Теория справедливости Адамса утверждает, что люди всегда субъективно оценивают соотношение между полученным вознаграждением и затраченными на его достижение усилиями, а также сравнивают его с соотношением других работников, выполняющих аналогичную работу. Люди склонны считать, что они работают много, а получают за эту работу мало, в то время как их коллеги работают меньше, а получают при этом больше. Если человек так считает, то он чувствует несправедливость. Джон Адамс в 1963 году выделил шесть возможных реакций работника на несправедливость:

- сокращение прикладываемых усилий: *«Я не намерен вкалывать за такие копейки!»*;
- попытка добиться увеличения вознаграждения за свой труд: *«Вы без меня пропадете! Хочу прибавку к зарплате и отдельный личный кабинет!»*;
- снижение самооценки: *«Мне платят так мало, потому что я неудачник...»*;

- попытка повлиять на зарплату или нагрузку других работников: *«Всю работу выполняю я один! Пусть коллеги тогда возьмут на себя дополнительные обязанности или не получают премию!»*;
- выбор другого объекта для сравнения: *«Ну, конечно, финансовый директор получает больше меня, но он имеет степень МВА. Зато я получаю больше ИТ-менеджера»*;
- попытка перейти в другой отдел или организацию: *«Меня здесь не ценят, уйду в другую компанию!»*.

Реакции на чувство несправедливости в теории Адамса могут быть различными (от снижения уровня прилагаемых усилий до попытки перейти на работу в другую фирму) и зависят от индивидуальных особенностей работника. Представления о справедливости могут различаться у работников, и не всегда воспринимаемое ими соотношение усилия/отдача соответствует действительности. Поэтому руководитель проекта должен отслеживать возникновение подобных противоречий и вовремя их устранять.

Теория X и Y МакГрегора

Теория МакГрегора опубликована в 1960 году, включает в себя теорию X и теорию Y, которые дают рациональное объяснение факторам мотивации.

Теория X Авторитарный стиль управления	Теория Y Демократический стиль управления
<ul style="list-style-type: none"> ● Люди изначально не любят трудиться и при любой возможности избегают работы 	<ul style="list-style-type: none"> ● Труд – это процесс естественный, а не вынужденная необходимость
<ul style="list-style-type: none"> ● У людей нет честолюбия, и они стараются избавиться от ответственности, предпочитая, чтобы ими руководили, больше всего люди хотят защищенности 	<ul style="list-style-type: none"> ● Важная задача менеджера – дать каждому человеку возможность раскрыть собственные способности
<ul style="list-style-type: none"> ● Чтобы заставить работников трудиться, крайне важно использовать принуждение, контроль и угрозу наказания 	<ul style="list-style-type: none"> ● Если работники приобщены к организационным целям, они будут использовать самоуправление и самоконтроль

В части мотивации и стимулирования по этой теории условно имеем такие подходы:

- теория Х – только «кнуты», «пряники» очень редко;
- теория Y – больше «пряников», «кнуты» просто не нужны (не нужно подгонять, все идет само, главное – не мешать и создать условия).

Коммуникация и конфликты

Говоря об участниках проекта внедрения ERP-системы, нельзя не затронуть важную задачу проекта – коммуникацию. Над проектом работают живые люди (пока еще роботизация не дошла до этой ниши), и то, как настроить взаимодействие, минимизировать конфликты и пробуксовки из-за непонимания и споров, поистине важно.

Руководители проектов тратят большую часть своего времени на осуществление коммуникаций с членами команды и с другими заинтересованными сторонами проекта, внутри своей организации или с внешними специалистами. Если этого не происходит, то, скорее всего, проект управляется без должного внимания и в нем копятся проектные риски, которые могут сработать в самый неожиданный момент.

Согласно РМВОК, **управление коммуникациями** – это процесс обеспечения своевременного и надлежащего сбора, создания, распространения, хранения, извлечения, управления, мониторинга и в конечном счете архивирования/утилизации информации проекта. Ключевая выгода данного процесса состоит в обеспечении эффективного и результативного обмена информацией между командой проекта и заинтересованными сторонами. Этот процесс осуществляется на протяжении всего проекта.

Руководитель проектов должен стремиться к построению доверительных отношений между всеми членами команды проекта, балансировать конфликтующие между собой задачи и цели, использовать навыки убеждения, ведения переговоров, поиска компромиссов. Нужно принять важный факт, что отношения между людьми в проекте важны не менее, чем сам проект. На решение коммуникационных задач нужно выделять достаточно времени (для некоторых РП в сложных проектах с большими командами это будет почти все рабочее время).

Нужно с признательностью принимать обратную связь от специалистов и пользователей, даже если она отрицательная (это же своевременно вскрывает проблемы и позволяет минимизировать проектные риски). В свою очередь, сам РП должен предоставлять конструктивную обратную связь (особенно в случае критики) – все четко по делу, когда из формулировок уже следует последовательность действий, что менять, когда что-то пошло не так. Важным навыком является умение выслушать и задать наводящий вопрос.

Нужно стараться поддерживать в коллективе уважительные отношения друг к другу за счет проявления вежливости, дружелюбия, честности (к себе в том числе), доверия, лояльности и соблюдения этических норм (включая культурные особенности – например, в ситуации заказчика из другого региона или страны).

Все это достигается с помощью различного инструментария:

- личные встречи;
- переписка – формальная и неформальная (почта, мессенджеры);
- телефонные разговоры и видеочаты;
- переговоры и совещания (интервью, опросы, планерки, мозговые штурмы);
- групповые переписки (форумы, чаты, группы в мессенджерах, переписка в комментариях к документу в общем доступе);
- шаблоны проектных документов;
- регламенты и процедуры проектного взаимодействия;
- обучение (по теме проекта, специальностям или личностного роста и эффективности);
- удаленное взаимодействие;
- командообразование (тимбилдинг);
- управление конфликтами.

Нужно искать выгоды для всех заинтересованных в проекте сторон, не забывать, что вокруг живые люди, а не просто решаемые задачи, их постановщики и спонсоры. Не нужно становиться сектой аскетов, сосредоточенных только на проекте и его задачах в ущерб всему остальному. Проекты на пути команды еще будут, сработанная команда – сама по себе ценность. Развитие команды проекта – это процесс совершенствования компетенций, взаимодействия членов команды и общих условий работы команды. Рост компетенций, мотивация, уменьшение текучки кадров, расширение навыков межличностных отношений – все это позитивно влияет на исполнение проекта.

Удаленное взаимодействие

Распределенные команды (виртуальные команды) в XXI веке стали нормой. Процесс глобализации проектов способствовал появлению потребности в виртуальных командах, члены которых при работе над одним и тем же проектом находятся в разных местах (городах и даже странах). Работа таких команд стала возможной благодаря средствам коммуникации через Интернет:

- электронная почта;
- аудио- и видеоконференции;
- мессенджеры;
- общий доступ к документам;
- онлайн-совещания и вебинары, основанные на веб-технологиях интерактивного показа удаленного рабочего стола, голосового и текстового общения;
- виртуальные сети и удаленный доступ;
- таск-трекеры.

Управление виртуальными командами открывает уникальные благоприятные возможности – например, возможность использовать особые знания и опыт в работе команды проекта даже в тех случаях, когда эксперт физически далеко от остальных членов команды.

Можно привлекать работающих на дому сотрудников, а также людей с ограниченными по мобильности возможностями.

Современные технологии позволяют проводить внедрение ERP-системы для ряда предприятий в большей части дистанционно, это позволяет заказчику привлекать исполнителя из любого города, минимизируя накладные расходы на командировки и место в своем офисе для всей команды. Со стороны исполнителя – экономия на аренде большого офиса, возможность привлечения специалистов из других городов (виртуальный офис).

Проблемы управления виртуальными командами связаны главным образом с коммуникациями, включая возможное чувство изолированности, проблемы в обмене знаниями и опытом между членами команды, а также трудности в отслеживании хода работ и эффективном расходовании времени (за спиной у сотрудника уже не постоишь, но у них и мотивация не по таймеру работать, а на результат).

Нужно также учитывать разницу во времени и часовых поясах. Если заказчик и исполнитель в разных часовых поясах, то исполнителю на время проекта логично перестроиться и минимизировать сдвиг времени с заказчиком. Особенно это актуально для поддержки на фазах опытной и промышленной эксплуатации системы. Если же команда самого исполнителя находится в разных часовых поясах, то это может стать преимуществом, если часть специалистов находятся в часовом поясе заказчика. Если же у заказчика филиальная структура в разных часовых поясах, то для его оперативной поддержки исполнителю тоже логично иметь специалистов в этих же часовых поясах, либо временные сдвиги и время реакции нужно явно зафиксировать в договоре и учитывать при планировании взаимодействий.

При всем многообразии доступного инструментария для удаленного взаимодействия нужно ограничить используемый инструментарий, чтобы не было ситуаций: «Я тебе скинул файл, ты смотрел? Куда скинул? В WhatsApp, Skype, на e-mail (на какой адрес?), в сетевую папку, гуглдок, яндексдок, в общий форум, в чат...»

Чем больше каналов связи, тем больше путаницы будет в них, т. к. кто-то любит одно, а другого у него нет, кто-то совсем другое и они не пересекутся и общаться, например, через единый мессенджер не смогут.

Нужно договориться внутри команды и со специалистами заказчика (у них могут быть свои ограничения на запуск софта из-за требований безопасности) – и выбрать единый набор инструментов, где на уровне регламентов зафиксировано, что, например, все рабочие документы передаются только по электронной почте с определенных адресов.

Для работы с проектами внедрения и поддержки 1С:ERP хорошо подходит специальный инструмент «1С-Коннект» (<https://1c-connect.com>) – это корпоративный мессенджер, центр уведомлений и ServiceDesk с множеством различных специальных опций, отсутствующих в обычных мессенджерах для простых пользователей.

Серьезным вызовом для «удаленки» стал 2020 год и карантин во многих странах из-за COVID-19. Даже стойкие противники такого взаимодействия были вынуждены настраивать режим удаленной работы для сотрудников и внешних партнеров. Это стало вопросом выживания бизнеса и дало практический опыт всем. Поэтому в настоящее время к «удаленке» уже нет такого пренебрежения, как было до кризиса.

Возможности удаленного взаимодействия нужно выяснять заранее и закладывать в сроки и бюджет как оптимизацию расходов или, наоборот, удорожание проекта и увеличение его сроков.

Но даже при хорошо налаженной удаленной работе стоит периодически встречаться очно, особенно в начале проекта, на этапе выстраивания отношений, и на этапе опытной эксплуатации, когда коммуникации при первичной поддержке наиболее востребованы. Поэтому проект нужно спланировать, исходя из периодов очного присутствия специалистов исполнителя на территории заказчика, на фоне основной удаленной работы.

Командообразование

Формирование команды и укрепление связей в ней состоит из действий, улучшающих социальные отношения и создающих атмосферы взаимопомощи и сотрудничества в команде. Спектр этих действий широкий: от ежедневных 5–10-минутных планерок с обзором статуса проекта до специальных выездных мероприятий с целью улучшения межличностных отношений среди членов команды.

Командообразование, или **тимбилдинг** (англ. team building – построение команды) – различный набор действий для создания и повышения эффективности работы команды. Идея командных методов работы заимствована из мира спорта и стала активно внедряться в практику менеджмента в конце XX века. Тимбилдинг направлен на создание групп равноправных специалистов различной специализации, сообща несущих ответственность за результаты своей деятельности и на равной основе осуществляющих разделение труда в команде.

Из личного опыта автора: идея тимбилдинга – веревочный курс (веревки где-то до 0,7 метра над землей, без страховок). Прохождение уровней в тишине (общение только жестами, мимикой и неречевыми звуками, типа «mmm»), когда одна рука на плече коллеги по команде, а второй нужно удерживать себя, а где-то и коллег (а где-то они вас). Уровень засчитывается, когда его прошли все, не отпуская рук, не падая. Неважно, сколько это времени займет, сколько попыток. Пока не будет выработана тактика у команды, распределены роли (кто идет первый, кто кого удержит, кто страхует, как вытащить последнего) – задание не выполнить. Может занимать часы. Одиночке не пройти, там специально расстояния такие, что нужно проходить парами или тройками, иначе просто не хватит длины рук. Потом все друг друга прекрасно с этого «mmm» понимали.

Тимбилдинг убирает стены и барьеры межличностных отношений, новые сотрудники быстро вливаются в коллектив, и команда, не имея совместных успешных проектов за плечами, становится сработанной – это увеличивает шансы на успех, т.к. экономит на времени притирки друг другу (до пары недель с начала проекта). И это время (а пара недель

умножить на часы и количество участников – это человеко-месяцы) будет сразу эффективным, даст быстрый старт. А видимые результаты – это тоже мотивация. Когда все сложно, непонятно и еще какие-то притирки в команде, то «руки опускаются». А тут все замотивированы с первого дня.

Потому какие-то совместные мероприятия по выстраиванию коммуникации полезны «до» проекта, а не только после, как отмечание успеха или хотя бы: «Ну, наконец-то это закончилось». Завершающее мероприятие, понятно, тоже нужно и полезно. Причем совместно для всех участников проекта, полной проектной командой исполнителя и заказчика.

Предложения по проведению тимбилдингов можно изучить самостоятельно в Интернете и выбрать предпочтительный формат по физическим и психологическим возможностям команды, некоторые мероприятия требуют определенной физической формы участников и хорошей погоды, что не всегда возможно, если проект уже начался. Любые конкурсы, корпоративные мероприятия, отмечания дней рождения и, конечно, национальные праздники – тоже варианты командообразования. Просто эффективность у них разная.

Цель укрепления команды – помочь отдельным ее членам результативно работать друг с другом. Это особенно важно, когда команда распределенная (кто-то удаленный, кто-то в офисе, кто-то «на клиенте») и нет возможности личного общения. Неформальное общение участников и соответствующие мероприятия могут помочь построить доверительные отношения и установить хорошие рабочие взаимоотношения. Руководитель проекта должен постоянно отслеживать взаимоотношения в команде, влияние их на функционирование, эффективность и результативность всей команды, выявлять, требуются ли какие-либо действия для предотвращения или устранения проблем и конфликтов в команде.

Управление конфликтами

В проекте внедрения ERP конфликтные ситуации встречаются обязательно. Природа конфликтов проявляется:

- в недостатке ресурсов (времени, людей, финансирования);

- приоритетах задач и их содержании;
- стиле работы и отношений между специалистами.

Формирование сработанной команды, добропорядочные отношения в коллективе, соблюдение социальных норм поведения и устоявшиеся практики управления проектом, такие как планирование коммуникаций и распределение ролей, способствуют снижению числа возникающих конфликтов, их интенсивности и последствий для проекта.

Успешное управление конфликтами дает высокую производительность работ при позитивных рабочих отношениях (эмоциональный фон в коллективе). Наличие разных мнений по каким-либо вопросам является положительным фактором, способствующим творческому подходу и выработке лучших решений, а не конфликтам, вырастающим из споров. Должно быть «в споре рождается истина», а не драка.

Когда разные мнения становятся причиной конфликта, то в первую очередь члены команды проекта несут ответственность за поиск компромисса и выработку общего решения. Если происходит обострение конфликта, то руководитель проекта должен способствовать его урегулированию. Конфликт следует гасить на ранней стадии и желательно конфиденциально: «не выносить сор из избы», а решать только между сторонами конфликта. Если конфликт переходит в деструктивную стадию, то для его решения могут быть использованы формальные процедуры (описывающиеся в уставе проекта, являющиеся регламентами по фирме), включая меры дисциплинарного воздействия.

Если «игра в демократию» (выбор решения большинством голосов) не дает эффекта, то какие-то противодействующие стороны умирятся в приказном порядке («будем делать так!»). Хотя, конечно, лучше всех убедить, но времени и эмоций может уйти больше, чем будет эффект – все зависит от того, кого нужно убеждать. Если это условный стажер на время летней практики, то затраты времени на его убеждение более разумного предела неоправданны. А если это системный архитектор и вопрос на уровне выбора архитектуры и работ на тысячи человеко-часов всей команды, то тут и неделю можно на исследование и поиск доводов потратить, собрать прототип, но все в рабочем порядке, приняв решение отложить конфликтный вопрос на время после необходимых работ по уточнению «за» и «против» предмета спора.

Автор лично попадал в различные конфликтные ситуации – как сторона конфликта, так и снаружи его, как РП. Однажды был долгий спор при проектировании ТЗ между консультантом и системным архитектором, для выхода из него пришлось потратить полдня разработчика, чтобы собрать живой, кликабельный (но без алгоритмов) прототип, показать его заказчику и выбрать вариант, устраивающий всех (это мог быть и предлагавшийся кем-то вариант, а не еще один, но вживую на прототипе его плюсы убрали предмет спора).

Конфликты возможны не только между людьми, но и с ресурсами, приоритетами и требованиями проекта – например, какие-то требования не реализуются в заявленном виде вообще или за разумное время, не хватает времени или бюджета. Такие конфликтные ситуации являются рисками проекта и после их выявления решаются между людьми, и тут уже будет компромисс или конфликт (уже конкретных участников проекта). Поэтому, так или иначе, невзирая на природу конфликтов, ответственными именно за перерастание рабочего момента в конфликт являются люди (конкретные люди конкретных психотипов, ролей и должностей).

Пример конфликта «заказчик – исполнитель», возникшего из-за сроков и бюджетов по изначальному конфликту в требовании (занижены оцениваемые сроки и трудозатраты).

Исполнитель: «В процессе проектирования задачи вскрылись ограничения реализации. Сделать можем, но это потребует дополнительный оплачиваемый человеко-месяц. Делаем?»

Заказчик: «Делаем, но денег не дадим!»

Успех РП в управлении командой проекта сильно зависит от способности разрешать конфликты. Можно применять различные методы их разрешения. Факторы, влияющие на методы разрешения конфликтов, включают в себя:

- важность и напряженность конфликта (мелкие конфликты можно решать фоном или жить с ними годами, критичные нужно сразу разбирать);
- ограниченность времени, доступного для разрешения конфликта (бросить все и заняться этим – или можно зафиксировать и решать в плановом порядке);
- относительная власть вовлеченных в конфликт людей (какой смысл долго и эмоционально спорить с лицом, не принимающим решения);
- важность сохранения хороших отношений (искушение заткнуть чьи-то доводы приемом «я начальник, ты дурак» не всегда хорошо и долго работать не сможет – нужно убеждать, если время позволяет);
- мотивацию к разрешению конфликта в долгосрочной или краткосрочной перспективе (конфликты – это тяжелый эмоциональный груз РП, их часто хочется пустить на самотек, но нужно найти в себе силы и решать такие задачи).

Существует пять основных методов, используемых для разрешения конфликтов:

- **Уклонение/избегание** – отступление от фактической или потенциальной конфликтной ситуации, перенос решения проблемы на более поздний срок, чтобы лучше подготовиться к ее разрешению или передать его другим лицам.
- **Сглаживание/приспосабливание** – подчеркивание точек соприкосновения вместо областей противоречий, отказ от своей позиции в пользу потребностей других, чтобы сохранить гармонию и взаимоотношения.

- **Компромисс/урегулирование** – поиск решений, которые будут в определенной степени удовлетворительными для всех сторон, чтобы временно или частично разрешить конфликт. Результатом этого подхода в некоторых случаях является ситуация, когда победители отсутствуют.
- **Принуждение/указания** – лоббирование чьей-либо точки зрения за счет других, предлагая только решения «один выиграл – все проиграли», обычно с позиции власти, чтобы разрешить критическую ситуацию. Результатом этого подхода часто является ситуация, когда кто-то выигрывает, а кто-то проигрывает.
- **Сотрудничество/разрешение проблем** – объединение множества точек зрения и взглядов с различных перспектив, готовность к сотрудничеству и открытому диалогу, которая обычно приводит к достижению консенсуса и поддержке решения всеми сторонами. Результатом этого подхода может быть ситуация, когда выигрывают обе стороны конфликта.

Как уже отмечалось, для РП управление конфликтами особенно важно, поэтому настоятельно рекомендуется более детально самостоятельно изучить возможный инструментарий для минимизации конфликтов (используя, например, материалы РМВОК).

Глава 5. Этапы и документация проекта

Жизненный цикл проекта внедрения ERP-системы

Ранее мы рассмотрели вопросы выбора ERP-системы, формирования требований к ней, технологии проектного управления и кто, собственно, участвует в таком проекте. Пришло время рассмотреть более конкретно, как именно проходит жизненный цикл проекта внедрения ERP, по каким этапам и фазам, какой документооборот на каждой фазе.

В общем случае, согласно РМВОК, проекты, которые отличаются размером и степенью сложности, отраслями приложения (например, не ИТ-проекты, а строительство зданий), структуру жизненного цикла имеют типичную:

- начало проекта;
- организация и подготовка;
- выполнение работ;
- завершение проекта.

Жизненный цикл проекта – это набор фаз, через которые проходит проект с момента его начала до момента завершения. Он определяет основные рамки управления проектом.

Мы будем рассматривать более конкретное приложение управления проектами – внедрение ERP-системы и исходить из этой специфики:

- начало проекта – инициализация работ по проекту внедрения ERP-системы;
- организация и подготовка – анализ, проектирование, планирование;
- выполнение работ – разработка, настройка, тестирование, обучение, поставка, запуск в эксплуатацию;
- завершение проекта – финализация работ по проекту, извлеченные уроки, переход к поддержке системы.

Начало проекта имеет особую важность, особенно когда сроки очень жесткие или же проект выполняется с приоритетом по срокам (часто бывает запуск в промышленную эксплуатацию с нового года). Для проектов автоматизации на базе 1С:ERP запуск с даты начала периода (особенно с начала года, как начало нового года в новой системе) – это частотное требование. А это, по сути, фиксирует одну из сторон проектного треугольника уже в начале проекта, если запаса по времени для маневра нет.

В начале пути возможность повлиять на конечные характеристики проекта и его стоимость максимальны, тогда как сами работы активно еще не ведутся, затраты и привлекаемые специалисты малочисленны. Есть время на «раскачку», вдумчивое проектирование и согласование решений. Когда же работы вовсю идут, то «запаузить» 10+ специалистов для «подумать и решить» очень сложно. Будет или простой в работе, или гонка, чтобы успеть принять решение, пока по нему не началась активная работа. С ходом проекта, особенно в его конце, способность повлиять на проект («а давайте все это переделаем по-другому») стремится к нулю (либо такие изменения равносильны запуску нового проекта).

Основной пик затрат и привлечения исполнителей приходится на фазу выполнения работ и к завершению проекта они снижаются. Тогда как вероятность успешного завершения проекта обратна вероятности срабатывания проектных рисков (вероятности рисков в начале проекта максимальны) и к концу проекта без сработанных рисков вероятность успешного завершения максимальна.

Схематически это можно представить в виде графиков.



Рис. 5.1. Графики влияния, затрат и вероятности успеха проекта

Сложные проекты могут быть разбиты на этапы. Имея общие фазы начала и завершения проекта для всех этапов, внутри каждого этапа повторяются фазы организации, подготовки и выполнения работ. При этом выходом этапа может быть запуск в промышленную эксплуатацию какого-то ограниченного набора блоков ERP-системы. А сами этапы могут идти с наложением друг на друга и в общем случае делаться даже разными командами.

Такая вариативность проектов внедрения ERP продиктована тем, что в основе лежит готовая к использованию система, а бизнес-процессы и требования к автоматизации у разных компаний разные, поэтому степень изменений системы и частота поставок (передача блоков и функциональности в эксплуатацию) в проектах отличаются.

Рассмотрим разные типы жизненных циклов (ЖЦ):

- **предиктивный** – содержание, сроки и стоимость определяются на ранних фазах, далее идет последовательное однократное выполнение фаз;

- **итеративный** – последовательные итеративные улучшения или модификация системы на основе обратной связи через настройку прототипа;
- **инкрементальный** – частая поставка готовой к использованию функциональности системы;
- **гибкий** – подход, который сочетает черты итеративного и инкрементального жизненных циклов и направлен на внедрение системы с увеличением частоты поставок (частый перевод функциональности в эксплуатацию, получение обратной связи от пользователей, agile-методики);
- **гибридный** – подход, который подразумевает комбинацию предиктивного, итеративного, инкрементального и/или гибкого подходов.

Характеристики жизненных циклов:

Тип ЖЦ	Требования	Действия	Поставка	Цель
Предиктивный	Фиксированные	Выполняются один раз за весь проект	Одна	Управление стоимостью
Итеративный	Динамические	Повторяются, пока не будет сделано правильно	Одна	Правильность решения
Инкрементальный	Динамические	Выполняются один раз для конкретной итерации	Частые более мелкие поставки	Скорость
Гибкий	Динамические	Повторяются, пока не будет сделано правильно	Частые небольшие поставки	Ценность для клиента через частые поставки и обратную связь



Рис. 5.2. Предиктивный жизненный цикл



Рис. 5.3. Итеративный жизненный цикл



Рис. 5.4. Инкрементальный жизненный цикл

Для каких-то компаний подходит запуск в эксплуатацию системы после небольшого периода настройки с последующей доработкой по результатам работы пользователей с системой (обратная связь), для других компаний нужно серьезно доработать систему под требования бизнеса, сделать интеграцию с другими системами и запускать все в сборе. От этого зависит выбор типа жизненного цикла проекта.

Место каждого типа ЖЦ в зависимости от степени изменений и частоты поставки работоспособной версии показано на графике ниже.



Рис. 5.4. Применение ЖЦ в зависимости от степени изменений и частоты поставок

Все типы жизненных циклов основываются на одинаковом наборе фаз внутри этапов (блоков из последовательных фаз).

Фаза проекта – совокупность логически связанных операций проекта, завершающихся достижением одного или ряда поставляемых результатов. Свойства конкретной фазы могут быть измеряемыми и уникальными. Свойства могут включать в себя, среди прочего:

- название (например, «Фаза А», «Фаза В», «Фаза 1», «Фаза 2», «Фаза подготовки предложения»);
- количество фаз (например, три фазы в проекте, пять фаз в проекте);
- длительность (например, 1 неделя, 1 месяц, 1 квартал);
- требования к ресурсам (например, человеческие ресурсы, сооружения, оборудование);
- критерии входа для проекта, чтобы перейти в данную фазу (например, необходимые одобрения задокументированы, необходимые документы разработаны);
- критерии выхода для проекта, чтобы завершить данную фазу (например, одобрения задокументированы, документы разработаны, поставляемые результаты завершены).

Для проектов внедрения ERP-системы обычно выбирают итеративный жизненный цикл, когда в проекте высокая степень изменений (или работ, связанных с настройкой множества блоков), а частота поставки низкая. Но допустимы и другие варианты, все зависит от конкретной ситуации.

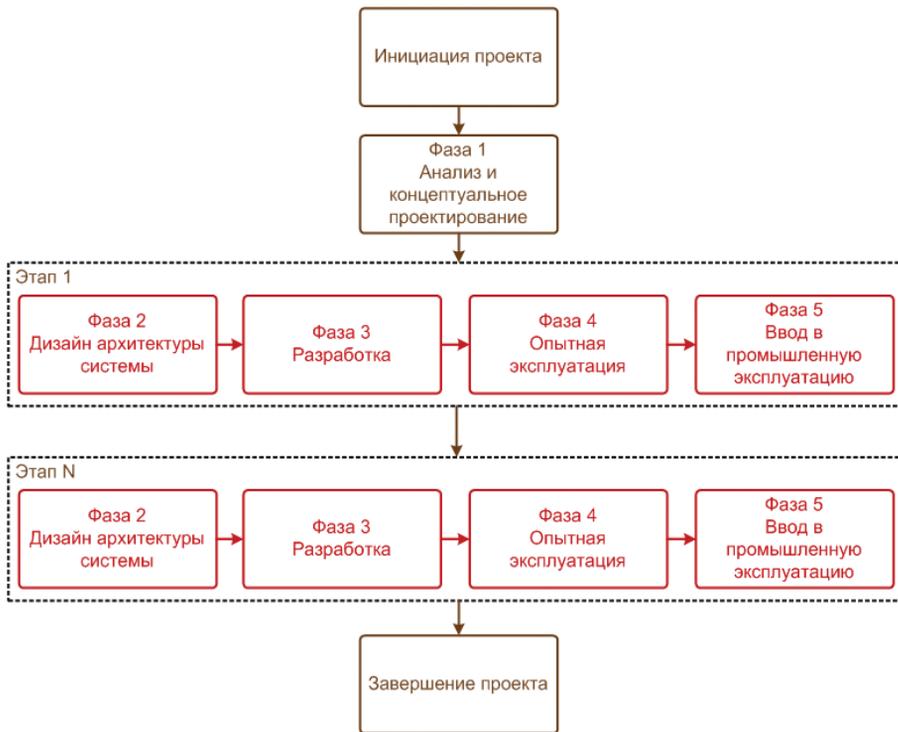


Рис. 5.5. Этапы и фазы проекта внедрения ERP-системы

При этом фазы группируются в этапы проекта, которые могут идти последовательно или с наложением, это переводит жизненный цикл в формат гибких или гибридных ЖЦ. После **инициации проекта** и всех начальных процедур проводятся **анализ и концептуальное проектирование** для всех этапов проекта (или ближайшего этапа, с высокоуровневым планом на остальные этапы). Далее по каждому этапу выполняются фазы:

- **дизайн архитектуры системы** – прототипирование и подготовка технического проекта этого этапа;
- **разработка** – фаза настройки, разработки, тестирования, подготовки руководств пользователей и обучения;
- **опытная эксплуатация** – начало работы в системе пользователей на предварительном переносе начальных данных, закрепление навыков работы, получение обратной связи

от пользователей, проверка работоспособности функциональности в реальных условиях;

- **ввод в промышленную эксплуатацию** – проверка и довод начальных данных, сверка с прошлыми системами (для перехода с других систем) и отказ от старых систем, перевод запущенных блоков в режим сопровождения.

Названия фаз могут отличаться, могут группироваться или, наоборот, выделяться отдельные фазы (например, обучение пользователей выделяется отдельно).

По завершении каждого этапа в системе прибавляется введенный в промышленную эксплуатацию функционал. После реализации всех запланированных работ на всех этапах выполняется фаза **завершения проекта**, на которой закрываются работы по проекту, подводятся итоги и достигнутые результаты, проект завершается. Система переходит в режим поддержки – уже как отдельный процесс, а не проект внедрения (силами того же исполнителя, либо внутренними специалистами, либо третьей стороной).

Ниже рассмотрим более детально каждую фазу и документооборот по ней.

Документация проекта

Документация по проекту внедрения ERP-системы имеет важное значение. Настоятельно рекомендуется вести как рабочие, так и отчетные документы своевременно и с должной детализацией.

Документация является средством коммуникации, управления, позволяет обезопасить обе стороны от возможных конфликтов и судебных разбирательств (либо отстоять в суде свою позицию, если до этого дойдет).

Если формулировки требований, изначально имеющие краткое описание, не переложить в более формально описанные технические задания, которые будут понятны всем участникам и подписаны специалистами заказчика (или хотя бы на уровне электронной переписки утверждены), то все стороны проекта не будут понимать, чего ожидать

от реализации этого требования, какие ограничения принимаются, какие алгоритмы закладываются. Остается простор для разных трактовок. А если все оформлено документами и формально описано, то все участники проекта будут «на одной волне».

Это сразу снимет ситуации завышенных ожиданий заказчика и защитит от этого исполнителя, например, уберет реплики пользователей вида: «Ой, а мы думали, что будет все иначе и делаться само по одной кнопке». Точнее, такие ситуации, конечно, возможны первые пару раз, пока специалисты заказчика не осознают степень ответственности работы над проектом, включая задачи по вычитке и утверждению рабочих документов проекта. Конфликт таких ожиданий по уже сделанному функционалу снимается показом утвержденного перед разработкой документа («сделано по ТЗ!») и предложением записать новые требования в журнал изменений для последующего пересмотра границ (сроков и бюджета) проекта, если потребуется. При этом утвержденные технические задания все равно остаются рабочими документами и могут пересматриваться по необходимости, но в управляемом режиме с пониманием ответственности за пересмотр и рисков изменений границ проекта. Сам процесс формального описания ТЗ тоже хорошо помогает в проектировании и понимании задачи и вариантов ее решения, так как кажущееся разработчику «мне все тут понятно» при формализации приводит к множеству вопросов у консультанта и конечного пользователя, и эти вопросы закрываются «на бумаге», а не переделкой кода в процессе тестирования и эксплуатации («ой, а этот сценарий мы совсем не учли»).

Обратная ситуация хорошей проектной документации – это защита интересов заказчика с помощью документации проекта. Зафиксированные в ТЗ моменты должны работать в системе, и если в процессе опытной эксплуатации выявляются недоработки и отступления от ТЗ, то они должны быть устранены исполнителем.

На заметку. Фраза: «Все, что вы скажете, может быть использовано против вас» – очень хорошо работает по отношению к написанному в проектной документации. Причем в обе стороны.

Поэтому не нужно писать в ТЗ какие-то несбыточные и фантастические вещи, нужно явно указывать, как это будет – «автоматически» или «вручную», иначе этого потом будут ожидать и нужно будет все делать или создавать риск конфликтной ситуации: «Ну, мы совсем не это имели в виду».

При использовании гибких agile-методик ведения проектов документация минимизирована согласно манифесту о важности людей и самой работающей системы, но для внедрения проектов ERP такой подход «без документации» имеет ряд минусов.

Применяемая проектная документация описывается по фазам проекта и фиксируется списком и шаблонами в уставе проекта. Это инструментарий для достижения целей проекта и визуальная часть хода проекта и его «бумажных» результатов. Дело в том, что систему и программный код в ней очень сложно представить и оценить неспециалисту, тогда как полноценная проектная документация вызывает доверие к результатам проекта и именно в ней описаны все возможности системы и как с ней работать пользователям.

Нужно помнить, что нет цели гнаться за объемом документации. Объемную документацию сложно готовить, трудоемко поддерживать в актуальном состоянии и, главное, сложно читать и согласовывать. Документация должна содержать необходимую и достаточную информацию для единого понимания всех участников проекта, без лишних, незначимых описаний и отступлений. А такая оптимальная документация и для гибких методик хорошо подойдет и лишней (даже в нарушение agile-манифеста) не будет.

Можно привести крылатое выражение А.П. Чехова: «Краткость – сестра таланта». Вот и проектная документация должна быть в меру краткой, а для этого нужен определенный талант и навык у специалистов, ее делающих. На помощь приходят шаблоны документации и внутренние тренинги, как и что в ней описывать.

У исполнителя должны быть готовые шаблоны на все типы документов и понимание, как и кто формирует эти документы. Возможна ситуация, когда у заказчика есть свои предпочтения по применяемой проектной методологии, тогда требования к документам и их шаблоны предоставляет сам заказчик.

Проектные документы разделяются на рабочие и отчетные:

- **рабочие документы** – используются для выработки и фиксации проектных решений, не подписываются всем управляющим комитетом в бумажном виде с печатями организаций, а могут быть только в электронном виде с утверждением по почте с цифровой подписью (опционально);
- **отчетные документы** – подписываемые на бумаге управляющим комитетом документы, имеют важное значение для фиксации принятых решений, как дополнительные соглашения к договору, влияющие на финансовые расчеты и обязательства сторон.

Так как различных видов документов на разных фазах проекта может быть довольно много, полезно описать, кто что делает. Это полезно не только для документации, конечно, но почти все в проекте документируется в том или ином виде, поэтому рассмотрим такой инструмент в данном разделе книги. Для таких целей хорошо подходит диаграмма RACI:

- **Responsible** – отвечает;
- **Accountable** – утверждает;
- **Consult** – консультирует;
- **Inform** – информируется.

Диаграмма RACI является удобным инструментом для четкого распределения ролей и сфер ответственности, когда в команде есть внутренние и внешние ресурсы. Часто ее представляют в виде таблицы: список документов и операций – по строкам, а по столбцам – сотрудники или роли в команде, которые в этом задействованы по R, A, C и I действиям. Пример такой таблицы представлен ниже.

Операция	ФИО_1 РП исполнителя	ФИО_2 РП заказчика	ФИО_3 Консультант	ФИО_4 Ключевой сотрудник
Устав проекта	R	A	I	I
Тех. задания	C	I	R	A
Запрос на изменение	A	C	I	R
Руководства пользователей	C	A	R	I

Далее рассмотрим более подробно фазы проекта и документацию по ним.

Шаблоны самой документации в данной книге не приводятся, их можно взять в технологии корпоративного внедрения 1С:ТКВ из «1С:ПрофКейс».

Фазы проекта

Инициация проекта

Инициация проекта – это фаза до начала активных работ над проектом и заключением самого договора на проект – так называемый «пред-проект».

В предыдущих главах рассматривались вопросы выбора заказчиком ERP-системы, внутреннего сбора требований к ней, проведения тендера. Это и есть предпроектные работы.

Фаза может быть от очень краткой (неделя) до вялотекущего внутреннего процесса на стороне заказчика (месяцы, полгода, год) – фоновый анализ ERP-систем на рынке, неспешный сбор требований и т.п. В какой-то момент начинается активность, приглашают потенциальных исполнителей, а для них эта фаза «бесплатная», потому затягивать ее им самим резона нет. Хотя если какие-то рассмотрения и согласования

требуют времени, то предпроект затягивается, но календарно в днях, а не из-за больших трудозатрат исполнителя на задачи в нем. Затраты в этой фазе несут обе стороны:

- заказчик – на фонд оплаты труда своих сотрудников, возможно, на командировки;
- исполнитель – на привлечение специалистов на экспресс-анализ, отчет по анализу, настройку прототипа для показа, презентации.

Целями фазы являются:

- Установить границы и рамки проекта – достижение понимания среди заинтересованных лиц проекта, что войдет в будущую систему и будет внедрено, а что не войдет.
- Определение участников проекта со стороны заказчика и выбор исполнителя – экспресс-анализ собранных требований и бизнес-задач.
- Заключить договор на работы и начать сам проект.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Приказ о начале работ над проектом	Отчетный	Директор заказчика
Спецификация требований	Рабочий	РП заказчика
RFI/RFP/RFQ	Отчетный	РП заказчика
Соглашение о неразглашении	Отчетный	РП заказчика
Коммерческое предложение	Отчетный	РП исполнителя
Опросники	Рабочий	Консультант
Протоколы совещаний	Рабочий	РП исполнителя

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Отчет об экспресс-обследовании	Отчетный	РП исполнителя
Презентации системы и проекта	Рабочий	РП исполнителя, консультант
Устав проекта	Отчетный	РП заказчика (по факту РП исполнителя)
План-график проекта (предварительный)	Отчетный	РП исполнителя
Договор на проект	Отчетный	РП исполнителя

Нужно отметить, что устав проекта, который, по идее (позиция РМВОК), должен делаться на стороне заказчика как первый документ, инициализирующий проект и назначающий руководителя проекта и его участников, по факту не делается вообще либо делается стороной исполнителя, чтобы зафиксировать все регламенты взаимодействия и состав команды (ситуации, когда данный документ не делается совсем, тоже бывают, но это уже не проектная методология, а ее отсутствие). Вместо устава на стороне заказчика проект инициируется приказом по компании о начале работ и выделении полномочий руководителю проекта (или назначении этой роли дополнительно какому-то сотруднику). Приказ может быть в виде электронного письма и формально отчетным (с подписью и печатью) документом не будет, либо вообще устным («давайте с понедельника этим займемся, ты будешь крайним»). Как и что анализировать, оценивать сроки и бюджет проекта, мы рассмотрим в следующих главах. Здесь же остановимся более подробно на уставе проекта. Он крайне важен для инициирования работ, с ним вся команда проекта одинаково понимает инструментарий взаимодействия, регламенты документооборота, цели проекта и критерии их достижения.

Устав проекта содержит разделы:

- **История изменений** – версии, даты, кто и что менял (устав – «живой» документ и может уточняться итерационно и потом по ходу проекта тоже).
- **Назначение документа** – для чего это все, показать важность документа.

- **Основные понятия и определения** – чтобы говорить на одном языке и ввести словарь сокращений и принятых в проекте терминов (особенно терминов заказчика, т.к. на фирмах имеют свой сленг и кучу аббревиатур, совсем непонятных внешнему исполнителю).
- **Основания и порядок изменения документа** – критерии, когда можно и даже нужно менять документ (например, пересмотр целей и границ проекта в ходе анализа и дизайна).
- **Цели проекта** – что именно ожидается от проекта внедрения ERP-системы, по пунктам, основные результаты проекта.
- **Границы проекта** – организационные и функциональные границы проекта, какие подразделения и функциональные блоки системы задействованы в проекте.
- **Критерии успешности проекта** – список критериев, по которым проект будет оцениваться в конце как выполненный (очень важно иметь понятный критерий завершения проекта, иначе можно попасть в вечный цикл: «А вы нам еще не сделали... и мы не можем закрыть вам акт выполненных работ»).
- **План-график проекта** – может быть отдельным документом, но часто включается в устав (хотя бы высокоуровневый, если отдельно ведется более детальный).
- **Основные контрольные точки проекта** – таблица событий и ожидаемых дат, когда они должны произойти: подписание документов, завершение фаз, запуск функциональности в работу, получение каких-то результатов от учета в системе и т. п.
- **Организационная структура проекта** – описание структуры команды проекта (заказчика и исполнителя), управляющего комитета и перечень конкретных специалистов (ФИО, контакты), преемственность, требования к квалификации.

- **Подход к выполнению работ** – описание проектной методологии, каждой фазы, документооборота по фазам и кто делает (согласует) документы, места и время выполнения работ, инструментарий коммуникаций (для фиксации возможности удаленного взаимодействия и программных средств).
- **Риски проекта** – список потенциальных рисков и инструментарий управления ими для своевременного выявления и минимизации.
- **Проектные процедуры** – порядок взаимодействия команды проекта: совещания, протоколирование решений, порядок согласования документов (время согласования на каждый тип документа), порядок планирования отпусков и обучения, управление изменениями, удаленное взаимодействие, управление конфликтами, управление рисками, урегулирование спорных ситуаций.
- **Ограничения и допущения проекта** – факторы внешней и внутренней среды, влияющие на проект.
- **Секция утверждения устава проекта** – реквизиты сторон, подписи, печати, как на договоре.
- **Приложения** – шаблоны проектных документов.

Успешное завершение этой фазы заключается в определении границ проекта и заключении договора на проект внедрения ERP-системы сразу на весь проект, либо только на первый этап, либо только на детальный анализ и концептуальное проектирование. Последнее – это самый удобный для исполнителя и компромиссный для заказчика вариант, позволяющий уточнить границы и бюджет проекта, а не «угадывать» после экспресс-анализа. Заказчику это дает время приглядеться и поработать со специалистами исполнителя, получить готовые проектные документы этой фазы и принять решение о продолжении (или замене исполнителя) работ на следующих фазах проекта. И, конечно, на этой фазе решаются финансовые вопросы: примерный бюджет проекта, покупка лицензий на систему, график оплаты работ по внедрению. В договоре фиксируется график платежей или формула/критерий для вычисления графика по ходу проекта. Тут возможна вариативность:

- аванс (30–50 %) и постоплата в конце;
- аванс (до 20–30 %) и оплаты по факту выполненных работ (по месячным актам, фазам или иным достижимым критериям);
- иные варианты, компромиссные для сторон проекта.

После всех работ по фазе и принятии решений о начале работ, подписания договора уместно провести совместный тимбилдинг команды проекта или иное мероприятие для налаживания более тесных контактов между ключевыми сотрудниками заказчика и специалистами исполнителя.

Анализ и концептуальное проектирование

Сразу отметим, что данной фазы может не быть отдельно от следующей в жизненном цикле фазы – дизайна архитектуры системы. В ситуации, когда требования на входе в проект детальные и все эти предварительные работы провел сам заказчик или сторонняя компания для него ранее (то есть эта фаза уже была завершена ранее, вне этого проекта). Если же понимания, что и для чего делаем, целевого вида системы и бизнес-процессов «как будет» нет, то такая фаза нужна, и она отделена от дизайна архитектуры системы. Также эта фаза требуется, если проект предполагает несколько этапов и нужно спроектировать целевую систему на все этапы в высокоуровневом виде, а более детальный дизайн архитектуры и работы по внедрению будут только по первому этапу.

Поэтому, по сути, эта фаза – продолжение предпроектных работ, но уже на возмездной основе (то есть оплачиваемых), с выделенным сроком, сформированной проектной командой (и выделением на эти работы сотрудников и их времени со стороны заказчика). Может быть оформлена как отдельный мини-проект с отдельным договором, например на аудит ИТ-инфраструктуры, бизнес-процессов и формирование детальных требований к ERP-системе.

Если ранее на фазе инициации проекта был экспресс-анализ, то тут идет полноценный анализ требований, документооборота заказчика, отчетов, бизнес-процессов, регламентов и должностных инструкций

сотрудников, готовится формальное описание бизнес-процессов и концепция их автоматизации в виде концептуального проекта будущей системы автоматизации заказчика.

Нужно отметить, что на данный момент заказчик еще может не определиться с выбором самой системы автоматизации и такой анализ и формулирование детальных требований могут быть совмещены с анализом их покрытия той или иной системой или отраслевым решением внутри единой платформы. Например, на данный момент может быть еще не ясно, что лучше подходит для нужд заказчика: «1С:ERP Управление предприятием» или «1С:Управление холдингом».

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Актуализированная спецификация требований	Рабочий	РП заказчика и РП исполнителя
Внутренние документы для анализа (отчеты, регламенты, базы данных)	Рабочий	РП заказчика
Развернутая 1С:ERP с прототипом	Система	РП исполнителя
Протоколы совещаний	Рабочий	РП исполнителя
Концептуальный проект (отчет об обследовании)	Отчетный	РП исполнителя
Презентации концепций	Рабочий	РП исполнителя, консультант, системный архитектор
Устав проекта (обновление)	Отчетный	РП исполнителя
План-график проекта (уточненный)	Отчетный	РП исполнителя
Договор на этап проекта (или доп. соглашение на уточнение срока и бюджетов)	Отчетный	РП исполнителя
Акт выполненных работ по фазе	Отчетный	РП исполнителя

Дизайн архитектуры системы

На этой фазе идет техническое проектирование по утвержденной ранее концепции и спецификации требований, проверяется и уточняется техническая реализуемость требований и концепций, покрывающих автоматизируемые бизнес-процессы. Для этого в развернутой ранее для демонстрации возможностей ERP-системе настраивается прототип на функциональности, что уже есть в системе (как «коробке»). Настраиваются опции, включающие и отключающие функциональность под потребности конкретного внедрения (все лишнее скрыть, нужное включить). Проектируется нормативно-справочная информация (НСИ), на базе которой будут работать внедряемые блоки.

Настройка НСИ – это крайне важный момент, определяющий успех будущей автоматизации. Собственно, это и есть те самые «кирпичики» фундамента архитектуры, на которых потом будет надстраиваться вся остальная система и используемая в ней функциональность. По ходу настройки прототипа уточняются места «разрывов» функциональности, которую предстоит доработать на фазе разработки.

Конечно, современные ERP-системы стремятся к покрытию всех потребностей бизнеса на уровне штатной функциональности. И в случае внедрения 1С:ERP очень многое можно настроить без дополнительного программирования, но бизнес и оперативный, и особенно управленческий, учеты не унифицированы под единые стандарты и ПБУ, потому готовые решения могут не подойти. Возможно, что какая-то особенность бизнес-процессов компании является ее конкурентным преимуществом на рынке, и тогда, конечно, нет смысла отказываться от такого процесса, а нужно поддержать его автоматизацией.

Все эти проектные решения описываются в сводном документе «Технический проект» (может называться иначе – например, «Дизайн решения»). Это еще не детальное техническое задание (ТЗ), это высокоуровневый свод, но уже и не концепция, а конкретные задачи, которые дальше могут уточняться в ТЗ на следующей фазе.

РМВОК вводит понятие иерархической структуры работ (ИСР) – детальной декомпозиции всех необходимых настроек, доработок и прочих работ. **Иерархическая структура работ** проекта – это дерево подзадач различного уровня, необходимых и достаточных для достижения целей проекта. Вот после фазы дизайна архитектуры системы

в техническом проекте и плане-графике работ должна быть полная картина работ по ИСР. Далее остается на последующих фазах их выполнить и по мере отметки выполнения задач мониторить ход проекта; когда все задачи будут выполнены, то и проект на этом заканчивается.

Если все спроектировать на этой фазе не получается, то такие моменты стоит выносить в отдельный этап (вторую очередь), записывать явно в ограничения либо использовать гибкие методики ведения проекта, когда уточнить требуемую реализацию будет возможно только после получения обратной связи по запущенным в работу блокам. Но такие «белые пятна» не позволяют четко зафиксировать бюджет и срок проекта на остальные фазы. Тогда все это нужно отразить в дополнительном соглашении к договору, чтобы проект на стал «резиновым» по доработкам при фиксированных сроках и бюджете.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Актуализированная спецификация требований	Рабочий	РП заказчика и РП исполнителя
Настроенный прототип 1С:ERP	Система	РП исполнителя
Протоколы совещаний	Рабочий	РП исполнителя
Технический проект (дизайн решения)	Отчетный	РП исполнителя
Презентации архитектуры решения	Рабочий	РП исполнителя, консультант, системный архитектор
Устав проекта (обновление по необходимости)	Отчетный	РП исполнителя
Уточнение сроков и бюджета остальных фаз проекта	Рабочий	РП исполнителя
План-график проекта (уточненный по необходимости)	Отчетный	РП исполнителя
Доп. соглашения к договору (по необходимости)	Отчетный	РП исполнителя
Отчет о выполненных работах (месячный или по фазе целиком)	Отчетный	РП исполнителя
Акт выполненных работ (месячный или по фазе целиком)	Отчетный	РП исполнителя

Разработка

На этой фазе выполняются подготовка и утверждение технических заданий на модификации системы, разработка и тестирование запланированных модификаций функциональности системы по этим ТЗ. Настраивается система из прототипа в конечные настройки для эксплуатации. Обновляются или формируются новые инструкции пользователей. Разрабатывается инструментарий переноса начальных данных из текущих систем или иных баз, вырабатываются методика и отчеты сверки и коррекции начальных данных.

Разворачивается инфраструктура «разработческих», тестовых и рабочих баз, настройка доступа пользователей и системы резервного копирования.

Проводится подготовка к опытной эксплуатации, выполняется развертывание системы в промышленной конфигурации и пробный перенос исторических данных (возможно, итеративно, с доработками и проверками). Проводятся обучение и аттестация пользователей системы.

В сложных проектах внедрения ERP-системы это самая длительная по времени фаза, длится от нескольких месяцев до нескольких лет (хотя логично ограничиться сроками в квантах по 2–9 мес., порезать задачи на этапы проекта и запускать блоки по мере готовности).

В простых проектах, где внедряется «коробка» без модификаций, фаза сводится к обучению пользователей и настройкам системы с переносом начальных данных (если это не новый бизнес вместе с новой системой), а это позволяет провести все работы в сжатые сроки (в один или несколько месяцев) и сразу перейти к опытной эксплуатации.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Протоколы совещаний	Рабочий	РП исполнителя
Технические задания	Рабочий	Консультант, системный архитектор
Презентации технических решения	Рабочий	РП исполнителя, консультант, системный архитектор
Запрос на изменение	Рабочий	РП заказчика
Журнал изменений	Рабочий	РП исполнителя
Уточнение сроков и бюджета остальных фаз проекта	Рабочий	РП исполнителя
План-график проекта (уточненный по необходимости)	Отчетный	РП исполнителя
Инструкции пользователя	Рабочий	Консультант
Презентации с обучающим семинарам	Рабочий	Консультант
График обучения пользователей	Рабочий	РП исполнителя
Акт передачи инструкций пользователей	Отчетный	РП исполнителя
Результаты аттестации пользователей	Рабочий	Консультант
Чек-лист тестирования системы	Рабочий	Консультант
Сообщения об ошибках	Рабочий	Пользователь
Система готова к опытной эксплуатации	Система	РП исполнителя
Регламент проведения опытной эксплуатации	Рабочий	РП исполнителя
Доп. соглашения к договору (по необходимости)	Отчетный	РП исполнителя
Протокол загрузки НСИ и начальных данных	Отчетный	РП исполнителя

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Отчет о состоянии проекта (периодически)	Рабочий	РП исполнителя
Отчет о выполненных работах (месячный или по фазе целиком)	Отчетный	РП исполнителя
Акт выполненных работ (месячный или по фазе целиком)	Отчетный	РП исполнителя

Опытная эксплуатация

Система запускается в опытную эксплуатацию. Пользователи нарабатывают практику использования системы, закрепляют навыки, полученные в ходе обучения, исправляются ошибки, обнаруживаемые в ходе опытной эксплуатации. В начале этой фазы планируется список мероприятий и ответственных за них:

- кто и когда настроит рабочие места пользователей в сети заказчика;
- настройка системы в промышленной конфигурации (перенос с тестовых серверов на рабочие);
- процедуры переноса и выверки начальных данных;
- графики подключения пользователей и отключения текущих систем (если отказ от них планируется сразу);
- прочие особенности хода фазы.

Опытная эксплуатация позволяет проверить работоспособность системы в условиях, максимально приближенных к режиму промышленной эксплуатации, а также отработать процедуры запуска системы в промышленную эксплуатацию (в том числе перенос исторических данных на нужную дату). Точный режим проведения опытной эксплуатации (например, возможно, параллельная работа пользователей одновременно в унаследованных системах и во внедряемой ERP-системе с периодической выверкой формируемых системами данных) определяется в начале фазы и фиксируется регламентом опытной эксплуатации.

После завершения опытной эксплуатации, исправления всех обнаруженных ошибок и актуализации инструкций пользователя выполняется запуск системы в промышленную эксплуатацию.

Фаза может длиться несколько недель или даже месяцев, проходить итерациями, если выявляются стоп-факторы, мешающие работе и нужно время на устранение проблем (а то и откат на предыдущую фазу с серьезной доработкой). Возможно бесшовное перетекание опытной эксплуатации в промышленную, когда работа идет на реальных данных и все бизнес-процессы покрыты, а фаза явно затягивается по времени. Тогда остается оформить это событие в следующей фазе формальным вводом системы в промышленную эксплуатацию.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Протоколы совещаний	Рабочий	РП исполнителя
Регламент проведения опытной эксплуатации (уточненный)	Рабочий	РП исполнителя
Журнал проведения опытной эксплуатации	Рабочий	РП исполнителя
Запрос на изменение	Рабочий	РП заказчика
Журнал изменений	Рабочий	РП исполнителя
Сообщения об ошибках	Рабочий	Пользователь
Уточнение сроков и бюджета остальных фаз проекта	Рабочий	РП исполнителя
План-график проекта (уточненный по необходимости)	Отчетный	РП исполнителя
Инструкции пользователя (уточненные)	Рабочий	Консультант
Результаты аттестации пользователей (новых пользователей)	Рабочий	Консультант
Система готова к эксплуатации	Система	РП исполнителя

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Доп. соглашения к договору (по необходимости)	Отчетный	РП исполнителя
Отчет о состоянии проекта (периодически)	Рабочий	РП исполнителя
Отчет о выполненных работах (месячный или по фазе целиком)	Отчетный	РП исполнителя
Акт выполненных работ (месячный или по фазе целиком)	Отчетный	РП исполнителя

Ввод в промышленную эксплуатацию

На фазе ввода в промышленную эксплуатацию сотрудники заказчика начинают полноценную работу в системе. Еще раз переносятся или доводятся начальные данные. Исполнителем совместно с сотрудниками центра компетенции осуществляются начальная поддержка, проверка данных, сбор запросов на изменение, коррекция пользовательской документации по результатам поддержки. По завершении фазы функционал системы передается в поддержку центру компетенции.

После успешной (пусть и не сразу, а итерациями) опытной эксплуатации эта фаза короткая, может занимать одну-две недели – на выверку начальных данных и формализацию передачи системы (запущенных блоков) в поддержку. Критерием завершения может быть успешно закрытый в системе учетный период (например, квартал). Если поддержкой занимается не сам исполнитель, а третья сторона или команда заказчика, то специалисты исполнителя приступают к следующему этапу проекта (если этого не произошло ранее, когда специалисты начали высвобождаться к концу фазы опытной эксплуатации). Если этап в проекте единственный или последний, то выполняется переход к завершающей фазе – формальному закрытию проекта. Если же фаза затягивается, то это, по сути, все еще идет опытная эксплуатация, на которой выявляются все огрехи внедрения (пользователи недообучены, система имеет ошибки, не все начальные данные перенесены). Просто такая опытная эксплуатация идет как промышленная, с отказом от старых систем (стратегия запуска «некуда отступать»). Могут потребоваться итерации переноса начальных остатков и доработок. В какой-то момент

все равно будет переход в нормальный режим промышленной эксплуатации (спад потока проблем, все работает штатно), что нужно оформить документами, передать систему в поддержку и завершить проект.

Поэтому как вариант обе фазы могут быть объединены в одну – «опытно-промышленная эксплуатация», но документооборот по ним тоже тогда сводный, и выход из проекта и передачу в поддержку запущенных блоков нужно оформить явным образом.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Протоколы совещаний	Рабочий	РП исполнителя
Регламент перехода к промышленной эксплуатации	Рабочий	РП исполнителя
Журнал проведения промышленной эксплуатации	Рабочий	РП исполнителя
Запрос на изменение	Рабочий	РП заказчика
Журнал изменений	Рабочий	РП исполнителя
Сообщения об ошибках	Рабочий	Пользователь
Инструкции пользователя (уточненные по необходимости)	Рабочий	Консультант
Работающая система передается в поддержку	Система	РП исполнителя
Отчет о состоянии проекта	Рабочий	РП исполнителя
Отчет о выполненных работах по фазе целиком	Отчетный	РП исполнителя
Акт выполненных работ по фазе целиком	Отчетный	РП исполнителя

Заккрытие проекта

Формальное завершение проекта и отношений по договору на внедрение ERP-системы, переход к процессу поддержки, расчеты за работы по проекту внедрения. Проводится совещание по закрытию проекта, посвященное оценке его результатов, достигнутых целей с участием всех заинтересованных сторон (спонсор проекта тоже участвует). Возможно, совещания и презентации разделены на две встречи: исполнитель и заказчик отдельно, заказчик и спонсор проекта – отдельно.

Документооборот и результаты фазы:

Результат (документ)	Рабочий/отчетный	Ответственный за создание
Протоколы совещаний	Рабочий	РП исполнителя
Презентация по результатам проекта	Рабочий	РП исполнителя
Акт сдачи-приемки работ по проекту	Отчетный	РП исполнителя
Отзывы, пресс-релизы, видеоролики о результатах внедрения	Отчетные	РП исполнителя

Ну и, конечно, после подписания всех результирующих документов уместно это дело отпраздновать совместной командой проекта. Хотя часто проект внедрения всех так утомляет, что «я бы вас всех больше не видел». Но такое неформальное общение нужно, чтобы разрядить обстановку и побыть командой (тимбилдинг), для реабилитации утомленных (скорее эмоционально, чем физически) специалистов.

После завершения проекта на стороне исполнителя нужно подвести свои итоги (так называемые «выученные уроки»), поделиться полученным опытом с коллегами из других проектных групп. Организовать эту внутреннюю презентацию, а также архивирование проектной документации и баз данных с прототипами и решениями тоже должен руководитель проекта. А еще нужно подготовить и опубликовать пресс-релиз об успешном завершении проекта, выступить с докладом на одном из мероприятий фирмы «1С», можно поучаствовать в номинации на проект года (<https://eawards.1c.ru>).

Но и это еще не все! Хороший, успешный проект и работающая ERP-система в компании заказчика является отличной площадкой для референс-визитов других потенциальных клиентов. А так как бизнес динамически меняется, то и задачи новые могут возникнуть, и к услугам исполнителя заказчик может еще не раз обратиться – уже для проектов развития своей используемой ERP-системы.

Глава 6. Что анализировать и как настроить прототип системы

С чего начать анализ

В процессе участия в тендере на внедрение ERP-системы обязательно встанет задача проведения вначале экспресс-анализа, а потом и полноценного анализа для выработки понимания границ проекта и его бюджета. Как и что анализировать?

Когда команда опытная и идет не первый проект у участников, то имеется отработанная последовательность действий (хотя бы у одного проектного ветерана), как и что делать, а вот когда кто-то участвует в этой фазе в первый раз (особенно для самих потенциальных пользователей на стороне заказчика), то понимания сути процесса анализа может не быть. Давайте разберемся.

Те, кто проводят анализ (специалисты исполнителя), до какого-то момента о компании заказчика могли просто не слышать вообще или представлять себе смутно из СМИ, что эта компания собой представляет, чем занимается, ее масштаб и региональных охват.

Первым источником информации, конечно, является сайт компании, раздел «О компании» на нем. Полезна любая иная информация, которую можно найти в поисковых системах, могут быть какие-то статьи и пресс-релизы из жизни компании на сайтах известных СМИ ресурсов.

Этой информации для инициирования работ и первого разговора исполнителя с заказчиком на одном языке достаточно. Важно все это изучить до, а не после первых встреч, презентаций и разговоров, так как потенциальному исполнителю крайне желательно говорить в терминологии (отраслевой специфики бизнеса) заказчика. Иначе вас просто не поймут.

Показ демопримера из ERP-системы необходимо адаптировать под специфику бизнеса. Если это производство, условно, спецтехники, то не нужно показывать плюсы системы на базе примера торговли продуктами питания. Не будет знакомых слов – будет сразу: «Ну, это система явно не для нашего бизнеса...»

Если в открытых источниках информации о компании мало, то нужно запросить вводную у заказчика (на этот момент уже может быть готов для передачи пакет документов, включая RFI/RFP). Если информации мало, то лучше разделить встречу на предварительную (кто вы, что хотите) и основную (показ системы и возможностей исполнителя). Если никакой формально описанной документации на вход не дают, то ее придется собирать (запрашивать) адресно. Бывают исключения, когда у заказчика уже проведена фаза анализа – самостоятельно или сторонней организацией – и все документы собраны, требования формально описаны. Но чаще это не так: приглашают исполнителя-внедренца из расчета, что там профессионалы и они сами все сделают. В общем случае всегда есть что анализировать, поэтому просим предоставить:

- учетную политику;
- должностные инструкции сотрудников (очень редко, когда они есть, но вдруг будут);
- регламенты по фирме (описание бизнес-процессов, документооборота, последовательности действий сотрудников);
- приказы по процессам в компании (аналог регламентов);
- схему организационной структуры (обычно точно есть устаревшая на пару лет, но ее охотно обновят, если попросить);

- таблицу структурных единиц и количество сотрудников – потенциальных пользователей КИС в них;
- список текущих информационных систем с комментариями, кто и что в них делает, какие потоки данных между системами (может не быть актуальной схемы, тогда рисуем в процессе опросов);
- инструкции пользователей от текущих информационных систем (если они вдруг есть, то, вероятно, актуальны на момент внедрения КИС);
- примеры управленческих отчетов, которые предстоит автоматизировать с уточнением: кто их делает, как часто, как долго, где берет информацию, кто и что потом анализирует (это все точно есть, это центральное для автоматизации место, но кто, что и как – придется выяснять опросами);
- какие-то аудиторские заключения по бизнесу заказчика и публичная отчетность, например по МСФО (если ранее привлекались аудиторы);
- типовые бухгалтерские проводки (скорее всего, их не дадут, но будут рады, если на вход исполнитель даст свое о них представление, и уже по нему уточнят свои особенности);
- список ключевых сотрудников по подразделениям и бизнес-процессам, с кем можно пообщаться для уточнения деталей;
- просим заполнить опросные листы (можно заочно по формуляру, можно очно консультантом исполнителя по мере задавания вопросов);
- какие-то иные документы в зависимости от специфики бизнеса, примеры бланков внутреннего документооборота.

Удивительно, но от первоначального «у нас ничего нет» не остается и следа, собирается увесистая папка файлов на несколько сотен мегабайт. В рамках экспресс-анализа может даже не хватить времени все поизучать, но будут понятны объемы работ для полноценного анализа.

Помним об обязательном соглашении о неразглашении (NDA – non-disclosure agreement), подписанном ранее. Обмен файлами производим по защищенному каналу, а если через e-mail, то как минимум через запароленный архив. Условия и каналы передачи файлов прописываются в NDA.

На стороне заказчика процесс сбора этих документов делается руководителем проекта через опрос (рассылку по электронной почте) по ключевым сотрудникам. Далее они уже сами рассылают своим коллегам или просто знают, где что лежит на серверах в сети компании.

Еще одно исключение, когда документации на входе нет – это стартап. Заказчик действительно может не иметь никакой документации, т.к. не оброс ей, как не иметь и ключевых сотрудников для интервью (еще не наняли или они работают первую неделю). От исполнителя и ERP-системы в этом случае ожидается не просто автоматизация, а «пусконаладка» самого бизнеса, когда по прошлым внедрениям и каким-то отраслевым решениям на базе ERP-системы исполнитель готов предоставить некую практику работы бизнеса «от и до» через систему автоматизации бизнес-процессов на конкретных сотрудниках компании заказчика. Это очень интересная задача, тут нет багажа из «у нас так было всегда, нам нужно это автоматизировать». Но и ответственность несколько иная, и требуемая квалификация специалистов (а от этого и стоимость услуги), чем просто при автоматизации. Это уже тот самый бизнес-консалтинг при внедрении ERP, когда запускается сам бизнес, а не просто система автоматизации в нем. В этой ситуации нужно запрашивать у идеолога новой компании, с кем планируется работа, какие бизнес-процессы уже определены, что и кто на них влияет (оборудование, его поставщики, кто клиенты). Тогда для анализа пойдет информация о сторонах (контрагентах, банках и т.п.), товарах и услугах, с которыми эта компания начинает выстраивать бизнес, это привнесет свои требования к автоматизации и даст вводную для анализа специалистам исполнителя.

Информацию получили, просмотрели, засылаем «десант специалистов» исполнителя к ключевым пользователям заказчика. Подспорьем тут будут опросные листы с заранее подготовленными вопросами (по мотивам прошлого проектного опыта или из шаблона методологии

внедрений). Такие вопросы являются хорошей точкой входа для начала общения. Далее по мере уточняющих вопросов можно вывести сотрудника заказчика на рассказ о бизнес-процессах, в которых он участвует, системах, базах и документах, которые использует (создает или потребляет). Все ответы нужно конспектировать, по результатам встреч подготовить протокол.

Встречи могут иметь разный формат:

- На рабочем месте сотрудника – он может показать конкретные системы и файлы, находится в комфортной для него ситуации; более открытое общение без лишних участников (нет простоя других сотрудников, т. к. других сотрудников на такой встрече нет).
- В переговорной комнате. Тут возможны варианты: «1 на 1» или «все на все» как формат совещания. Совещание «все на все» – это не интервью ключевых сотрудников и его не заменяет, информация в таком формате будет неполной, люди будут скованы присутствием коллег, будут лишние реплики и увод обсуждения в сторону, затягивание встречи с меньшей отдачей.
- Показ внедряемой ERP-системы и сбор обратной связи: «Да, у нас так», «Нет, у нас не так», «Почти как у нас, можно доделать...» В формате «1 на 1» или «все на все». Нужно отметить, что чем больше прототип системы настроен под заказчика, тем качественнее будет обратная связь и лояльность к показываемой системе (знакомые термины, родные подразделения и НСИ и т. п.).

В целом уместно использовать все три формата встреч последовательно: индивидуально, показ прототипа, обсуждение коллегиальное. Только между встречами нужно закладывать время на «переваривание» информации, подготовку тезисов, настройку прототипа.

Выделяемое на такие интервью время, количество ключевых пользователей заказчика и привлекаемых консультантов исполнителя (возможность параллельного или последовательного опроса) будут определять длительность и стоимость таких работ. Также нужно учесть потребность в командировках. При всей любви к дистанционным работам

и удаленному доступу, фазу анализа в части опроса и показа за одним компьютером рабочих мест ключевого сотрудника эффективнее проводить очно. После таких плотных совместных работ можно перейти к удаленному взаимодействию, т. к. уже будет первичная сработанность консультанта исполнителя и сотрудника заказчика.

Собрав всю необходимую (возможную к предоставлению за разумное время) информацию, проведя опросы и интервью с сотрудниками, специалисты исполнителя приступают к составлению отчета об экспресс-обследовании, а после (через несколько итераций встреч, если требуется) – и к составлению концептуального проекта.

Готовим отчет

Проведение анализа и составление итогового отчета по концепции проекта – задача весьма интересная, но для специалистов, которые не имеют большого опыта таких работ, изначально сложная. Поручать такую работу консультанту без должного проектного опыта – неблагоприятное дело, так можно подвести все стороны проекта и породить ненужные проектные риски.

Этой задачей должен заниматься опытный специалист, прошедший несколько аналогичных фаз в других проектах. Всегда бывает «первый раз» – и как тогда его пройти? Нужно привлекать новичков в помощь такому «проектному ветерану». Задач хватит на всех, но контроль за логикой, подачей информации в отчете, проектными решениями и оформлением должен осуществлять понимающий и опытный специалист. То есть итоговый отчет по фазе анализа готовят несколько консультантов под чутким руководством системного архитектора либо самого руководителя проекта (если он в прошлом имеет большой проектный опыт и «собаку съел» на всем этом).

Хороший шаблон из проектной методологии тоже является секретом успеха: он содержит структуру и описание разделов, и даже «пустой» шаблон может быть на 10 страницах только из-за детализации структуры, по сути, являясь опросником для консультанта, – остается узнать то-то и то-то и сформулировать выявленное в виде формального описания.

Вроде все просто, но процесс «пропускания через себя» большого потока информации, четкое формулирование текста и выработка проектных решений требуют определенного склада ума и понимания предметной области.

Можно вспомнить известную фразу древнекитайского философа Лао-цзы: «Даже путь в тысячу ли начинается с первого шага». Вот и подготовку отчета нужно с чего-то начать, перебороть страх перед пустой первой страницей.

Наличие хорошего шаблона и подсказок в нем – это хорошо, но как понять, какие подразделы нужны и что в них будет? Для этого занимаемся максимально возможной декомпозицией «непонятного» на части, чем меньше эти части будут, тем проще их сделать «понятными», а поняв, формально описать.

У нас на входе довольно много информации, полученной от заказчика, ее нужно переработать, используя конспект проведенных интервью и опросные листы. Начинаем описывать бизнес-процессы и роли пользователей «как есть» – это понятно, т.к. вся информация есть, всегда можно уточнить у ключевых сотрудников заказчика какой-то непонятный момент, проверить и согласовать схемы.

Далее это «как есть» критически рассматриваем применительно к системе и входящим требованиям, попутно проверяя какие-то гипотезы в прототипе системы (настраиваем цепочки документов). Проводим мозговые штурмы внутри команды по необходимости. Проводим fit-gap анализ, рисуем схемы ИТ-ландшафта – текущего и целевого (подробнее про это было в главе 2).

Обычно системный архитектор знает систему, что называется, наизусть, и ему достаточно прочесть требование и посмотреть схему бизнес-процесса, чтобы понять, как это настроить в системе и может ли она такое вообще, а если не может, то что можно доделать, чтобы смогла.

Попутно из «как есть» приходим к схемам «как будет», это продиктовано возможностями системы и здравым смыслом свежего пересмотра бизнес-процессов в ходе анализа – реинжиниринг бизнес-процессов на этой фазе очень к месту.

Крайне нежелательно слепо переключившись в систему все как есть без попыток пересмотра, мотивируя тем, что специалистам заказчика виднее, это их фирма, их бизнес. Сыграем в простую игру: представьте, что вас наняли в эту фирму на эти самые ключевые должности. Вы бы все так и оставили или, может, что-то поменяли, исходя из свежего взгляда со стороны на компанию (и опыта других компаний по другим проектам)? Вот что вы бы сами себе меняли, то и нужно предлагать менять сотрудникам заказчика.

Стоит давать рекомендации (консультации), рассматривать разные варианты на выбор, будучи консультантом со стороны исполнителя, руководителям (и ключевым специалистам) подразделений со стороны заказчика. Они на своих местах нормально это воспринимают: потому и зовут специалистов со стороны, что сами изнутри не могут что-то заметить (а своих подчиненных, даже если те скажут то же самое, что и вы, скорее всего, не услышат). Если предложение имеет минусы – вам прямо на это укажут (новая информация к анализу), а вот если предложение пройдет: «А действительно, что это мы так раньше не сделали?» – то это победа.

Ситуация со стороны заказчика – нужно чаще спрашивать консультантов исполнителя: «А как можно еще это сделать?», «Какую альтернативу

вы можете предложить?» На данной фазе коммуникация особенно важна. Специалисты исполнителя и сотрудники заказчика еще не сработались до понимания друг друга с полуслова. Исполнитель может принять (под давлением или по иным причинам) тактику «любой каприз за ваши деньги» («что скажете, то и будем делать»). Нужно со стороны заказчика не подавлять («нам виднее, это наша фирма!») инициативу рассмотрения разных вариантов, а, наоборот, поощрять креатив специалистов исполнителя. Помним график из главы 5 про возможность влияния на проект в начале, при проектировании? Вот на этой фазе, где все пока еще слова, даже документа описания может не быть, бумага с рисунками, схемы на флипчарте и маркерных досках (не забываем, кстати, все с них фотографировать, потом может пригодиться для перерисовки в документ) – это все быстрая коммуникация и рассмотрение вариантов.

Нужно понимать, что в реальности между мыслями одного человека, их формулировкой (речью и текстом) и пониманием другого человека этого всего лежат фильтры «искажений», меняющие и урезающие поток информации. Наглядно это можно представить на схеме.



Рис. 6.1. Схема искажения понимания при коммуникации «исполнитель – заказчик»

Схема работает в обе стороны: консультанты исполнителя также фрагментарно понимают рассказ или описание регламентов от специалистов заказчика. Мало того, это и внутри группы работает – между, например, консультантом и разработчиком. Собственно, это один из факторов того, что в какой-то момент наращивание проектной команды сотрудниками

останавливает ускорение в решении задач, т.к. накладные расходы на коммуникациях съедают этот прирост от дополнительных «человеко-часов». Какие методики есть для налаживания коммуникаций, рассматривалось в главе 4 «Кто участвует в проекте внедрения». Показ прототипа и презентация тезисов и выводов готового отчета – тоже важный инструмент выстраивания коммуникации. Со стороны не видно, что думает человек при прочтении документа, зато реакция на озвучивание тезисов и показ схем на очной презентации очень заметна, можно быстро задать уточняющие вопросы в обе стороны, проверить, что все теперь имеют одинаковое представление о проектных задачах и границах. После таких встреч отчет можно уточнять итерационно и после рассылать текстовую версию на согласование.

Рассмотрим типовую структуру документа «Концептуальный проект» (он же «Отчет об обследовании»):

- **История изменений** – версии, даты, кто и что менял (документ может уточняться итерационно по мере согласования и обсуждения презентаций).
- **Принятые сокращения, понятия и определения** – словарь сокращений и принятых в проекте и используемых далее по тексту терминов.
- **Введение** – назначение документа, когда и как проходил анализ и концептуальное проектирование, кто участвовал.
- **Общие сведения о заказчике и области автоматизации** – описание контекста для последующих пунктов: о компании, бизнес-направлении, оргструктура, используемое программное обеспечение, текущая схема ИТ-ландшафта.
- **Бизнес-требования** – кратко со ссылкой на таблицу-приложение (в таблице работать с требованиями удобнее), входящие требования для автоматизации (по мере концептуального проектирования уточняемые).
- **Основные бизнес-процессы** – описания бизнес-процессов в виде схем (или иерархических списков) и пояснений к ним «как есть» (может не быть) и «как будет» (как целевая модель).

- **Концепция решения** – долгосрочное видение системы (включая перспективный ИТ-ландшафт), которая должна быть создана и внедрена и которая будет удовлетворять всем бизнес-требованиям, разделение требований по этапам проекта и целевые схемы системы и ИТ-ландшафта на каждый этап.
- **Факторы успеха проекта** – критерии, по которым можно будет измерить и определить, что проект завершился успешно.
- **Секция утверждения** – подписи согласующих (может быть на самом титульном листе, так нагляднее).
- **Приложения** – необходимые приложения, вынесенные из основного текста: схемы, таблицы, графики, ссылки на другие файлы.

Как же подготовить отчет? Простые шаги, с чего начать:

- Взять шаблон документа и заполнить шапку шаблона и историю изменений.
- Составить свое оглавление (последовательность подпунктов). Оглавление делать заголовками с пустым содержанием.
- Расположить полученные от заказчика и нарисованные на флипчартах схемы и диаграммы в целевых разделах документации.
- Декомпозировать подзаголовки дальше – получаем иерархическую структуру документа, что будет соответствовать ИСР (иерархической структуре работ) проекта.
- Начинаем наполнять все подразделы, уточняем непонятные моменты у заказчика и коллег, проводим мозговые штурмы, настраиваем прототип, проектируем.
- Включаем в приложения описание используемой нотации для схем бизнес-процессов.

При завершении документа:

- Прочитать полученный документ самому так, как будто это совершенно новый документ, который писал кто-то другой (очень хороший трюк для взгляда со стороны).
- При этом нужно задавать себе вопросы: «А это что?», «Откуда это?», «А что будет, если?..»
- Если ответов на эти вопросы по тексту нет, дописать пояснения по месту возникновения.
- После итерации вычитки отдать на проверку коллегам и далее по процедуре специалистам заказчика.
- Полученные в ходе анализа и проектирования уточненные требования к автоматизации критически рассматриваем и приоритезируем с учетом всех изученных и проанализированных документов и процессов заказчика.

Приоритизация требований

В главе 2 рассматривались вопросы сбора требований и fit-gap анализа по ним. На фазе анализа пересматриваются изначальные приоритеты по требованиям уже с учетом возможностей ERP-системы и разумной компоновки границ проекта по срокам и бюджету. Для приоритизации можно воспользоваться методом **MoSCoW** (легко запомнить и понять). Метод получил свое название от акронима, образованного следующими классификациями приоритета (буква «о» добавлена для созвучности):

- **Must have** – обязательно должно быть (жизненно необходимые требования, без чего не будет работать);
- **Should have** – следовало бы иметь (обязательные требования, без чего сложно обойтись);
- **Could have** – может быть (стоит сделать для удобства работы);
- **Would like** – хотелось бы иметь (можно не делать, «бантики»).

Вся эта методика прекрасно иллюстрируется на примере шариковой ручки, состоящей:

- из стержня – без него никак нельзя что-то написать (must have);
- корпуса – писать одним стержнем крайне неудобно, нужная вещь (should have);
- зажима на корпусе, автомата убирания стержня – полезные для удобства работы опции, но без них уже можно пользоваться инструментом для письма (could have);
- логотипа и фонарика в торце – на функции работы шариковой ручки никак не влияют, просто опции (would like).



Рис. 6.2. Приоритизация MoSCoW на примере шариковой ручки

Эта простая методика позволяет сгруппировать требования в этапы автоматизации и согласовать их с заказчиком. Без привлечения заказчика специалисты исполнителя могут расставить приоритеты только примерно, на свое усмотрение и исходя из понимания замкнутости цикла учета в системе, но потребности ведения бизнеса могут вносить свои коррективы. И, наоборот, приоритеты только со стороны заказчика не учитывают специфику настройки и работы системы, что-то ненужное на первый взгляд может оказаться крайне нужным для целостного учета в системе. Потому работа с приоритетами – это совместное мероприятие.

И тут исполнителю тоже нельзя занимать позицию «как скажете, так и сделаем», нужно все время проверять полученный пул задач на реализуемость и сочетаемость между собой. Например, нельзя сделать какое-то требование с приоритетом *must have*, если оно делается поверх функциональности, помеченной *would like* и в этот этап вообще не входящей.

В торге и спорах за приоритеты хорошо помогают оценки реализации (время и стоимость) этих требований. Они сразу отрезвляют поток пожеланий и расставляют все точки над «i»: когда что-то по факту не нужное (но очень желаемое – *would like*) стоит дорого, оно сразу становится не критичным. Как оценивать сроки и бюджет проекта, рассмотрим в следующей главе.

Описание бизнес-процессов

Описание бизнес-процессов в виде графических схем в определенной нотации – это очень интересная, творческая деятельность фазы анализа и концептуального проектирования. При проведении экспресс-анализа на такое описание время не выделяется, в лучшем случае на составление списка самих бизнес-процессов к последующему описанию и автоматизации. Конечно, можно все описывать только словами в виде последовательного по пунктам текста, но в силу вариативности и «ветвистости» некоторых процессов текст такой очень сложно понять, в отличие от наглядной схемы. Также используемая нотация описания бизнес-процессов накладывает требования на определенное сочетание элементов разного типа (действие, информация, условия, исполнитель), что вынуждает более тщательно продумывать процесс, чем при описании его текстом. В схеме просто не получится нарисовать иначе, как ответив на значимые вопросы. Такое моделирование «вытягивает» из консультанта информацию для формализации в схемах для отчета, а если этой информации нет, то такие вопросы переадресуются ключевым сотрудникам заказчика. За счет этого сама процедура описания бизнес-процессов позволяет выявить всю значимую информацию.

Под методологией создания модели бизнес-процесса понимается совокупность способов, при помощи которых объекты реального мира и связи между ними представляются в виде модели в определенной графической нотации. Основное в методологии – дать пользователю последовательность шагов, которые приводят к заданному результату.

Некоторые используемые во внедрениях ERP-систем нотации для описания бизнес-процессов:

- **Business Flow Chart** – функциональные блок-схемы (бизнес-схемы). Адаптация использования блок-схем описания программных алгоритмов для бизнес-схем. Детальное описание блок-схем приведено в ГОСТ 19.701-90 (ИСО 5807-85) «Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».
- **UML** (Unified Modeling Language) – унифицированный язык моделирования. Семейство нескольких нотаций для описания любых систем, включая описание бизнес-процессов. Позволяет перейти от описаний системы непосредственно к написанию компьютерных программ. Большинство нотаций посвящено именно архитектуре программ, нотации для бизнес-процессов довольно ограничены по возможностям и визуализации.
- **IDEF** (Integrated DEFinition) – семейство нотаций для решения задач моделирования сложных систем. Для задачи описания бизнес-процессов интерес представляют IDEF0 и IDEF3:
 - **IDEF0** (Function Modeling) – методология функционального моделирования. С помощью схем IDEF0 изучаемая система (предприятие) предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков). Моделирование средствами IDEF0 обычно является первым этапом изучения и высокоуровневого описания любой системы (предприятия);
 - **IDEF3** (Process Description Capture) – методология документирования процессов, происходящих в системе (предприятии), описываются сценарий и последовательность операций для каждого процесса. IDEF3 имеет прямую взаимосвязь с IDEF0 – каждая функция (функциональный блок) может быть представлена в виде отдельного процесса схемой IDEF3.
- **DFD** (Data Flow Diagrams) – диаграммы потоков данных. Инструмент структурного анализа и проектирования

информационных систем. Является структурной нотацией, используемой в бизнес-анализе и в анализе информационных систем. Схемы DFD хорошо сочетаются с IDEF0 и IDEF3. Смысл DFD-блока, отображающего функцию, совпадает со смыслом блоков IDEF0 и IDEF3, заключается в преобразовании входов в выходы. DFD-блоки также имеют входы и выходы, но не поддерживают управление и исполнителей. DFD может быть высокоуровневой схемой, которая далее раскрывается в IDEF0.

- **ЕРС** (Event-Driven Process Chain) – событийная цепочка процессов, ключевыми элементами которой являются «События» и «Функции». Очень распространенная нотация для моделирования и описания бизнес-процессов, хорошо сочетается с IDEF0, как более наглядная замена нотации IDEF3. Имеет расширенный вариант – eЕРС:
 - **eЕРС** (extended Event-Driven Process Chain) – нестрогий вариант ЕРС, когда можно использовать дополнительные элементы схем для большей наглядности – например, «база данных» символом цилиндра.
- **ВРМН** (Business Process Model and Notation) – нотация для моделирования бизнес-процессов. Имеет простой и понятный конечному пользователю набор элементов. Нотация современная (появилась в 2004 году, стабилизировалась в 2008–2013 г.) и сегодня активно используется для описания бизнес-процессов, для чего она и была изначально предназначена, а не адаптирована из описания ИТ-систем.

Приведенный список не конечен, читатель может самостоятельно изучить иные средства моделирования, но приведены основные используемые инструменты, с которыми предстоит столкнуться в проектах внедрения ERP-систем.

В силу того, что чтение схем в разных нотациях требует определенных знаний самой нотации и принципов моделирования, а сотрудники на стороне заказчика этого могут не знать, нотацию нужно выбирать осторожно, исходя из ее понятности неспециалистам и быстроты освоения «чтения» схем. По этой причине очень хороши в использовании блок-

схемы, они сразу понятны большинству специалистов, по нотации блок-схем есть ГОСТ 19.701-90.

Впрочем, порог вхождения для чтения схем сильно ниже, чем для самого их рисования и моделирования процессов компании. Поэтому достаточно предоставить заказчику описание нотации и провести небольшую презентацию по используемой (предлагаемой к использованию) в проекте методике описания бизнес-процессов – и можно фиксировать свою любимую нотацию, в которой уже что-то типовое описано (как наработки с других проектов или часть поставки ERP-системы), в уставе проекта.

Нужно отметить, что у заказчика могут быть свои требования к нотации – например, если уже до этого были проекты с описанием бизнес-процессов или параллельно идут несколько разных проектов, где производится описание в какой-то нотации, логично, что она должна быть одна и та же, несмотря на, возможно, разных исполнителей.

Инструментарий для подготовки описания бизнес-процессов схемами может быть как специализированный, под какую-то определенную нотацию, так и универсальный, который позволяет делать схемы в любой из приведенных выше в списке нотаций.

Важно учитывать, что со схемами предстоит работать не только на экране в инструменте проектирования (где может быть удобная навигация между подчиненными и родительскими схемами), но и на бумаге (как часть отчета).

Цветная схема требует цветной печати, либо нужно подбирать цвета текста и фона графических элементов для читабельности и возможности различения полутонов при черно-белой печати. Также важным является расположение элементов: не всегда простой задачей будет вписаться в лист А4 для сохранения приемлемого масштаба и размера шрифтов текста. Нужно следовать правилу минимизации элементов функций (действий) на одной схеме до 5–8 штук. Иначе за счет элементов других типов схема превратится в паутину соединительных линий с потерей наглядности. Минимизация элементов облегчает восприятие, нужно выносить группируемые элементы в подпроцессы, использовать переходы между схемами. Такое требование актуально для схем любых нотаций.

Далее рассмотрим некоторые из приведенных выше нотаций.

Функциональные блок-схемы

Блок-схемы изначально предназначены для описания алгоритмов и программ, организованных согласно структурному подходу, и не могут отразить, например, алгоритм, который реализуется во взаимодействии абстракций при объектно-ориентированном подходе. Для объектно-ориентированного программирования (ООП) отлично подходят схемы UML. Но для логики описания бизнес-процессов таких схем достаточно, там нет элементов ООП.

Для описания бизнес-процессов к блок-схемам добавляются элементы (отступление от ГОСТ), описывающие функции (элементы дорожек для горизонтальных и вертикальных разделений, например, по ролям) и какие-то графические артефакты рабочего процесса (иконки, оборудование, здания, серверы и прочие). Цветом можно выделять действия вне системы. Получается расширенная нотация функциональных блок-схем для описания бизнес-процессов.

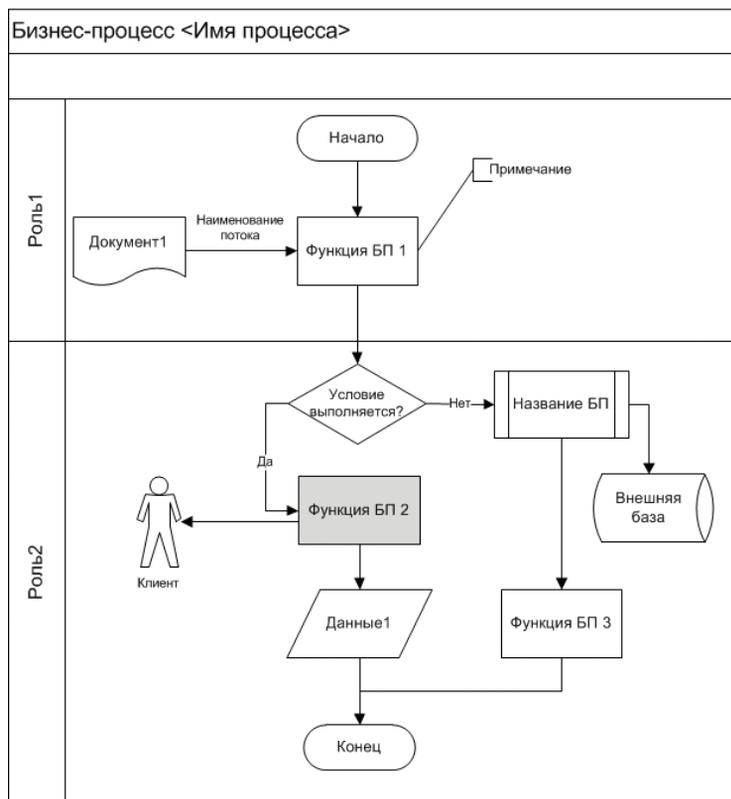


Рис. 6.3. Пример функциональной блок-схемы

Функциональные блок-схемы используются для описания бизнес-процессов в продуктах «1С». Ниже приведен скриншот из 1С:ERP процесса согласования документов. В такой нотации можно также описывать любые бизнес-процессы в процессе анализа компании заказчика, создавая файлы графических схем (.grs) в «1С:Предприятии 8» через команду **Файл – Новый – Графическая схема**. То есть платформа «1С» предоставляет свой инструментарий для графического описания бизнес-процессов в нотации блок-схем.

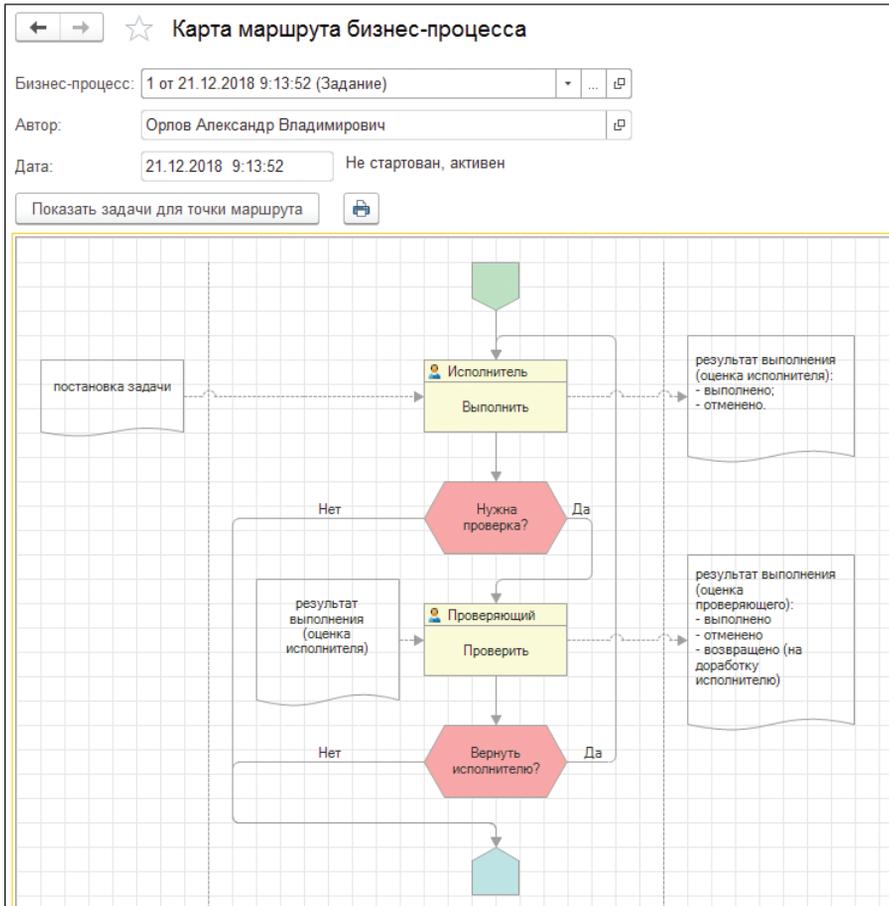
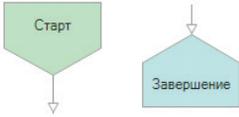
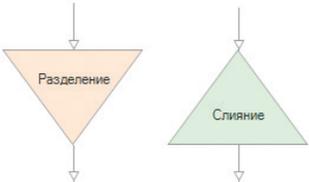
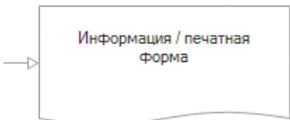
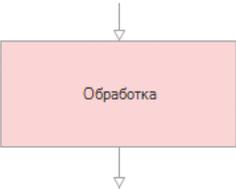


Рис. 6.4. Карта маршрута согласования документа в 1С:ERP

Рассмотрим возможные элементы схемы бизнес-процессов «1С» подробнее.

Элемент схемы	Графическое представление	Описание
Точка начала/ конца		Начало и завершение бизнес-процесса, является обязательным
Действие		Соответствует конкретному действию конкретного исполнителя
Условие		Условие, у которого может быть только 2 состояния (да/нет или с иным названием от контекста)
Выбор варианта		Возможность выбора дальнейших действий исходя из проверки условия, у которого может быть более двух состояний
Точки разделения и слияния		Соответствуют моментам разделения и слияния выполнения разных действий параллельно и независимо друг от друга
Информация		Блок для входящей и выходящей информации, документов и печатных форм

Элемент схемы	Графическое представление	Описание
Вложенный бизнес- процесс		Соответствует выполнению другого бизнес-процесса, который является составной частью текущего и рисуется отдельной схемой
Обработка		Отвечает за выполнение определенного алгоритма системой

IDEFO

Нотация **IDEFO**, как правило, используется для описания бизнес-процессов верхнего уровня компании. Она позволяет просто и наглядно изобразить состав основных процессов, выходы бизнес-процессов, изображающих заданный результат их выполнения, и входы, показывающие, какие ресурсы нужны для получения результата. Декомпозиция функциональных блоков уместна до 3–6-го уровня глубины.



Рис. 6.5. Нотация IDEFO

В зависимости от того, в какую грань функционального блока входят стрелки или из какой грани они выходят, стрелки делятся на четыре вида:

- **Входы функций** (входят в левую грань функционального блока) – информационные ресурсы, которые используются при выполнении функции для достижения ожидаемого результата.
- **Выходы функций** (выходят из правой грани функционального блока) – результат выполнения сотрудниками (пользователями) действий в соответствии с выполняемой функцией. Выход функции может быть использован как вход или управление для другой функции или же направлен на обеспечение внешней связи информационной системы с окружающей средой. Например, функция анализа какого-то вида деятельности заключается в получении информации, которая используется для принятия управленческого решения вне информационной системы.
- **Управление** (входят в верхнюю грань функционального блока) – это элемент модели, который привязывает все действия к системе регламентов компании, четко обозначая, какие правила и требования должны быть соблюдены в процессе выполнения функции.
- **Исполнитель функции** (входят в нижнюю грань функционального блока) – изображают ресурсы, необходимые для выполнения функции, но не изменяющиеся в процессе работы (например, сотрудники с определенными правами из профилей пользователя).

Функциональная модель 1С:ERP поставляется в составе конфигурации «1С:Система проектирования прикладных решений» (СППР) в виде набора схем стандарта IDEF0, что очень полезно к изучению для освоения возможностей прикладного решения, принципов и методологии взаимодействия отдельных подсистем.

1.7.4 Бюджетирование

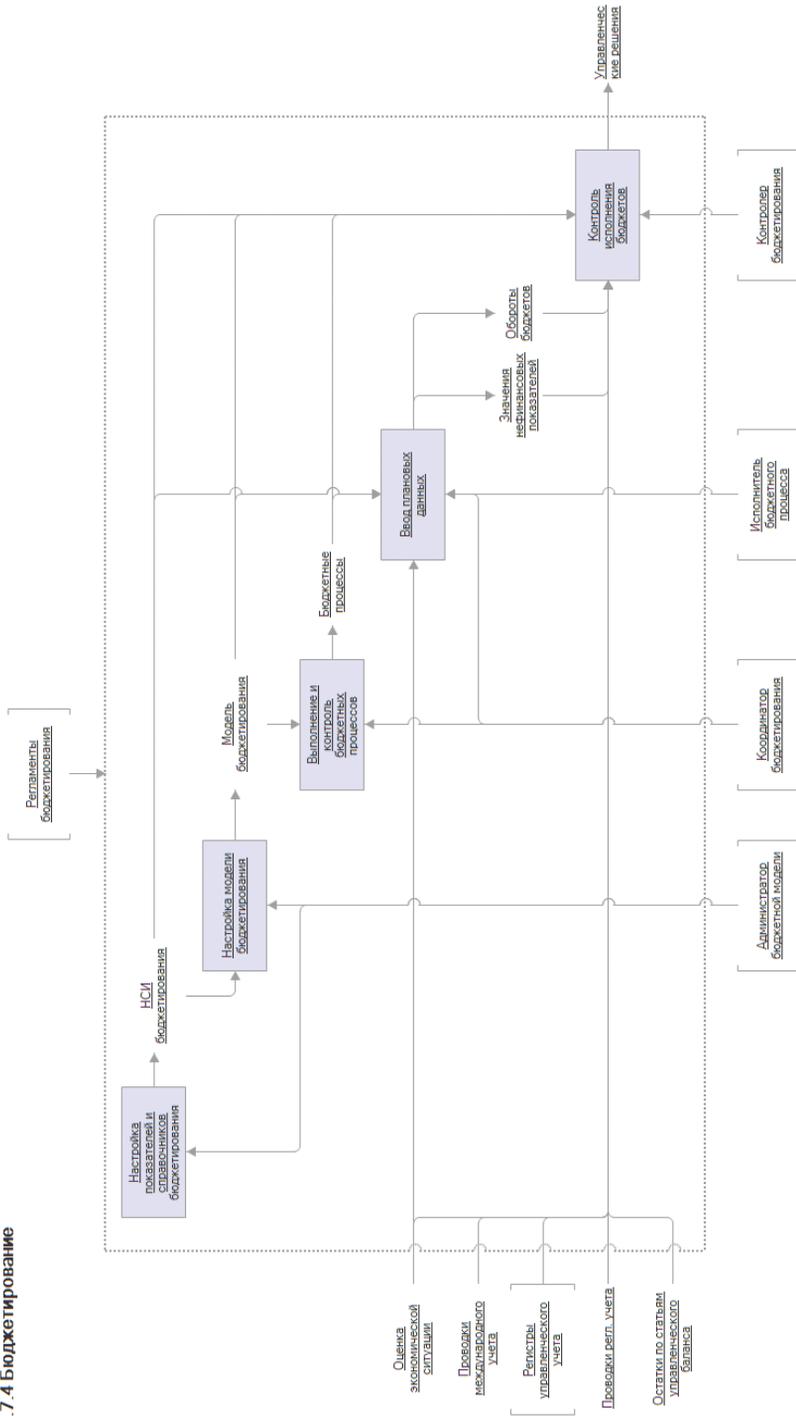


Рис. 6.6. Схема IDEFO из 1С:СПП описания бюджетирования в 1С:ERP

«1С:Система проектирования прикладных решений» предназначена для проектирования прикладных решений (конфигураций) на платформе «1С:Предприятие 8» и ведения технической документации проекта. СППР может быть использована как инструмент для проектирования новых информационных систем, разрабатываемых в среде «1С:Предприятие 8», а также для описания, документирования, организации разработки, тестирования (включая автотесты), сбора и исправления ошибок при внедрении 1С:ERP или иных конфигураций.

Партнеры, разрабатывающие отраслевые решения на базе 1С:ERP, должны оформлять модель своих решений в 1С:СППР. Такие модели по отраслевым решениям публикуются в сервисе «1С:Облачная карта прикладных решений»: <http://platform.demo.1c.ru/solutionscloud>. Можно самостоятельно изучить различные отраслевые решения и исходную модель от 1С:ERP.

Подробная информация по СППР представлена на странице <http://v8.1c.ru/model/>. Мы еще вернемся к инструментарию 1С:СППР в главе 9. Пока же примем факт, что вместе с 1С:ERP поставляется 1С:СППР с готовой функциональной моделью в нотации IDEF0, которую можно дополнить необходимыми блоками, возникающими в ходе ведения проекта внедрения ERP-системы.

ЕРС

Для описания бизнес-процессов нижнего (операционного) уровня можно использовать нотацию **ЕРС** (Event-Driven Process Chain). ЕРС-метод был разработан Августом-Вильгельмом Шеером в рамках работ над созданием ARIS в начале 1990-х годов.

ARIS (Architecture of Integrated Information Systems) – методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций.

Нотация используется многими организациями для моделирования, анализа и реорганизации бизнес-процессов. Пришла на смену IDEF3 и хорошо сочетается с IDEF0, как раскрытие функциональных блоков. Ключевая особенность ЕРС-диаграмм – описание бизнес-процесса как последовательности чередующихся событий и функций.

Основные графические элементы диаграммы EPC:

- функции;
- события;
- роли или организационные единицы, ответственные за исполнение функций;
- информационные или материальные объекты, которые используются при выполнении функций;
- коннекторы (AND, OR, XOR).

Основные правила при моделировании просты, но между тем строгие (дисциплинируют и вынуждают выяснять детали):

- каждая функция инициируется событием и завершается также событием;
- в каждую функцию не может входить более одной стрелки, «запускающей» выполнение функции;
- из каждой функции может выходить не более одной стрелки, описывающей завершение выполнения функций.

При моделировании процессов консультант очень часто сталкивается с ситуацией, когда одно событие в рамках процесса может инициировать выполнение одновременно нескольких функций, и наоборот – функция может быть результатом нескольких событий. Каким образом отразить это в диаграмме модели? Такие логические разветвления в потоке процесса отображаются при помощи операторов:

- логическое **И** (AND) – все ветки должны отработать;
- логическое **ИЛИ** (OR) – любая одна, несколько или все ветки;
- логическое **взаимоисключающее ИЛИ** (XOR) – только одна из веток отработает.

При моделировании процесса стоит придерживаться определенных правил расположения графических элементов на диаграмме:

- графические элементы процесса (события и функции) следует располагать сверху вниз;
- графические элементы, отображающие исполнителей функций (сотрудников и подразделений), следует располагать справа от функций;
- документы, используемые при выполнении функций, располагаются слева от функций, а формируемые в результате выполнения функций – располагаются справа от функций (вариант – тоже слева, чтобы не пересекаться с исполнителями, но тогда нужно рисовать у стрелки начало-конец).

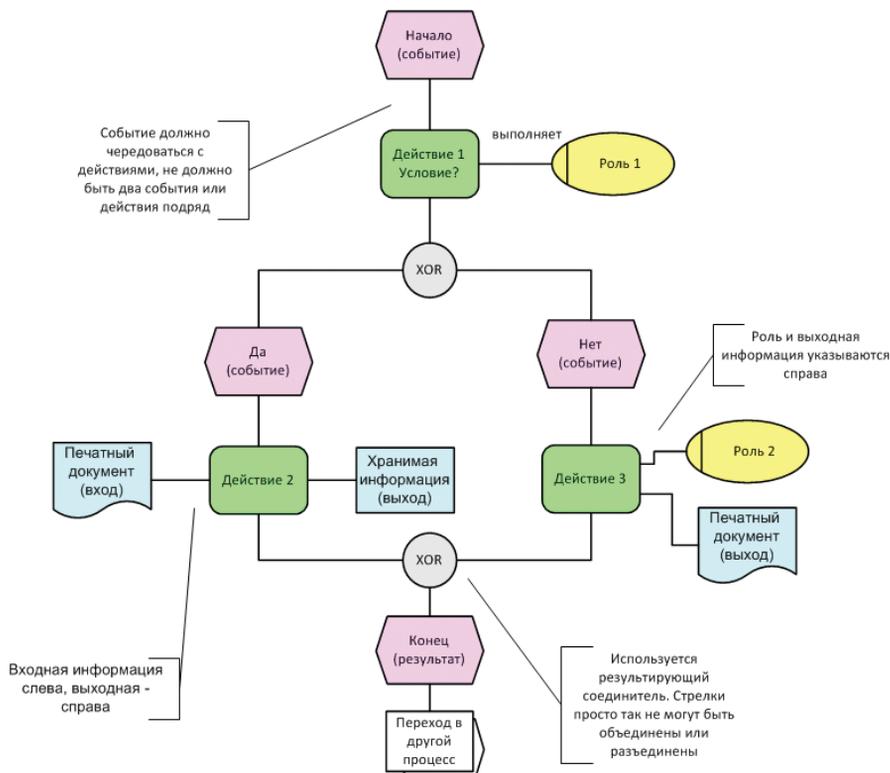


Рис. 6.7. Элементы графической схемы в «1С»

Такие требования последовательности изображения элементов схемы затрудняют возможность уместить схему на печатном листе формата А4, схема ЕРС склонна к «распуханию», нужно аккуратно ее декомпонировать на подпроцессы.

Правила построения схем ЕРС:

- если в рамках бизнес-процесса формируются или поступают какие-либо документы (учетные документы), они должны быть отражены на схеме в обязательном порядке;
- функции на схемах следует именовать отглагольными существительными, например «Регистрация заявки»;
- использование отглагольных существительных для других элементов, кроме функций и бизнес-процессов, запрещено;
- каждой функции и бизнес-процессу следует давать код – например, код функции может быть формата **<Код бизнес-процесса>. <Код функции>**;
- в схемах допускается использование меток на естественном языке. Метками можно пояснять соединительные линии (стрелки связи элементов модели). Стрелки показывают, как функции системы связаны между собой, как они обмениваются данными;
- объект схемы не может быть замкнут сам на себя;
- в модели разрешены циклы между несколькими объектами;
- разрешено только одно соединение между двумя объектами (все деления и слияния через коннекторы);
- одна функция (действие, работа) может иметь несколько последователей и несколько предшественников. Для отображения логических условий пересечения связей последовательности следует использовать специальные элементы **И**, **ИЛИ**, **XOR**;

- по возможности следует декомпозировать функции бизнес-процесса до такого уровня, чтобы у одной функции не было нескольких исполнителей – ролей;
- можно перечислять через запятую названия сущностей на схеме внутри одного элемента, например: «Движения по складу, расчеты по поставщику, проводки по БУ».

BPMN

Нотация **BPMN** (Business Process Model and Notation) интересна своей простотой и совместимостью между разными программными средствами, т. к. диаграммы, нарисованные в одной программе, можно открывать в другой, формат файла унифицирован.

Принцип составления схем не отличается от рассмотренных выше приемов, отличается графическая нотация. Можно провести параллели с другими нотациями:

- есть дорожки для ролей, как в функциональных блок-схемах;
- события начала и конца, как в блок-схемах;
- логические разделители по **И**, **ИЛИ** и **XOR**, как в EPC (и IDEF3);
- группировка в подпроцессы – как везде;
- графические артефакты для обозначения баз данных, документов, информации – как в большинстве других нотаций.

Таким образом, нотация соответствует функциональным блок-схемам, EPC, IDEF3 и может их заменять, сочетаясь с IDEF0 для детального описания функциональных блоков.

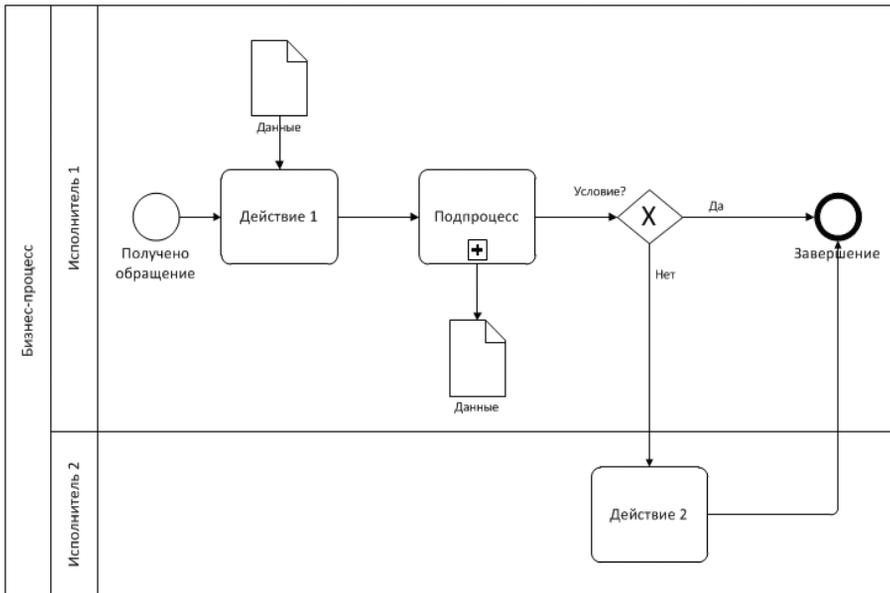


Рис. 6.8. Пример схемы в нотации BPMN

Подробное описание нотации приведено на сайте <http://www.bpmn.org/>.

Особенности описания бизнес-процессов

Рассмотрим некоторые определения для полноты понимания описания бизнес-процессов:

- **Процесс** – набор взаимосвязанных и взаимодействующих операций (действий), которые преобразуют входы в выходы.
- **Бизнес-процесс** – периодически повторяющаяся часть деятельности предприятия, отвечающая следующим критериям:
 - имеется цель деятельности;
 - известны условия начала деятельности;
 - постоянные участники и ожидаемые результаты;
 - известны условия завершения.

Описание процессов должно отражать не только отдельные процессы, но также взаимосвязи и взаимодействия между ними. Процессы вместе с взаимосвязями и взаимодействиями образуют **сеть бизнес-процессов** организации.

С чего начать описание бизнес-процессов? Для начала нужно иметь информацию для анализа. Выше рассматривалось, как готовить отчет и что запрашивать, это необходимые условия начала работ над схемами.

Шаги подготовки схем описания бизнес-процессов:

- Готовим общую схему. Ставим точку начала и конца и заполняем шаги между ними.
- Детализируем схему в диаграмму (то есть в итоговую схему в требуемой нотации), попутно разбивая на подсхемы, и понимаем, что не все понимаем сами.
- Просим дополнительную информацию для анализа.
- Рисуем дальше диаграммы. Могут появиться альтернативные окончания процессов, все уточняется по ходу наращивания детализации.
- Гнаться за максимальной детализацией на схеме смысла нет, схема может превратиться в нечитабельную паутину из стрелок, используем группировку в подпроцессы.
- При завершении диаграмм начинаем описывать бизнес-процессы словами. Описывать диаграммы по неясным из схем вещам. Может быть схема совсем без описания, когда все ясно без пояснений. Дословное описание схемы делать не нужно, это не имеет смысла, только перегрузит итоговый документ.

Заблуждением будет использовать подход: «А, мне и так все понятно, можно сразу начинать использовать систему или делать ТЗ на доработку системы». Раз все понятно, то можно потратить время и нарисовать! А в процессе рисования схемы выяснится удивительным образом, что «понятно», оказывается, было далеко не все, сразу пойдут вопросы:

- Что на выходе?
- Что на входе?
- Кто это делает?
- Исполнитель или система уже располагает информацией для этого действия на этом шаге?
- Что является причиной (управленческим воздействием) для этого действия? И так далее.

Всегда помним о балансовом правиле: если где-то что-то получается на выходе, то в другом месте оно потребляется на входе. Это касается не только товаров и денежных потоков или закрытия бухгалтерских счетов по дебету и кредиту, но и документооборота в бизнес-процессе. Если какие-то исходящие документы нигде потом не нужны – это подозрительно. И, наоборот, если на вход подается что-то, оно где-то должно создаваться или явно обозначаться как внешняя информация (для схемы и системы учета) с понятным источником возникновения вне контура рассматриваемых процессов. Это процессный подход к анализу функций: мало описать конкретную функцию исполнителя, нужно понимать ее место среди всего бизнес-процесса с общими данными/документами на входах и выходах.

Обязательно ли нужно описывать бизнес-процессы «как есть»? С одной стороны, для проекта внедрения нужна целевая схема «как будет», и можно было бы сразу рисовать ее. Но если предполагается реинжиниринг процессов, то для понимания заказчиком процесса перехода из состояния «было» в «будет» схемы «как есть» будут необходимы. Также отметим, что со схем «как есть» проще начать проектирование и перейти потом к подготовке целевых схем за счет моделирования на готовых схемах новых их состояний (перетасовкой элементов).

Бизнес-процессы «как есть» могут не описываться, в случае если внедрение программного продукта сопровождается использованием заранее выбранных и утвержденных процессов (или стартапа, у которого еще нет исторических процессов). Компания принимает схемы процессов от системы и переходит на них в процессе внедрения.

В случае автоматизации бизнес-процессов компании переход от схем «как есть» в схемы «как будет» еще наглядно показывает автоматизируемые действия, которые выполняются вручную и вне учетных систем в данный момент (до внедрения ERP-системы). Такая визуальная разница перехода к автоматизации в том числе является критерием обоснования затрат на автоматизацию для заказчика и спонсора проекта.

Рассмотрим критерии значимости (то есть нужно ли их описывать детально) бизнес-процессов для внедрения КИС:

- Процессы должны находить свое отражение в системе – нет смысла рассматривать процессы, которые ведутся вне системы автоматизации целиком (например, как работает пропускная система на предприятии), если это не цель автоматизации.
- Бизнес-процессы, которые еще не представлены в системе (в системе нет такого функционала для процесса), но сам бизнес-процесс планируется автоматизировать.
- Бизнес-процессы представлены в системе, но имеют отраслевую специфику или специфику конкретного предприятия – скорее всего, существующая функциональность системы не будет покрывать всех аспектов бизнеса компании из-за отраслевой специфики.

Цели описания бизнес процессов – это найти ответы на вопросы:

- Какая цель данного бизнес-процесса, его место и роль в общих задачах (процессах) компании?

- Какие функции и в какой последовательности выполняются в рамках процесса?
- Кто является ответственным за выполнение каждой из функций?
- Условия начала выполнения каждой функции, какие события инициализируют выполнения функции?
- Документы и данные, необходимые для выполнения бизнес-процесса, функций и их источники?
- Документы, создаваемые в результате выполнения бизнес-процесса, и их получатели?
- Какие программные и аппаратные комплексы задействованы, в каких операциях?
- Что является результатом работы, какие проблемы возникают при выполнении работы?

Важно максимально полно выявить информацию о процессах, протекающих в компании заказчика. Проверить список ролей и реально работающих сотрудников: всем ли назначена какая-то роль в системе, есть ли эти роли в схемах бизнес-процессов. По этому же списку сотрудников потом можно будет планировать график обучения и аттестации, чтобы все были охвачены и задействованы во внедрении, а не «случайно забыты».

Необходимо в фазе анализа сразу выявить требования к разграничениям прав доступа пользователей к данным и функциональности КИС:

- часто про это забывается, а это нужно проектировать, нужно список ролей знать для оценки трудозатрат на настройку, подготовку инструкций пользователей и презентаций для обучения;
- возможно, сразу заложить в проекте внедрения работы на какие-то специализированные рабочие места и отчеты, если показ прототипа системы выявит недостатки рабочих мест по конкретным ролям;

- зафиксировать требования и выделить время на разграничение доступа на уровне записей и полей (RLS – Record Level Security).

Не нужно из-за кажущейся простоты («а, это мелочи, потом настроим») откладывать формализацию доступа по ролям на конец проекта – лучше, когда «потом» наступит, всем этим управлять в обычном режиме, как плановыми работами, а не в авральном, когда уже все заходят в систему, а понимания, кому какие права выдать, нет вообще.

В завершение раздела про бизнес-процессы и важность их понимания до начала работ по внедрению – небольшая история про «забытый» отдел. Автор лично встречал ситуацию почти по Михаилу Задорнову с его рассказом «Девятый вагон» и фразой: «Мой вагон пустой!» А именно: идет вовсю запуск системы, перевод логистики со старой системы в ERP, все уже спроектировано, пользователи обучены, отказ от прошлых систем, полет нормальный. Через неделю выясняется, что ключевые сотрудники заказчика забыли включить в границы проекта целый бизнес-процесс группы из 5 сотрудников, которые все еще продолжали работать в старой системе, сидели в отдельной комнате, ажиотаж внедрения не видели, обучение не проходили, в ERP их никто не переводил. И забеспокоились они сами, когда у них закрылся список отработки заявок в старой системе, а новые перестали появляться, стало пусто и нечего обрабатывать.

Вывод: нужно тщательно выяснять все бизнес-процессы по всем сотрудникам, явным образом отметить по ним график обучения и перевода их в систему или отметить, что они вне проекта. Иначе такое «ой, мы про них забыли» – станет реальностью.

Прототип и его демонстрация

Внедрение ERP-системы отличается от разработки программного обеспечения под заказ с нуля хотя бы тем, что изначально система уже есть и, вообще говоря, готова к работе без каких либо доработок, в ней реализуется замкнутый бизнес-цикл учета и ведения бизнес-процессов. Другое дело, насколько эта готовая система покрывает потребности компании заказчика и требования конкретных ключевых пользователей. В главе 2 рассматривался fit-gap анализ ERP-систем в процессе участия в тендере. На том этапе проекта не было выделено достаточно времени

для полноценного прототипирования и моделирования бизнес-процессов и ведения учета в системе для покрытия всех требований, оценка давалась больше экспертно с незначительной проверкой гипотез на существующей функциональности системы. А вот на фазе анализа и концептуального проектирования для развертывания и начала настройки прототипа системы – самое время. Результат этой работы покажет достоверный список требуемых модификаций и концепцию по ним, достаточный для оценки бюджета и сроков проекта. Сам прототип является отличным инструментом для коммуникации между специалистами исполнителя и заказчика, так как разговоры о том, как все будет, будут проходить на живой системе, что сразу минимизирует риски высказываний пользователей типа: «А мы думали, что будет совсем не так!»

Проведем аналогию с известным высказыванием. Однажды великого итальянского скульптора и живописца Микеланджело Буонаротти спросили «Как вы делаете свои скульптуры?» – на что он ответил: «Я беру камень и отсекаю все лишнее». С внедрением ERP похоже, но ровно наоборот. Можно перефразировать: «Нужно взять готовую ERP-систему и дописать в ней все недостающее для конкретной компании». То есть нужно мысленно представить идеальный итоговый результат из текущей ERP-системы и описать (а потом и сделать) недостающее функции до идеала в ней.

Прототип будущей корпоративной информационной системы позволяет заинтересованным сторонам экспериментировать с моделью ERP-системы, а не ограничиваться обсуждением абстрактных представлений о своих требованиях и их возможной реализации. Прототипирование поддерживает концепцию последовательного уточнения в итеративных циклах создания макетов, проверки гипотез, проведения экспериментов конечным пользователем совместно с консультантом, формирования отзывов и пересмотра прототипа. После проведения достаточного числа циклов обратной связи через демонстрацию прототипа в рабочих встречах (один на один с ключевыми пользователями или в групповых обсуждениях) требования уточняются или формируются новые. Этих уточненных требований достаточно для перехода к фазам дизайна архитектуры системы и разработки. Сбор требований при демонстрации

прототипа – это отличный инструмент вовлечения заинтересованных сторон. Время, потраченное на этом этапе, окупится за счет сокращения в дальнейшем количества лишней работы. Помним график из раздела 5.1 жизненного цикла проекта внедрения: такая проверка требований на прочность на демонстрации прототипа позволяет гибко влиять на проект на начальной стадии с минимумом усилий, т. к. многое можно еще пересмотреть, уточнить приоритеты и очередность работ, пока нет жесткого утверждения границ проекта и разработанной функциональности.

Интересная статистика. На исправление логической ошибки в требованиях, выявленной на стадии работы над требованиями, тратится в среднем 30 минут, а на исправление уже во внедряемой функциональности той же ошибки, выявленной в ходе тестирования, необходимо от 4 до 16 часов. Поэтому польза от уточнений требований на прототипе несомненна.

Обязательно нужно фиксировать и максимально полно (исходя из понимания на момент прототипирования и обсуждения с пользователями) описывать выявленные потребные модификации системы в таблице в сводном списке по разделам (подсистемам). Такие полезные идеи, как это все можно сделать, возникающие по ходу анализа и показа прототипа, могут забыться со временем или из-за ротации в команде стать утерянными, тогда позже придется «продумывать эту идею еще раз». Полученный список модификаций будет основой для уточнения оценок и планирования сроков разработки.

Если фаза анализа не отделяется от фазы дизайна, то такие необходимые модификации сразу проектируются в достаточном для утверждения заказчиком и начала разработки виде.

Важным моментом при показе прототипа (и вообще фазы анализа) является уточнение требований к переносу начальных данных, исходя из возможностей системы по инициализации начальных данных и того, в каком виде эти начальные данные можно получить из текущих учетных систем заказчика. Задачи на подготовку начальных данных требуют внимания и планирования: для исполнителя эти задачи могут казаться находящимися за рамками проекта, т.к. начальные данные находятся

в распоряжении заказчика, но вот сроки, когда эти данные предоставят, влияют на успешность и последовательность работ уже исполнителя. Потому лучше явно и заранее прописать работы по подготовке начальных данных и сроки по ним с ответственностью заказчика за его задачи. На прототипе нужно продемонстрировать возможности по вводу начальных данных для выработки у заказчика понимания, что потребуется системе для инициализации работы, и анализа, какие данные и как получить для этого.

Если исполнитель может как-то оптимизировать (автоматизировать) подготовку начальных данных (выгрузки из текущих систем, мэппинг по таблицам соответствий, агрегирование), то эти возможности нужно озвучить и предложить. Стоимость таких обработок и удорожание на них бюджета проекта могут быть незначительными на фоне трудозатрат (и оплаты труда) сотрудников заказчика. То есть при явной экономии сроков и сил сотрудников заказчика лучше на работы по автоматизации переноса начальных данных привлекать специалистов исполнителя, а не экономить по принципу «ну, это мы и сами подготовим».

А что делать, если в системе нет какой-то функциональности, но для целостности картины у ключевых сотрудников заказчика при моделировании бизнес-процессов в прототипе она нужна? Недостающие части (которые предстоит создать или доделать на следующих фазах проекта) могут быть обозначены схематически в виде упрощенных эскизов форм рабочих мест и макетов отчетов, показанных в виде слайдов презентации, или даже рисунков на флипчарте или маркерной доске, рисуемых в процессе демонстрации прототипа. То есть уместны фразы типа: «Тут мы сделаем такое вот рабочее место, – рисуем форму или показываем слайд, – а тут будет вызов этого отчета» (открываем файл Excel) в процессе показа прототипа системы и его обсуждения.

В процессе дизайна форм для работы с ними пользователей можно использовать прием наглядного макетирования. Такой прием практикуется в дизайне интерфейсов и проверке их юзабилити (от англ. usability – удобство и простота использования). При разработке и дора-

ботке ERP-систем есть средства визуального создания макета интерфейсных форм в самой системе без дополнительной логики (кнопки не жмутся, алгоритмов нет), но интерфейсно все наглядно.

Полноценное проектирование новых интерфейсов будет на следующей фазе (дизайна), а на фазе анализа детализация может быть ограничена достаточностью для обсуждаемых вопросов. В качестве макетов новых отчетов, которых нет в системе, можно брать текущие Excel-файлы отчетов, что используются у заказчика. Если по внешнему виду формы отчета есть сомнения, что итоговый отчет будет точно таким же, как был в файлах, а полноты понимания нового дизайна еще нет, то лучше такие моменты явно отразить в описании концепции: что итоговые отчетные формы будут согласовываться на фазе дизайна архитектуры системы.

При проведении демонстраций прототипа системы и возможностей функциональности можно использовать готовые презентации от поставщика ERP-системы. Так, по 1С:ERP довольно много готовых презентаций от самих разработчиков, доступных на сайте ИТС (Технологическая поддержка прикладных решений/Методическая поддержка «1С:Предприятия 8»/ERP Управление предприятием/Доклады и презентации), часть выложена в открытом доступе на странице описания конфигурации <http://v8.1c.ru/erp/>.

На фазе дизайна архитектуры системы финализируются решения по НСИ и используемым опциям настроек системы. В прототипе все эти настройки находят свое отражение уже в конечном для будущей рабочей версии виде. Ниже представлен скриншот 1С:ERP со списком пунктов из раздела **НСИ и администрирование**. К настройкам относятся:

- базовая НСИ (организации, структура предприятия, банковские счета, контрагенты, номенклатура и прочие);
- настройка опций разделов (подсистем) – позволяет включать или отключать подсистемы и особенности функций внутри подсистем;

- настройки интеграции с другими системами, например с «1С:Документооборотом»;
- различные общие настройки по сервисным возможностям и подключаемому оборудованию;
- подключаемые дополнительные отчеты и обработки.

<p>Начальное заполнение</p> <ul style="list-style-type: none"> Начальное заполнение Корректировка регистров <p>НСИ</p> <ul style="list-style-type: none"> Номенклатура Классификаторы номенклатуры Партнеры Организации Структура предприятия Склады и магазины Банковские счета Кассы предприятия Кассы КММ Проекты Графики работы Лица с правом подписи Физические лица <p>Администрирование</p> <ul style="list-style-type: none"> Обслуживание Общие настройки Интерфейс Настройки пользователей и прав Интернет-поддержка и сервисы Организатор Настройки работы с файлами Печатные формы, отчеты и обработки РМК и оборудование 1С:Маркировка 1С:Номенклатура Настройки интеграции с ВетИС Настройки интеграции с ИС МОТП 	<p>Настройка НСИ и разделов</p> <ul style="list-style-type: none"> Предприятие Номенклатура Бюджетирование и планирование CRM и маркетинг Продажи Закупки Склад и доставка Производство Кадры Зарплата Казначейство и взаиморасчеты Финансовый результат и контроллинг Учет НДС и ВЭД Внеоборотные активы Регламентированный учет Международный финансовый учет <p>Настройка интеграции</p> <ul style="list-style-type: none"> 1С: Документооборот Система проектирования прикладных решений Синхронизация данных Обмен электронными документами Обмен с сайтом <p>См. также</p> <ul style="list-style-type: none"> Валюты Банки Страны мира Производственные календари 	<p>Отчеты</p> <ul style="list-style-type: none"> Дополнительные отчеты <p>Сервис</p> <ul style="list-style-type: none"> Дополнительные обработки Обращение в техническую поддержку фирмы "1С" Персональные настройки Перенос данных из БП Помощник замены ставки НДС на 20% Свертка информационной базы
--	--	--

Рис. 6.10. Пункты раздела «НСИ и администрирование» в 1С:ERP

Как мы видим, настроек большое количество, и правильное, подходящее компании заказчика, их сочетание – это залог успеха работы системы на многие годы ее использования заказчиком. Особенно важными являются настройки номенклатуры, учета затрат и расчета себестоимости.

Например, заказчик изначально хочет детализацию учета себестоимости выпускаемой продукции до последнего винтика (хорошее, в общем-то, желание максимальной точности всех сумм). Но на практике так настроенная система может не заработать, т. к. по факту такой детальной информации нет на уровне организации самого бизнес-процесса в компании: отслеживание и обособление по продукции всех затрат затруднительно для сотрудников на местах. Поэтому в процессе анализа прототипа на реальном бизнес-процессе исполнителем и заказчиком совместно принимается решение по упрощению схемы учета за счет распределения незначимых статей расходов без явного их отнесения на конкретный экземпляр продукции.

При этом систему можно настроить на изначальные требования заказчика по максимальной детализации, но проблемы без прототипирования выявятся только на уровне опытной эксплуатации в момент начала работы пользователей, что «откатит» проект обратно в перенастройку и согласование изменений в настройках. Этого можно избежать за счет моделирования в прототипе на малых наборах данных бизнес-процессов компании с участием ключевых сотрудников заказчика.

Начав настраивать прототип на фазах анализа и дизайна, далее остается поддерживать его настройки в актуальном состоянии по мере доработок системы. В результате прототип станет тестовой версией, в него можно провести тестовый перенос начальных данных. Так постепенно прототип перерастет в полноценную настроенную версию с реальными настройками, по которым в чистой базе на введенных на дату запуска начальных данных будет продолжаться опытная, а затем и промышленная эксплуатация.

Глава 7. Как оценить срок и бюджет проекта

Сроки и бюджет проекта

С первого дня проекта (еще на этапе предпроектных работ) встают два вопроса: «Сколько будет стоить проект внедрения ERP-системы?» и «Как быстро все это будет сделано?» И на эти вопросы нужно отвечать – сначала примерной оценкой, позже – более точной, а далее регулярно уточнять оценки и состояние бюджета проекта со сроками на всем его протяжении. Рассмотрим в этой главе вопросы оценок бюджета и сроков проекта.

Бюджет проекта (смета проекта) включает в себя все денежные средства, выделяемые для исполнения проекта.

Срок проекта – это календарное время, за которое проект (и его фазы) будут выполнены.

Определение бюджета – процесс консолидации оценочных стоимостей отдельных операций или пакетов работ для создания авторизованного базового плана по стоимости всего проекта.

В результате экспресс-анализа, а позже и полноценного анализа и концептуального проектирования, создается иерархическая структура работ (ИСР), содержащая весь перечень того, что нужно сделать для успешного завершения проекта. Остается оценить стоимость и сроки каждого пункта ИСР и получить итоговые оценки бюджета и сроков.

Проект может проходить при следующих подходах к бюджету:

- Бюджет фиксирован (fix price) – заранее утверждается стоимость проекта, что требует серьезного и скрупулезного подхода к оценкам: учесть все, заложить риски.
- Оплата по факту (time & material) – примерная оценка на весь проект и детальные оценки на ближайшие работы с фиксированием реальных трудозатрат по результатам сделанного.
- Микс из двух подходов на разные блоки внутри проекта.

Задача точной оценки сроков и трудозатрат по проекту является довольно сложной (особенно для подхода fix price) и, к сожалению, не имеет однозначного формализованного решения, которое давало бы 100-процентный результат.

На разных стадиях проекта можно выделить четыре типа оценок бюджета:

- **грубый порядок величины (row order of magnitude)** – стоимостные ожидания проекта, находящегося на фазе замысла или идеи;
- **порядок величины (order of magnitude)** – предположения о стоимости проекта, рассчитанные на фазе инициализации проекта, отражаемые в первичном коммерческом предложении;
- **бюджетная оценка (budget estimate)** – оценка стоимости проекта, полученная на основе данных, предоставленных заказчиком после анализа исполнителем;
- **точная оценка (definitive estimate)** – оценка стоимости, включаемая в бюджет при определении окончательной плановой стоимости проекта перед переходом к фазе разработки.

Таким образом, на начальной фазе проекта (инициализация проекта) дается первичная оценка «порядок величины». Точность такой оценки колеблется в диапазоне от -25 % до 75 %. То есть проект может оказаться дешевле на 1/4 или дороже на 3/4.

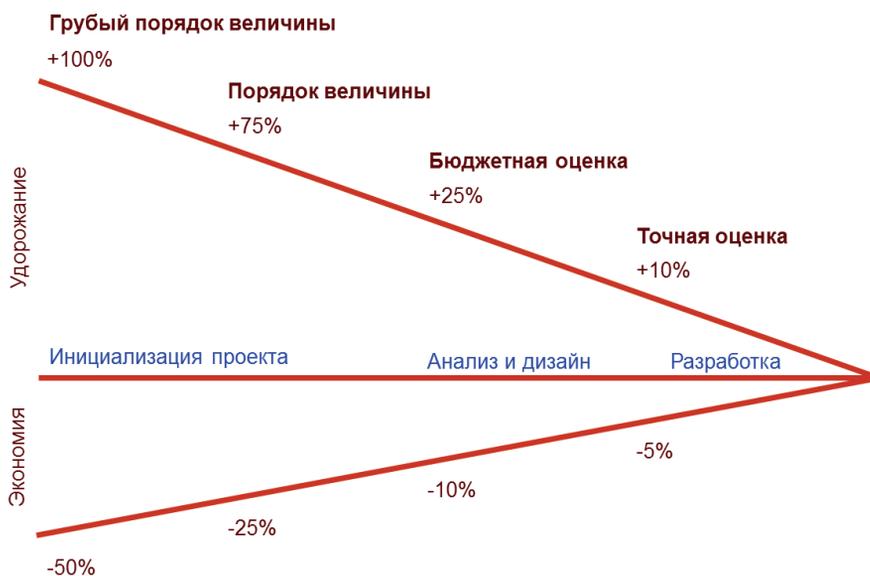


Рис. 7.1. Диапазоны точности оценок стоимости проекта

При завершении планирования после фазы анализа точность оценки бюджета вырастает до бюджетной оценки с точностью от -10 % до 25 %, после дизайна архитектуры системы оценка уточняется и колеблется уже в диапазоне от -5 % до 10 %.

Можно ли на начальных стадиях проекта «угадать» бюджет проекта с точностью до 10–15 %? При большой практике типовых внедрений у исполнителя, когда не ожидаются какие-то значительные особенности в бизнес-процессах компании заказчика, такое возможно: можно провести аналогию с прошлыми проектами, похожими на этот. Но в какой-то момент у команды проекта случается их первый проект, и аналогов еще нет или проекты значительно различаются. Какие методы и инструменты для оценок бюджета и сроков доступны руководителю проекта?

Используемый инструментарий для оценки бюджета проекта:

- нет инструментария – все в голове и на бумагах. Для крупных проектов для детальной оценки такой подход не годится, но как быстрая примерная оценка «на лету» применяется;

- электронные таблицы – позволяют оценить таблицу ИСР по трудозатратам, но планирование сроков и рисование диаграммы Ганта осуществляются вручную;
- специальное программное обеспечение для проектной деятельности – позволяет оценить ИСР и получить календарный план-график сроков с визуальной диаграммой Ганта.

Что такое диаграмма Ганта? **Диаграмма Ганта** (англ. Gantt chart, ленточная диаграмма, график Ганта, календарный график) – это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов.

Генри Гант пришел к идее визуального отображения процессов и прогресса, чтобы облегчить просмотр и учет производственных графиков в начале XX века. Он занялся визуальным планированием с гистограммами, чтобы дать понять супервайзерам, было ли производство успешным или отстало от графика.

Что можно увидеть и отследить с помощью диаграмм Ганта:

- какие задачи включает в себя проект;
- даты начала и окончания задач проекта и проекта в целом;
- сколько времени займет каждая задача;
- кто работает над каждой конкретной задачей;
- последовательность выполнения задач и их группировка.

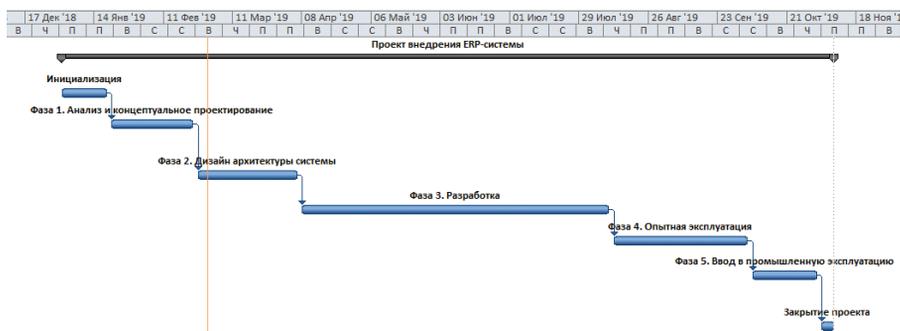


Рис. 7.2. Пример диаграммы Ганта

В Excel можно на основе таблицы вида «задача – начало – конец» через условное оформление ячеек и формулы нарисовать диаграмму Ганта по недельному плану. Но вот для автоматического пересчета дат по связанным задачам при моделировании различных сценариев (когда мы «играем» последовательностью их выполнения) придется прилагать большие усилия. Тогда как специальное ПО для управления проектами позволяет задавать связи между задачами и их трудозатраты, а график с планированием дат начала и конца работ формируется сам.

Диаграмма Ганта – это отличный инструмент для расчета сроков проекта, если знать трудозатраты по каждой задаче ИСР и возможную последовательность их выполнения (параллельно, последовательно, зависимые между собой задачи).

Еще одним очень важным аспектом при расчете бюджета и сроков является **правило Парето 80–20** (англ. The Pareto Principle): 80 % задач по проекту делается с помощью 20 % усилий, а остальные 20 % задач потребуют 80 % усилий.

Само правило Вильфредо Федерико Дамасо Парето вывел в конце XIX века при анализе статистики по сбору урожая, когда 20 % кустов давали 80 % урожая. Это правило работает во многих областях, например: 20 % сотрудников компании приносят 80 % пользы; 20 % покупателей (и товаров) приносят 80 % выручки.

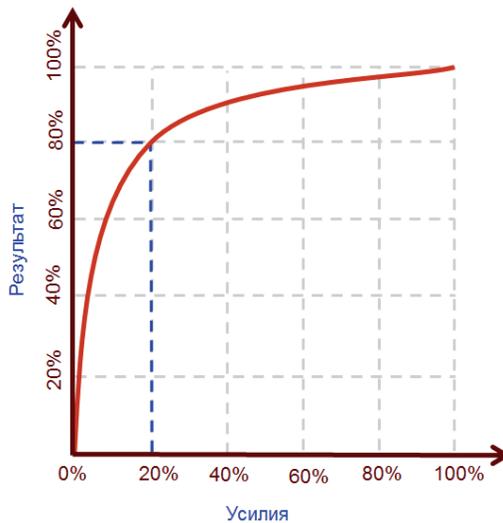


Рис. 7.3. График результата и усилий по правилу Парето 80–20

В случае проекта автоматизации и внедрения ERP-системы «видимый результат» может достигаться быстро, 80 %-ную готовность получить легко. Но система и проект должны быть завершены на 100 %. То есть сроки разных процентов готовности проекта нелинейные. Получить внешне рабочую систему можно просто и быстро, а вот отладить и запустить ее в полноценную работу – сложно. Это правило 80/20 – вредный трюк для мнимого соблюдения сроков, когда сдается система по списку спецификации требований, но при углубленном изучении (в работе) все начинает сыпаться и требуются значительные усилия (те самые 80 %) для доведения всего до рабочего состояния. Это нужно учитывать при оценке трудозатрат и планировании сроков, т. к. задача состоит в оценке бюджета проекта (и сроков), исходя из его завершенности на 100 %.

Для оценки трудозатрат на реализацию проекта применяются следующие методы:

- оценка по аналогам;
- анализ предложений исполнителей;
- оценка «снизу вверх»;
- оценка «сверху вниз»;

- экспертная оценка;
- параметрическая оценка;
- оценка по трем точкам (метод PERT).

Рассмотрим их подробнее в следующем разделе.

Методы оценки трудозатрат

Оценка по аналогам

Оценка длительности и трудоемкости по аналогам строится на данных о фактической длительности и трудоемкости ранее сделанной аналогичной работы. Этот метод часто используется при оценке длительности проекта в условиях недостатка детальной информации о проекте – например, на ранних фазах проекта. Оценка по аналогам использует историческую информацию, накапливаемую в компании-исполнителе, а также экспертные оценки сотрудников (их проектный опыт у других работодателей).

Как правило, оценка по аналогам обходится дешевле и занимает меньше времени, чем другие методы, но при этом она обычно оказывается и менее точной. Оценки длительности по аналогам могут применяться ко всему проекту или к его частям, а также могут использоваться вместе с другими методами оценки.

Оценка длительности по аналогам наиболее надежна в тех случаях, когда ранее выполнявшиеся задачи схожи по сути, а не только по форме, а у членов команды проекта, подготавливающих оценки, есть необходимый (аналогичный) проектный опыт выполнения таких задач. Например, необходимо быстро оценить, сколько времени понадобится на запуск в эксплуатацию 1С:ERP на очередном филиале производственного предприятия. До этого уже прошла автоматизация нескольких аналогичных филиалов данного предприятия, поэтому известны средние сроки и бюджет на автоматизацию одного филиала. С высокой долей вероятности можно утверждать, что на автоматизацию нового филиала потребуется то же время и аналогичный бюджет. Происходит так называемое тиражирование решения.

Анализ предложений исполнителей

Очень простой метод получения оценок – это попросить кого-то другого оценить эти задачи. Применяется при наличии нескольких исполнителей или подрядных организаций, желающих выполнить данный объем работ. Тендерная документация, требования или иная проектная документация рассылается по исполнителям-претендентам с просьбой предоставить свои оценки стоимости и сроков выполнения данных работ.

Понятно, что внутри себя эти исполнители будут использовать какой-то иной метод оценки, применимый для конечного исполнителя. Полученная информация может использоваться для анализа и уточнения различающихся между собой оценок, например методом PERT.

Оценка «снизу вверх»

Это метод оценки больших объемов работ за счет суммирования оценок, полученных для более мелких (и понятных для оценки) составляющих данной работы. Чем более подробно и точно разработана ИСР проекта, тем точнее и корректнее могут быть получены стоимостные оценки по проекту.

Для оценки самих низкоуровневых задач ИСР используются другие методы оценок (по аналогам, экспертная, параметрическая, PERT).

Метод «снизу вверх» считается одним из самых точных при должной декомпозиции задач в ИСР.

Оценка «сверху вниз»

Этот метод применяется в условиях отсутствия детальной ИСР, нехватки информации о ресурсах и материалах, необходимых для реализации работ. Метод «сверху вниз» значительно менее точный по сравнению с методом «снизу вверх».

Очевидно, что подход «сверху вниз» наиболее оптимально использовать на самых ранних сроках выполнения проекта, и он требует значительного проектного опыта от специалистов, проводящих оценку, взяв за основу оценки по аналогам и экспертную оценку.

Технология «сверху вниз» предполагает ровно обратные шаги по отношению к методу «снизу вверх». Сначала дается укрупненная оценка всего пакета работ, а затем она детализируется и декомпозируется на отдельные элементы (по работам, исполнителям и др.).

На ранних этапах проекта, когда выполняется оценка его жизнеспособности и еще непонятно, следует ли расходовать ресурсы на более детальное планирование и оценку, метод «сверху вниз» позволяет с минимальными усилиями получить данные для анализа судьбы проекта.

Экспертная оценка

Трудоемкость и продолжительность выполнения задач часто очень трудно поддаются оценке в силу большого количества влияющих на них факторов (квалификация и производительность сотрудников, специфические риски, окружение проекта и т.п.). Эксперт, который делает оценку конкретной задачи или пула задач (или всего проекта), опирается на историческую информацию и свой проектный опыт.

Условно можно сказать, что эксперт моделирует все задачи на себя: «За сколько я могу это сделать сам? А если не сам, а кто-то с более низкой квалификацией? А если при этом нужно будет что-то переделывать, ошибки какие-то вскроются и нужно будет время их поправить?»

Члены команды проекта могут брать информацию по оценке длительности и трудоемкости из похожих задач прошлого опыта. При этом надо отдавать себе строгий отчет в том, что если такой информации нет, то оценка длительности получается более неопределенной и рискованной.

Данный метод рекомендуется применять вместе с PERT-методом для получения более достоверной оценки и диапазона точности оценки.

Можно получить детальные экспертные оценки по шаблонам задач разных видов и использовать их в методе параметрической оценки.

Параметрическая оценка

Параметрическая оценка – метод оценки, использующий алгоритм для вычисления стоимости или длительности на основе исторических данных и параметров проекта.

Оценка дается по параметрам задач проекта. При этом известны (опытно, по факту, по аналогам или как экспертные оценки) нормативы для таких параметров. Данный метод может обеспечивать более высокую степень точности в зависимости от опыта и данных, заложенных в основу модели. Параметрическая оценка может применяться ко всему проекту или к его частям вместе с другими методами оценки.

Например, необходимо определить время, необходимое для разработки новых 10 отчетов. Известно из предыдущего опыта, что на разработку одного отчета уходит примерно 20 человеко-часов. Таким образом, общее время разработки всех отчетов составит 200 человеко-часов.

Если же использовать оптимистичные-реалистичные-пессимистичные оценки параметров и применить параллельно с этим метод PERT, то можно получить более точную оценку и размер ее погрешности.

В качестве параметров задач, связанных с доработкой ERP-системы, можно использовать детальные готовые шаблоны для оценок модификаций системы по градациям: очень простая, простая, средняя и сложная модификация. Подробнее об оценке бюджета и его сроков с помощью шаблонов модификаций поговорим в одном из следующих разделов этой главы.

Оценка по трем точкам (метод PERT)

В методе PERT (от англ. Project Evaluation and Review Technique) обрабатываются три экспертных оценки срока, необходимого для выполнения задачи одним условным специалистом.

Три точки:

- оптимистическое время (англ. optimistic time) **Ot** – «раньше не справлюсь точно, даже если повезет (ни один риск не «выстрелит»)»;

- пессимистическое время (англ. pessimistic time) **Pt** – «успею гарантированно, даже если все риски овеществятся»;
- наиболее вероятное время (англ. most likely time) **Mt** – «скорее всего, успею».

Эти три точки позволяют определить ожидаемое время (англ. expected time) **Et**.

Формула PERT:

$$Et = (Ot + 4 * Mt + Pt) / 6.$$

$$\text{Отклонение PERT} = (Pt - Ot) / 6.$$

В результате экспертного опроса руководитель проекта получает несколько оценок по длительности выполнения задачи. Используя формулы PERT, можно получить некую реалистичную оценку и наиболее вероятное отклонение от нее в определенном диапазоне.

Рассмотрим использование формулы на примере. Необходимо оценить задачу создания нового отчета.

Опрашиваем трех экспертов (специалистов проектной команды, кто в состоянии сделать такой отчет), просим назвать свои оценки по длительности такого рода работы. Предположим, что были названы сроки: $Ot = 8$, $Mt = 16$ и $Pt = 40$ часов.

Таким образом, ожидаемое время Et по длительности составит $(8 + 4 * 16 + 40) / 6 = 18,67$ человеко-часа, а погрешность оценки PERT +/- $(40 - 8) / 6 = 5,33$ часа.

Мифический человеко-час

Обычно трудозатраты по работам в консалтинговых проектах по внедрению ERP-систем оцениваются в человеко-часах работы специалистов над задачами, часы суммируются в рабочие дни, дни в недели, недели в месяцы итоговых сроков проекта. Рассмотрим подробнее, что собой представляет человеко-час и сколько их может предоставить специалист, участвующий в проекте.

Данный раздел книги назван по мотивам известной и рекомендуемой к самостоятельному изучению книги «Мифический человек-месяц, или Как создаются программные системы» (англ. The Mythical Man-Month: Essays on Software Engineering). Книга Фредерика Брукса об управлении проектами в области разработки программного обеспечения (на основе опыта IBM OS/360) впервые была опубликована в 1975 году. В 1995 году было добавлено несколько новых глав, актуализирующих книгу.

Современный мир повысил темпы создания новых продуктов и технологий, поэтому счет сегодня логично вести на часы, тем более мы оцениваем бюджет в человеко-часах, пусть и в сотнях или тысячах. Но выводы книги для нас очень интересны, и они реально работают на практике, какими бы удивительными ни казались заключения автора на первый взгляд.

Фредерик Брукс сделал вывод: «Время выполнения проекта не обратно пропорционально числу программистов, по крайней мере, по двум причинам:

- В программировании, в отличие от, например, сбора хлопка, работа не может быть произвольно разделена на несколько независимых частей. Части проекта зависят друг от друга, и некоторые задачи можно начинать выполнять только после того, как будут закончены другие.
- Программисты должны тратить часть времени на взаимодействие друг с другом.

Если есть N специалистов, то количество пар специалистов (для взаимодействия между собой, в нашем случае «консультант – разработчик») равно $N(N-1)/2$, то есть с ростом числа участников проекта затраты времени на взаимодействие растут квадратично. Поэтому начиная с какого-то N рост числа участников проектной команды замедляет выполнение проекта».

Фактический пример из реальной оценки одной проектной задачи. Есть задача на 320 ч, один разработчик делает ее за 8 недель. Привлекаем второго разработчика, задача делается за 4 недели (линейный прирост). Далее привлекаем еще специалиста, результат уже не кратный, а те же 3–4 недели. А если добавить сюда еще кого-то (явно лишнего), то срок будет такой же (3–4 недели, если не будет мешать работать остальным своим желанием «помочь») или увеличится до 6–8 недель, т. к. время на коммуникацию всех со всеми начнет расти в ущерб самой задаче.

Если сроки сорваны, наем новых специалистов замедляет выполнение проекта по причине того, что новичкам требуется время на изучение текущего состояния задач, подходов к разработке. В книге сформулирован **закон Брукса**: «Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше».

В примере, что был рассмотрен выше, где 1 специалист делал задачу за 8 недель, предположим, что он уже потратил 4 недели, 4 остается, но очень хочется ускорить процесс за счет привлечения дополнительных сил. Добавим еще одного специалиста – в идеале можно было бы поделить оставшиеся 4 недели на два, но фактически экономия будет где-то 1 неделя, т. к. половину времени новый специалист будет «въезжать» в задачу, а тут еще и 50-процентная готовность, чужой код... А если привлечь двух специалистов в помощь? Тогда срок будет 9+ недель, то есть вырастет! Срок не изменится, если эти новые специалисты не будут мешать первому и он продолжит делать все сам, но если задачи поделить и «ускорять», то будут накладные расходы на интеграцию наработок и коммуникацию.

При очень большом числе специалистов проект может быть вообще никогда не закончен: из-за общей неразберихи попытки исправить существующие ошибки в программном обеспечении порождают новые ошибки, так что система не улучшается.

Можно привести любимую руководителями проектов цитату, когда нужно это все обосновать в одном предложении: «Даже если собрать вместе 9 беременных женщин, то ребенок через месяц не родится».

Какой вывод можно сделать из рассмотренных тезисов? При планировании сроков работ нельзя просто делить итоговое число часов трудозатрат, полученное по оценкам пунктов ИСР, на число специалистов и их часы работы. Срок должен учитывать последовательность задач и их возможную декомпозицию на подзадачи.

В своей книге Фредерик Брукс приводит также 2 способа снизить стоимость разработки программного обеспечения:

- Привлекать в проект программистов только после того, как построена архитектура системы. Иначе при длительности этой стадии, например, в несколько месяцев программистам будет нечего делать.
- Купить часть программного обеспечения у других разработчиков.

Эти тезисы актуальны и для процесса внедрения ERP-системы: на начальном этапе команда проекта должна быть разумной, а не максимальной, участники добавляются и убираются по необходимости в зависимости от задач на разных фазах проекта (см. рис. 5.1 – график привлекаемых специалистов). Предполагается, что у исполнителя этот проект – не единственный источник доходов и специалисты могут переключаться на задачи других проектов на время простоя, например в процессе согласования. На практике это не всегда так, иногда команда выделяется сразу и содержится за счет именно этого проекта.

Второй тезис для проекта внедрения – это возможность закрыть какие-то потребности готовыми отраслевыми решениями или отчуждаемыми функциональными блоками, которые могут встраиваться в выбранную ERP-систему. Стоимость этих блоков может быть значительной, но расчеты трудозатрат на разработку, отладку и запуск в эксплуатацию,

а главное, сроки на все это могут говорить в пользу приобретения готового решения.

Давайте рассмотрим еще один интересный вопрос: сколько же «мифических» человеко-часов может «выдать» в месяц условный специалист?

Тут нужно учесть, сколько сотрудник реально работает часов в средний месяц условного года с учетом больничных, отпусков и других факторов.

Обычный расчет говорит нам, что 8 ч в день, 5 дней в неделю дают 40 ч в неделю. Что при 20 рабочих днях в месяц дает 160 ч, а при 23 рабочих днях – уже 184 ч. Но в среднем по году в месяце что получится?

Учтем факторы: отпуск сотрудника 4 недели, болезни условные 2 недели в год – это минус 240 ч в год, еще есть множество государственных праздников (минус еще 2 недели в году). Получается, из 52 недель в календарном году реально рабочие у сотрудника 44 недели, это 1760 ч. Делим на 12 месяцев, получаем 146 часов в условный месяц.

Именно эти 146 часов в месяц нужно планировать на проектах более 6 месяцев длительностью. Относительно 160 и 184 ч (для 20 и 23 рабочих дней в месяце) получаем разницу почти в 10 % и 20 % соответственно. Сверхурочное время и переработки (overtime) будут запасом на случаи форс-мажоров и обычно оптимистичных оценок проектов. Если же на это «святое» время закладываться сразу, то это заведомый срыв сроков или снижение качества, т. к. в режиме аврала специалисты долго в приемлемом качестве работать не смогут.

Эти условные 146 ч. в условный месяц и есть наши искомые «мифические человеко-часы», которые мы в следующем разделе будем учитывать в расчете себестоимости человеко-часа по годовому бюджету расходов.

Еще одним важным фактором в учете человеко-часов является доля участия того или иного специалиста.

Если в известной сказке «В стране невыученных уроков» ответ для математической задачи «полтора землекопа» был явно неверным, то вот в консалтинге 1,5 разработчика или 2,5 консультанта – это нормальное дело. Может встретиться даже 0,25 системного архитектора.

Долевое участие в проекте для специалистов применяется при наличии в компании-исполнителе других проектов и задач. Оно позволяет рассчитать для заказчика приемлемый бюджет проекта за счет более точного распределения работ по специалистам (привлечение их по мере необходимости на разных фазах проекта). Если же вся работа для команды проекта предполагается как единственный монопроект, то там предполагается 100-процентное участие всех – возможно, в разных ролях на разных фазах проекта. При этом нужно понимать, что на ранних стадиях проекта всем специалистам просто нет задач (нечего разрабатывать разработчикам, пока нет утвержденных технических заданий, нечего тестировать, пока что-то не разработано и т. п.).

Руководитель проекта со стороны заказчика должен понимать потребность в специалистах и загрузку всех специалистов исполнителя на всех фазах проекта и логику расчета бюджета (из трудозатрат всех участников), т. к. желание компании-исполнителя включить в проект содержание проектной команды в режиме «простоя» понятно, но и желание не переплачивать за это у заказчика тоже понятно.

Расчет себестоимости человеко-часа

В проектах внедрения ERP-систем и консалтинге вообще услуги оказываются обычно по часовым ставкам участвующих в проекте специалистов. Для компании-заказчика ставки специалистов исполнителя могут быть едиными или зависеть от должности либо роли исполнителя (например, руководитель проектов и системный архитектор идут по повышенной часовой ставке, а остальные – по обычной). Практикуется разделение ставок консультантов и разработчиков или градации внутри них на ведущих специалистов и просто специалистов. Тарифные ставки (или сетка тарифных ставок) определяются компанией-исполнителем

самостоятельно, исходя из внутрифирменных правил и среднерыночных ставок по отрасли и региону.

В этом разделе мы рассмотрим вопрос расчета минимальной ставки – условной себестоимости человеко-часа для компании.

Варианты расчета могут сильно различаться в зависимости от принятой в компании-исполнителе системы оплаты труда и мотивационных схем. Так возможны варианты:

- сотрудники работают на фиксированных окладах;
- оклад состоит из долей: фиксированной и переменной части. Тут возможны разные пропорции: фиксированная часть небольшая или большая по отношению к переменной части, образованной отработанными и оплачиваемыми часами;
- сотрудники работают по фрилансовой схеме, фиксированной части окладов не имеют вообще, все по часовым ставкам по факту работы.

Для расчета себестоимости человеко-часа работ нужно учесть все расходы по компании и знать количество часов, которое сотрудники могут работать на проектах и которое можно реализовать за рассматриваемый период. К расходам компании относятся:

- оплата труда сотрудникам, участвующим в проектах (фиксированная и переменная часть, которую считаем по максимально возможным часам, выделяемым на работу сотрудниками);
- оплата труда сотрудников, не участвующих в проектах (это руководство компании, бухгалтерия, маркетинг, ИТ-специалисты, администраторы, секретари и т. п.);
- налоги на фонд оплаты труда;
- социальные выплаты сотрудникам, добровольное медицинское страхование (ДМС);
- расходы на аренду офиса;

- расходы на мебель и оборудование;
- расходы на программное обеспечение;
- расходы на командировки (не включаемые в проекты явным образом);
- периодические расходы на маркетинг, продвижение и обучение (статьи в СМИ, сайт компании, участие и проведение конференций и тренингов, обучение сотрудников);
- расходы на поддержание партнерских отношений (партнерство с поставщиком ERP-системы, взносы на участие в различных отраслевых ассоциациях и т. п.);
- расходы на проценты по привлекаемому финансированию (кредиты и займы);
- прочие налоги (зависят от системы налогообложения компании);
- прочие расходы на поддержание работы компании (интернет, телефоны, услуги банка и прочее).

Доходами компании являются доходы от основной деятельности – продажи программного обеспечения и оказания услуг по проектам внедрения ERP-систем. Прочими доходами от неосновной деятельности в расчете можно пренебречь из-за их незначительности или отсутствия вовсе (это могут быть проценты по депозитам остатков денежных средств на расчетных счетах, доходы от проведения мероприятий, например тренингов, и т. п.).

Так как в консалтинге нет товарооборота, а в основном оказание услуг, то основная себестоимость работ – это фонд оплаты труда и налоги с него. Нужно понимать, что увеличение расходов на оплату сотрудникам за счет налогов составляет почти 50 %. Например, для того чтобы на руки сотрудник получил 100 руб., в расходы нужно планировать почти 150 руб. (13 % НДФЛ с оклада и 30 % налоги в различные фонды): оклад 115 руб., НДФЛ с него 14,95 руб., пенсионные взносы, социальное и медицинское страхование – 34,5 руб.

Составляем таблицу бюджета доходов и расходов (БДР). При расчете ставок человеко-часа мы еще не знаем, какие у нас будут доходы, нам нужно определить себестоимость ставок, то есть окупаемость всех расходов за счет ведения деятельности. Поэтому расходы нужно учесть максимально полно. БДР нужно составлять ежемесячно на год или более, исходя из предполагаемой длительности проектов. Далее делим полученную сумму расходов на количество человеко-часов, которые могут обеспечить специалисты компании (с учетом реальной возможности все эти часы реализовать, иначе нужны понижающие коэффициенты в расчете). Получаем примерную себестоимость часа, ниже которой компания начнет получать убытки от своей деятельности.

Тут не заложены проектные риски, прибыль компании – все это начнет повышать ставку часа до того самого среднерыночного уровня.

Можно продолжить моделирование в таблице БДР, заложив в нее ожидаемые учредителями компании дивиденды (и налоги с них) в расходную часть, туда же – риски на непредвиденные расходы. Это будет не совсем уже БДР, но как расчетная модель для нашей задачи подходит. Оптимистичные и пессимистичные суммы в плане при моделировании позволят получитьвилку ставок человеко-часа.

При моделировании можно учесть чистый доход от продажи самой ERP-системы, но скорее это нужно учитывать в составе конкретного проекта, в рамках этой суммы можно управлять рисками и прибыльностью проекта.

Далее эти доходы и расходы нужно переложить в бюджет движения денежных средств (БДДС) для расчета и поддержания в компании положительного остатка денежных средств на расчетных счетах и минимизации кассовых разрывов. Это уже несколько другая тема, она выходит за рамки этого раздела книги. Отметим только, что предполагаемые денежные потоки по расходам наложат свои требования на схему расчетов с заказчиками по этапам проекта, чтобы деньги выплачивались (и авансировались) по закрытым актами фазам проекта своевременно и компания могла функционировать (выплачивать зарплату сотрудникам и платить по счетам для других расходов). Либо нужно планировать привлечение дополнительного финансирования (кредитов) на закрытие таких кассовых разрывов между поступлениями и расходами денежных средств по периодам (месяцам).

Снижение себестоимости ставки человеко-часа – это конкурентное преимущество для компании при участии в тендерах проектов автоматизации. Снижение ставки возможно за счет снижения расходов по компании. Конечно, прямое влияние оказывает фонд оплаты труда прямых участников проектов внедрения – самих специалистов. Для формирования успешной и квалифицированной команды в ней должны присутствовать сильные специалисты, а они будут готовы работать за зарплаты, сопоставимые со средними по рынку для отрасли и региона. То есть сильно снижать эту статью расхода не получится – некому будет работать над проектами. Оптимизация возможна за счет отказа от непрофильных видов работ в компании с переводом на аутсорсинг выполнения возникающих задач (например, компании не нужен штатный бухгалтер с полноценным окладом и всеми налогами с него). Есть сервисы по ведению бухучета организации, например сервис 1С:БухОбслуживание (<https://1cbo.ru>), месячные ставки по которому явно ниже содержания штатного сотрудника. Аналогично могут быть переданы на аутсорсинг возникающие задачи по маркетинговым мероприятиям и исследованиям. Минимизация расходов на офис (площади, мебель, оборудование), перевод большинства работ в дистанционный режим. Любая оптимизация в итоге снижает внутрифирменную себестоимость человеко-часа, что позволяет повысить доходность о т проектов при использовании среднерыночной ставки.

Следующей возможностью оптимизации расходов может служить выбор системы налогообложения. Например, применение компанией-исполнителем упрощенной системы налогообложения (УСН) вместо основной системы налогообложения (ОСН) с ее 20 %-ной ставкой налога на прибыль и прочими налогами и сборами. Выбор системы налогообложения может заметно влиять на суммы налогов и итоговые суммы выплат в бюджет. УСН можно применять при определенных условиях, обычно небольшим компаниям. УСН может быть двух вариантов:

- «доход» – 6 % налог с доходов;
- «доходы минус расходы» – 15 % с доходов, уменьшаемых на признаваемые расходы.

При упрощенной системе может не быть налога на добавленную стоимость (НДС), а это еще 20 %. Но для заказчика компания-исполнитель – это поставщик услуг автоматизации, и НДС за услуги консалтинга может быть признан и уменьшать исходящий НДС с продаж самого заказчика, так что не всегда это заказчику хорошо. Но вот если сама стоимость проекта может быть ниже на 1/5 по сравнению с использующими ОСН компаниями-исполнителями, то это тоже может быть существенным конкурентным преимуществом.

При использовании УСН «доходы» в расчете БДР нужно заложить налог на входящий денежный поток, а это все расходы на ФОТ и налоги с него в том числе.

Детально рассматривать особенности налогообложения компаний в этой книге не будем. Изучить подробнее тему действующих в РФ налогов и сборов можно на официальном сайте налоговой службы (www.nalog.ru).

Важно принять тот факт, что выбор системы налогообложения существенно влияет на рассматриваемый в этом разделе вопрос расчета ставок человеко-часа.

Лежащим на поверхности инструментом минимизации затрат может выглядеть использование в проекте фрилансеров, так как оптимизация возможна за счет оформления таких специалистов не как сотрудников, а как индивидуальных предпринимателей (ИП) на УСН, это существенно экономит на налогах с ФОТ (до 44 %), но требует соблюдения условия, что такой проект и работодатель у специалиста не единственные.

Фрилансер (англ. freelancer, буквально «свободное копьё», наемник) – специалист, привлекаемый на время, на конкретную задачу. Работа по фрилансу часто удаленная, хотя возможно очное присутствие и командировки. Многим сильным специалистам выгоднее работать по такой схеме, как ИП, чем быть в штате одной компании, т.к. итоговая загрузка (а значит, и доход) фрилансера ограничена только его трудолюбием и навыками.

Ставки при этом у таких специалистов выше, включают в себя отпуска, больничные и простои, но зато и работают они только на задачи проекта, не неся риски обеспечения им 100 % загрузки. Например, не нужно оплачивать месячный оклад специалисту (со всеми налогами с него), когда задача есть только на 40 часов, платим по факту (time & material в действии). Но у этого кажущимся привлекательным подхода для проектов внедрения ERP-систем есть неприятная обратная сторона. Это риски низкого качества, несоблюдения сроков и неуправляемости специалистов (они под давлением могут отказаться от задачи вообще). Для проекта в жестких рамках сроков, бюджета, качества такой путь оптимизации принесет больше вреда. Хороших специалистов рекомендуется брать в штат в команду исполнителя или заказчика (в центр компетенции). При планировании ресурсов стоит рассматривать специалистов с нужными компетенциями от проверенных партнеров, являющихся центрами компетенции по ERP («1С: Центр ERP» – <https://1c.ru/rus/partners/ckerp.jsp>), которых можно взять на субподряд. Тогда работа этих специалистов в проекте будет управляемой с гарантированным ожидаемым результатом и даст искомую экономию на неполной загрузке по time & material.

Как мы видим, на расчет себестоимости человека-часа влияют множество факторов, главное – прийти в итоге к окупаемости проекта, достойной компенсации труда всех участников, прибыли для владельцев компании и возможности успешно завершить проект внедрения для получения ожидаемых выгод заказчику автоматизации.

Как получить итоговый срок проекта

В предыдущих разделах мы рассмотрели особенности того, что календарная длительность проекта не зависит линейно от трудозатрат в оценках задач в расчете на одного специалиста. Разделение задачи между несколькими специалистами вызывает дополнительные затраты на обучение и обмен информацией, что приводит, как правило, к падению производительности на 10–30 % в расчете на каждого дополнительного участника команды проекта, участвующего или влияющего на разработку. То есть получается, что небольшие команды более эффективны. Но следует отметить, что при большой масштабности задач небольшая команда может просто не справиться в разумные сроки или с приемлемым уровнем качества. В общем случае нельзя просто взять и поделить итоговое число часов, полученное оценками ИСР проекта, на 8 рабочих часов в день и число специалистов в команде для получения итогового срока в рабочих днях. Но тогда как же получить календарный срок проекта, зная его трудозатраты?

Прежде чем ответить на этот вопрос, вернемся к особенностям самой оценки трудозатрат, т.к. этот вопрос не был закрыт рассмотрением методов оценок до конца. Все приведенные выше методы оценки трудозатрат, используемые при оценках бюджета проекта, показывают парадоксальную картину: все сводится к экспертным оценкам как основе для прочих моделей («снизу вверх», PERT, аналогов). А как именно думает эксперт – не формализованный процесс! Это же человек, а человеку свойственно ошибаться. Разработчики часто склонны быть оптимистами и оценивать любую задачу только со стороны разработки. А это время изучения задачи и самого процесса программирования ее реализации. Отладка, тестирование, документирование, обучение пользователей, настройка данных для рабочей базы – это все остается за бортом, если спросить: «За сколько сделаешь?» – у разработчика. Что логично: остальное же делает не он. В проектах автоматизации много работы, вообще не связанной непосредственно с разработкой.

Поэтому спрашивать как эксперта нужно системного архитектора, который понимает, когда задача действительно завершается, сколько людей и каких над ней работают.

Как бороться с рисками обычно оптимистичных оценок экспертов со стороны разработки, мы рассмотрим в следующей главе. Если кратко, то нужно вводить коэффициент-мультипликатор к оценкам экспертов (причем у каждого эксперта он может быть своим и определяется руководителем проекта опытно).

Рассмотрим подробнее прием применения параметрических оценок, когда задачи ИСР классифицируются по сложности реализации и применяется соответствующий шаблон оценок. Например, шаблоны от сложности модификаций:

- очень простая;
- простая;
- средняя;
- сложная.

Сами шаблоны оценок получаются экспертными оценками, аналогами и уточняются формулой PERT.

Ключевое для нас – это понять состав работ внутри такого шаблона, то, что даже самая простая на первый взгляд задача («да что тут делать, за 15 мин справлюсь») имеет определенный пул сопроводительных работ и задач на 15 минут в проектах внедрения ERP-систем просто не бывает.

Рассмотрим в качестве примера шаблон оценки разработки простого отчета: какие работы входят в такую модификацию системы, кто участвует (в часах) и сколько календарного времени это может занять. Оценки даны примерные и могут различаться внутри разных команд.

Как получить итоговый срок проекта

Описание работы	Консультант	Разработчик	Системный архитектор	Календарных дней
Формирование шаблона отчета с предварительным описанием заполнения по столбцам	4		1	0,5
Обсуждение и согласование шаблона с заказчиком	1		0,5	0,5
Подготовка ТЗ с описанием шаблона и особенностей алгоритмов	4		1	0,5
Обсуждение, согласование и уточнение ТЗ	2	1	1	0,5
Разработка по утвержденному ТЗ		8	0,5	1
Тестирование модификации системы, устранение дефектов	3	2	0,5	1
Обновление или подготовка нового руководства пользователя	2		0,5	0,5
Перенос модификации, настройка данных для демонстрации и тестирования на сервере заказчика	0,5	0,5		0,25
Демонстрация работы нового функционала специалистам заказчика	0,5			0,25
Поддержка тестирования модификации и ее приемки специалистами заказчика	2	2	1	
Итого	19	13,5	6	5

Отметим, что прямой конвертации трудозатрат в человеко-часах в календарные дни путем деления часов на 8 часов в рабочем дне в шаблоне нет. В случае задач с согласованием (внутри команды и тем более с заказчиком) возникают накладные расходы на ожидание, когда специалисты смогут встретиться (удаленно подключиться) и обсудить или посмотреть функциональность. Во время простоя могут выполняться другие задачи, но календарно именно разработка некоего простого отчета быстрее не завершится.

Поддержка процесса тестирования специалистами заказчика календарно в этом примере шаблона не оценена, т.к. считаем, что модификация от исполнителя уже сдана заказчику. При сложной структуре баз данных проекта (отдельные базы для разработки, тестирования исполнителем, тестирования заказчиком и рабочей базы) календарное время на тестирование заказчиком тоже нужно закладывать в шаблон, т.к. без завершения этого этапа модификация не попадет в рабочую базу. Так как логично общее время задачи считать до полного завершения жизненного цикла разработки – передачи функциональности конечному пользователю и перевода ее в промышленную эксплуатацию.

Как мы видим из рассмотренного примера, довольно простой отчет, который, если спросить конечного разработчика: «За сколько сделаешь?» – то он ответит: «За 1 день», выливается в почти 40-часовые работы на 5 календарных дней. Вот если спрашивать системного архитектора как эксперта, то он такой отчет оценит – уже с учетом всех участников и запаса времени на согласования с заказчиком – как раз в рабочую неделю на 40 часов, что и соответствует детализированному шаблону.

Такие детальные шаблоны на разные виды задач и разные по сложности модификации хорошо показывают руководителю проекта со стороны заказчика логику расчета трудозатрат и снимают вопросы о том, почему так много часов по такой, казалось бы, простой модификации системы. Так как в ИСР оценки будут сводные на задачу целиком, а не с такой детализацией, как приведено в шаблоне выше. Все это в итоге позволяет защитить сводные оценки трудозатрат, итоговый бюджет и обосновать календарные сроки проекта.

Есть еще один важный психологический аспект, связанный со сроками, и **закон Паркинсона** к нему: «Работа заполняет все время, отпущенное на нее» (для консалтинга можно добавить: «И еще чуть-чуть сверху»). В идеале руководителю проекта нужно планировать разные версии

сроков: для заказчика и для специалистов внутри команды. Установить внутренние сроки меньше внешних, чтобы был запас по времени, о котором не должен знать конечный исполнитель задач. Как это сделать при прозрачности проектной документации и доступности ее всей команде проекта?

Эмпирический закон был сформулирован историком Сирилом Норткотом Паркинсоном в его сатирической статье, напечатанной в британском журнале *The Economist* в 1955 году и изданной позднее в книге «Закон Паркинсона» (англ. *Parkinson's Law: The Pursuit of Progress*; Лондон, John Murray, 1958).

При моделировании в системе управления проектами и задачами можно задачу разделять не просто на «постановку – разработку – тестирование», но добавить еще задачу «внедрение», где стоит отдельный исполнитель, не разработчик или консультант. Эта подзадача будет влиять на сроки завершения всей задачи целиком, но не мешать, например, разработчику переключаться на другие задачи разработки встык с завершенной задачей по связям. И будет фиксировать срок завершения разработки (внутренний срок) при итоговом большем внешнем сроке по всей задаче для заказчика.

Все это делает график запутаннее и сложнее для восприятия, но как подход расчета внешних сроков работает. Для данной книги по введению в тему управления проектами это все может быть преждевременно и излишне усложнит материал и его понимание, до этого нужно прийти на практике самостоятельно. Потому обозначим, что есть проблема планирования внешних сроков и внутренних дедлайнов для команды, это нужно понимать и учитывать при планировании календарных сроков.

Вообще поддерживать в актуальном состоянии максимальной детализации план-график очень сложно. При использовании для планирования Excel (с нарисованной там диаграммой Ганта) перебить все связи задач, их даты и длительность в них будет очень сложно. Потому нужно либо использовать специализированное ПО для проектного управления, либо актуализацию отчетного план-графика проводить только на высокоуровневых задачах и фазах целиком. А отклонения и прогресс исполнения

конкретных задач оставить в системе класса Task Tracking, позволяющих учитывать плановое и фактическое время по задачам. Например, система 1С:СППР позволяет решать вопросы отслеживания и планирования исполнения задач. Для заказчика высокоуровневой актуализации плана-графика проекта будет достаточно, т. к. внутренняя декомпозиция до подзадач разных исполнителей – это внутренние планы исполнителя.

Если руководитель проекта со стороны заказчика настаивает на максимальной детализации и контроле за ходом проектных работ, то ему можно дать доступ в такую систему Task Tracking или строить отчет из нее.

Вернемся с расчету календарного срока проекта. Необходимо учесть время отпусков специалистов (как исполнителя, так и заказчика), заложить риски на болезни (если работы попадают на ежегодные сезоны простуд). Если ИСР имеет достаточную детализацию, то расчет сроков можно доверить инструментарию управления проектом, способному выстроить последовательность исполнения задач диаграммой Ганта на календарь с учетом выходных и праздничных дней. Остается аккуратно задать последовательность и связи исполнения задач, заложить время ожидания реакции со стороны заказчика (этим сложно управлять исполнителю; хотя сроки реакции и прописываются в уставе проекта, но календарно нужно иметь некий дополнительный запас). Получаем итоговый план-график.

В процессе согласования проекта срок его начала может сдвигаться – нужно будет актуализировать план-график: по мере сдвига срока начала проекта смещаются условия, заложенные в план (сезонность, отпуска). Недостаточно только в одной задаче изменить дату начала работ и ждать того, что весь график сместится сам. В идеале – да, и ПО управления проектами такую задачу существенно облегчает, но критически посмотреть и скорректировать новые сроки руководителю проекта тоже нужно. В итоге от переноса сроков начала работ по проекту может измениться календарная длительность срока проекта – например, стать больше на полмесяца за счет того, что происходит наложение на январь, в котором первая неделя праздничная, а февраль короткий месяц.

Нужно также помнить и регулярно обновлять план-график проекта по мере внесения изменений в согласованные ранее границы проекта. В ходе проекта могут возникать дополнительные работы, не заложенные в проект изначально. Расширение границ проекта может быть

оформлено дополнительными соглашениями к договору по обоюдной договоренности сторон, нужно помнить, что любые дополнительные работы (или выявленные задержки), очевидно, влияют на итоговый срок, и лучше это явно отражать в соглашениях. Сами соглашения могут менять или не менять бюджет проекта, но вот срок они, скорее всего, увеличат в любом случае.

Ведение руководителем проекта со стороны исполнителя план-факта по исполнению задач, выполняемых сотрудниками заказчика, и соблюдению их контрольных сроков позволит в итоге исполнителю бесконфликтно сдвинуть срок на накопленные заказчиком задержки. Например, исполнитель не сможет реализовать свою задачу по загрузке начальных данных, если заказчик их не предоставит своевременно. Или будет все время переноситься начало обучения из-за невозможности собрать сотрудников заказчика в группу.

Руководителю проекта нужно учесть такие задержки при планировании работ, чтобы не приходилось регулярно менять заявленный срок проекта, пусть и не по вине исполнителя, так как частое продление срока смотрится не очень хорошо – больше похоже на плохое планирование и управление рисками.

Как выявлять риски и бороться с ними на ранних стадиях, рассмотрим в следующей главе.

Глава 8. Риски проекта и управление ими

Что такое риски проекта

Любые проекты подвержены риску, поскольку они являются уникальными предприятиями с различным уровнем сложности, которые осуществляются с целью получения тех или иных выгод. Проекты осуществляются в контексте ограничений и допущений, а также ожиданий заинтересованных сторон, которые могут противоречить друг другу и изменяться со временем с ходом работ по проекту.

Организации (заказчика и исполнителя) берут на себя осознанный и контролируемый риск по выполнению проекта с целью создания ценности, соразмеряя при этом риски и выгоды.

Риск проекта – это неопределенное событие или условие, которое в случае его наступления будет иметь положительный или отрицательный эффект на один или несколько целевых параметров проекта (сроки, стоимость, содержание, качество).

Если не управлять проектными рисками, они имеют потенциал вызывать отклонение проекта от плана и приводить к тому, что проект вообще не может достигнуть установленных целей.

«Чтобы управлять проектом, достаточно управлять его рисками», – утверждает Том ДеМарко, известный консультант по программной инженерии, в своей книге «Deadline. Роман об управлении проектами» (написана в 1997 г.). Действительно, всю работу руководителя проекта можно свести к одному: борьба с рисками, которые могут помешать проекту завершиться в срок, вписаться в бюджет и выполняться с должным уровнем качества.

Риск внутри каждого проекта существует на двух уровнях: каждый проект имеет индивидуальные риски, которые могут оказать влияние на достижение его целей, и рискованность проекта в целом, которая вытекает из сочетания индивидуальных рисков проекта и других источников неопределенности. Уровни рисков проекта можно определить следующим образом:

- **индивидуальный риск проекта** – это неопределенное событие или условие, наступление которого позитивно или негативно сказывается на одной или нескольких целях проекта;
- **совокупный риск проекта** – это воздействие неопределенности на проект в целом, возникающее из любых источников неопределенности, включая индивидуальные риски, представляющие собой влияние последствий вариаций результатов проекта (как позитивных, так и негативных) на заинтересованные стороны.

В самом слове «риск» слышится что-то негативное. Как риск может быть позитивным? Рассмотрим примеры событий, которые положительно сказываются на состоянии одной или обеих сторон проекта:

- изменение курса валюты в выгодную сторону;
- изменение налогового законодательства в сторону послабления, приводящее к экономии средств;

- утверждение завышенной оценки по какой-то задаче дает запас времени и бюджета для покрытия рисков заниженной оценки по другим задачам.

Риски в проекте внедрения ERP-системы типовые и могут быть прописаны в уставе проекта, как и инструментарий по их профилактике, выявлению срабатывания и минимизации последствий. Подробнее типовые риски проектов внедрения ERP-систем рассмотрим в следующем разделе.

Сами риски делятся:

- на **событийные риски** – что-то происходит (событие): поставщик перестал предоставлять обновления для внедряемой системы, изменение требований заказчика, изменения бизнеса заказчика;
- **несобытийные риски** – не имеют явного события, разделяются:
 - на **риск вариативности** – отклонения от целевых ожиданий (производительность, ошибки и т.п.) как в большую, так и меньшую сторону;
 - **риск неоднозначности** – неопределенность относительно будущих состояний и потребности (пригодности) в разрабатываемой функциональности. Проблему неоднозначности можно решать путем инкрементной поэтапной разработки, создания прототипов или моделирования.

Компоненты риска:

- **событие** – что происходит, когда риск становится реальностью;
- **вероятность** – насколько вероятно наступление события;
- **влияние** – степень влияния на стоимость, сроки или качество проекта.

Для управления рисками руководителю проекта важно понять их источники, определить список рисков, оценить вероятность их наступления и степень влияния и, самое главное, что с этими рисками делать, когда они проявятся.

Управление рисками включает в себя правила и процедуры, относящиеся к планированию управления рисками:

- идентификация рисков;
- анализ и оценка рисков;
- планирование реагирования на риски;
- мониторинг и контроль рисков.

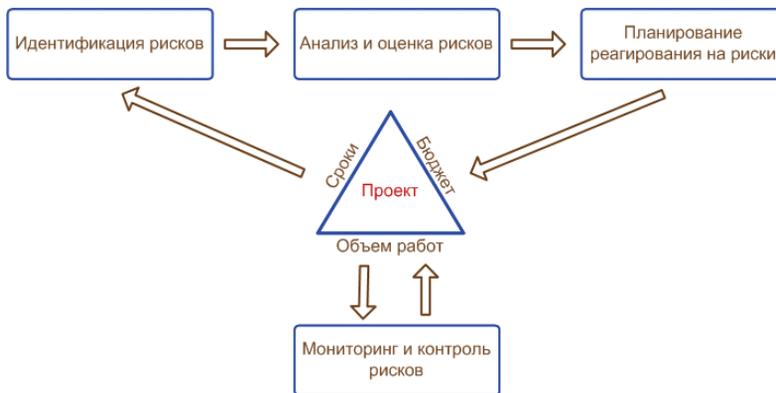


Рис. 8.1. Управление рисками проекта

Важно понимать, что управление рисками – это управление причинами (событиями), к ним приводящими, или событиями, следующими из проявления рисков. Сам по себе риск – некая абстракция возможности и вероятности события. Выгода от управления рисками в первую очередь связана с результатами проекта, с соблюдением заданных рамок процессов выполнения работ.

Можно сказать, что любая проблема с проектом – это сработавший проектный риск. Существует вероятность, что игнорирование риска

пройдет безнаказанно, но это не означает, что это заслуга команды проекта или руководителя проекта.

Чтобы подробнее узнать о процессах управления рисками, входах и выходах в процессы, рекомендуется изучить главу «Управление рисками проекта» в РМВОК.

Источниками рисков в проекте внедрения ERP-системы являются:

- **зафиксированные границы проекта** по бюджету, срокам, содержанию – это основной источник рисков проекта, так как всегда существует вероятность выйти за ограничения. Без ограничений проекта не было бы рисков вообще, но, исходя из определения проекта, он в принципе – ограниченное предприятие для достижения конечных целей за конечное время;
- **заинтересованные стороны** (их требования и ожидания) – заказчик проекта может отказаться принять сделанную исполнителем работу, т.к. система не решает его целевые задачи (или решает не так, как ожидалось); заказчик четко не представляет, чего хочет; два ключевых сотрудника озвучивают прямо противоречащие друг другу требования, при этом от исполнителя ожидается, что система будет покрывать такие требования;
- **технические источники рисков** – используемые программные и аппаратные технологии, «технический долг» (неоптимальный код, ограничения архитектуры системы), производительность;
- **организационные источники рисков** – использование или неиспользование проектных методологий; стабильность и достаточность финансирования проекта; участие сотрудников заказчика (приоритет работ в проекте, достаточно времени и внимания); квалификация специалистов – как со стороны заказчика, так и со стороны исполнителя; сопро-

тивление пользователей на местах; затягивание принятия решений и согласования проектных документов;

- **внешние условия** – это события за рамками проекта: действия третьих лиц и государственных органов, природные явления и прочее.

Внешние условия для проекта могут стать событием непреодолимой силы – **форс-мажором**. Это непредсказуемое событие (например, санкции для государства, изменение законодательства, стихийное бедствие, военные действия, забастовки, революции и др.), не зависящее от воли сторон, участвующих в проекте, но ведущее к невозможности исполнения договорных обязательств. Форс-мажором не признаются ситуации, связанные с коммерческим риском (изменение цен, неблагоприятная конъюнктура, изменения конкурентной среды), – такие риски должны обрабатываться по ходу проекта.

Есть риски, управление которыми затруднено, т. к. влияние на них вне полномочий команды проекта. Команда проекта не может управлять в рамках проекта такими рисками, их можно только выявить и мониторить, но не влиять на них. Такие риски сильно зависят от окружающей среды проекта, процессов организации, параметров проекта. У руководителя проекта нет достаточных полномочий, чтобы минимизировать вероятность срабатывания риска, можно только планировать мероприятия по разбору последствий от него.

Рассмотрим далее, какие риски бывают в проекте внедрения ERP-системы, более подробно.

Какие риски бывают в проекте внедрения ERP-системы

Любые проекты уникальны по-своему, даже в случае тиражирования решения, например, на филиалы холдинга, в силу того, что меняются участники, другой сезон за окном (отпуска, болезни), есть свои особенности на местах. В этом разделе рассмотрим типовые проектные риски для внедрения ERP-системы. Очевидно, это не исчерпывающий список возможных событий и состояний, которые могут происходить в жизненном цикле проекта внедрения.

Риски границ проекта:

- Ошибка в оценке трудозатрат на реализацию и конечных сроков – срыв сроков, неверная оценка бюджета, недостаток финансирования работ, как следствие – риски потери части специалистов.
- Постоянные изменения бизнеса заказчика. Меняется организационная структура компании, увеличивается или уменьшается число отделений, меняется продуктовая линейка, и т. д. Зафиксированные границы проекта перестают покрывать реальные потребности бизнеса, нужно оперативно регистрировать и планировать изменения.
- Задержки в выполнении другими подрядчиками работ (соблюдение ими своих установленных границ проекта), которые необходимы для начала или завершения работ по проекту.
- Качество и соответствие требованиям предоставляемого подрядчиками результата работ, влияющие на возможность продолжения работ (интеграции и доработки) и в результате – на сроки проекта.

Риски со стороны заинтересованных сторон:

- Не формализованы процессы, нет требований к автоматизации, только общая потребность.
- Работа нескольких команд (или команды разработчиков заказчика) без согласования общей архитектуры.
- Не так поняли (коммуникационный риск) – риск неудовлетворенных ожиданий заказчика. Работает в обе стороны (заказчика и исполнителя): подразумевалось одно, поняли другое, сделали как поняли (с каскадным усилением риска на коммуникации внутри команды – например, разработчик понял иначе консультанта, который по-своему интерпретировал слова ключевого пользователя, – принцип «испорченного телефона»). Нужна единая правильная версия постановки задачи, но даже документация в виде ТЗ не всегда помогает, если оставляет места неоднозначных трактовок.

- Уход ключевого специалиста или замена команды (актуально для команды заказчика и исполнителя). Риск ухода инициатора проекта приводит, как правило, к закрытию или к существенному торможению проекта. Риск ухода руководителя проекта со стороны заказчика приводит к тому, что не выделяются ресурсы для выполнения проекта – как в финансовом плане, так и в человеческом (нет возможности поставить задачу сотрудникам заказчика).

Технические риски:

- Прочеты в архитектурном проектировании системы и ее компонентов – может привести к неработоспособности системы в сборе при массовой работе пользователей.
- Сложности с интеграцией с другими системами. Заявленные на уровне проектирования связи систем между собой на практике нереализуемы или требуют значительно больше трудозатрат и времени.
- Проблемы с подключением аппаратного обеспечения и т.д. Задержки на поставку или пусконаладку оборудования.
- Особенности реализации функциональности в конкретной ERP-системе.
- Слабая аппаратная часть для серверов тестирования и разработки. Зависания, задержки в отладке, общее замедление процесса (и эмоциональное негодование команды проекта). Тестирование выявляет проблемы, которых нет на рабочем сервере, ложные задачи и усилия по оптимизации некритических мест.

Организационные риски:

- Обучение пользователей и отторжение ими системы. Мотивация и эмоциональная составляющая проекта внедрения и ожиданий от системы сотрудниками, которым с этим работать дальше.
- Саботирование работ сотрудниками заказчика. Игнорирование административных процедур, прикрываясь другими задачами, неучастие в обучении и опытной эксплуатации. Без полноценных и своевременных данных система не сможет работать: «Мы не работаем в системе, т. к. в ней ничего нет» – порочный замкнутый круг, в системе ничего не будет, если в ней не начать работать.
- Нет времени у пользователей на изучение системы. Предполагается, что проект идет фоном к основной работе сотрудников, но, чтобы стать пользователями, они должны приложить определенные усилия, иметь запас времени на знакомство с системой.
- Пользователи не могут пройти аттестацию (не изучают систему, не хватает квалификации). Итерации на аттестацию, сдвиги сроков или повышенная нагрузка на поддержку в момент начала работ пользователей без должных навыков, риски неверных данных.
- Начальные данные в текущей системе имеют недостаточную детализацию или ошибки, сложности с инициализацией данными внедряемой системы.
- График оплат и его соблюдение. Исполнитель, живущий за счет финансирования выполняемого проекта, несет риски своевременности поступления платежей (это при успешных и вовремя закрытых актах фазах проекта). Возникновение кассовых разрывов и задержки в оплате сотрудникам могут повлечь потерю квалифицированного персонала (не будут без оплаты работать).
- Усталость и демотивация специалистов. В попытке уложиться в сроки – длительный период авралов и сверхурочной работы,

потеря внимательности, рост вероятности ошибок; эмоциональные реакции, межличностные конфликты.

- Наложение периода выполнения фаз проекта на сезон отпусков или болезней: связанные задачи сдвигаются по цепочке, ждут исполнителей или перепоручаются другим исполнителям, которым задачу делать сложнее – нужно больше времени, ниже качество.
- Утверждение технических заданий «не читая». Не выделяется достаточно времени (или, скорее, внимания специалистов, т. к. календарно времени может быть достаточно) на согласование документации проекта, затягивается процесс согласования. Формальное согласие с ТЗ по принципу «потом в процессе работы доделаем, как нужно». Задержки согласования со стороны заказчика приводят к простоям специалистов исполнителя.
- Отказ от согласованных технических заданий, по которым ведется (или уже завершена) разработка. Запросы на изменения функциональности (уровня «без этого мы работать не будем»), по которой уже прошло обучение и которая перенесена на рабочую базу и готова к работе.
- Риск привнесения ошибок в рабочую базу, если не разделены базы на разработку-тестирование и работу.
- Функциональные разрывы и противоречия в требованиях: системный архитектор не анализирует концептуальные проекты от разных консультантов по разным блокам, но конфликты возможны на стыках (общих справочниках и документах).
- Регрессионное тестирование при постоянных доработках не может завершиться, приходится начинать заново, тесты (тем более ручные) не могут пройти. Альтернатива – вообще нет регресс-тестирования, запускаем «кота в мешке», что само по себе риск.

Риски внешних условий. На что, как правило, нет возможности повлиять со стороны управления проектом внедрения – работаем с тем, что есть:

- изменение законодательства;
- доступность квалифицированного персонала;
- территориальная структура организации заказчика;
- природные явления.

Профилактика и как бороться с проявлениями рисков

Нельзя сидеть и ничего не делать до появления события риска, наоборот, необходимо предпринять корректирующие действия, направленные на то, чтобы риск либо не овестествился, либо его овестествление имело меньшие последствия для проекта.

Народная мудрость о профилактике рисков: «Кабы знал, где упасть, так соломки бы подостлал».

При этом разумность усилий тоже важна: затраты на ослабление рисков должны быть не больше затрат на устранение последствий рисков. Нужно определить, какими рисками мы будем управлять, а за какими только наблюдать. Для этого нужно:

- определить триггеры для управляемых рисков (критерии и симптомы срабатывания);
- составить план действий по ослаблению для каждого риска;
- назначить ответственных за управление рисками в целом и/или по каждому риску в отдельности.

Инструменты для управления рисками – это в первую очередь шаблоны проектной документации и использование проектной методологии, где предусмотрены:

- устав проекта – задает регламент управления рисками проекта;
- отчет о состоянии проекта – регулярно составляется и имеет раздел текущих проектных рисков;
- журнал рисков – сводный перечень рисков как часть отчета о состоянии проекта;
- карточка риска – каждый значимый риск описывается отдельно.

«Врага нужно знать в лицо» – в нашем случае возможные риски нужно заранее хотя бы проговорить, зафиксировать и учесть в планировании работ. Если сформулированы возможные риски проекта, оценены их вероятность овеществления и степень воздействия на проект, то можно рассчитать, какой резерв (сроки, финансы, персонал, дополнительные командировки, оборудование и т.д.) нужно заложить в проект, чтобы минимальными усилиями предотвратить и/или устранить последствия рисков.

Устав проекта – сам по себе минимизатор рисков, т.к. фиксирует и вынуждает соблюдать проектную методологию. Само понимание, что есть риски и ими нужно управлять – уже хорошо. Не будет сюрпризов и пущенного на самотек хода проекта.

Даже в гибких методиках есть риск-менеджмент, но он идет сессиями по принятым квантам разработки, а не целиком на весь проект.

Управление рисками осуществляется за счет использования:

- **экспертных оценок** – прошлый опыт и профессиональные суждения специалистов;
- **анализа данных** – текущие и перспективные состояния проекта и что на него влияет;

- **совещаний** – обмен мнениями, мозговые штурмы, единое понимание решаемых задач и проблем в них;
- ведения **отчета о состоянии проекта** – сам факт его составления и периодическое заполнения раздела рисков и текущих проблем позволяет снизить влияние рисков и своевременно принимать управленческие решения для корректировки состава и последовательности работ по проекту.

Согласно РМВОК, возможны четыре метода реагирования на риски:

- уклонение от риска;
- передача риска;
- снижение риска;
- принятие риска.

Уклонение от риска предполагает изменение плана управления проектом таким образом, чтобы исключить угрозу, вызванную негативным риском, оградить цели проекта от последствий риска или ослабить цели, находящиеся под угрозой (например, уменьшить содержание проекта).

Некоторых рисков, возникающих на ранних стадиях проекта, можно избежать при помощи уточнения требований, получения дополнительной информации или проведения экспертизы. Например, уклониться от риска можно, если отказаться от реализации рискованного функционального требования или самостоятельно разработать необходимый программный компонент вместо ожидания поставок продукта от субподрядчика.

Передача риска подразумевает переложение негативных последствий угрозы с ответственностью за реагирование на риск на третью сторону. Передача риска просто переносит ответственность за его управление другой стороне, но риск при этом никуда не девается.

Частый пример такого подхода в ИТ-проектах, даже для fixed price проектов, – переложить риск на заказчика. Это достигается за счет следующих возможностей:

- обосновать, что нужен отдельный бюджет на предпроектное обследование, в котором найдутся ответы на неизвестные вопросы (технические, организационные, методологические), и как следствие – риск неопределенности перестанет существовать;
- составить перечень рисков, сделать их оценку и в явной форме озвучить заказчику, что в случае наступления определенных событий потребуется дополнительный бюджет на проект. Если следовать здравой логике, то заказчик должен оставить резерв на известные риски. Исполнитель может явно прописать в договоре критерии увеличения бюджета проекта или заложить такие риски сразу в бюджет проекта (что даже выгоднее исполнителю, но не заказчику, так как бюджет вырастет, а риск может не сработать).

Снижение риска предполагает понижение вероятности и последствий негативного события до приемлемых пределов. Принятие предупредительных мер по снижению вероятности наступления риска или его последствий часто оказывается более эффективным, нежели усилия по устранению негативных последствий, предпринимаемые после наступления события риска.

Так, например, раннее решение архитектурных задач (проектируем архитектуру решения до начала активной разработки) существенно снижает технические риски. Регулярная демонстрация промежуточных результатов заказчику может снизить вероятность риска его неудовлетворенности конечным результатом. Если в проектной команде высокая текучка сотрудников, то введение на начальной стадии в проект дополнительных (избыточных) людских ресурсов снижает потери при увольнении некоторых членов команды, поскольку не будет затрат на адаптацию новых участников и риска задержки работ.

Принятие риска означает, что команда проекта осознанно приняла решение не изменять план проекта в связи с риском или не нашла подходящей стратегии реагирования на него.

В процессе работы с выявленными рисками нужно их проанализировать и выработать пути минимизации и устранения последствий. Характеристики рисков, по которым можно проводить такой анализ, согласно РМВОК:

- **Срочность.** Период времени, в течение которого меры реагирования на риск должны быть осуществлены, чтобы они дали ожидаемый результат. Короткий период показывает высокую срочность.
- **Близость.** Период времени до того, как риск может оказать влияние на одну или несколько целей проекта. Короткий период показывает высокую степень близости.
- **Латентность.** Период времени, который может пройти после наступления риска до обнаружения его воздействия. Короткий период показывает низкую степень латентности.
- **Управляемость.** Насколько просто владелец риска (или организация – владелец риска) может управлять наступлением или воздействием риска. В случаях, когда управление не представляет особой сложности, степень управляемости является высокой.
- **Контролируемость.** Степень, в которой владелец риска (или организация – владелец риска) способен контролировать последствия риска. В случаях, когда контроль последствий риска не представляет особой сложности, степень контролируемости является высокой.
- **Выявляемость.** Насколько просто можно выявить и опознать признаки наступления или высокой вероятности наступления риска. В случаях, когда наступление риска можно обнаружить без особого труда, степень выявляемости считается высокой.
- **Сопряженность.** Степень, в которой риск связан с другими индивидуальными рисками проекта. В случаях, когда риск

связан с несколькими другими рисками, степень сопряженности является высокой.

- **Стратегическое воздействие.** Потенциал риска оказать позитивное или негативное воздействие на стратегические цели организации. В случаях, когда риск может оказать значительное воздействие на стратегические цели, степень стратегического воздействия является высокой.
- **Восприятие.** Степень значимости риска с точки зрения восприятия по крайней мере одной или несколькими заинтересованными сторонами. В тех случаях, когда риск воспринимается как очень значительный, его восприятие считается высоким.

Далее рассмотрим несколько ситуаций, которые возникают на практике, и то, как бороться с такими рисками.

Срывы сроков – как с ними бороться?

Том ДеМарко в своей книге утверждает: «У проекта должно быть два срока сдачи: запланированный и желаемый. Эти сроки должны быть разными».

Полезным инструментом руководителя проектов со стороны исполнителя для минимизации рисков затягивания сроков является использование разных оценок сроков и дедлайнов (дат, когда все должно быть готово):

- для руководства заказчика – на уровне договора фиксируется пессимистичный срок;
- своего руководства (исполнителя) – озвучивается реалистичный срок;
- для команды проекта (конечных исполнителей) – задается оптимистичный срок как целевой.

Оптимистичный целевой срок работает как часть мотивационной схемы и не дает по закону Паркинсона («работа занимает все выделяемое на нее время») впустую тратить сроки проекта. Резерв на риски будет реальным резервом для команды, а не запасом для неспешной работы с первого дня по принципу «времени много, успеем», но уже без возможного запаса, когда в последний момент никакие усилия не позволят доделать задачу в срок.

Для борьбы за соблюдение сроков нужно вводить контрольные сроки на согласование документов. Прописывать их в уставе проекта (можно разные в зависимости от типа документа). Иначе заложенные, например, 3 дня на согласование технического задания заказчиком «пролетят», и заказчик может не выдать документ, а срок проекта «поедет» от этого у исполнителя (нельзя начать разработку без согласования ТЗ). Любые срывы сроков нужно считать в обе стороны (не только исполнитель затягивает свои задачи, но и заказчик тоже). Наличие контрольных сроков по документообороту проекта стимулирует сотрудников заказчика не откладывать задачи проекта на потом. Ведь все задержки будут учтены и могут в результате сдвинуть срок проекта на сумму таких задержек без санкций исполнителю за провал сроков. Ну, или по крайней мере у исполнителя появится рычаг воздействия, если сроки «поплывут» (по разным причинам), но уже не только по вине одного исполнителя.

Нужно отметить, что нарушать сроки также склонны и сотрудники заказчика, т.к. для них проектные ERP-задачи не профильные и формат проектной работы для них может быть в новинку.

Еще один инструмент минимизации просрочек – не терять время, когда договориться в рабочем порядке не получается (это просто подвесит вопрос и всю ситуацию, это уже сработавший риск коммуникации, но пострадает срок проекта) – нужно использовать эскалацию проблемы в управлении рисками для минимизации последствий. Например, конфликты и поиск компромисса на более высоком уровне кураторов проекта, если в рабочем порядке руководителя проекта со стороны исполнителя и заказчика не могут договориться.

Конечно, необходимо добавлять дополнительное время, называемое резервом на непредвиденные обстоятельства (временной резерв или буфер), в общий план-график проекта для смягчения последствий овеществления рисков, которые могут повлиять на ход выполнения работ по проекту. Как оценивать срок и бюджет проекта, рассматривалось в предыдущей главе, важно понимать необходимость управления рисками (и знать сам перечень рисков) на момент оценок проекта.

В части согласования и утверждения документов есть риск «включить заднюю передачу» («я этого не согласовывал!») – он снимается прописанным в уставе проекта форматом утверждения документов и их защиты. Если реальные подписи на бумажных документах не всегда удобны (много бумаги, удаленно документы сложно подписывать), то можно использовать цифровые подписи.

Когда в проекте участвует несколько сторон (заказчик, подрядчик и субподрядчики), у каждой из них возникают «свои» риски. Каждый, на первый взгляд, платит сам за «свои» риски, и возникает иллюзия разделения ответственности за «свои» риски. Основным принципом в этой ситуации является ответственность за риск, приписываемая той стороне, которой придется расплачиваться при нежелательном результате, вызванном осуществлением данного риска.

Проект, в котором участвуют несколько сторон, обязательно столкнется с конфликтом интересов. Нужно следовать принципу: «Мы находимся по одну сторону баррикад. По другую сторону находится сама проблема».

Без установленного управления рисками искусные переносы ответственности за риск частенько могут пройти незамеченными, что приводит к «перебрасыванию» рисков между участниками проекта, но никак не улучшает управления ими. Поэтому должны быть известны ответственные за риски, чтобы сделать этот процесс понятным и прозрачным.

При реализации проекта нужно помнить, что все «в одной лодке, потому что:

- если овеществился «риск заказчика», то деньги за проект не дополучит исполнитель;
- если овеществился «риск подрядчика», то больше денег за проект заплатит заказчик за устранение последствий (или замену подрядчика);
- если овеществился «риск субподрядчика», то отвечать за результат перед заказчиком будет подрядчик (основной исполнитель).

Нет смысла осуществлять управление рисками только руководителю проекта, когда все остальные исправно используют принцип «будет сделано», без оглядки на реальные возможности и ситуацию в целом. Поддерживая перечень рисков и количественно оценивая неопределенность, этот одинокий руководитель («борец с ветряными мельницами») добьется лишь того, что в конце концов станет выглядеть паникером и вечным пессимистом в глазах всех остальных участников проекта. Говорить правду в обстановке, где нормой является показной оптимизм (часто ложный, сознательно или несознательно), – значит оказаться в крайне невыгодном положении.

В известной сказке только маленький мальчик (без груза ответственности и обязательств) рискнул выкрикнуть: «А король-то голый!» Тогда как все придворные и даже народ боялись санкций за такую правду.

Если руководитель проекта утверждает, что есть лишь 10 %-ная вероятность сдать проект в желательный заказчику срок, то он предоставляет возможность компании-конкуренту сказать: «Поручите это нам, и мы все сделаем вовремя». Поэтому пагубная практика тендеров на понижение сроков и ставок встречается в нашей жизни, а что при этом закрываются

глаза на риски или вообще проектные методологии, вторично. Но такие проекты редко становятся успешными и достигают всех изначальных целей в заявленные сроки, бюджет и при должном качестве.

Рассмотрим практический пример риска коммуникации между исполнителем и заказчиком по срокам проекта подробнее.

Руководитель проекта со стороны исполнителя понимает объем работ и потребные сроки, и они не соответствуют ожиданиям заказчика. Сообщение заказчику о том, что его пожелания о начале эксплуатации системы «с нового года» являются призрачными и недостижимыми, наверняка будет встречено фразой: «Мы найдем более профессиональных подрядчиков!» Прямой обман из серии «будет сделано» – приведет к ожидаемому скандалу в конце проекта, когда цель не будет достигнута. Что делать?

В такой ситуации детальный план проекта и объективный перечень рисков с планами по их ослаблению – практически единственный способ аргументированно добиться взаимопонимания по части сроков, когда действительно может быть выполнен проект. Голословные заявления со ссылками на интуицию аргументами быть не могут, а вот прилежно оформленная проектная документация с оценками иерархической структуры работ позволит поискать компромиссы: снять (перенести в другой этап) часть требований, изменить очередность разработки, увеличить команду (и бюджет, при фиксации сроков). Такой конструктивный диалог также покажет несбыточность решения задачи силами других подрядчиков, когда они без оснований заявляют обратное, чтобы войти в проект.

Следующая проблема – система готова, но не начинает эксплуатироваться: нет начальных данных, пользователи игнорируют новую систему. Что такое работающая система? Для заказчика это люди, которые работают в системе, а не сама система и пусть даже успешный проход ей тестов на прототипе. А на людей исполнитель особо не влияет, даже проводя обучение. Нужны регламенты и административные рычаги со стороны заказчика и кто-то «с кнутом и пряником» (не исполнитель, а руководитель проекта от заказчика). Иначе система «не взлетит». Эти риски исполнителю лучше явно фиксировать как риски заказчика и зону его влияния. Иначе при неоднозначном критерии успеха проекта из него может не быть выхода для исполнителя: просто будет

идти время, система (готовая) будет ждать своих пользователей, а они будут ходить мимо, или начальные данные, которые должен предоставить заказчик, все никак не предоставляются, а заказчик будет только торопить исполнителя (а еще и на штрафы/пени за просрочку поставит). Из этого капкана должен быть заранее оформленный в документах выход. Система проходит все тесты, принимается центром компетенции заказчика, если нет – должно быть обоснование причин для возможности их устранения. А работает система или нет на людях, при всей эгоистичности такого похода – это «дело рук самих утопающих», то есть заказчика. Для этого у руководства со стороны заказчика есть свой инструментарий, например личный пример – использование системы топ-менеджерами с самых ранних этапов внедрения. Отчеты по эффективности менеджеров строятся из системы, где должны вводиться все данные (мотивация менеджеров вводить все свое оперативно). Отказ от приема отчетов и документов, подготовленных вручную (в старой системе), с определенного момента и прочие возможности.

Существуют риски проектной команды, ее квалификации, коммуникации и распределения задач в ней. Проблемы, возникающие в связи с наличием большой команды с самого начала проекта (а не постепенным наращиванием ее на фазах проекта):

- Участие большой команды с первого дня проекта, когда еще нет всей архитектуры: каждого участника нужно озадачить чем-то, а это не всегда возможно и рационально.
- Нет единой картины, но уже что-то делается – потом придется переделывать или вообще отказаться от результатов работ.
- Дополнительные собрания по уточнению, что и как делать, отвлекают аналитиков и системного архитектора от задач фазы анализа и дизайна, затягивают ее.

Профилактика этого риска – в планировании привлечения специалистов на конкретные задачи, свои на разных фазах проекта: динамический состав участников.

Сработанность и эффективность команды – важный фактор успеха (или неуспеха) проекта внедрения ERP-системы. Рекомендации, которые дает Том ДеМарко в книге «Deadline. Роман об управлении проектами»:

- Не пытайтесь создавать новые команды без необходимости, поищите в коллективе уже сложившиеся и сработавшиеся команды.
- Оставляйте команды работать вместе и после окончания проекта (если они сами того хотят), чтобы у пришедших вам на смену руководителей было меньше проблем с плохо сработавшими командами.
- Считайте, что команда, которая хочет продолжать работать вместе и дальше, – это одна из основных целей любого проекта.

Когда команда – «сборная солянка», еще и с использованием удаленного взаимодействия, то управлению ей нужно уделять особое внимание.

Простои команды из-за болезней в сезон простуд могут минимизироваться профилактическими прививками, поощрением занятий спортом и правилом лечения дома, когда заболел (при этом можно работать удаленно, у кого есть возможность по состоянию здоровья), чтобы не разносить инфекцию по офису, заражая коллег.

Риски, связанные с проектированием архитектуры и разработкой, минимизируются единым ответственным за архитектуру всей системы. Иначе возможны наложения функциональных разрывов:

- В команде специалисты описывают разные блоки, но все хотят поменять, например, справочник «Договоры». При этом хотят изменить по-разному (закупщикам и продавцам нужно разное) – это противоречия на уровне требований заказчика, но это может пройти, т.к. занимаются разные специалисты исполнителя.
- Нужно сводить все изменения воедино, и системный архитектор должен проверить их на непротиворечивость. Если противоречия есть, то ключевые сотрудники заказчика должны договориться между собой (работа с требованиями

и ключевыми сотрудниками), и только потом систему настраиваем и модифицируем под сводные требования.

- Если эту работу не проводить, то проблема выявится на уровне разработки или еще позже, в процессе эксплуатации.

Бывают конфликты внутри организации заказчика – например, бухгалтерия против ERP-системы как небухгалтерской парадигмы учета, потери монополии влияния на учет (на первую роль выходят оперативный контур и первичные документы, которые вводят «небухгалтеры»). Этот тот самый конфликт интересов, который был описан в первой главе. Профилактикой этого риска может быть разъяснительная работа, мотивирование сотрудников, возможность настройки контроля (аудита) всех вводимых менеджерами документов, которые попадают в бухгалтерский учет, закрытие периода и конкретных проверенных документов от изменений или контроль за этими изменениями. В 1С:ERP есть специальное рабочее место для бухгалтера, где можно проверять и защищать от изменений документы, вводимые всеми остальными «небухгалтерами», – его демонстрация на ранних фазах проекта снимает негативное отношение к проекту у бухгалтеров, переводит их в ряды защитников проекта, т. к. система предоставляет им много полезного для ведения регламентированного учета, и компромиссы достижимы.

Эффект от проводимой профилактики рисков при видимой трате усилий (ресурсов в лице руководителя проекта) на эту задачу окупается экономией трудозатрат. Например, для проекта с командой в 10 человек расходы на управление рисками составят условно 40 человеко-часов в месяц. При этом будут идентифицироваться около 20 потенциальных или сработавших рисков текущего периода и состояния проекта. Из этих рисков минимум 5 самых важных будет подавляться комплексом мероприятий. Исходя из того, что критичный риск отбирает у проекта минимум 40 человеко-часов дополнительных усилий на переделки, исправления, доработки, то получаем от 200 человеко-часов экономии ежемесячно. Таким образом, отношение усилий на мониторинг и контроль рисков к положительной экономии усилий команды проекта явно в пользу проведения работ над рисками. Это лучше, чем получить небольшую экономию времени и бросить все на самотек, что может (и это обязательно произойдет) привести к задержкам проекта, конфликтам и общей неудовлетворенности процессом или результатами работы.

Когда риски можно игнорировать

Существуют три основных свойства конкретного риска, которые оправдывают «неуправление» этим риском и его игнорирование:

- вероятность овеществления риска настолько мала, что можно ее игнорировать;
- при овеществлении риска обесмысливается цель проекта;
- ослабление риска стоит значительно дороже, чем устранение последствий риска.

Причем критерии могут учитываться по одному, в комбинациях или все вместе.

Наглядно ситуацию можно продемонстрировать на примере риска падения метеорита – «метеоритный риск».

В сети и СМИ несколько лет назад был значительный всплеск к этой теме после падения метеорита под Челябинском. В страховых компаниях этот риск страхуется по-разному: где-то выделен явно, где-то входит в «прочие предметы» или «летающие объекты». И компенсацию ущерба от этого тоже выдавали/не выдавали по-разному.

Допустим, мы всерьез рассматриваем риск того, что на завод, который мы автоматизируем с помощью ERP-системы, упадет метеорит и уничтожит все результаты проекта. Проанализируем этот риск:

- по данным аналитиков, небольшие метеориты весом 2–5 килограммов падают на Землю каждые два месяца, вероятность причинения ущерба от них равна 0,00001 %. И это ущерб в принципе, а не нашему объекту, это слишком низкая вероятность для проектного риска;
- если будет уничтожен объект автоматизации, то отпадает смысл в самом проекте автоматизации;

- стоимость отслеживания метеорита и его отклонения от заданной траектории огромна и явно несоизмерима со стоимостью проекта автоматизации.

Вывод: таким риском управлять бессмысленно.

В целом можно обобщить ситуации, когда не стоит управлять рисками:

- организации (заказчик, исполнитель и подрядчики) принципиально и осознанно не поддерживают управление рисками;
- конкретным риском управлять бессмысленно («метеоритный риск»);
- у риска незначительные (относительно масштабов проекта) последствия;
- конкретный риск не является нашим риском – это чужой риск, влияния на него у нас нет.

Процедура управления рисками в проекте внедрения ERP-системы

Рассмотрим в заключение главы конкретный перечень шагов, используемых для управления рисками на проектах внедрения ERP-системы.

Еще раз кратко повторим важность управления рисками в простом списке ответов на вопрос: «Что будет, если про риски не думать в начале проекта?»:

- Все хорошо (повезло).
- Все плохо (начали срабатывать риски).
- Все очень плохо (сработавшие риски никак не решались).

Шансы того или иного сценария разные на разных проектах и зависят от многих факторов, в общем случае для проектов внедрения систем класса ERP рекомендуется рисками управлять. Процедура на проекте может состоять из следующих шагов (можно их в уставе проекта прописать):

1. **Идентификация и регистрация риска проекта.** Любой член сводной проектной команды заказчика и исполнителя может обратить внимание руководства проектом на потенциальный риск проекта.
 - Инициатор обращения о возникновении риска формулирует описание проектного риска и направляет его руководителю проекта со стороны исполнителя с копией руководителю проекта от заказчика.
 - Инициатор отражает возможное влияние данного риска на проект, в случае если он не будет разрешен. Руководитель проекта со стороны исполнителя отражает данный риск в документе **Отчет о состоянии проекта**, присваивает ему статус.
2. **Анализ риска.** Руководители проекта со стороны заказчика и исполнителя анализируют данный риск и определяют дальнейшие действия по нему:
 - Данный риск уже учтен в составе другого риска. Может потребоваться обновить информацию по существующим рискам для отражения текущей ситуации. Руководитель проекта со стороны исполнителя производит соответствующее обновление информации.
 - Данный риск требует подготовки плана действий по его устранению. Руководители проекта назначают ответственного за подготовку плана действий и сроки, доводят это решение до ответственного исполнителя. Руководитель проекта со стороны исполнителя регистрирует статус риска в «Отчете о состоянии проекта».
 - Данный риск представляется как не влияющий на проект и не требует дальнейших действий. Руководители проекта

со стороны заказчика и исполнителя принимают это решение и доводят его до инициатора. Руководитель проекта со стороны исполнителя регистрирует это решение в «Отчете о состоянии проекта».

3. **Подготовка плана действий по устранению риска.** Ответственный за риск, назначенный руководителями проекта со стороны заказчика и исполнителя, анализирует данный риск, его причины и возможные последствия, и готовит план действий по его устранению. План действий включает в себя также необходимые ресурсы и изменения в бюджете проекта, если необходимо. План действий передается на рассмотрение и согласование руководителям проекта со стороны заказчика и исполнителя.
4. **Эскалация риска** (при необходимости). В случае если данный риск находится вне зоны компетенции руководителей проекта со стороны заказчика и исполнителя, он передается на рассмотрение управляющего комитета. Управляющий комитет рассматривает данный риск и назначает ответственного (возможно, вне рамок проектной группы) за подготовку плана действий по устранению риска, либо присваивает ему статус не относящегося к проекту, либо принимает решение, определяющее действия по его устранению.
5. **Принятие решения.** Принятое решение фиксируется в протоколе заседания управляющего комитета и является обязательным для исполнения. Данное решение регистрируется руководителем проекта со стороны исполнителя в «Отчете о состоянии проекта».
6. **Периодический анализ рисков проекта.** Руководители проекта со стороны заказчика и исполнителя проводят периодический контроль (например, 1 раз в 2 недели) активных рисков проекта и процедур по их устранению (в рамках периодических совещаний по состоянию проекта).

Данный алгоритм действий позволяет выявлять новые риски (тут главное создать атмосферу доверия, чтобы члены проектной команды не скрывали доступную на их уровне информацию), контролировать и минимизировать выявленные ранее риски.

Впрочем, не так важно, как именно построено управление рисками в проекте (или оно отсутствует вовсе), он должен двигаться по своему плану к заветной цели – достижению результатов проекта. Рассмотрим в следующей главе некоторые аспекты фаз разработки и опытной эксплуатации и то, какие действия позволяют успешно пройти эти фазы.

Глава 9. Как пережить фазы разработки и опытной эксплуатации

Сколько и каких баз нужно в проекте на фазе разработки

К началу фазы «Разработка» проект внедрения ERP-системы успешно прошел путь концептуального и архитектурного проектирования, имеется настроенный прототип, но не все в нем получается покрыть текущей функциональностью системы, требуется доработать систему, устранить функциональные разрывы. К началу фазы «Разработка» считаем, что все потребные доработки выявлены и описаны в документе «Технический проект», спланирована последовательность работ, остается только все это окончательно формализовать в технических заданиях, согласовать с заказчиком и сделать.

Прежде чем начинать активную разработку, нужно задаться вопросом организации совместной работы специалистов (разработчиков и консультантов исполнителя, ключевых сотрудников заказчика), минимизировать ситуации попадания непротестированного функционала на рабочую версию к конечным пользователям.

Рассмотрим варианты инфраструктуры из информационных баз для разработки, тестирования и эксплуатации. Выбор подхода зависит от бюджета проекта и выбора проектной команды (как привыкли, как хотят минимизировать риски).

Возможны подходы от максимальных по числу используемых баз до минималистических:

- 4 базы: разработка – тестирование – мастер – рабочая;
- 3 базы: разработка и тестирование – мастер – рабочая;
- 2 базы: разработка и тестирование – рабочая.

Явно опасный в использовании вариант, когда информационная база развернута в единственном экземпляре, она же рабочая база пользователей, она же база для тестирования и разработки, детально рассмотреть не будем – этот не подход для внедрения ERP-системы.

Подход «разработка – тестирование – мастер – рабочая».

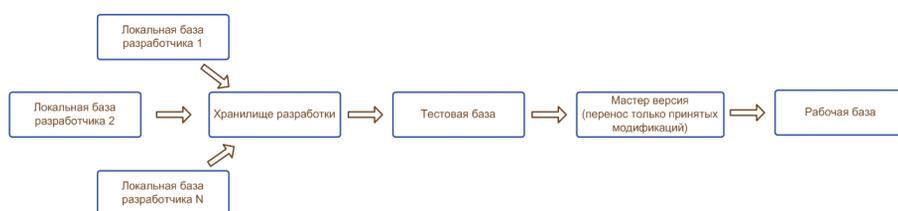


Рис. 9.1. Подход «разработка – тестирование – мастер – рабочая»

При таком способе используется 4 базы:

- **База разработки** – в нее подключаются разработчики, получают срез конфигурации из общего хранилища, закладывают туда свои доработки, устраняют конфликты и пересечения в коде с другими разработчиками, проводят отладку (возможно, запуск автотестов) и затем переносят проверенные наработки в базу для тестирования.
- **Тестовая база** – в нее подгружаются прошедшие отладку модификации, в этой базе работают консультанты, настраивают и тестируют учетную модель системы, тестируют существующий и доработанный функционал. Ключевые сотрудники заказчика изучают и тестируют новый функци-

онал. Прошедшие тестирование и утверждение модификации переносятся дальше на мастер-версию.

- **Мастер-версия** – конфигурация для консолидации прошедших тестирование и запланированных к переносу на рабочую базу в данный релиз модификаций, а также база для финализации настроек учетной модели системы. Наиболее стабильная версия в сравнении с версией тестирования, где может быть еще что-то не прошедшее тесты. «Мастер» – это будущая конфигурация рабочей базы, куда конфигурация переносится целиком, а не по избранным модификациям. На этой базе может проходить финальное тестирование переносимых модификаций и общее регрессионное тестирование на целостность замкнутого цикла учета и бизнес-процессов.
- **Рабочая база** – база в эксплуатации, в которой работают конечные пользователи с реальными учетными данными. Тестирование и разработка в этой базе под запретом.

Подход «разработка и тестирование – мастер – рабочая».



Рис. 9.2. Подход «разработка и тестирование – мастер – рабочая»

При таком подходе используется 3 базы:

- **База разработки и тестирования** – в нее подключаются разработчики, получают срез конфигурации из общего хранилища, закладывают туда свои доработки, устраняют конфликты и пересечения в коде с другими разработчиками, в этой же базе работают консультанты, настраивают и тестируют учетную модель системы, тестируют существующий и доработанный функционал. Ключевые сотрудники заказчика

изучают и тестируют новый функционал. Прошедшие тестирование модификации переносятся дальше на мастер-версию.

- **Мастер-версия** – аналогично варианту из 4 баз.
- **Рабочая база** – аналогично варианту из 4 баз.

Объединение разработки и тестирования, с одной стороны, оптимизирует работу, убирая шаг (и трудозатраты) переноса модификаций в дополнительную базу, с другой – привносит в процесс временное состояние базы, когда тестировать невозможно, т.к. «все разломано», а также потенциальные потери данных настроек для тестирования или частичную потерю данных документов или регистров.

Подход «разработка и тестирование – рабочая».



Рис. 9.3. Подход «разработка и тестирование – рабочая»

При таком подходе используется 2 базы:

- **База разработки и тестирования** – в нее подключаются разработчики, получают срез конфигурации из общего хранилища, закладывают туда свои доработки, устраняют конфликты и пересечения в коде с другими разработчиками, в этой же базе работают консультанты, настраивают и тестируют учетную модель системы, тестируют существующий и доработанный функционал. Ключевые сотрудники заказчика изучают и тестируют новый функционал. Прошедшие тестирование модификации переносятся дальше на рабочую базу.
- **Рабочая база** – аналогично варианту из 4 баз.

В этом подходе фокус предварительного тестирования и отладки может быть вынесен на локальные базы разработчиков, туда же могут заходить тестировщики (или брать копию к себе для тестов). Тогда итоговая сборка общего хранилища может быть достаточно стабильна, но в любом случае «мастер» не заменяет, т. к. может содержать лишние для переноса на рабочую базу модификации, не согласованные к переносу на рабочую базу в этот пул обновления. Поэтому при таком подходе на рабочую базу переносить целиком конфигурацию затруднительно – нужно выделять период на стабилизацию, вводить мораторий на перенос с локальных версий модификаций, которые не входят в данный релиз к переносу на рабочую базу.

Условно, при этом подходе все участники знают график, когда будет сборка версии, поэтому за некоторое время до этого происходит остановка помещения изменений в хранилище, наступает мораторий на изменения и консультанты тестируют, проводятся регресс-тесты (автоматические или ручные). Версия стабилизируется, вносятся только исправления ошибок. После отмашки, что версия стабильна, рабочая база получает изменения из хранилища разработки и тестирования и обновляется. После чего мораторий снимается до сборки следующего билда. Такой подход приводит к паузам, когда разработчики не могут поместить в хранилище конфигурации новые модификации, а тестировщики – начать их тестировать, пока идут процессы стабилизации и тестов.

Использование версии «мастер» для сборки готовых модификаций позволяет исключить ошибки ручного переноса на рабочую базу сложных модификаций, дает время на тестирование результата, а не сразу влияет на работу конечных пользователей. Ошибки уровня «ой, я не тот код перенес, а нужный вообще забыл» – это частые ошибки, связанные с невнимательностью и просто человеческим фактором. Поэтому когда на тестовой базе все работает, то это совсем не значит, что с первого раза все заработает после переноса на «мастер» или сразу на рабочую базу. Если для «мастера» есть время выявить проблемы без последствий для реальной работы и рабочих данных, то такие ошибки (пусть даже они быстро исправляются «допереносом забытого») на рабочей базе могут порождать проблемы, которые потребуют значительных усилий на ликвидацию последствий: развал данных потребует отката базы из бэкапа или сложных обработок восстановления данных. Поэтому наличие промежуточной (относительно тестовой версии) стабильной

сборки, близкой к рабочей конфигурации, – это залог минимизации таких проблем.

Во всех трех подходах заявлено, что на рабочей базе нет разработки и тестирования. При этом возможны ситуации критичных ошибок, которые исправлять по длительной процедуре (через цепочку конфигураций «разработка – тестирование – мастер» и замену всей рабочей конфигурации целиком) сложно, т.к. нет достаточно времени для этого, работа пользователей стоит. Такие исключения могут приводить к внесению исправлений (проверенных, очевидно, предварительно на тестовой версии) сразу в рабочую базу. Эти же исправления ошибок далее проходят весь путь миграции по всем базам до мастер-версии и при следующей замене конфигурации с «мастера» на рабочую базу не должны потеряться.

Некоторые ситуации, которые требуют отладки для поиска и исправления ошибок, воспроизводятся только на данных рабочей базы. Нужно снимать с нее копию и заменять периодически базу тестирования свежими данными с рабочей базы. Если параллельно с этим идет настройка данных и тестирование прототипа настроек для других этапов проекта, еще не перенесенных на рабочую базу, то такая замена данных на текущие данные из рабочей базы приводит к потере тестовых настроек. Мы приходим к тому, что баз может стать еще больше, т.к. может потребоваться иметь тестовую базу в двух версиях и нужно время на перенос настроек в обновленную рабочими данными тестовую базу из старой тестовой базы.

Рекомендуется в разных базах делать какие-то отличия: менять заголовки, цвет стилей оформления интерфейса (платформа «1С» это позволяет) – чтобы специалисты визуально, на глаз определяли, что это за база, и случайно не тестировали в рабочей базе. Также стоит отобрать полные права на рабочей базе у разработчиков и консультантов – это тоже хороший способ для защиты от «ой, я перепутал базы». Сделать специальный логин или выдавать права по необходимости. Когда баз много – их будут путать обязательно.

Чем больше баз используется, тем больше накладных расходов на миграцию данных и модификаций между ними возникает, а также растут требования к аппаратной части, особенно с ростом объема рабочей базы. Для получения копии рабочей базы для тестирования при значительном ее объеме потребуется аппаратное обеспечение для стенда тестирования такого же уровня, как для рабочей базы. Поэтому выбор того или иного подхода зависит не только от желания проектной команды, но и от возможностей бюджета проекта: в нем должно быть заложено достаточное аппаратное обеспечение, поддерживающее разработку по цепочке из нескольких баз.

В самом начале фазы «Разработка» может не быть рабочей базы: еще не закуплено оборудование, не готово к развертыванию базы в ее промышленной конфигурации. На ранней стадии фазы это допустимо. Обучение можно проводить на тестовой или мастер-базе, данных еще мало, требования к аппаратной части серверов для разработки ниже, чем для рабочей системы на множество пользователей. Но в какой-то момент рабочая база будет нужна в ее итоговой целевой конфигурации, на рабочих серверах, а задержки настройки серверной аппаратуры могут стать рисками проекта и повлиять на сдвиг сроков начала опытной эксплуатации.

Серверное оборудование заказывает заказчик, поставляет его третья сторона для проекта автоматизации, а сроки из-за задержек аппаратуры и ее настройки «поедут» у исполнителя.

При использовании облачных технологий и (или) виртуальных машин снижается зависимость от аппаратного обеспечения. Тестовые стенды и конфигурация разработки могут разворачиваться на виртуальных серверах в локальной сети или облаках. При частой поставке и сборке билдов на помощь проектной команде приходят практики DevOps.

Организация миграции данных для тестирования с рабочей базы, обновление из хранилища конфигурации разработки и тестирования, обновление рабочей базы из конфигурации мастера – все это можно (и нужно) автоматизировать соответствующими скриптами, что минимизирует ручной труд (и ошибки, связанные с человеческим фактором).

DevOps (акроним от англ. development и operations, произносится как «девопс») – набор практик, нацеленных на активное взаимодействие специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимную интеграцию их рабочих процессов. DevOps базируется на идее о тесной взаимозависимости разработки и эксплуатации программного обеспечения и нацелен на помощь организациям быстрее создавать и обновлять программные продукты и услуги.

Еще один важный вопрос – это выбор версии, на которой начинать разработку и настраивать учетную модель прототипа системы автоматизации. Например, у 1C:ERP есть ознакомительные версии, а есть поддерживаемые предыдущие под редакции системы. Для уже внедренного проекта переходить на ознакомительную версию смысла не имеет (если там не заявлен требуемый функционал, критичный для работы, конечно). А вот для нового проекта внедрения нужно выбирать и переходить на актуальную версию системы, пусть и в ознакомительном статусе. Потом будет проще обновляться, нет дополнительных работ по переводу проекта, готового к запуску, со старой под редакции в конце проекта (ее могут снять с поддержки в процессе проекта), когда все уже протестировано и запущено, и сразу доступна вся новая функциональность. Нужно соотносить планы проекта внедрения с планами выпуска новых версий ERP-системы, закладывать риски.

Стандарты разработки

В проектной команде разработчики должны соблюдать лучшие практики (best practices) и стандарты разработки, применяемые для платформы и конфигурации ERP-системы. Это позволяет делать код единообразным, обезличенным (без авторских стилистических особенностей), читабельным для любого другого разработчика, совместимым с исходным кодом ERP-системы. Соблюдение стандартов может требовать дополнительных усилий от разработчиков, т.к. «быстрый» код без соблюдения стандартов тоже может работать, но его масштабируемость и отчуждаемость для передачи другим разработчикам (команде поддержки со стороны заказчика) может быть низкой.

Система стандартов и методик разработки конфигураций для платформы «1С:Предприятие 8» (<https://its.1c.ru/db/v8std>) – это свод правил, позволяющий разработчику избежать многих возможных проблем на этапе разработки и поддержки прикладного решения. Описание стандартов является частью официальной документации по информационно-технологическому сопровождению «1С:Предприятия». Рекомендуется следовать стандартам при доработке 1С:ERP в проектах внедрения системы автоматизации.

Требование соблюдения стандартов разработки и правила именования проектных метаданных (для отделения их от штатной функциональности системы) можно прописать в уставе проекта. Руководитель проекта со стороны заказчика особенно заинтересован в этом пункте, т. к. ему потом обеспечивать дальнейшее обслуживание системы (возможно, силами других подрядчиков), а для этого код в ней должен быть «правильным».

Фирма «1С» предоставляет и постоянно развивает необходимую информацию по разработке и администрированию конфигураций на платформе «1С:Предприятие 8»:

- книги и стандарты по разработке и администрированию – <https://its.1c.ru/section/dev>;
- технологические вопросы крупных внедрений – <https://its.1c.ru/db/metod8dev#browse:13:-1:1989:2035>;
- блог разработчиков платформы «1С» – <https://wonderland.v8.1c.ru>;
- ютуб-канал платформы «1С» – «1С Platform» <https://www.youtube.com/channel/UC8C9c4GFLs6cNqBVWSM9jLw>;
- курсы и экзамены – <http://1c.ru/rus/partners/training/default.jsp>.

При разработке новой функциональности нужно помнить о так называемом «техническом долге» – когда код пишется неоптимально, задачу свою решает, но производительность или последующая масштабируемость (возможность развития и доработки) низкие. В качестве примера можно привести какой-нибудь алгоритм, который обрабатывает данные системы на протяжении часа, тогда как после оптимизации время работы

может сократиться на порядок – до 5–7 минут. Есть разница? Для конечного пользователя точно есть. Некоторые неоптимальные алгоритмы создают нагрузку на всю систему, мешая работе других пользователей, а результат работы алгоритма оперативным не назвать: он может быть доступен через несколько часов или только на следующий день.

Тестирование

Тестирование – это важная часть процесса разработки программного обеспечения и инструментарий достижения целей проекта в обеспечении должного качества модификаций и доработок ERP-системы.

Рассмотрим определение тестирования, которое дает РМВОК. **Тестирование** – это организованное и проводимое по определенному плану исследование с целью получить объективную информацию о качестве продукта или услуги, проходящих тестирование, в соответствии с требованиями проекта. Тестирование предназначено выявить ошибки, дефекты, неисправности или другие проблемы несоответствия в продукте или услуге. Тип, объем и глубина тестов, необходимых для оценки каждого требования, входят в состав плана управления качеством проекта и зависят от характера, сроков, бюджета и других ограничений проекта. Тесты могут проводиться на всем протяжении проекта, по мере того как становятся доступными различные компоненты проекта, а также в конце проекта, когда тестируются конечные поставляемые результаты. Проведение тестирования на ранних стадиях помогает выявить проблемы несоответствия и снизить стоимость исправления несоответствующих требованиям компонентов.

Подробное формальное описание процессов и методик тестирования, рекомендуемое для самостоятельного изучения, приведено в серии стандартов ISO 29119 (и ГОСТ Р по ним) «Системная и программная инженерия. Тестирование программного обеспечения»:

- Часть 1. Понятия и определения (ГОСТ Р 56920-2016/ISO/IEC/IEEE 29119-1:2013).
- Часть 2. Процессы тестирования (ГОСТ Р 56921-2016/ISO/IEC/IEEE 29119-2:2013).

- Часть 3. Документация тестирования (ГОСТ Р 56922-2016/ISO/IEC/IEEE 29119-3:2013).
- Часть 4. Методики тестирования (ISO/IEC/IEEE 29119-4:2015).

Далее рассмотрим основные моменты и понятия, которые нужно учитывать в проектах внедрения ERP-системы.

План тестирования – приложение к ТЗ, содержит входные и ожидаемые выходные данные (или действия) для функциональности системы для проверки работоспособности алгоритмов и соответствия их исходным требованиям к системе. Разработка плана тестирования позволяет лучше понять требование и его реализацию, т. к. заставляет формально описать данные на входе в функциональность и ожидаемые данные на выходе из нее. Такая информация позволяет разработчику лучше понять требования и отладить свой код и алгоритмы. План тестирования должен покрывать сценарии использования системы в бизнес-процессах компании. Успешное прохождение тестов (ручных или автоматических) – это критерий отметки успешности для данной функциональности в общем чек-листе тестирования.

Формирование плана тестирования модификации системы позволяет подключить процесс тестирования на стадии проектирования, до активной разработки модификации, и проверить «в уме» описанную в ТЗ функциональность. Иногда при подготовке плана тестирования вскрываются недостатки проектирования, когда в алгоритм не заложены значимые сценарии работы пользователей. Пренебрежение планом тестирования создает иллюзию экономии на проекте, но все в головах не удержать: разработчик закладывал что-то одно, тестировщик проверял совсем другое, а пользователь будет ожидать что-то третье. А план тестирования, как часть ТЗ, утверждается и является критерием приемки работоспособности функциональности, одинаковым для всех специалистов исполнителя и заказчика.

Специалистам заказчика порой проще понять и утвердить само ТЗ по описанию его плана тестирования, т. к. они как пользователи оперируют понятиями данных и действий с системой, а не ее алгоритмами и тонкостями их реализации.

Чек-лист тестирования системы – сводный высокоуровневый список спецификаций возможностей системы, необходимых для покрытия автоматизируемых бизнес-процессов компании. Является критерием приемки работоспособности системы для эксплуатации и, заполненный успешно пройденными тестирование пунктами, необходим для завершения фазы «Разработка». Причем отмечать (или хотя бы проверять достоверность) факт успешного прохождения должен руководитель заказчика как принимающая работу сторона.

Ручное тестирование – тесты выполняются вручную, без использования особых средств автоматизации. Тестировщик просто следует заранее написанному плану действий, фиксируя результат: было поведение функциональности системы ожидаемым или нет (в этом случае записывается, что пошло не так). Если детального плана тестирования нет, то тестировщик проявляет креатив, представляет себя на месте пользователя и моделирует его работу в различных сценариях по бизнес-процессам. Качество теста в этом случае сильно зависит от квалификации специалиста.

Автоматизированное тестирование – потребность в нем возникает, когда проект большой и ведется его активное развитие, чтобы не тратить время на повторный ручной прогон тестов, которые приходится повторять регулярно. Часто под автотестами понимают только юнит-тесты, хотя автоматизировать можно практически любой вид тестирования.

Unit testing – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Юнит-тесты пишутся разработчиками для своего кода, проверяют код, передавая входные параметры и ожидая (проверяя) результат.

Нужно учесть, что часто бывает так, что затраты на поддержание автотестов значительно превышают затраты на ручное тестирование. Поэтому надо действовать аккуратно, оценив плюсы и минусы: какие-то редкие (уникальные) тесты оставить ручными, а автоматизировать массовые, повторяемые периодически тесты, например набор тестов для регрессионного тестирования.

Регрессионное тестирование – это проверка работоспособности системы после внесения модификаций и доработок, направленная на подтверждение того, что функциональность системы с новыми изменениями работает так же, как и раньше, без ухудшений. Тестирование затрагивает всю используемую функциональность, а не только модифицируемые в рамках текущей поставки (сборки релиза) участки. Необходимость проверки на возникновение ухудшений (собственно, самого регресса системы, от лат. regressio – «движение назад») обусловлена возможностью возникновения ошибок в коде, который не планировался к изменению, после исправления ошибок или добавления новых возможностей.

Бывает, что разработчики при исправлении какой-то ошибки или модификации системы в одном месте привносят ошибки в другие места. Такие ошибки не находятся ручными тестами по плану тестирования данной модификации, т.к. проявляются вне контура, который тестируется.

Поэтому требуется проверка на «неразвал» всей системы, так как локальные тесты модификаций могут успешно проходить, а места, которые никто специально не трогал, могут перестать работать корректно. Перед выпуском очередной сборки для переноса на рабочую базу конфигурацию пропускают через набор тестовых сценариев, подготовленных ранее. Если сами тесты не проходят по причине запланированного изменения функциональности, а не ошибок системы, то тесты нужно актуализировать.

Смоук-тест (англ. smoke testing, «дымовое тестирование») – минимальный набор тестов на явные ошибки. «Дымовой тест» обычно выполняется самим разработчиком в процессе отладки, или это минимальный набор автотестов проверки на неразвал (система «задымилась»). Не прошедшую этот тест систему не имеет смысла отдавать на более глубокое тестирование, нужно оперативно исправлять.

Нагрузочное тестирование (англ. load testing) – это вид тестирования производительности, проводимый с целью оценить поведение системы (или какой-то ее функции) под увеличивающейся нагрузкой (число

одновременно работающих пользователей и/или количество транзакций). Нагрузочные тесты требуют создания обработок по генерации массива данных, которые будут создавать нагрузку для проверки алгоритмов системы. Без тестов этого типа неоптимальный код ручными тестами на малых выборках данных выявить сложно, опытная эксплуатация тоже пропустит такую функциональность, т.к. за несколько месяцев наработать критическую массу данных, когда начнет проявляться проблема, возможности нет. Массовую работу пользователей можно проверить симуляцией работы или практикой, организовав нагрузку на систему путем договоренности со всеми пользователями о выполнении ими их основных бизнес-процессов (эмуляция активной работы: ввод документов, построение отчетов) в системе в одно и то же время. В этот момент следим за показателями нагрузки на аппаратуру (процессоры, диски, память серверов) и мониторим состояние СУБД.

Игнорирование нагрузочных тестов (экономия на них, т.к. они требуют трудозатрат, а это сроки и бюджет) является «отложенным возмездием» или проектным риском, который может проявиться через годы использования системы (а может и не проявиться, если код изначально оптимальный или прирост данных не влияет на алгоритм), когда в системе накопится критический объем данных. Тогда нужно будет заниматься рефакторингом и оптимизацией алгоритмов. И эта задача будет уже на этапе сопровождения системы, за рамками жизненного цикла проекта внедрения.

Рассмотрим подробнее вопрос, для чего нужно автоматизировать процесс тестирования и какие инструменты для этого возможны. Цели автоматизации тестирования:

- повысить эффективность внутреннего тестирования;
- уменьшить количество и критичность внешних ошибок (найденных конечными пользователями);
- быстрый и точный интеграционный тест (на стыке разных подсистем и модулей), как регресс-тест функционала на неразвал, при внесении изменений;
- ускорить приемочное тестирование.

Автоматизация тестирования требует усилий: создание автотеста может быть в 2–3 раза более трудозатратно, чем проведение аналогичного ручного теста. Но если этот ручной тест нужно будет повторять 4–10 раз, то выигрыш от автотеста очевиден. Автотест может идти значительно быстрее, чем это делает человек, хотя сам автотест может эмулировать работу пользователя (кликает в интерфейсе системы) в реальном времени, но делает он это быстрее человека. Автотесты могут запускаться на ночь (так как множество тестов могут выполняться несколько часов) и высвободить консультантов для других задач.

Инструментарий для автоматизации процесса тестирования проектов на 1С:ERP:

- 1С:АПК <https://v8.1c.ru/acc/> – проверка кода на соответствие стандартам разработки;
- 1С:СППР <http://v8.1c.ru/model/> – сценарные автотесты;
- 1С:КИП <https://v8.1c.ru/expert/etp.htm> – нагрузочные тесты, оценка масштабируемости системы.

«1С:Автоматизированная проверка конфигураций» предназначена для автоматизированной проверки конфигураций, разработанных на платформе «1С:Предприятие 8», на соответствие стандартам и иным требованиям технического характера. АПК существенно расширяет платформенную проверку конфигурации и выполняет статический анализ технического качества конфигураций в автоматическом режиме, не требуя их запуска. Тестирование кода позволяет еще до тестирования самой функциональности отловить и устранить ряд проблем.

Это отдельная конфигурация для проверки других конфигураций, она может находить и регистрировать ошибки по коду конфигурации. АПК может интегрироваться с 1С:СППР и передавать найденные ошибки туда.

Типы проверок можно гибко настраивать. Какие-то ошибки могут быть просто особенностью – можно отметить их как исключение, и АПК не будет больше уведомлять об этой ошибке.

Для тестирования своего проектного кода на соответствие стандартам разработки нужно предварительно выполнить проверку АПК стандартной конфигурации и отметить все найденное как особенности, потом запускать проверку модифицированной конфигурации. Это позволит получить сообщения о проверках только по проектному коду.

«1С:Система проектирования прикладных решений» содержит возможности автоматизации тестирования: подготовка, выполнение сценариев тестирования и регистрация ошибок по результатам выполнения сценария. Поддерживаются сценарные тесты по методологии BDD (англ. Behavior-Driven Development, разработка через поведение). Для пользователей и партнеров 1С:ERP доступна также специализированная демонстрационная база, включающая в себя интегрированный инструмент «Тест-центр» и необходимые дополнительные тестовые обработки с готовыми тестовыми сценариями.

«1С:Корпоративный инструментальный пакет» предназначен для повышения производительности, масштабируемости и надежности информационных систем, работающих на платформе «1С:Предприятие 8», за счет решения широкого круга технических задач, возникающих на всех этапах жизненного цикла информационной системы.

Отдельный вопрос, который стоит рассмотреть в завершение раздела, – это прозрачность прохождения процесса тестирования для заказчика. Специалисты исполнителя используют для организации процесса регистрации ошибок и отслеживания статусов их исправления систему баг-трекера (например, тот же 1С:СППР, в котором эта опция есть). Нужно ли предоставлять доступ к ней сотрудникам заказчика? Если руководитель проекта со стороны заказчика хочет полностью контролировать ход процесса, то такой доступ он может запросить сам.

Организацию сбора ошибок от пользователей рекомендуется осуществлять с использованием баг-трекера, но с разделением доступа на внутренние ошибки команды исполнителя (их не видно пользователям) и внешние от самих пользователей (плюс им видны внутренние ошибки, отмеченные как внешние). Тогда до момента начала обучения пользователей процесс тестирования и ошибки, найденные в нем, от конечных

пользователей скрыты, им дается готовая система, стабилизированная на момент начала ознакомления с ней. А вот дальше все ошибки пользователям интересны и должны быть видны, т. к. они затрагивают функциональность, с которой они работают непосредственно сейчас.

Готовимся к обучению пользователей

На фазе «Разработка» начинается активная подготовка к опытной эксплуатации. А для того, чтобы начать эксплуатировать систему пользователям (пусть пока еще только в опытом режиме), нужно уметь с ней работать, а для этого – пройти обучение.

Но для того чтобы пройти обучение, нужно иметь функциональность системы (и ее описание в виде руководств пользователей), а фаза разработки еще не прошла и не все готово. Получается парадокс: начинаем готовиться к обучению, когда система еще не готова.

С одной стороны, можно выделить из фазы «Разработка» все, что касается обучения и подготовки начальных данных, в отдельную фазу «Подготовка к опытной эксплуатации», но, с другой стороны, нет смысла делать эту фазу отдельной после фазы разработки. Ведь система большая и осваивать ее и готовить учебные материалы можно с самого начала (конечно, не по тем блокам, которые разрабатываются отдельно с нуля внутри фазы), а на подготовку начальных данных сотрудникам заказчика нужно время, и может получиться, что если фаза будет отделена от разработки, то часть команды исполнителя будет простаивать и ждать обратной связи от пользователей, которые еще не начали осваивать систему, а активная разработка уже завершилась. Из этого следует еще один плюс совмещения процессов разработки и обучения – это обратная связь от пользователей в процессе изучения системы на фазе разработки, которая влияет на разработку (юзабилити, понятность, ошибки, найденные самими пользователями).

Понятно, что обучение пользователей не нужно начинать с первого дня фазы разработки – этот процесс начнется по готовности функциональности к показу пользователям, но сама фаза завершится, когда пользователи пройдут обучение, аттестацию и будут готовы начать опытную эксплуатацию системы. А параллельно с началом обучения будут проходить доработки и настройки функциональности для следующих тем обучения.

Важно отметить, что поставляемая с ERP-системой документация – это не инструкции для конечных пользователей, нельзя просто выдать книжку или ссылку на электронную документацию ИТС и сказать: «Там все есть». Эти материалы для специалистов. Конечным пользователям нужно знать только то, что реально нужно им в работе, у них конечный набор сценариев работы в их бизнес-процессе, новые, возникающие в процессе деятельности компании ситуации и изменения бизнес-процессов требуют рассмотрения центром компетенции и закрытия их через дообучение и обновление инструкций. Поэтому инструкции пишем для конечных пользователей в процессе внедрения под бизнес-процессы компании, детализируя под их комфортное понимание и ввод в дело новых сотрудников, смену роли сотрудника и ротации между отделами. Закладываем это время в сроки и бюджет проекта.

Детализация инструкций пользователей не должна быть излишней: описывать, как нажимать кнопку ОК или пользоваться формой выбора (автор видел и такие инструкции) на каждом скриншоте, конечно, не нужно, это некое неуважение к нормальным пользователям. На крайний случай (так как пользователи в компании могут быть далеки от уверенного владения ПК) нужна инструкция по работе с кнопками и мышкой в принципе как вводная для неподготовленных пользователей ПК, а для функциональности ERP-системы – нормальная (достаточная) для рабочего места инструкция по работе в системе, наложенная на бизнес-процесс компании.

Вообще ERP-система развивается по мере обновлений самим поставщиком, поэтому нужно планировать регулярное дообучение пользователей, обновление инструкций по ходу появления и внедрения в работу новых возможностей после обновления системы на новую версию. Стоит поставить это на поток: делать какие-то рассылки по корпоративной почте, например «новые возможности корпоративной ERP-системы», по факту переноса проектных доработок на рабочую базу и появления новых возможностей функциональности системы при обновлении версии самой ERP-системы. Такая визуализация и информирование об эволюции системы в компании создают хорошее к ней отношение: пользователям приятно видеть в обновлении то, что они сами заказыв-

вали ранее, что позволит им работать в системе лучше и больше сосредоточиться на самих бизнес-задачах, а не рутинных операциях.

По ходу проведения обучения и самообучения пользователей, и тем более по результатам опытной эксплуатации, становятся понятны места оптимизации «юзабилити» (англ. usability – удобство и простота использования). Банальное удобство для рутинных действий может экономить, например, 10 секунд (вроде немного), но при 100 повторах в час это будет 1000 с (16 минут), а за рабочие 8 часов экономия будет уже 2 часа! Это и есть эффект от оптимизации. Можно увеличить поток данных в бизнес-процессе компании через того же сотрудника – автоматизация бизнес-процессов в действии. За счет такой оптимизации за 4 рабочих дня сотрудник получит дополнительные 8 часов для работы (или перестанет зашиваться с ростом нагрузки при расширении бизнеса).

Такие «бантики» и доработки, вскрываемые консультантами по ходу подготовки системы к показу, проведения обучения и обратной связи от пользователей, должны рассматриваться и включаться в доработки системы (см. раздел «Ведем список запросов на изменение» в этой главе).

Как совет консультантам: если что-то в процессе тестирования начинает напрягать и становиться рутинной и есть понимание, как это сделать оптимальнее, нужно это сделать! Ведь вас это утомило за несколько часов тестирования, а пользователям с этим работать каждый день по 8 часов. Разработчики такие места склонны игнорировать и не думать о таких «бантиках» как о глобальной проблеме – нужно их в этом убедить. Система не просто должна работать корректно, она должна работать на конкретных пользователях, то есть они должны работать в системе без лишних сложностей. И мелочь для разработчика может быть очень важной опцией для пользователя. Система ценностей тут разная.

Само обучение может проходить:

- индивидуально (неэффективно при массе пользователей с одинаковыми ролями);
- занятиями в группах (логично проводить общие темы обучения в больших группах, а ролевые особенности уже в более мелких);
- комбинация подходов – вначале групповые занятия, потом индивидуальные (у пользователей разный темп усвоения информации и ритм обучения, кому-то после групповых занятий нужен будет дополнительный индивидуальный подход).

Обучение конечных пользователей проводят:

- консультанты исполнителя;
- центр компетенции заказчика (обученный предварительно специалистами исполнителя) – позволяет тиражировать знания о системе, сохраняет экспертизу внутри компании после завершения проекта.

Использование проектора и презентаций с тезисами, схемами и основными скриншотами системы позволяет фокусировать внимание на важных вещах. Наличие возможности делать практические задания (требует компьютеров у пользователей) зависит от инфраструктуры заказчика: есть в офисе дисплейный класс для проведения обучения или это только переговорные комнаты с большим экраном, тогда обучение состоит из лекций и самостоятельной работы на своем рабочем месте. В любом случае нужно организовать поддержку самостоятельной работы пользователей: быстрые ответы на вопросы (телефон, мессенджеры, корпоративный форум), а также доступ консультантов к рабочему месту пользователя или удаленное подключение к его рабочему столу.

Процент усвоенного материала и качество освоения системы зависят от формата подачи информации. Согласно исследованиям Эдгара Дейла, проводившимся в 1960–1970-х годах и позднее его последователями, человек способен усваивать только часть изучаемой информации.

Наглядно это можно продемонстрировать через пирамиду обучения Дейла.

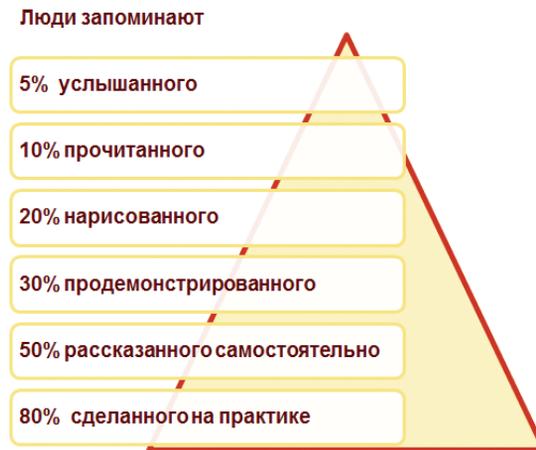


Рис. 9.4. Пирамида обучения Эдгара Дейла

Поэтому наличие скриншотов и схем в инструкциях пользователей, живая демонстрация системы в процессе обучения и, конечно, личная практика использования повышают качество обучения пользователей за счет большего объема усвоенной информации.

В принципе, обучение может проходить удаленно полностью, в формате вебинара, вопросов-ответов и подключения к рабочим столам пользователей. Но эффективность такого обучения может быть ниже очного – это происходит больше из-за психологии восприятия информации и категории/поколения пользователей. Если большинство пользователей – молодежь и привыкли к формату подачи информации в виде роликов и вебинаров, то текстовые инструкции можно заменить видеороликами. Вообще проводимые занятия полезно записывать либо готовить как видеоконтент заранее и раздавать пользователям для изучения (позволяет не отрывать сотрудников от работы массово, они изучают материалы в своем режиме). Такие записи будут хорошим подспорьем для подключения новых сотрудников (или тех, кто болел, был в отпуске), что сэкономит время и усилия консультантов на итерациях повтора обучения. Но нужно понимать, что инструкции пользователей приходится обновлять по мере изменения функциональности (меняются интерфейсы, появляются новые возможности

функциональности), замена инструкций видеороликами не снимет задачу их обновления. Поэтому запись с очного (или удаленного) обучения будет актуальна только некоторое время.

Завершением обучения пользователей должна быть аттестация на готовность работы в системе по своей роли и бизнес-процессам в системе. Консультантам исполнителя нужно подготовить аттестационные задания и вопросы. Сам факт аттестации является хорошей мотивацией для приложения сотрудником заказчика должных усилий и усидчивости в освоении системы. Аттестация не должна быть формальностью, когда все проходят ее по умолчанию. Аттестация может и должна выявлять проблемы и может проходить итерациями. Результаты доводят до руководителя проекта заказчика, а он – до руководителей подразделений, там делают соответствующие выводы и мотивационные воздействия на сотрудников, вплоть до их ротации при невозможности дальше выполнять должностные обязанности с учетом автоматизации бизнес-процессов компании.

Исполнителю рекомендуется в договоре заранее прописать конечное количество «бесплатных» передач аттестации сотрудниками заказчика, чтобы не попадать в вечный цикл: «Ну, он уже все выучил, проверьте его еще раз». Руководитель проекта со стороны заказчика и центр компетенции заказчика тогда будут больше замотивированы в поддержке процесса обучения своими силами и аттестации пользователей специалистами исполнителя с минимумом итераций.

Некоторые сотрудники могут саботировать процесс обучения и аттестации по разным причинам: в силу природного консерватизма, опасений того, что автоматизация за счет прозрачности вскроет (закроет) возможности каких-то схем и махинаций, покажет реальную эффективность сотрудников (а не декларативно-показную). Но такая прозрачность, статистика и показатели работы персонала – это цели самой автоматизации для владельцев бизнеса. И стоящих у этих целей на пути сотрудников должен убеждать сам заказчик. Руководитель проекта со стороны заказчика должен организовать устранение таких возникающих проблем, используя соответствующие административные рычаги воздействия.

При подготовке к обучению пользователей нужно озаботиться настройкой разграничения доступа:

- Настройка прав доступа к функциональности и разграничение по записям (RLS).
- Нужно проектировать, кому что видно, что можно редактировать.
- Настройка и тесты прав доступа занимают время.
- Часто про разграничение доступа вообще забывают (даже не закладывается в бюджет).

Изучение системы пользователями должно проходить в обстановке, приближенной к «боевой» (как в будущей рабочей базе), то есть под правами доступа, какие у них будут потом. Это позволяет сразу оттестировать права доступа на реальных бизнес-процессах.

Настройка прав доступа на ранних стадиях снимает ряд вопросов вида: «А что это за кнопки/поля, когда мне ими пользоваться?» – когда под полными правами пользователь видит лишнюю для себя функциональность системы.

Нужно отметить, что скриншоты для инструкций пользователей также нужно снимать в конечном виде интерфейса для данной роли пользователей, то есть под неполными правами, иначе неминуемы обращения: «А у меня форма в системе не такая, как на картинке, у меня не все кнопки есть». Консультанты исполнителя часто мыслят категориями функциональности под полными правами и мысленно игнорируют лишнее, пользователи так не умеют, для них система новая, все поля и кнопки равнозначны, должны иметь значение и использоваться в работе.

Параллельно с обучением пользователей нужно проводить процессы подготовки начальных данных. Начальные данные, перенесенные в тестовом режиме, будут полезны при проведении обучения:

пользователи будут видеть знакомые документы по знакомым номенклатурам и контрагентам. Настройки системы, даже для обучения, должны быть унаследованы от настраивавшегося на прошлых фазах прототипа системы: обучение на абстрактных демо-данных не столь эффективно, как на реальных учетных данных компании.

Когда пользователи прошли обучение и аттестованы, система успешно прошла тестирование, разработка завершена, можно переходить к следующей фазе проекта – «Опытной эксплуатации».

Опытная эксплуатация

Начать рассмотрение особенностей фазы «Опытная эксплуатация» дополнительно к описанному составу работ фазы в главе 5 уместно с метафоры, которую ввел Том ДеМарко для момента запуска системы: «Руководитель проекта – как главнокомандующий на поле битвы: к началу сражения работа главнокомандующего уже закончена».

Действительно, когда проект доходит до запуска системы, пусть еще только в опытную эксплуатацию, то все необходимые планы и управленческие воздействия руководитель проекта уже применил ранее, остается наблюдать за результатом (своей и команды проекта) – как система автоматизации начинает работать в руках пользователей.

При этом от качества проделанной ранее работы по управлению и выстраиванию процессов внутри команды зависит успех (или неуспех) прохождения этой фазы. Судорожно что-то менять, надеясь исправить ситуацию, тут уже поздно. Нужно суметь признать неготовность системы к опытной эксплуатации, если это объективно так, и взять время на необходимые доработки. Но даже при видимой готовности системы, когда процесс эксплуатации начинается, руководителю проекта рано расслабляться: пусть все спроектировано, разработано и есть регламент проведения опытной эксплуатации, но контролировать его исполнение, поддерживать команду и коммуникацию специалистов исполнителя и заказчика тоже нужно.

Опытная эксплуатация – комплексная проверка готовности автоматизированной системы. Опытная эксплуатация имеет своей целью проверку алгоритмов, отладку программ и технологического процесса обработки данных в реальных условиях.

Исходя из определения опытной эксплуатации, готовность системы на момент перевода в опытную эксплуатацию подвергается оправданному сомнению. Да – прошли тесты, да – обучен персонал, но как это все в реальности начнет работать? Поэтому по завершении разработки не осуществляется полноценный запуск в промышленную эксплуатацию, а идет проверка системы на профпригодность для ведения бизнеса, вскрываются практикой работы все недостатки теоретических допущений при проектировании, шлифуется регламент работы пользователей, уточняются инструкции пользователей, проверяется корректность переноса начальных данных.

Отказ от прошлых учетных систем в процессе опытной эксплуатации рассматривается, и чаще принимается решение о ведении параллельного учета в новой системе, т. к. для отказа от прошлых систем нужна высокая степень уверенности в новой. А при переходе учета на ведение только в новой системе это будет уже не опытная, а промышленная эксплуатация. Такие ситуации в проектах внедрения ERP-систем возможны, фаза может быть сводной: «опытно-промышленная эксплуатация». Работу системы в таком режиме уже не получится временно остановить на необходимые доработки обнаруженных недостатков. Для команды проекта режим опытно-промышленной эксплуатации сложнее, чем просто опытной: нет права на ошибку, система работает в рабочем режиме, но сама по себе еще требует доработок, которые нужно вносить «по-живому», стараясь не мешать пользователям.

В опытной эксплуатации может участвовать ограниченная выборка сотрудников заказчика, которые потом вынесут вердикт о пригодности системы к переводу в промышленную эксплуатацию и подключению всех сотрудников к работе в системе. Это снимает нагрузку со всех остальных сотрудников по двойной работе в двух системах.

На этой фазе нужно выявлять, фиксировать и оперативно рассматривать все возникающие запросы на изменения функциональности или исходных требований к автоматизации. Рассмотрим подробнее ведение списка запросов на изменение в следующем разделе этой главы.

Так как с системой начинают работать конечные пользователи, на этой фазе нужно организовывать первичную поддержку пользователей, которая дальше перерастет в полноценное сопровождение работающей системы, нужно выстраивать процессы взаимодействия пользователей и специалистов поддержки. Если в процессе обучения пользователей было выстроено взаимодействие (система регистрации ошибок, доступ к рабочему столу пользователя для консультаций), то на этой фазе такое взаимодействие продолжится уже в новом качестве. Первичную поддержку пользователей должен взять на себя центр компетенции заказчика, а его поддержку осуществляют специалисты исполнителя.

Руководителю проекта нужно следить за тем, чтобы фаза опытной эксплуатации не затягивалась неоправданно.

Сотрудники заказчика на фоне основной деятельности могут с трудом выделять время на активную работу в системе, и проверки работы системы могут идти неспешно, тогда как у проекта фиксированные границы и держать команду в режиме простоя и вялой загрузки (конечно, если система не «сыпется» и доделывается на лету) расточительно.

Для этого в регламенте проведения опытной эксплуатации должны быть сроки проведения и критерии успеха (например, отсутствие критичных замечаний, мешающих работе, на момент окончания выделенного срока проведения опытной эксплуатации), достижение которых позволит фазу закрыть, подписав соответствующий акт.

Успешное завершение опытной эксплуатации переводит систему в режим полноценной работы. Об этом речь пойдет в следующей главе.

Ведем список запросов на изменение

По ходу разработки, проведения обучения и самой опытной эксплуатации возможно возникновение запросов на изменение функциональности системы. Это не учтенные ранее требования к системе, или пересмотр реализации заявленных ранее требований, или изменение самих требований по разным причинам (меняется бизнес, стало больше понимания после знакомства с системой). Регистрировать запросы на изменение можно с помощью специализированной информационной системы поддержки проекта. Запрос на изменение должен обязательно содержать:

- описание источника проблемы (что не так, чем недовольны);
- описание влияние проблемы (почему не так, почему недовольны);
- предложение по решению (что должно быть сделано).

Жизненные циклы запроса на изменение и сообщения о дефекте/несоответствии функциональности достаточно похожи, но есть различия в следующем:

- запрос на изменение – почти всегда повод для пересмотра бюджета и сроков проекта. несоответствия, как правило, устраняются «бесплатно»;
- обычно устранение несоответствий/дефектов имеет более высокий приоритет, чем реализация запросов на изменение.

Чем сложнее система, тем сложнее оценить последствия, которые будут вызваны тем либо иным изменением. Что необходимо определить в первую очередь при таком анализе:

- Возникнут ли при реализации данного запроса на изменение противоречия с теми функциями и характеристиками системы, которые уже реализованы или утверждены к реализации?
- Потребуется ли изменения архитектуры системы, и если да, то насколько масштабные?

- Так ли уж важна реализация данного изменения?
- Какие могут быть альтернативы предложенному изменению (в том числе организационно-административного характера)?

На основе результатов анализа, предпринятого на предыдущих шагах, даются оценки запрашиваемого изменения:

- трудоемкость;
- срок реализации;
- стоимость.

Необходимо понять наличие (доступность) человеческих и других ресурсов, которые нужны для реализации запроса на изменение. Насколько вообще можно реализовать возникшее новое требование на данном этапе проекта? Оценки даются, в том числе, с учетом возможных переделок того, что уже работает, то есть бюджет и время на проект потрачены.

Например, запрос на изменения уровня «давайте это не делать, мы подумали, оно не нужно», когда что-то уже сделано, будет иметь стоимость и сроки (нужно будет удалять сделанное, менять инструкции) и экономией не станет.

На основе полученных оценок по ресурсам и срокам делаем вывод о влиянии на бюджет, сроки и требуемые ресурсы по проекту:

- все (или некоторые) параметры проекта изменятся;
- изменений параметров проекта не последует.

Как правило, если реализация запроса на изменение не приведет к изменению бюджета и срока проекта, то решение о реализации запроса на изменение принимается практически автоматически – делаем!

В случае если параметры проекта (бюджет и сроки) будут изменены, реализацию такого запроса на изменение необходимо согласовать в управляющем комитете проекта. Возможно, что после оценки изменения сроков и бюджета проекта они снизят приоритет или отменят вовсе такой запрос на изменение. Либо он перенесется в следующий этап проекта, к которому бюджет и состав работ еще не окончательно утвержден.

Запрос на изменение должен быть:

- сформулирован в терминах, понятных проектной команде, в четких, однозначно трактуемых формулировках (аналогично исходным требованиям для проекта);
- запрашиваемые изменения должны пройти итерацию проектирования и проверку системным архитектором на отсутствие противоречий текущей архитектуре системы (это не всегда возможно в процессе регистрации запроса на изменение, тогда эти работы входят в оценку сроков и бюджета реализации изменения).

После реализации запроса на изменение необходимо провести тестирование, которое должно быть направлено на установление факта достижения целей:

- изменение внесено в соответствии с требованиями инициатора запроса на изменение;
- уже существующая в системе функциональность и другие характеристики не ухудшились в результате реализации изменения.

Сама реализация запросов на изменения не отличается от разработки по требованиям к автоматизации. Важным является само ведение списка запросов на изменения, т.к. их может быть большое количество и реализация их не закладывалась в границы проекта. А все, что выходит за границы проекта, – это повод их пересматривать или отказываться от реализации, процесс должен быть управляемым.

Прямой отказ от всех запросов на изменения, позиция: «У нас все зафиксировано по договору, и ничего дополнительно мы не делаем», – вызовет отрицательные эмоции у инициаторов запросов на изменения, а это часто ключевые сотрудники заказчика – конечные пользователи, которым работать с системой. Рассмотрение же запросов на изменения и их реализация (или откладывание, или отказ) их по согласованию снимает любой негатив, а также убирает проектные риски недостатков проектирования и сбора требований на предыдущих фазах: всегда можно внести корректировки по ходу проекта, выявляя лучшие решения или оптимизируя бизнес-процессы. Система автоматизации и согласованный ранее проект автоматизации не должны стать якорем для бизнеса. Нужно быть открытым к изменениям, когда они оправданны.

Глава 10. Запуск системы и что дальше

Ввод в промышленную эксплуатацию и закрытие проекта

Основные мероприятия и документооборот на фазах «Ввод в промышленную эксплуатацию» и «Закрытие проекта» рассматривались в главе 5 «Этапы и документация проекта». Рассмотрим дополнительно некоторые особенности, возникающие на практике.

Основная сложность с проектами внедрения ERP-системы на финальных фазах – это выйти из состояния бесконечной опытной эксплуатации. В какой-то момент зачастую опытная эксплуатация переходит в опытно-промышленную, даже если нет явного закрытия актом опытной эксплуатации и перехода в следующую фазу. Заметный критерий этого процесса – это когда система становится основной, в ней идет первичный учет, нет дублирующих систем, формируется и сдается полноценная отчетность.

Акт завершения опытной эксплуатации и вообще завершения проекта подписать у заказчика руководителю проекта со стороны исполнителя обычно крайне сложно. Находятся различные причины: по закону подлости за день до подписания акта что-то отваливается, какая-то ошибка и поднятый вокруг нее эмоциональный фон сдвигают подписание акта на завтра (или через неделю), когда все будет исправлено.

Со стороны заказчика есть страх подписанием акта принять не работающую корректно систему. Практики владения системой класса ERP у заказчика еще нет, и сам факт того, что в таких системах что-то может пойти не так в процессе эксплуатации, приводит к завышенным ожиданиям по стабильности системы и работе с ней. Хотя при этом, например, в офисе, где периодически что-то требует обслуживания и мелкого ремонта, когда идет вечный бой за «чтобы все работало», перегорают лампочки, начинают протекать краны, нужно менять картриджи в принтерах, какие-то сбои на компьютерах пользователей – и все это воспринимается нормально, регулярно чинится и обслуживается в процессе эксплуатации. В случае системы автоматизации накладки могут возникать и в самих бизнес-процессах, о чем система (и требования к ней) могли не знать, и потому система оказывается не готова к таким ситуациям в процессе ее использования – нужно доработать по ходу эксплуатации саму систему или регламенты ее использования.

Четкие критерии завершения фазы опытной эксплуатации позволят перейти к полноценной промышленной работе с системой, а исполнителю важно на протяжении всего проекта готовить заказчика к мысли, что с завершением проекта внедрения сами работы по поддержке системы (и затраты на них) не завершаются, ведь у систем ERP класса ненулевая стоимость владения (TCO – Total Cost of Ownership, см. подробнее главу 1). Завершается именно проект внедрения. Системы не было до проекта, ее внедрили – это конечное действие, оно имеет срок и границы, что и соответствует определению проектных работ. Но система потом остается в компании и продолжает работать много лет, что потребует сопровождения и обслуживания. Например, как это происходит с техобслуживанием автомобилей или сложного производственного оборудования. В случае оборудования потребности в его обслуживании и ремонте, затраты и усилия на это воспринимаются нормально,

так как со школьного курса физики все представляют, что такое износ и почему оборудование требует периодического техобслуживания. А вот в отношении программного обеспечения психология восприятия дает иные ожидания – чего-то нематериального, а потому не подверженного износу. Сделанное один раз, оно должно работать вечно. Если что-то и может сломаться, то только серверная часть оборудования, на котором установлено это ПО. Но ведь внутри программы тоже идут процессы, в ней работают пользователи, человеческий фактор при работе с данными – это неиссякаемый источник проблем для учетных систем: «забыл ввести», «сдали отчетность, но нужно поправить данные», «ввел не то», «ввел не так».

Насколько подробно ни были бы проведены проектное обследование и описание бизнес-процессов, как бы хорошо ни была спроектирована и настроена система, как бы качественно ни было проведено обучение – с переходом к промышленной эксплуатации возникнет множество вопросов и проблем из серии: «Мы делаем эту операцию очень редко и поэтому не сказали о ней, но прямо сейчас ее надо учесть. Как это сделать, что нажимать в системе?»

Сами ERP-системы часто проектируются и при внедрении дорабатываются исходя из сценариев правильного с ними обращения, и обычно инструментарий по работе с проблемными данными имеют ограниченный или не имеют вовсе. Сами проблемы сложно типизировать и формализовать, т.к. ситуации могут не повторяться и возникать как новые проблемы именно этого внедрения. Для повторяемых ситуаций нужно предусматривать инструментарий исправлений или методики корректировок на существующих возможностях функционала системы. Такие ситуации, по идее, должны вскрываться опытной эксплуатацией, но ее конечная длительность по срокам может не позволить полного охвата всех проблем (например, рамки опытной эксплуатации не затронули перехода календарного года и закрытия периода, когда начинается «самое интересное», т.к. этот период подведения финансовых результатов и подготовки отчетности – отдельный источник разных ситуаций).

Сам процесс прохождения фазы промышленной эксплуатации и ее поддержки не отличается от проведения опытной эксплуатации. Скорее, ожидается спад обращений от пользователей и минимизация выявленных проблем и, как следствие, доработок и изменений инструкций пользователей. На практике спад происходит редко, скорее наоборот, т.к. подключается больше пользователей, которые до этого опытно эксплуатировали не в полную силу.

Поэтому завершение фазы ввода в промышленную эксплуатацию лучше совмещать с успешным прохождением какого-то финансового периода и его благополучным закрытием. Если это не полноценный год, то хотя бы полугодие, квартал.

Мы приходим к формулированию критерия выхода из фазы ввода в промышленную эксплуатацию – это выполнение условий:

- прошел заявленный срок фазы;
- система успешно эксплуатируется;
- в системе закрыт финансовый период, выбранный заранее как фактор приемки системы и перехода к закрытию проекта.

Собственно, для заказчика факт приемки системы и выглядит завершением всего проекта внедрения, хотя подведение итогов и достигнутых результатов (и бизнес-целей автоматизации) для заказчика тоже востребовано. Полезно показать внешнему миру, что бизнес работает в новых условиях: новая система – это, как переезд в красивый престижный офис, повод для гордости. А в некоторых случаях этот информационный повод позитивно влияет на стоимость акций компании и позволяет привлекать дополнительное финансирование.

Для исполнителя на фазе закрытия проекта, после подписания всех актов, остается важный ряд дел:

- подведение итогов («извлеченные уроки», англ. lessons learned, в терминах РМВОК) и извлечение выводов для учета в последующих проектах;

- получение отзыва от заказчика для публикации на сайте и портфолио успешных проектов;
- подготовка и согласование пресс-релизов по проекту;
- участие в профильных конференциях с докладами совместно со специалистами заказчика;
- договоренности о референс-визитах для потенциальных клиентов;
- подготовка материалов для конкурса «проект года» (<https://eawards.1c.ru>).

Опасения со стороны заказчика о том, что проект завершается и команда исполнителя уходит, снимаются выращенным за время проекта центром компетенции по ERP-системе внутри компании заказчика. Исполнитель проекта уходит, но компетенции по системе остаются в компании – это залог бесперебойной работы системы. А также снятие рисков и устранение сбоев в системе в процессе ее эксплуатации возможны за счет дополнительного договора на сопровождение корпоративной системы (с этим же исполнителем или иным подрядчиком). Но жизненный цикл именно проекта внедрения, как любой конечной работы, завершается на фазе закрытия проекта. А сама система остается работать в компании заказчика, но это уже совсем другая история...

Система продолжает работать – сопровождение корпоративной системы

Данную книгу об управлении проектами внедрения ERP-системы можно было бы завершить на предыдущем разделе – вместе с завершением проекта внедрения. Но нужно рассмотреть для полноты понимания процессы, которые выходят за его рамки, т. к. все же проект внедрения – это не самоцель для компании, цель – работающая система, автоматизация бизнес-процессов и получение различных бизнес-преимуществ от автоматизации. А для того, чтобы система продолжала успешно работать, нужно приложить определенные усилия, в том числе и по ходу самого проекта, заложив необходимые возможности для обновлений и развития системы в процессе ее сопровождения.

К задачам сопровождения корпоративной системы относятся:

- поддержка существующих пользователей, ответы на их вопросы;
- обучение и подключение новых пользователей;
- настройка и доработка прав доступа пользователей;
- дообучение пользователей при добавлении новой функциональности или ротации внутри компании на другую роль;
- доработка системы по требованиям;
- анализ инцидентов и устранение проблем;
- исправление выявленных ошибок;
- доведение до поставщика ERP-системы информации о найденных ошибках стандартной функциональности и консультирование по способам обхода или исправлениям;
- обновление системы на актуальные исправительные релизы и новые подредакции системы;
- снижение рисков, связанных с эксплуатацией сложных КИС (выполнение требований по производительности, надежности, отказоустойчивости, масштабируемости и т. п.).

Важно отметить, что сопровождение корпоративной системы помимо самой системы учета и автоматизации бизнес-процессов включает поддержку аппаратной части и прочего программного обеспечения:

- рабочие станции пользователей и серверы;
- доступ к сетевым ресурсам;
- принтеры и прочее офисное оборудование.

Так как процесс поддержки ERP-системы и всей необходимой для ее работы ИТ-инфраструктуры является важным условием для успешного функционирования корпоративной системы на многие годы в будущем, то фирма «1С» разработала технологию «1С:Технология корпоративного сопровождения» (сокращенно 1С:ТКС) для повышения квалификации специалистов на стороне заказчика или партнеров «1С», оказывающих услуги по сопровождению. По этой технологии в учебных центрах «1С» проводится обучение практикам корпоративного сопровождения.

ITIL (англ. IT Infrastructure Library) – библиотека инфраструктуры информационных технологий, описывает лучшие из применяемых на практике способы организации работы подразделений или компаний, занимающихся предоставлением услуг в области информационных технологий.

Вся деятельность в области повышения квалификации специалистов по корпоративному сопровождению опирается на стандарты и своды лучших практик:

- свод лучших практик и библиотека ITIL;
- стандарты серии ГОСТ Р ИСО/МЭК 20000 «Информационные технологии. Управление услугами»;
- стандарты жизненного цикла ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Процессы жизненного цикла программных средств» и ISO/IEC 14764:2006 Software Engineering – Software Life Cycle Processes – Maintenance;
- стандарт ГОСТ Р ИСО 9001 «Системы менеджмента качества. Требования».

1С:Технология корпоративного сопровождения (<https://consulting.1c.ru/technologies/tks/>) – это технология управления предоставлением услуг сопровождения информационных систем, разработанных на платформе «1С:Предприятие 8». Разработанная технология корпоративного сопровождения состоит из трех основных элементов:

- **описание технологии** – общие положения, описание процессной модели предоставления услуг, а также рекомендации по использованию технологии;
- **каталог услуг** – концепция и типовой каталог услуг, рекомендации по его адаптации, а также типовое соглашение об уровне услуг (SLA) и рекомендации по его формированию;
- **типовые регламенты процессов** – управление каталогом и уровнем услуг, управление обращениями, включая инциденты и запросы на обслуживание, управление проблемами, изменениями, релизами, конфигурациями, знаниями и работами.

Учитывая то, что вопросы начальной поддержки пользователей возникают уже на фазе «Разработка», когда начинается обучение пользователей и их самостоятельная работа в системе, имеет смысл вопросами последующего сопровождения внедренной системы задаться как можно раньше и начинать использовать инструментарий для сбора обращений пользователей и их поддержки, который после запуска системы можно использовать для поддержки промышленной эксплуатации.

Если предполагается, что последующую поддержку будет осуществлять команда заказчика или иной подрядчик, а не исполнитель проекта внедрения, то развернутый инструментарий первичной поддержки лучше делать на базе аппаратных и программных средств заказчика, а не исполнителя. Конечно, после завершения проекта, когда исполнитель в своем инструментарии осуществлял самостоятельно поддержку пользователей заказчика, можно будет перейти на другую систему регистрации обращений и отработки обнаруженных ошибок в инфраструктуре заказчика. Но это двойная работа: система на фазах опытной и промышленной эксплуатации уже будет работать на конкретных пользователях, они привыкнут к используемой системе поддержки и ее замена будет для них, по сути, небольшим перевнедрением – пусть не самой корпоративной системы, а связанной с ней части (инструмента поддержки). Поэтому лучше изначально продумать и настроить систему не только для процесса внедрения, но и последующего сопровождения системы.

Нужны будут четкие договоренности по предоставляемому сервису поддержки и времени реакции (так называемое SLA) на возникающие запросы пользователей. Это необходимо не только для внешних подрядчиков, но даже для сотрудников центра компетенции заказчика и пользователей.

SLA (англ. Service Level Agreement, соглашение об уровне предоставления услуги) – термин методологии ITIL, обозначающий формальный договор между заказчиком услуги и ее поставщиком, содержащий описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги.

Такие понятные правила взаимодействия понадобятся как специалистам поддержки, так и конечным пользователям, чтобы исключить ситуации хождения по офису пользователей в отдел поддержки дополнительно к созданному в системе поддержки обращению и скандалов там с целью повысить приоритет рассмотрения их обращений на фоне запросов от других пользователей. Все процессы должны быть регламентированы, время реакции должно быть утверждено и соблюдаться.

Для организации процессов сопровождения корпоративной системы напрашиваются средства автоматизации. Сбор обращений по электронной почте и телефону с неформализованными процессами их обработки не позволяет достигнуть требуемого уровня качества и управляемости. Система автоматизации процессов поддержки решает множество задач и позволяет:

- снизить затраты на информационные сервисы и поддержку ИТ-инфраструктуры;
- получить обоснование расходов на ИТ и эффективный ИТ-сервис как основу для развития бизнеса;
- обеспечить соответствие основных ИТ-процессов задачам бизнеса;

- повысить прозрачность и измеримость работы ИТ-службы, заинтересованность ИТ-подразделения в предоставлении качественного сервиса;
- повысить удовлетворенность пользователей за счет получения ими дополнительных удобств при обращении в службу техподдержки;
- установить взаимоотношения между ИТ-службой и бизнес-подразделениями компании по принципу «клиент – заказчик» для обоснования запросов на финансирование сервисов требуемого качества;
- вести учет имеющихся ресурсов;
- эффективно планировать работы, активы и бюджет на ИТ-инфраструктуру и ее сопровождение;
- осуществлять автоматизированный аудит и анализ состояния ИТ-ресурсов, определять проблемные места;
- оперативно устранять сбои и сократить время возможных простоев из-за проблем с ИТ;
- наглядно представлять результаты работы руководству компании;
- оптимизировать работу с внешними контрагентами.

Таким инструментом автоматизации может выступить решение «1С:ИТИЛ Управление информационными технологиями предприятия». Подробно с возможностями конфигурации можно ознакомиться на сайте <http://www.1c-itil.ru>.



Рис. 10.1. Возможности решения 1С:ITIL КОРП

Рассмотрим некоторые понятия и определения, вводимые ИТIL.

- **Запрос на обслуживание** (Service Request) – запрос от пользователя на предоставление информации на стандартное изменение или на доступ к ИТ-услуге.
- **Инцидент** (Incident) – незапланированное прерывание предоставления ИТ-услуги или снижение качества ИТ-услуги.
- **Изменение** (Change) – добавление, модификация или удаление чего-нибудь, что имеет влияние на ИТ-услуги.
- **Проблема** (Problem) – неизвестная причина одного или более инцидентов.
- **Событие** (Event) – изменение состояния, которое имеет значение для управления конфигурационной единицей или ИТ-услугой.

- **Сбой** (Failure) – потеря способности функционировать в соответствии со спецификацией или предоставлять требуемый результат.
- **Ошибка** (Error) – изъян в проекте или неверное функционирование, вызывающее сбой одной или более конфигурационной единицы или ИТ-услуги. Ошибка, совершенная сотрудником, или нарушение процесса, которое влияет на конфигурационную единицу или ИТ-услугу, также являются ошибкой.
- **Известная ошибка** (Known Error) – проблема, для которой определены и документированы корневая причина и обходное решение.
- **Релиз** (Release) – совокупность аппаратных средств, программного обеспечения, документации, процессов или других компонентов, требующихся для внедрения одного или нескольких согласованных изменений в ИТ-услугах.
- **Центр обслуживания пользователей** (Service Desk) – система, предназначенная для обеспечения поддержки пользователей, быстрого и удобного поиска необходимых документов, запросов на обслуживание, инцидентов, проблем, запросов на изменение, релизов, задач.

Разница между следованием рекомендациям, лучшим практикам и самостоятельным изобретением методик для корпоративной поддержки состоит в стоимости полученного опыта (эксперименты, ошибки, потраченное время и деньги). О многих процессах сопровождения и решениях в них не задумываешься, пока не возникает необходимость (не появляются инциденты, требования, проблемы, ошибки). Использование формализованных технологий и инструментов для них делает ненужным «изобретение велосипеда» при выстраивании процессов сопровождения корпоративных систем конкретной компании.

При организации поддержки корпоративной системы нужно решить, какой временной формат будет использоваться: 24/7 или 8/5, то есть работы службы круглосуточно или только в рабочие часы.

Потребность режима 24/7 может быть вызвана наличием филиалов в различных часовых поясах или форматом работы предприятия, если там организована посменная работа без выходных. От нагрузки на службу поддержки зависит состав ее участников (посменная работа или специалисты из разных часовых поясов). Возможно, компании подойдет промежуточный вариант 12/5, когда у специалистов поддержки будет временной сдвиг прихода-ухода в офис, чтобы перекрыть 12-часовой диапазон в рабочие дни.

Когда система уже запущена в промышленную эксплуатацию, грань между доработками в рамках сопровождения системы и полноценными проектами, как отдельными этапами проекта с полным жизненным циклом из всех фаз, размывается. Для работающих систем без четкого плана развития, который может быть оформлен в отдельный проект, хорошо работают гибкие методики ведения проекта с небольшими периодами поставки новой функциональности (например, раз в 2 недели). Система развивается в этом случае по запросу, исходя из реальных потребностей. Доработки системы могут осуществляться как силами специалистов поддержки, так и с привлечением на усиление проектных команд других подрядчиков.

В случае подключения новых филиалов возможно тиражирование системы автоматизации с поставленным на поток процессом обучения новых пользователей, инициализацией системы необходимыми начальными данными для подключаемой бизнес-единицы. Если бизнес-процессы у филиала типовые, то такое тиражирование не будет полноценным проектом внедрения. Но это еще и не поддержка промышленной эксплуатации. Хотя система и работает во всей компании, но в новой бизнес-единице предприятия она требует внедрения, пусть и с укороченным списком работ. С задачами тиражирования решения обычно справляется центр компетенции компании, а вот если бизнес-процессы подключаемой бизнес-единицы отличаются от автоматизированных ранее, то необходим полноценный проект – со сбором и анализом требований, по полному жизненному циклу проекта, с переводом в сопровождение после запуска филиала в промышленную эксплуатацию.

Доработки системы в рамках такого проекта должны не ухудшать (не разваливать) работающую в других подразделениях функциональность системы. Такой проект, как и любые доработки по развитию уже запущенной системы, требует повышенного внимания, т.к. несет риски привнесения ошибок в используемую в системе функциональность,

что может приводить к простоям в бизнес-процессах компании и, как следствие, к финансовым потерям. Ведь опытная эксплуатация для части сотрудников (запускаемого филиала) – это продолжение промышленной эксплуатации для всех остальных подразделений. И если система единая и в ней сквозной учет по группе компаний, то доработки функциональности будут вестись «по живому». Поддержка пользователей и оперативное устранение проблем в таком случае особенно актуальны.

Когда система работает, стабильна, зачем ее обновлять? С другой стороны, в компании могут быть автоматизированы процессы, учет которых регламентируется законодательством или по которым нужно готовить отчетность либо обмениваться данными с фискальными органами. Такую поддержку законодательства и изменение системы осуществляет сам поставщик ERP-системы. И эти изменения нужно применить в компании, где внедрена ERP-система, то есть без обновлений не обойтись. При всей стабильности системы для управленческого учета и нежелании прикладывать усилия для ее обновления придется планировать (и закладывать в бюджет сопровождения) регулярные обновления. Чем более модифицирована была система, тем сложнее может протекать обновление. Необходимо проводить сравнение и объединение новой версии от поставщика с используемой конфигурацией, проверять работоспособность доработок, сделанных в расширениях конфигурации. Делать это нужно, очевидно, на копии базы. Проверить полученный результат, провести замеры времени, потребного для обновления данных всеми обработчиками обновления. Провести регрессионное тестирование (вот тут автотесты пригодятся особенно) по основным бизнес-процессам. Доработать по необходимости выявленные проблемы и уточнить инструкции пользователей. И уже после вердикта о стабильности обновленной системы обновлять рабочую базу.

В случае если компания работает не в режиме 24/7, можно изыскать так называемый «технический час» – время простоя системы, когда с ней можно совершать операции по обслуживанию, что не будет мешать работе пользователей. Например, в выходные дни или ночью. Но если база данных большого объема или компания работает 24/7 и не имеет существенного запаса в часах простоя от работы пользователей, то обработчики обновления данных для перехода на новую версию системы могут не успеть отработать за выделяемое время.

При обновлении 1С:ERP обработчики обновления делятся:

- на **монопольные** – выполняются последовательно, блокируют работу пользователей;
- **фоновые** – выполняются в несколько потоков (параллельно), работу пользователей не блокируют, но могут ограничивать доступ к необработанным документам или искажать данные в отчетах.

Обработка данных на многоядерных серверах в несколько потоков (например, 16 ядер сервера дают 16 потоков обработки) выполняется быстрее, практически с кратным числом потоков ускорением обработки, и в итоге фоновые обработчики выполняются за время самого медленного из них.

При этом некоторые обработчики обновления данных сами могут работать в многопоточном режиме, что еще больше нагружает аппаратную часть в процессе обновления, но существенно сокращает время обновления.

Если совсем не удастся найти время для останова работы пользователей и обновления данных, можно использовать следующую последовательность действий:

- обновление конфигурации (после всех необходимых тестов) и обновление данных проводятся на копии рабочей базы;
- останов рабочей базы нет, идет продолжение работы пользователей после снятия ее копии для обновления;
- после завершения процедур обновления на копии происходит переключение пользователей на новую систему (останов работы занимает минуты на замену путей запуска системы с другого сервера);
- далее выполняется фоновый перенос введенных за время перехода документов (с момента, когда забрали копию).

Такой подход требует мощного серверного оборудования для тестовой базы, на которой будет проходить обновление перед заменой рабочей базы, также будут замедления в работе пользователей, пока будут идти копирование и замена баз. Запущенные после замены баз процедуры автоматического переноса данных, вводимых после взятия копии, из старой версии в новую позволяют проводить обновление без длительного останова работы пользователей на технические часы простоя. Синхронизация введенных данных может потребовать некоторого времени, но это может проходить параллельно с работой пользователей.

Четко выстроенные в компании процессы сопровождения системы и пользователей, регулярные обновления и подключение новых сотрудников и бизнес-процессов позволяют ERP-системе стать частью компании, инструментом автоматизации и оптимизации протекающих в ней бизнес-процессов. Все это позволяет компании функционировать более эффективно, сотрудникам – сосредоточиться на бизнес-задачах увеличения (или удержания при большой конкуренции) доли рынка и максимизации прибыли, а не на самих процессах автоматизации или поддержки внедренной системы.

Заключение

Вот и подошло к концу рассмотрение вопросов управления проектами внедрения ERP-системы. В книге были рассмотрены необходимые (но не достаточные) понятия, методики и некоторые практические аспекты, дающие читателю начальное представление об управлении проектами и процессах, происходящих при внедрениях. Автор надеется, что такое относительно краткое структурированное описание будет полезно на пути становления новых специалистов.

Реальный практический опыт управления проектами внедрения можно получить, конечно, только на самих проектах, но быть достаточно подготовленными к ним в теории тоже полезно. Для этого теории и чужой практический опыт нужно постоянно изучать, руководителю проектов всегда нужно поддерживать «хорошую форму» своих компетенций. Дальнейшее углубленное изучение темы управления проектами можно продолжить с помощью приведенных в списке литературы книг.

Для более полного понимания процессов, протекающих на предприятиях, и возможностей по их автоматизации рекомендуется дополнительно ознакомиться с книгами серии «Академия ERP», а также с документацией к системе 1С:ERP и тематическими статьями профильных сайтов в Интернете.

Отзывы и предложения по улучшению этой книги и всей серии можно присылать на электронную почту publishing@1c.ru.

Список использованной литературы

- PMBOK, 6th Edition // <https://www.pmi.org>.
- Журнал «Управление проектами» // <https://pmmagazine.ru>.
- Журнал «Управляем предприятием» // <http://upr.ru>.
- Эдвард Йордон. Путь камикадзе. Полное руководство для разработчика программного обеспечения по выживанию в безнадежных проектах // Лори, 2003.
- Чапел Хилл, Брукс Фредерик. Мифический человеко-месяц, или Как создаются программные системы // Символ-Плюс, 2007.
- Карл Вигерс, Джой Битти. Разработка требований к программному обеспечению // ВНУ, 2013.
- Том ДеМарко. Deadline. Роман об управлении проектами // М.:Манн, Иванов и Фербер, 2016.
- Том ДеМарко, Тимоти Листер. Вальсируя с медведями: управление рисками в проектах по разработке программного обеспечения // Компания p.m. Office, 2005.

- Алан Купер. Психбольница в руках пациентов // Символ-Плюс, 2009.
- Эванс Эрик. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем // Вильямс, 2010.
- Элияху М. Голдратт, Элия Шрагенхайм, Птак А. Керол. Цель-3. Необходимо, но не достаточно // Баланс Бизнес Букс, 2009.
- Максим Батырев. 45 татуировок менеджера // М.: Манн, Иванов и Фербер, 2019.

© ООО «1С-Публишинг», 2021

© Оформление. ООО «1С-Публишинг», 2021

Все права защищены.

Материалы предназначены для личного индивидуального использования приобретателем. Запрещено тиражирование, распространение материалов, предоставление доступа по сети к материалам без письменного разрешения правообладателей.

Фирма «1С»

123056, Москва, а/я 64, Селезневская ул., 21.

Тел.: (495) 737-92-57, факс: (495) 681-44-07.

1c@1c.ru, <http://www.1c.ru/>

Издательство ООО «1С-Публишинг»

127434, Москва, Дмитровское ш., 9.

Тел.: (495) 681-02-21, факс: (495) 681-44-07.

publishing@1c.ru, <http://books.1c.ru>

О найденных опечатках просьба сообщать по адресу publishing@1c.ru.