

В.В. Братищенко

**ПРОЕКТИРОВАНИЕ
ЭКОНОМИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Учебное пособие

Министерство науки и высшего образования Российской Федерации
Байкальский государственный университет

В.В. Братищенко

**ПРОЕКТИРОВАНИЕ
ЭКОНОМИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Учебное пособие

Текстовое электронное издание

Иркутск
Издательский дом БГУ
2022

© Братищенко В.В., 2022
© ФГБОУ ВО «БГУ», 2022

УДК 004.82 (075.8)
ББК 32.97я7

Издается по решению редакционно-издательского совета
Байкальского государственного университета

Рецензенты

д-р физ.-мат. наук А.В. Боровский (БГУ)
канд. техн. наук, доц. Т.Ю. Новгородцева (ИГУ)

Братищенко, В.В. Проектирование экономических информационных систем : учеб. пособие / В.В. Братищенко. – Иркутск : Изд. дом БГУ, 2022. – 132 с. – URL: <http://lib-catalog.bgu.ru>. – Текст : электрон.

Содержит сведения о компонентах информационных систем, стадиях и этапах проектирования информационной системы; описание последовательности и методологии выполнения работ на разных этапах проектирования; методы организации работ по проектированию.

Предназначено для студентов направлений подготовки 09.03.03 Прикладная информатика, 38.03.05 Бизнес-информатика.

Учебное электронное издание

Минимальные системные требования:

веб-браузеры: Microsoft Edge версии 79, Google Chrome версии 51,
Mozilla Firefox версии 52, Safari версии 11 (или более новые);
средства просмотра файлов Portable Documents Format: Adobe Acrobat версии 7.0,
Adobe Reader версии 7.0, Sumatra PDF версии 1.1 (или более новые),
Foxit Reader всех версий, PDF24 Creator всех версий.

Доступ к сети Интернет.

Минимальные требования к конфигурации и операционной системе компьютера определяются требованиями перечисленных выше программных продуктов.

Издается в авторской редакции

Подписано к использованию 14.11.2022.

Объем 2,3 Мб.

Байкальский государственный университет.
664003, г. Иркутск, ул. Ленина, 11.
<http://bgu.ru>.

© Братищенко В.В., 2022
© ФГБОУ ВО «БГУ», 2022

ОГЛАВЛЕНИЕ

Предисловие	5
1. Общие сведения	8
1.1. Определение и компоненты информационной системы	8
1.2. Экономические информационные системы	12
1.3. Подсистемы предприятия.....	15
1.3.1. Обеспечение стратегического управления.....	16
1.3.2. Управление производством	22
1.3.3. Маркетинг и сбыт.....	24
1.3.4. Материально-техническое снабжение	25
1.3.5. Управление персоналом и оплата труда.....	26
1.3.6. Управление финансами (планово-финансовый отдел)	27
1.3.7. Бухгалтерский учет	28
2. Методология разработки информационных систем	31
2.1. Жизненный цикл информационной системы.....	31
2.2. Принципы разработки информационных систем	37
2.3. Общие методы разработки	43
3. Анализ бизнес-процесса. Разработка требований к информационной системе.....	47
3.1. Изучение и анализ предприятия в целом.....	48
3.2. Описание бизнес-процесса.....	49
3.2.1. Определение и компоненты бизнес-процесса.....	49
3.2.2. Модель IDEF0.....	52
3.2.3. Управление бизнес-процессом	54
3.2.4. Информационная составляющая бизнес-процесса.....	60
3.3. Инновации управления на основе информационных технологий	63
3.4. Описание и анализ существующей информационной инфраструктуры	66
3.5. Определение требований и приоритетов. Разработка технического задания.....	68
4. Конструирование.....	74
4.1. Работы стадии конструирования	74
4.2. CASE-технология проектирования информационных систем	76
4.3. Язык объектно-ориентированного моделирования UML	80
4.3.1. Диаграмма прецедентов (use case diagram)	81
4.3.2. Диаграмма классов (Class diagram)	83
4.3.3. Диаграмма последовательностей (Sequence diagram) и диаграмма кооперации (Cooperation diagram).....	85
4.3.4. Диаграмма деятельности (Activity diagram)	87
4.3.5. Диаграмма состояний (State diagram)	89
4.3.6. Диаграммы пакетов (Package diagram)	90
4.3.7. Диаграммы компонентов (Component diagram).....	91
4.3.8. Диаграммы размещения (Deployment diagram)	91

4.4. Проектирование интерфейса пользователя, структуры и логики работы программы	93
5. Реализация и сопровождение информационной системы	96
5.1. Разработка и внедрение	96
5.2. Обеспечение эксплуатации	97
6. Типовые проектные решения	103
6.1. Быстрая разработка программ	103
6.2. Параметрически-ориентированное проектирование.....	104
6.3. Модельно-ориентированное проектирование.....	105
7. Организация процесса разработки информационной системы	108
7.1. Управление основными характеристиками проекта	108
7.2. Организация коллективной работы над проектом	109
7.3. Управление процессом проектирования	111
7.4. Экономические показатели проектирования	113
7.5. Гибкие методы проектирования информационных систем.....	116
Заключение	124
Приложение. Требования стандартов к содержанию проектных документов	125
Список использованной и рекомендуемой литературы.....	132

ПРЕДИСЛОВИЕ

В данном пособии рассматривается проектирование информационных систем для управления бизнес-процессами. В современных условиях информационная система (ИС) становится необходимым инструментом управления. ИС обеспечивает использование компьютерных функций пользователям, выполняющим функции учета и управления. Каждый пользователь получает в ИС функции по вводу, обработке и получению данных, соответствующие его профессиональным обязанностям. Пользователи образуют большой коллектив, решающий задачи управления некоторым бизнес-процессом. Значительная часть функций ИС обеспечивает учет и накопление сведений о выполнении операций бизнес-процесса. По накопленным данным определяются многочисленные показатели, измеряющие разные свойства бизнес-процесса. Вот только некоторые из них:

- денежный результат бизнес-процесса – доход, добавленная стоимость;
- результат в натуральных единицах: штуках, единицах, килограммах, метрах – в зависимости от результатов бизнес-процесса;
- расход материалов (в натуральных и денежных единицах);
- потребление ресурсов (в натуральных и денежных единицах), особенной сложностью отличается нормирование и учет трудовых затрат;
- временные характеристики бизнес-процесса;
- показатели качества;
- показатели удовлетворенности клиентов.

Разные бизнес-процессы измеряются разными показателями. Некоторые из них характеризуют многие бизнес-процессы. Такими являются финансовые показатели: доход, расход, прибыль, индекс доходности и многие другие. Другие показатели отражают специфику предприятия.

Измерение характеристик бизнес-процесса является основой управления. Планируется бизнес-процесс в виде набора значений показателей. Выполнение плана измеряется фактическими значениями этих показателей. Сопоставление плана и факта является основой для принятия управленческих решений. Решения принимаются разными менеджерами на разных уровнях управления, и каждый участник этого процесса должен получать из информационной системы необходимые сведения для принятия решений. Большое количество данных, пользователей, увязывание действий всех пользователей в единую технологическую цепочку накопления и использования данных приводит к высокой сложности информационных систем и процессов их проектирования.

Все затраты на разработку, приобретение и внедрение ИС должны окупаться повышением качества учета (меньше затрат, меньше ошибок, удобнее ввод, электронный обмен документами с партнерами и клиентами) и управления (выше оперативность, достовернее данные, ниже риски, выше качество решений). Важную роль в повышении качества играют инновации на основе компьютерных технологий: интернет-коммуникации, электронный документооборот, интеллектуальная аналитика, автоматизация расчетов, оптимизация планов, современная бизнес-графика, выделение и использование феноменов и зависимостей, прогнозирование.

Кроме организационных вопросов: кто и как будет использовать функции ИС необходимо решать технические: какие устройства будут применять пользователи (персональные компьютеры, планшеты, смартфоны, сканеры, принтеры, регистраторы...), как они будут соединяться вместе (интернет, локальная вычислительная сеть, облачные технологии, мобильная связь...), какие серверы будут обеспечивать хранение и обработку данных, как будет выполняться передача данных.

Конечно, одним из главных вопросов функционирования ИС является использование программ. Какие программы будут обеспечивать интерфейс пользователя, передачу, хранение и обработку данных, как программы будут взаимодействовать друг с другом? Все эти сложные задачи не имеют однозначного решения, отличаются по затратам и эксплуатационным свойствам. Все больше программ применяются готовыми или собираются из готовых компонентов. Все больше программ объединяются в единую систему с единым хранением данных и общим администрированием. Доставка и развертывание программ все чаще производится через глобальные или локальные сети. В условиях повсеместного использования компьютеров актуальным становится организация взаимодействия информационных подсистем и систем. Например, сбыт должен учитывать производство товаров, а график производства товаров создает план потребления материалов и ресурсов, который следует синхронизировать с внешними поставщиками и вспомогательными производствами. Информационный обмен между системами и подсистемами существенно снижает риски, уменьшает негативное влияние человеческого фактора, сокращает затраты на документирование и согласование, но в то же время требует проработки регламентов обмена данными.

Данные образуют информационное ядро современных ИС. Базы данных предоставляют удобный и надежный инструмент для хранения больших объемов сведений. Жестко структурированные реляционные базы данных обеспечивают высокую эффективность хранения и обработки, но требуют изменений программ в случае изменения структуры. Гибкие структуры существенно усложняют обработку и снижают ее эффективность, но позволяют эффективно адаптировать ИС к изменениям в управлении бизнес-процессами. Проектирование структур хранения данных – поиск компромисса между эффективностью и гибкостью – является одной из важнейших задач проектирования. Другой задачей является выбор способа хранения данных: единое хранилище, или распределение данных по нескольким серверам сети, или использование облачных ресурсов. Часто происходит разделение данных на условно-постоянные для – хранения нормативно-справочной информации, оперативные – для хранения детальных учетных данных и аналитические – для эффективного выполнения аналитической обработки.

Проектирование информационной системы должно учитывать все особенности работы всех пользователей ИС. Функции отдельного пользователя должны быть реализованы в программе на подходящем устройстве и должны использовать определенные структуры данных, хранимых в системе. Иногда изменение в какой-то одной части ИС вызывает целую цепочку изменений в других частях, а на первый взгляд простые пожелания пользователей приводят к суще-

ственной переделке всех компонентов ИС. Поэтому и сам процесс проектирования ИС является в высокой степени неопределенным процессом с высокими рисками по соответствию функций ИС потребностям пользователей, по соблюдению временных и стоимостных ограничений разработки. Управление процессом разработки отражает все эти неопределенности и сложности.

Проектирование информационных систем включает много направлений и технологий. Известные традиционные подходы [1–4] рассматривают процесс проектирования по компонентам и по этапам. Корпорация Microsoft предлагает свою технологию проектирования [5]. Объектно-ориентированный подход предоставляет свою технологию процессов проектирования [6]. Этот подход представлен многочисленными графическими моделями [7–9], описывающими различные компоненты информационных систем. Большую популярность приобретают гибкие методы проектирования [10]. В данном пособии в разной степени подробности отражены все перечисленные подходы. Разные подходы к проектированию ИС не противоречат друг другу, а акцентируют внимание на различных сторонах процесса проектирования. Выбор подходов, стандартов и моделей каждый проектировщик делает сам, исходя из специфики задачи проектирования.

В первой главе приводятся сведения о компонентах информационных систем, описывается специфика экономических информационных систем, деление информационной системы предприятия на подсистемы.

Во второй главе рассматриваются стадии и этапы разработки информационной системы, образующие жизненный цикл. Приводятся несколько распространенных вариантов жизненного цикла. Обсуждаются принципы проектирования и общие подходы и методы решения задач проектирования.

Третья глава посвящена постановке задач проектирования. Рассматриваются методы описания бизнес-процесса и соответствующей системы управления. Приводятся возможные инновации управления на основе информационных технологий. Представлен шаблон постановки задачи проектирования ИС в виде технического задания.

В четвертой главе рассматриваются методы проектирования отдельных компонент информационной системы на основе языка UML. Приведены рекомендации по проектированию интерфейса пользователя.

В пятой главе описаны работы и особенности оформления документов на этапах реализации и сопровождения информационной системы.

В шестой главе рассматриваются методы применения готовых решений на уровне отдельных компонентов информационной системы или применения готовых продуктов с параметрической или модельной настройкой.

В седьмой главе приводятся методы календарного планирования и управления процессом разработки ИС и рассматриваются вопросы управления коллективом разработчиков.

Самая главная вещь при разработке программ – ясно представлять конечную цель.

B. Stroustrup

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Определение и компоненты информационной системы

После первых успешных применений вычислительной техники для решения задач учета и управления на предприятиях появился соответствующий класс экономических информационных систем, который значительно отличается от первых применений компьютеров для решения вычислительных задач. Для обозначения такого применения в Советском Союзе был введен специальный термин – автоматизированная система управления [11]. Существует руководящий документ РД 50-680-88, который определяет понятие автоматизированной системы (АС): *АС – это организационно-техническая система, обеспечивающая выработку решений на основе автоматизации информационных процессов в различных сферах деятельности (управление, проектирование производства и т.д.) или сочетаниях.* Более поздний стандарт ГОСТ 34.003-90 дает похожее определение. *Автоматизированная система – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.* Применение термина «автоматизированная» подчеркивает участие человека в функционировании системы, а именно то, что машина не заменяет пользователя, но обеспечивает инструменты для решения задач управления. Системы, предназначенные для управления, так и стали называться автоматизированные системы управления – АСУ. Этот класс систем отделили от систем управления технологическими процессами – АСУТП. В дальнейшем термин АСУ стал активно заменяться термином «информационная система» без принципиальной замены классификационных признаков.

Следующее определение выделяет характерные составляющие информационной системы и основные операции обработки информации, без которых система не может называться информационной. *Информационная система (ИС) – это организационные, технические, программные и информационные средства, объединенные в единую систему с целью сбора, хранения, обработки и выдачи информации, предназначенной для решения определенного множества задач.*

Характерно, что все определения ставят человеческий, организационный фактор на первое место. Действительно, мало разработать программы, системы хранения данных, мало разместить все это в вычислительной среде, необходимо, чтобы пользователи работали с системой: вводили данные, получали результаты запросов и вычислений для более эффективного и продуктивного выполнения профессиональных обязанностей. Для этого информационная система на практике, во-первых, должна освобождать пользователей от выполнения рутинных, ручных операций, а во-вторых, обеспечивать нужными вычислениями процессы обоснования и принятия решений.

В информационных системах выделяют процессы, образующие технологическую цепочку использования данных: сбор, хранение, обработка, выдача. Сбор данных в определении информационной системы подразумевает превращение данных в электронную форму, пригодную для дальнейшей обработки в вычислительных системах. В информационных системах управления можно выделить ввод данных пользователями вручную на рабочих местах, подключенных к вычислительной системе. Ввод данных вручную с помощью соответствующих программ является основным способом ввода данных в систему и, одновременно, основным источником ошибок в информационной системе. Для предотвращения ошибок используют различные приемы: выбор из выпадающего списка, переключатели, радиокнопки, календари, выбор данных из справочников. Для обнаружения ошибок применяют синтаксические и семантические проверки. Известны и применяются методы ввода проверочных данных, таких как контрольные суммы, в редких случаях (например, задание паролей) для исключения ошибок используют двойной ввод. Общей тенденцией является снижение объема ручных операций ввода данных за счет применения экспорта-импорта и автоматической регистрации (штрих-коды, магнитные карты, smart-карты...).

Накопление и хранение данных всегда было неотъемлемой частью информационных технологий. Современные системы должны обеспечивать одновременный доступ к данным многим пользователям с возможностью разделения прав доступа к данным. Кроме этого, часто требуется распределенное хранение данных на нескольких узлах вычислительной сети. Структуры хранения данных должны обеспечивать эффективную обработку данных. Перечисленным выше требованиям удовлетворяют базы данных (БД) [12], доступ к которым обеспечивается соответствующей системой управления базами данных (СУБД).

В современных СУБД выделяют функции оперативного доступа к данным для извлечения и изменения первичных данных (OLTP – OnLine Transaction Processing) и функции аналитической обработки (OLAP – OnLine Analitical Processing). Технология OLTP обеспечивает ввод и хранение всех детальных данных об операциях, выполняемых в рамках бизнес-процесса. Очевидно, что объем этих данных непрерывно растет. Удаление данных практически никогда не происходит по следующим причинам: с одной стороны, есть нормативные сроки хранения информации, с другой – данные нужны для решения многочисленных аналитических задач. Накопление данных, как правило, не приводит к потере производительности благодаря эффективным методам индексации и росту производительности вычислительной техники и объемов накопителей.

Технология OLAP [13] основана на сборе данных из разных систем, баз данных и вообще из любых источников, доступных в том числе в сети Интернет. Из Интернета можно получать регистрационные данные предприятий и персонала, курсы валют, котировки ценных бумаг, производственные календари и много другой нормативной и справочной информации. Технология OLAP обеспечивает автоматический сбор, агрегирование и объединение данных из разных источников. Собранные данные могут сохраняться в специализированных хранилищах данных в структурах (кубах), ориентированных на эффективную обработку данных.

Обработка заключается в многомерном анализе данных – изучении изменений показателей (например, дохода, объемов поставок) вдоль одного измерения или вдоль комбинации измерений (например, по дням и видам продукции). Доступ к результатам многомерного анализа реализуется многими готовыми инструментами, такими как сводные таблицы в MS Excel или Power BI desktop. Технология OLAP успешно заменяет многочисленные бумажные отчеты, предоставляя дополнительную функциональность в виде выбора показателей и измерений и построения соответствующих графиков.

Обработка в современных системах носит распределенный характер. В них выделяют три уровня логики (функциональности): презентационная логика, логика обработки данных и логика доступа к данным. *Логика представления данных* реализуется клиентскими персональными компьютерами или мобильными устройствами и предназначена для ввода пользователем команд и данных и демонстрации запрошенных данных и результатов обработки. *Логика обработки данных* обычно сосредоточена на сервере приложений и содержит некоторые общие для всей системы правила и алгоритмы обработки. Часть общезначимой логики обработки данных также может быть передана серверу БД или OLAP-серверу. *Логика доступа к данным* реализуется сервером БД или OLAP-сервером и предназначена для выполнения запросов пользователей в рамках их полномочий.

Кроме многомерной обработки активно применяются методы Data Mining [13], которые способны выделять из исходных, «сырых» данных зависимости, шаблоны, прогнозы и другие закономерности. Использование этих закономерностей позволяет решать самые разные задачи: определять риски невозврата кредитов, выделять подозрительные операции, прогнозировать покупки клиентов, автоматически классифицировать обращения для направления нужному специалисту и многие другие.

В управлении используются различные виды обработки данных, учитывающие условия производства [8]. Это может быть оптимизация планирования, построение календарных графиков производства и загрузки оборудования, вычисление графиков потребления материалов и ресурсов, расчеты смет, бюджетирование, управление запасами и многие другие задачи.

Отображение данных в наиболее удобной для пользователя форме выделяется в самостоятельную операцию, хотя может рассматриваться как часть обработки данных. Общей тенденцией в построении интерфейса пользователя является увеличение доли графических средств отображения. Средства визуализации становятся все более интерактивными: изображения можно вращать, находить наиболее выразительные проекции. Выводится подробная информация по элементу графика при указании его курсором. Активно используются интерактивные карты и средства мультимедиа.

Определение информационной системы выделяет ее составные части, без которых классифицируемый объект перестает быть информационной системой. Они включают следующие компоненты:

- Организационные средства, к которым относят пользователей, использующих ИС для решения профессиональных задач, и обслуживающий персонал,

обеспечивающий функционирование ИС. Кроме этого к организационным средствам относят все инструкции и документы, регламентирующие использование ИС и взаимодействие персонала.

– Информационные средства – данные, хранимые и циркулирующие в системе, и документы, описывающие структуры данных.

– Технические средства хранения, передачи и обработки данных обеспечивают хранение, передачу, обработку и вывод данных.

– Программные средства, выполняющие алгоритмы обработки, передачи и доступа к данным.

С каждым компонентом системы можно связать несколько видов обеспечения, которые представлены в табл. 1.

Таблица 1

Компоненты информационной системы

Компоненты ИС	Компоненты автоматизированной системы (АС) в соответствии с ГОСТ 34.003-90
Пользователи	<ul style="list-style-type: none"> – Пользователь АС – лицо, участвующее в функционировании АС или использующее результаты функционирования АС. – Эксплуатационный персонал – персонал, обеспечивающий эксплуатацию вычислительной техники и программ
Инструкции пользователям	<ul style="list-style-type: none"> – Организационное обеспечение АС – совокупность документов, устанавливающих организационную структуру, права и обязанности пользователей и эксплуатационного персонала АС. – Правовое обеспечение АС – совокупность правовых норм, регламентирующих правовые отношения, возникающие в процессе эксплуатации АС, и юридический статус результатов функционирования АС. – Эргономическое обеспечение АС – совокупность решений по согласованию психологических, антропометрических и физиологических характеристик и возможностей пользователей АС с характеристиками интерфейса пользователя. – Лингвистическое обеспечение АС – совокупность правил формализации естественного языка, используемых в интерфейсе пользователя АС
Данные	<ul style="list-style-type: none"> – Информационное обеспечение – формы существования, объемы и размещение информации в АС
Технические средства хранения, передачи и обработки данных	<ul style="list-style-type: none"> – Техническое обеспечение АС – совокупность всех технических средств, используемых при функционировании АС
Программы	<ul style="list-style-type: none"> – Математическое обеспечение – математические методы, модели и алгоритмы, применяемые в АС. – Программное обеспечение АС – программы и программные документы для отладки, проверки работоспособности и функционирования АС

Для проектирования наиболее значимыми и трудоемкими являются следующие виды обеспечений:

- Организационное обеспечение в виде инструкций, руководств, описаний технологии работы в условиях эксплуатации информационной системы.
- Программное обеспечение, предоставляющее основные функции по вводу, хранению и обработке данных.
- Информационное обеспечение для надежного и эффективного хранения данных и предоставления доступа к данным.

Техническое обеспечение разрабатывается достаточно редко. Это связано с повсеместным распространением информационных технологий. Как правило, внедрение новых компьютерных технологий происходит в условиях оснащения всех рабочих мест персональными компьютерами, объединенными в единую сеть, уже включающую необходимые для проектируемой подсистемы (системы) серверы. Обработка экономической информации обычно не требует увеличения имеющихся вычислительных и информационных мощностей. При необходимости изменения в технической, программной и информационной инфраструктуре, выполняются заказчиком.

Классификация информационных систем позволяет точнее формулировать требования к проектируемым видам обеспечений. Существует много подходов к классификации информационных систем. Руководящий документ РД 50-680-88 предлагает следующую классификацию автоматизированных систем в зависимости от сферы автоматизируемой деятельности:

- автоматизированные системы управления (АСУ), которые в свою очередь делят на АСУ предприятием (АСУП) и АСУ технологическими процессами (АСУТП);
- системы автоматизации проектирования;
- автоматизированные системы научных исследований;
- автоматизированные системы обработки и передачи информации;
- автоматизированные системы технологической подготовки производства;
- автоматизированные системы контроля и испытаний.

Приведенная классификация частично устарела и не отражает многих распространенных сфер применения. В частности, отсутствуют системы, связанные с глобальными сетями. Невозможно привести достаточно полную систему, так как количество областей применения компьютерных технологий увеличилось в сотни раз по сравнению с 1988 годом издания цитируемого стандарта, и этот процесс продолжается. В данном пособии рассматривается применение информационных систем в экономике.

1.2. Экономические информационные системы

В пособии будут рассматриваться АСУП или экономические информационные системы. *Экономическая ИС – ИС, связанная с управлением некоторым экономическим объектом.*

В экономическом объекте выделяют (рис. 1) систему управления (СУ) и объект управления (ОУ). ОУ выпускает товары или выполняет некоторые

услуги. СУ осуществляет воздействие на ОУ для организации функционирования и достижения целей управления, как правило связанных с развитием и совершенствованием функционирования ОУ. При этом ОУ получает из внешней среды необходимые ресурсы и материалы для производства товаров и услуг, а СУ – требования к производству, правила взаимодействия с другими экономическими объектами и сведения о взаимодействии. ОУ возвращает в среду произведенные товары и услуги, а СУ – информацию о деятельности экономического объекта.

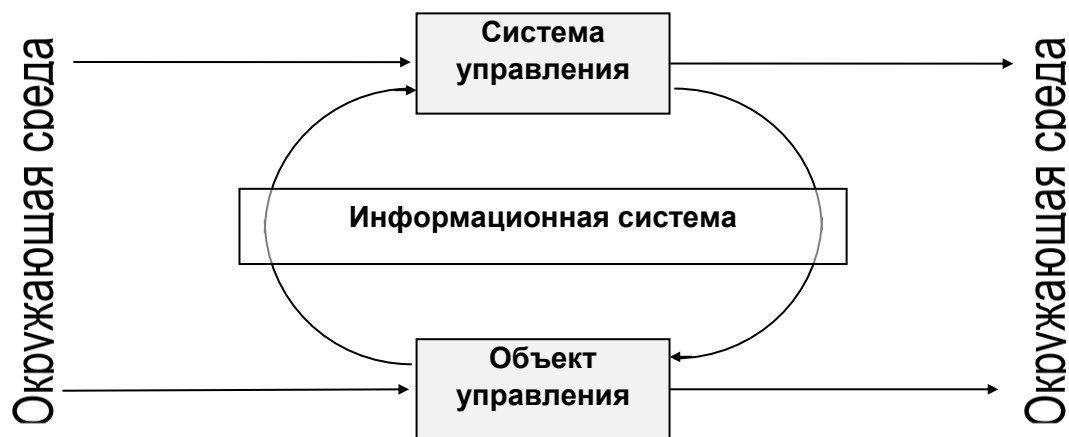


Рис. 1. Экономический объект

Экономические ИС характеризуются:

- большим объемом информации,
- цикличностью управления: получения и использования информации, многообразием источников и потребителей информации.

Экономическая информационная система становится необходимым компонентом контура управления. Она хранит все необходимые данные, описывающие состояние и динамику ОУ: данные планирования (планы, графики, расписания, плановые показатели...), детальные данные, характеризующие выполнение бизнес-процессов и позволяющие контролировать выполнение планов, данные, требуемые для учета взаимодействия с окружающей средой.

Относительно автоматизации функций управления определенного уровня (оперативного, тактического, стратегического) можно выделить следующие типы ИС [14] (рис. 2):

– COT – системы обработки транзакций (Transaction Processing/Data Systems – TPS) соответствуют самому низкому уровню управления (иногда его выделяют в отдельный операционный уровень) и связаны с регистрацией транзакций, под которыми здесь понимаются элементарные экономические события (сообщения). Обработка транзакций – это даже не уровень управления, а уровень операционистов, выполняющих рутинную работу по оформлению документов

(накладных, счетов и т.д.) и регистрации первичных документов. Задача управления на этом уровне заключается в организации делопроизводства, в измерении количества и качества выполняемой работы.

– УИС – управляющие информационные системы (Management Information Systems – MIS) предназначены для автоматизации точно определенных, формализованных информационных процедур, связанных, в основном, с уровнем тактического управления. Такие системы используют данные систем обработки транзакций для вычисления показателей деятельности предприятия и выявления отклонений фактических значений от плановых, а также для статистической обработки и прогнозирования.

– СППР – системы поддержки принятия решений (Decision Support Systems – DSS) применяют для содействия принятию решений плохо формализованных задач в условиях недостатка информации и большой неопределенности. Здесь могут применяться методы извлечения информации по запросам пользователя, позволяющие ему оценивать ситуацию с различных точек зрения, сценарии вида «что, если», методы имитационного моделирования, методы многомерного анализа (OLAP) и Data Mining. Системы этого типа связаны с высшими уровнями управления, достаточно специфичны и в технологическом плане требуют применения компонентов искусственного интеллекта.

– ЭС – экспертные системы (Expert Systems – ES) предназначены для выделения плохо формализуемых знаний и навыков специалистов – экспертов и для последующего тиражирования и использования экспертных систем в качестве консультантов для анализа и принятия решений. Экспертные системы находят широкое применение для диагностики, анализа различных ситуаций и оценки решений. Системы такого типа также применяются, в основном, на высших уровнях управления.

– ИСВР – информационные системы для высшего руководства (Executive Information Systems – EIS) предназначены для удовлетворения информационных запросов высшего руководства. Они предоставляют информацию в виде набора наиболее значимых для управления сведений, показателей и отклонений плановых и фактических значений. Однако, при необходимости они способны показать и детальную информацию, объясняющую значение того или иного показателя. В основном они предназначены не для ввода новых данных, а для извлечения информации по различным аспектам деятельности предприятия.



Рис. 2. Классификация ИС по автоматизации задач
определенного уровня управления

1.3. Подсистемы предприятия

Одной из самых полезных является классификация ИС по видам автоматизируемой деятельности. Такая классификация, в основном, совпадает с делением предприятия на подразделения и лежит в основе разделения информационной системы организации на взаимодействующие подсистемы. Функциональные характеристики ИС организации сильно зависят от отраслевой направленности предприятия. Планирование и учет деятельности сельскохозяйственного предприятия и банка будут существенно разными.

На предприятии выделяют основные и вспомогательные виды деятельности и соответствующие подсистемы должны учитывать их специфику. Исходными данными для планирования производства являются данные спроса на соответствующие товары и услуги и план сбыта продукции, а также сведения о производственных мощностях, персонале, ресурсах и материалах. На предприятии, как правило, создается несколько планов: укрупненный план стратегического развития, более детальный годовой план, и, наконец, детальный календарный график работы. Учет выполнения планов включает огромное количество событий: потребление материалов и ресурсов, выполнение операций, учет готовой продукции, передача изделий от одного производственного участка другому и многое другое.

График производства определяет практически все бизнес-процессы предприятия:

- материально техническое снабжение – обеспечение производства материалами и ресурсами в соответствии с графиком потребления;
- сбыт готовой продукции в соответствии с результатами производства;
- управление кадрами как важнейшим ресурсом производства и управления;
- управление бюджетами и денежными потоками – подсистема финансового управления;
- учет движения денежных и материальных ресурсов – подсистема бухгалтерского учета.

Перечисленные подсистемы часто дополняются многими другими для автоматизации многих других видов деятельности, например:

- управление взаимоотношениями с клиентами;
- электронный документооборот;
- маркетинг;
- конструкторская и технологическая подготовки производства;
- управление проектами.

Для исключения обмена данными между разными подсистемами используют ERP (Enterprise Resources Planning) – системы управления предприятием, реализующие функции всех перечисленных подсистем на единой технологической основе. Такие системы отличаются высокой сложностью, так как изменение в одной части системы затрагивают многие контуры учета и управления.

1.3.1. Обеспечение стратегического управления

Управление предусматривает увязывание всех уровней: стратегического – для достижения долгосрочных целей, тактического, определяющего способы достижения стратегических целей управления в виде комплекса задач, оперативного, составляющего графики решения задач и организующее их выполнение. Одним из подходов к стратегическому управлению является сбалансированная система показателей – ССП (Balanced Scorecard – BSC) [15], которая предназначена для создания системы целей и измерения их достижения, а также для вовлечения каждого подразделения и сотрудника в этот процесс.

Создатели ССП – Р. Каплан и Д. Нортон – осознали, что только финансовые цели и показатели не охватывают все стороны деятельности предприятия. Ими были предложены следующие четыре области (проекции или перспективы) управления:

Финансы – финансовые эффекты и показатели бизнес-процессов.

Клиенты – эффекты и показатели взаимоотношений с клиентами.

Внутренние бизнес-процессы – эффекты и показатели использования внутренних ресурсов.

Обучение и развитие – показатели состояния и развития человеческого капитала фирмы.

Области достаточно тесно связаны друг с другом (рис. 3). Чем выше квалификация персонала, тем эффективнее выполняются бизнес-процессы. Конечно, на бизнес-процессы влияют и многие другие факторы (качество сырья и ресурсов, техническая оснащенность, технологии...), однако, человеческие ресурсы составляют в зависимости от отрасли половину или большую долю всех затрат, а квалификация персонала напрямую влияет на количество и качество результатов работы.

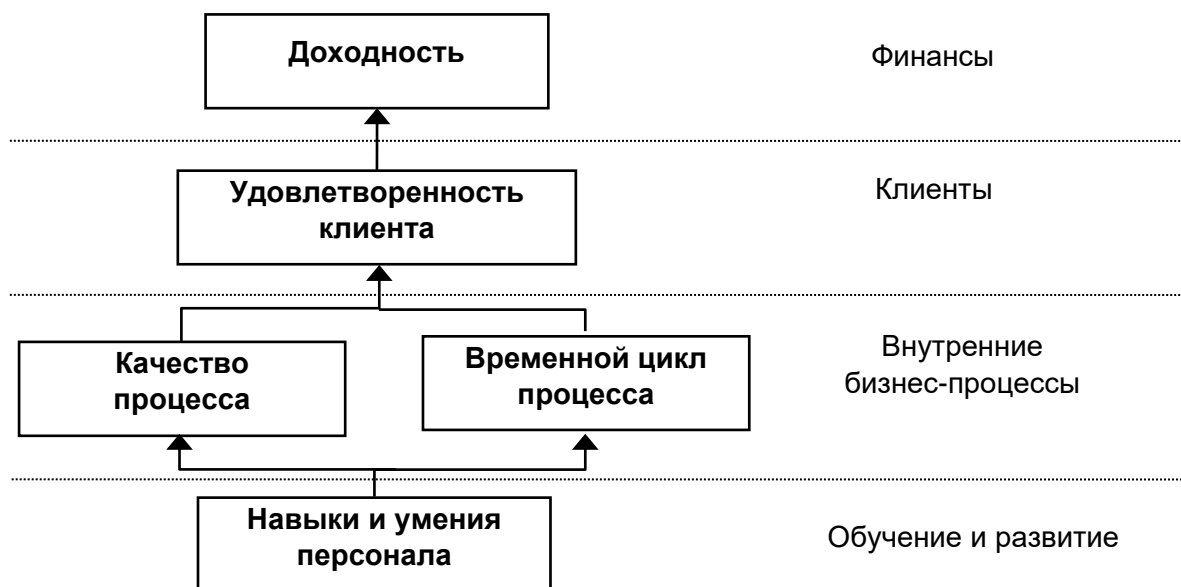


Рис. 3. Связи компонент ССП

Качество организации внутренних бизнес-процессов влияет на удовлетворенность клиентов. Многочисленные стандарты качества, начиная с ISO9000, переносят фокус с качества результата на качество исполнения, которое и должно гарантировать качество результата.

Удовлетворенность клиентов приводит к увеличению объемов и стоимости проданных товаров или оказанных услуг и, в конечном итоге, к возрастанию финансовых результатов.

Одно из положений ССП заключается в измеримости достижения целей. Для этого с каждой целью связываются ключевые показатели эффективности – KPI (Key Performance Indicators) и достижение цели связывается с достижением целевого значения KPI.

Цели могут противоречить друг другу. Например, повышение качества и снижение затрат. Очевидно, что для повышения качества нужно проводить обучение, выполнять модернизацию, приобретать более качественные материалы, усложнять технологию, что естественно приводит к увеличению затрат. Сбалансированность предполагает такую стратегию, которая обеспечивает достижение всех целей, а не какой-то одной.

Разработка сбалансированной системы показателей заключается в разработке изменения показателей по времени и распределении показателей по должностям (сотрудникам). Последнее особенно важно для мобилизации каждого работника на достижение стратегических целей.

ССП предназначена для получения следующих эффектов:

- Определение ключевых показателей эффективности (матрица KPI) для всех должностей по периодам.
- Периодическое получение систематизированных итоговых отчетов, содержащих иерархический перечень целей и KPI по всей компании. Отчеты должны отражать динамику достижения целей и вклад в достижение целей всех подразделений и сотрудников.

Для реализации управления на основе ССП необходима информационная подсистема, которая позволяла бы задавать матрицу KPI и контролировать фактически достигнутые значения KPI. Для определения фактических значений KPI необходимо собирать данные из разных подсистем, учитывающих затраты и результаты работы. Особую сложность такой подсистемы составляет то, что она должна предоставлять возможность определять любую возможную систему целей и KPI. Соответствующие компоненты ССП должны встраиваться в компьютерные инструменты для высшего руководства, чтобы каждый руководитель мог анализировать плановые и фактические KPI своих подразделений и подчиненных.

На рис. 4 приведен фрагмент карты стратегий вуза. В отношении всех показателей необходимо определение технологии формирования фактических значений показателей. Для приведенного примера большая часть показателей может быть получена из разных подсистем вуза. Однако, ряд показателей (отзывы, повышение квалификации) придется вводить вручную, другие показатели (данные трудоустройства) возможно получать из государственных систем учета.

Применение ССП заключается в создании карты стратегий, на которой отражают стратегические цели, разбитые по перечисленным выше четырем направлениям и их взаимное влияние. Цели могут влиять друг на друга и даже противоречить друг другу. Далее определяют критические факторы успеха (КФУ) – наиболее значимые факторы, которые влияют на достижение целей. Одни и те же факторы могут влиять на разные цели. Наконец для измерения достижения целей и использования факторов определяют ключевые показатели результативности (KPI).

Для описания показателя формируется паспорт показателя следующего содержания:

- название KPI;
- единица измерения KPI;
- цель, которую измеряют;
- частота измерения показателя;
- алгоритм расчета показателя;
- данные, используемые при расчете показателя (описание, источник и алгоритм расчета);

- подразделение, отвечающее за измерение КРІ;
- целевое значение КРІ.



Рис. 4. Фрагмент карты стратегий учебного заведения

Устанавливаются плановые значения целевых показателей и ключевых показателей результативности на стратегический и тактический периоды. Для каждого подразделения разрабатываются сбалансированные счетные карты, отражающие цели, КФУ, КРІ подразделения. Достижение этих значений периодически отслеживается и используется для принятия решений по изменению планов/целей/стратегий.

Показатели, полученные в результате применения ССП и измеряющие достижение стратегических целей, называют стратегическими, в отличие от диагностических, которые сигнализируют об отклонении от нормального выполнения бизнес-процесса.

Инновационный проект проходит стадии, изображенные на рис. 5:

1. Инновации – длинная волна, включающая определение потребительских свойств, конструирование, разработку технологии, создание или модификация технологической линии.

2. Операции – короткая волна от получения заказа до доставки, финансовые показатели короткой волны включают нормативные и фактические издержки, бюджет, отклонения, к нефинансовым относят показатели качества, продолжительности операций.

3. Послепродажное обслуживание характеризуется набором показателей, который сильно зависит от вида продуктов.

В разные моменты времени на первое место выходят разные показатели в зависимости от целей управления:

1. Для стадии роста показатели должны характеризовать потенциал, который нужно реализовать, а также затраты на инвестирование. Типичными целями этой стадии являются рост дохода, расширение деятельности.

2. Для устойчивого состояния показатели должны характеризовать эффективность бизнес-процесса и его доходность. На этой стадии целями являются сокращение издержек, увеличение производительности.

3. На стадии сбора «урожая», когда нет инвестиций, а на первый план выходит доходность и целью является увеличение доходности.

В зависимости от стадии применяют разные показатели, характеризующие стоимостную цепочку (рис. 5).

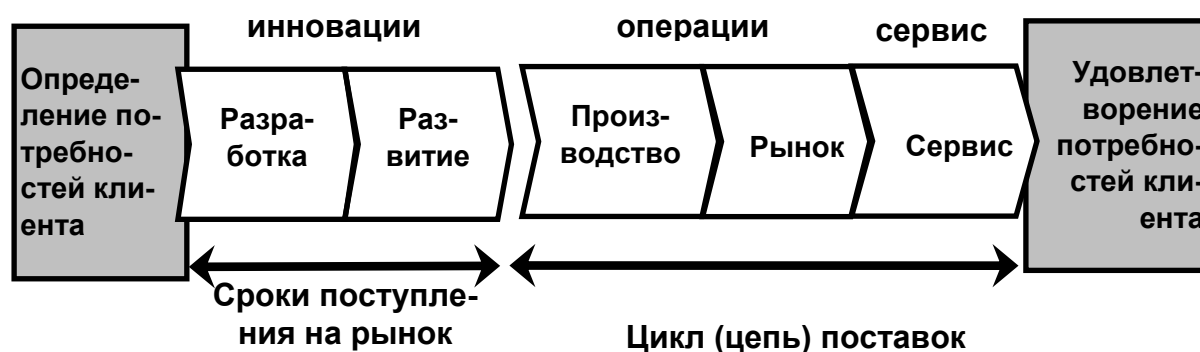


Рис. 5. Стоимостная цепочка продукта

Организационная единица ССП – подразделение с самостоятельными стратегией, клиентами (внешние или внутренние), внутренним бизнес-процессом. ССП – инструмент для:

- четкой формулировки стратегии и выработки общего мнения;
- доведения стратегии до всех работников;
- согласования цели и задач каждого подразделения, работника со стратегией компании;
- согласования стратегических задач с долгосрочными целями и годовыми бюджетами;
- идентификации и систематизации стратегических инициатив;
- систематического проведения обзоров достигнутых стратегических результатов;

– создания обратной связи для получения информации о реализации стратегии и корректировки ее при необходимости.

Методика исследования показателей, основанная на детализации и агрегировании, является простейшим, и в то же время достаточно эффективным инструментом, который лежит в основе технологии OLAP.

Одним из методов стратегического управления является проектное управление. Проект определяют как совокупность действий, имеющих временный характер и общую цель по созданию уникального продукта, услуги или любых других уникальных результатов. На практике же проектом можно назвать любую деятельность, которую руководство компании решает контролировать отдельно от операционных задач.

Любой проект начинается с планирования. В плане отражаются цели проекта, план мероприятий, ресурсы, необходимые для реализации проекта, финансовый план и бюджет проекта. Особенностью проектного управления является высокая неопределенность и риски проекта. Выполнение проекта сопровождается постоянными корректировками параметров и плана проекта. Системы управления проектами ориентированы на построение бизнес-плана и контроль его выполнения. В качестве примеров можно привести Project Expert¹ и MS Project². Программа перспективного планирования должна учитывать следующие факторы:

– параметры среды (валюта, инфляция, налоги, условия переоценки основных фондов):

– уровень инфляции по группам: деньги, материалы, зарплата, фонды...;

– условия платежей (средние по отрасли дебиторская и кредиторская задолженности);

– налоги: с продаж, на валовую прибыль (доход), на прибыль, на активы, на ресурсы, на зарплату...;

– условия предоставления займов: процент по долгосрочным (более года) и краткосрочным (менее года) кредитам, уровень процентных ставок;

– инвестиционные данные (затраты, сроки, условия амортизации) на подготовительный период, землю, здания, коммуникации, оборудование;

– данные о сбыте: товары / услуги, цены, объемы, затраты на маркетинг (реклама, продвижение);

– данные об издержках: прямые (на единицу продукции), постоянные, административные, торговые.

Программа позволяет получить:

– расчет потребностей в капитале (отчет о прибылях и убытках, баланс, план денежных потоков);

– план формирования капитала (формирование собственного и заемного капитала – источники, условия);

¹ URL: <https://www.expert-systems.com/financial/pe>.

² URL: <https://www.microsoft.com/ru-ru/microsoft-365/project/project-management-software?market=ru>.

- расчет показателей эффективности: рентабельности, показателей платежеспособности и ликвидности, период окупаемости, индекс прибыльности, чистая приведенная величина прибыли, внутренняя норма доходности;
- распределение свободного капитала (дивиденды, депозиты);
- анализ чувствительности – нахождение интервальных оценок.

Для управления проектами используют специальные технологии гибкого управления¹: канбан, agile и другие. Одной из распространенных программ обеспечения гибкого управления проектами является Trello², доступная бесплатно в усеченной функциональности.

1.3.2. Управление производством

Подсистемы управления производством являются одними из самых сложных. Планирование в таких системах имеет несколько уровней:

- Долгосрочные планы отражают стратегические цели.
- Среднесрочные планы включают укрупненные планы производства и потребления ресурсов, планы загрузки, модернизации и ремонта оборудования, планы поставок.
- Краткосрочные планы (несколько недель, месяцев) включают графики выпуска продукции, производства компонент, материально-технического снабжения, загрузки оборудования.

Планирование включает следующие этапы:

- Прогноз продаж и определение спроса по видам продукции.
- Вычисление по объемам спроса численности рабочих, производственных мощностей, материальных ресурсов.
- Построение оптимального графика загрузки оборудования и использования ресурсов с учетом множества ограничений: производственные мощности, материальные, финансовые, человеческие ресурсы, проведение ремонтов и других мероприятий.

Исходные данные планирования включают спрос на выпускаемую продукцию, описание конструкции, описание технологии.

В зависимости от типа производства и состояния рынка применяют разные стратегии планирования. Например, сбалансированный план обеспечивает выпуск продукции, соответствующей спросу (минимизируется запас готовой продукции). План с фиксированным уровнем мощности ориентирован на равномерную загрузку производства, а отклонение производства от спроса компенсируется запасами готовой продукции.

Конструкторская подготовка производства предусматривает не только создание конструкции и документации к ней, но ведение данных, описывающих составы изделий. Технологическая подготовка производства создает информацию о разработанной технологии производства. Описание технологического процесса должно предусматривать для каждого изделия последовательность

¹ URL: <https://vc.ru/services/99244-obzor-po-dlya-upravleniya-proektami>.

² URL: <https://trello.com>.

операций и характеристики всех операций. По перечисленным данным выполняются следующие расчеты:

- разузлование – вычисление для каждого изделия (сборочной единицы) составляющих ее сборочных единиц, деталей и материалов. Для сборочных единиц вычисляют уровень вхождения – уровень иерархии в дереве состава изделия;
- расцеховка – установление порядка цехов и операций в технологическом маршруте изготовления изделия;
- потребности в материалах, готовых деталях и сборочных единицах;
- потребности в трудовых ресурсах;
- потребности в оборудовании, оснастке, инструментах;
- определение себестоимости как суммарной стоимости материалов и комплектующих и стоимости трудозатрат, увеличенных на процент накладных расходов;
- определение плановой цены – себестоимости, увеличенной на процент отчислений и плановой прибыли.

Перечисленные расчеты создают нормативную базу для планирования производства.

Оперативное планирование производства выполняется на основании контрактов на поставку товаров/услуг и/или плана производства:

- выбираются контракты на плановый период и формируется план-график выпуска продукции,
- рассчитываются потребности в материалах, финансах, рабочей силе, составляется план загрузки оборудования.

Результатом является

- план мероприятий по подготовке и обеспечению производства,
- план-график производства по изделиям и исполнителям,
- ведомости загрузки оборудования,
- потребности в материалах, стандартных комплектующих по времени (лимитно-заборные карты).

В процессе реализации производственного плана ИС позволяет вести учет затрат на производство и вычислять соответствующие показатели:

- учет фактических объемов выпуска:
 - готовых изделий;
 - учет незавершенного производства;
- расчет:
 - фактических затрат:
 - по статьям;
 - по подразделениям и предприятию в целом;
 - фактической себестоимости.

Составной частью управления производством является управление ремонтом. Выделяют следующие виды ремонта: техническое обслуживание, текущий и капитальный ремонты. Для управления ремонтами необходима нормативная база, описывающая оборудование, его классификацию и обслуживание. График

ремонт должен быть синхронизирован с планом-графиком производства. По графику ремонтов вычисляются:

- потребности в запасных частях;
- анализ обеспеченности ремонтов материальными и трудовыми ресурсами.

В процессе производства необходимо учитывать:

- выполнение ремонтов;
- простои.

Данные о загрузке и простоях оборудования позволяют вычислять показатели загрузки оборудования.

1.3.3. Маркетинг и сбыт

Управление сбытом является важным направлением, так как именно сбыт приводит к получению дохода. Доход позволяет компенсировать расходы и развивать предприятие. Маркетинг – это деятельность по продвижению продукции от производства до её употребления. Повышенное внимание уделяется методам коммуникации с потребителем, учитываются его взгляды, предпочтения, вкусы. Центральное место в концепции маркетинга занимает реклама, то есть активное информирование потенциального покупателя о свойствах продукции. Сбыт – это деятельность, направленная на реализацию продукции, произведённой предприятием. Отдельные сферы экономики могут обходиться и без маркетинговых схем продвижения продукта. Это относится к военно-промышленному комплексу, добывающей промышленности, лесозаготовке и т.д. Здесь покупатель часто сам ищет продавца. Сбыт – это прямые продажи, которые могут осуществляться с предварительным заказом или без него.

Автоматизация управления маркетингом предусматривает выполнение следующих операций:

- ведение информации о товарах и услугах;
- управление каналами сбыта;
- анализ рынка рекламных услуг, планирование рекламы, размещение рекламы, анализ эффективности рекламы;
- сбор и обработка отзывов о товарах;
- ведение досье на конкурентов и продукты-аналоги;
- контроль жизненных циклов продуктов;
- регистрация серийных продаж, учет рекламаций, гарантий;
- маркетинговый анализ по каналам сбыта, товарам, группам товаров, направлениям реализации.

Управление сбытом обеспечивает выполнение следующих функций:

- регистрация и обработка контрактов;
- ведение отпускных цен и скидок;
- оформление документов на продажу: документ – основание продажи со списком товаров, накладная;
- учет наличия товаров на складе готовой продукции;
- резервирование товаров;
- учет возврата товаров;

- прогнозирование продаж;
- формирование отчетов:
 - об исполнении заказов;
 - о реализации по товарам, группам, партиям, классификаторам, получателям;
- анализ реализации по периодам;
- пересчет себестоимости;
- учет расчетов с заказчиками.

В ИС хранятся данные о заказчиках (в том числе потенциальных), заказах, данные складского учета, данные расчетов с заказчиками, данные маркетинговых исследований.

Оперативное управление связано с отслеживанием документооборота: кто, когда и как принимает заказы, как организована комплектация партии отгрузки, отгрузка и транспортировка, как учитывается товар на складе, как соблюдаются правила учета дебиторской задолженности.

Тактическое управление связано с оценкой процессов сбыта: выполняется ли продвижение товара на рынке в соответствии с планом, насколько продуктивно выполняется оформление заказов и отгрузка, каково среднее и максимальное время оплаты.

Стратегическое управление решает следующие вопросы:

- исследование рынка, прогнозирование объемов и ассортимента продаж;
- формирование рыночных потребностей (реклама, продажи);
- прогнозирование цен на продукты;
- организация и стимулирование сбыта и послепродажного обслуживания;
- ориентация научно-технических и производственных подразделений на потребности рынка;
- разработки стратегии продвижения продукции на рынок;
- формирование и изменение списка заказчиков;
- выбор технологии выполнения заказов – привлечения субподрядчиков, организации складского учета (количество, расположение, техническая оснащенность), выбор методов уменьшения времени оплаты и дебиторской задолженности).

1.3.4. Материально-техническое снабжение

Материально-техническое снабжение – процесс обеспечения предприятия всеми видами материальных и технических ресурсов в сроки и в объемах, необходимых для бесперебойного осуществления его производственной деятельности. Сбои в снабжении болезненно сказываются на деятельности предприятия. Именно со снабжения начиналась автоматизация управления производством.

Приобретение товаров и услуг сторонних организаций и учет расходов на эти товары и услуги предусматривает:

- ведение картотеки предложений;
- учет заявок подразделений;
- составление плана закупок;

- составление заказов, учет их выполнения;
- распределение материалов по складам и организация хранения;
- учет взаимоотношений (поступление, оплата, рекламации) с поставщиками;
- складской учет:
 - приходование;
 - возвраты и рекламации;
 - отпуск;
 - внутренние перемещения;
 - нормативные запасы;
 - сверхнормативные запасы;
 - неликвиды;
 - учет партий и сроков хранения;
 - формирование документов складского учета: доверенностей, приходно-расходных ордеров;
 - ведение учетных цен и поддержание методик списания (LIFO, FIFO, средневзвешенные);
 - формирование оборотных ведомостей по складам;
- формирование отчетов:
 - по товарам, группам, партиям, классификаторам, поставщикам;
 - о состоянии запасов;
 - о состоянии заказов на закупку.

В подсистеме хранятся данные о поставщиках, каталогах, прайс-листах, заказах, фактах получения, оплате и других событиях, связанных с закупкой и поставкой заказанных материалов.

Оперативное управление связано с отклонениями в графиках снабжения: полнота документации о заказываемой продукции, соответствие заказываемой продукции потребностям, исполнение сроков и условий поставки, правильность учета кредиторской задолженности и своевременность оплаты.

Тактическое управление связано с выбором поставщиков, товаров, управлением соответствия «цена-качество», соответствием динамики поступления и расходования материалов, процессами порчи, пропажи, потери; управлением процессом оплаты (применение скидок и дисконтов, кредитов).

Стратегическое управление включает разработку стратегии и политики размещения заказов, определение страховых запасов и размеров складов, эффективность расходования денег на приобретение товаров и услуг.

1.3.5. Управление персоналом и оплата труда

Управление персоналом – специализированная функция в организациях, нацеленная на управление наемным трудом. Эту функцию выполняет особое подразделение («отдел кадров», «служба управления персоналом», «департамент HR»). Потребности в трудовых ресурсах определяются на основе планов

работы всего предприятия и оформляются в виде штатного расписания. В штатном расписании перечисляются все подразделения и должности с указанием для каждой должности вилки оклада и требований к квалификации. Подбором сотрудников занимается, как правило, руководитель подразделения, оформлением документов – отдел кадров.

В подсистеме решаются следующие задачи:

- кадровый маркетинг – прогнозирование потребностей предприятия, формулирование требований к персоналу;
- подбор кадров;
- аттестация, расстановка и продвижение кадров;
- организация материального и нефинансового стимулирования;
- управление и совершенствование социально – психологического климата;
- автоматизация ведения личных дел сотрудников, в личное дело включают:
 - общие сведения;
 - данные об образовании;
 - анкетные данные;
 - наличие родственников;
 - данные воинского учета;
 - список назначений и перемещений;
 - отпуска;
 - болезни;
- планирование и управление штатным расписанием предусматривает:
 - описание структуры предприятия;
 - ведение списка рабочих места с привязкой к подразделениям;
 - учет заполнения вакансий;
 - учет трудовых затрат;
 - учет объема выполненной работы;
 - регистрация табелей (по одному на ставку);
 - учет больничных;
- формирование стандартных отчетов.

1.3.6. Управление финансами (планово-финансовый отдел)

Планирование и управление финансами включает следующие виды деятельности:

1. Создание, поддержка актуальности, контроль и оценка различных бюджетов (бюджет доходов и расходов, бюджет движения денежных средств, операционные бюджеты продаж, производства, запасов, прямых материальных затрат, прямых затрат труда).
2. Разработка и мониторинг кредитно-инвестиционного плана, бюджета реконструкции производства, бюджета налоговых выплат.
3. Анализ операционной деятельности (в сравнении с депозитами).
4. Разработка и сравнение схем проведения коммерческих операций (проектов).

5. Анализ влияния различных прогнозируемых факторов (курс рубля, сезонность, рост оплаты труда) на бюджеты.

Управление финансами предусматривает следующую последовательность действий:

1. Разработка структуры бюджетов предприятия (иерархия статей поступлений и затрат).
2. Составление бюджетов по видам деятельности.
3. Детальный анализ плана по видам деятельности, подразделениям по периодам.
4. Обобщение финансовых планов в единый план.
5. Проведение анализа влияния факторов на показатели плана.
6. Регистрация выполнения финансового плана и разноска поступлений и затрат по видам деятельности и подразделениям.
7. Контроль хода выполнения плана за прошедший период (план – факт).
8. Анализ выполнения плана по направлениям и видам деятельности.
9. Управление операциями в сфере денежного обращения, на рынках инвестиций, ценных бумаг и валюты для поддержания нужного баланса между активами и пассивами, доходами и расходами.

Задачи:

- контроль за балансом активов и пассивов и соответственно, поддержание кредитоспособности предприятия (капитал = активы – пассивы; при коэффициенте задолженности (долги/капитал) больше 2 начинается проверка платежеспособности);
- контроль за балансом наличных расходов и поступлений для поддержания платежеспособности (наличность на начало периода, поступление, расход, остаток на конец периода);
- контроль за счетом прибылей и убытков: определение порога безубыточности – минимальной активности предприятия;
- операции с ценными бумагами;
- эмиссии акций, векселей и других ценных бумаг;
- выплата дивидендов и паев;
- взаимодействие с банками и инвестиционными фондами – получение и возврат кредитов;
- валютные расчеты;
- финансовый контроль.

1.3.7. Бухгалтерский учет

Бухгалтерский учет объединяет и измеряет деятельность всех подразделений, составляет отчеты для высшего руководства. Учет ведется в виде журнала хозяйственных операций. Каждой операции соответствует набор проводок (переводов сумм со счета на счет). Для удобства используется справочник типовых хозяйственных операций (ТХО), который для каждой ТХО содержит:

- наименование ТХО;
- корреспонденцию счетов и алгоритмы проводок.

Выбирая ТХО, бухгалтер выполняет проводку операции по счетам. Например, покупателям отгружена готовая продукция и выставлен счет:

1. Проводится себестоимость:

- Дебет 46 – Реализация продукции.
- Кредит 40 – Готовая продукция.

2. Проводится договорная стоимость:

- Дебет 62 – Расчеты с покупателями и заказчиками.
- Кредит 46 – Реализация продукции.

3. Если себестоимость > договорная стоимость, то проводится прибыль = Договорная стоимость – Себестоимость:

- Дебет 46 – Реализация продукции.
- Кредит 80 Прибыли и убытки.

4. Если себестоимость < договорная стоимость, то проводится убыток = Себестоимость – Договорная стоимость:

- Дебет 80 – Прибыли и убытки.
- Кредит 46 – Реализация продукции.

Регистрация операций включает:

- ввод сведений о документе;
- проводка по счетам синтетического учета;
- ввод данных для аналитического учета.

Перечислим некоторые первичные документы:

- складской учет:
 - приходный складской ордер;
 - расходный складской ордер;
 - накладные на внутреннее перемещение;
- покупка – продажа:
 - документы – основания (договоры);
 - счета – фактуры;
 - акты оказания услуг;
 - накладные на прием – отпуск;
 - накладные на возврат;
- производство:
 - накладные на отпуск материалов в производство;
 - накладные на прием готовой продукции;
- касса:
 - приходные кассовые ордера;
 - расходные кассовые ордера;
 - документы – основания;
 - ведомости на оплату / выплату;
- расчетный счет:
 - платежное поручение;
 - платежные требования;
 - чеки;

– зарплата;

• документы по учету количества труда: табель, наряд на выполненные заботы, акт о выполнении работы;

• оплаты (>160 видов);

• удержания (>60 видов);

• отчисления в ПФ, фонд обязательного медицинского страхования, фонд социального страхования;

• подоходный налог.

В журнале должен быть представлен аналитический учет по направлениям:

– объекты (товарно-материальные ценности, основные средства, нематериальные активы, ценные бумаги, капитальное строительство);

– центры ответственности (подразделения, подотчетные лица);

– центры затрат (затраты производства, изделия, заказы, договоры);

– прочие (виды деятельности, виды прибылей и убытков, льгот и т.д.).

Учет товарно-материальных ценностей (ТМЦ) по следующим видам:

– малоценное и быстроизнашивающееся оборудование;

– объекты физического учета.

Основные операции по учету ТМЦ:

– приход, расход (счета, ордера);

– перемещение;

– отчеты;

– инвентаризация (ввод остатков);

– переоценка.

Учет материальной ответственности по подотчетным лицам:

– учет денежных средств;

– учет материальных ценностей.

На основании журнала операций формируются многие другие документы:

– документы внутреннего использования:

• журнал хозяйственных операций;

• главная книга – по счетам: начальное сальдо, обороты, конечное сальдо;

• оборотный баланс;

• ведомости аналитического учета;

– исходящая отчетность:

• баланс с приложениями;

• отчеты по налогам и отчислениям;

– экономические показатели:

• аналитический баланс;

• расчет рентабельности;

• ликвидность;

• оборачиваемость.

Системы автоматизации бухгалтерского учета должны обладать значительными возможностями по адаптации, в связи с изменениями в формах и правилах ведения учета и формирования отчетности.

2. МЕТОДОЛОГИЯ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1. Жизненный цикл информационной системы

Жизненный цикл информационной системы (программы) – это процесс ее построения и развития. В жизненном цикле ИС выделяют стадии, а каждую стадию разбивают на этапы. Для каждого этапа описывают выполняемые работы и участие в них системных аналитиков, программистов и пользователей.

Существует несколько стандартов описания жизненного цикла ИС. Один из них – это стандарт ГОСТ 34.601-90 [9], который предусматривает следующие стадии и этапы создания автоматизированной системы (АС):

1. Формирование требований к АС.
 - 1.1. Обследование объекта и обоснование необходимости создания АС.
 - 1.2. Формирование требований пользователя к АС.
 - 1.3. Оформление отчета о выполнении работ и заявки на разработку АС.
2. Разработка концепции АС.
 - 2.1. Изучение объекта.
 - 2.2. Проведение необходимых научно-исследовательских работ.
 - 2.3. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователей.
 - 2.4. Оформление отчета о проделанной работе.
3. Техническое задание.
 - 3.1. Разработка и утверждение технического задания на создание АС.
4. Эскизный проект.
 - 4.1. Разработка предварительных проектных решений по системе и ее частям.
 - 4.2. Разработка документации на АС и ее части.
5. Технический проект.
 - 5.1. Разработка проектных решений по системе и ее частям.
 - 5.2. Разработка документации на АС и ее части.
 - 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку.
 - 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации.
6. Рабочая документация.
 - 6.1. Разработка рабочей документации на систему и ее части.
 - 6.2. Разработка и адаптация программ.
7. Ввод в действие.
 - 7.1. Подготовка объекта автоматизации к вводу АС в действие.
 - 7.2. Подготовка персонала.

7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями).

7.4. Строительно-монтажные работы.

7.5. Пуско-наладочные работы.

7.6. Проведение предварительных испытаний.

7.7. Проведение опытной эксплуатации.

7.8. Проведение приемочных испытаний.

8. Сопровождение АС.

8.1. Выполнение работ в соответствии с гарантийными обязательствами.

8.2. Послегарантийное обслуживание.

Первые три стадии связаны с определением того, что нужно автоматизировать и как это следует делать. Этим занимаются бизнес-аналитики и эксперты в области применения информационных технологий. Подробно изучается выбранный для автоматизации бизнес-процесс, описывается соответствующая циркуляция информации, изучается технология его управления. Особое внимание уделяется возникающим при этом проблемам. Отыскиваются пути их решения на основе информационных технологий. Варианты автоматизации оцениваются с точки зрения их результативности и стоимости. Принятые решения фиксируют в виде технического задания на проектирование автоматизируемой системы.

Эскизный, технический проекты и рабочая документация – это, по существу, последовательное построение все более точных проектных решений по всем видам обеспечения информационной системы. Допускается исключать стадию «Эскизный проект» и отдельные этапы работ на всех стадиях, объединять стадии «Технический проект» и «Рабочая документация» в «Технорабочий проект», выполнять отдельные этапы работ до окончания предшествующих стадий, разрешается параллельное выполнение этапов работ и включение дополнительных этапов работ. На всех стадиях и этапах разрабатывается проектная документация (табл. 2).

Таблица 2

**Документационное обеспечение жизненного цикла
информационной системы**

Стадии создания АС в соответствии с ГОСТ 34.601-90	Документы
1. Формирование требований к АС	Отчет и заявка на создание АС
2. Разработка концепции АС	Отчет о НИР, технико-экономическое обоснование АС
3. Техническое задание	Техническое задание
4. Эскизный проект	Схема организационной структуры, схема функциональной структуры, схема комплекса технических средств, модель данных «сущность – связь»
5. Технический проект	Описание функций и задач, описание информационного обеспечения, описание математического обеспечения

Стадии создания АС в соответствии с ГОСТ 34.601-90	Документы
6. Рабочая документация	Общее описание системы, описание технологического процесса обработки данных, описание программного обеспечения, руководство пользователя
7. Ввод в действие	План-график работ, протокол испытаний, протокол опытной эксплуатации, акты приемки в опытную и промышленную эксплуатацию
8. Сопровождение АС	Заявки пользователей, акты выполнения работ

Существует много других стандартов, регламентирующих разработку информационных систем или их компонент. Популярный среди разработчиков зарубежных стандартов [16] предусматривает следующие стадии жизненного цикла программного средства:

1. Системный анализ и разработка спецификации. Спецификация содержит требования к программе и по содержанию аналогична техническому заданию.

2. Предварительное (внешнее) проектирование. На этой стадии принимаются базовые решения по архитектуре программного средства: выделение самостоятельных программных компонент, определение регламента их взаимодействия, функций, входных, выходных и хранимых данных, интерфейса пользователя.

3. Детальное (внутреннее) проектирование – детализация структур входных, выходных и хранимых данных, построение алгоритмов обработки данных, структуры и внутреннего интерфейса программных компонент.

4. Кодирование и отладка компонентов – создание программной реализации компонентов программного средства, выявление и исправление ошибок.

5. Интеграция и комплексная отладка – объединение компонентов в единое программное средство, выявление и исправление ошибок взаимодействия компонент.

6. Испытание и документирование – проверка работоспособности и создание различных видов документации для развития, сопровождения и использования программного средства.

7. Поддержка эксплуатации – выполнение штатных процедур (резервное копирование, проверка работоспособности, контроль целостности данных...).

8. Сопровождение – выявление и устранение ошибок, развитие и адаптация программ, измерение и увеличение производительности.

Несмотря на различие объектов (автоматизированная система и программа) двух приведенных стандартов можно считать, что они относятся к одному классу, так невозможно разрабатывать программу без учета ее применения (без организационного обеспечения), без учета структур данных (информационного обеспечения), с которыми программа будет работать, наконец, без учета

программно-аппаратной среды (технического обеспечения). Следует подчеркнуть близость приведенных стандартов: оба используют пошаговое уточнение проектных решений и предлагают подробную систему стадий и этапов.

Жесткая последовательность стадий и этапов разработки ИС известна как каскадная схема и была сформулирована в 1960-е годы. Достоинствами ее являются:

- управляемое построение ИС;
- юридически зафиксированная ответственность участников разработки: как разработчиков, так и заказчиков;
- возможность применения структурных методов разработки;
- развитие ИС в соответствии с определенными функциями и бюджетом.

В то же время каскадной методологии разработки присущи следующие недостатки:

- избыток согласующей документации;
- в начале разработки предполагается идеальный прогноз будущей ситуации;
- последовательный характер выполнения работ;
- обнаружение ошибок откладывается на последний этап;
- неспособность произвести работающую ИС за короткое время.

Излишняя жесткость каскадного подхода мешала оперативно вносить изменения в проект при обнаружении ошибок или изменений условий функционирования. Для устранения недостатков каскадного проектирования в 70-х годах был предложен итерационный подход, который, не меняя содержания работ, предлагает по результатам каждого этапа переосмысление и изменение при необходимости результатов предыдущих этапов. Гибкость процесса проектирования была достигнута, но ценой постоянного пересмотра проектных решений, планов и бюджетов.

В 1980-х годах появился спиральный подход, который заключается в построении работающего ядра или прототипа системы с последующими доработками, модификациями, а главное – развитием и расширением функциональных возможностей.

Спиральный жизненный цикл информационной системы (рис. 6) по методике Microsoft Solution Framework включает следующие фазы [5]:

- Анализ – разработка концепции продукта.
- Планирование – создание архитектуры системы и плана проектирования.
- Разработка – выработка полнофункционального продукта.
- Стабилизация – доработка продукта до стабильного состояния, готового к развертыванию и эксплуатации.

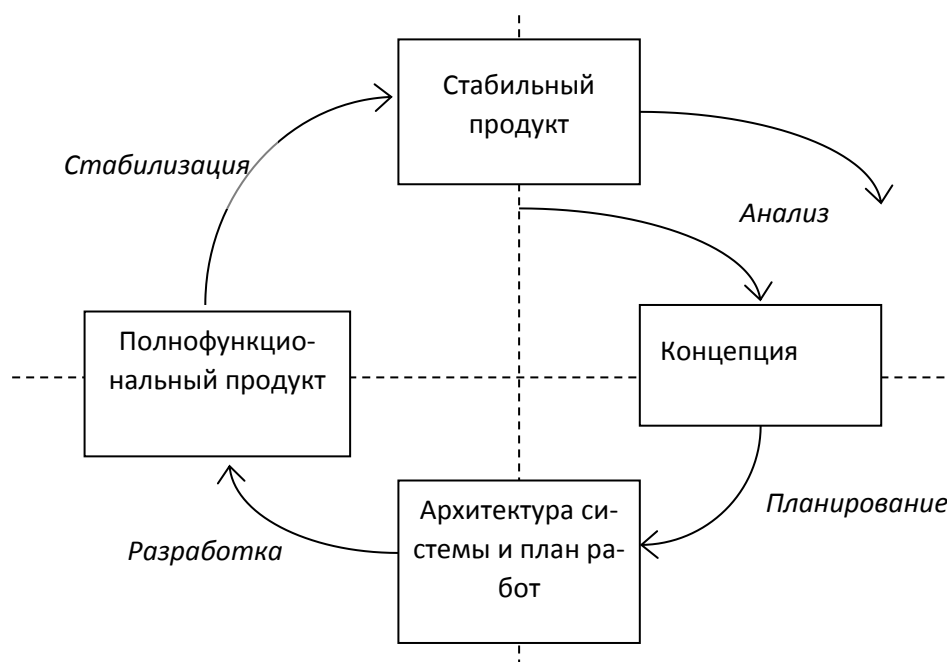


Рис. 6. Спиральный подход к проектированию информационных систем
Microsoft Corporation

Продукт проходит все четыре фазы неоднократно с целью последовательного уточнения характеристик и архитектуры для изменения и увеличения количества автоматизированных функций.

Спиральный подход во многом снимает проблемы каскадного проектирования. В приемлемое время заказчик получает первую очередь продукта, может оценить полученный результат и принять обоснованное решение об изменении и дальнейшем развитии продукта.

В то же время спиральный подход порождает ряд проблем:

- задержки в получении полнофункциональной информационной системы;
- возможность получить «никогда не заканчивающуюся» разработку;
- сложность в определении стоимости всех итераций проекта;
- отсутствие стабильности продукта в силу постоянных его изменений.

Объединению стадий (или «размыванию» границ между стадиями) способствуют современные технологии проектирования:

1. Joint Application Development (JAD) – организация совместной работы пользователей, менеджеров и разработчиков ИС для определения требований, приоритетов и, собственно, проектирования ИС.

2. Prototyping – построение и/или использование прототипов для определения требований, приоритетов и проектирования.

3. Rapid Application Development (RAD) – методы быстрой разработки программ.

4. Computer Aid System Engineering (CASE) – автоматизированные технологии разработки ИС.

Спиральные технологии проектирования ИС и инструментальные методы разработки ИС привели к появлению гибких (agile) методов проектирования, в

которых разработка ведется короткими итерациями (около четырех недель). За итерацию заказчик получает полностью разработанную часть функций. Проект начинается с планирования функций, набор и свойства которых могут изменяться в процессе проектирования. Гибкие методы требуют особых отношений заказчика и разработчика. Персонал заказчика принимает участие в проектировании. Это снижает риски, но процесс проектирования становится менее определенным.

Между стадиями различных моделей жизненного цикла можно установить соответствие. Для любой модели жизненного цикла выполняются следующие виды работ:

Анализ бизнес-процесса. Исследуются и документируются операции, составляющие бизнес-процесс: учет операций и результатов бизнес-процесса, потребности в материалах и ресурсах, а также управление бизнес-процессом: планирование, организация выполнения, учет выполнения работ, потребление материалов и ресурсов, результатов выполнения, контроль и корректировка планов. Это позволяет понять потребности пользователей в компьютерных инструментах для выполнения функций учета и управления. При этом важно предложить качественные улучшения работы пользователей на основе информационных технологий. Это наиболее сложная деятельность для проектировщика, так как она требует изучения профессиональных навыков различных специалистов. На этом этапе особенно важно взаимодействие с персоналом заказчика. В результате должно появиться представление, как следует автоматизировать работу объекта в целом и каждого пользователя.

Разработка требований к информационной системе. Требования включают разработку архитектуры проекта в виде отдельных серверных и клиентских компонент. К серверным компонентам относят сервер баз данных, сервер приложений, интернет-сервер. Клиентские компоненты обеспечивают, прежде всего, пользовательский интерфейс и, возможно, некоторые функции по обработке данных. К ним относят десктопные приложения, мобильные приложения, веб-страницы сайта, веб-приложения. Решаются также вопросы о распределении обработки данных среди компонент системы.

Другая часть требований связана с выделением отдельных функций ИС и разработкой особенностей реализации каждой функции в ИС. Приводятся также и другие требования к разным компонентам ИС. Требования документируют в виде технического задания (ТЗ). Несмотря на «древность» этого документа, он является основным для определения объемов работ, сроков и качества их исполнения. Техническое задание включается в договор между заказчиком и разработчиком. В гибких технологиях ТЗ может изменяться на каждой итерации, но изначально является основным документом для предварительной оценки стоимости и сроков разработки.

Конструирование и кодирование. Создаются и документируются компоненты информационной системы: структуры хранения данных, запросы на сохранение и извлечение данных, обработка данных в серверных и клиентских компонентах, интерфейс пользователя для ввода данных, команд и парамет-

ров, отображение данных в экранных формах приложений или на сайте. Решаются вопросы распределения полномочий и разрабатываются инструкции пользователям.

Внедрение и поддержка. Эти процессы сопровождаются большой организационной работой, которая осуществляется заказчиком при активном участии разработчиков. Выполняется развертывание системы на стороне заказчика, распределяются полномочия – создаются условия для начала эксплуатации системы, происходит обучение пользователей и запускается опытная эксплуатация. Ввод в эксплуатацию сопровождается приказами и актами, в которых определяется комиссия, выполняющая ввод в эксплуатацию, программа и сроки испытаний. В процессе опытной эксплуатации ведется протокол согласования, в котором фиксируется необходимость исправления ошибок и доработок. Заканчивается опытная эксплуатация приказом и актом о вводе в промышленную эксплуатацию, которые констатируют готовность системы и обязательства персонала использовать информационную систему при выполнении служебных обязанностей. Поддержка системы заключается в обеспечении ее нормального функционирования и соответствующих характеристик: надежность, безопасность, производительность, доступность и других, определенных в ТЗ.

В последующих главах приведенный список работ рассматривается более подробно с указанием применяемых моделей, схем и регламентов.

2.2. Принципы разработки информационных систем

Принципы разработки информационных систем, с одной стороны, являются результатом осмысления большого опыта проектирования, с другой – представляют неформальное дополнение стандартов. Следование принципам требует внутренней дисциплины всей команды и каждого разработчика в отдельности, но все затраты окупаются меньшим количеством переделок и качеством выполнения проектирования. Нарушение принципов приводит к проблемам создания и эксплуатации системы.

Периодически появляются новые списки принципов, принадлежащие различным коллективам и отражающие изменения в технологии проектирования. Исторически первыми в СССР появились принципы проектирования АСУ, предложенные академиком В.М. Глушковым [17]:

1. **Принцип системности** заключается в повсеместном применении системного подхода:

- установление основных направлений хозяйственной деятельности системы и ее основных функций;
- выявление структурных элементов и взаимосвязей между ними:
 - макроанализ: выделяются функции и связи элементов;
 - микроанализ: выделяется и изучается структуры элементов;
- формирование многоуровневой иерархии: система, подсистемы...

2. **Принцип непрерывного развития** предполагает постоянное изменение и наращивание возможностей ИС. Несомненно, что высокая адаптивность ИС остается актуальной и сейчас.

3. **Принцип совместимости** заключается в реализации информационного обмена между системами и подсистемами. В настоящее время он нашел свое воплощение в построении корпоративных ИС на основе совместного использования распределенных данных.

4. **Принцип стандартизации и унификации** предусматривает широкое использование стандартных решений и компонентов.

5. **Принцип эффективности** – создание АСУ должно быть экономически выгодным.

Кроме перечисленных общих принципов были предложены более детальные частные принципы:

1. **Принцип декомпозиции** – разделение системы на части для последующего анализа и реализации.

2. **Принцип первого руководителя** говорит о том, что в организационном плане возглавлять разработку должен первый руководитель. В современной трактовке принцип значительно шире – требуется привлечение и интерес к разработке ИС всех ключевых пользователей.

3. **Принцип новых задач** заключается в отказе от механической передачи функций обработки информации от человека компьютеру. Вместо этого должно быть найдено такое применение вычислительной техники, которое позволило бы решать качественно новые задачи (например, оптимизации загрузки оборудования).

4. **Принцип автоматизации информационных потоков и документооборота** предусматривает использование технических средств на всех этапах создания и обработки документов.

5. **Принцип автоматизации проектирования** предполагал использование информационных технологий для создания, ведения и доступа к проектной документации, автоматическое генерирование заготовок проектных документов, структур данных, программ по проектным описаниям. Сейчас эта идея реализована в CASE-средствах.

Наконец, в общий свод вошли организационно-технологические принципы:

1. **Принцип абстрагирования** заключается в выделении существенных для решения задачи аспектов системы и исключении несущественных.

2. **Принцип формализации** заключается в получении строго формальных описаний. Для этого были разработаны и успешно применяются несколько схем, систем записи, языков для отображения различных проектных решений.

3. **Принцип концептуальной общности** заключается в следовании единой методологии построения системы.

4. **Принцип непротиворечивости и полноты** заключается в построении всех необходимых для решения задач компонентов и в отсутствии противоречий между ними. Принцип должен соблюдаться при разработке всех моделей и описаний на каждом этапе проектирования.

5. Принцип независимости данных – данные должны проектироваться независимо от процессов их обработки в соответствии с информационными потребностями системы. Последовательное применение этого принципа привело к объединению всех экземпляров хранимых данных в единую базу данных и созданию СУБД как единого средства доступа к данным.

6. Принцип структурирования данных предполагает выделение единиц данных и связей между ними, описание хранимых структур и запросов обработки данных на единой методологической основе.

7. Принцип доступа конечного пользователя заключается в наличии средств оперативного доступа конечного пользователя к данным системы.

8. Принцип однократного ввода был очень важен для первых ИС, в которых в условиях отсутствия совместного использования данных одна и та же информация вводилась неоднократно в разных подсистемах, что приводило к созданию и накоплению противоречий. Ввод данных и поныне остается одной из самых сложных задач и источником различных ошибок, хотя в этом отношении накоплен большой опыт использования технических средств (магнитные карточки, штрих-коды) и организационных мероприятий (специальные методы проверки, совмещение ввода данных и формирования первичного документа).

Многие из принципов Глушкова не потеряли актуальности несмотря на то, что были сформулированы в 70-х годах, другие (как принцип доступа конечного пользователя) стали очевидной реальностью. Предлагаемый далее свод признаков отражает взгляд зарубежных специалистов на проектирование информационных систем [14]:

1. Принцип привлечения пользователей. Следование этому принципу снижает риски несоответствия системы требованиям бизнес-процесса. Привлекаемые пользователи должны быть компетентными, опытными, достаточно образованными в области компьютерных технологий, заинтересованными в результатах разработки. Пользователи должны привлекаться на вполне законных основаниях: по распоряжению или приказу руководства с точно определенными правами и обязанностями. Существуют специальные методы привлечения пользователей к процессу разработки, например, Joint Application Development. Пользователей следует привлекать не только на этапах постановки задач, но и в процессе конструирования и тестирования. Ознакомление пользователей с прототипом позволяет избегать ошибок в создании интерфейса и уменьшать количество проблем с обучением использования и внедрения системы.

2. Принцип инноваций. Этот принцип не менее важен, чем принцип привлечения пользователей. Если, привлекая пользователей, разработчики лучше учитывают их потребности, то следуя принципу инноваций, они качественно улучшают технологии работы пользователей на всех уровнях. Отказ от инноваций на основе информационных технологий приводит к тому, что проектируемая система ориентирована на поддержку устаревших технологий управления, и это еще более затрудняет внедрение новых. Если система решает только учетные задачи, то необходимость ввода данных в систему увеличивает трудоемкость учета: добавляет к проблемам документационного учета проблемы адекватности

компьютерного учета. Часто пользователи не могут предложить новые задачи, потому что слишком заняты решением текущих проблем и не представляют возможностей информационных технологий. Одним из методов преодоления этого является **проблемный подход**, который предусматривает следующие этапы решения:

- Идентификация проблем, возможностей или изменения условий функционирования системы (например, изменение законодательства или правил ведения бухгалтерского учета).
- Выявление причин, вызвавших проблему, и последствий решения проблемы.
- Разработка требований к решению.
- Выявление альтернативных решений.
- Выбор наилучшего решения.
- Реализация решения.
- Оценка результатов решения.

Другими инструментами инноваций управления являются известные методики повышения качества управления. Например, реинжиниринг бизнес-процессов [18] (Business Process Reengineering) направлен на радикальные изменения системы управления и всей организации бизнес-процессов. Широкое распространение для моделирования бизнеса получил процессный подход [15]. Процессные модели позволяют точнее определить место информационных технологий в организации бизнес-процессов. Новое качество управления может быть достигнуто за счет применения сбалансированной системы показателей [15] – инструмента вовлечения всех участников бизнес-процесса в достижение стратегических целей управления.

3. Принцип управления процессом разработки. Очевидно, что неуправляемый процесс дает непредсказуемые результаты и чаще плохие, чем хорошие. Процесс управления проектом требует описания всех работ, оценки ресурсов каждой работы и определения последовательности выполнения работ. Сложность управления проектированием информационных систем обусловлена многовариантностью программных решений, сложностью оценки ресурсов – особенно трудоемкости работ, противоречивыми интересами разных групп разработчиков. Все это делает управление проектированием ИС еще более актуальным. При бессистемном управлении никто не знает, как идет работа над проектом: программисты решают задачи в удобной для себя последовательности, руководитель не узнает результатов проектирования, пока программист не продемонстрирует ему работающую программу. Руководитель пытается разобраться с каждой проблемой – программисты подолгу ждут его руководящих решений и не могут разобраться с их логикой. Участники проекта простаивают в ожидании результатов своих коллег. Сроки затягиваются, программа не отвечает требованиям пользователей. Невозможно найти виновника несоответствий – программист реализовал пожелания первого попавшегося пользователя.

4. Принцип управления качеством процесса и продукта проектирования. В обработке данных особенно сильно проявляется влияние реализуемых

функций друг на друга. Например, считается, что исправление одной ошибки ведет к появлению трех новых, а в каждой программе есть хотя бы одна ошибка. Поэтому методики, обеспечивающие простоту, ясность и эффективность программ, являются просто необходимыми для сколько-нибудь объемных проектов. Со времен возникновения программирования известно понятие концептуальной целостности, которое требует выработки некоторых общих архитектурных правил, их документирования и неуклонного следования этим правилам. За время развития программирования как отрасли придумано и успешно внедрено достаточно много хороших методик и стандартов: структурные методики, объектно-ориентированные методики, парное программирование и многие другие. От проектировщиков требуется выбирать, устанавливать и следовать (это самое сложное) стандартам для разработки и документирования проектных решений. Отсутствие концептуальной целостности, нарушение стандартов обычно приводит к следующим ситуациям:

- Отсутствие или несоблюдение концепции приводит к появлению большого количества «заплат» в программах и огромным изменениям при изменении схем учета и правил вычисления показателей.
- Каждая форма имеет свои особенности интерфейса, которые пользователь должен знать.
- В системе есть «дыры» – которые приходится затыкать администратору или сопровождающему систему программисту.
- Фрагменты системы не стыкуются между собой из-за чего одна и та же работа выполняется неоднократно.
- Пользователи вынуждены выполнять не свойственные им функции.
- Разные стили и принципы построения модулей и документов приводят к проблемам преемственности разработчиков и специалистов сопровождения.

5. Принцип документирования. Документированию подлежат практически все значимые решения и результаты проектирования. Прежде всего следует разработать и утвердить сам стандарт документирования. Следует предусмотреть документирование взаимоотношений с заказчиком и не только юридическую сторону взаимодействий (договоры, акты, протоколы...), но и рабочую документацию (согласование интерфейса, пожелания пользователей...). Документация должна сопровождать взаимоотношения внутри группы разработчиков (протоколы совещаний, договоренности разработчиков, журналы испытаний...). В противном случае, невозможно будет определить причину несоответствия ИС ожиданиям и потребностям пользователей, одни и те же проблемы проектирования будут решаться неоднократно просто потому, что решения не фиксируются. Отсутствие общих описаний приведет к большому объему переделок. Уход разработчика из проекта будет иметь характер катастрофы – его преемнику придется начинать все заново.

6. Принцип структурирования. Принцип «разделяй и властвуй» известен с древнейших времен. В программировании он с успехом применяется для разделения сложных объектов на более простые, а большая сложность является отличительной чертой современных экономических ИС, охватывающих большую

часть процессов управления предприятием. Поэтому очень важно строить информационную систему по частям, тщательно планируя разбиение на контуры управления (подсистемы) и регламентируя их взаимодействие. Структурные методы лежат в основе автоматизации самого процесса проектирования. Важно применять структурные методы не только к программированию, но и ко всем процедурам проектирования: созданию информационных структур, организации эксплуатации ИС, документированию процесса и продукта разработки. Нарушение структурированности ведет к следующим проблемам:

- Одни и те же формы и функции проектируются неоднократно.
- Функции системы не систематизированы, никто не знает всех функций системы, описание функций разбросаны по разным документам. Появляются «дыры», которые приходится затыкать вводом дублирующей информации и другой ручной работой.
- Отсутствие обзорной документации делает изучение функций системы практически невозможным: понять последовательность работ можно только удерживая в голове описание сотен детальных функций.
- Недокументированные функции и исправления делают сопровождение чрезвычайно трудоемким.

7. Принцип эффективности требует рассматривать ИС как капитальное вложение. Это означает, что альтернативные решения должны оцениваться с помощью различных методик экономического анализа: затраты – прибыль, цена – качество, совокупная стоимость владения и других [19]. Однако, если затратная часть проектирования рассчитывается достаточно просто, то прибыль от внедрения информационных технологий оценить сложно, так как она может быть связана с изменением качества управленческих решений. Тем не менее сравнение с экономической точки зрения различных вариантов решения возможно и необходимо. Другое важное следствие этого принципа заключается в периодической оценке целесообразности продолжения проектирования, т.е. периодически рассматриваются вопросы переоценки стоимости проекта, его соответствия потребностям бизнеса и при обнаружении нецелесообразности проекта принимается болезненное решение о прекращении работ.

8. Принцип гибкости и адаптивности ориентирует на создание систем, способных расти и изменяться. Ясно, что «жесткие» системы придется заменять всякий раз при изменении условий эксплуатации или технологии управления. С другой стороны, большие адаптационные возможности, во-первых, сложны в настройке и требуют специальных знаний, во-вторых, увеличивают стоимость разработки. Поэтому особую значимость приобретают специальные методы проектирования (структурные и объектные методы, компонентный подход), позволяющие отделять данные от программ доступа и обработки, строить систему из независимых модулей, соблюдать соответствие программ и их описаний. Тем не менее, какие бы методы не применялись, гибкость и адаптивность всегда будет увеличивать стоимость разработки. Этот принцип требует разумного компромисса между степенью гибкости и адаптивности, с одной стороны, и затратами на достижение этой степени – с другой. Сама степень адаптивности может находиться в очень широких пределах: от минимального заранее определенного

набора параметров до специализированных инструментальных систем (например, таких как 1С Предприятие).

При всей разнице формулировок очевидно, что различные наборы принципов содержат по-разному сформулированные похожие рекомендации. Отличия обусловлены пристрастиями авторов и технологическими изменениями в области информационных технологий. Очевидна и некоторая противоречивость принципов. Так, адаптивность требует от программного обеспечения большей гибкости и функциональности, чем необходимо в момент разработки и это, несомненно, увеличивает стоимость работ и противоречит принципу эффективности. В целом, принципы являются концентрированным отражением культуры разработки информационных систем.

2.3. Общие методы разработки

На разных этапах разработки ИС приходится решать общие в методическом отношении задачи:

- Управление сложностью разработки – невозможно сразу создать (и описать) все детали информационной системы, приходится выделять и разрабатывать более или менее самостоятельные блоки.

- Задача определения границ является своеобразным продолжением структурных методов – при выполнении любой части проекта требуется тщательно определять область разработки (внешние границы) и уровень детализации (внутренние границы).

- Задача выработки и соблюдения требований к качеству и готового изделия, и процесса его разработки.

Известны и широко применяются следующие подходы к управлению сложностью:

1. Декомпозиция «сверху вниз». Исходное целое разбивается на некоторые части, каждая из которых в дальнейшем подвергается аналогичному делению до получения некоторых простейших единиц.

2. Синтез «снизу вверх» противоположен предыдущему подходу – из простых частей пытаются скомбинировать более сложные блоки, и так далее до получения требуемого результата.

3. Подход «от центра к границам» (Centre-out) комбинирует два первых подхода. Изучение начинается с некоторого среднего по уровню детализации объекта. Исследуется его структура (декомпозиция) и связь с другими объектами (синтез). Подход применяется при изучении различных структур управления.

Неправомерно противопоставление этих подходов – на самом деле они имеют различную область применения. Пошаговая декомпозиция применяется, когда объект и его свойства достаточно точно определены. Подход «снизу вверх» подходит для создания инструментальных средств, библиотек подпрограмм, предназначенных для решения широкого (и не вполне определенного) круга задач в некоторой предметной области.

Пошаговая декомпозиция является наиболее распространенной методологией решения задач и породила множество методов, которые называют структурными:

1. Структурное программирование – ограничение управляющих конструкций следованием, выбором и циклом (отказ от Go To) с возможностью их вложения друг в друга. Эти ограничения в сочетании с определением процедур и функций и общим принципом пошагового уточнения образуют стройную концепцию разработки программ.

2. Структурный анализ или процедурно-ориентированный подход. Выполнение задачи представляют в виде иерархической системы функций. На верхнем уровне иерархии находится функция, решающая задачу, которая делится на более элементарные. Разбиение прекращают при получении «неделимых» в некотором смысле функций. Для функций описываются входные и выходные данные, которые также могут быть структурированы.

3. Моделирование данных – описание системы данных в виде совокупности единиц информации. При этом выделяют элементарные единицы, такие как числа, строки, логические данные, и составные единицы, такие как массивы (наборы пронумерованных единиц одного типа) и структуры (наборы единиц разных типов).

4. Объектные модели. Объекты хранят данные и предоставляют методы для доступа и обработки хранимых данных. В отношении объектов можно рассматривать несколько иерархий (декомпозиций). Одну из них составляет иерархия классов, реализованная через механизм наследования – определения нового класса при помощи модификации существующего класса. Другая иерархия описывает вложенность объектов друг в друга. Третья определяет взаимодействие объектов в процессе выполнения программы.

Приведенный список отражает только некоторые распространенные структурные методы.

Определение границ, т.е. содержания и уровня детализации является важным с методологической точки зрения – слишком часто встречаются описания, в которых либо не указаны значимые для понимания моменты, либо указаны несущественные детали в большом количестве. При определении границ описания выделяют:

- внешние границы, определяющие описываемую часть реального мира,
- внутренние границы, задающие уровень детализации.

Границы описания рекомендуется устанавливать до начала проектирования и приводить во введении документов. Для сложных объектов, каковыми являются ИС, должно существовать несколько описаний с разными границами. Например, внешней границей описания ИС фирмы будут все виды деятельности и информационные источники, а внутренней – подсистемы и информационные блоки.

Требования к качеству должны быть составной частью процесса проектирования. Плохим вариантом является сначала выполнение работы, а затем проверка ее качества. В этом случае объем работ на переделку может существенно возрасти. Причины этого заключаются в запаздывании времени обнаружения

ошибок и несоответствия требуемому уровню качества (рис. 7). Так, наиболее значимые ошибки в определении требований обнаруживаются на самых последних этапах разработки, когда с продуктом начинают работать конечные пользователи.

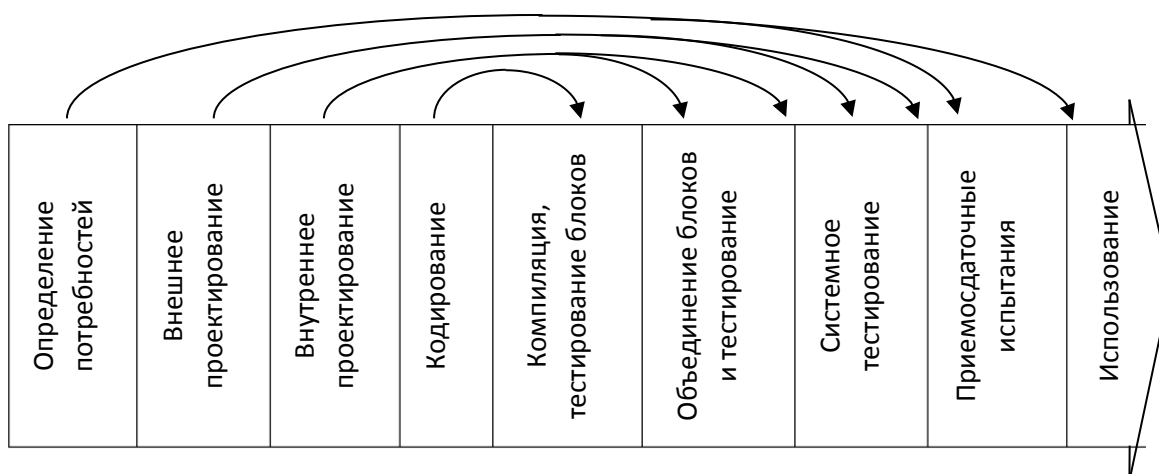


Рис. 7. Наиболее вероятные моменты обнаружения ошибок

Отдельно рассматривают качество конструкции и качество изготовления (табл. 3).

Таблица 3

Факторы, влияющие на качество информационной системы

	Соответствие ожиданиям/ требованиям пользова- теля	Гибкость, Расширяемость, возможности сопровож- дения	Универсальность
Качество кон- струк- ции	<ul style="list-style-type: none"> – Степень соответствия ожиданиям / требованиям пользователя – Изменение требований в процессе разработки 	<ul style="list-style-type: none"> – Независимость частей (высокая внутренняя связанность компонент каждой части, низкая степень зависимости частей друг от друга) – Нормализация структур данных – Структурированность процессов обработки данных 	<ul style="list-style-type: none"> – Применение стандартных технологий и систем записи – Понятность документации и конструкции в целом – Управление изменениями
Качество изготовле- ния	<ul style="list-style-type: none"> – Производительность – Доступность описания проектных решений – Количество ошибок – Количество ненормальных завершений 	<ul style="list-style-type: none"> – Модульность – Структурный код 	<ul style="list-style-type: none"> – Применение стандартных технологий и систем записи – Читаемость кода – Управление изменениями

Для каждого вида работ важны следующие характеристики качества:

1. Соответствие ожиданиям / требованиям пользователя. Качество соответствия обеспечивается:

- участием пользователя в планировании и конструировании ИС;
- применением технологий и систем записи, соответствующих подготовке пользователя;
- использованием специальных приемов и технологий общения.

2. Гибкость ИС. Гибкости добиваются посредством независимости систем и частей ИС.

3. Универсальность (portable). Способами достижения универсальности являются разработка более сложных процедур, чем требуется в данном проекте с учетом прогноза изменения ИС и тщательное документирование разработки.

В отношении моделей, используемых в процессе проектирования должны соблюдаться следующие нормы:

1. Модели должны соответствовать реальному миру. Для этого необходимо обеспечить следующее:

- понятность моделей для пользователей;
- реальное участие пользователей в разработке: все аспекты ИС, которые касаются пользователей обсуждаются и согласовываются с ними;
- построение моделей на сопоставительном анализе «как есть» и «как должно быть».

2. Каждая модель должна быть полной и непротиворечивой.

3. Набор моделей должен быть полным и модели не должны противоречить друг другу. Для этого модели должны формально следовать одна из другой, а подсистемы моделей должны соответствовать друг другу.

Неэффективные механические системы способны приносить убытки в пару центов на каждой производимой детали, однако из-за некачественных информационных процессов можно потерять целую компанию.

Алан Купер

3. АНАЛИЗ БИЗНЕС-ПРОЦЕССА. РАЗРАБОТКА ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

Любое проектирование начинается с постановки задачи. Это справедливо для любой модели жизненного цикла и способа организации работ. Различие в том, что гибкие методы разработки допускают неопределенности в постановках задач, которые исключаются в процессе проектирования.

Постановка задачи проектирования экономической информационной системы требует понимания того, как выполняется бизнес-процесс, и как он управляется. Кто и как составляет планы, как организован учет, что происходит при отклонениях от плана. Анализ выполняется для бизнес-процесса или системы связанных бизнес-процессов в рамках некоторого проекта. Работы, рассматриваемые в этой главе, соответствуют следующим стадиям разных моделей жизненного цикла:

- «Формирование требований к АС», «Разработка концепции АС» в жизненном цикле автоматизированной системы (ГОСТ 34.601-90);
- «Анализ» в спиральном жизненном цикле информационной системы по методике Microsoft Solution Framework.

Анализ бизнес процесса и разработка требований к ИС являются наиболее значимыми для успеха проектирования. Получаемое в результате представление о новой организации бизнес-процесса на основе информационных технологий определяет основное направление проектирования. Ошибки и недоработки на этих стадиях приводят к наибольшим издержкам.

Прежде всего, необходимо представить место анализируемого бизнес-процесса в системе процессов предприятия. Для этого приводится общее описание предприятия: виды деятельности (основные и обеспечивающие), принципы и технология управления, характеристика информационной системы предприятия в целом. В этой общей картине выделяется анализируемый бизнес-процесс, определяется его значение для предприятия в целом, устанавливаются правила его синхронизации с другими процессами.

Далее создают подробное описание выбранного для автоматизации бизнес-процесса: выделяют вход, ресурсы и выход процесса, описывают технологию преобразования входа и ресурсов процесса в выходные продукты или услуги, систему и технологию управления процессом.

Особое внимание следует уделить инновациям в системе управления бизнес-процессом на основе информационных технологий. Фиксация существующей технологии управления в виде соответствующей информационной системы

является самой распространенной ошибкой проектирования – информационная система из инструмента развития и совершенствования управления превращается в мощный тормоз для изменений в управлении.

Следует использовать все возможности поиска инноваций: выделение проблем и неиспользуемых возможностей, оптимизация исполнения бизнес-процесса, применение специальных методик анализа управления [15], таких как реинжиниринг бизнес-процессов, процессный подход, сбалансированная система показателей. Инновации могут быть получены улучшением существующей информационной системы, переходом на новые, более производительные и удобные технологии, развитием средств поддержки принятия решений и аналитических возможностей, применением интернет-технологий и методов искусственного интеллекта.

В результате анализа и поиска инноваций бизнес-процесса формируется представление «как должно быть», которое должно быть согласовано с пользователями и заказчиком, проанализировано с точки зрения осуществимости, затрат на реализацию, ожидаемых эффектов от внедрения и целесообразности проекта. Представление о будущей ИС излагается в техническом задании на разработку ИС, которое после утверждения становится основой для всех последующих проектных решений.

3.1. Изучение и анализ предприятия в целом

Начинается изучение с общего описания организации, в которое рекомендуется включить следующие пункты:

1. Миссия (место организации в общественном разделении труда), цели (набор значений показателей, таких как доля или сегмент рынка, которые организация хотела бы достигнуть), задачи (мероприятия, которые должны привести к достижению целей), методы достижения целей и решения задач (если целью является захват доли рынка, то какими средствами фирма стремится это сделать).

2. Характеристика организации:

- виды деятельности и их взаимосвязи и значимости;
- применяемое оборудование (здания, технологические линии, транспорт...);
- внешняя среда (поставщики, потребители, органы государственного регулирования) и взаимодействие с ней;
- технология планирования деятельности (периоды и показатели планирования);
- перспективы развития.

3. Описание системы бизнес-процессов включает основные процессы по производству продукции и / или оказанию услуг и вспомогательные, обеспечивающие нормальное функционирование основных. Кроме перечисления процессов необходимо выстроить логистические цепочки взаимодействия процессов, в которых выход одного процесса является входом для других. Наконец,

нужно выделить анализируемый бизнес-процесс в системе процессов предприятия, и более подробно остановиться на его взаимодействии с другими бизнес-процессами.

4. Описание организационно-функциональной структуры ограничивают основными подразделениями. Для подразделения указывают выполняемые виды деятельности, цели и задачи управления, взаимодействие с другими подразделениями. В этой части следует описать функционирование системы управления анализируемого бизнес-процесса, согласование плана выполнения этого бизнес-процесса с планом функционирования предприятия в целом.

5. Описание информационной инфраструктуры предприятия, которая обеспечивает управление предприятием. На крупных и средних предприятиях информационная инфраструктура является сложным объектом, который включает множество компонентов:

- технические средства обработки информации (серверы, персональные компьютеры, компьютерные коммуникации и сетевое оборудование);
- сетевая архитектура (сетевые протоколы, сегментация сети, операционные системы, система распределения полномочий и предоставления сервисов);
- применяемое программное обеспечение (специализированные программы для автоматизации различных видов деятельности);
- автоматизированные функции управления;
- набор информационных услуг и порядок их предоставления;
- систему хранения данных;
- географию узлов обработки данных.

Важно, чтобы описание давало полную картину о деятельности предприятия, но не было бы перегружено очевидными и излишними деталями.

3.2. Описание бизнес-процесса

3.2.1. Определение и компоненты бизнес-процесса

Процесс – устойчивая, целенаправленная совокупность взаимосвязанных видов деятельности, которая по определенной технологии преобразует входы в выходы, представляющие ценность для потребителя [15].

В соответствии с процессным подходом на предприятии выделяется и описывается система процессов. Есть сквозные (межфункциональные) процессы, получающие входы и ресурсы извне и поставляющие выходной результат работы внешним потребителям – клиентам. Сквозной (межфункциональный) процесс является бизнес-процессом, полностью или частично включающим деятельность подразделений с разной функциональной и административной подчиненностью. Сквозные процессы подразделяются на самостоятельные подпроцессы – процессы подразделений, которые, в свою очередь, делятся на операции (функции). На определенном уровне декомпозиции выявляется сеть бизнес-процессов – совокупность взаимосвязанных и взаимодействующих процессов, включающая все виды деятельности предприятия.

Например, небольшое предприятие по производству мягкой мебели на заказ выполняет один сквозной процесс – производство мягкой мебели, который включает следующие процессы подразделений: прием заказа и дизайн мебели, приобретение материалов и комплектующих, изготовление, доставка и сборка мебели. На более сложных предприятиях вместо линейной цепочки подпроцессов появляется сеть, в которой в рамках одного подразделения (например, лакокрасочного цеха) могут выполняться подпроцессы для разных технологических цепочек (например, производство мягкой и корпусной мебели).

Процессы можно классифицировать следующим образом:

1. Основные процессы – связанные с непосредственным созданием товаров и услуг, продаж и обслуживанием и добавляющие ценность продукта.
 - 1.1. Входная логистика – приобретение материалов и ресурсов производства.
 - 1.2. Производство – преобразование материалов в товары и услуги.
 - 1.3. Выходная логистика – предоставление товаров и услуг потребителям.
 - 1.4. Маркетинг и продажа товаров и услуг.
 - 1.5. Обслуживание.
2. Вспомогательные процессы – процессы, обеспечивающие нормальное выполнение основных процессов и добавляющие стоимость продукта.
 - 2.1. Инфраструктура фирмы – транспорт, склады, оборудование, коммуникации, информационные технологии.
 - 2.2. Управление персоналом.
 - 2.3. Конструкторская и технологическая подготовка производства.
3. Процессы управления бизнес-процессами.

Описание бизнес-процесса включает [9; 15] следующие компоненты:

Выход (продукт) процесса – это материальный или информационный объект или услуга, являющийся результатом выполнения процесса и потребляемый внешними по отношению к процессу клиентами. Для мебельного производства выходом является мебель.

Вход бизнес-процесса – это материалы, которые процессом преобразуются в выход. Для входа характерным является поступление материалов извне по отношению к рассматриваемому бизнес-процессу и планирование поступления материалов на определенный объем выхода. Для производства мебели входом могут быть детали крепежа и фурнитура, определенное количество мебельного поролона, ламинированной ДСП, кромочной ленты и других материалов. Для каждой единицы мебели можно установить нормы расхода всех компонентов входа.

Ресурс бизнес-процесса – это материальный или информационный объект, постоянно используемый для выполнения процесса и не являющийся его входом. В отличие от входа объем ресурса планируется на длительный период работы процесса. Примерами ресурса могут быть помещения и оборудование, необходимые для производства мебели. К ресурсам относятся и работники, занятые в производстве.

Преобразование входов в выходы при помощи ресурсов выполняется по определенной **технологии**, которая включает порядок и правила выполнения отдельных операций.

Процесс нуждается в **управлении**, которое основано на планировании и измерении различных показателей: объем производства, объемы потребления материалов и ресурсов, денежные доходы и расходы, временные характеристики производства, показатели качества выходной продукции, показатели удовлетворенности клиентов и многие другие. Планирование включает определение плановых показателей и построение календарных графиков работы. Выполнение бизнес-процессов сопровождается учетом использования материалов, ресурсов, изготовления создаваемой продукции. По данным учета вычисляются фактические значения показателей, которые постоянно сравниваются с плановыми. Отклонение фактических показателей от запланированных значений запускает процесс корректировки планов и выработки управляющих воздействий.

Владелец процесса – это должностное лицо (орган управления), имеющее в распоряжении ресурсы, необходимые для выполнения процесса, и отвечающее за организацию и результаты процесса.

Организация процесса должна удовлетворять следующим требованиям:

– Требования к владельцу процесса:

- должен существовать один владелец процесса;
- полномочия и ответственность владельца процесса должны быть четко определены;

• не должно быть пересечений полномочий и ответственности с другими руководителями организации.

– Требования к границам процесса:

• границы процесса должны быть четко определены (по функциям и ответственности руководителей) и зафиксированы документально;

• границы функциональных подразделений процесса должны быть четко определены.

– Требования к регламентирующим документам:

- должно существовать действующее описание процесса в целом;
- должны существовать действующие положения о подразделениях;
- должны существовать действующие должностные инструкции;
- должны существовать действующие методики (внутренние стандарты);
- должна функционировать система актуализации документации.

– Процесс должен соответствовать существующим законодательным актам и нормативным документам, регламентирующим выполнение процесса.

– Требования к выходам процесса:

• выходы процесса должны быть четко определены;

• пользователи каждого выхода процесса должны быть четко определены, потребности пользователей оформлены документально;

• должны существовать спецификации, содержащие требования на каждый выход процесса;

• каждый выход должен быть закреплен за ответственным исполнителем;

• должна функционировать система контроля качества выходов процесса.

– Требования к входам процесса:

- входы должны быть четко определены;
- поставщики каждого входа процесса должны быть четко определены, требования к поставщикам – специфицированы;
- должны существовать спецификации, содержащие требования на каждый вход процесса;
- каждый вход должен быть закреплён за ответственным исполнителем;
- должна существовать система входного контроля качества.
- Требования к ресурсам:
 - ресурсы должны быть четко определены;
 - должны существовать спецификации, содержащие требования к каждому ресурсу;
 - каждый ресурс должен быть закреплён за ответственным исполнителем (материально ответственным лицом).
- Требования к показателям процесса:
 - показатели эффективности процесса должны быть определены и постоянно использоваться на практике;
 - должны быть определены и использоваться показатели продукта процесса;
 - должна существовать система сбора и использования данных удовлетворенности клиентов процесса.

3.2.2. Модель IDEF0

Модель IDEF0 разработана задолго до появления процессного подхода. Протообразом модели IDEF0 является система SADT (Structured Analysis and Design Technique), предложенная Дугласом Россом в 70-х годах. Позже вооруженные силы США адаптировали SADT в IDEF0 [9]. Можно утверждать, что именно эта модель легла в основу процессного подхода.

Модель представляет собой описание функций системы. Основным компонентом описания является работа (поименованный процесс, функция или задача, которые имеют временные рамки и результаты). Работы изображаются на диаграммах прямоугольником (рис. 8). Взаимодействие работы с внешним миром и другими работами описывается стрелками: входом (материалы, информация), управление (правила, стратегии, графики, регламенты, нормативы), выход (материалы, услуги, документы), механизмы – ресурсы (персонал, станки, оборудование, энергия...), вызов (аутсорсинг, выполнение работы сторонними организациями).

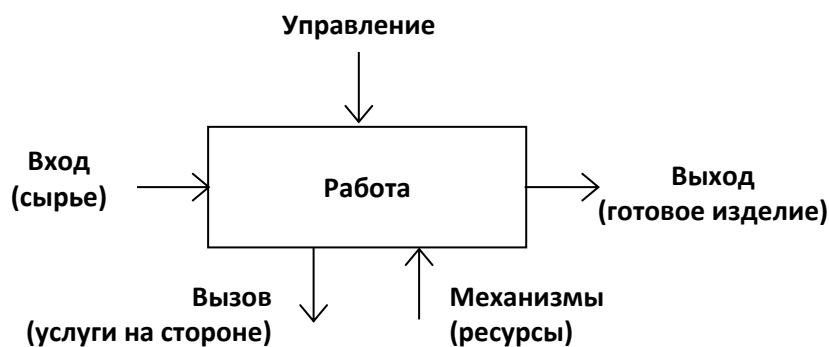


Рис. 8. Графическое изображение работы на диаграммах IDEF0

Модель строят в виде иерархической системы диаграмм. На верхнем уровне находится контекстная диаграмма, соответствующая всему моделируемому контуру управления. Стрелки на контекстной диаграмме отображают взаимодействие процесса с внешней средой. Диаграммы следующего уровня описывают декомпозицию некоторой работы. Эта модель соответствует структурным методам пошаговой детализации.

Диаграммы IDEF0 позволяют наглядно представить структуру и последовательность выполнения и связи операций бизнес-процесса. Многие детали выполнения и особенности взаимодействия не отображаются диаграммами IDEF0. Для детального описания операций используют модели IDEF3 [9], ARIS [19]. Возможно построение нескольких моделей: модель AS-IS (как есть) и модель TO_BE (как должно быть) для сравнения и выделения объема проектных работ.

В качестве примера рассмотрим бизнес-процесс оказания услуг кинотеатром. На рис. 9 представлена контекстная диаграмма оказания услуг демонстрации фильмов. Предполагается, что кинотеатр использует независимую от него организацию, оказывающую услуги по распространению билетов. Контекстная диаграмма разбивается (рис. 10) на детальные операции: выбор сеанса и мест, продажа билетов и, собственно, демонстрация фильма. По итогам продаж выполняется анализ оказания услуг. Планирование репертуара выполняется на основе этого анализа и ряда других сведений, в частности маркетинговых исследований кинорынка.

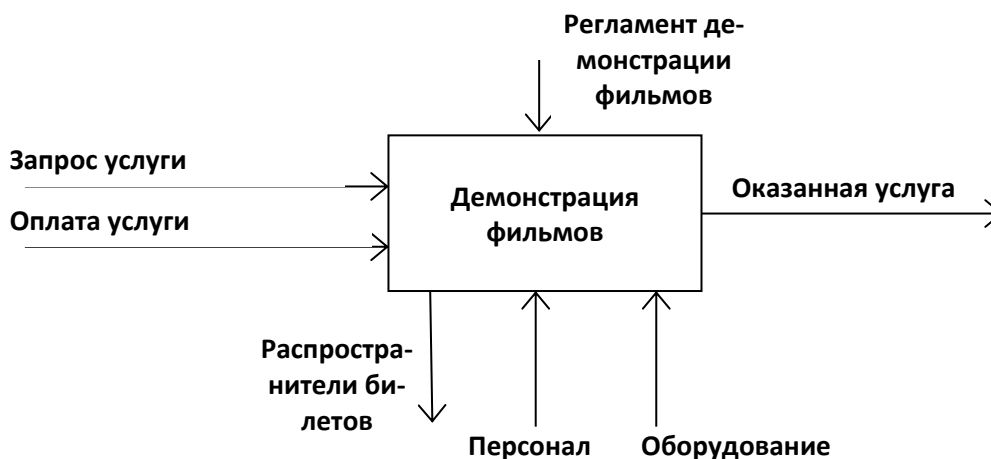


Рис. 9. Оказание услуг кинотеатром



Рис. 10. Детализация оказания услуг кинотеатром

Детализация может быть продолжена до уровня операций, которые одновременно выполняются на некотором рабочем месте. Диаграмма детализации должна соответствовать по входам, выходам, управлению и ресурсам родительской диаграмме.

3.2.3. Управление бизнес-процессом

Управление процессом включает различные виды деятельности, которые можно представить в виде цикла Деминга [15] P – D – C – A:

- Планирование (Plan) – определение сроков исполнения бизнес-процесса, требований к исполнению, порядок и нормы потребления ресурсов.
- Выполнение процесса (Do) – организация исполнения процесса.
- Контроль (Check) – измерение показателей выполнения процесса.
- Управление (Act) – изменение графиков, ресурсов, требований.

Описание технологии управления должно раскрыть последовательность и характер операций на каждом этапе управления. Управление рассматривается как непрерывный процесс. Однако, это в большей степени характерно для оперативного управления, стратегическое и тактическое управление имеют определенные периоды планирования и выполнения. Планирование начинается с определения целей. Целью не может быть просто организация исполнения бизнес-процесса. Цель должна заключаться в достижении некоторых характеристик бизнес-процесса, измеряемых через систему показателей.

Цели управления должны быть [15]

- *достижимыми* – нельзя планировать нереальные цели;

– *измеримыми* – иначе невозможно определить, приближаемся мы к целям или удаляемся от них. «Вы не можете управлять тем, что невозможно измерить... Все, что измеримо, может быть достигнуто» (Билл Хьюлетт);

– *конкретными* – в полном соответствии с предыдущим утверждением нельзя планировать «Борьбу за непрерывное улучшение качества услуг»;

– *гибкими* – чрезмерная конкретность целей может привести организацию в тупик. «Достигнуть рентабельности продукта на 20 %». За год могут подняться цены на сырье, упасть стоимость продукта N на рынке, появиться новый демпинговый конкурент, цели должны направлять деятельность организации, но не загонять ее в жесткие рамки;

– *совместимыми* – цели не должны противоречить стратегии и миссии организации.

Цели управления могут быть противоречивыми. Например, повышение качества требует увеличения затрат, что противоречит цели снижения издержек. Возможна декомпозиция целей, или влияние достижения одной на несколько других.

Известный стандарт – МС ИСО 9000:2000 – содержит принципы управления бизнес-процессами:

1. *Фокус на потребителя* – учет и стремление удовлетворить потребности потребителя.

2. *Лидерство руководства* – не только распределение полномочий и поручений, но и личное участие в решении задач.

3. *Вовлечение персонала* – мотивация персонала на активное участие в решении задач управления, а не только в выполнении порученных функций.

4. *Процессный подход* – применение процессной методологии для понимания и управления процессом.

5. *Системный подход* – выявление взаимосвязанных процессов и управление ими как единой системой.

6. *Постоянное улучшение* – улучшение качества бизнес-процесса является постоянной целью управления.

7. *Принятие решений на основе фактов* – создание системы точного и надежного измерения показателей бизнес-процесса и учет результатов измерений при принятии решений.

8. *Взаимовыгодные отношения с поставщиками* – учет выгод поставщиков в управлении бизнес-процессом.

Описание технологии управления должно включать периоды планирования, содержание планов, показатели, характеризующие выполнение бизнес-процесса и достижение целей. В описание контроля нужно включить процедуру сбора данных о фактическом выполнении бизнес-процесса. По собранным данным определяются фактические значения показателей. Сопоставление этих значений с плановыми, выявление выбросов, феноменов, выделение закономерностей помогает принимать решения по корректировке планов и проведению мероприятий для достижения целей управления.

Ключевую роль в управлении играют показатели. Все показатели можно разделить на три категории:

– *показатели процесса* могут быть определены как числовые величины, характеризующие течение самого процесса и затраты на него (временные, финансовые, ресурсные, человеческие и др.). Показатели процесса могут быть абсолютными и относительными (приведенными к объему услуг, сезонным колебаниям, тарифным изменениям и другим внешним факторам, не зависящим от управления проверяемым процессом);

– *показатели продукта (услуги)* – числовые величины, характеризующие продукт (услугу) как результат выполнения процесса (абсолютный объем услуг, объем услуг относительно заказанного или необходимого, количество ошибок и сбоев при оказании услуги, номенклатура оказанных услуг, номенклатура оказанных услуг относительно необходимой и т.д.);

– *показатели удовлетворенности клиентов процесса* – числовые величины, характеризующие степень удовлетворенности потребителя результатами процесса (выходом, услугой и т.д.). При этом следует различать удовлетворенность потребителя (внутреннего и внешнего) выходом процесса и удовлетворенность конечного потребителя полученной продукцией или услугой.

Абсолютные показатели используют для измерения характеристик бизнес-процесса (стоимость, объем услуг...), относительные рассчитываются на основе абсолютных показателей процесса путем формирования различных отношений между ними, применяются для сравнения однотипных показателей (доля (процент) увеличения объема продаж).

Абсолютные показатели делят на:

– *технические*, которые характеризуют объемы выполненных работ, произведенных товаров или выполненных услуг, технологию выполнения процесса, используемое оборудование, программное обеспечение, среду и т.д. Технические показатели отличаются по отраслям;

– *временные*, характеризующие временные свойства, такие как суммарное и среднее время выполнения процесса в целом, время простоев, время выполнения отдельных функций процесса и другие;

– *стоимостные*, измеряющие денежные характеристики, доход или увеличение стоимости бизнес-процесса, расходы на материалы, оплату труда и другие;

– *показатели качества*, измеряющие качество процесса и продукта.

Приведем примерный список возможных показателей.

Показатели процесса:

1. Показатели результативности выполнения процесса.

1.1. Выход процесса в натуральных единицах (количество услуг, объемы произведенной продукции...).

1.2. Выход процесса в денежных единицах.

2. Показатели времени выполнения процесса.

2.1. Среднее время выполнения процесса в целом и отдельных его подпроцессов.

2.2. Среднее время простоев.

2.3. Среднее время выполнения отдельных функций процесса.

3. Технические показатели.

- 3.1. Загруженность производственных мощностей.
 - 3.2. Количество функций процесса, выполняемых на рабочих местах.
 - 3.3. Численность персонала процесса, в том числе руководителей и специалистов.
 - 3.4. Количество транзакций за период.
 - 3.5. Количество автоматизированных рабочих мест.
 - 4. Показатели стоимости.
 - 4.1. Стоимость процесса в целом.
 - 4.2. Стоимость сырья и материалов.
 - 4.3. Затраты на оплату труда исполнителей.
 - 4.4. Амортизация оборудования и нематериальных активов.
 - 4.5. Затраты на тепло- и энергоносители.
 - 4.6. Затраты на связь.
 - 4.7. Затраты на получение информации.
 - 4.8. Затраты на повышение квалификации исполнителей.
 - 4.9. Дебиторская и кредиторская задолженности, объемы «замороженного» капитала в виде складских запасов сырья или неликвидов).
 - 4.10. Амортизация оборудования.
 - 5. Показатели качества.
 - 5.1. Степень дефектности продукции процесса.
 - 5.2. Количество возвратов и рекламаций на продукцию процесса.
 - 5.3. Количество жалоб и рекламаций на качество обслуживания, поступивших от клиентов.
 - 5.4. Количество некомплектных (не соответствующих спецификациям) отгрузок.
 - 5.5. Сохранность готовой продукции.
 - 5.6. Количество внештатных ситуаций, потребовавших оперативного вмешательства руководства верхнего уровня.
 - 5.7. Способность процесса быстро адаптироваться к изменяющимся требованиям заказчика.
 - 5.8. Способность процесса сохранять свои параметры при изменении внешних условий (стабильность процесса, минимальные вариации).
 - 5.9. Независимость процесса от изменений в части персонала.
 - 5.10. Управляемость процесса.
 - 5.11. Способность процесса к улучшениям.
- Показатели 5.1–5.6 достаточно просто измерить, разработав методики сбора и обработки соответствующей информации. Показатели 5.7–5.11 интуитивно понятны, однако на практике их измерить затруднительно.
- 6. Относительные показатели.
 - 6.1. Показатели «план / факт».
 - 6.2. Сравнение с другим процессом.
 - 6.3. Рентабельность процесса = прибыль по процессу / стоимость процесса.
 - 6.4. Рентабельность оборотных активов процесса = прибыль по процессу / объем используемых оборотных активов.

6.5. Выработка на одного сотрудника = объем продукции процесса / численность сотрудников.

6.6. Фондоотдача процесса = объем продукции / величина основных фондов.

6.7. Оборачиваемость оборотных активов процесса = величина выручки / средние остатки оборотных активов процесса.

6.8. Эффективность использования ресурсов на единицу продукции: коэффициенты использования оборудования, ресурсов, сырья, материалов, времени на проведение единицы работ или услуг).

6.9. Доля накладных расходов = величина накладных расходов / стоимость процесса.

6.10. Степень загрузки персонала = общее время работы по выполнению функций процесса / общее рабочее время всех сотрудников.

6.11. Степень автоматизации = количество автоматизированных функций процесса / общее количество функций процесса.

6.12. Величина офисной площади на одного сотрудника.

6.13. Количество персональных компьютеров на одного работающего.

7. Затраты на брак.

7.1. Предупреждение на этапах разработки и производства; проведение альтернативных расчетов, проектов и испытаний; отбраковка в процессе производства.

7.2. Исправление, доработка или переделка продукции (услуги), забракованной по результатам контроля.

7.3. Замена и возврат брака, компенсация потребителю понесенного ущерба, а также транспортных и таможенных расходов.

Показатели продукта:

1. Функциональные показатели.

1.1. Количество функций продукта; полнота функций продукта; диапазон параметров.

1.2. Сравнение с мировыми образцами продукта (услуги).

2. Характеристики надежности.

2.1. Гарантийный срок; наработка на отказ.

2.2. Гарантированное количество ошибок (дефектов при массовой поставке, AQL – acceptable quality level – приемлемый уровень качества).

3. Показатели безопасности.

3.1. Экологические показатели.

3.2. Наличие сертификатов соответствия для обязательной сертификации товаров и услуг; фактические показатели безопасности товаров и услуг.

4. Наличие дополнительных услуг.

4.1. «Пакетные» закупки или сервис.

4.2. Скидки для постоянных или оптовых клиентов; премии для клиентов; послепродажный сервис.

4.3. Другие способы поддержки клиента (юридическая, техническая, аудиторская, инвестиционная, кредитная и др.).

Показатели удовлетворенности клиента. Показатели удовлетворенности можно получить двумя способами:

1. Анкетирование представителей клиента, потребителя или заказчика по специально созданным анкетам.

2. Анализ взаимоотношений с клиентами, потребителями и поставщиками на основании собственных данных (косвенная оценка удовлетворенности).

Косвенно степень удовлетворенности клиента можно оценить по следующим показателям:

- рост объема продаж продукции (услуг);
- длительность деловых связей;
- число (процент) постоянных клиентов;
- количество претензий, рекламаций, жалоб, замечаний;
- число (процент) клиентов, «потерянных» за период времени;
- доля рынка по продукту (услуге);
- число «имиджевых клиентов».

Приведенный список показателей не претендует на полноту. Для конкретного бизнес-процесса может быть предложен набор, который не содержит некоторых из перечисленных выше показателей и дополняет его другими показателями. Важно, чтобы показатели отражали все существенные характеристики бизнес-процесса. В отношении каждого показателя должны быть определены данные, по которым он рассчитывается и правила вычислений.

Для кинотеатра в показатели процесса можно включить:

- показатели процесса:
 - количество проданных билетов;
 - стоимость проданных билетов;
 - суммарное время демонстрации фильмов;
 - суммарное время технологических перерывов;
 - заполняемость мест;
 - количество сбоев в демонстрации;
 - количество возвратов билетов по причинам неудовлетворительного качества демонстрации;
- показатели продукта:
 - характеристики фильмов;
 - технические характеристики зала;
 - характеристики аппаратуры;
 - акустические характеристики;
- показатели удовлетворенности клиентов:
 - количество претензий, рекламаций, жалоб, замечаний;
 - количество отзывов в социальных сетях.

Кроме содержания и правил вычисления показателей нужно определить измерения, для которых будет анализироваться изменение показателей. Рассматривая изменения дохода от фильма к фильму можно выделить наиболее доходные фильмы и характеристики, влияющие на доход. Измерения состоят из меток и, как правило, имеют сложную структуру. Для каждого измерения можно ввести несколько классификаций. Например, для дат можно ввести классификации год-квартал-месяц-

день, год-сезон. Кроме этого, с метками измерений можно связать различные свойства. Каждый день может быть рабочим, или выходным, или праздничным. Наличие свойств позволяет исследовать их влияние на показатели бизнес-процесса.

Для кинотеатра можно предложить следующие измерения:

- дни с иерархиями: день-месяц-год, день-день недели;
- фильмы со свойствами: жанры, производители, режиссёры, актеры, характеристики показа (2D, 3D), возрастные категории, рейтинг...;
- сеансы с различными свойствами (утро, день, вечер);
- зрители со свойствами: пол, возрастная категория, социальное положение;
- персонал.

Для отзывов можно дополнительно добавить измерение эмоциональной окраски отзыва или явную оценку в какой-либо шкале. Система показателей и измерений в информационной системе будет основой для проектирования отчетов, многомерного анализа, исследования зависимостей и других аналитических процедур.

3.2.4. Информационная составляющая бизнес-процесса

Основные события в рамках бизнес процесса оформляются соответствующими документами. Такое оформление позволяет юридически зафиксировать ответственность и результаты бизнес-процесса, а также образует документационную основу учета. Моменты регистрации событий в виде документов представлены на рис. 11. Большая их часть сосредоточена на границах бизнес-процесса и предназначена для учета поступления входных материалов и ресурсов, а также фиксации результатов бизнес-процесса (выхода). Кроме этого, важным является сбор данных для измерения всех показателей. Отдельную и сложную задачу составляет построение системы сбора данных для определения удовлетворенности потребителя.

Описание циркуляции информации, возникающей при выполнении и управлении бизнес-процессом особенно важно для проектирования ИС, так как позволяет понять участие каждого будущего пользователя в функционировании проектируемой ИС, выделить наиболее трудоемкие операции и узкие места в обработке данных, предложить наилучший способ применения информационных технологий для управления бизнес-процессом.

Для описания циркуляции документов собирают образцы исходных, выходных, отчетных документов и описывают порядок их обработки и использования, строят схемы документооборота и описывают его регламент. Строится информационная модель прикладной области, описывается география рабочих мест и действия, выполняемые на каждом рабочем месте.

Для детального описания циркуляции документов используют схемы документооборота (рис. 12). Система обозначений для них предложена фирмой IBM и приведена в табл. 4. Символы должны иметь стандартные пропорции (3 × 2) и одинаковые в рамках диаграммы размеры. Для удобства ссылок на компоненты схемы их можно снабжать номерами или другими обозначениями. Направления передачи документов указываются стрелками, кроме направлений сверху вниз и справа налево, которые считаются стандартными. Диаграммы потоков данных

разделяют на зоны (полосы или столбцы), указывая в одной зоне операции, выполняемые в одном подразделении.

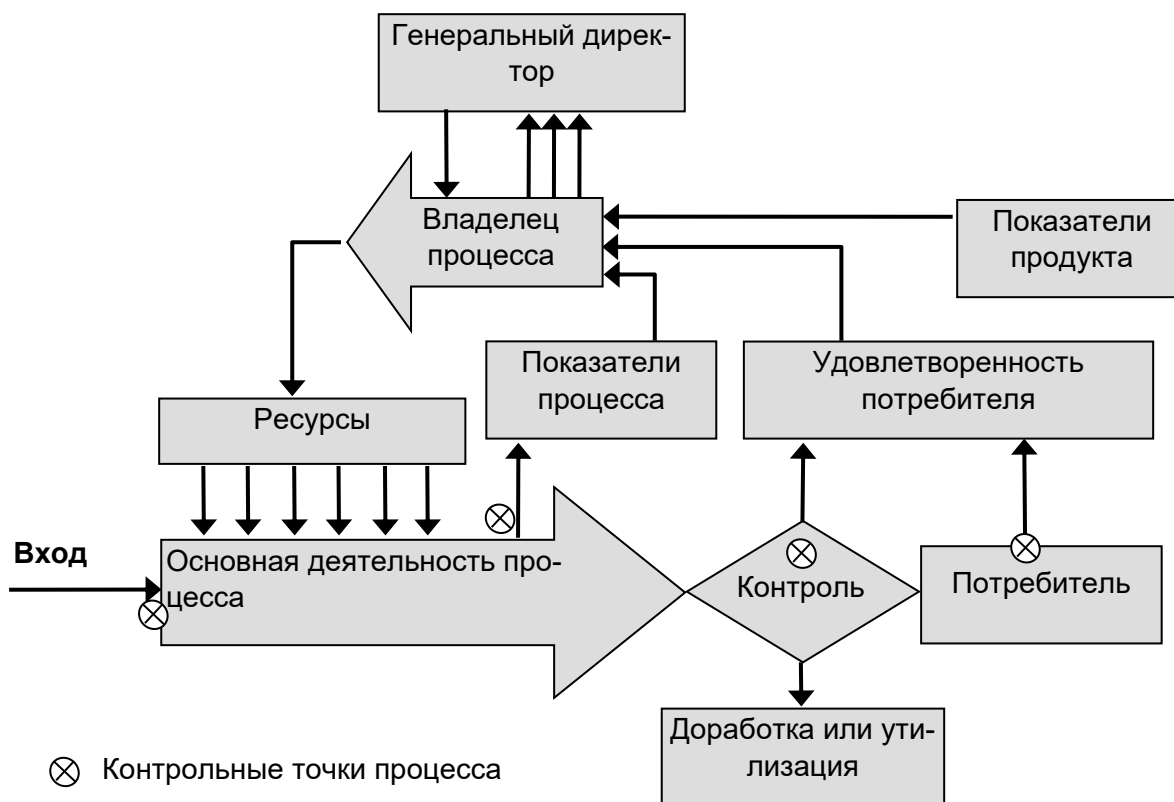


Рис. 11. Регистрация выполнения бизнес-процесса

Основное отличие схемы документооборота от блок-схем является то, что линии обозначают передачу информации, а не управления. Для каждой операции на схеме должны быть указаны входные и выходные документы. Составление должно следовать этой логике обозначений. Например, недопустимо соединение двух операций линией транспортировки носителей без указания носителя, который передается от операции к операции. Появление документов, которые никем не используются (которые не имеют выходящих дуг), также не допустимо. Символ источника (получателя) информации используется для обозначения внешних, по отношению к рассматриваемой предметной области, объектов, детализация документооборота внутри которых не нужна.

Схемы информационных потоков и документооборота дополняют пояснениями, образцами форм документов, правилами заполнения и обработки документов, временным регламентом обработки документов (сроками поступления документов и временем выполнения операций). По описанию документооборота и образцам исходных и отчетных документов можно построить информационную модель бизнес-процесса – описание содержания структуры и источников данных, а также определить сотрудников, ответственных за ввод и использование данных. Отчетные формы помогают выделить показатели и измерения, по которым выполняется анализ изменения показателей.

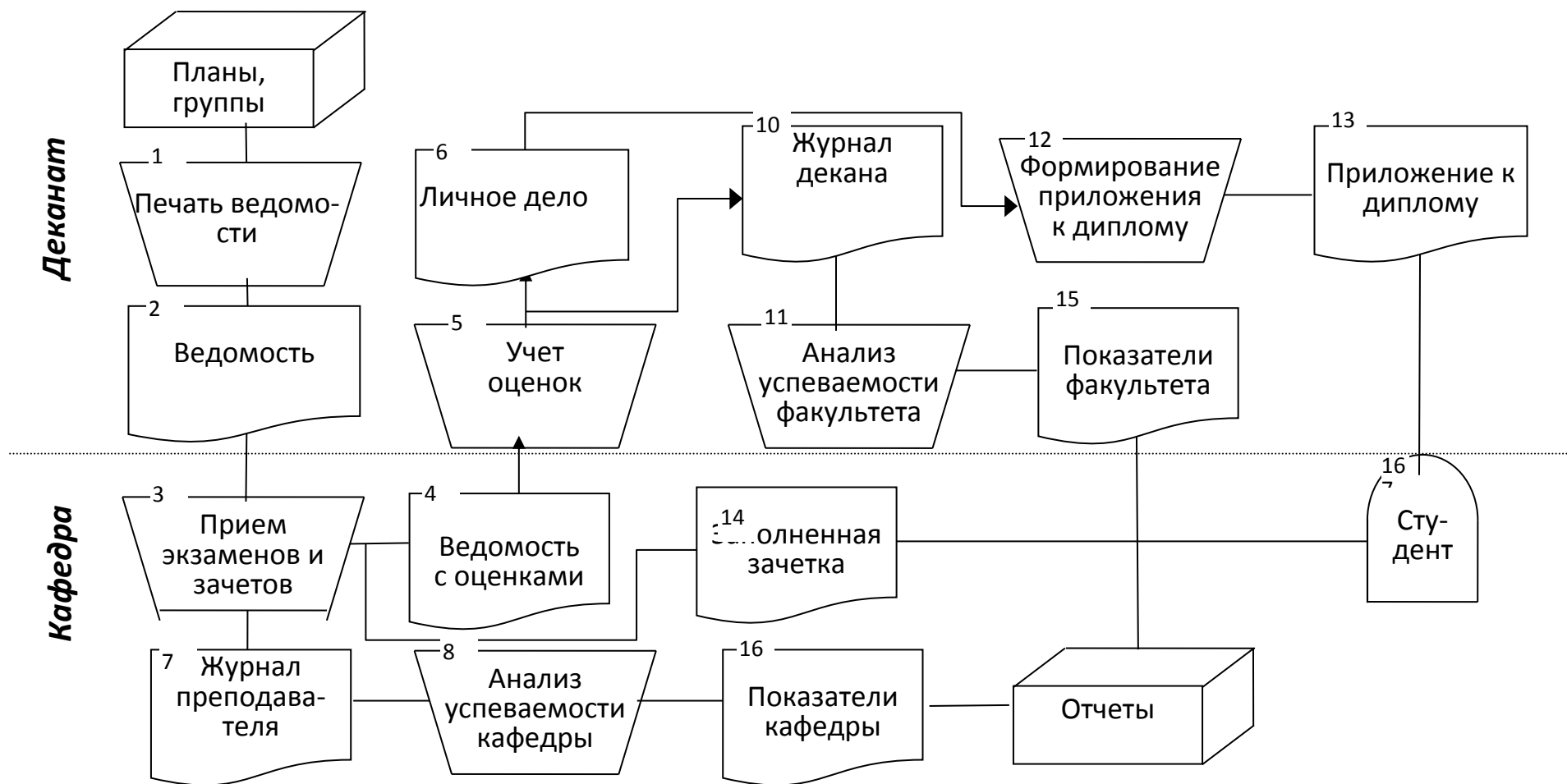
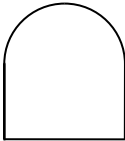

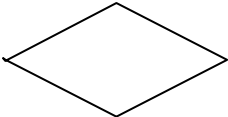
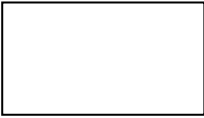
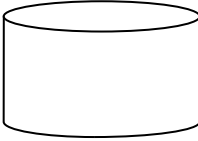



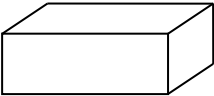
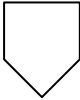
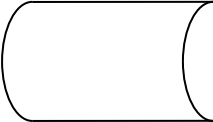
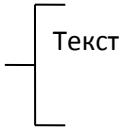



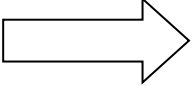


Рис. 12. Документооборот учета успеваемости

Таблица 4

Символы схем описания документооборота

Символ	Пояснения	Символ	Пояснения
	источник-получатель информации		канал связи
	развилка		операция, выполняемая на ЭВМ
	магнитный диск		пуск – останов
	печатный документ		внутристраничный соединитель
	архив		межстраничный соединитель
	хранимые данные		комментарий
	ручная операция		ВВОД-ВЫВОД
	ввод с клавиатуры		материальный поток

3.3. Инновации управления на основе информационных технологий

Инновации в управлении должны стать основной целью разработки или модернизации информационной системы. Очевидно, что инновационный компонент информационной системы создается на основе компонента учета, который должен обеспечивать сбор данных для принятия решений на основе точной и актуальной информации.

Достижению этой цели препятствуют отсутствие представления о возможностях ИТ у менеджеров, и о проблемах управления у проектировщиков. Поэтому довольно часто постановка задачи на создание ИС включает, в основном, учетные задачи, дополненные построением стандартной отчетности. В данном разделе приводятся несколько подходов к выделению инноваций в управлении.

В главе 1 уже перечислялись основные положения проблемного подхода, нацеленного на выявление проблем, возможностей или изменения условий управления.

На основании анализа бизнес-процесса предлагаются цели и ограничения новой системы. Цели формулируются как показатели успеха в решении проблем и использовании возможностей. Ограничения устанавливают границы (временные, стоимостные, технологические, политические), в которых ищется решение.

Эта достаточно простая методика в последнее время все чаще заменяется более глубокими преобразованиями, получившими название реинжиниринг бизнес – процессов [18].

Предшественником технологии реинжиниринга является «Трансформация и непрерывное усовершенствование бизнес-процессов» Эдвардса Деминга. Ее основными декларациями являются следующие:

1. Цель – повышение качества услуг, а не прибыль.
2. Трансформация организации работ, исходя из интересов потребителя.
3. Исследуются и устраняются недостатки производства, а не работника.
4. Внедряются командные (артельные) методы работы, в результате возрастает роль и ответственность каждого работника.

Появление реинжиниринга бизнес-процессов (Майкл Хаммер, 1990) обусловлено качественными изменениями на рынке. Стремление получить конкурентные преимущества за счет более полного удовлетворения пользователя заставляет переходить от массового производства к индивидуальному даже в традиционно массовых отраслях, таких как производство мебели и автомобилей. При этом существенно возрастает нагрузка на управленческий аппарат, и автоматизация управления становится необходимой частью реинжиниринга.

Реинжиниринг начинается с переосмысления положения фирмы на рынке. Однако с точки зрения построения ИС наиболее существенную роль играют изменения в технологии управления. Для этого применяются следующие преобразования:

1. Горизонтальное сжатие: несколько последовательно выполняемых рабочих процедур объединяют в одну.
2. Вертикальное сжатие: исполнители принимают самостоятельные решения по большому множеству задач, не обращаясь к вышестоящим руководителям.

3. Отказ от линейного упорядочения рабочих процедур в пользу их параллельного исполнения.
4. Процессы имеют разные варианты исполнения в зависимости от ситуации.
5. Работа выполняется в том месте, где целесообразно.
6. Уменьшается количество проверок и управляющих воздействий.
7. Минимизируется количество согласований.
8. Вводится «Уполномоченный менеджер» для принятия решений в рамках некоторого проекта.

В отличие от старых иерархических строго централизованных структур управления применяется централизованно-децентрализованный подход: происходит централизация хранения данных для их децентрализованного использования (в отличие от жестких иерархических структур, в которых «правая рука не знает, что делает левая»). Из приведенного выше описания становится понятно, что автоматизированная обработка данных становится необходимым компонентом реинжиниринга бизнес-процессов.

Информационная система объединяет большое количество пользователей. Значительная часть функций связана с учетными задачами – вводом данных. Инновации по вводу данных в значительной степени связаны со снижением трудоемкости ввода и уменьшением количества ошибок. Для достижения этих целей применяют нормативно-справочные данные, заменяют ввод с клавиатуры выбором из справочников, применяют автоматические проверки вводимых данных. Более перспективным является применение автоматической регистрации на основе штрих-кодирования, QR-кодов¹, бесконтактных смарт-карт, RFID-меток² и других подобных технологий. Дистанционные коммуникационные технологии также существенно снижают трудоемкость учета – документ формируется, передается и поступает в систему учета сразу в электронной форме. Это исключает и повторный ввод, и связанные с ним ошибки. Применение мобильных устройств позволяет решать задачи учета и управления без привязки к рабочему месту.

При всей важности задач учета наибольший эффект от внедрения информационной системы следует ожидать от повышения качества управления. Поэтому автоматизация поддержки принятия решений является основой для инноваций.

Инструменты многомерного OLAP анализа являются первой ступенькой автоматизации управления. Внедрение такого анализа может быть быстро и с минимальными затратами реализовано с использованием сводных таблиц MS Excel, Power BI³, облачных технологий Yandex DataLens⁴ или других аналогичных технологий. Для этого строятся запросы для вычисления (выбора) значений показателей для каждой комбинации меток измерений и определяются сводные таблицы для анализа зависимостей показателей от измерений.

¹ URL: <https://ru.wikipedia.org/wiki/QR-%D0%BA%D0%BE%D0%B4>.

² URL: <https://ru.wikipedia.org/wiki/Rfid>.

³ URL: <https://powerbi.microsoft.com/ru-ru/what-is-power-bi>.

⁴ URL: <https://cloud.yandex.ru/services/datalens>.

Многомерный анализ позволяет выявить тенденции и выдвинуть предположения о наличии и характере зависимостей. Для математического описания зависимостей применяют различные модели Data Mining:

- Регрессионные модели устанавливают формулы зависимости средних значений показателей от других переменных.
- Модели классификации позволяют отнести объект к определенному классу.
- Модели кластеризации группируют объекты по степени близости характеристик.
- Модели ассоциаций находят часто встречающиеся множества объектов.
- Модели последовательностей предсказывают события по цепочке наблюдений.
- Модели числовых рядов используют для предсказания числовых значений наблюдений.

Эти и многие другие модели помогают принимать более обоснованные решения.

Многие задачи управления имеют формальную постановку и это является основой для применения различных математических методов и моделей оптимизации от симплекс-методов до динамического управления.

Наиболее сложными являются задачи управления персоналом. Методикой мобилизации коллектива для решения задач управления является сбалансированная система показателей – ССП (см. 1.3.1). В ее основе лежит измерение не только финансовых результатов, но и всех факторов, влияющих на бизнес-процесс. С одной стороны, это усложняет систему учета (дополнительно необходимо учитывать участие персонала, временные параметры, качественные характеристики), с другой – появляются возможности выявления «узких» мест бизнес-процесса и мотивации персонала.

Многие из перечисленных инноваций относятся скорее к менеджменту, а не к информационным технологиям. Все новые возможности, конечно, нужно обсуждать с заказчиком и пользователями, учитывая при этом не только преимущества, но и стоимости реализации и сопровождения.

3.4. Описание и анализ существующей информационной инфраструктуры

Достаточно часто автоматизируемый бизнес-процесс уже обеспечивается некоторой компьютерной поддержкой. В таких случаях возникает дополнительно задача перехода на новую информационную систему с переносом накопленных данных. Существующую информационную систему (подсистему) бизнес-процесса нужно описать, указав следующие компоненты:

1. Технические средства обработки информации (серверы, персональные компьютеры, компьютерные коммуникации и сетевое оборудование).
2. Сетевая архитектура (сетевые протоколы, сегментация сети, операционные системы, система распределения полномочий и предоставления сервисов).

3. Автоматизированные виды деятельности и применяемое для этого программное обеспечение (специализированные программы для автоматизации различных видов деятельности).

4. Система хранения данных (локальные БД и БД коллективного доступа с указанием СУБД и серверов, файловая система (серверы) коллективного доступа с указанием регламента использования и т.д.).

5. Набор информационных услуг и порядок их предоставления (к таким относятся коммуникационные сервисы, доступ в интернет, информационно-справочные системы и т.д.).

6. Географию узлов обработки данных (размещение компьютеров в одном или нескольких зданиях или регионах).

Основным в описании должен стать пункт 3) – именно в нем должны быть раскрыты функциональные возможности существующей ИС.

Анализ существующих информационных услуг и технологий должен включать следующие пункты:

- полнота автоматизации – сравнение функциональных возможностей существующей ИС с некоторым идеальным уровнем (лучшими образцами в соответствующей отрасли);

- производительность системы – выделяются узкие места ИС, которые существенно тормозят работу пользователей;

- степень интеграции различных подсистем – выделяются одинаковые данные различных подсистем, оценивается необходимость и возможность обмена данными между ними;

- анализ предоставляемых (выходных) данных по направлениям:

- недостаток информации для решения;
- форма информации неудобна для управления;
- несвоевременность информации;
- избыток информации;
- неточность информации;

- анализ хранимых данных – выявление следующих свойств:

- полнота;
- целостность;
- избыточность;
- гибкость (возможность выполнения сложных запросов);

- анализ сервиса:

- точность;
- надежность;
- простота использования;
- гибкость (управление исключительными, нестандартными ситуациями);
- координированность;

- соответствие современному уровню информационных технологий – сравнение с некоторым технологическим идеалом, при этом следует помнить,

что передовые технологии и дороже, и требуют больше ресурсов (например, применение технологии «клиент-сервер» требует квалифицированного сопровождения администратором БД и вряд ли оправданно для ведения локальной БД);

- проблемы, возможности и ограничения информационной системы – связаны чаще всего с реализацией в существующей ИС устаревшей технологии управления;

- экономический анализ (стоимость и прибыльность проектных решений);

- анализ управляемости (много или мало управления) и безопасности;

- анализ эффективности (возможность увеличить и / или уменьшить производительность работы пользователей).

Анализ завершается предложениями по модернизации архитектуры существующей ИС. Эти предложения вместе с анализом технологии управления и возможностями ее улучшения на основе информационных технологий нужны для выработки представления о том, какая ИС нужна для данного бизнес-процесса и что для этого нужно сделать.

3.5. Определение требований и приоритетов. Разработка технического задания

На данном этапе определяются требования к информационной системе. Приоритетными являются функциональные требования: что и как должно быть автоматизировано. Кроме этого, формулируются технологические требования по всем видам обеспечения. Требования формируются с учетом трудовых и временных затрат, а также других ресурсов, необходимых для реализации проекта и поддержания эксплуатации системы. Учитываются совместимость с существующей информационной системой и перспективы развития. В результате создается и утверждается принципиально важный документ – техническое задание на разработку ИС.

Работы на этом этапе требуют профессиональных знаний в области информационных технологий, хорошей эрудиции и навыков проектировщика. Нужно представить множество вариантов проектируемой ИС, оценить каждый на соответствие потребностям всех пользователей, ограничениям по стоимости проекта, затратам на поддержание ИС, возможностям развития. Ошибки в постановке задачи потребуют впоследствии существенных переделок и дорого будут стоить и разработчику, и заказчику.

Техническое задание регламентирует взаимоотношения заказчика и исполнителя работ, поэтому кроме требований оно содержит план выполнения работ, размеры и порядок финансирования.

Для формирования технического задания выполняются следующие работы:

1. Идентификация требований.

2. Назначение приоритетов требованиям: обязательные, желательные (реализуемые позже), по выбору.

3. Построение плана и бюджета проекта.

4. Оформление и утверждение технического задания.

Существует ГОСТ 34.602-89, регламентирующий содержание и оформление технического задания на создание автоматизированной системы. По причине важности этого документа ниже приводятся основные положения ГОСТа.

Техническое задание включает следующие пункты:

1. Общие сведения:

- наименование системы и ее условное обозначение;
- номер договора или шифр темы;
- наименования и реквизиты предприятий разработчика и заказчика;
- перечень документов, на основании которых создается система;
- сроки выполнения работ;
- сведения об источниках и порядке финансирования;
- порядок предъявления результатов.

2. Назначение и цели создания (развития) системы:

- назначением являются виды автоматизируемой деятельности, дополненные перечнем объектов автоматизации;
- цели формулируют как требуемые значения показателей, которые должны быть достигнуты в результате внедрения системы.

3. Характеристики объекта автоматизации:

- краткие сведения об объекте или ссылка на соответствующие документы;
- условия эксплуатации системы на объектах автоматизации.

4. Требования к системе:

- требования к системе в целом:
 - к структуре и функционированию системы: компоненты системы, способы и средства связи компонент; взаимосвязи с другими системами;
 - к режимам функционирования; к диагностическим процедурам;
 - перспективы развития и автоматизации;
 - к численности, квалификации и режиму работы персонала;
 - к надежности: показатели надежности; перечень аварийных ситуаций;
- требования к надежности технических средств и программного обеспечения, требования к методам оценки надежности;
 - к эргономике и технической эстетике;
 - требования к защите информации от несанкционированного доступа;
 - к сохранности информации при авариях;
- требования к функциям (задачам):
 - перечень функций (задач, комплексов);
 - регламент реализации каждой функции;
 - для каждой функции требования к качеству, форме представления выходных данных, точности, ресурсам для выполнения;
- требования к видам обеспечения:
 - к математическому обеспечению: состав алгоритмов, типовые алгоритмы, требования к алгоритмам;

- к информационному обеспечению:
 - к составу, структуре и способам организации данных;
 - к информационному обмену между компонентами системы;
 - к информационной совместимости с другими системами;
 - к использованию классификаторов;
 - к применению СУБД;
 - к процессу сбора, обработки, передачи и представления данных;
 - к защите от разрушения при авариях и сбоях;
 - к контролю хранения, к обновлению и восстановлению данных;
 - к процедуре придания юридической силы документам, полученных при помощи технических средств системы;
 - к программному обеспечению: требования к независимости программ от вычислительной техники и операционной системы; к качеству, способам обеспечения качества, к контролю качества;
 - к техническому обеспечению: к видам и характеристикам технических средств;
 - к организационному обеспечению: к структуре и функциям подразделений, участвующих в функционировании и обеспечивающих эксплуатацию; к организации функционирования и порядку взаимодействия персонала АС и персонала объекта автоматизации; к защите от неправильных действий персонала.
5. Состав и содержание работ по созданию системы:
- перечень стадий и этапов разработки, сроки выполнения, исполнители;
 - перечень документов, предъявляемых по окончании каждого этапа работ.
6. Порядок контроля и приемки системы:
- виды, состав, объем и методы испытаний;
 - общие требования к приемке работ, порядок согласования и утверждения приемочной документации;
 - состав приемочной комиссии.
7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие:
- приведение поступающей в систему информации к виду, пригодному для обработки;
 - изменения в объекте автоматизации;
 - создание условий функционирования объекта автоматизации для соответствия условия эксплуатации системы;
 - создание подразделений и служб, необходимых для функционирования системы;
 - сроки, порядок комплектации штатов и обучения персонала.
8. Требования к документированию – перечень комплектов документации на бумажных и машинных носителях.
9. Источники разработки – документы, на основании которых разрабатывается ИС.

При наличии утвержденных методик в техническое задание включают расчет ожидаемой эффективности системы; оценку научно-технического уровня системы.

Оформление титульного листа технического задания приведено на рис. 13. Листы нумеруют, начиная с листа, следующего за титульным. Заключительный лист включает две приведенные ниже таблицы с подписями.

СОСТАВИЛИ

Наименование предприятия	Должность исполнителя	Фамилия имя, отчество	Подпись	Дата

СОГЛАСОВАНО

Наименование предприятия	Должность исполнителя	Фамилия имя, отчество	Подпись	Дата

<p>УТВЕРЖДАЮ</p> <p><руководитель (должность, наименование предприятия-заказчика)></p> <p><личная подпись> <расшифровка></p> <p><Печать> <дата></p> <p><наименование системы (проекта)></p> <p><наименование объекта автоматизации></p> <p><сокращенное наименование автоматизированной системы></p> <p>ТЕХНИЧЕСКОЕ ЗАДАНИЕ</p> <p>На <количество листов> листах</p> <p>Действует с <дата></p> <p>СОГЛАСОВАНО</p> <p><руководитель (должность, наименование согласующей организации)></p> <p><личная подпись> <расшифровка></p> <p><Печать> <дата></p>	<p>УТВЕРЖДАЮ</p> <p><руководитель (должность, наименование предприятия-разработчика)></p> <p><личная подпись> <расшифровка></p> <p><Печать> <дата></p>
--	---

Рис. 13. Оформление титульного листа технического задания

Стандарт предлагает полный набор пунктов технического задания. На практике незначимые для конкретного проекта пункты опускают. Например, если организация выполняет проект силами своих работников, то из пункта «Общие сведения» исчезают номер договора или шифр темы, реквизиты разработчика.

Исполнитель представляет проект технического задания на согласование и утверждение. Согласование проекта не должно превышать 15 дней. Замечания по проекту технического задания представляются с техническим обоснованием. Разногласия заказчика и разработчика оформляются протоколом. Копии технического задания в 10-дневный срок рассылаются исполнителям.

В характеристике объекта автоматизации приводят только те сведения, которые значимы для разработки: перечень подразделений (рабочих мест), характеристики сети и компьютеров, используемые операционные системы, квалификация персонала и т.д.

Наиболее важный пункт – «Требования к системе». В требованиях к системе в целом главным является структура проектируемой системы. Перечисляют компоненты системы – обычно ее составляют серверы (такие как сервер баз данных и интернет-сервер) и клиентские приложения. Для каждого компонента приводят назначение и описывают регламент взаимодействия компонент. Функции каждого компонента должны быть более подробно описаны в п. «Требования к функциям».

В «Требованиях к функциям» для функций ввода и редактирования данных недостаточно только указать предмет описания (клиент, товар...), требуется перечислить поля, используемые справочники, процедуры проверки корректности вводимых данных. Использование нормативно-справочных данных должно сопровождаться включением в ТЗ функций по корректировке или импорту соответствующих справочников. Описание поиска и сортировки должны включать списки полей поиска и сортировки, а также правил формирования условий поиска.

Функции вывода данных связаны обычно с формированием отчетов. Недостаточно просто перечислить наименования отчетов. Каждый отчет имеет табличную структуру. Для его описания нужно перечислить столбцы отчета, группировки строк, правила формирования итогов по группам и общего итога отчета. Также нужно указать параметры выбора данных для отчета. Обычно в такие параметры включают отчетный период.

Отчетные функции можно заменить или дополнить многомерным анализом (OLAP функциональностью). Для него нужно перечислить формируемые показатели и измерения, по которым анализируются изменения показателей. Для показателей нужно привести правила вычисления – функции агрегирования или другие формулы. Для измерения нужно определить применяемые классификации. В функции многомерного анализа включают построение соответствующих диаграмм.

Как правило, алгоритмы обработки данных общеизвестны. Это могут быть, например, правила учета или определения финансовых показателей. Описание таких алгоритмов не включают в требования к математическому обеспечению. Требования к математическому обеспечению включают, если есть необходимость разработки или использования сложных математических алгоритмов, таких как поиск оптимального решения.

Простое перечисление стандартных характеристик операционных систем и вычислительной техники обычно составляет содержание пунктов «требования к программному обеспечению» и «требования к техническому обеспечению».

Требования к информационному обеспечению в случае использования СУБД сводятся к описанию ее класса (реляционная, документационная, графовая...), характеристик, технологии связи с другими серверами, процессу редактирования данных, защите данных.

Требования к документированию содержат список документов, который можно разделить на две части: во-первых, инструкции по использованию и сопровождению, во-вторых, описание компонентов ИС. Все прочие пункты технического задания стандартны и их разработка не вызывает затруднений.

Гибкие (agile) методы проектирования концентрируют усилия на разработке небольших наборов функций. Детальные требования к таким наборам определяются в тесном взаимодействии с пользователями и не оформляются в виде ТЗ. Однако, такой процедуре разработки всегда предшествует этап планирования, на котором определяется общая архитектура системы и набор функций. Такой план является достаточно близким по структуре к ТЗ. В него включают решения по архитектуре системы и основные блоки функций.

Большой программистский проект ... включает в себя много отдельных задач, каждая из которых может зависеть от окончания другой. Вероятность того, что каждая задача будет идти нормально, становится исчезающе малой.

Фредерик Брукс-мл.

4. КОНСТРУИРОВАНИЕ

4.1. Работы стадии конструирования

На этой стадии предстоит разработать архитектуру – каркас будущей информационной системы. На основании требований к системе с учетом мнений и рекомендаций пользователей вырабатываются принципиальные информационные решения по разработке функций информационной системы, методов и алгоритмов вычислений, структур хранения данных, структуры сети, интерфейса пользователей.

Работы, рассматриваемые в этой главе, соответствуют следующим стадиям разных моделей жизненного цикла:

- «Эскизный проект», «Технический проект» в жизненном цикле автоматизированной системы (ГОСТ 34.601-90).
- «Предварительное (внешнее) проектирование», «Детальное (внутреннее) проектирование» в жизненном цикле программного средства.
- «Планирование» в спиральном жизненном цикле информационной системы по методике Microsoft Solution Framework.

Однако, прежде надо решить стоит ли разрабатывать информационную систему или лучше воспользоваться готовыми решениями. Использование готовых продуктов, безусловно, намного проще и надежнее, если на рынке есть продукты с нужным набором функциональных характеристик. Покупка выполняется обычно по следующей схеме:

1. Формулировка требований (техническое задание).
2. Сбор предложений поставщиков.
3. Оценка и ранжирование предложений.
4. Разработка контрактов.

Если все же выбирается вариант разработки, то в этом случае и нужно выполнять конструирование и реализацию системы. Конструирование включает следующие виды работ:

1. Построение функциональной модели системы. Выполняется декомпозиция функций системы для получения некоторого элементарного уровня. Для каждой функции описываются входные и выходные данные, алгоритмы выполнения, взаимодействие с другими функциями, регламент выполнения функции.
2. Построение модели данных. Определяется, какие данные надо хранить для выполнения функций информационной системы. Кроме этого, должны учитываться все информационные потребности организации и перспективы развития

информационной системы. После этого нужно решить вопросы о форме хранения данных на основании требований к доступу, надежности, безопасности данных.

3. Определение набора приложений. Обычно приложения определяются количеством и функциональными характеристиками рабочих мест.

4. Проектирование функций, входных и выходных данных каждого приложения. Одна и та же функция может применяться в разных приложениях. Для минимизации затрат на разработку однотипных функций следует определить модульную или компонентную структуру проекта.

5. Проектирование интерфейса пользователя. Вырабатываются принципы построения интерфейса, способы задания параметров, ввода данных, использования справочников, определяются стандарты терминологии и компонентов интерфейса.

6. Проектирование структуры и логики работы программ.

На стадии конструирования особенно целесообразно использование единого для всех разработчиков репозитория для хранения документации на подсистему и всех описаний, связанных с проектом. Применение репозитория решает задачи распространения изменений в документации и заменяет передачу данных от этапа к этапу предоставлением их по мере готовности, что создает предпосылки для распараллеливания и децентрализации работ.

Конструктивные решения оформляются набором документов. Ниже приведен список документов, разработку которых предусматривают ГОСТы и которые целесообразно разрабатывать на стадии конструирования:

- Пояснительная записка к эскизному (техническому) проекту.
- Общее описание системы.
- Схема организационной структуры.
- Описание организационной структуры.
- Схема функциональной структуры.
- Описание автоматизируемых функций.
- Описание постановки задачи (комплекса задач).
- Описание технологического процесса обработки данных.
- Схема автоматизации.
- Описание комплекса технических средств.
- Описание информационного обеспечения.
- Описание организации информационной базы.
- Описание алгоритма (проектной процедуры).

Требования к содержанию этих документов приведены в приложении.

Известно, что программисты не любят писать документацию. Среди них даже бытует мнение, что это форма наказания для неугодных. Есть одно оправдание такой ситуации: много документов создается раньше программ, и они теряют актуальность по мере внесения изменений в проект на стадии реализации. Таким образом, создается впечатление, что документы оформлены зря, так как их все равно приходится переделывать. Тем не менее, все разработчики, а особенно руководители проектов должны знать, что если разработка не документируется, то риск неудачи существенно возрастает.

С другой стороны, изменения в технологии проектирования требуют новых подходов к документированию. Наверное, никто уже не вычерчивает эскизы окон, поскольку все компоненты интерфейса пользователя просто размещаются на экранной форме при помощи графических инструментальных средств проектирования. В данном случае описание непосредственно становится программой. Такой «симбиоз» описаний и программ является идеальным вариантом документирования проекта.

К сожалению, значительная часть проектных решений не имеет жестких связей со структурами программ и данных. Это является основным препятствием создания конверторов из описаний в программы и наоборот. К тому же для такого конвертирования требуется создание формальных языков, которые обычно плохо воспринимаются неподготовленным пользователем. Тем не менее, постоянно предпринимаются усилия автоматизировать сам процесс проектирования и документирования информационной системы. Результатом таких усилий являются различные CASE-средства.

Применение современных инструментальных систем программирования и CASE-средств приводит к размыванию границ между конструированием и реализацией программного и информационного обеспечений. Часть программного текста появляется в результате проектирования интерфейса, также как модели данных легко превращаются в соответствующие структуры данных современных СУБД.

4.2. CASE-технология проектирования информационных систем

CASE – это акроним Computer-Aided System / Software Engineering, объединяющий методы автоматизации самого процесса проектирования информационных систем [8; 15]. CASE средства поддерживают работы на всех этапах жизненного цикла системы. Целью CASE-технологии является ускорение процесса создания систем и повышение их качества.

Идея CASE-технологии весьма привлекательна: по имеющимся описаниям проектных решений автоматически получать программы, схемы данных, инструкции и другие эксплуатационные документы (рис. 14). Противоположные средства – reverse engineering (design recovery) – восстанавливают описания более высокого уровня по описаниям низкого уровня (по выполняемому модулю восстанавливается текст программы, по программе – структурные схемы, диаграммы потоков, деревья решений). Программа, как конечный продукт проектирования, содержит не только идеи программиста, но также и идеи системного аналитика и требования пользователей. Все это может быть использовано в новой системе.

Применение этих идей требует введения формализованных описаний для преобразуемых компонент. Если для прямого проектирования этого можно добиться, применяя строго определенные конструкции для описания проектных ре-

шений, то для обратного проектирования формализовать процесс удастся гораздо реже – не так часто тексты программ удовлетворяют формальным требованиям.

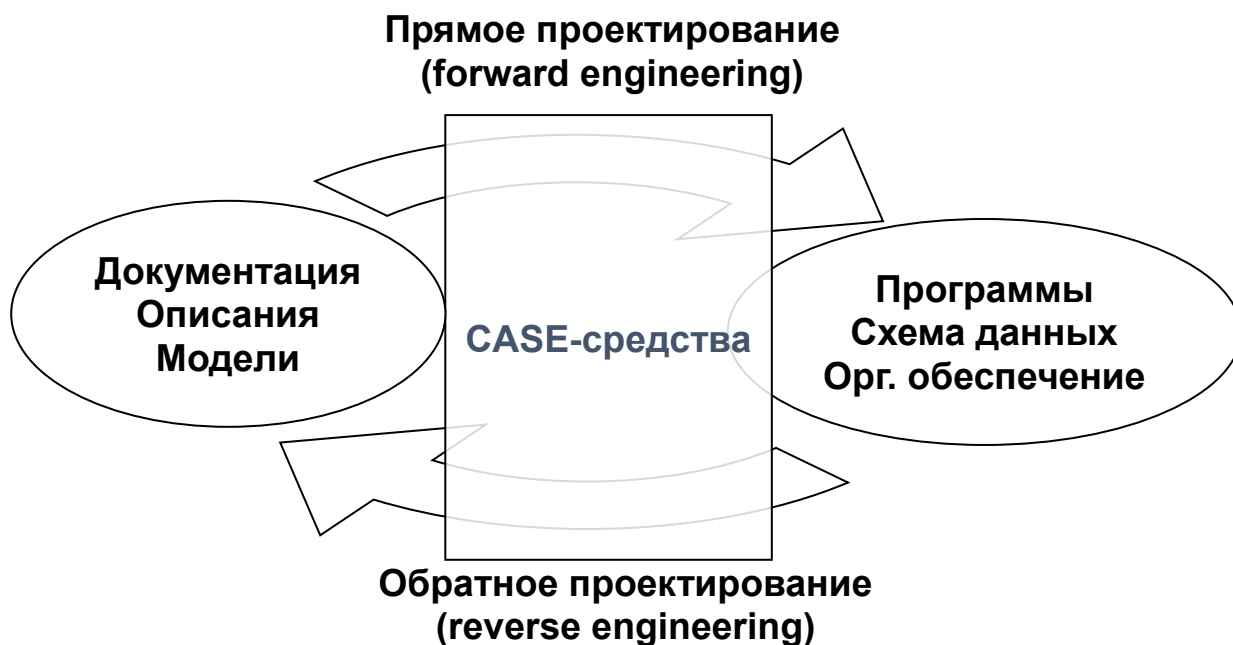


Рис. 14. Использование CASE-средств

В 1970-х годах Daniel Teichrowe создает Problem Statement Language (PSL) и для него разрабатывает программу Problem Statement Analyzer (PSA), выполняющую анализ требований пользователя на полноту и последовательность. В 1984 году компания INTERSOLV (прежнее название IndexTechnology) выпустила Exelerator. Его успех породил термин CASE и индустрию CASE-средств.

По степени охвата этапов жизненного цикла CASE-программы классифицируют следующим образом:

- Upper-CASE для поддержки планирования, анализа и общего конструирования систем;
- Lower-CASE для детального конструирования, разработки и сопровождения;
- Cross life cycle-CASE для организации работ на всех этапах жизненного цикла.

В состав Upper-CASE включают:

- средства планирования развития информационных систем, автоматизирующие описание целей и задач, определение критических факторов успеха, ключевых показателей результативности, формулирование проблем, описание организационной структуры, информационных и функциональных требований;
- средства системного анализа и конструирования, позволяющие описать границы проектируемой системы, модель существующей системы, выполнить конструирование системы, отвечающей требованиям пользователей, создать прототипы специфических частей системы (отчетов, экранных форм...).

В состав Lower-CASE включают:

– средства конструирования и реализации систем:

- инструментальные средства прикладного программиста, поддерживающие кодирование, компиляцию, тестирование и отладку на интеллектуальных терминалах;

- генераторы компонент, выполняющие генерацию соответствующих компонент по их описанию (генераторы БД, создание графического интерфейса пользователя, средства реализации технологии «клиент-сервер»);

- генераторы программ (кода), создающие текст программ по описаниям системы;

– средства поддержки ИС:

- помогают программистам реструктурировать текст программы, чтобы сделать ее более мобильной;

- помогают изменять систему по требованиям пользователя;

- облегчают переход на новые технологии;

- помогают определить, стоит ли еще поддерживать ИС или необходимо создавать ее заново;

- восстанавливают полезную информацию из устаревшего программного обеспечения.

Главным в средствах поддержки является проектирование новых программ (forward engineering), а также изменение и реструктуризация существующих программ в соответствии с описаниями. Гораздо сложнее реализовать противоположную процедуру reverse engineering. Помимо логической сложности разбора логики программы по ее тексту, следует учитывать, что старые программы были написаны без соблюдения стандартов и извлекаемая в таких случаях информация формулируется в физических терминах (например, в адресах переменных).

В Cross life cycle CASE включают следующие средства:

– управление проектом – планирование, составление расписаний и бюджетов, составление отчетов, управление ресурсами;

– управление процессами;

– оценивание времени и стоимости проекта, а также прочих ресурсов;

– создание плановой и отчетной документации по выполнению проекта.

Архитектура CASE-средств продемонстрирована на рис. 15. *Репозиторий* (энциклопедия, словарь данных) является ядром системы и представляет собой специализированную базу данных, на основе которой выполняется синхронизация всех описаний (диаграмм) и координация усилий всех разработчиков. Репозиторий содержит информацию об объектах проектируемой ИС и взаимосвязях между ними:

– проектировщики и их права доступа к различным компонентам системы;

– диаграммы, отражающие организационные, функциональные, информационные, технологические и другие аспекты системы;

– компоненты диаграмм;

– связи между диаграммами;

– структуры данных;

– программные модули;

- процедуры;
- библиотеки модулей и т.д.

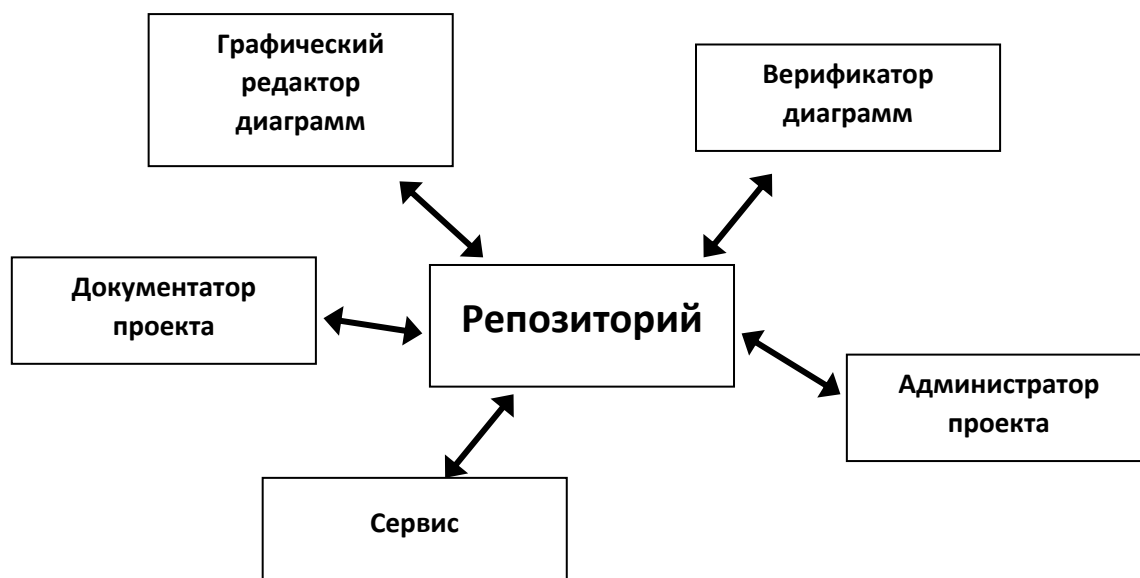


Рис. 15. Архитектура CASE-средств

Графический редактор диаграмм позволяет создавать различные описания системы в виде схем и может выполнять следующие операции:

- создавать элементы диаграмм и взаимосвязи между ними;
- задавать описания элементов диаграмм;
- задавать описания связей между элементами диаграмм;
- редактировать элементы диаграмм, их взаимосвязи и описания.

Верификатор диаграмм контролирует правильность диаграмм и синхронизирует диаграммы между собой.

Документатор проекта генерирует документацию проекта в виде различных отчетов. Отчеты могут группироваться по нескольким признакам, например, по времени, автору, элементам диаграмм, диаграмме или проекту в целом.

Администратор проекта автоматизирует выполнение следующих административных функций:

- инициализации проекта;
- задания начальных параметров проекта;
- назначения и изменения прав доступа к элементам проекта;
- мониторинга выполнения проекта.

Сервис выполняет функции архивации данных, восстановления данных и создания нового репозитория. Основные возможности CASE-средства определяются набором моделей, который поддерживает графический редактор диаграмм. Некоторые такие модели, используемые на стадиях планирования и анализа (организационная, функциональная, схемы документооборота...), уже были рассмотрены. Модели, применяемые на стадии конструирования, образуют метафору проектирования или подход к проектированию.

4.3. Язык объектно-ориентированного моделирования UML

CASE-технология проектирования основана на применении различных графических моделей, дополненных описаниями каждого элемента. Графические модели для описания различных видов обеспечения ИС должны учитывать соответствующую специфику. Наиболее универсальную систему моделей предлагает язык UML [20].

Язык UML – Unified Modeling Language проявился сравнительно недавно. В период с 1975 по 1989 идеологию объектно-ориентированного программирования активно пытались распространить на смежные с программированием области, в том числе и на проектирование. Основные понятия: объект, класс достаточно успешно пытались применить для моделирования отдельных предметных областей реального мира. В результате с 1989 по 1994 год появилось до 50 разных языков объектно-ориентированного моделирования. С 1994 года наиболее продвинутые в этой области ученые: Grady Booch, Ivar Jacobson, James Rambo и другие – договорились объединить усилия по созданию языка, который включил бы наиболее удачные конструкции. В результате и появился UML и продукты, использующие этот язык.

Сложность и многоплановость языка UML является следствием сложности решаемых с его помощью задач:

1. Задачи конструирования.
2. Задачи визуализации.
3. Задачи документирования различных аспектов информационной системы, которые включают:
 - требования к системе;
 - архитектуру системы;
 - проектирование системы;
 - тексты программ;
 - планы работ;
 - тестирование;
 - прототипы.

Для отражения разных сторон информационной системы предложены диаграммы следующих типов:

- диаграммы прецедентов (Use-case diagram) для описания функциональности ИС и ее использования;
- диаграммы классов объектов (Class diagram) для описания объектных структур программ и данных;
- диаграммы состояний (State diagram) для описания поведения объекта, как последовательности смены его состояний под действием входных сигналов (сообщений);
- диаграммы последовательностей (Sequence diagram) и диаграммы кооперации (Cooperation diagram) для описания взаимодействия объектов;
- диаграммы деятельности (Activity diagram) для построения алгоритмов – некий аналог блок-схем с возможностью распараллеливания и синхронизации действий и описания использования данных;

- диаграммы пакетов (Package diagram) для описания пакетной (модульной) структуры программы;
- диаграммы компонентов (Component diagram) для применения компонентного подхода к построению программного обеспечения;
- диаграммы размещения (Deployment diagram) для описания развертывания информационной системы по узлам вычислительной сети.

Диаграммы отражают различные стороны построения, поведения и использования объектов, составляющих информационную систему, и поэтому они должны использоваться в комплексе целостном и непротиворечивом. Соответствующие CASE-средства не только обеспечивают построение диаграмм и определение структуры объектов, но и предоставляют средства проверки целостности и непротиворечивости, администрирования и документирования всего проекта в целом, а также средства переноса объектной структуры в другие среды программ и данных.

4.3.1. Диаграмма прецедентов (use case diagram)

Диаграммы прецедентов могут использовать для моделирования:

- контекста системы в виде входных и выходных данных, объектов, функций информационной системы;
- организационной структуры предметной области;
- взаимодействия функций.

На диаграммах прецедентов используются обозначения, представленные на рис. 16.

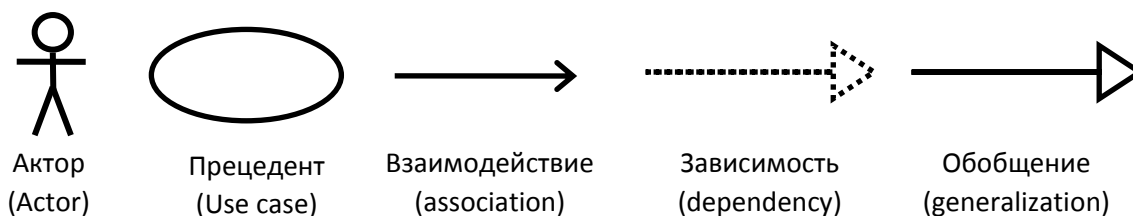


Рис. 16. Обозначения диаграмм прецедентов

Под актором понимают логически связанное множество ролей (люди, программы), взаимодействующих друг с другом и с информационной системой. Чаще всего актор является пользователем. Прецедент (вариант использования) является описанием множества действий. Между акторами и прецедентами могут быть различные отношения (связи):

- ассоциация (взаимодействие) – это отношение, показывающее, что один объект как-то связан с другим объектом. На диаграммах ассоциация отображается линией. При необходимости ассоциация может иметь направление, указываемое стрелкой. Например, так можно отображать передачу данных от одного объекта другому.

– зависимость – это отношение, показывающее, что один объект зависит от другого. Для зависимостей можно вводить стереотипы. Для зависимостей двух прецедентов создатели UML предусмотрели два стандартных стереотипа:

– <<uses>> использование – применение одним прецедентом другого прецедента в процессе выполнения;

– <<extend>> расширение – один прецедент является вариантом исполнения другого.

– обобщение – это отношение, показывающее, что один объект (родитель) обобщает свойства и поведение других объектов (потомков). Потомки наследуют свойства и методы родителя.

На рис. 17 приведен прецедент формирования списка экзаменов сессии некоторой студенческой группы. Формирует этот список инспектор деканата. При этом он может использовать другую функциональную возможность – прецедент выбора списка экзаменов из учебного плана группы.

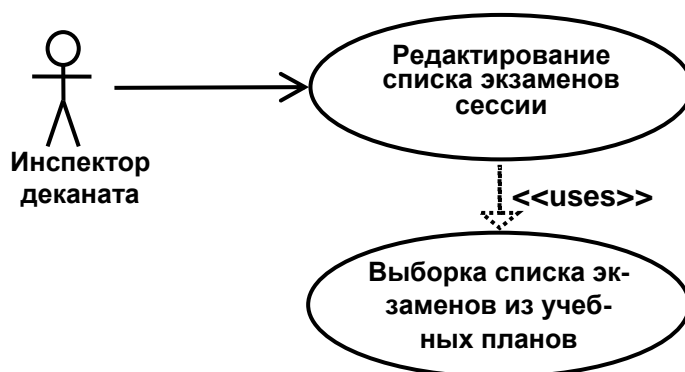


Рис. 17. Пример включения (<<uses>>)

Для печати экзаменационной ведомости (рис. 18) инспектор деканата указывает группу и экзамен. При этом возможно два варианта печати ведомости со всеми студентами или только с задолжниками.

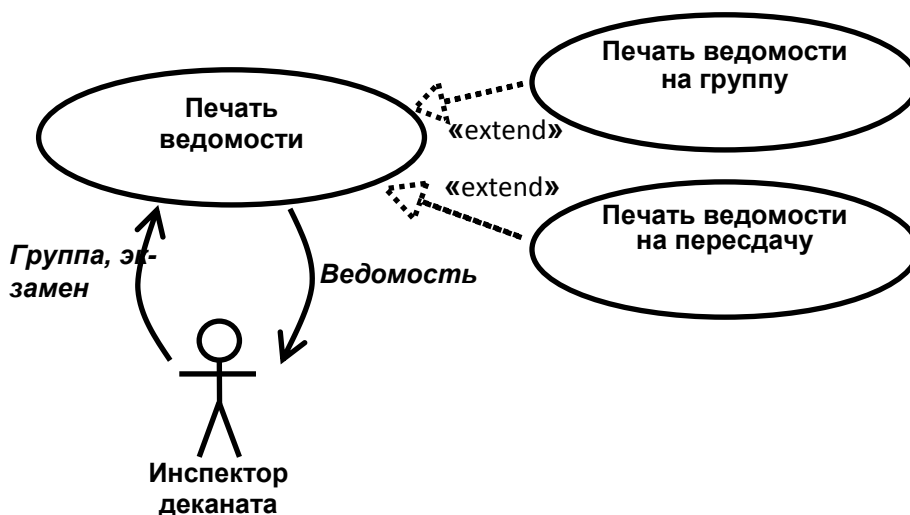


Рис. 18. Пример использования (<<extend>>)

4.3.2. Диаграмма классов (Class diagram)

Диаграмма классов описывает классы и связи между классами. Класс описывает множество однотипных объектов, которые характеризуются одинаковыми наборами свойств (атрибутов), операций (методов), а также одинаковым смысловым значением и связями. Класс является шаблоном для создания объекта (рис. 19).

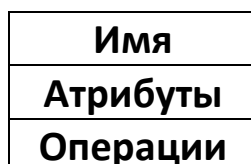


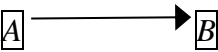
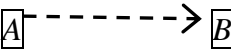
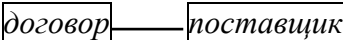
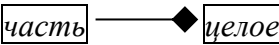
Рис. 19. Изображение класса

Свойство – это не только значение атрибута. Оно также отражает состояние объекта. Изменение свойства сопровождается действиями по приведению объекта в соответствие со значением свойства (например, присваивание Visible значения true приводит к визуализации объекта).

Вызов операции – это способ заставить объект выполнить некоторые действия и / или изменить состояние. С операциями связано понятие полиморфизма – возможности определения класса и конкретной операции объекта по имени операции. Например, вызов операции Show приводит к определению класса объекта и выполнению операции Show именно этого класса. Между классами могут устанавливаться связи различных видов (табл. 5).

Таблица 5

Виды связей классов

Вид связи	Описание вида связи
	<i>Обобщение</i> (generalization) – наследование: А является потомком В (по числу родителей: одиночное и множественное)
	<i>Зависимость</i> (dependency) – А зависит от В, если изменения в описании В могут повлиять на А (но не наоборот)
	<i>Ассоциация</i> (association) – объекты одной сущности связаны с объектами другой. К ассоциациям применимы четыре базовых дополнения: имя, роли, кратность
	<i>Агрегирование</i> (aggregation) – один класс является частью другого

Одним из фундаментальных видов связей является наследование. Наследование является наиболее эффективным способом определения класса и связывает классы в иерархические структуры, подобные генеалогическому дереву. Класс может связываться с другим классом использованием его атрибутов и методов.

В программах классы (объекты) связываются отношениями принадлежности (зависимости). Например, экранная форма является владельцем элементов,

размещенных на форме. Отношение подчиненности определяет порядок обработки событий программными объектами. С другой стороны, подчиненность формирует иерархию программных объектов, создаваемую во время работы программы.

Диаграмма классов моделирует статическую структуру предметной области и может использоваться для моделирования объектной структуры приложений, для моделирования структуры БД и для моделирования других компонентов информационной системы. Для часто встречающихся классов определены стереотипы классов и введены соответствующие обозначения (рис. 20).

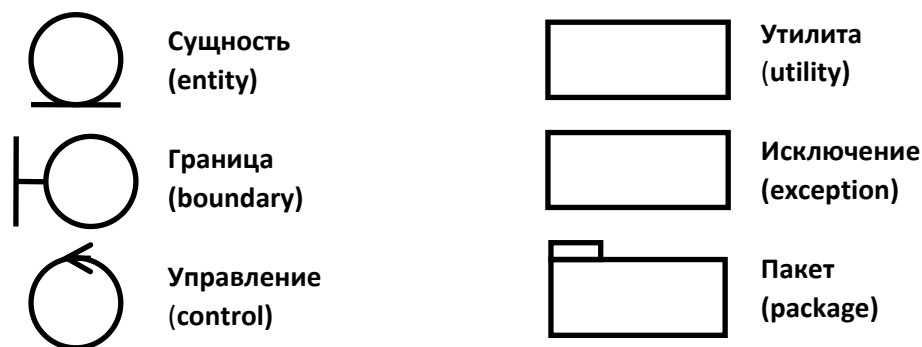


Рис. 20. Стереотипы классов в UML

Класс «сущность» предназначен для моделирования структур данных. Отличается от сущности в модели «Сущность – связь» возможностью дополнения списка атрибутов набором операций. Класс «граница» описывает процессы взаимодействия системы с пользователем или с другими системами. Класс «управление» задает управление потоками событий и обработку данных в процессе функционирования системы. Класс «утилита» позволяет определять использование вспомогательных процедур. Класс «исключения» можно использовать для описания обработки исключительных ситуаций. Класс «пакеты» предоставляет механизм объединения объектов в контейнер (аналог модуля в Delphi, C++). Включает интерфейсную часть (доступную для использования) и скрытую часть (особенности реализации).

На рис. 21 представлен пример диаграммы классов для описания структур данных, описывающих оценки студентов. Между классами «Группа» и «Студент» введена связь специального типа «агрегирование» – частный случай ассоциации, который описывает отношение между целым и частями. Как и в модели «сущность-связь» каждая связь имеет наименование (или два для каждого направления) и описание характеристик. Для сущности «Группа» определена операция CopyStudyPlanToExList, которая выбирает список экзаменов для указанной параметрами сессии и дополняет соответствующими объектами класс «Аттестация».

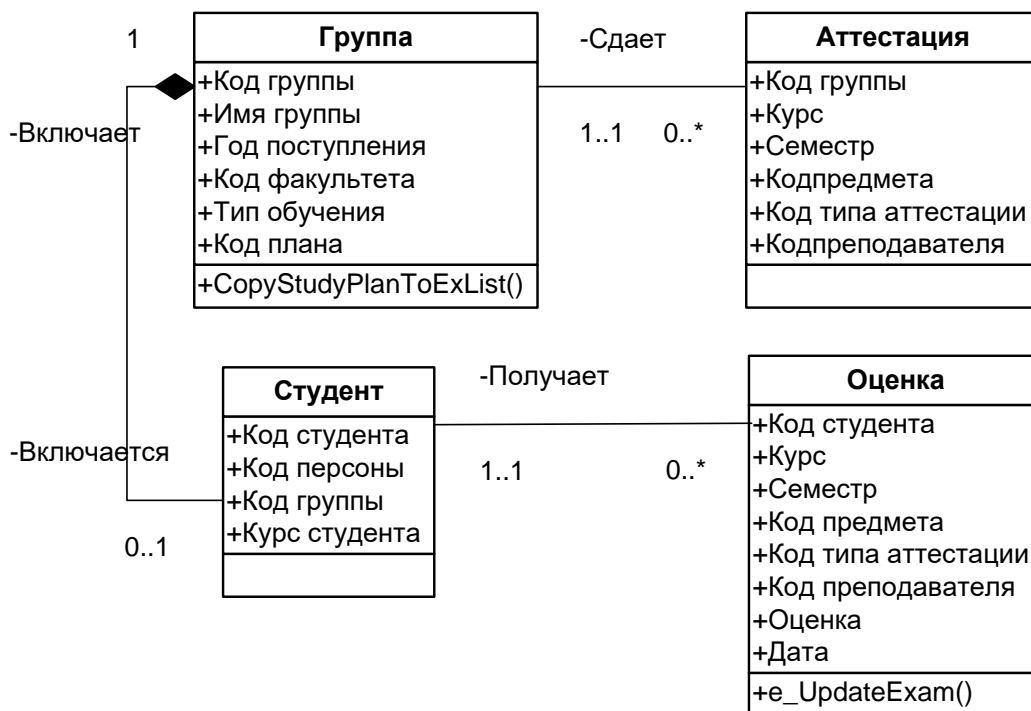


Рис. 21. Пример диаграммы классов для описания данных

Атрибуты и операции помечаются признаком видимости. В языке UML определены следующие уровни видимости:

- **public** (общий) – операцией или атрибутом могут пользоваться другие классы. Обозначаются знаком «+» перед именем атрибута или операции;
- **protected** (защищенный) – операцией или атрибутом могут пользоваться только потомки данного класса. Обозначаются знаком «#»;
- **private** (закрытый) – операцией или атрибутом можно пользоваться внутри класса. Обозначаются символом «-».

4.3.3. Диаграмма последовательностей (Sequence diagram) и диаграмма кооперации (Cooperation diagram)

Диаграмма последовательностей описывает временной порядок взаимодействия компонентов системы в процессе функционирования. На диаграмме отражаются акторы, обычно представляющие пользователей, сообщения, возникающие в процессе работы и объекты, участвующие в обработке сообщений. Диаграмма дает точное представление о сценариях выполнения отдельных функций в информационной системе.

Аналогичные возможности предоставляют диаграммы кооперации, которые применяются для моделирования структурной организации потоков управления. На диаграммах этого типа используются такие же объекты, как и на диаграммах взаимодействия. Семантически диаграммы последовательностей и кооперации эквивалентны и могут быть получены одна из другой. Отличие заключается в способах изображения взаимодействия: диаграмма последовательности

отображает последовательность и продолжительность взаимодействия, а диаграмма кооперации – потоки данных между акторами и объектами.

Диаграммы последовательностей отражают порядок взаимодействия объектов в процессе реализации некоторого прецедента. На диаграмме (рис. 22) в верхней части размещают акторов и объекты, линия вниз от актора или объекта указывает его участие в обмене сообщениями, которые изображаются стрелками от линии к линии. Полосы на линиях отображают время работы объекта.

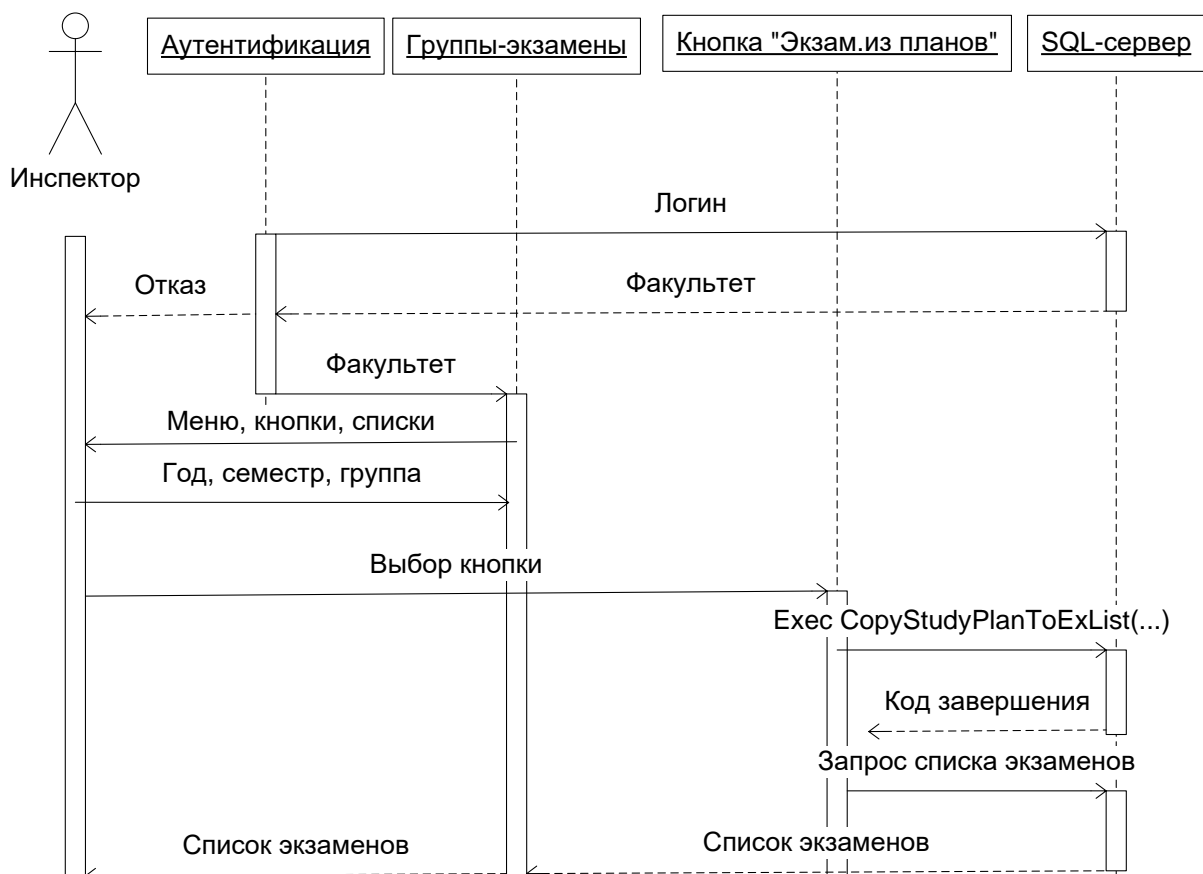


Рис. 22. Диаграмма последовательности для копирования списка экзаменов группы из учебных планов

На рис. 22 инспектор деканата запускает свое приложение, в котором программный объект «Аутентификация» получает логин инспектора, по которому SQL-сервер в специальной таблице определяет факультет. Если факультет найден, то запускается форма «Группы-экзамены», которая предлагает инспектору меню и панель с параметрами и кнопками. Инспектор выбирает год, семестр, указывает группу и выбирает кнопку «Экзамены из планов». В результате SQL-серверу отправляется команда запуска процедуры CopyStudyPlanToExList с соответствующими параметрами. При успешном завершении процедуры выполняется запрос списка экзаменов группы. Полученный список демонстрируется формой «Группы-экзамены» инспектору.

Диаграмма кооперации семантически эквивалентна диаграмме последовательностей и может быть автоматически получена из нее (рис. 23). На ней объекты и акторы соединяются стрелками, изображающими сообщения. Нумерация сообщений отражает последовательность обмена.

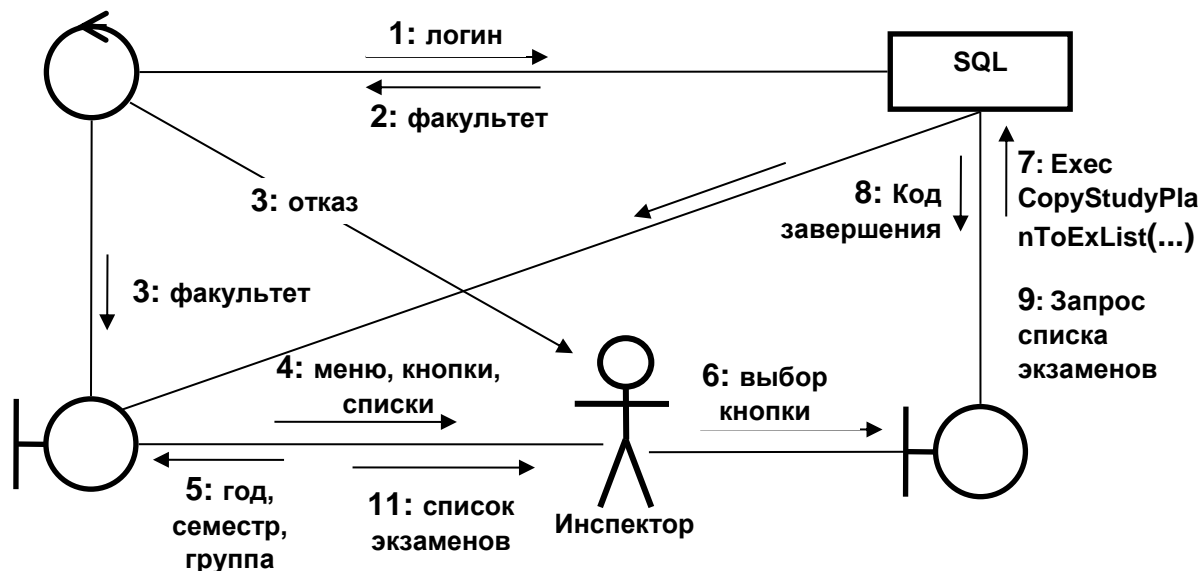


Рис. 23. Диаграмма кооперации для копирования списка экзаменов группы из учебных планов

4.3.4. Диаграмма деятельности (Activity diagram)

Диаграмма деятельности описывает поток переходов от одной к другой. Деятельность в обработке данных представляет собой шаг вычислений в информационной системе. Кроме деятельности на диаграмме отображаются переходы и могут отображаться объекты, используемые в процессах обработки (табл. 6). Диаграмма деятельности аналогична традиционным блок-схемам и, в отличие от них, отражает действия, выполняемые параллельно и порядок синхронизации таких действий. Компоненты диаграммы деятельности могут группироваться в дорожки для указания места выполнения операций.

Таблица 6

Компоненты диаграммы деятельности

Элемент	Описание
	Начальный узел
	Деятельность (activity)
	Поток от деятельности к деятельности
	Использование объектов
	Решение

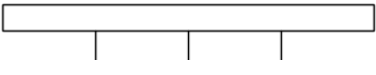
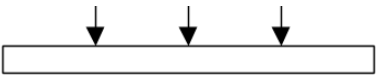

Элемент	Описание
	Разделение потока на деятельности, выполняемые параллельно или произвольно
	Синхронизация
	Выход

Диаграмма (рис. 24) деятельности описывает поток переходов от одной операции (деятельности) к другой. Деятельность в обработке данных представляет собой шаг вычислений в информационной системе. Кроме деятельности на диаграмме отображаются переходы и могут отображаться объекты, используемые в процессах обработки. Диаграмма деятельности аналогична традиционным блок-схемам и, в отличие от них, отражает действия, выполняемые параллельно и порядок синхронизации таких действий.

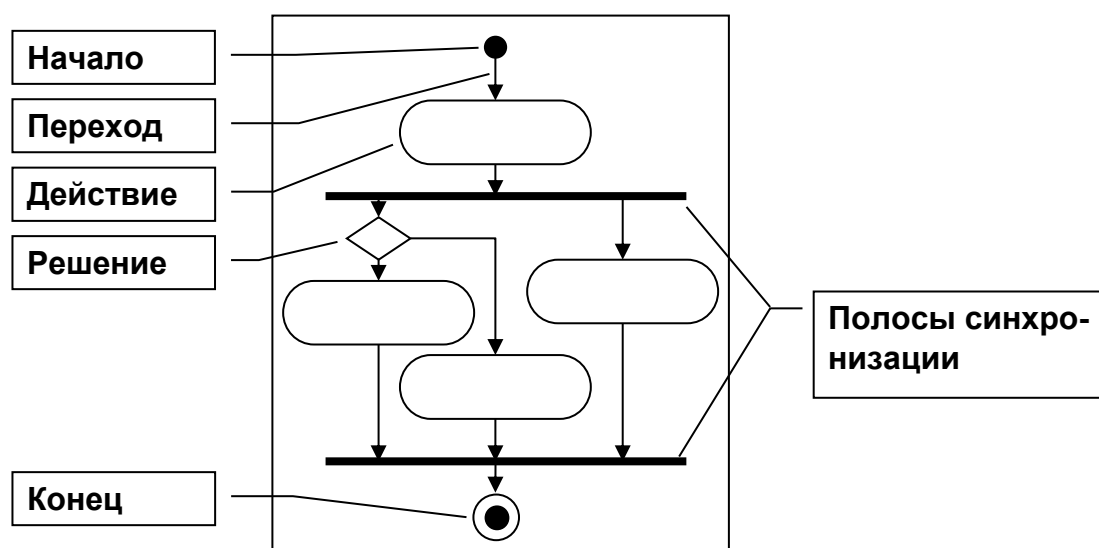


Рис. 24. Обозначения диаграммы деятельности

Рисунок 25 содержит диаграмму деятельности для описания функционирования АРМ инспектора деканата. После аутентификации программа переходит в цикл выбора и выполнения функций до тех пор, пока не будет выбрана функция «Выход». При выполнении функций происходит обращение к СУБД для команд работы с базой данных. На рис. 25 продемонстрировано использование таблицы «Аттестации» в базе данных.

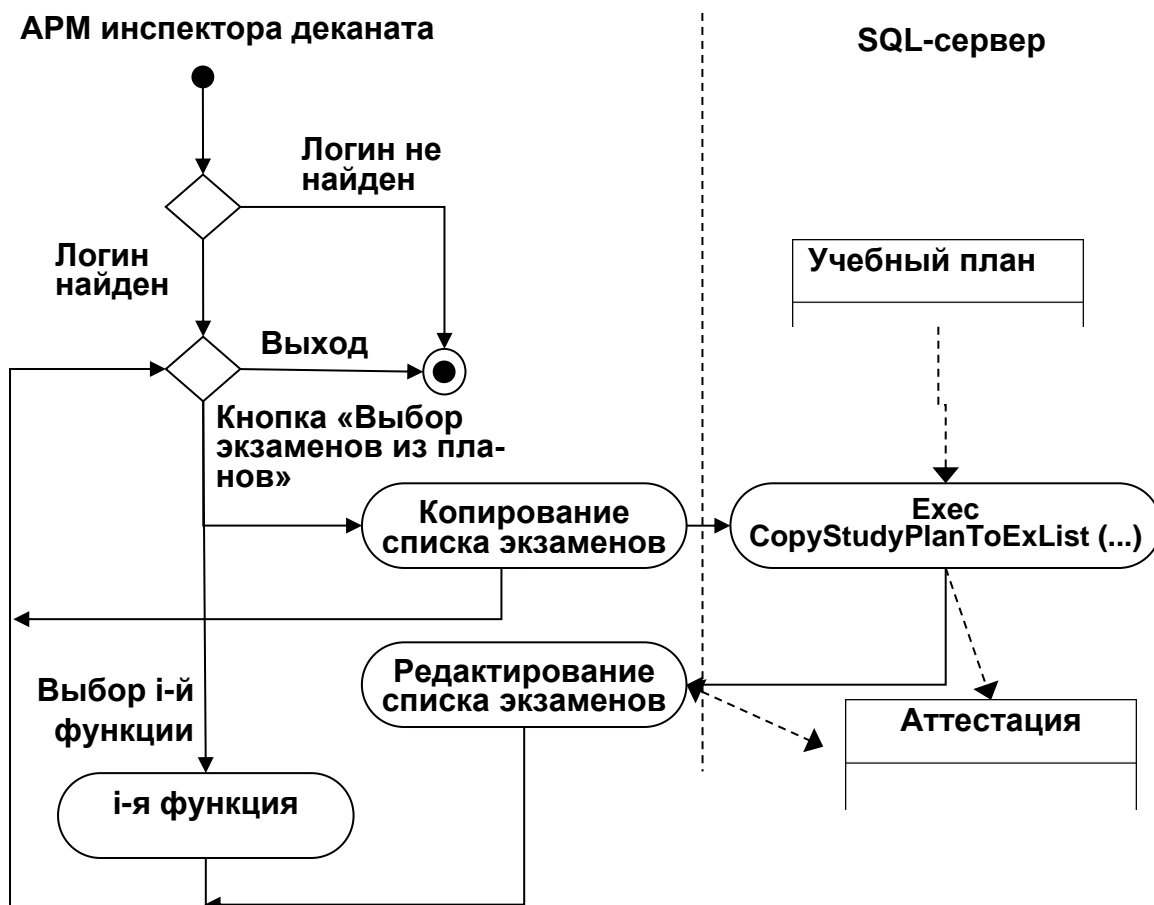


Рис. 25. Пример диаграммы деятельности

4.3.5. Диаграмма состояний (State diagram)

В диаграмме состояний поведение системы моделируется автоматом (state machine). Автомат задается множеством состояний (начальное, конечное, промежуточное). Состояния соответствуют вершинам графа и изображаются прямоугольниками. Из состояния в состояние выполняется переход под действием некоторого события и при выполнении заданных условий в процессе перехода выполняются назначенные действия. Переходы изображаются дугами, нагруженными событиями, условиями и действиями. Автомат – это еще один способ определения алгоритма функционирования объекта.

Для автомата определяются следующие компоненты:

- множество состояний (state) (начальное, конечное, промежуточное);
- переходы (transition) из одного состояния в другое под действием некоторого события (event) и условия (condition), в процессе перехода может выполняться некоторое действие (activity).

На рис. 26 приведен пример моделирования обработки заказа автоматом. Из начального состояния (обозначается окружностью), которое соответствует отсутствию заказа, выполняется переход в следующее при появлении заказа. Переход из состояния «Оформлен» может быть выполнен в два разных состояния, в зависимости от выполнения условий. Заканчивается работа автомата

переходом в конечное состояние (вписанная окружность). Все возможные пути от начального состояния к конечному соответствуют всем вариантам обработки заказа.



Рис. 26. Пример диаграммы состояний для объекта «строка заказа»

4.3.6. Диаграммы пакетов (Package diagram)

Диаграммы пакетов описывают способ организации элементов системы в более крупные блоки, которыми можно манипулировать как единым целым. Понятие пакета является близким понятию модуля, который содержит структуры переменных, классы, процедуры и функции для реализации некоторой функциональности, например, экранной формы. Изображают пакет папкой с закладкой. Пакет владеет своими элементами и определяет область действия имен. Элементом пакета может быть другой пакет. Взаимодействие пакетов описывается отношениями экспорта и импорта. На рис. 27 представлено взаимодействие приложений и сервера баз данных.



Рис. 27. Диаграмма пакетов для описания взаимодействия приложений и SQL-сервера

4.3.7. Диаграммы компонентов (Component diagram)

Диаграмма компонентов является частным случаем диаграммы классов и описывает программу, построенную в виде системы взаимодействующих компонент. Кроме традиционных связей классов для компонент появляется возможность описания интерфейсов компонент.

Компонент – физически заменяемая часть системы, обеспечивающая реализацию некоторого интерфейса. Общими характеристиками компонента с классом являются имена, связи, создание на основе класса компонента соответствующих экземпляров (объектов).

В отличие от классов компоненты физически существуют и могут размещаться в узлах сети, не имеют атрибутов – обладают только операциями, доступными через интерфейс. Интерфейс – набор операций, используемый для специфицирования услуг, предоставляемых классом или компонентом (описывается стереотипным классом). Интерфейс может быть:

- экспортируемый – реализуемый компонентом;
- импортируемый – используемый интерфейс другого компонента.

Интерфейсы моделируют стыковочные узлы системы. Стандартными стереотипами компонент являются executable, library, table, file, document. На рис. 28 представлены компоненты, обеспечивающие взаимодействие приложения и сервера баз данных.

4.3.8. Диаграммы размещения (Deployment diagram)

Диаграмма развертывания (рис. 29) предназначена для моделирования взаимодействия узлов вычислительной сети. Среди узлов выделяют устройства, обеспечивающие интерфейсы пользователей, такие как персональные компьютеры, планшеты, смартфоны, и серверы, обеспечивающие хранение данных – серверы баз данных, серверы предоставляющие коммуникации – интернет серверы, серверы, предлагающие другие услуги, например, почтовые серверы, принт-серверы. Диаграмма развертывания определяет взаимодействие всех узлов сети в виде потоков команд и данных.

На диаграмме рис. 29 показаны персональные компьютеры пользователей и их взаимодействие с серверами кинотеатра.

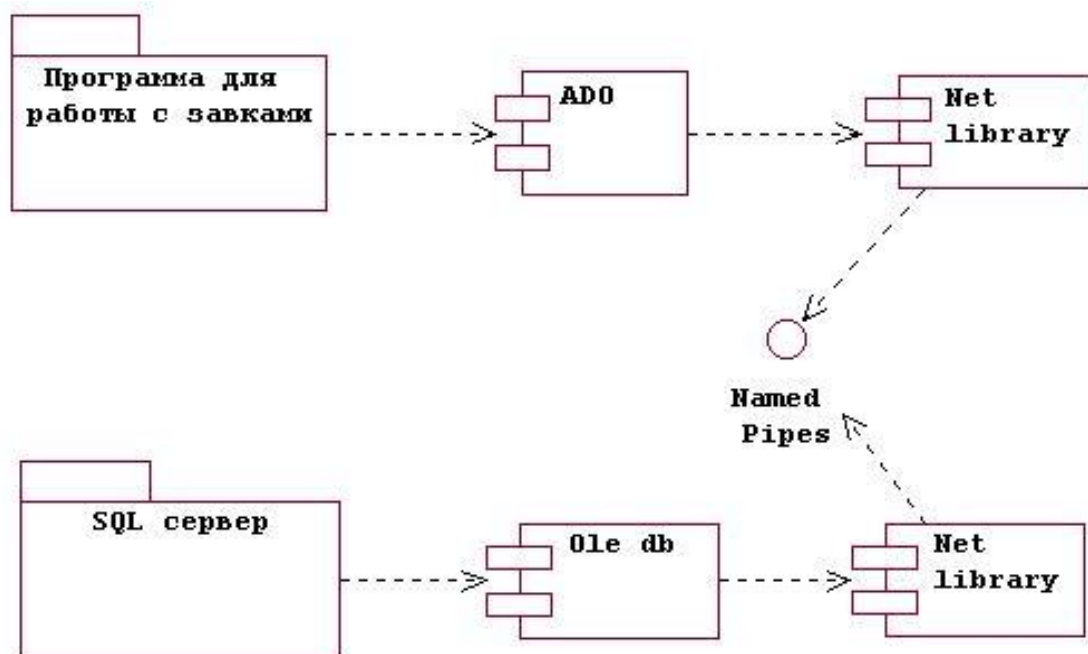


Рис. 28. Диаграмма компонент для описания взаимодействия приложений и SQL-сервера

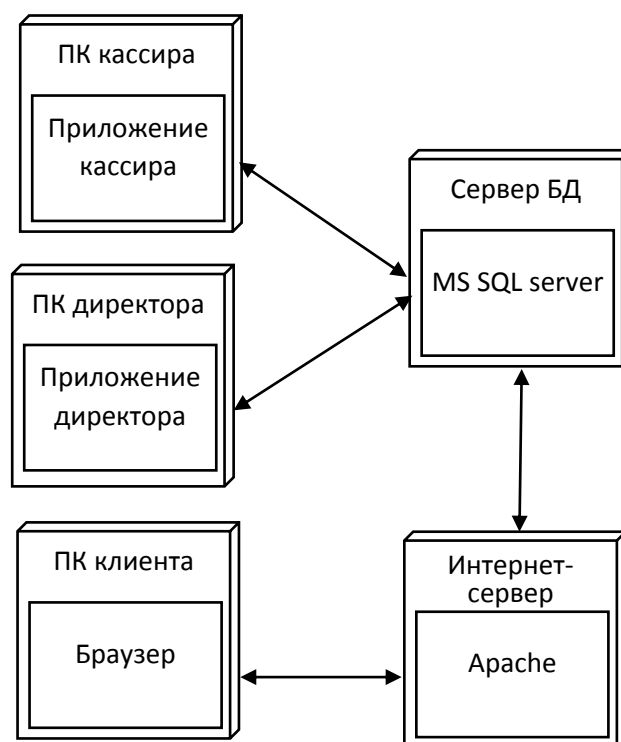


Рис. 29. Диаграмма развертывания для ИС оказания услуг демонстрации фильмов

4.4. Проектирование интерфейса пользователя, структуры и логики работы программы

Выделяют четыре подхода к проектированию [1]:

- функционально-ориентированное проектирование, которое использует структурные методы для построения функциональной, информационной и других моделей информационной системы;
- объектно-ориентированное проектирование [6] предлагает набор объектных моделей для описания предметной области. Для этого применяется специальный язык UML (Unified Modeling Language) [7];
- параметрически-ориентированное проектирование основано на применении готового программного обеспечения (например, конфигурации на платформе 1С:Предприятие, системы электронного документооборота, облачные сервисы), которое адаптируется к потребностям заказчика с помощью параметрической настройки, ввода нормативно-справочной информации, создания списка пользователей с определенными полномочиями и модификации и дополнения программного обеспечения;
- модельно-ориентированное проектирование основано на настройке и доработке типовой конфигурации информационной системы в среде специализированных инструментальных систем, таких как R/3, BAAN.

Выделение приложений выполняется на основе анализа событий, прав и обязанностей персонала. Для каждого приложения указывают назначение, персонал, использующий приложение, функции, автоматизируемые приложением, и регламент их выполнения, взаимодействие с серверами.

Для описания функций можно использовать стандартный документ «Описание алгоритма (проектной процедуры)» (см. прил.). Существует достаточно много средств описания алгоритмов:

- блок-схемы (диаграммы деятельности);
- деревья решений;
- диаграммы Nassi-Shneiderman;
- таблицы решений;
- структурный псевдокод;
- конечные автоматы (диаграммы состояний).

Какой бы вариант не был избран, важно, чтобы описание было точным и полным.

Проектирование интерфейса в современных инструментальных системах выполняется специальным графическим редактором, который наряду со схемой генерирует программу, создающую интерфейс. Поэтому достаточно ограничиться общепринятыми рекомендациями.

Рекомендации по проектированию входных и выходных форм:

1. Минимизируйте объем ввода:

- вводите только переменные данные, не вводите постоянные;
- не вводите значения, которые вычисляются;
- используйте подходящие коды (например, «м» и «ж» вместо «мужской» и «женский»).

2. Формы должны быть удобны для заполнения:
 - размещайте подсказки;
 - минимизируйте ручные действия (используйте переключатели, радиокнопки, поля с выпадающими списками).
3. Технология ввода должна быть простой: последовательность размещения полей в форме должна совпадать с последовательностью ввода.
4. Выходные формы должны быть просты для понимания и использования:
 - должен быть заголовок документа;
 - должны быть заголовки колонок;
 - должны быть пояснения к сокращениям и условным обозначениям;
 - не следует использовать жаргон.
5. Включайте в форму дату и время формирования.
6. При проектировании выходной формы необходимо учитывать ее использование.

Рекомендации по проектированию контрольных функций

1. Регистрируйте объемы введенных данных в журналах.
2. Контролируйте вводимые данные:
 - полноту (все ли поля заполнены?);
 - допустимые значения полей;
 - допустимые комбинации значений полей;
 - используйте шаблоны для уменьшения ошибок ввода;
 - применяйте контрольные суммы.
3. Совмещайте изготовление первичного документа и ввода данных.
4. Контролируйте время и объем операций вывода.
5. Контролируйте распределение выходных документов по пользователям.
6. Используйте контрольные суммы в выходных документах.

Рекомендации по проектированию интерфейса

1. Грамотно применяйте компоненты интерфейса:
 - пассивный и активный диалоги;
 - главные, дочерние и диалоговые окна;
 - командные кнопки,
 - меню;
 - блоки радиокнопок;
 - переключатели;
 - поля ввода;
 - поля со списками выбора.
2. Система должна подсказывать, что пользователь должен сделать дальше.
3. Поля ввода должны размещаться в ожидаемых пользователем зонах и единообразно.
4. Диалог должен быть ограничен окном.
5. Визуальные эффекты должны быть строго дозированы.

6. Умолчания должны быть размещены в полях.
7. Выводите предупреждения в случае опасных действий.
8. Предоставляйте возможность отмены изменений.
9. Используйте в подсказках простые правильные предложения.
10. Избегайте сокращений.
11. Последовательно используйте термины (например, используйте или термин «редактирование», или «изменение»).
12. Добивайтесь точности инструкций.

Часто не хватает времени на то, чтобы выполнить работу правильно. Однако время на переделку всегда находится.

Законы Мерфи

5. РЕАЛИЗАЦИЯ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

5.1. Разработка и внедрение

На стадии реализации создаются структуры данных, программы и документация к ним, инструкции по использованию программ, выполняется обучение пользователей и запускаются в эксплуатацию готовые программы. Реализация включает:

- создание и тестирование сети и базы данных;
- создание, тестирование и документирование отдельных программ;
- сбор, установка, тестирование и документирование системы в целом;
- ввод в эксплуатацию (разработка инструкций пользователям, обучение, консультирование).

Вследствие быстрого развития инструментальных средств разработки информационных систем сложно разделить этапы конструирования и реализации. Практически создание базы данных и приложений (в виде прототипов) уже выполняется на этапе конструирования по результатам разработки соответствующих описаний.

Стандартные документы стадии реализации приведены в приложении. Некоторые из них сохранили актуальность: «Описание системы (подсистемы)», «Описание организационной структуры», «Описание алгоритма (проектной процедуры)», «Руководство пользователя», организационно-распорядительные документы; другие – (например, документация по техническому обеспечению) редко разрабатываются в рамках подсистемы, так как описывают свойства системы в целом; третьи – (документы по информационному обеспечению, «Описание программы») достаточно сложно адаптировать, вследствие глубоких изменений технологий программирования.

Какой бы стандарт не был выбран разработчиками, он должен содержать:

- проектные решения:
 - описание архитектуры системы (подсистемы): приложения, серверы, регламенты решения задач;
 - описание информационного обеспечения: структуры данных, ограничения, бизнес-правила ведения и использования данных;
 - описание математического и программного обеспечений: математические модели, структуры программ (например, объектные и / или компонентные);
- эксплуатационные документы:
 - общий регламент управления в условиях автоматизации;
 - инструкции пользователям (порядок и правила использования информационных технологий);

- инструкции администраторам (управление полномочиями, развертывание информационного и программного обеспечений, поддержка работоспособности, безопасности и надежности системы, восстановление системы в случае аварий и сбоев);

- руководства программистам по сопровождению и развитию.

Ввод системы в опытную (промышленную) эксплуатацию обязательно должен быть оформлен юридически соответствующими документами (требования к актам о приемке в опытную (промышленную) эксплуатацию и к протоколам испытаний и согласований приведены в приложении). В соответствующие комиссии должны входить представители заказчика, исполнителя и ключевые пользователи.

Для обеспечения достоверности испытаний заказчиком должен быть составлен контрольный пример: варианты исходных данных и значения показателей, которые должны быть получены программным обеспечением по этим данным. Однако, заказчик зачастую пренебрегает этим правилом и испытание производится по контрольному примеру исполнителя.

В случае положительных результатов испытаний издается приказ о начале опытной эксплуатации, в котором должны быть определены сроки, пользователи и регламент использования информационной системы, а также состав и порядок работы соответствующей комиссии.

В период опытной эксплуатации выявляются и оформляются протоколом согласования все ошибки проектирования, в том числе и ошибки в постановке задачи. Согласованные ошибки исправляются. Возможно также дополнение системы новыми функциональными возможностями. Вносятся изменения в конструктивные и эксплуатационные документы.

Опытная эксплуатация заканчивается сдачей в промышленную эксплуатацию, которая оформляется соответствующими актом и приказом.

5.2. Обеспечение эксплуатации

Сопровождение заключается в поддержании в рабочем состоянии программного и информационного обеспечений. На этой стадии выполняются консультации и дополнительное обучение пользователей, восстановление после сбоев, исправление ошибок и адаптация программы к изменениям в условиях эксплуатации.

Исправление ошибок предполагает:

- определение проблемы;
- понимание программ;
- изменение и тестирование в сравнении с предыдущей версией;
- изменение документации.

Адаптация программы обычно включает:

- анализ требований;
- реструктуризация файлов и баз данных;
- анализ и переделка комплекса программ.

Развитие автоматизации предприятия проходит несколько уровней зрелости, зафиксированных в нормативных материалах COBIT¹ (Control Objectives for Information and Related Technologies – «Задачи управления для информационных и смежных технологий»).

Нулевой уровень – отсутствие автоматизации. Полное отсутствие каких-либо процессов управления ИТ. Организация не признает и даже не осознает существования проблем в ИТ, которые нужно решать, и, таким образом, нет никаких сведений о проблемах. На нулевом, самом нижнем уровне зрелости, речь об автоматизированной системе вообще не идет, системы управления ИТ как таковой не существует, а необходимость ее создания не осознается. Наблюдается полное отсутствие управляемых ИТ-процессов. Все ключевые ИТ-роли выполняются «незаменимыми» сотрудниками, часто являющимися «универсалами», мастерами на все руки. Общая стратегия развития ИТ отсутствует. Часто действия «универсалов» между собой не согласованы. При необоснованно завышенных расходах, например, на обеспечение информационной безопасности в такой организации совокупный результат, как правило, близок к нулю. Если зависимость предприятия от ИТ большая, то в такую организацию инвесторам не выгодно вкладывать деньги.

Первый уровень – начальный – автоматизация эпизодическая и бессистемная. Есть свидетельства того, что организация осознает существование проблем и необходимость их решения. При этом не существует никаких стандартизированных процессов. Существуют случайные одномоментные решения, принимаемые кем-то персонально или от случая к случаю. Организованный подход к управлению отсутствует. В организациях, находящихся на первом уровне зрелости, руководство начинает осознавать необходимость реализации комплексного подхода к управлению ИТ, что вызвано большой зависимостью этих организаций от ИТ и значительными расходами, которые не дают видимых результатов. На этом уровне еще не существует стандартизированных ИТ-процессов и преобладает фрагментарный подход к их реализации. Руководство только начинает задумываться о получении возврата инвестиций, сделанных в ИТ, не располагая, однако, методикой оценки их эффективности. Для решения каждой задачи ИТ-специалистами вырабатываются собственные подходы. Связь между бизнес-целями и деятельностью ИТ-департамента отсутствует. На этом уровне в настоящее время находятся многие российские предприятия, вкладывающие значительные средства в развитие и поддержание работоспособности своих информационных систем.

Второй уровень – повторяющийся, но интуитивный. Существует всеобщее осознание проблем управления ИТ. Процессы достигли уровня, при котором разные сотрудники, выполняющие одну и ту же задачу, используют сходные процедуры. Показатели деятельности и ИТ-процессов находятся в развитии, охватывая процессы планирования, функционирования и мониторинга ИТ. Деятельность по управлению информационными технологиями описана и интегри-

¹ URL: <https://ru.wikipedia.org/wiki/Cobit>.

рована в процесс управления организацией. Выбраны для улучшения и/или контроля те ИТ-процессы, которые влияют на основные бизнес-процессы предприятия. Руководство организации регламентировало меры по управлению ИТ, но процесс не был принят в организации. Не существует формализованного обучения. Организация сильно зависит от знаний отдельных лиц, вследствие чего велика вероятность ошибок. Ограниченные инструменты управления выбираются и внедряются для сбора метрик управления, но не используются в полном объеме из-за недостатков в оценке их функциональности.

На втором уровне *зрелости* уже существуют стандартизированные ИТ-процессы, однако они не документированы и пока не являются частью системы управления. Фактически они реализуются в виде стандартной практики отдельных ИТ-специалистов. На этом уровне необходимость планомерного внедрения системы управления ИТ уже ни у кого не вызывает сомнений, активно разрабатываются показатели эффективности ИТ-процессов, внедряются процессы планирования, мониторинга и предоставления ИТ-услуг, устанавливаются взаимосвязи между ИТ и бизнес-процессами, разрабатывается стратегия развития ИТ. Руководство организации принимает активное участие в формировании управляемых ИТ-процессов, для которых уже существуют базовые методы оценки эффективности. Однако сказывается недостаток опыта управления ИТ, используется ограниченное количество механизмов управления и показателей эффективности.

Третий уровень – определенный. Развивается базовый набор показателей управления ИТ: определена связь между результатом и показателями производительности, она зафиксирована и внедрена в стратегические процессы планирования и мониторинга. Процедуры стандартизованы и документированы, проводится обучение сотрудников по выполнению этих процедур. Показатели производительности всех видов деятельности зафиксированы и отслеживаются, что приводит к повышению эффективности работы всей организации. Процедуры не сложны, они являются формализацией существующей практики. Идеи сбалансированных карт оценки бизнеса принимаются организацией. Ответственность за обучение, выполнение и применение стандартов возложена на сотрудников организации. Анализ первопричин применяется время от времени. Большинство процессов управляются в соответствии с некоторыми основными метриками, и, как правило, отдельными сотрудниками, поэтому ни о каких отклонениях руководители не знают. Однако всеобщая отчетность о выполнении ключевых процессов является четкой, и руководство премирует сотрудников на основе измерения ключевых результатов.

Если на предыдущих уровнях *зрелости* преобладает администратор-универсал, то, начиная с третьего уровня, доминирующей становится роль системы управления. Здесь все процедуры стандартизированы и документированы, а сотрудники обучены их использованию. Деятельность ИТ-отдела регламентирована этими процедурами. Однако механизмы контроля качества выполнения процедур пока не работают, а сами процедуры далеки от совершенства, и об их оптимизации говорить не приходится.

Четвертый уровень – управляемый и измеряемый. Ведется мониторинг процессов в измеряемых показателях. Существует полное понимание проблем управления ИТ на всех уровнях организации, постоянно происходит обучение сотрудников. Определены и поддерживаются в актуальном состоянии соглашения об уровне обслуживания. Четко распределена ответственность, установлен уровень владения процессами. Процессы ИТ соответствуют бизнесу и стратегии ИТ. В первую очередь улучшения в процессах ИТ основываются на измеряемых количественных показателях. Существует возможность управлять процедурами и метриками процессов, измерять их соответствие. Все совладельцы процесса осознают риски, важность ИТ и возможности, которые они предоставляют. Руководство организации определило допустимые отклонения, при которых процессы должны работать. Если процессы не работают эффективно и продуктивно, действия предпринимаются во многих (но не всех случаях). Процессы постоянно совершенствуются, их результаты соответствуют «лучшим практикам». Формализован порядок анализа первопричин. Присутствует понимание необходимости постоянного совершенствования. Ограниченно применяются передовые технологии, основанные на современной инфраструктуре и модифицированных стандартных инструментах. Все необходимые ИТ-специалисты вовлечены в бизнес-процессы. Управление ИТ превращается в процесс уровня всей организации. Деятельность управления ИТ интегрируется в процесс управления организацией. Четвертый уровень зрелости характеризуется наличием системы контроля качества ИТ-процессов. Эта система осуществляет непрерывный мониторинг ИТ-процессов, устанавливает стандарты качества и контролирует соответствие ИТ-процессов данным стандартам. Наличие системы контроля качества позволяет выявлять неэффективно действующие механизмы управления ИТ-системой и постоянно работать над повышением их эффективности.

Пятый уровень – оптимизированный. В организации существует углубленное понимание управления ИТ, проблем и решений ИТ, а также перспектив. Обучение и коммуникация поддерживаются на должном уровне, самыми современными средствами. В результате непрерывного улучшения процессы соответствуют моделям зрелости, построенным на основании «лучшей практики». Внедрение этих процедур привело к появлению организаций, людей и процессов, максимально адаптируемых к изменяющимся условиям, а также полностью соответствующих требованиям управления ИТ. Первопричины всех проблем и отклонений тщательно анализируются, по результатам анализа выполняются результативные действия. Информационные технологии интегрированы в бизнес-процессы, полностью их автоматизируют, предоставляя возможность повышать качество и эффективность работы организации.

На высшем уровне система управления ИТ отличается от предыдущего по существу лишь более высоким уровнем оптимизации ИТ-процессов, которые являются управляемыми и измеряемыми. Информация о выполнении каждого ИТ-процесса фиксируется. ИТ являются эффективным инструментом бизнеса, а система управления ими – одной из составных частей системы управления организацией.

Информационная система организации на высоких уровнях развития обладает следующими характерными чертами:

- большое количество пользователей и приложений приводит к повышению сложности в результате увеличения функций, структур данных и их взаимосвязей;

- многообразие хранимых данных. Создание единой системы требует объединения различных, возможно непостоянных и противоречивых компонент;

- стандартизованные способы представления и обработки данных. Достижению этого мешают:

- разные представления данных;
- разные способы обработки данных;
- разная терминология;
- страх утраты монопольного распоряжения данными;

- тщательный контроль изменений – внедрение корпоративных подсистем начинает ограничивать пользователей в свободе манипулирования данными;

- значительное увеличение сложности и сроков разработки;
- изменение требований в процессе разработки;
- большой штат разработчиков;
- сложности в понимании системы в целом.

Потребность управления ИС требуется по следующим причинам:

- разные типы требуемых систем;
- конкуренция подразделений;
- внедрение подсистемы может повлиять на взаимодействие подразделений.

В задачи управления ИС входят следующие:

- внедрение эффективного использования информационных ресурсов, стимулирование изучения и применения достижений в области информационных технологий;

- обеспечение эффективного управления ресурсами ИТ, предоставление информации высшему руководству для инвестиционных решений;

- обеспечение взаимодействия пользователей и службы информационного обеспечения.

Организационно управление ИС предприятия в целом включает следующие направления деятельности:

- обеспечение обработки данных:

- *группа развития ИС* решает вопросы развития и поддержания ИС. Составляют группу системные аналитики и программисты, которые разрабатывают и развивают ИС при участии пользователей;

- *системные администраторы* занимается инсталляцией, настройкой, поддержкой системного программного обеспечения;

- *администратор данных* контролирует доступ к данным, отвечает за сохранность и безопасность данных, поддерживает ограничения целостности и непротиворечивости данных, восстанавливает данные при сбоях;

- обеспечение поддержки пользователей:

- *группа исследования и развития* изучает новые программные продукты;
- *группа снабжения* разрабатывает политику оснащения предприятия программным обеспечением и техническими средствами;
- *группа поддержки* выполняет консультации, обучение, в небольших количествах программирование (настройка, макросы и т.д.).

Сопровождение и развитие информационной системы средней или крупной организации является сложным процессом, управление которым должно учитывать интересы и потребности большого количества подразделений. Основной целью управления является соответствие информационной системы потребностям организации. Современным стандартом сопровождения информационных систем и технологий является ITIL¹ (IT Infrastructure Library) – библиотека инфраструктуры информационных технологий.

¹ URL: <https://ru.wikipedia.org/wiki/ITIL>.

6. ТИПОВЫЕ ПРОЕКТНЫЕ РЕШЕНИЯ

Типовое проектное решение (ТПР) – тиражируемый продукт (программное, информационное, организационное обеспечение, проектная документация), пригодный к многократному применению. Идея, методы и продукты ТПР появились практически одновременно с первыми разработками программного обеспечения. Причины появления этой методологии очевидны – это высокая трудоемкость разработки и легкость тиражирования. Однако, у нее есть и препятствия широкого распространения – это специфика управления однотипными бизнес-процессами и сложность адаптации готовых проектных решений к особенностям конкретного предприятия. Поэтому во все ТПР включают возможности настройки и программирования.

Реализация ТПР возможна на уровне:

- элементов;
- подсистем;
- объектов.

Проектирование на основе ТПР подразделяется на следующие направления:

- быстрая разработка программ;
- параметрически-ориентированное проектирование;
- модельно-ориентированное проектирование.

6.1. Быстрая разработка программ

Типовые решения на основе элементов широко применяют в современных инструментальных системах программирования. В них программист использует готовые элементы интерфейса, доступа к данным, формирования документов, построения графических изображений. К такого рода системам относятся языки программирования четвертого поколения (4GL), которые обладают следующими характеристиками:

- объектная структура;
- графический построитель интерфейса (генераторы меню, форм для ввода и отображения данных);
- генераторы отчетов;
- доступ к базам данных (компоненты для доступа к данным, графический построитель запросов).

Языки четвертого поколения лежат в основе быстрой разработки программ (Rapid Application Development – RAD). Характерными чертами этой методологии являются следующие:

1. Разработка приложения итерациями.
2. Обязательное вовлечение пользователей в процесс проектирования и построения системы.
3. Высокая параллельность работ.
4. Повторное использование частей проекта.
5. Необходимое применение CASE-средств.
6. Применение средств управления конфигурациями.

7. Использование автоматических генераторов (мастеров).

8. Использование прототипирования, позволяющего полнее выяснить и удовлетворить потребности конечного пользователя.

9. Тестирование и развитие проекта, осуществляемые одновременно с разработкой нескольких версий прототипа.

Современные инструментальные среды разработки обеспечивают быстрое построение программ на основе готовых классов, обеспечивающих нужную функциональность программного обеспечения: построение интерфейса, реализация взаимодействия с базами данных, типовые функции обработки данных в виде строителей отчетов и многомерной обработки данных. Полученный продукт обычно передается «как есть» без возможности модификации его заказчиком (если такая возможность не предусмотрена техническим заданием). Такая технология разработки применяется все реже, так как на рынке ПО предлагается все больше готовых программ для решения широкого спектра задач.

6.2. Параметрически-ориентированное проектирование

Продукты этого семейства ориентированы на удовлетворение требований заказчика с помощью большого количества параметров, начиная с атрибутов организации и должностных лиц (пользователей) и заканчивая различными справочниками. В настройку, кроме прочего, входят параметры, описывающие организационную структуру организации: пользователи и их функциональные полномочия. Как правило, такой настройки достаточно для обеспечения требуемой функциональности. Кроме этого, системы этого класса включают описание обработки данных при регистрации различных данных или для формирования выходных документов. Обычно для этого используется специальный макроязык, описывающий обработку данных различных объектов, таких как экранные формы, документы, отчеты. Программные модули хранятся в специализированной базе и определяют функциональные возможности системы. Регистрируемые данные накапливаются в базе данных и используются для формирования выходных документов. Настройка и функционирование такой системы представлены схемой на рис. 30.

Представителями параметрически-ориентированных систем являются различные конфигурации, построенные на платформе 1С: Предприятие.

Конфигурация в 1С: Предприятие включает следующие компоненты:

- константы;
- перечисления;
- календари;
- справочники (простые, многоуровневые, связанные);
- документы и операции по обработке документов;
- журналы документов;
- регистры – средства накопления сводной информации;
- выбор и обработка данных и формирование отчетов;

– модули (глобальные или связанные с конкретными объектами: документами, журналами, списками...), которые содержат: переменные; процедуры и функции.

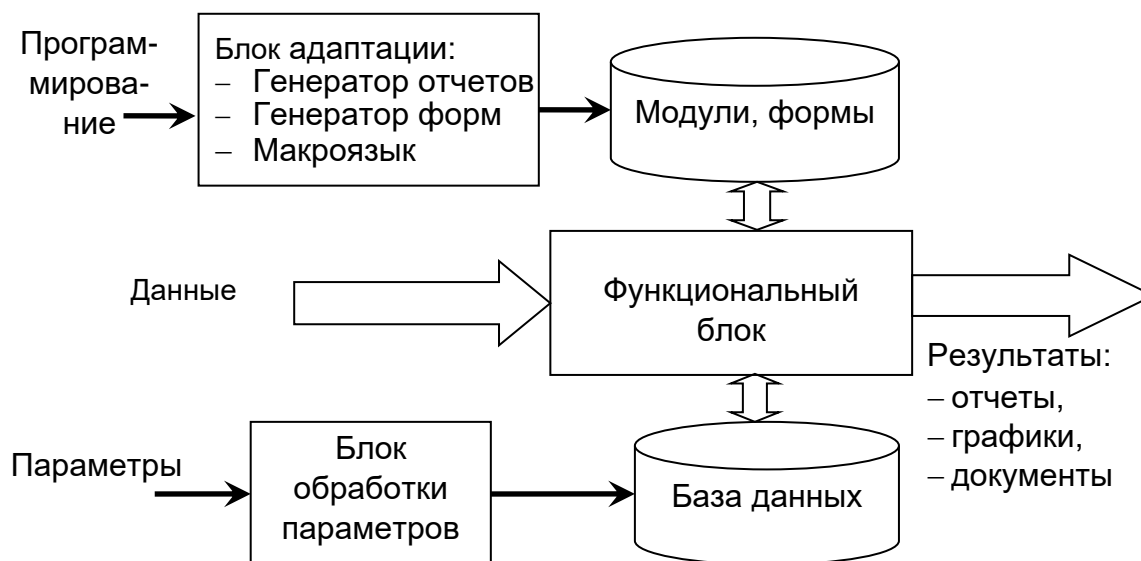


Рис. 30. Схема параметрически-ориентированного проектирования.

Конфигуратор является средством разработки и предоставляет следующие возможности:

- разработку программных модулей;
- настройку интерфейса и полномочий пользователей;
- отладку модулей;
- мониторинг действий пользователей.

База данных конфигурации содержит файлы конфигурации, параметры и учетные данные. Однако прямой доступ к таблицам БД выполняется через информационные объекты конфигурации (справочники, документы, регистры...).

Администрирование конфигурации выполняется регистрацией пользователей в системе и назначения им прав использования определенных интерфейсов. Войдя в систему, пользователь видит и использует «свои» функциональные возможности.

6.3. Модельно-ориентированное проектирование

Основой модельно-ориентированного подхода является постоянно развиваемая *модель проблемной области (предприятия)*, поддерживаемая в специальной базе метаинформации – *репозитории*. Для моделирования проблемной области используется специальный программный инструментарий, например SAP Business Engineering Workbench (BEW)¹ и BAAN² Enterprise Modeler.

¹ Зимненко И. Система SAP R/3. URL: <https://www.osp.ru/os/1998/02/179421>.

² Общая характеристика корпоративной системы BAAN IV. URL: <http://www.market-pages.ru/infteh/52.html>.

Методология BAAN¹ включает следующие модели:

1. ESM – Enterprise Structure Model – модель метаструктуры предприятия. Модель позволяет описать географически распределенную структуру компании, а также материальные и информационные потоки между ними.

2. BCM – Business Control Model – модель управления. Процессная модель описывает бизнес-процессы компании в стандарте диаграмм потоков данных (DFD²). Применяется для описания бизнес-процессов верхнего уровня

3. BPM – Business Process Model – процессная модель. Модель описывает бизнес-процессы компании в стандарте WFD³. Применяется для описания бизнес-процессов нижнего уровня.

4. BFM – Business Function Model – функциональная модель. Модель описывает функции, выполняемые в компании и их иерархию.

5. BOM – Business Organization Model – организационная модель. Модель описывает организационную структуру компании. BOM используется для описания подразделений и должностей организации, а также связей линейного и функционального подчинений. На данной модели также показываются роли, которые играет должность в тех или иных бизнес-процессах.

6. ERM – Entity-Relationship Model – информационная модель типа «сущность – связь». Описывает структуру информации, используемой при реализации бизнес-процессов. Позволяет описать структуру базы данных

Аналогичный набор моделей обеспечивает SAP EA Designer⁴. Это инструментальное средство поддерживает создание и редактирование следующих видов моделей:

1. Бизнес-процесс. Диаграммы бизнес-процессов помогают определить, описать и разложить бизнес-процессы. SAP EA Designer поддерживает и описательную BPMN 2.0, которая предоставляет небольшую подгруппу объектов, предназначенных для конструирования и анализа бизнес-процессов, и исполняемую BPMN 2.0, которая включает все стандартные объекты BPMN 2.0 и предназначена для разработчиков технических моделей.

2. База данных. Физическая модель данных позволяет анализировать и оптимизировать структуру базы данных. Можно перепроектировать любую поддерживаемую базу данных, чтобы создать физическую модель данных.

3. Диаграмма архитектуры предприятия. Диаграмма архитектуры предприятия позволяет анализировать и документально описывать организацию, ее функции и процессы, поддерживающие их приложения и системы, а также физическую архитектуру, с помощью которой они реализованы.

4. Карта процесса. Карта процесса дает графическое представление бизнес-архитектуры и помогает определить бизнес-функции и высокоуровневые процессы вне зависимости от выполняющих их людей и бизнес-единиц.

¹ Методология BAAN. URL: <http://sergeeva-i.narod.ru/inform/page505.htm>.

² URL: <https://ru.wikipedia.org/wiki/DFD>.

³ URL: <https://cesencom.ru/blog/metody-notacii-modelirovaniya-biznes-protsessov>.

⁴ Моделирование с помощью SAP Enterprise Architecture Designer. URL: https://help.sap.com/docs/SAP_HANA_PLATFORM/0e60f05842fd41078917822867220c78/49c9f5245a7d492fa35149be2733c8b6.html?version=2.0.00.

5. Список требований. В документах требований отображается иерархический список письменных требований.

Совокупность всех взаимосвязанных диаграмм с указанием свойств всех элементов образует модель предприятия и превращается в готовую информационную систему. Создано большое количество готовых моделей как отдельных бизнес-процессов, так и типовых предприятий для различных отраслей. Наличие готовых моделей существенно облегчает и ускоряет процесс разработки информационной системы. На основании анализа предприятия и определения требований к ИС подбирается типовая модель предприятия, в которую вносятся изменения в соответствии со спецификой предприятия. В результате получается готовая модель (рис. 31), которая определяет функционирование информационной системы. В такой системе учтено все: и технологии выполнения отдельных функций, и полномочия их выполнения, связанные с организационной структурой предприятия, и структуры хранения данных, и алгоритмы их обработки. В результате проектирование и программирование выглядят как построение графических моделей, соответствующих деятельности предприятия.



Рис. 31. Информационная система на основе модели предприятия

Модельно-ориентированный подход отличается полнотой охвата всех бизнес-процессов предприятия и значительной стоимостью инструментария разработки и полученной информационной системы.

Чтобы выносить ребенка, нужно девять месяцев, независимо от того, сколько женщин будет к этому привлечено.

7. ОРГАНИЗАЦИЯ ПРОЦЕССА РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Организация разработки и внедрения информационной системы включает следующее:

- управление основными характеристиками проекта: цели, назначение, автоматизируемые функции, свойства программного, информационного и других видов обеспечения;
- организация коллективной работы над проектом;
- управление процессом проектирования: выбор модели жизненного цикла (определение стадий и этапов разработки), оценка ресурсов (трудовых, временных, стоимостных), необходимых для выполнения работ на каждом этапе, оценка и управление рисками, оценка целесообразности продолжения работ в запланированном объеме, составление календарных графиков работ и отслеживание их выполнения.

7.1. Управление основными характеристиками проекта

Основными целями проекта являются следующие:

- удовлетворение требований заказчика;
- соблюдение ограничений проекта;
- соответствие спецификации (техническому заданию);
- выпуск только после выявления и устранения проблем;
- повышение эффективности труда пользователя;
- простота развертывания и организация сопровождения продукта.

Для успеха проекта принципиально важно соблюдать приоритет требований заказчика (системы управления). Для этого необходимо добиться, с одной стороны, понимания задач управления проектировщиками, с другой – понимания менеджерами стоимости информационных технологий и их значимости для эффективного решения задач управления. Привлечение проектировщиков к реинжинирингу бизнес-процессов, а менеджеров к принятию решений по информационным технологиям является ключом к созданию действительно полезной и нужной информационной системы.

Довольно часто пользователи под впечатлением «легкости» применения компьютерных программ не осознают сложности создания информационной системы, которая охватывает множество подразделений и объединяет усилия многих пользователей, имеющих зачастую противоречивые интересы. С другой стороны, менеджеры плохо представляют новые возможности, которые предоставляют информационные технологии, и поэтому склонны ограничиться автоматизацией тех функций, с которыми они справлялись раньше «вручную».

Наиболее частой ошибкой проектировщиков является построение системы на основании приблизительных представлений о целях, задачах и технологии управления.

Соблюдение требований к функциям и характеристикам – это задача менее творческая, чем постановка задачи или подгонка системы к интересам бизнеса. Она требует дисциплины и понимания со стороны коллектива разработчиков и постоянного внимания со стороны руководства.

На качестве проектирования положительно сказывается правильное разбиение проекта на итерации. В первую очередь должны быть автоматизированы ключевые операции управления. К сожалению, наиболее часто приходится начинать с рутинных учетных задач, которые обеспечивают накопление данных, нужных для решения задач управления. Необходимо, чтобы первые очереди системы действительно облегчали работу операционистов (ускоряли подготовку документов, автоматизировали подведение промежуточных итогов, уменьшали количество ошибок) и, кроме этого, позволяли извлекать информацию, полезную для управления.

7.2. Организация коллективной работы над проектом

Первоначально коллектив разработчиков строился по стандартной схеме единоначалия с элементами авторитарной власти. Распределение работ в коллективе можно выполнять по функциональному принципу (по подсистемам, комплексам задач), что требует разработчиков-универсалов, компетентных во всех направлениях проектирования. Так как, обычно, разработчик имеет определенную специализацию, то чаще работу распределяют по технологическому принципу (системные аналитики, постановщики задач, системные архитекторы, кодировщики, тестировщики).

Особенностью такой организации являются высокие требования, предъявляемые к руководителю разработки: он должен быть и политиком для достижения баланса интересов заказчика и разработчика, и менеджером для равномерного и плотного распределения работ, и архитектором для принятия важнейших системных решений, и достаточно компетентным во всех областях проектирования. Ясно, что сложно найти такого специалиста. Иерархической системе управления проектированием присущи следующие недостатки:

- Медленное согласование решений – проблема и методы ее решения согласуются в нескольких подразделениях и принимает окончательное решение руководитель проекта.
- Разработку замедляют проблемы организации сотрудничества разных подразделений, каждое из которых «тянет одеяло на себя».
- Сложно загрузить равномерно всех сотрудников.
- Низкая заинтересованность сотрудников в общем решении – каждый заинтересован в решении только своих специальных задач.
- Жесткая авторитарная система управления плохо подходит для объединения в единый коллектив высокоинтеллектуальных исполнителей.

Этой жесткой иерархической структуре нашли альтернативу в виде бригады с коллективной ответственностью за результат работы.

«Бригада главного программиста» включает руководителя работ, главного программиста, второго пилота, тестировщика, редактора, проблемных программистов. Руководитель работ осуществляет взаимодействие с пользователем, определяет цели и подцели, утверждает все проектные решения.

Проектирование ведется главным программистом и вторым пилотом. Такая система весьма эффективна, так как второй пилот постоянно подвергает критике решения главного программиста и испытывает прочность его концепций, а кроме того, предлагает свои.

Тестировщик оказался очень полезной фигурой и по важности сравнился с главным программистом. Именно тестировщик, и никто другой, лучше всех знает текущее состояние проекта, и с вопросами, насколько готов модуль, обращаются не к автору модуля, а к тестировщику. Критерием работы тестировщика является количество не пройденных модулем тестов – чем больше, тем лучше.

Редактор разрабатывает проектную документацию (оформляет идеи главного программиста и второго пилота) и документацию пользователя.

Такой модели присуща довольно высокая степень мобильности, однако демократизация остается по-прежнему низкой (поскольку главный программист единолично принимает все принципиальные решения по проекту). Вследствие этого проект очень сильно зависит от компетенции и личных качеств главного программиста.

В последнее десятилетие появилась модель команды-сообщества (community team), ярким примером которой является команда Microsoft [5] и методология разработки программных продуктов Microsoft Solutions Framework. Это наиболее демократичная модель, поскольку в ней нет явно выделенного центра. Команда-сообщество – это команда равных. В подобной команде взаимодействие участников проекта происходит на уровне сотрудничества. В группу MS Framework входят:

- **менеджер продукта** – представляет заказчика и его интересы, вместе с менеджером программы находит компромисс между функциями проекта, с одной стороны, и, сроками и бюджетом – с другой. У заказчика менеджер продукта представляет разработчиков, объясняя сроки и бюджет работ, достигает с ним компромисс о финансировании работ;

- **менеджер программы** – организует выполнение всех стадий и этапов разработки для выпуска нужного продукта в нужные сроки, координирует при этом деятельность других участников группы. Принимает и следит за выполнением графика работ (успешный способ составления графиков – «снизу вверх» на основе графиков исполнителей) и всего бюджета проекта. Отвечает за набор функциональных возможностей;

- **разработчики** выбирают применяемые технологии и создают продукт. Оценивают затраты на создание;

- **тестировщик** выполняет испытание продукта для оценки работоспособности отдельных функций и продукта в целом. Разрабатывает стратегию, планы, графики и сценарии тестирования;

– **инструктор** отвечает за выпуск удобного и полезного для пользователей продукта. Проектирует документацию пользователя, систему подсказок и, при необходимости, учебные курсы;

– **логистик** отвечает за доставку, поддержку и сопровождение продукта, составляет график развертывания, порядок передачи продукта, готовит продукт к развертыванию.

Допускается совмещать роли с учетом отсутствия противоположных интересов: роль разработчика не совмещается ни с одной другой, противоположные интересы у менеджера продукта и менеджера программы. Роль тестировщика не совместима с ролью разработчика.

К преимуществам данной модели можно отнести: высокую функциональную гибкость, высокую масштабируемость и производительность команды. Однако, на пути воплощения в жизнь подобной модели встают несколько существенных преград: подбор участников должен вестись не только исходя из компетенции, но и с учетом психологической и социокультурной совместимости, а также с учетом умения работать в команде (артельного духа); для формирования подобной команды нужны равные (одинаково квалифицированные и равно заинтересованные) участники.

7.3. Управление процессом проектирования

Управление процессом проектирования включает:

- выбор модели жизненного цикла (определение стадий и этапов разработки);
- оценку ресурсов (трудовых, временных, стоимостных), необходимых для выполнения работ на каждом этапе;
- оценку и управление рисками;
- составление календарных графиков работ и отслеживание их выполнения;
- оперативное управление процессом проектирования, включающее не только проверку выполнения и корректировку календарного графика, но и периодическую оценку целесообразности продолжения работ в запланированном объеме.

Выбор модели жизненного цикла основывается на масштабах проекта. Для проектирования экономических информационных систем или подсистем наиболее подходящими являются спиральные модели, предусматривающие достаточно короткие циклы и последовательное расширение набора автоматизируемых функций.

Основными ресурсами проектирования являются трудовые – время необходимое для выполнения каждой работы по проектированию. Оценка трудовых затрат наиболее сложная задача. Другие ресурсы: технические, программные, информационные, денежные – оценивать гораздо проще. А неоднократные попытки создать нормы работы проектировщиков и программистов не привели к надежным методикам. Практически все методики основаны на некоторых экспертных оценках. Поэтому чаще всего применяют экспертные

оценки трудоемкости самих исполнителей, добавляя к ним для страховки некоторое резервное время.

Оценка и управление рисками требуют, прежде всего, выделения и классификации возможных рисков. Обычно выделяют следующие риски:

- проектные риски при создании системы – риски, связанные с созданием системы в срок, с планируемыми затратами, с запланированными функциями и характеристиками;
- технические риски, состоящие в простоях, отказах, потере или искажении данных и т.п.;
- риски бизнес-потерь, связанные с эксплуатацией системы (возникающие, в конечном счете, из-за технических рисков). Такие риски бизнес-потерь назовем бизнес-рисками. Каждый бизнес-риск принадлежит, по крайней мере, одному из вариантов бизнес-использования (business use case) системы;
- риски бизнес-потерь, связанные с вариативностью бизнес-процессов.

При этом потери происходят по следующим причинам:

- бизнес-процессы надо изменять, а информационная система не готова к этому, и потери связаны с неоптимальным функционированием бизнеса;
- приходится платить за модификацию системы.

Для каждого вида рисков следует выполнить оценки вероятностей и продумать технологии компенсаций возможных негативных моментов, а также оценить затраты на подобные компенсации.

Для календарного планирования применяют методы сетевого планирования. Основой всего планирования является структурирование всего объема проектных работ до уровня отдельных проектных процедур и определение отношения предшествования: для каждой процедуры указывается, что должно быть выполнено до начала ее разработки. Для каждой процедуры устанавливается ее трудоемкость (обычно методом экспертной оценки разработчика процедуры) и необходимость в других ресурсах. На этой основе строится сетевая модель, в которой процедуры разработки представлены дугами, а вершины соответствуют завершенным работам.

По сетевой модели рассчитывается **критический путь** – путь на графе (последовательность смежных дуг без циклов), связывающих начальное и конечное состояния проекта, с максимальным суммарным временем выполнения, которое и определяет время выполнения проекта. Выделяют критические процедуры (лежащие на критическом пути), увеличение времени выполнения которых приводит к увеличению времени выполнения проекта. Для некритических процедур можно вычислить ранее и позднее время начала разработки, а также резерв времени как разность между поздним и ранним временами начала разработки. По критическому пути составляют графики выполнения работ, финансирования и потребления ресурсов, а также определяют контрольные точки для проверки выполнения графика работ.

Оперативное управление заключается в проверке выполнения календарного графика, анализе причин отставания от графика и, при необходимости, внесении корректив в сам график работы. Кроме этого, анализируется целесообразность выполнения работ в запланированном объеме.

Возможно применение более сложных моделей сетевого графика с вероятностными оценками длительностей работ или просто выделение резервного времени.

7.4. Экономические показатели проектирования

Календарный график работ является основой для определения финансовых, временных и других ресурсов. Можно выделить следующие виды расходов, связанных с информационной системой:

1. Расходы на разработку и внедрение:

- зарплата разработчиков;
- стоимость машинного времени;
- накладные расходы;
- материалы;
- приобретение вычислительной техники (ПК, серверы, сетевое оборудование);
- покупка программного обеспечения;
- обучение пользователей;
- накладные расходы (до 20 %), учитывающие затраты на управление разработкой ИС.

2. Расходы на эксплуатацию:

- трудовые затраты на сопровождение ИС (зарплата администраторов, группы сопровождения, операторов);
- амортизация оборудования;
- затраты на материалы (картриджи, диски, бумага...);
- расходы на ведение информационной базы;
- накладные расходы (до 20 %), учитывающие затраты на управление эксплуатацией ИС.

В определении ресурсов наиболее сложным является определение затрат труда. Общей практикой оценки трудозатрат является применение экспертных оценок исполнителей с выделением резервов от 30 до 50 %. Эти оценки используются для построения календарных графиков и определения затрат на оплату труда.

Источниками доходов от внедрения и модернизации информационных систем и технологий могут быть следующие:

- качественное улучшение процесса управления;
- снижение трудовых затрат (сокращение штатов);
- увеличение производительности труда;
- снижение стоимости инфраструктуры;
- уменьшение неопределенности и рисков;

– уменьшение совокупной стоимости владения (Total Cost of Ownership – TCO).

Для определения инвестиционной привлекательности проекта используют традиционные оценки. Для оценки эффективности проекта анализируются инвестиции (расходы) I и доходы $CF(i)$. При этом используют следующие показатели:

– чистая дисконтированная величина дохода (NPV – Net Present Value) – оценка дохода за некоторый период времени T , пересчитанная на момент вложения по процентной ставке γ

$$NPV = -I + \sum_{i=1}^T CF(i)/(1+\gamma)^i$$

– срок окупаемости проекта (BRP – Payback Period) – это время, за которое дисконтированная величина инвестиций сравняется с дисконтированной суммой доходов, находится как решение уравнения

$$NPV = -I + \sum_{i=1}^T CF(i)/(1+\gamma)^i = 0$$

относительно T ;

– индекс прибыльности (PI – Profitability Index)

$$PI = \frac{\sum_{i=1}^T CF(i)/(1+\gamma)^i}{I}$$

– внутренняя норма прибыльности (IRR – Internal Rate of Return) – это такая процентная ставка, при которой чистая приведенная величина дохода за все будущие периоды равна 0, т.е. когда приведенные доходы за все будущие периоды будут равны инвестициям. Находится как решение уравнения

$$-I + \sum_{i=1}^{\infty} CF(i)/(1+\gamma)^i = 0$$

относительно γ .

Методика оценки совокупной стоимости владения – ССВ (Total Cost of Ownership – TCO) была предложена Cartner Group в 1996 году. ССВ предназначена для полного учета ежегодных расходов предприятия (а не только его ИТ-отдела), связанных с приобретением и использованием информационных технологий в бизнесе. Все расходы делятся на следующие категории:

1. Прямые, т.е. бюджетлируемые затраты, которые включают в себя:

- капитальные затраты на приобретение оборудования и ПО;
- затраты на управление информационной службой (ИС);
- затраты на поддержку и разработку информационных систем;
- затраты на услуги аутсорсинга;

- затраты на закупку расходных материалов;
- обучение персонала и сотрудников ИС.

2. Косвенные, т.е. небюджетируемые затраты, которые включают в себя:

- потери от простоев пользователей – потери производительности вследствие запланированной и незапланированной недоступности системы;
- потери и затраты, связанные с самоподдержкой пользователей, т.е. с решением самими пользователями проблем, находящихся в компетенции ИТ-специалистов;

- потери и затраты, связанные с взаимоподдержкой пользователей.

ССВ ИТ-инфраструктуры складывается из суммы ССВ ее элементов:

- рабочие места пользователей (ССВ – затраты, связанные с использованием стандартного офисного рабочего места);
- распределенные информационные системы, (ССВ – затраты, связанные с эксплуатацией рабочих мест в системе, и затраты, связанные с использованием системы в целом – затраты на сервер системы, базу данных, простои системы в целом);
- совместно используемые инфраструктурные объекты – компьютерная сеть, источники бесперебойного питания, серверы, а также другое оборудование и программное обеспечение, обслуживающее группу рабочих мест и не связанное непосредственно с той или иной прикладной распределенной системой.

Расчет ССВ можно выполнить по следующей формуле:

$$CCB = A + N \cdot B,$$

где А – ССВ централизованного оборудования и ПО; В – ССВ одного рабочего места; N – число рабочих мест.

ССВ используется для решения следующих задач управления информационными системами предприятия:

- бенчмаркинг – сопоставление ССВ предприятия с ССВ сходных предприятий данной отрасли, страны и т.д.;
- оценка достаточности мощности службы технической поддержки. Проводится путем сопоставления затрат на техническую поддержку и потерь от оплачиваемых простоев пользователей;
- сопоставление альтернативных проектов информационных систем. Проводится путем сопоставления ССВ рассматриваемых систем;
- выбор между разработкой и закупкой (обычно в отношении программного обеспечения);
- принятие решения о полном или частичном аутсорсинге эксплуатации информационных систем, а также выбор поставщика услуг аутсорсинга;
- бюджетирование ИС.

Можно отметить следующие достоинства ССВ модели:

- подсчет текущих стоимостных параметров;
- оценка эффективности выполнения отдельных функций;
- наличие схемы учета и контроля расходов на информационные технологии.

В то же время в силу игнорирования доходов от эксплуатации информационных технологий ССВ присущи следующие недостатки:

- ССВ не учитывает риски;
- ССВ не позволяет соотнести ИТ со стратегическими целями.

Кроме оценки инвестиционной привлекательности и совокупной стоимости владения используют другие методологии оценки информационных технологий.

7.5. Гибкие методы проектирования информационных систем

Гибкая методология разработки¹ (*agile-разработка*) – обобщающее название для целого ряда подходов и практик [10]. Манифест agile-разработки декларирует следующие ценности:

1. **Люди и взаимодействие** важнее процессов и инструментов.
2. **Работающий продукт** важнее исчерпывающей документации.
3. **Сотрудничество с заказчиком** важнее согласования условий контракта.
4. **Готовность к изменениям** важнее следования первоначальному плану.

Таким образом, подход не отрицает важности того, что справа, считает более ценным то, что слева. Сформулированные ценности дополняются следующими принципами:

- наивысшим приоритетом является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке программного обеспечения;
- изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества;
- работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев;
- на протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе;
- над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им;
- непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды;
- работающий продукт – основной показатель прогресса;
- инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. Agile помогает наладить такой устойчивый процесс разработки;
- постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта;

¹ URL: [https://ru.wikipedia.org/wiki/Гибкая методология разработки](https://ru.wikipedia.org/wiki/Гибкая_методология_разработки).

- простота – искусство минимизации лишней работы – крайне необходима;
- самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд;

- команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

К гибким методологиям, в частности, относят:

- экстремальное программирование¹ применяет полезные традиционные методы и практики разработки программного обеспечения, подняв их на новый «экстремальный» уровень;

- метод разработки динамических систем (Dynamic Systems Development Method, DSDM)² – методика разработки программного обеспечения, основанная на концепции быстрой разработки приложений (Rapid Application Development, RAD) ;

- SCRUM (термин из регби, обозначает стартовое состояние команд перед вбросом мяча) – минимально необходимый набор мероприятий, артефактов, ролей, на которых строится процесс разработки, позволяющий за фиксированные небольшие промежутки времени, называемые спринтами (*sprints*), предоставлять конечному пользователю работающий продукт;

- Feature driven development³ (FDD, разработка, управляемая функциональностью) – итеративная методология разработки программного обеспечения – объединение наиболее признанных методик, принимающих за основу важную для заказчика функциональность (свойства) разрабатываемого программного обеспечения. Основной целью данной методологии является разработка реального, работающего программного обеспечения систематически, в поставленные сроки.

Кроме перечисленных к agile-методологиям относят и многие другие, следующие перечисленным выше принципам.

В гибких методах проектирования применяется организации труда небольших групп (которые делают однородную творческую работу) в объединении с управлением ими комбинированным (либеральным и демократическим) методом.

Большинство гибких методологий нацелены на минимизацию рисков путем сведения разработки к серии коротких циклов, называемых итерациями, которые обычно длятся две-три недели. Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для выдачи мини-прироста по функциональности: планирование, анализ требований, проектирование, программирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки.

¹ Экстремальное программирование – Википедия (wikipedia.org).

² DSDM – Википедия (wikipedia.org).

³ Feature driven development – Википедия (wikipedia.org).

Agile-методы делают упор на непосредственное общение лицом к лицу. Большинство agile-команд расположены в одном офисе. Как минимум, она включает и «заказчиков» (заказчик или его полномочный представитель, определяющий требования к продукту; эту роль может выполнять менеджер проекта, бизнес-аналитик или клиент). Офис может также включать тестировщиков, дизайнеров интерфейса, технических писателей и менеджеров.

Основным аргументом в пользу гибких методов проектирования является высокий уровень неопределенности процесса разработки информационных систем. В [10] приводится график (рис. 32) точности определения ресурсов, необходимых для выполнения проекта. На начальном этапе разброс оценивается в границах от -40% до $+60\%$. В процессе разработки этот интервал уменьшается до нуля в момент окончания работ.

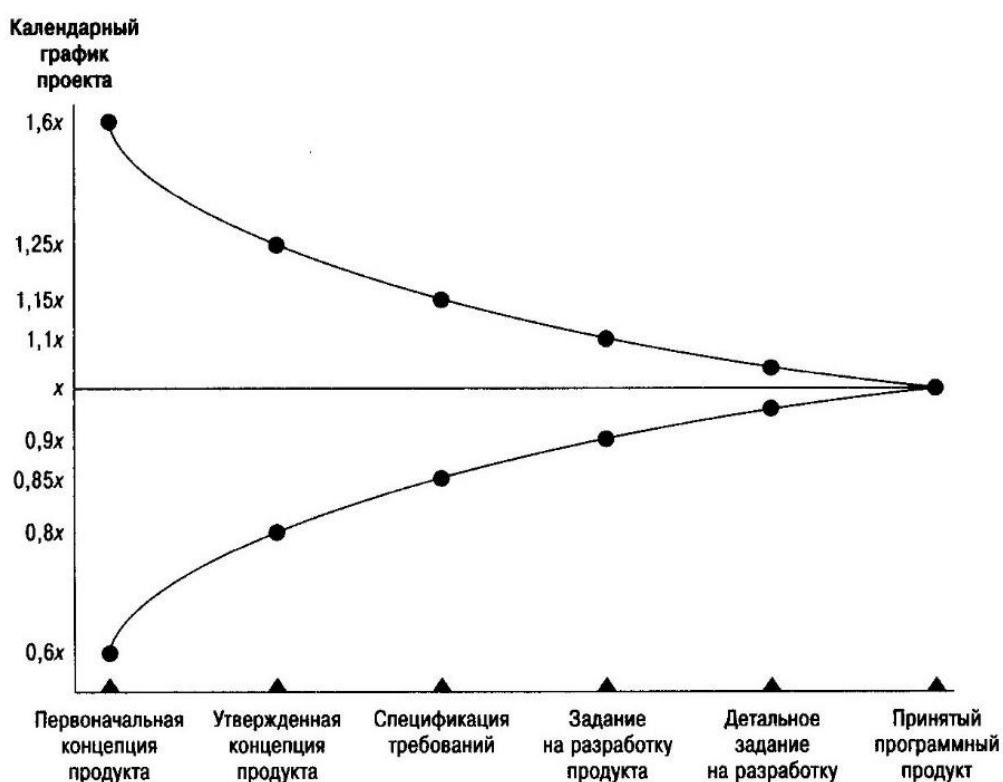


Рис. 32. Конус неопределенности проектирования ИС

Это означает, что заказчик должен быть готов нести издержки в указанных пределах. Для снижения рисков и понимания необходимости расходов для достижения нужных качеств проектируемой системы важно постоянное участие заказчика в процессе проектирования. Короткие итерации обеспечивают оперативный контроль процесса разработки со стороны заказчика и обоснование затрат на получаемую функциональность.

Одним из основных недостатков agile-подхода является отсутствие точного списка функций и требований к ИС. Гибкий подход к управлению требованиями не подразумевает далеко идущих планов (по сути, управления требованиями просто не существует в данной методологии), а подразумевает возможность

заказчика вдруг и неожиданно в конце каждой итерации выставляют новые требования, часто противоречащие архитектуре уже созданного и поставляемого продукта. Такое иногда приводит к катастрофическим «аврамам» с массовым рефакторингом и переделками практически на каждой очередной итерации.

Кроме того, считается, что работа в agile мотивирует разработчиков решать все поступившие задачи простейшим и быстрейшим возможным способом, при этом зачастую не обращая внимания на правильность кода с точки зрения требований нижележащей платформы (подход «работает – и ладно»), при этом не учитывается, что код может перестать работать при дальнейшем изменении. Это приводит к снижению качества продукта и накоплению дефектов.

Для исключения перечисленных недостатков целесообразно сочетать создание достаточно точной постановки задачи проектирования с гибкими методами проектирования, внося изменения общей постановки задачи и архитектуры системы в процессе разработки по согласованию с заказчиком.

Agile технология применяет командные методы управления. Это означает и коллективную ответственность, и коллективное принятие решений. Тем не менее в Agile-команде есть разделение на компетенции – роли. Список ролей может отличаться от проекта к проекту. Однако есть роли, общие для всех проектов:

- владелец продукта – это представитель интересов заказчика. Он отвечает за общее видение проекта, устанавливает приоритеты, контролирует рентабельность и полезность продукта, влияет на финансирование проекта. У заказчика он представляет интересы разработчиков, у разработчиков – интересы заказчика;

- пользователь отвечает за соответствие принимаемых решений и результатов проектирования потребностям персонала;

- разработчик непосредственно участвует в создании системы, может отвечать за определенную его часть, если в команде принято соответствующее разделение труда;

- руководитель проекта контролирует график работы, координирует деятельность по проектированию, отслеживает соответствие проекта требованиям, но не вмешивается в работу разработчиков.

Любое управление начинается с планирования. Agile подход устанавливает непрерывность планирования. Это очевидно, когда речь идет о коротких итерациях. Менее очевидным является наличие более высоких уровней планирования.

На вершине иерархии планов располагается «общее видение», которое описывает в общих чертах какую бизнес-проблему пытается решить проект. Это делается в первую очередь заказчиками проекта, но они должны работать в сотрудничестве с разработчиками для точного понимания возможностей и потребностей в ресурсах. Описание проекта на данном этапе содержит самые общие формулировки, например, «Продавать билеты на самолет через новый веб-сайт». Общее видение должно быть измеримым – следует определить показатели, которые будут достигнуты за счет реализованного проекта. Это могут быть денежные показатели (чистый дисконтированный доход, срок окупаемости), показатели снижения трудоемкости, показатели популярности созданного

ресурса – любые показатели, характеризующие полезность реализованного проекта для заказчиков.

Следующий шаг детализации – выделение «тем» – множества связанных функций, объединяющих несколько «пользовательских историй» – сценариев работы пользователей с информационной системой. Например, тема выбора рейса может включать несколько историй: прямой рейс, стыковочные рейсы, подбор с минимизацией стоимости, учет временных ограничений и т.д. Среди тем расставляются приоритеты в соответствии с пожеланиями заказчика.

Начальное планирование проекта не должно заходить дальше этих пределов, поскольку для понимания, что и каким образом должно быть разработано, нужно собрать команду (разработчиков, аналитиков, тестировщиков и т.д.). Это должно быть сделано на следующем этапе – планировании «релиза». На данном уровне полезно произвести оценку относительной сложности, трудоемкости и временных ограничений проекта.

Планирование релиза – это обязанность разработчика, но в сотрудничестве с заказчиком. Разработка плана релиза – это баланс потребности в новой функциональности и необходимости учитывать соответствующие технологические возможности, проблемы инфраструктуры, взаимозависимости и другие соображения.

Планирование релиза основывается на приоритетах функциональности. Часто это трудно сделать, поскольку каждая отдельная функция сама по себе важна. Лишь обсуждение сравнительной важности функций относительно друг друга поможет уточнить приоритеты. После того, как определены приоритеты в перечне функций, можно начать делить эти функции на истории. Истории – краткое описание функциональности с точки зрения пользователя. Что-то станет пользовательской историей, что-то, возможно, технической – например, потребности в новых технических инструментах, базах данных и т.д. Истории – это более детальная разбивка каждой функции на части, которые можно разработать за одну итерацию, таким образом, чтобы у каждой истории был отдельный результат.

Теперь, когда есть набор историй для первого релиза, снова нужно расставить приоритеты, чтобы знать, с чего начать и на чем сосредоточить усилия в первую очередь. Приоритеты определяются технологическими связями историй (например, анализ данных следует реализовать после функций их регистрации) и приоритетами заказчика.

Далее требуется оценить истории, и на этой основе определить сколько времени понадобится на весь объем работы по выпуску релиза. После этого можно начать обсуждение, не удалить ли какой-то функционал, если это время слишком велико, либо поменять ожидания относительно релиза.

Работа над релизом делится на итерации. Каждая итерация заключается в разработке и сдаче заказчику функций для пользователя на основании пользовательской истории. Формирование итерации основано на оценке сроков выполнения. Одним из методов такой оценки является измерение трудоемкости в «пунктах». Пункт – это абстрактная единица, применяемая для сравнения историй. Минимальная «пользовательская история» – 1 пункт, средняя – 5. Реализация

итерации за какой-то период позволяет более-менее точно сопоставить пункты и сроки выполнения.

Другой единицей измерения является «идеальный день» – чистое время, затраченное на проектирование, без ожиданий, собраний и других отвлечений. По разным оценкам реальный день работы соответствует 50–70 % идеального дня. Независимо от выбранной единицы измерения оценивается история целиком, а не затраты отдельного исполнителя.

Оценка сроков выполнения может быть выполнена следующими методами:

- экспертная оценка (оценивается срок выполнения пользовательской истории всеми разработчиками);
- оценка по аналогии (относительно разных уже реализованных историй с усреднением полученных оценок);
- декомпозиция на части с оценкой трудоемкости по частям.

Каждый метод обладает своими достоинствами и ограничениями. Универсальным является экспертная оценка. М. Кон для ее выполнения предлагает «покер планирования» [10]:

1. Каждый эксперт получает карточки со сроками 0, 1, 2, 3, 5, 8, 13, 20, 40, 100.
2. Каждый эксперт выполняет оценивание. Оценки экспертов предъявляются синхронно.
3. Эксперты объясняют отклонения от средней оценки.
4. В случае большого разброса выполняется повторение оценки.

Сроки итерации определяются с учетом совещаний, документирования, тестирования, устранения ошибок. Процесс разработки постоянно контролируется. Фиксируется прогресс проектирования. Изменения объема работ приводят к пересмотру всех планов. Часть функций откладывается из-за временных ограничений или по причинам необходимой переделки уже реализованных функций и образует «незавершенку». Обо всех таких обстоятельствах обязательно информируется заказчик.

Инструментом оперативного проекта являются Agile-доски – инструмент для совместной работы, в его основе лежат принципы Канбан-методологии. Канбан («рекламный щит, вывеска») является методом управления разработкой. Задачи по мере поступления заносятся на доску, откуда каждый разработчик может извлечь требуемую задачу. Технология применения Agile-доски включает следующие шаги:

1. **Определите цель** – какой процесс (процессы) вам важно отслеживать, так вы разграничите рабочие пространства:

- создайте доску для направления или продукта, отдела, команды или персонального пользования;
- дайте доске имя, соответствующее назначению.

2. **Определите участников.** Для защиты информации и порядка в процессах ограничьте круг лиц, имеющих доступ к рабочему пространству команды. В зависимости от того, будет ли пользоваться доской один сотрудник, команда или отдел наделите всех участвующих сотрудников правами доступа.

3. Определите этапы процесса, соответствующие колонкам (столбцам) доски. Начать пользоваться досками можно со стандартных этапов:

- «новые» для размещения новых задач;
- «в работе» для задач, распределенных исполнителям;
- «выполнено» для реализованных задач.

Можно переименовывать колонки, добавлять новые и удалять лишние, менять их места.

4. Зафиксируйте задачи, это быстро войдет в привычку и позволит забыть про досадные неприятности вроде потерянных стикеров и идей:

- в электронных досках аналоги бумажных наклеек называются тикетами, их можно создавать в колонках: один тикет – одна задача;
- крупные задачи разбивайте на несколько менее масштабных, так исполнитель быстрее включится в работу и «съест слона по кусочку»;
- название тикета должно отражать суть задачи, быть кратким и емким; детали укажите в описании задачи;
- если доска создана для команды или отдела – назначьте тикетам исполнителей;
- если исполнитель не определен – вы можете вернуться к этому пункту позже, перед тем, как задача будет передана в работу;
- определите срок, к которому должна быть выполнена та или иная задача, укажите их в тикетах.

5. Расставьте приоритеты, так исполнители выстроят для себя очередность задач и эффективно распределят рабочее время:

- выделите наиболее важные задачи;
- перетащите их в верхнюю часть «ожидающей» колонки, менее важные и менее срочные будут располагаться внизу;
- если решение позволяет, назначьте приоритеты прямо в тикетах: установите более высокий для важных задач.

6. Сохраните полезную для выполнения задач информацию. Благодаря этому исполнитель максимально быстро погрузится в работу.

В некоторых решениях к тикетам можно прикреплять ссылки, например, на стандарт предприятия, инструкцию, папку, любой другой информационный ресурс, который пригодится во время выполнения задачи сотрудником.

7. Раздайте задачи исполнителям. Договоритесь о правилах взаимодействия:

- если вы руководите работой команды или отдела – расскажите сотрудникам, как устроена Agile-доска;
- договоритесь, кто может фиксировать новые задачи, назначать исполнителей, перемещать тикеты, а главное – как исполнитель должен узнать о появлении новых задач;
- в командах с менее формальным подходом задачи могут распределяться во время коротких онлайн-совещаний, фиксироваться на доске и перемещаться по колонкам самими исполнителями.

Как бы вы не построили процесс, и какие бы не установили правила взаимодействия, главным остается то, что доска – это единый «источник правды» в

команде. Надо сделать – запиши. Сделал – отметь. Не идет работа – важно вовремя выяснить почему и принять меры.

8. Отслеживайте текущее состояние работ и загрузку сотрудников для определения «узких мест» и эффективного распределения ресурсов:

- для оценки работы в том или ином разрезе можно использовать цветные метки (теги), фильтры;
- метками (тегами) может отмечаться все, что дополнительно характеризует особенности ваших процессов;
- фильтры помогают оценить текущее состояние работ по проекту, направлению или цели;
- фильтры по исполнителям в совокупности со статусами работ позволят оценить текущую загрузку сотрудника: не злоупотребляйте «многозадачностью», ресурсы и фокус внимания каждого человека ограничены.

В некоторых инструментах Agile сотрудники могут оставлять комментарии в тикетах. Так они остаются в курсе состояния задачи, не открывая документов.

Многие решения предлагают дополнительный функционал, позволяющий оценивать ход процесса. Как правило, эффективность определяется скоростью прохождения тикета по колонкам от старта до финиша.

9. Перемещайте завершенные работы в колонку «Выполнено». Этот пункт особенно важный для сотрудника с психологической точки зрения – фиксировать результаты и ставить точки в бесконечном потоке задач.

Agile-доски успели зарекомендовать себя как простой и эффективный инструмент для планирования и отслеживания состояния работ в едином пространстве команды или проекта, позволяющий удерживать фокус на приоритетных задачах, ограничить «многозадачность» сотрудников рамками здравого смысла, выявить «узкие места» в процессах и эффективно управлять ими.

Электронные Agile-доски дополнительно предоставляют ряд преимуществ в виде удаленного доступа, возможности прикрепления к тикетам файлов и пояснений, формирования различных отчетов и показателей выполнения проектов. Можно воспользоваться онлайн-досками, например Trello¹.

Agile-технология проектирования не меняет логики проектирования: постановка задачи – проектные решения, но делает этот процесс максимально гибким. Мелкие итерации значительно снижают риски проектирования, однако неопределенность технологии требует особых взаимоотношений заказчика и разработчика – фактически они должны стать единомышленниками, образовать команду, учитывать интересы друг друга для достижения общего результата – создания информационной системы, нужной заказчику и удовлетворяющей профессиональным амбициям разработчика.

¹ URL: <https://trello.com/ru>.

ЗАКЛЮЧЕНИЕ

Проектирование информационных систем – это сложный творческий процесс. Большое количество методик проектирования только подтверждает это. Выбор методики зависит от опыта работы и навыков команды проектировщиков. Нет сомнений, что новые методы проектирования соответствуют новым требованиям и технологиям, а традиционные методы проектирования отражают накопленный опыт. В целом между «старыми» и «новыми» методами нет существенных противоречий. Все методики начинаются с постановки задачи и проработки архитектуры системы. Следует отметить всё более возрастающую популярность различных диаграмм для описания проектных решений. Языки диаграмм применяются не только для иллюстрации конструкций компонент ИС. Они применяются для генерации соответствующих компонент. Эти возможности убедительно демонстрируют и CASE-средства, и системы модельного проектирования.

Наиболее значимыми проблемами в проектировании информационных систем остаются проблемы взаимопонимания пользователей и программистов, организации последовательности проектных работ, управление взаимоотношениями в команде разработчиков, документирование проектных решений. Для решения этих проблем созданы и создаются компьютерные инструменты их решения:

- управление проектами;
- календарное планирование и управление;
- ведение совместной документации и поддержание версий;
- доски для управления работами.

Сама технология программирования включает все больше готовых систем и объектов, из которых собирается решение, нужное заказчику. Все больше появляется «No-code-платформ», которые позволяют создавать сайты, приложения, аналитические процедуры без кодирования на алгоритмическом языке.

Общей тенденцией в проектировании ИС является отказ от жестких решений: команды разработчиков строят в виде бригады равноправных участников с разными ролями, последовательность работ адаптируется к пожеланиям и приоритетам пользователей.

ПРИЛОЖЕНИЕ

Требования стандартов к содержанию проектных документов

Пояснительная записка к эскизному (техническому) проекту

1. Общие положения:

- наименование;
- разработчики;
- цели, назначение и области использования системы;
- соответствие нормам техники безопасности и нормативным документам;
- сведения о НИР, изобретениях, открытиях, использованных при разработке;
- очередность создания системы.

2. Описание процесса деятельности (состав процедур, требования к организации работ).

3. Основные технические решения:

- структура системы, связь между структурными единицами;
- связь со смежными системами;
- функционирование и диагностирование;
- состав, квалификация, функции, режим работы персонала системы;
- состав функций, комплексов задач (задач), реализуемых системой (подсистемой);
- комплекс технических средств;
- состав, объем, способы организации информации, виды носителей информации, входные и выходные документы и сообщения, последовательность обработки;
- состав программных средств, языки, алгоритмы и методы их реализации.

4. Мероприятия по подготовке объекта автоматизации к вводу системы в действие:

- по приведению информации к виду, пригодному для обработки на ЭВМ;
- по обучению и проверке квалификации персонала;
- по созданию необходимых подразделений и рабочих мест;
- по изменению объекта автоматизации.

Общее описание системы

1. Назначение системы:

- вид автоматизируемой деятельности;
- перечень объектов автоматизации;
- перечень функций, реализованных системой.

2. Описание системы:

- структура системы и назначение ее частей;
- сведения о системе в целом и ее частях, необходимых для обеспечения эксплуатации системы;

- описание функционирования системы и ее частей.

3. Связи с другими системами:

- перечень систем;
- описание связей;
- регламент связей;
- связи с подразделениями объекта автоматизации.

4. Описание подсистем (выполняется при необходимости и подобно описанию системы).

Описание организационной структуры

1. Изменения в организационной структуре системы управления: проектные решения по изменению и их обоснования; изменение во взаимосвязях.

2. Организация подразделений: описание организационной структуры и функций создаваемых подразделений; описание регламента работ; категории работников и их количество.

3. Реорганизация подразделений.

Схема организационной структуры

1. Состав подразделений, обеспечивающих функционирование системы или использующих информацию из системы.

2. Основные функции и связи, а также их подчиненность.

Схема функциональной структуры

1. Элементы функциональной структуры (функции и/или задачи (комплексы задач)); действия, выполняемые автоматически и вручную.

2. Информационные связи между элементами и с внешней средой с указанием содержания сообщений, связи других типов (вложенности, подчинения и др.).

3. Детализация (при необходимости).

Описание автоматизируемых функций

1. Исходные данные:

- перечень исходных материалов и документов для разработки функциональной части системы;
- особенности объекта автоматизации, влияющие на проектные решения по автоматизируемой функции;
- сведения о системах, связанных с разрабатываемой системой, и об информации, которой они обмениваются;
- информационная модель объекта.

2. Цели системы и автоматизированные функции (описание функций).

3. Характеристики функциональной структуры:

- перечень подсистем системы с указанием функций (задач);
- описание процесса выполнения функции (при необходимости);
- пояснения к разделению функции на действия, выполняемые техническими средствами и человеком.

4. Типовые решения (при наличии).

Описание постановки задачи (комплекса задач)

1. Характеристики комплекса задач:

- назначение;
- перечень объектов, при управлении которыми решается данный комплекс;
- периодичность и продолжительность решения;
- условия прекращения использования (при необходимости);
- связи с другими комплексами;
- распределение действий между персоналом и техническими средствами.
- выходная информация – перечень и описание выходных сообщений:
 - идентификатор;
 - форма (документ, видеокадр, файл, ...);
 - периодичность выдачи;
 - получатели и назначение;
 - перечень и описание структурных единиц сообщений, имеющих самостоятельное значение,
- входная информация – перечень и описание входных сообщений:
 - наименование;
 - точность (при необходимости);
 - источник (документ, устройство, кодограмма, файл...);
 - идентификатор.

Описание технологического процесса обработки данных

1. Технологический процесс сбора и обработки данных на периферийных устройствах при децентрализованной обработке данных:

- состав и последовательность выполнения операций по сбору, регистрации, подготовке, контролю, передаче, обработке и отображению информации;
- перечень документации на каждую операцию.

2. Технологический процесс обработки данных на ВЦ:

- состав и последовательность выполнения операций по приему, контролю, обработке, хранению, выдаче данных и других;
- перечень документации на каждую операцию.

Схема автоматизации

1. Упрощенное изображение объекта или его части.
2. Средства технического обеспечения, участвующие в процессе, отображенном на схеме, за исключением вспомогательных.
3. Функциональные связи между средствами технического обеспечения.
4. Внешние функциональные связи с другими средствами технического обеспечения.
5. Таблица условных обозначений, не предусмотренных стандартами.

Описание комплекса технических средств

1. Общие положения (исходные данные для проектирования).
2. Структура комплекса технических средств:
 - обоснование выбора структуры комплекса технических средств;
 - описание функционирования комплекса технических средств, в том числе в пусковых и аварийных режимах;
 - описание размещения технических средств;
 - обоснование применения и требования к изготавливаемым в индивидуальном порядке технических средств;
 - обоснование методов защиты технических средств, в том числе от несанкционированного доступа;
 - оценка надежности технических средств.
3. Средства вычислительной техники:
 - обоснование и описание решений по выбору ЭВМ,
 - периферийные устройства;
 - структурная схема технических средств на ВЦ и рабочих местах;
 - расчет числа технических средств и потребностей в машинных носителях;
 - обоснование численности персонала, обслуживающего технические средства;
 - оснащение рабочих мест персонала, включая описание рабочих мест и расчет площадей;
 - особенности функционирования технических средств в пусковом, нормальном и аварийном режимах.
4. Аппаратура передачи данных:
 - выбор каналов связи и расчет их числа, технических средств сопряжения с каналами связи;
 - размещение абонентов и объемно-временные характеристики передачи данных;
 - надежность, достоверность и другие характеристики.

Описание информационного обеспечения

1. Состав информационного обеспечения (наименование и назначение всех баз и наборов данных).
2. Организация информационного обеспечения:
 - принципы организации информационного обеспечения;
 - обоснование выбора носителей и принципы распределения данных по типам носителей;
 - виды и методы контроля;
 - решения по обеспечению информационной совместимости с другими системами.
3. Организация сбора и передачи данных:

- источники и носители информации с указанием интенсивностей и объемов;

- общие требования к сбору, передаче, контролю и корректировке.

4. Построение системы классификации и кодирования.

5. Организация внутримашинной информационной базы:

- принципы построения, характеристики состава и объема;

- описание структуры на уровне баз данных, с описанием характера взаимосвязей баз данных и указанием функций системы, при реализации которых используют каждую базу данных, характеристики данных, содержащихся в каждой базе данных.

6. Организация немашинной информационной базы: состав, объем, принципы построения, связи с функциями системы.

Описание организации информационной базы

1. Описание внутримашинной информационной базы: логическая структура (состав, форматы, взаимосвязи); перечни БД и взаимосвязи между ними; физическая структура (расположение данных на носителях).

2. Описание немашинной информационной базы: логическая структура (состав, форматы, взаимосвязи); перечень документов и сообщений с указанием функций системы, при реализации которой используют документ или сообщение; физическая структура (расположение данных на носителях).

3. Организация ведения информационной базы: для внутримашинной базы – последовательность процедур с регламентом, средствами защиты от разрушения и несанкционированного доступа для немашинной базы – последовательность процедур по маршруту движения документов.

Описание массивов информации

1. Наименование.

2. Обозначение.

3. Носитель.

4. Перечень реквизитов в порядке следования в записях с указанием: обозначения, длины и диапазона изменения, логических и семантических связей.

5. Оценка объема.

6. Другие характеристики.

Описание алгоритма (проектной процедуры)

1. Назначение и характеристика:

- назначение;

- ссылка на документ с постановкой задачи, для решения которой он предназначен;

- сведения об объекте и влияние решения на объект;

- ограничения, условия применения, качество решения;

- требования к данным.

2. Используемая информация (массивы).

3. Результаты решения (массивы).
4. Математическое описание.
5. Алгоритм решения:
 - последовательность этапов счета;
 - точность;
 - контроль достоверности;
 - связи между частями и операциями;
 - порядок расположения значений в выходных документах.

Алгоритм должен предусматривать все ситуации. Следует использовать обозначения таблиц, файлов, реквизитов, документов, и.т.д.

Акт завершения работы

1. Наименование завершенной работы.
2. Список представителей разработчика и заказчика, составивших акт.
3. Дата завершения работы.
4. Наименование документов, на основании которых производилась работа.
5. Основные результаты заверченной работы.
6. Заключение о результатах заверченной работы.

Акт о приемке в опытную эксплуатацию

1. Наименование АС или ее части, принимаемой в опытную эксплуатацию, и соответствующего объекта автоматизации.
2. Наименование документа, на основании которого разработана АС.
3. Состав приемочной комиссии и основания для ее работы.
4. Период времени работы комиссии.
5. Наименования разработчика и исполнителя.
6. Состав функций, принимаемых в опытную эксплуатацию.
7. Перечень составляющих технического, программного, информационного и организационного обеспечений, принимаемых в опытную эксплуатацию.
8. Перечень документов, предъявляемых комиссии.
9. Оценка соответствия принимаемой АС техническому заданию.
10. Основные результаты приемки в опытную эксплуатацию.
11. Решение комиссии о принятии АС в опытную эксплуатацию.

Акт о приемке в промышленную эксплуатацию

1. Наименование АС или ее части, принимаемой в промышленную эксплуатацию, и соответствующего объекта автоматизации.
2. Наименование документа, на основании которого разработана АС.
3. Состав приемочной комиссии и основания для ее работы.
4. Период времени работы комиссии.
5. Наименования разработчика и исполнителя.
6. Состав функций, принимаемых в промышленную эксплуатацию.

7. Перечень составляющих технического, программного, информационного и организационного обеспечений, принимаемых в промышленную эксплуатацию.

8. Перечень документов, предъявляемых комиссии.

9. Заключение о результатах опытной эксплуатации.

10. Оценка соответствия принимаемой АС техническому заданию.

11. Краткая характеристика и основные результаты выполненной работы по созданию АС.

12. Оценка научно-технического уровня АС

13. Оценка экономической эффективности АС.

14. Решение комиссии о принятии АС в эксплуатацию.

15. Рекомендации комиссии по дальнейшему развитию системы.

Протокол испытаний

1. Наименование объекта испытаний.

2. Список должностных лиц, проводивших испытание.

3. Цель испытания.

4. Сведения о продолжительности испытания.

5. Перечень пунктов ТЗ, на соответствие которым проводились испытания.

6. Сведения о результатах наблюдений.

7. Сведения об отказах, сбоях и аварийных ситуациях, возникших в процессе испытаний.

Протокол согласования

1. Перечень рассмотренных отклонений с указанием документа, устанавливающего требования.

2. Список должностных лиц, составивших протокол.

3. Обоснование принятых отклонений.

4. Перечень согласованных отклонений и срок внесения необходимых изменений в техническую документацию.

СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Смирнова Т.Н. Проектирование экономических информационных систем : учебник / Т.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов. – Москва : Финансы и статистика, 2001.
2. Автоматизированные информационные технологии в экономике : учебник / под ред. Г.А. Титоренко. – Москва : Компьютер, ЮНИТИ, 1998.
3. Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы / Ф. Брукс. – Санкт-Петербург : Символ-Плюс, 1999.
4. Мишенин А.И. Теория экономических информационных систем / А.И. Мишенин. – Москва : Финансы и статистика, 2002.
5. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения : учеб. курс MCSD. – Москва : Изд.-торг. дом «Русская редакция», 2000.
6. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч. – Москва : ЗАО Изд-во «Бином» ; Санкт-Петербург : Невский диалект, 1999.
7. Буч Г. Язык UML. Руководство пользователя / Г. Буч, А. Джекобсон, Д. Рамбо. – Москва : ДМК, 2000.
8. Калянов Г.Н. CASE-технологии: Консалтинг в автоматизации бизнес-процессов / Г.Н. Калянов. – Москва : Горячая линия – Телеком, 2000.
9. Маклаков С.В. BPWin и ERWin. CASE-средства разработки информационных систем / С.В. Маклаков. – Москва : Диалог-МИФИ, 1999.
10. Кон М. Agile: оценка и планирование проектов / М. Кон. – Москва : Альпина Диджитал, 2006. – Т. 91. – 417 с. (Библиотека Сбербанка).
11. Информационная технология: Комплекс стандартов и руководящих элементов на автоматизированные системы. – Москва, 1991.
12. Дейт К.Дж. Введение в системы баз данных / К.Дж. Дейт. – 6-е изд. – Москва : Вильямс, 2000.
13. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям : учеб. пособие / Н. Паклин, В. Орешков. – 2-е изд., испр. – Санкт-Петербург : Питер, 2013. – 701 с.
14. Whitten J.L. System analysis and design methods / J.L. Whitten. – 3rd ed. – IRWIN, 1994.
15. Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов / В.Г. Елиферов., В.В. Репин. – Москва : РИА «Стандарты и качество», 2004. – 408 с.
16. Липаев В.В. Документирование и управление конфигурацией программных средств. Методы и стандарты / В.В. Липаев. – Москва : СИНТЕГ, 1998.
17. Глушков В.М. Введение в АСУ / В.М. Глушков. – Киев : Техника, 1974.
18. Тельнов Ю.Ф. Реинжиниринг бизнес-процессов. Компонентная методология / Ю.Ф. Тельнов. – Москва : Финансы и статистика, 2004. – 320 с.
19. Шматалюк А. Моделирование бизнеса. Методология ARIS: Практическое руководство / А. Шматалюк и др. – Москва : Серебряные нити, 2001.
20. Трофимов С.А. CASE-технологии: практическая работа в Rational Rose / С.А. Трофимов. – Москва : ЗАО «Изд-во БИНОМ», 2001.