

Денис Михайлович Лысанов

**Мобильные технологии в сфере
санаторно- курортных услуг**

LAP LAMBERT Academic Publishing RU

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

LAP LAMBERT Academic Publishing

is a trademark of

International Book Market Service Ltd., member of OmniScriptum Publishing Group

17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

ISBN: 978-620-2-07828-3

Copyright © Денис Михайлович Лысанов

Copyright © 2017 International Book Market Service Ltd., member of
OmniScriptum Publishing Group

All rights reserved. Beau Bassin 2017

Содержание

Введение.....	4
1. Теоретические основы моделирования бизнес-процессов	6
1.1. Моделирование бизнес-процессов на основе мобильных приложений .	6
1.1.1. Теоретические основы создания мобильных приложений.....	6
1.1.2. Архитектура Android-приложений	9
1.2. Android и iOS: концепции распространения приложений	22
1.3. Зарубежный и отечественный опыт использования мобильных приложений	35
1.4. Теоретические основы реинжиниринга	38
1.4.1. Реинжиниринг бизнес-процессов	38
1.4.2. Инструментальные средства реинжиниринга бизнес-процессов....	43
1.4.3. Бизнес-процессы ЛПЧУП «Санаторий Ижминводы».....	48
2. Предпроектный анализ деятельности ЛПЧУП «Санаторий Ижминводы»	57
2.1. Анализ функционирования санатория	57
2.1.1. Общая характеристика.....	57
2.1.2. Анализ технико-экономических показателей предприятия	59
2.1.3. Состав и структура действующей информационной системы	61
2.2. Анализ бизнес-процесса «Сбыт»	62
2.3. Основные проблемы бизнес-процесса «Сбыт»	67
3. Проектирование информационной системы ЛПЧУП «Санаторий Ижминводы»	68
3.1. Характеристика проекта.....	68
3.2. Модель усовершенствованного бизнес-процесса «Сбыт»	70
3.3. Разработка и программная реализация задач бизнес-процесса «Сбыт»	74

3.3.1. Алгоритм решения задач	74
3.3.2. Программная реализация задач бизнес-процесса «Сбыт»	76
3.4. Экономическое обоснование разработки программного обеспечения	86
3.4.1. Бюджет затрат на разработку программного обеспечения	86
3.4.2. Бюджет решения задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки»	106
3.4.3. Расчет показателей экономической эффективности	109
Список использованных источников	116

Введение

В настоящее время использование информационных систем на предприятии становится неотъемлемой частью организации производственного процесса. Применение данной системы на предприятии позволило бы автоматизировать операции по оформлению путевок, передачи информации, а также повысить общую степень контроля наличия и состояния объектов основных средств данного предприятия. Это бы привело к получению материальных выгод по следующим направлениям:

- а) уменьшение количества времени, затрачиваемого на оформление документации;
- б) сокращение объема документооборота, что позволит снизить расходы на хранение и утилизацию промежуточной документации;
- в) наглядность представления и быстрота получения информации позволила бы ускорить и облегчить процесс принятия управленческих решений, что могло бы благоприятно сказаться на деятельности предприятия в целом.

К сожалению, можно сказать о том что, лечебно-профилактические учреждения теряют былую популярность, так как весь поток клиентов предпочитает туристические услуги. В таких условиях возникает высокая потребность в клиентах.

Чтобы привлечь отдыхающих, необходимо иметь ряд преимуществ перед конкурентами, которые будут привлекать все большее и большее количество клиентов, и обеспечивать максимально возможную прибыль.

Расположение санатория уже не достаточно для обеспечения должного потока клиентов и многие учреждения пытаются привлечь их путем создания различных скидок, акций и новых технологий.

Сегодня информационные технологии нашли широкое применение как в общем в сфере санаторных услуг, так и в отдельных его частях, таких как бронирование, резервирование, интегрированные коммуникационные сети, системы мультимедиа и так далее.

Актуальность работы состоит в том, что современные предприятия вынуждены постоянно заниматься улучшением своей деятельности. Это требует разработки новых технологий и приемов ведения бизнеса, повышения качества конечных результатов деятельности и, конечно, внедрения новых, более эффективных методов управления и организации деятельности предприятий. Разработка мобильного приложения – это новый метод усовершенствования деятельности предприятия и его бизнес-процессов.

Целью работы является разработка приложения для мобильных телефонов и планшетов. Основной характеристикой разрабатываемого приложения является использование сервисов и процессов не зависящих от основного приложения и выполняющих обработку данных в фоновом режиме.

Решаемые задачи:

- изучить бизнес-процессы предприятия и выявить проблемы;
- изучить технологии разработки, используемые для мобильных платформ Android и IOS;
- разработать информационную систему выполняющую задачи «Оформление и бронирование путевки» и «Онлайн-предоплата санаторной путевки» бизнес-процесса «Сбыт».

1. Теоретические основы моделирования бизнес-процессов

1.1. Моделирование бизнес-процессов на основе мобильных приложений

1.1.1. Теоретические основы создания мобильных приложений

Возможности мобильных приложений давно вышли за рамки индустрии развлечений, и их использование позволяет приобрести дополнительные преференции в сфере бизнес интересов. Мобильные информационные технологии способствуют улучшению сервисного обслуживания потребительской аудитории, а также развитию торговых и производственных отношений. Кроме того, не стоит сбрасывать со счетов коммерческий потенциал развлекательных приложений, этот рынок по-прежнему чрезвычайно привлекателен и имеет растущие перспективы. Целесообразность заказа мобильных приложений для бизнеса можно обосновать сразу несколькими факторами:

1. Пользователи интернета (потенциальные потребители) все чаще выходят в сеть при помощи мобильных устройств, а ряд приложений позволяет фактически привязать к себе клиентов и целевую аудиторию.
2. Возможность подключаться к серверам, которые работают по отраслевым стандартам, и быть на связи с корпоративными сетями обеспечивают практически полную независимость и оперативное реагирование при решении неотложных вопросов.
3. Динамика развития рынка мобильных приложений подтверждает прогнозы о высоком росте доходов в краткосрочной перспективе. Причем это не очередной банковский «мыльный пузырь», раздутый из очередных махинаций, а реальный продукт, который способен принести отличную прибыль.
4. Обеспечивается более тесный контакт с представителями наиболее платежеспособной целевой аудитории, так как в основном именно они являются владельцами многофункциональных мобильных устройств.

5. Возможность получения дополнительных доходов за счет размещения рекламы, а также платного скачивания приложений и их обновленных версий.

Мобильные приложения – это программные продукты, разработанные специально для мобильных устройств, смартфонов, планшетных компьютеров или других. Мобильные приложения распространяются через магазины приложений: Apple App Store, Google Play, Windows Phone Store, BlackBerry App World и др. Мобильные приложения помогают решать различные прикладные задачи: от мобильной картографии и приема электронной почты до узкоспециализированных функций. Они призваны облегчить жизнь пользователей мобильных устройств, а также ее разнообразить.

К концу 2014 года 93% всего рынка мобильных телефонов приходится на смартфоны с операционными системами Android и iOS. У каждого пятого российского пользователя есть iPhone, у каждого второго – Android. Статистики за последние три года:

- Продано 500 000 000 телефонов на системе Android
- Продано 300 000 000 iPhone
- Загружено 20 000 000 000 приложений посредством обеих систем

За последние два года количество Интернет-запросов с мобильных устройств увеличилось в пять раз.

Смартфоны Android – самый массовый продукт на мобильном рынке, число пользователей смартфонов под управлением ОС Android в России составило 12,5 млн. человек. Мобильные приложения для Android это отличный канал коммуникации, расширение охвата аудитории для бизнеса, одна из лучших платформ для модернизации предприятия и моделирования ее бизнес-процессов.

Моделирование бизнес-процессов позволяет проанализировать не только, как работает предприятие в целом, как оно взаимодействует с

внешними организациями, заказчиками и поставщиками, но и как организована деятельность на каждом отдельно взятом рабочем месте.

Существует несколько подходов к определению понятия «моделирование бизнес-процессов»:

1) моделирование бизнес-процессов - это описание бизнес-процессов предприятия, позволяющее руководителю знать, как работают рядовые сотрудники, а рядовым сотрудникам - как работают их коллеги и на какой конечный результат направлена вся их деятельность;

2) моделирование бизнес-процессов - это эффективное средство поиска возможностей улучшения деятельности предприятия;

3) моделирование бизнес-процессов - это средство позволяющее предвидеть и минимизировать риски, возникающие на различных этапах реорганизации деятельности предприятия;

4) моделирование бизнес-процессов - это метод, позволяющий дать оценку текущей деятельности предприятия по отношению к требованиям, предъявляемым к его функционированию, управлению, эффективности, конечным результатам деятельности и степени удовлетворенности клиента;

5) моделирование бизнес-процессов - это метод, позволяющий дать стоимостную оценку каждому процессу, взятому в отдельности, и всем бизнес-процессам на предприятии, взятым в совокупности;

6) моделирование бизнес-процессов - это всегда верный способ выявления текущих проблем на предприятии и предвидения будущих.

Современные предприятия вынуждены постоянно заниматься улучшением своей деятельности. Это требует разработки новых технологий и приемов ведения бизнеса, повышения качества конечных результатов деятельности и, конечно, внедрения новых, более эффективных методов управления и организации деятельности предприятий. [13, С. 224-225]

Моделирование бизнес-процессов - это эффективное средство поиска путей оптимизации деятельности компании, средство прогнозирования и минимизации рисков, возникающих на различных этапах реорганизации

предприятия. Этот метод позволяет дать стоимостную оценку каждому отдельному процессу и всем бизнес-процессам организации в совокупности.

Целью моделирования является систематизация знаний о компании и ее бизнес-процессах в наглядной графической форме более удобной для аналитической обработки полученной информации. Модель должна отражать структуру бизнес-процессов организации, детали их выполнения и последовательность документооборота. [5, 112 с.]

1.1.2. Архитектура Android-приложений

Как это часто бывает в IT, многие вещи не могут быть объяснены в отрыве от истории возникновения конкретного программного обеспечения. Вот почему мы должны обратиться к истокам ОС Android.

Разработка ОС Android была начата в 2003 молодой компанией Android Inc. В 2005 году эта компания была куплена Google. Я считаю, что главные особенности архитектуры Android были определены именно в этот период. Это заслуга не только Android Inc; архитектурные концепции и финансовые ресурсы Google оказали решающее влияние на архитектуру Android.

Архитектурные уровни Android

Операционная система Android имеет три весьма различных и сильно отделённых друг от друга уровня:

1. В основе лежит модифицированная и урезанная версия Linux
2. Над уровнем Linux находится уровень инфраструктуры приложения, содержащий виртуальную машину Dalvik, веб-браузер, базу данных SQLite, некие инфраструктурные «костыли» и Java API.
3. И, наконец, уровень написанных в Google Android-приложений. Вообще говоря, они являются расширением уровня инфраструктуры, поскольку разработчик может использовать эти приложения или их части как строительные блоки для собственных разработок.

Архитектура Android является фреймворк-ориентированной (framework-based), в противовес вольной (free-style) архитектуре.

Вольные приложения, написанные на Java, начинаются с класса, имеющего метод main(), и разрабатываются в полном соответствии с настроением разработчика.

В противоположность этому фреймворк-ориентированные приложения основываются на существующем фреймворке. Их разработка сводится к расширению неких классов или реализации интерфейсов, предоставленных фреймворком. Такое приложение не может быть запущено вне фреймворка или без него. Примером могут быть веб-приложения на Java, в которых разработчики реализуют интерфейс Servlet или расширяют одну из его реализаций, или приложения Eclipse RCP, в котором разработчик расширяет один из классов Editor или View.

Фреймворк-ориентированная архитектура ограничивает свободу разработчиков, предписывая им что и как следует делать. Но, в итоге, исчезают повторяющиеся участки кода (boilerplate code), и разработчикам приходится (хотелось бы в это верить) тщательно следовать шаблонам проектирования.

Для Java существует множество фреймворков. Тем не менее, команда Android решила создать свой собственный. Возможно, одной из причин, по которой они так поступили, была необходимость поддерживать уникальную систему управления памятью.

В «обычной» Java объекты находятся в памяти до тех пор, пока сборщик мусора не доберётся до них. Происходит это только тогда, когда объект нет ни одной ссылки от «живых» объектов (подробнее можно почитать здесь). В Android это не так. Если некий интерфейс пользователя скрывается (то есть он более не видим на экране), то нет никакой гарантии, что он находится в памяти, даже есть приложение в дальнейшем собирается использовать его. Если у ОС Android есть достаточно свободной памяти, эти объекты могут храниться в ней, но сборщик мусора может уничтожить их в

любой момент, когда ОС решит, что памяти осталось слишком мало. То же верно и для процессов. Процесс, который в данный момент времени не показывает пользователю никакого графического интерфейса, может быть уничтожен ОС Android на абсолютно законных основаниях (есть одно исключение из этого правила — сервисы; об этом мы поговорим позже).

В плане обработки взаимодействия между пользовательским интерфейсом и его логикой Android следует архитектурному шаблону «Model-View-ViewModel» (MVVM). Для разработчиков это хорошая новость, поскольку MVVM — самая лучшая архитектура GUI-приложений на настоящий момент.

Архитектура MVVM была создана с целью разделения труда дизайнера и программиста, которое невозможно, когда Java-разработчик пытается построить GUI в Swing или разработчик на Visual C++ пытается создать пользовательский интерфейс в MFC. Разработчики — сообразительные парни и имеют множество навыков, но создание удобных и привлекательных интерфейсов требует абсолютно других талантов, нежели те, которыми они обладают разработчики. Эта работа больше подходит для дизайнеров интерфейсов. Хорошие дизайнеры интерфейсов лучше знают, чего хотят пользователи, нежели эксперты в области проектирования и написания кода. Понятно, будет лучше, если дизайнер интерфейсов создаст интерфейс, а разработчик напишет код, который реализует логику этого интерфейса, но технологии типа Swing или MFC просто-напросто не позволяют поступать таким образом.

Архитектура MVVM решает эту проблему ясным разделением ответственности:

Разработка пользовательского интерфейса совершается дизайнером интерфейсов с помощью технологии, более или менее естественной для такой работы (XML)

Логика пользовательского интерфейса реализуется разработчиком как компонент ViewModel

Функциональные связи между пользовательским интерфейсом и ViewModel реализуются через биндинги (bindings), которые, по сути, являются правилами типа «если кнопка А была нажата, должен быть вызван метод `onButtonAClick()` из ViewModel». Биндинги могут быть написаны в коде или определены декларативным путём (Android использует оба типа).

Архитектура MVVM используется в том или ином виде всеми современными технологиями, например Microsoft WPF и Silverlight, Oracle JavaFX, Adobe Flex, AJAX.

Было упомянуто, что различные части Android-приложения могут вызывать друг друга и взаимодействовать между собой только формально. Как же это достигается? Фреймворк Android использует несколько шаблонов взаимодействия:

Обмен сообщениями с помощью класса Intent

Наблюдатель с использованием классов Intent и BroadcastReceiver

Позднее связывание с последующим вызовом метода используется для доступа к контент-провайдерам (ContentProviders) и локальным (внутрипроцессным) сервисам (Services)

Позднее связывание и межпроцессное взаимодействие (Inter-process Procedure Communication, IPC) для вызова сервисов (AIDL)

Android-приложение состоит из:

Java-классов, являющихся подклассами основных классов из Android SDK (View, Activity, ContentProvider, Service, BroadcastReceiver, Intent) и Java-классов, у которых нет родителей в Android SDK.

- Манифеста приложения
- Ресурсов наподобие строк, изображений и т.п.
- Файлов
- Java классы

На следующей диаграмме представлена иерархия основных классов из Android SDK, с которыми предстоит иметь дело разработчику:

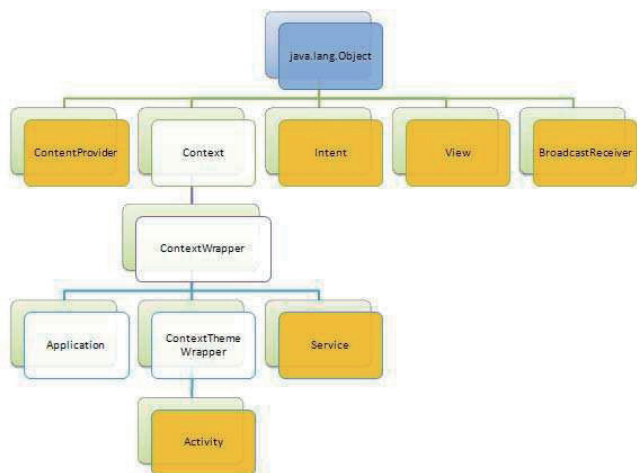


Рисунок 1.1 - Иерархия классов из Android SDK

На самом деле классов намного больше, но это основные. Выделенные жёлтым — те, с которыми разработчик работает непосредственно (в частности, наследуются от них). Остальные так же важны, но они реже используются напрямую.

`View` — базовый класс для всех виджетов пользовательского интерфейса (GUI widgets). Интерфейс Android-приложения представляет собой дерево экземпляров наследников этого класса. Можно создать это дерево программно, но это неправильно. Пользовательский интерфейс определяется с помощью XML (файлы словей, layout files), а во время исполнения автоматически превращается (inflate, термин Android) в дерево соответствующих объектов.

Класс `Activity` и его подклассы содержат логику, лежащую за пользовательским интерфейсом. При ближайшем рассмотрении этот класс соответствует `ViewModel` в архитектурном шаблоне Model-View-ViewModel (MVVM). Отношение между подклассом `Activity` и пользовательским

интерфейсом — это отношение один к одному; обычно каждый подкласс Activity имеет только один связанный с ним слой пользовательского интерфейса, и наоборот. Activity имеет жизненный цикл.

В течении жизненного цикла Activity может находиться в одном из трёх состояний:

Активно и выполняется — этот пользовательский интерфейс находится на переднем плане (говоря технически — на вершине стека активити)

Приостановлено — если данный интерфейс пользователя потерял фокус, но всё ещё видим. В таком состоянии никакой код не выполняется.

Завершено — если интерфейс пользователя невидим. В таком состоянии код не выполняется.

Код activity выполняется только когда соответствующий интерфейс пользователя видим и имеет фокус. Нет гарантии, что объект Activity и связанные с ним объекты находятся в памяти, когда активити приостановлено или завершено (очень важно помнить об этом; ранее мы уже обсуждали этот момент управления памятью в Android).

Класс ContentProvider и его подклассы представляют model в архитектуре MVVM. В большинстве практических случаев это обёртка над базой данных SQLite с немного причудливым способом доступа на основе URI. Теоретически, никто не мешает разработчику создать ContentProvider на основе чего-то ещё, кроме базы данных. Тем не менее, существующий метод query() контент-провайдера возвращает объект Cursor, который крайне похож на JDBC ResultSet интерфейсом и тем, как он работает. Поэтому вряд ли кто-то усомнится, что настоящее назначение контент-провайдеров — инкапсулировать базу данных.

Основная идея контент-провайдеров, похоже, базируется на архитектуре AJAX приложений. AJAX приложения обычно используют архитектуру MVVM, где модель представлена как URI на стороне сервера (тем не менее, это изменилось с появлением HTML5, который позволяет хранить данные локально). В самом деле, тот факт, что контент-провайдеры

запрашиваются с помощью URI и создают расширение с помощью типов MIME указывает на то, что в основе лежит AJAX. Напомню, ребята из Google создали большое количество AJAX приложений, таких как Gmail, Google Docs и т.п., поэтому вполне естественно, что идеи заимствовались из архитектуры AJAX.

Возможно, кто-то ещё пришёл с ещё одной отличной идеей: как было бы здорово иметь полноценную реляционную базу на мобильном устройстве. И, как результат, они соединили две хороших идеи в один класс `ContentProvider`. Как это обычно и случается в разработке ПО, соединение двух хороших идей не всегда даёт в результате хорошую идею; в случае Android мы имеем несколько обескураживающий дизайн контент-провайдеров. [3, с. 30-35]

Сотрудники Google испытывают те же трудности (прочтите, пожалуйста, их документацию). Их классификация, в основном, говорит, чем этот класс не является. Сервис — это разновидность `Model`, обслуживающая несколько иные варианты использования, нежели `ContentProvider`.

Архитектурный дизайн `Android Service` навеян сервисами `OSGI`.

Сервисы были созданы ребятами из Google как решение логической проблемы, возникшей из-за модели потоков Android.

`Activity` активно и выполняется только когда его пользовательский интерфейс находится на переднем плане. Как только интерфейс другого `Activity` закрывает собой текущее, последнее останавливается, даже если оно что-то делало. А что, если вам нужно выполнять некую операцию, даже если процесс, которые её выполняет, не на переднем плане? С помощью `Activity` вы не сможете этого сделать. Вы не сможете это сделать и с помощью `ContentProvider`, поскольку у них нет собственного жизненного цикла, и они могут выполняться только пока `Activity`, использующее его, активно.

И тут на помощь приходят сервисы. Они могут выполняться даже когда процесс, в котором они работают, не на переднем плане. Так, если вы разрабатываете активити, выполняющее растянутую во времени операцию,

которая должна завершиться даже работая в фоне, вы должны создать Service, реализующий эту операцию, и запустить его из Activity.

Service так же имеет жизненный цикл. Это означает, что он может быть инстанцирован и запущен Android-приложением по некому условию (мы обсудим это позже).

Service, как model, преследует более общие цели, нежели ContentProvider. Он может использовать базу данных, но его API не связано с БД, как в случае ContentProvider. В большинстве случаев сервисы используются для связи с внешними серверами.

Класс BroadcastReceiver и его подклассы представляют собой «подписчика» в механизме взаимодействия издатель/подписчик, реализованном в архитектуре Android.

Конечно, разработчик под Android не ограничен одним только расширением классов из Android SDK. Он может писать собственные классы так, как захочет. Но все они будут только хелперами («helper classes») для классов из Android SDK.

Манифест Android

Манифест Android — ещё одна важная часть Android-приложения. Идея была навеяна манифестами плагинов для Eclipse.

Манифест Android представляет собой XML файл и выполняет несколько функций. Вот как их описывает Google:

Определяет имя Java-пакета приложения. Имя пакета представляет собой уникальный идентификатор для приложения.

Описывает компоненты приложения — активити, сервисы, бродкастерсерверы и контент-провайдеры. Определяет имена классов, реализующие каждый из компонентов и оглашает их возможности (например, какие Intent-сообщения они могут обрабатывать). Эти объявления позволяют системе Android знать, какие компоненты и при каких условиях могут быть запущены.

Предопределяет процессы, которые будут содержать компоненты приложения.

Объявляет разрешения, которые приложение должно иметь для доступа к защищённым частям API и взаимодействия с другими приложениями.

Также объявляет разрешения, которые требуются для доступа к компонентам приложения.

Перечисляет классы Instrumentation, которые предоставляют профайлинг и другую информацию во время работы приложения. Эти объявления присутствуют в манифесте только пока приложение разрабатывается и тестируется; они удаляются перед публикацией приложения.

Объявляет минимальный уровень Android API, который требует приложение.

Перечисляет библиотеки, с которыми приложение должно быть связано.

Обратите внимание на второй пункт. Имеется ввиду, что если некий класс расширяет Activity, ContentProvider, BroadcastReceiver или Service в вашем приложении, этот класс не может быть использован до тех пор, пока он не описан в манифесте.

Ресурсы

Каждое современное GUI приложение в той или иной форме использует ресурсы. Android-приложения — не исключение. Они используют следующие типы ресурсов:

- Изображения

- Слои GUI (XML файлы)

- Объявления меню (XML файлы)

- Текстовые строки

Способ, которым ресурсы связываются с Android-приложением — это что-то необычное. Как правило, в Java ресурсы идентифицируются строками.

Такие строки могут содержать, например, путь и имя файла, содержащего изображение, или ID данной строки и т.п. Проблема в том, что ошибки в таких ссылках не могут быть обнаружены в процессе трансляции кода.

Давайте рассмотрим следующий пример. Файл с именем `mybutton.png` содержит картинку для кнопки. Разработчик совершает ошибку и набирает `mybuton.png`, ссылаясь на ресурс из кода. Как итог, код пытается использовать несуществующий ресурс, но компиляция пройдет успешно. Ошибка может быть обнаружена только в ходе тестирования (а может и не быть обнаружена вовсе).

Ребята из Google нашли элегантное решение этой проблемы. При сборке Android-приложения генерируется специальный Java-класс с именем `R` (всего лишь одна буква). Этот класс содержит несколько `static final` наборов данных. Каждый такой набор данных — ссылка на отдельный ресурс. Эти ссылки используются в коде приложения для связи с ресурсами. Теперь каждая ошибка в ссылке на ресурсы проявляет себя в процессе компиляции.

Файлы

Android-приложение использует несколько разных типов файлов:

Файлы «общего назначения»

Файлы БД

Файлы `Opaque Binary Blob (OBB)` (они представляют собой зашифрованную файловую систему, которая может быть монтирована для приложения)

Закешированные файлы

Хотя, в конечно итоге, все они только файлы Linux, имеет смысл рассматривать их как отдельные типы файлов только в разрезе обработки их различным API и отдельного хранения. Также они отделены от файлов, хранящихся во внутреннем или внешнем хранилище (последнее может отсутствовать или исчезнуть/появиться в любой момент).

API для работы с файлами реализован классом Context, от которого порождены классы Activity и Service.

Обмен сообщениями

Очевидно, это первый механизм взаимодействия, с которым сталкивается разработчик, начав программировать под Android. Он используется для запуска активности или сервиса. Механизм реализуется с помощью трёх методов класса Context (о котором говорилось в одной из предыдущих статей) и одного метода класса Activity:

startActivity() класса Context запускает активность

startActivityForResult() класса Activity запускает активность

startService() или bindService() класса Context запускает сервис, если это необходимо, и связывается с ним (между этими двумя методами существует некоторая разница)

В обоих случаях класс Intent выполняет роль передаваемого сообщения.

Как Android узнаёт, какому активити или сервису адресуется сообщение? Для этого есть два способа:

Intent может явно указывать имя компонента. Такие интенды называются явными («explicitIntents»). Целевой компонент определяется методами setComponent() и setClass(). Этот тип интенд-ориентированного механизма взаимодействия не реализует шаблон позднего связывания, в отличие от всех остальных.

Целевой компонент определяется весьма необычным механизмом принятия решения, основанном на фильтрах интендов.

Имейте ввиду, что в случае отправки сообщения (экземпляра класса Intent) механизм взаимодействия доставит его только одному компоненту, или не доставит вовсе.

Издатель/подписчик

Механизм взаимодействия издатель/подписчик использует тот же класс Intent в качестве сообщения и те же фильтры, что и механизм определения

целевого компонента, но работает несколько иначе. Интененты, используемые в данном случае, называются «широковещательными». Эти интененты могут быть доставлены только `BroadcastReceiver`.

В отличие от интенентов, используемых для отправки сообщений, один широковещательный интенент может быть доставлен нескольким компонентам.

Фактически, `Android` реализует два разных механизма издатель/подписчик, каждый из которых имеет по две версии (обычную и «липкую», «sticky»):

Нормальное вещание реализуется методами `sendBroadcast()` и `sendStickyBroadcast()` класса `Context`.

Упорядоченное вещание реализуется методами `sendOrderedBroadcast()` и `sendStickyOrderedBroadcast()`.

Нормальный механизм вещания доставляет интенент до всех подходящих бродкаст-ресиверов асинхронно, и они реагируют на них независимо. То есть одни бродкаст-ресивер не может влиять на то, как другой бродкаст-ресивер реагирует на интенент, или как интенент доставляется другому бродкаст-ресиверу. Это обычный шаблон издатель/подписчик.

Упорядоченный механизм вещания доставляет интененты до всех подходящих бродкаст-ресиверов последовательно, то есть только одному за раз. Поэтому каждый бродкаст-ресивер потенциально может:

Использовать результаты работы предыдущего бродкаст-ресивера, который работал над тем же интенентом

Если необходимо, прервать дальнейшую обработку интенента

Компоненты, инициализировавшие упорядоченное вещание, получают возможность принимать результат обработки интенента от всей цепи бродкаст-ресиверов.

Упорядоченное вещание является реализацией шаблона цепочка обязанностей.

«Липкое» вещание — разновидность обычного вещания. Разница в том, что интент доступен до тех пор, пока он не будет явно удалён методом `removeStickyBroadcast()` класса контекст. Помните, что этот тип вещания следует использовать с осторожностью, поскольку не удалённые «липкие» интенты приводят, в итоге, к утечке памяти.

Выводы по интент-ориентированному механизму взаимодействия

Все описанные механизмы взаимодействия используют класс `Intent` в качестве сообщения. Обратите внимание, большая часть представленных методов реализована классом `Context`, а это значит, что теоретически они могут быть использованы всеми его подклассами.

Отсюда можно сделать вывод, что все механизмы взаимодействия Android используют класс `Intent`. Но это не совсем так.

В большинстве случаев класс `ContentProvider` используется как обёртка вокруг базы данных SQLite. Хотя, теоретически, он может быть использован для других механизмов хранения\получения данных. Об использовании этого класса можно почитать здесь.

Существует множество способов использования контент-провайдеров. Одни включают получение объектов, которые работают как прокси над контент-провайдером (обычно это экземпляр или подкласс `ContentProviderClient`, который реализует шаблон прокси). Другие возвращают экземпляр класса `Cursor`, который позволяет перебирать множество данных, возвращённых контент-провайдером. В любом случае, конкретный `ContentProvider` должен идентифицироваться по URI (подробнее см. тут).

В стороне от класса `ContentProvider` находится класс `Service`, который выполняет роль модели в архитектуре MVVM. В предыдущей статье мы обсуждали гипотетические причины, по которым Android имеет два разных класса, выполняющих одну и ту же роль.

Локальный сервис запускается в том же процессе, где и вызывается. Архитектурно он похож на контент-провайдер, за исключением того, что у

первого имеется собственный жизненный цикл и отсутствует обязательная оглядка на БД. Для доступа к локальному сервису используется разновидность позднего связывания.

Удалённый сервис работает в другом процессе, а не в том, из которого он был вызван. Связь с таким сервисом происходит через разновидность IPC, называемую AIDL (Android Interface Definition Language). [6, с. 110-111]

1.2. Android и iOS: концепции распространения приложений

Сейчас смартфоны под управлением Android и iOS являются одними из самых популярных среди потребителей во всем мире, хотя по количеству проданных устройств и наблюдается существенный разрыв. Так, согласно отчету NPD Group, доля Android-смартфонов на рынке США составляет 61%, в то время как доля iOS — 29%. Несмотря на все возрастающую популярность двух конкурирующих платформ, проблемы их безопасности разительно отличаются. В то время как сообщения об очередной атаке злоумышленников на пользователей Android-устройств появляются с завидной регулярностью, владельцы «яблочных» i-продуктов фактически не испытывают никаких опасений, а большинство из них могло даже не слышать об имевших место атаках.

В чем же причина столь большой разницы? Для начала рассмотрим технические особенности каждой операционной системы.

В основе ОС Android лежит Ядро Linux, которое выступает мостом между аппаратными возможностями смартфона и программными функциями Android. Большинство приложений для этой системы создаются при помощи языка Java и выполняются в виртуальной среде Dalvik, являющейся интерпретацией виртуальной Java-машины. Помимо языка Java, для написания программ могут использоваться и другие языки, но вне зависимости от способа создания приложения работают в специальной защищенной среде — песочнице (sandbox). Благодаря этой песочнице

приложения не могут получить доступ к процессам других программ или их данным.

В Android всем приложениям присваивается уникальный идентификатор пользователя User ID или UID. Во время работы каждая программа выполняется под своим пользователем и в отдельном процессе, что является составной частью песочницы. У одного и того же приложения, установленного на разных устройствах, этот идентификатор может быть разным, однако он всегда сохраняется за приложением на протяжении всего времени пребывания его в системе. По умолчанию доступ к root (учетной записи суперпользователя с неограниченными правами в системе) имеют только ядро и некоторые ключевые системные приложения.

Обязательным условием корректной работы приложений в ОС Android является наличие у них цифровой подписи. Если ее нет, система просто не позволит выполнить установку программы. Цифровая подпись используется для идентификации разработчика, а также для обеспечения безопасной взаимной работы приложений. Когда разработчик создает и тестирует программу, она подписывается временным ключом, используемым для отладки. Для готового приложения необходима личная цифровая подпись разработчика, которая создается после компиляции приложения и затем используется для подписи уже готового программного пакета. Особенностью Android является то, что разработчик самостоятельно подписывает свое приложение, без вмешательства кого бы то ни было со стороны, после чего приложение сразу готово к использованию.

Из-за ограничений, накладываемых песочницей, два и более приложений не могут работать под одним и тем же процессом, так как у каждого из них должна быть своя уникальная учетная запись пользователя. Тем не менее, у разработчиков существует возможность назначить для нескольких программ единый идентификатор UID, но для этого все приложения должны иметь одну и ту же цифровую подпись. Если это

условие выполняется, приложения могут работать с одной учетной записью пользователя.

Базовое приложение Android не содержит в себе функций, требующих доступа к контролируемым и защищаемым песочницей данным или возможностям смартфона. В случае необходимости задействования определенного функционала или доступа к данным разработчик должен указать для приложения соответствующие разрешения (Permissions) в конфигурационном файле `AndroidManifest.xml`. Во время установки подобного приложения пользователь увидит сообщение, содержащее информацию о функциях, необходимых для его работы. Операционная система оставляет за пользователем право решать, стоит ли доверять выбранной программе. После установки выполнение той или иной функции уже не потребует участия пользователя.

В Android предусмотрена возможность дистанционного удаления установленных приложений. Компания Google может задействовать эту функцию, например, в случае если то или иное приложение представляет угрозу.

Операционная система iOS, как и OS X, на которой она базируется, построена на основе ядра Mach и интерфейсов BSD и представляет собой UNIX-подобную систему. Системное ядро отвечает за управление памятью, работу с файловой системой, доступ к аппаратным функциям, сети и т. п. Драйвера выполняют связующую роль между аппаратной частью и системными функциями. Доступ к ядру и драйверам ограничен, и его имеют лишь некоторые системные приложения и библиотеки.

Для обеспечения безопасности каждое устанавливаемое приложение помещается в песочницу — `sandbox`. Приложениям отводится отдельное место в файловой системе (каталог, содержащий саму программу и ее данные), доступ к которому для других приложений ограничивается. Помимо этого, песочница контролирует доступ к различным функциям системы, сети, аппаратным возможностям смартфона и т. д. Например, ограничения

затрагивают доступ к СМС- и почтовой переписке, выполнению скрытого набора номера, отправке сообщений. Если в работе приложения предусмотрен обмен определенными данными с другими программами, у разработчиков существует возможность предоставить такой доступ. Все приложения, устанавливаемые пользователями, работают с ограниченными правами.

Приложения для iOS должны быть подписаны цифровым сертификатом. Этот сертификат служит идентификатором разработчика и гарантирует, что приложение было создано именно им, а также что разработчик является зарегистрированным членом Developer Program. Для получения сертификата разработчик должен направить запрос в Apple. В случае положительного решения этого вопроса он получает цифровой сертификат безопасности, который в дальнейшем используется при создании приложений.

Как и Google, компания Apple при необходимости может дистанционно удалять со смартфонов пользователей опасные или нарушающие какие-либо правила приложения.

Как мы видим, обе операционные системы обеспечивают вполне приемлемый уровень безопасности. Почему же сообщения об очередных угрозах для владельцев Android-устройств звучат гораздо чаще, чем об угрозах для смартфонов от Apple?

Рассмотрим, какие методы наиболее часто встречаются в арсенале злоумышленников, преследующих свои корыстные цели. Можно выделить три основных вектора атак:

- онлайн-мошенничество;
- социальная инженерия;
- создание вредоносных и нежелательных программ.

Практически всегда эти «инструменты» встречаются в комбинированном виде, что позволяет киберпреступникам использовать их наиболее эффективно.

Владельцы мобильных устройств под управлением Android сталкиваются сразу со всеми тремя проблемами, среди которых наиболее остро выделяются вредоносные программы. Для смартфонов от Apple, напротив, проблема вредоносных программ практически отсутствует. Чтобы понять, почему это происходит, необходимо более детально изучить логику работы обеих операционных систем с приложениями, а также особенности процесса их создания и распространения среди конечных потребителей.

Что касается ОС Android, прежде всего, стоит отметить тот факт, что SDK — средства разработки — доступны для Windows, Linux и Mac OS X. Этим обеспечивается охват самого широкого круга разработчиков.

Созданные приложения могут распространяться несколькими способами. Первый — через основную официальную площадку — каталог Google Play. За небольшую единовременную плату в размере \$25 любой желающий может зарегистрировать аккаунт разработчика и распространять в этом каталоге созданные им приложения. Помещаемые в Google Play программы практически сразу становятся доступными для загрузки пользователями. До недавнего времени публикуемые в этом каталоге приложения не подвергались какой-либо проверке, однако из-за того, что злоумышленники периодически размещали в нем вредоносные программы, компании Google пришлось ввести специальную систему Bouncer, которая должна искать подозрительные и нежелательные приложения на основе уже имеющихся данных. Тем не менее, ни о какой жесткой политике проверки и тестирования приложений перед размещением речи не идет.

У каталога Google Play имеется интернет-версия. В ней содержится не только информация о доступных в нем приложениях, но также существует возможность их установки непосредственно с веб-сайта. Для этого владелец смартфона должен выполнить вход под своей учетной записью электронной почты, которая привязана к его мобильному устройству. Простым нажатием кнопки можно инициировать установку, которая при наличии доступа к

Интернету будет сразу же выполнена, при этом на самом устройстве не возникнет никаких запросов.

Второй способ — свободная установка приложений вне официального каталога: это может быть загрузка из различных веб-порталов, сторонних каталогов, через компьютер или обмен между устройствами. По умолчанию данная возможность отключена, однако ничто не мешает пользователю ее активировать. Для разработчиков, распространяющих свои приложения, минуя каталог Google Play, никакая регистрация не требуется, и созданные ими программные продукты сразу готовы к использованию.

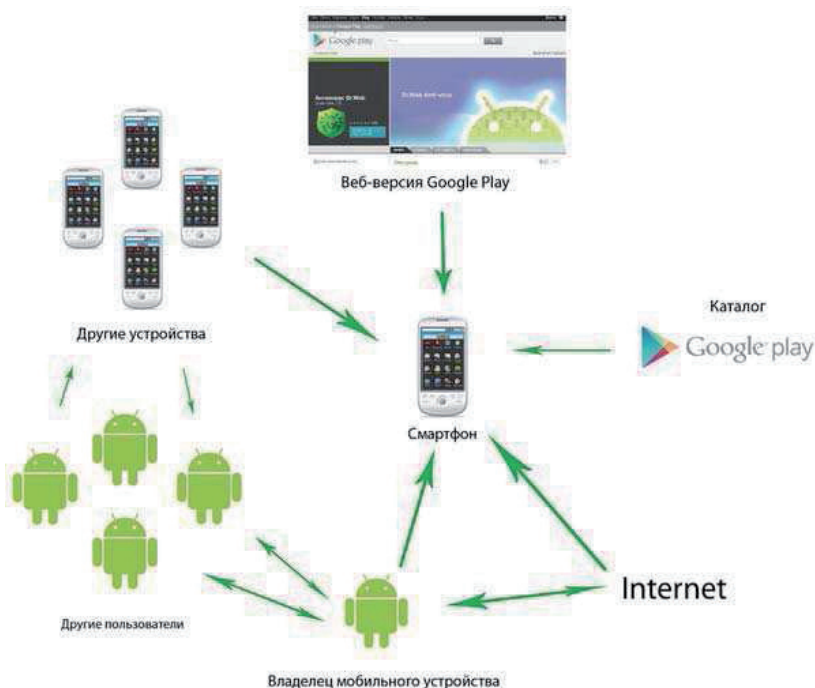


Рисунок 1.2 - Схема возможных каналов получения приложений в ОС Android

Пользуясь такой свободой, вирусописатели быстро освоились. Вполне демократичная стоимость аккаунта в Google Play дает им возможность без

особых финансовых потерь создавать новые учетные записи в случае их блокировки. Различные методы социальной инженерии позволяют легко заставить пользователей загрузить и установить троянские программы, распространяющиеся в том числе вне официального каталога, например, под видом некоего обновления системы или популярного приложения (популярная тактика, применяемая для доставки СМС-троянцев семейства Android.SmsSend — одной из главных и наиболее распространенных угроз).

Система разрешений Android предоставляет пользователю информацию о функциях, которые сможет выполнять устанавливаемая программа, однако далеко не всегда можно однозначно сказать, как именно поступит приложение с той или иной функцией, чем и пользуются злоумышленники. Так, например, необходимость доступа к сети Интернет, закладкам браузера, контактам, координатам GPS, номеру телефона и т. п. может иметь как вполне безобидная программа, так и троянец, распространяющийся под ее видом.

Не стоит забывать, что компания Google позиционирует Android как открытую операционную систему. Несмотря на то что некоторые компоненты все же являются недоступными (например, ключевые системные драйверы и проприетарные приложения), имеющийся в открытом доступе код позволяет эффективнее обнаруживать различные уязвимости, которые в том числе берут на вооружение и используют в своих вредоносных программах злоумышленники. Одним из наиболее ярких примеров таких приложений являются троянцы Android.DreamExploit, Android.Wukong и Android.Gongfu, которые эксплуатируют уязвимости операционной системы, позволяющие повышать привилегии программного окружения до уровня root. Это дает им более широкие возможности по выполнению своих вредоносных функций, например, установке других приложений без участия пользователя.

Так как Android доступна большому числу производителей, выпускающих смартфоны с различными техническими характеристиками и

версиями операционной системы, на рынке присутствует большое разнообразие мобильных устройств, что вызывает фрагментацию. Несмотря на попытки Google оперативно выпускать обновления системы, закрывающие обнаруживаемые уязвимости, фрагментация платформы часто мешает пользователям своевременно получать эти обновления, так как каждый производитель имеет свою стратегию по их выпуску. Случается и так, что производитель вообще отказывается от дальнейших обновлений некоторых моделей по причине их устаревания, экономическим или техническим соображениям или же по каким-либо другим причинам. Таким образом, пользователи становятся беззащитными перед угрозой троянцев, использующих root-эксплойты, о которых говорилось выше.

Доступный исходный код Android также может применяться различными энтузиастами, создающими свои сборки операционной системы. Вирусописатели освоили и эту нишу. Они могут распространять троянские программы (такие как, например, Android.SmsHider), имеющие цифровую подпись одного из образов такой сборки. Благодаря этому методу приложение получает недоступные в обычных условиях права суперпользователя в системе, которой принадлежит использованный для подписи сертификат безопасности. А некоторые злоумышленники могут и вовсе встраивать троянские программы в подобные сторонние прошивки.

Таким образом, у вирусописателей существует несколько способов доставки своих творений пользователям.

Создание и распространение приложений для операционной системы iOS имеет ряд отличий. Во-первых, средства разработки доступны только для компьютеров Mac OS X. Во-вторых, разработчик должен выбрать подходящую ему модель распространения созданных приложений:

— iOS Developer Program Individual или iOS Developer Program Company — для индивидуальных разработчиков и компаний, которые создают приложения для массового рынка;

— iOS Developer Enterprise Program — для корпоративных клиентов, которым необходимы приложения для внутренних нужд;

— iOS Developer University Program — для образовательных учреждений с целью повышения квалификации студентов и ознакомления их с процессом создания приложений для мобильной системы от Apple.

Первая модель является стандартной, рассчитанной на общий потребительский рынок (именно эта модель в дальнейшем будет использоваться в качестве основы для рассмотрения). Созданные на ее основе приложения доступны только через официальный каталог приложений App Store, причем прежде чем увидеть свет, они проходят определенную проверку. Процесс создания приложения, проверки и добавления в App Store выглядит следующим образом: сначала разработчик должен пройти простую регистрацию на сайте developer.apple.com, после чего ему будет доступна среда разработки. Однако для того, чтобы иметь возможность распространять в App Store созданные приложения, ему необходимо выбрать одну из подходящих моделей распространения (в случае с iOS Developer Program Individual или iOS Developer Program Company членство стоит \$99 в год).

Для того чтобы разместить созданное приложение в каталоге App Store, разработчику необходимо направить его на рассмотрение в Apple. Для этого требуется уже откомпилированная и готовая к использованию программа. При проверке внимание обращается на соответствие многим требованиям, среди которых — оформление приложения и логика построения интерфейса, отсеивание малосодержательных приложений, а также программ, которые дублируют уже имеющийся функционал и вводят пользователя в заблуждение своим сходством со стандартным ПО. Также не допускаются приложения, иконка которых похожа на иконку системных приложений. Обязательным является проверка на следование установленным правилам безопасности. Среди них стоит выделить следующие:

— функционал соответствует заявленным возможностям, то есть не имеется скрытых и необъявленных функций;

— приложение может выполнять загрузку файлов, если это необходимо для ее основной работы, причем пользователь должен быть уведомлен об этом;

— если требуемый объем загрузки через мобильную сеть превышает лимит в 50 МБ, загрузка должна происходить только по Wi-Fi либо отдельными сессиями;

— приложение не выполняет установку или запуск других программ;

— приложение выполняется в своей среде и не имеет доступа к областям других приложений, включая их файлы;

— в приложении не используются недокументированные функции системы;

— пользователь должен быть уведомлен о сборе персональной информации (включая координаты местоположения), выполняемом приложением, и должен дать согласие на такие действия перед их выполнением;

— приложение должно уведомить пользователя о целях сбора персональной информации и возможных областях их применений.

Проверка приложений выполняется сначала в автоматическом, а затем в ручном режиме. В случае успешного прохождения этого процесса приложение попадает в App Store, откуда становится доступным пользователям.

Помимо установки программ непосредственно со смартфона, существует еще один стандартный способ, который выполняется при помощи программы iTunes, устанавливаемой на компьютер владельца устройства. Подключив смартфон к компьютеру, пользователь легко может загрузить нужную программу из каталога App Store.

Вторая модель — iOS Developer Enterprise Program — ориентирована на корпоративных клиентов, которым требуется создание приложений для

внутренних нужд. Как и в предыдущем случае, для доступа к ней необходима платная регистрация, которая стоит уже \$299 в год. Созданные впоследствии приложения могут быть распространены среди сотрудников компании и установлены следующими способами: через компьютер с имеющейся на нем программой iTunes, при помощи администрирующего приложения iPhone Configuration Utility, либо через специально созданный веб-сайт. Помимо корпоративных, также возможна установка общедоступных приложений, размещенных в каталоге App Store.

Третью модель — iOS Developer University Program — рассматривать не стоит, так как она служит образовательным целям. Кроме описанных способов установки приложений на смартфоны с iOS, существует и третий. Он становится возможным после выполнения операции jailbreak — снятия стандартных ограничений системы и получения прав доступа к файловой системе устройства. Сделав jailbreak, пользователь может устанавливать приложения, минуя App Store. Доступ к таким приложениям осуществляется при помощи специализированных программ-каталогов, наиболее известный и широко применяемый из которых — Cydia. Маловероятно, что корпоративный сектор будет использовать такой метод, поэтому он в большей степени присущ определенной части обычных пользователей.



Рисунок 1.3 - Схема возможных каналов получения приложений в iOS

Учитывая серьезные ограничения в каналах получения приложений, а также долгий и детальный процесс проверки перед размещением их в App Store, неудивительно, что злоумышленники не торопятся создавать вредоносные программы для смартфонов Apple. Высокая цена аккаунта разработчика также помогает снизить число киберпреступников, желающих попробовать свои силы, ведь в случае неудачи, например на этапе проверки приложения, аккаунт мошенника будет заблокирован, и он потеряет ощутимую сумму, так и не успев толком заработать.

Среди того немногочисленного числа вредоносных программ, существующих для iOS, самой известной, пожалуй, является червь Icke. Этот червь функционировал только на взломанных смартфонах, где был выполнен jailbreak. Для своего распространения он использовал терминал SSH и стандартный root-пароль устройства: alpine. Червь устанавливал в качестве

фонового изображения фотографию певца Рика Эстли (Rick Astley). Другая версия червя уже имела в своем функционале возможность принимать и выполнять команды от своих создателей. Более того, он менял стандартный пароль, чтобы снизить вероятность того, что пользователь сможет от него избавиться.

Помимо вредоносной программы Ike, для iOS существуют и коммерческие программы-шпионы. Однако для их установки необходимо, чтобы на устройстве также был выполнен jailbreak, так как они распространяются через сторонние площадки, отличные от App Store. Для успешного внедрения и настройки такого шпиона требуется непосредственное участие злоумышленника.

Также следует помнить, что iOS, являясь закрытой операционной системой, принадлежащей исключительно компании Apple, менее охотно поддается анализу. Это вовсе не значит, что уязвимости в ней найти невозможно. Тем не менее, обнаруженные уязвимости операционной системы, которые применяются на практике, больше всего направлены на выполнение операции jailbreak. Может сложиться впечатление, что смартфоны с jailbreak'ом являются хорошей мишенью для злоумышленников, и это отчасти справедливо. Однако нельзя забывать, что пользователей таких смартфонов все-таки существенно меньше, чем тех, кто предпочел ничего не модифицировать, поэтому экономическая эффективность (а именно финансовая сторона является одной из главных причин деятельности злоумышленников) здесь также под вопросом.

Ко всему прочему, iOS не лицензируется сторонним производителям. Это помогает сохранять контроль над целостностью платформы, обеспечивая тем самым более высокую надежность и позволяя оперативно выпускать обновления системы для все еще поддерживаемых устройств. Это, в свою очередь, также снижает вероятность создания злоумышленниками вредоносных программ, использующих различные уязвимости системы, как, например, происходит в случае Android. Хорошим примером такого подхода

может служить недавний выпуск обновления системы, iOS 5.1.1, устраняющего уязвимости, которые могли привести к подмене отображаемого адреса веб-страницы, межсайтовому скриптингу, а также выполнению произвольного кода. Это обновление доступно сразу для нескольких поколений мобильных устройств от Apple.

Таким образом, мы можем видеть в концепции создания и распространения приложений на Android и iOS принципиальные различия, которые приводят к совершенно разным результатам в плане наличия вредоносных программ — одного из главных инструментов киберпреступников. На данный момент создание таких программ для мобильной операционной системы от Apple невыгодно злоумышленникам, так как этот процесс сопряжен с определенными финансовыми рисками. К тому же им незачем направлять часть своих усилий на еще одну платформу, когда другая — Android — обеспечивает их всем необходимым. [7, с. 20-23]

1.3. Зарубежный и отечественный опыт использования мобильных приложений

Рынок мобильных приложений относительно молод, однако за последние несколько лет он существенно вырос. Его активному развитию во всём мире способствовал быстрый рост рынка мобильных устройств и проникновение мобильных технологий во все сферы частной и общественной жизни. На сегодняшний день мобильные приложения являются одним из наиболее привлекательных для инвестиций сегментов рынка.

Хотя мобильный рынок России существенно уступает в размерах рынкам США, Азии и Европы, многие эксперты разделяют мнение, что он обладает значительным потенциалом роста. Согласно исследованиям, в 2014 году объем российского рынка мобильных приложений оценивался в \$246 млн., и аналитики ожидают, что он продолжит расти. Исследование показывает, что более 50% пользователей планшетов и смартфонов минимум раз в день выходят в интернет, и каждый день

запускают видео-приложения и играют в мобильные игры. Сообщается также, что 40% пользователей мобильного интернета скорее предпочтут лишиться телевизора, нежели смартфона.

Россия занимает пятое место по общему числу загрузок на iOS, уступая только США, Китаю, Японии и Великобритании. Этот же отчет показывает, что по загрузкам на Android Россия на четвертом месте после США, Индии и Бразилии. Однако при более внимательном рассмотрении доходов от приложений для iOS и Android становится очевидным тот факт, что Россия еще далека от раскрытия своего полного потенциала, поскольку она занимает 10 и 9 место соответственно. [9, с. 50-51]

Прогноз мобильного рынка России

Несмотря на недавний спад российской экономики, прогнозы относительно ее мобильного рынка довольно оптимистичные. Согласно прогнозам, в 2015 году российский рынок мобильных приложений в среднем вырастет на 186%, а его объем к 2016 году достигнет \$1,3 млрд. Рост обеспечат такие факторы, как растущее количество мобильных устройств, быстрое развитие сетей 4G и общее смещение потребительских и коммуникационных привычек в сторону мобильности (например, мобильный шопинг, развитие мобильного банковского обслуживания и мобильных платежных систем, распространение мобильных социальных сетей и мобильных мессенджеров, растущая популярность геолокационных сервисов и пр.)

2014 год был отмечен резким ростом продаж смартфонов: в России было продано 26 млн. устройств - на 47% больше, чем в 2013 г. Смартфоны составили 61% от общего числа (43 млн.) проданных мобильных телефонов. Однако стоит отметить, что общий рост рынка регистрируется за счет увеличивающихся продаж бюджетных смартфонов (стоимостью \$50-\$110), в то время как остальные сегменты мобильного рынка идут на спад.

Продажи

■ смартфоны, 26 млн. ■ мобильные телефоны, 16,9 млн

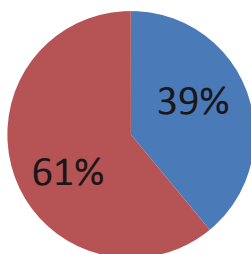


Рисунок 1.4 - Количество проданных смартфонов и мобильных телефонов за 2014 год

Если рассматривать операционные системы, на российском мобильном рынке доминирует Android, удерживающий 50,65% рынка по состоянию на декабрь 2014 г. На втором месте iOS с 43,59% (на 15,74% больше, чем в 2013 г.), в то время как Windows Phone теряет позиции с долей рынка 2,42% (по сравнению с 3,34% в 2013 г.). Суммарная доля всех остальных операционных систем составляет 3,34% (данные от The Statistics Portal, декабрь 2014).

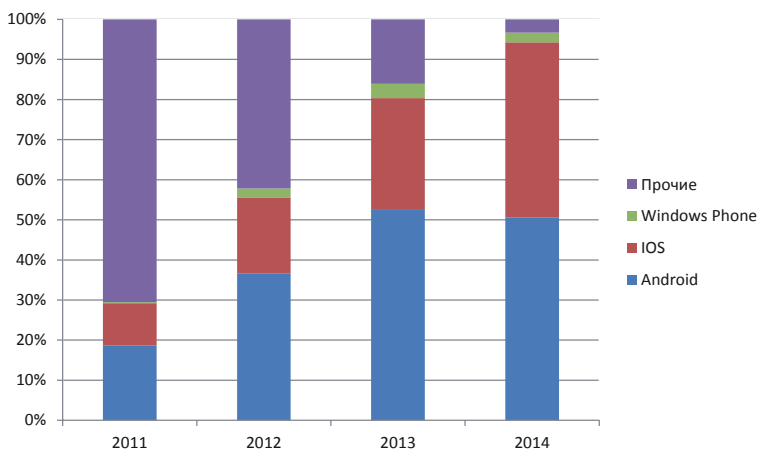


Рисунок 1.5 - Мобильные операционные системы в России, 2011-2014 г.г.

1.4. Теоретические основы реинжиниринга

1.4.1. Реинжиниринг бизнес-процессов

Основной целью реинжиниринга или инжиниринга бизнес-процессов предприятия является проектирование такой системы управления процессами предприятия, которая обеспечивает разработку стандарта управления.

Стандарт управления предприятия - это документальная процедура всех процессов, их взаимосвязи, правила принятия управленческих решений и измерители качества и количества этих процессов.

Предпроектный анализ деятельности предприятия устанавливает состояние предприятия «как есть». Реинжиниринг (инжиниринг) устанавливает состояние предприятия «как надо» с учетом состояния «как есть».

Реинжиниринг бизнес-процессов или BPR (Business Process Reengineering), начиная с 1990 года, вызывает активный интерес специалистов в области менеджмента и информационных технологий.

Сегодня методы BPR взяты на вооружение уже практически всеми ведущими компаниями мира.

BPR является научно-практическим направлением, возникшим на стыке двух различных дисциплин, и поэтому он требует новых специфических средств представления и обработки проблемной информации понятных и удобных как менеджерам, так и разработчикам информационных систем. Подобные средства требуют интеграции ключевых достижений информационных технологий и создания соответствующих инструментальных средств поддержки реинжиниринга, ориентированных на использование как специалистами в области информационных технологий, так и менеджерами.

Большой вклад информационных технологий в организацию деятельности компании - это автоматизация бизнес-процессов (Business Process Automation). Но автоматизация приводит лишь к ускорению существующих процессов, которое не может, в подавляющем большинстве случаев, привести к тому многократному улучшению эффективности, которое предусматривает подлинный реинжиниринг. На самом деле можно выделить два способа влияния информационных технологий на деятельность организаций: применение методов ИТ для анализа и конструирования бизнес-процессов, например, объектно-ориентированное моделирование; появление новых бизнес-процессов, позволивших коренным образом изменить базовые правила работы организаций.

Бизнес-процесс - это совокупность взаимосвязанных задач или работ, для решения которых используются ресурсы предприятия с целью достижения определенных измеримых результатов или продукции для удовлетворения внутренних или внешних потребителей.

Внедрение единой системы управления бизнес-процессами, позволяющей комплексно решать основные задачи управления и развития, дает предприятию следующие возможности:

1) Система процессного управления позволяет оптимизировать систему общего корпоративного управления, сделать ее прозрачной для руководства и способной гибко реагировать на изменения внешней среды. Система процессного управления регламентирует:

- порядок планирования целей и деятельности;
- взаимодействие между бизнес-процессами и подразделениями предприятия; ответственность и полномочия исполнителей бизнес-процессов; систему показателей, характеризующих результативность и эффективность деятельности предприятия в целом и его бизнес-процессов;
- порядок рассмотрения результатов деятельности и принятие управленческих решений по устранению отклонений и достижению плановых показателей.

Внедрение на предприятии системы управления бизнес-процессами подразумевает работу по описанию и регламентации, в рамках которой:

- проводится распределение ответственности за результаты задач, входящих в состав бизнес-процессов;
- определяется система взаимодействия бизнес-процессов между собой и с внешними поставщиками и потребителями;
- определяются входные и выходные информационные массивы, необходимые для функционирования бизнес-процессов;
- устанавливаются показатели деятельности и эффективности бизнес-процессов.

2) Система процессного управления позволяет получить и использовать систему показателей и критериев оценки эффективности управления на каждом этапе производственной и управленческой цепочки.

3) Система управления бизнес-процессами обеспечивает уверенность предприятия в том, что существующая система управления нацелена на постоянное повышение эффективности и максимальный учет интересов заинтересованных сторон, поскольку система основана на измерении

показателей деятельности предприятия, планировании и достижении непрерывного улучшения результатов деятельности.

4) Внедрение процессного подхода гарантирует четко определенный порядок и ответственность за разработку, согласование, утверждение и ведение документации.

Методы реинжиниринга бизнес-процессов

Формирование миссии предполагает определение стратегии поведения предприятия на рынке в части расширения границ рынка или глубокого проникновения на рынок, диверсификации деятельности или повышения качества товаров и услуг, глобализации или локализации деятельности и т.д. В качестве основного метода формирования стратегии предприятия обычно используется метод анализа иерархий Саати. В качестве инструментальных средств анализа иерархий используются статические экспертные системы с возможностью обработки качественных (нечетких) оценок, такие, как Expert Choice, Guru, Level5.

Выбор сегментов рынка предполагает конкретизацию стратегических целей предприятия в части определения регионов, потребителей, каналов распределения продукции и услуг. Основными методами исследований на этом этапе выступают методы статистического анализа и прогнозирования рынков сбыта, нейронных сетей, интеллектуального анализа данных современных информационных хранилищ. Наиболее мощными инструментальными средствами анализа и прогнозирования для выявления основных сегментов рынка являются ППП SAS, SPSS, NeurOn-Line, Brain Maker, PolyAnalyst и др. [2, с. 25-27]

Формирование продуктовых портфелей для выделенных перспективных сегментов рынка предполагает оценку возможностей предприятия в плане эффективности распределения капиталовложений по различным проектам и продуктам. Для решения этой задачи обычно используются математические модели и методы оптимизации. Одним из наиболее известных средств бизнес-планирования является ППП Project

Expert, который позволяет проектировать и оценивать бизнес-планы предприятия для различных вариантов стратегий.

Проектирование бизнес-процессов. Выявленный на третьем этапе продуктовый портфель определяет содержание бизнес-процессов, для которых необходимо детально разработать технологию их осуществления с позиции достижения заданных критериев эффективности. Для выбора конкретных технологий требуется разработать статические и динамические модели бизнес-процессов, позволяющие выполнить оценку их эффективности.

Существуют различные методы и средства моделирования бизнес-процессов, которые в основном сводятся либо к функциональному (диаграммы рабочих потоков Oracle Designer 2000, SilverRun, Natural Engineering Workbench, функциональные диаграммы Design/IDEF), либо к объектно-ориентированному моделированию (язык UML, средство Natural Engineering Workbench).

Проектирование использования ресурсов. После графического построения статической модели бизнес-процесса требуется выполнить анализ ее реализации с позиции эффективности и достаточности ресурсов. Для этого используются методы стоимостного анализа функций и динамического имитационного моделирования. Так, статический стоимостной анализ функций позволяет выявить:

- Наиболее трудоемкие и затратные функции;
- Функции, не вносящие вклад в образование прибыли;
- Функции с низким коэффициентом использования ресурсов.

Стоимостной анализ функций реализуется либо с помощью средств CASE-технологий, таких, как Design/IDEF, либо с помощью систем комплексной автоматизации предприятий, например, R/3, либо с помощью специализированных программных продуктов, таких, как Easy ABC+.

Для динамического анализа бизнес-процесса используются методы имитационного моделирования, которые позволяют генерировать статистику

выполнения множества бизнес-процессов одного или нескольких типов за длительный период времени. При этом большое значение придается анализу узких мест в организации бизнес-процессов, связанных с перегрузкой ресурсов, образование очередей, или наоборот недогрузкой ресурсов. К известным средствам имитационного моделирования относят ППП РДО, Workflow Analyser, Pilgrim, Ithink, ReThink и др.

1.4.2. Инструментальные средства реинжиниринга бизнес-процессов

Для автоматизации моделирования процессов сложился специальный класс программных продуктов. Наиболее известными являются такие продукты, как ARIS, IDEF, AllFusion ERwin Process Modeler (старое название — BPwin), хотя в большом количестве случаев стандартных графических пакетов типа Microsoft Visio, текстового редактора и электронной таблицы бывает достаточно.

ARIS — методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций. Продукт и методология принадлежат немецкой компании Software AG. Любая организация в методологии ARIS рассматривается с пяти точек зрения: организационной, функциональной, обрабатываемых данных, структуры бизнес-процессов, продуктов и услуг. При этом каждая из этих точек зрения разделяется ещё на три подуровня: описание требований, описание спецификации, описание внедрения. Для описания бизнес-процессов предлагается использовать около 80 типов моделей, каждая из которых принадлежит тому или иному аспекту. ARIS предоставляет визуальный инструментарий для обеспечения наглядности моделей. Также инструментарий поставляется с набором референтных моделей, заранее разработанных для типичных процессов в различных отраслях. Общий принцип в инструментарии — возможность интеграции моделей разных типов в рамках одного репозитория посредством декомпозиции (детализации) объектов. [15, с. 110-119]

IDEF — методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования сложных систем, позволяет отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах. При этом широта и глубина обследования процессов в системе определяется самим разработчиком, что позволяет не перегружать создаваемую модель излишними данными. В настоящий момент к семейству IDEF можно отнести следующие стандарты:

- IDEF0
- IDEF1
- IDEF2
- IDEF3
- IDEF4
- IDEF5
- IDEF6
- IDEF7
- IDEF8
- IDEF9
- IDEF10 — IDEF14

All Fusion ERwin Process Modeler (старое название — BPwin) — CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных. Модели данных помогают визуализировать структуру данных, обеспечивая эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологий баз данных и среды развертывания.

Принципы и методы процессно-задачного, процессно-продуктового реинжиниринга бизнес-процессов предприятия

Процессно-задачная концепция бизнеса основана на следующих положениях:

1. Предприятие или бизнес представляет систему последовательно-параллельно взаимосвязанных бизнес-процессов, бизнес-процессы представляют систему последовательно-параллельно взаимосвязанных бизнес-задач.

2. Любой бизнес-процесс, бизнес-задача имеют входной поток используемых ресурсов (материальных, информационных, трудовых и др.).

3. Каждый бизнес-процесс, бизнес-задача имеют свои внутренние правила, алгоритмы, законы и технологию преобразования входов в выходы.

4. Бизнес-процесс, бизнес-задача имеют выходной продукт (поток).

5. Связи между бизнес-процессами реализуются как связи между бизнес-задачами.

6. Бизнес-процесс, бизнес-задача имеют свою систему показателей эффективности. Эти показатели эффективности характеризуют выходные их продукты.

7. Бизнес-процесс протекает во времени.

8. Бизнес-процесс, бизнес-задача имеют исполнителя, владельца.

9. Каждый бизнес-процесс может состоять из подпроцессов, бизнес-задача из подзадач.

10. Бизнес-задачи (подзадачи) решаются выполнением множества функций и работ.

Понятия бизнес-процесс и подпроцесс, а также бизнес-задача и подзадача определяют лишь отношения их вложенности и возможность их детализации, структурно они эквивалентны. Ширина охвата и глубина детализации определяют границы моделирования бизнеса. Бизнес-процесс отражает ширину охвата, а бизнес-задача - глубину детализации.

Важным является понятие бизнес-функции. Функция в данном случае рассматривается как подчиненный элемент по отношению к задаче, чаще она представляет, так называемую, работу-действие. Результаты выполнения

функции не представляют самостоятельной ценности для бизнес-процесса и поэтому функция не пригодна для совершенствования экономических и производственных показателей деятельности предприятий и управления ими. Принцип продуктивности при выделении бизнес-задач является основным критерием отличия функции от задачи.

Процессно-задачный подход при выделении бизнес-процессов и их состава (бизнес - задач) использует следующие принципы (критерий):

- 1) полноты состава бизнес-процессов, бизнес-задач;
- 2) детерминированности (определенности) содержания каждого бизнес-процесса, бизнес-задачи;
- 3) нормализованности (отсутствие пересечения задач бизнес-процессов);
- 4) связанности бизнес-процессов и бизнес-задач, т.е. наличие у них не менее одной связи с другими бизнес-процессами, бизнес-задачами;
- 5) автономности (логической завершенности бизнес-процесса, бизнес-задачи);
- 6) ресурсоемкости (наличия входных ресурсов);
- 7) продуктивности (наличия выходного продукта);
- 8) динамичности (зависимость входных ресурсов, внутренних правил, алгоритмов, законов и технологий, следовательно, и выходного продукта от времени). [17, с. 69-71]

У каждой бизнес-задачи существует продукт решения (материальный, информационный), поддающийся технико-экономической оценке. Именно эта оценка является основой оценки эффективности исполнителя и управляемого процесса, для принятия новых управленческих решений. Поэтому все бизнес-процессы должны иметь в своем составе бизнес-задачу «Расчет показателей эффективности бизнес-процесса». Без решения такой задачи не может быть управляемого процесса. Все показатели эффективности можно разделить на количественные и качественные показатели. Такие показатели существуют как для отдельных бизнес-задач, так и для всего

бизнес-процесса – агрегированные показатели. При этом, как правило, агрегированные количественные показатели формируются аддитивной сверткой, а качественные показатели – мультипликативной сверткой.

Информационная система предназначена для сбора, хранения, обработки данных и для повышения эффективности управления с помощью информационных технологий, подготовки и применения управленческих решений.

Функционирование информационной системы направлено на обеспечение:

1. Повышения эффективности управления.
2. Полноты информации для обеспечения принимаемых решений.
3. Представления информации с максимальной высокой скоростью.
4. Максимального удобства взаимодействия ИС.
5. Эффективного использования ресурсов ИС.
6. Сокращения расходов на создание, эксплуатацию и развитие ИС.
7. Максимального извлечения выходной информации.
8. Сокращения избыточности в базе данных.

С помощью проектированной информационной системы могут быть решены следующие проблемы предприятия:

1. Совершенствование системы кадрового обеспечения.
2. Проблема недостаточной автоматизации обработки входящей информации.
3. Упорядочение функций и совершенствование организационных структур управления для повышения оперативности управления.

Совершенствование организации управления путем объединения структурных подразделений перераспределения функций и исключения возможности их дублирования.

При рассмотрении процессов возникают следующие сложности. Процессы не удастся описывать так же легко, как, например, организационные иерархические структуры. У организационных

подразделений есть «имена» («производство продукции», «доставка продукции»), с ними связаны ответственные должностные лица («президент», «начальник подразделения»). Процессы же обычно невидимы, не имеют описаний и имен. Однако понятие «процесс» возникает более естественным образом, чем организационные иерархии, тогда, когда люди кооперируются для достижения обещанного клиенту результата. При традиционной структуре внимание фокусируется на заданиях, работах, людях, структурах, но не на процессах, хотя процессы пронизывают традиционные организационные структуры. [22, с. 50-57]

1.4.3. Бизнес-процессы ЛПЧУП «Санаторий Ижминводы»

1. Предоставление санаторных услуг - основной задачей является предоставление услуг и отражение всего лечебного процесса. В качестве бизнес задач будут отражены услуги компании.

1. Материально-техническое обеспечение – обеспечивает предприятие всеми видами материально-технических, сырьевых ресурсов. Поэтому имеет две составляющие:
 - обеспечение складскими запасами;
 - обеспечение всех рабочих мест необходимыми ресурсами.
2. Маркетинг – адаптирует деятельность предприятия к условиям рыночной экономики. Учитывая, что доход предприятия зависит от его положения на рынке, основными задачами маркетинга являются:
 - сегментирование рынка;
 - анализ рынка;
 - продвижение товаров и услуг;
 - разработка новых товаров и товарных свойств, востребованных на рынке.
3. Финансы – в состав бизнес-процесса входит:
 - бюджетирование;

- составление сметы доходов, а затем сметы расходов;
- нормирование;
- формирование цен и расчет себестоимости;
- распределение финансов по проектам, по статьям затрат;

4. Бухгалтерский учет – отражение во времени количественных характеристик, влияющих на финансовые результаты деятельности предприятия.

5. Кадровое обеспечение – включает в себя прием, перемещение, увольнение кадровых ресурсов, планирование потребности трудовых ресурсов по всем направлениям. В соответствии с решаемыми задачами и выполняемыми работами во всех бизнес-процессах, в соответствии с нормативами трудовых ресурсов производится планирование трудовых ресурсов по категориям, по бизнес-процессам.

6. Транспортное обеспечение – необходим для решения некоторых задач материально-технического обеспечения и сбыта. Основной задачей транспорта является перемещение ресурсов во времени. Можно выделить:

- технологический транспорт, используемый в основном в производстве;
- основной транспорт (автотранспорт, железнодорожный транспорт и т.п.).

7. Энергетическое обеспечение – это обеспечение производства предприятия всеми видами энергии (электрическая, тепловая, химическая, твердая, жидкая, газообразного видов, механическая, топливо, вода).

8. Ремонтное обеспечение – можно отнести к вспомогательному производству. К ремонтному обеспечению относятся работы и задачи текущего капитального ремонта основных и производственных фондов предприятия.

9. Социальное обеспечение – включает в себя содержание, развитие социально-культурного быта, строительство жилья, пенсионно-медицинское обеспечение.

10. Дополнительное обеспечение основной деятельности – юридическое обеспечение, обеспечение безопасности и т.д.

11. Информационное обеспечение – обеспечение всех бизнес-процессов предприятия, решение задач и принятие эффективных решений.

12. Управление – состоит из двух подпроцессов:

- стратегическое управление – определение стратегических задач, стратегических параметров деятельности предприятия, стратегическое планирование;

- оперативное управление – управление основными бизнес-процессами предприятия в краткосрочном периоде.[18, с. 78-91]

Основные задачи и работы бизнес-процессов предприятия

1. Предоставление санаторных услуг

2. МТО

2.1. Обеспечение складских запасов.

2.1.1. Планирование или расчет потребности в медицинском оборудовании, материалах, топливе, энергии и медикаментах;

2.1.2. Заключение договоров на поставку всех видов материально-технических ресурсов;

2.1.3. Пополнение складских запасов в соответствии с плановыми потребностями материально-технических ресурсов;

2.1.4. Количественный и качественный контроль поступающих материально-технических ресурсов;

2.1.5. Расчет потребностей в транспорте и составление заявок на транспорт;

2.1.6. Анализ складских запасов;

2.1.7. Контроль над соблюдением норм расхода сырьевых ресурсов.

2.2. МТО рабочих мест.

- 2.2.1. Составление план-графика обеспечения рабочих мест;
- 2.2.2. Планирование потребности в транспорте;
- 2.2.3. Учет перемещения материально-технических ресурсов;
- 2.2.4. Учет браков и работа с поставщиками по браку.

3. Маркетинг

3.1. Сегментирование рынка

3.1.1. Сегментирование рынка предоставляемых услуг по следующим основным признакам:

- географическому
- социально-экономическому

3.1.2. Определение сегментов рынка, целесообразных для данного предприятия, т.е. экономически обоснованные сегменты

3.2. Анализ рынка (по сегментам)

3.2.1. Анализ рынка используемых медикаментов (по ценам, по потребительским свойствам, по поставщикам);

3.2.2. Анализ рынка используемого оборудования (по ценам, по поставщикам, по использованию).

3.3. Продвижение товаров и услуг

3.3.1. Реклама (планирование, оценка эффективности, разработка новых видов рекламы);

3.3.2. Развитие сети дилеров (расчет потребности в дилерах по сегментам рынка, определение состава дилеров);

3.3.3. Организация интерактивных систем реализации услуг (на базе интернет-технологий);

3.3.4. Разработка адаптивных методов продвижения услуг для каждого конкретного сегмента (сетевая реализация, сетевой маркетинг, система льгот).

3.4. Разработка новых и современных услуг

3.4.1. Анализ потребительских свойств услуг конкурентов;

3.4.2. Анкетирование потенциальных клиентов;

- 3.4.3. Разработка диапазона параметров будущих услуг;
- 3.4.4. Оценка цен на проектируемую услугу.
- 4. Сбыт
 - 4.1.1. Оформление и бронирование путевки;
 - 4.1.2. Онлайн-оплата санаторной путевки;
 - 4.1.3. Заключение договоров на реализацию услуг;
 - 4.1.4. Учет заказов по договорам в базе санатория;
 - 4.1.5. Учет реализованных услуг в базе санатория;
- 5. Финансы
 - 5.1.1. Составление бизнес-планов по всем видам деятельности;
 - 5.1.2. Составление сметы доходов и сметы расходов;
 - 5.1.3. Распределение финансов по направлениям деятельности;
 - 5.1.4. Нормирование расходов;
 - 5.1.5. Расчет нормативной себестоимости продукции;
 - 5.1.6. Расчет фактической себестоимости продукции;
 - 5.1.7. Формирование договорных и розничных цен;
 - 5.1.8. Расчет показателей рентабельности деятельности предприятия по всем направлениям;
 - 5.1.9. Расчет показателей эффективности деятельности подразделений (эффективности решения задач);
 - 5.1.10. Контроль исполнения бюджета;
 - 5.1.11. Контроль платежей в государственный бюджет (контроль исполнения по налогам) и отношений с кредитными учреждениями.
- 6. Бухгалтерский учет
 - 6.1.1. Учет основных средств и нематериальных активов;
 - 6.1.2. Учет материально-производственных запасов;
 - 6.1.3. Начисление заработной платы;
 - 6.1.4. Учет затрат на производство;
 - 6.1.5. Учет готовой продукции;
 - 6.1.6. Учет финансовых операций;

- 6.1.7. Учет общехозяйственных расходов;
- 6.1.8. Учет транспортных расходов;
- 6.1.9. Учет ремонтного фонда;
- 6.1.10. Составление бухгалтерской отчетности.

7. Кадровое обеспечение

7.1.1. Расчет потребностей в трудовых ресурсах по всем производственным программам и бизнес-процессам в разрезе профессий, квалификаций с учетом используемых технологий;

7.1.2. Анализ трудовых ресурсов предприятий;

7.1.3. Прием на работу;

7.1.4. Учет движения трудовых ресурсов;

7.1.5. Подготовка, переподготовка, повышение квалификации трудовых ресурсов;

7.1.6. Увольнение.

8. Транспортное обеспечение

8.1.1. Планирование внешних транспортных перевозок сотрудников, клиентов и поставляемых медикаментов;

8.1.2. Планирование внутренних транспортных перевозок (планирование технологического транспорта);

8.1.3. Ремонт и техническое обслуживание транспортных средств;

8.1.4. Подготовка транспортной документации (путевые листы);

8.1.5. Учет транспортных затрат (ГСМ, рабочее время и т.д.);

8.1.6. Расчет технико-экономических показателей эксплуатации транспортных средств (рентабельность использования транспортных средств и т.д.);

8.1.7. Расчет потребностей транспортных средств;

8.1.8. Планирование и управление закупки транспортных средств.

9. Энергетическое обеспечение

9.1.1. Расчет потребностей предприятия в энергоресурсах;

9.1.2. Нормирование расходов;

9.1.3. Планирование капитального и текущего ремонта объектов энергообеспечения;

9.1.4. Разработка правил, контроль над соблюдением правил эксплуатации энергообеспечения;

9.1.5. Учет наличия и движения электрооборудования;

9.1.6. Учет затрат энергоресурсов;

9.1.7. Контроль над объемами расходования энергоресурсов;

9.1.8. Планирование развития энергообеспечения в соответствии с потребностями производства;

9.1.9. Текущее обслуживание электрооборудования, капитальный ремонт электрооборудования, закупка электрооборудования и т.д.

10. Ремонтное обеспечение

10.1.1. Планирование профилактических ремонтных работ по всем видам оборудования предприятия;

10.1.2. Планирование капитального ремонта;

10.1.3. Расчет потребностей ресурсов для плановых работ;

10.1.4. Организация ремонтных работ (документирование);

10.1.5. Проведение ремонтных работ;

10.1.6. Учет фактических ремонтных работ;

10.1.7. Расчет технико-экономических показателей ремонтных работ;

11. Социальное обеспечение

11.1.1. Охрана труда, обеспечение условий труда;

11.1.2. Медицинское обеспечение;

11.1.3. Организация, проведение спортивно-оздоровительных, корпоративных мероприятий.

12. Дополнительное обеспечение основной деятельности

14.1. Правовое обеспечение

14.1.1. Разработка нормативных актов и порядка обеспечения соблюдения законодательных норм (юридическая экспертиза);

14.1.2. Обеспечение инструментариями правовой деятельности;

14.2. Делопроизводство

14.2.1. Документооборот, типы и формы, владельцы, порядок согласования;

14.2.2. Учет и контроль движения документов;

14.2.3. Контроль исполнения;

14.2.4. Архивация, хранение и доступ к документам.

14.3. Охрана и безопасность жизнедеятельности предприятия

14.4. Хозяйственное обслуживание

13. Информационное обеспечение

13.1. Управление информационными ресурсами предприятия

13.1.1. Расчет и анализ потребности в информационных ресурсах:

- технических;
- математических
- информационных
- программных
- организационно-правовых
- трудовых

13.1.2. Ремонт и техническое обслуживание и модернизация оргтехники;

13.1.3. Информационная безопасность

13.2. Управление информационными потоками на предприятии

13.2.1. Реинжиниринг бизнес-процессов предприятия;

13.2.2. Проектирование информационной системы предприятия (информационных потоков);

13.2.3. Разработка и внедрение сегментов ИС (в том числе на базе интегрированных ИС);

13.2.4. Модернизация ИС, перепроектирование информационных потоков;

13.2.5. Анализ в информационном обеспечении взаимодействия бизнес-процессов.

13.2.6. Показатели эффективности решения задач и бизнес-процессов

14. Управление

14.1. Стратегическое управление

14.1.1. Определение стратегических задач, позиционирование предприятия в экономике региона, государства, во времени;

14.1.2. Стратегическое планирование финансов, развития основных средств и производственных фондов;

14.1.3. Стратегический мониторинг развития предприятия, рынка.

14.2. Среднесрочное управление

14.2.1. Бизнес-планирование;

14.2.2. Учет реализации бизнес-проектов;

14.2.3. Контроль, анализ и управление проектами;

14.3. Оперативное управление предприятием (бизнес-процессами предприятия)

14.3.1. Подготовкой производства;

14.3.2. Основным производством;

14.3.3. Вспомогательным производством;

14.3.4. МТО (материально-техническим обеспечением);

14.3.5. Финансами;

14.3.6. Бухгалтерским учетом

14.3.7. Маркетингом;

14.3.8. Сбытом;

14.3.9. Кадровым обеспечением;

14.3.10. Транспортном;

14.3.11. Энергообеспечением;

14.3.12. Ремонтным обеспечением;

14.3.13. Социальным обеспечением;

- 14.3.14. Дополнительным обеспечением основной деятельности;
- 14.3.15. Информационным обеспечением.

2. Предпроектный анализ деятельности ЛПЧУП «Санаторий Ижминводы»

2.1. Анализ функционирования санатория

2.1.1. Общая характеристика

Сегодня в России здоровье людей – проблема государственного масштаба, и одним из самых эффективных средств его укрепления является оздоровление в санаторно-курортных условиях. Ярким представителем современного высокотехнологичного оздоровительного сервиса является знаменитая старейшая здравница Татарстана «Ижевские минеральные воды. На протяжении многих лет здесь заслуженным успехом пользуется лечение, об эффективности которого свидетельствует то, что после долечивания на курорте больные в 3-4 раза чаще и в 1,5-2,5 раза быстрее возвращаются к производительному труду.

Санаторий расположен на высоком берегу реки Кама в окружении живописных лугов и лесов. Уникальность курорта «Ижевские минеральные воды» заключается в наличии редких вод для внутреннего и наружного применения, которые настаиваются в огромных известковых резервуарах из песчаника и известняка, различных глин с примесями кальцита, пирита, медного и свинцового блеска и особенно в голубоватых и зеленоватых глинах, отличающихся по полезному воздействию на человека. Минеральные воды создавались в подземной лаборатории самой природой многие миллионы лет. И теперь они пробиваются наружу сквозь трещины и рукотворные скважины.

Вода, получившая за свои исключительные качества фирменное название «Шифалы Су» («полезная вода») и в 2001 году на конкурсе минеральных вод в Италии получившая Золотую медаль, является основным благоприятным фактором в лечении заболеваний органов пищеварения. Она

имеет химический состав, схожий с показателями вод из знаменитых европейских источников Виши, Вильдунген и Фахинген. Вода питьевого назначения, отмеченная также признанием не одного поколения практикующих врачей, способна облегчить течение и даже излечить легкие формы таких заболеваний, как гастриты, язвенные болезни, холециститы, циститы, желчнокаменные и почечнокаменные болезни, остеохондрозы, невроты, невралгии, заболевания половой сферы, эндокринной и дыхательной систем. Эту поистине волшебную смесь используют для приготовления лечебных ванн.

Виды деятельности:

1. Первичный прием врача-терапевта
2. Повторный прием врача-терапевта
3. Консультация специалистов
4. Лабораторные исследования (по показаниям)
5. ЭКГ (по показаниям)
6. Питье минеральной воды
7. Лечебные ванны (один из видов)
8. Лечебные души (один из видов)
9. Тепловое лечение (грязевое или парафин)
10. Физиотерапия (по основному заболеванию)
11. Спелеотерапия (по показаниям)
12. Лечебная физкультура
13. Массаж (один из видов)
14. Фитотерапия (один из сборов)
15. Музыка- или ароматерапия
16. Сауна

Дополнительные процедуры:

- инфракрасный саунарий
- гирудотерапия
- мониторная очистка кишечника АМОК

- андрологический и гинекологический массажи
- мануальная терапия
- стоматологические услуги
- УЗИ, ЭКГ (суточное мониторирование)
- массажная кушетка
- криотерапия
- биохимические анализы крови
- массаж ручной и точечный

2.1.2. Анализ технико-экономических показателей предприятия

Основные средства на 15.12.2013. составляют 35635,23 тыс.руб.

Результаты работы за 2013 год:

Таблица 2.1 – Показатели деятельности

Показатель	Значение (тыс. руб)
Итого по разделу I - Внеоборотные активы	234367
Нематериальные активы	0
Результаты исследований и разработок	0
Нематериальные поисковые активы	0
Материальные поисковые активы	0
Основные средства	174820
Доходные вложения в материальные ценности	0
Показатель	Значение (тыс. руб)
Финансовые вложения	0
Отложенные налоговые активы	75423
Прочие внеоборотные активы	25418
Итого по разделу II - Оборотные активы	208446
Запасы	118819
Налог на добавленную стоимость по приобретенным ценностям	1429
Дебиторская задолженность	54863
Финансовые вложения (за исключением денежных эквивалентов)	0

Показатель	Значение (тыс. руб)
Денежные средства и денежные эквиваленты	44456
Прочие оборотные активы	12
Итого по разделу III - Собственный капитал	285195
Уставный капитал (складочный капитал, уставный фонд, вклады товарищей)	25946
Собственные акции, выкупленные у акционеров	0
Переоценка внеоборотных активов	0
Добавочный капитал (без переоценки)	534242
Резервный капитал	0
Нераспределенная прибыль (непокрытый убыток)	274993
Итого по разделу IV - Долгосрочные пассивы	1561
Заемные средства	0
Отложенные налоговые обязательства	1561
Оценочные обязательства	0
Прочие обязательства	0
Итого по разделу V - Краткосрочные пассивы	197254
Заемные средства	122427
Кредиторская задолженность	69993
Доходы будущих периодов	0
Оценочные обязательства	4834
Прочие обязательства	0
БАЛАНС (актив)	484010
БАЛАНС (пассив)	484010
Валовая прибыль (убыток)	-65706
Выручка	228303
Себестоимость продаж	294009
Прибыль (убыток) от продаж	-109493
Коммерческие расходы	8537
Управленческие расходы	35250
Прибыль (убыток) до налогообложения	-121731
Доходы от участия в других организациях	0
Проценты к получению	992
Проценты к уплате	15112

Показатель	Значение (тыс. руб)
Прочие доходы	23995
Прочие расходы	22113
Чистая прибыль (убыток)	-98212
Текущий налог на прибыль	0
постоянные налоговые обязательства (активы)	0
Изменение отложенных налоговых обязательств	982
Изменение отложенных налоговых активов	24501
Прочее	0

В таблице 2.1 мы видим что на конец 2014 года совокупный финансовый результат периода составил -98212 тыс.руб. Результат от прочих операций, не включаемый в чистую прибыль (убыток) периода равен 0. Чистые активы равны 285194 тыс.руб.

2.1.3. Состав и структура действующей информационной системы

Информационные системы предприятия - это комплекс аппаратных и программных средств, позволяющие автоматизировать процедуры и процессы в повседневной деятельности предприятия, повышая производительность труда сотрудников и предоставляя возможность руководителям организации оперативно и адекватно принимать решения по управлению и планированию деятельности предприятия.

Современные информационные системы являются неотъемлемой частью инфраструктуры управления предприятием и содержат в себе различные инструменты обеспечения и управления бизнес-процессами. [21, 335 с.]

В настоящий момент в компании, в эксплуатации находится система "1С: Бухгалтерия 8.2.". Это комплексная конфигурация "Бухгалтерия; Торговля; Склад; Зарплата; Кадры" представляет собой универсальную программу конструктор, которая позволяет вести учет в одной

информационной базе от имени нескольких организаций. Так же используется система MSOffice.

2.2. Анализ бизнес-процесса «Сбыт»

Рассматриваются модели бизнес-процессов и их использование в качестве инструмента повышения эффективности управления основной деятельностью и как вспомогательное средство для анализа и планирования деятельности предприятия на основе моделей IDEF0 и IDEF3. Модель бизнес-процесса позволяет заранее дать оценку затрат ресурсов и времени на выполнение каждой конкретной задачи бизнес-процесса, т.е. провести так называемое временное нормирование задач и работ. В свою очередь, это нормирование даст и управленческий, и экономический эффекты, т.е. позволит:

- Определить действительную стоимость производства продукта на выходе из задач (или бизнес-процесса);
- Повысить эффективность принятия управленческих решений по каждой конкретной задаче бизнес – процесса или по всему бизнес–процессу в целом;
- Идентифицировать задачи и работы, которые должны быть улучшены в первую очередь.

Схема в IDEF0 позволяет наглядно структурировать процессы организации и графически отобразить взаимодействие между этими процессами. Особенно важно, что в идеологии IDEF0 при помощи специальных стрелок можно показывать управленческие воздействия, что дает возможность описывать систему управления процессами и организаций.

Используя вместе модели IDEF0 и IDEF3 можно производить календарное и ресурсное планирование и нормирование выполнения работ, выявлять отклонения от плана и анализировать их. [12, 110 с.]

Организационно-логическая сущность решения задач бизнес-процесса (IDEF3) Сбыт

Подсистема предназначена для процесса онлайн-бронирования путевок и операции по их подбору. Целью выделения подсистемы является более эффективный процесс оформления путевок на предприятии, а также выполнение контроля бронирования в соответствии с запланированными сроками.

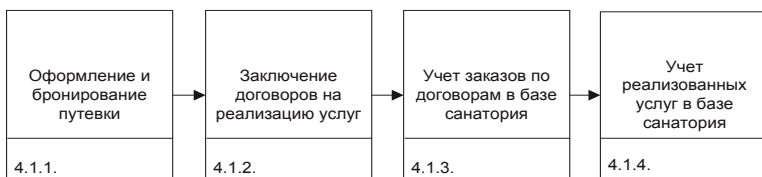


Рисунок 2.2 - Организационно-логическая сущность подсистемы управления «Сбыт» (IDEF3)

Взаимосвязь бизнес-процессов рассматривается как взаимосвязь задач, входящих в состав этих бизнес-процессов. Основой информационных связей между задачами являются информационные массивы. Любая задача имеет входные информационные массивы и результирующие выходные информационные массивы. Кроме того, задача может иметь локальные информационные массивы, используемые только этой задачей. Выходные информационные массивы можно использовать для решения других задач или для принятия управленческих решений. [13, с. 224-225]

Решение задач любого бизнес-процесса или выполнение комплекса работ требует наличия:

1. Входных информационных массивов.
2. Алгоритма решения задач, выполнения работ.
3. Выходных информационных массивов.

Таблица 2.2 - Таблица потоков

№	Входы		Бизнес-задачи		Выходы	
	Управление, входные ресурсы, продукты других б.з., механизмы	Тип входа	№	Наименование	Класс задачи	Наименование выходных бизнес-продуктов
4. Сбыт						
1.1.10.	План проведения санаторных услуг	У	4.1.1.	Оформление и бронирование путевки	А	Заключение договоров на реализацию услуг
3.2.1.2.	Потенциальные потребители, заявленные службой маркетинга	В				
	Сайт санатория Ижмивольды	М				
	Торговый агент, представитель	М				
14.3.8.1.	Приказы, распоряжения по сбыту путевок	У	4.1.2.	Заключение договоров на реализацию услуг	А	Ключи от номера и санаторная книжка
4.1.1.	Оформление и бронирование путевки	В				
	Данные о клиентах	В				
	Заместитель коммерческого директора	М				
4.1.3.	Ключи от номера и санаторная книжка	У	4.1.3.	Учет заказов по договорам в базе санатория	А	Учет заказов по договорам в базе санатория
	Договора	У				
	Заявка клиента	В				
	Офис-менеджер	М				
	Общий план реализованных услуг	У	4.1.4.	Учет реализованных услуг в базе санатория	А	
4.1.3.	Учет заказов по договорам в базе санатория	В				
	Менеджер по продажам, помощник	М				

Для описания схемы создания стоимости используется модель IDEF0.

Методология функционального моделирования IDEF0 — это технология описания системы в целом как множества взаимозависимых действий, или функций. Важно отметить функциональную направленность IDEF0 — функции системы исследуются независимо от объектов, которые обеспечивают их выполнение. "Функциональная" точка зрения позволяет четко отделить аспекты назначения системы от аспектов ее физической реализации.

Наиболее часто IDEF0 применяется как технология исследования и проектирования систем на логическом уровне. По этой причине он, как правило, используется на ранних этапах разработки проекта, до IDEF3 моделирования для сбора данных и моделирования процесса "как есть". Результаты IDEF0 анализа могут применяться при проведении проектирования с использованием моделей IDEF3 и диаграмм потоков данных. [9, с. 50-51]

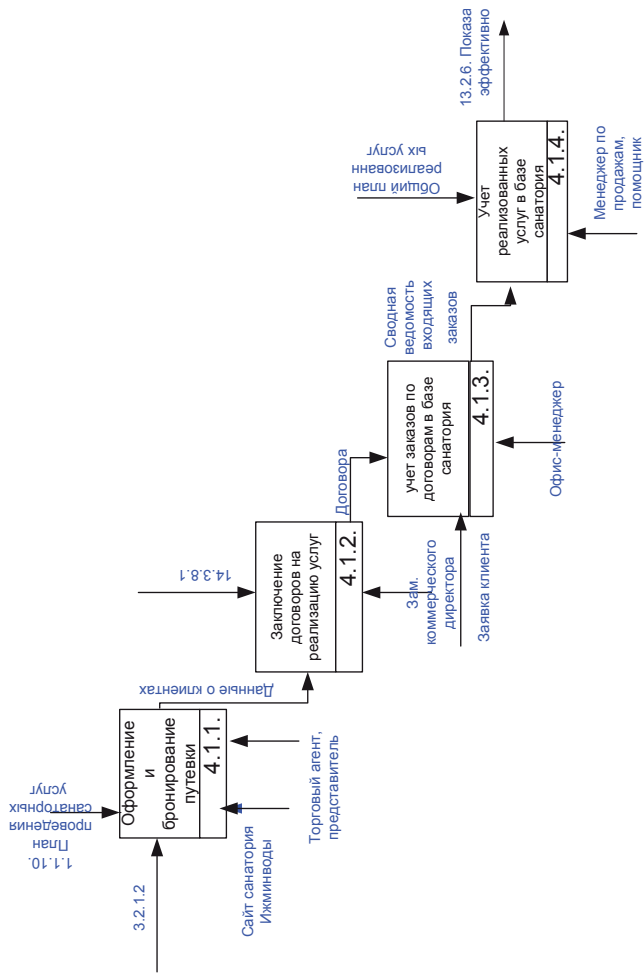


Рисунок 2.3 – Схема информационных связей задач подсистемы «Сбыт» (IDEF0)

2.3. Основные проблемы бизнес-процесса «Сбыт»

Исследование бизнес-процесса санатория «Ижминводы» и его рыночное окружение представляет собой систематические мероприятия по получению оперативной информации в отношении данных, описывающих текущее или перспективное (прогнозное) состояние самого бизнес-процесса, а также анализ и отражение данных в форме, необходимой для решения стратегических и тактических задач, стоящих перед санаторием. Проведенные в плановом порядке исследования дают возможность получать конкурентные преимущества, снижать финансовые и коммерческие риски предпринимательской деятельности, определять отношение клиентов к оказываемой услуге, давать оценку стратегической и тактической деятельности санатория, повышать эффективность коммуникационных рыночных мероприятий, определять оптимальные сегменты позиционирования процессов, характер их жизненного цикла.

Для оформления путевки у данного санатория есть сайт, но через него онлайн-оплату сделать не возможно, существует лишь функция бронирования. В качестве проекта моделирования бизнес-процесса санатория «Ижминводы» я предлагаю добавить дополнительную задачу онлайн-оплаты. А так же для привлечения более молодого и современного контингента осуществлять бронирование и оплату путевки через мобильное устройство. Это позволит будущему отдыхающему санатория «Ижминводы» легко и быстро организовать себе отдых.

Данный бизнес процесс, в данном случае бизнес-процесс «Сбыт» будет усовершенствована новыми задачами.

3. Проектирование информационной системы ЛПЧУП «Санаторий Ижминводы»

3.1. Характеристика проекта

1. Требования к структуре программного обеспечения и общая информация.

Язык интерфейса приложения русский, в будущем можно предусмотреть добавление еще 2-3 языков. Разработанное программное обеспечение должно быть совместимо с Android, серверная часть, будет написана на языке программирования PHP и реализована на базе сервера nginx с поддержкой MySQL. ОС сервера freeBsd. Клиентская часть будет состоять из следующих функциональных модулей:

- Модуль регистрации
- Модуль объектов
- Модуль оплаты
- Модуль маршрутов

2. Внешний вид и описание взаимодействия пользователей с приложением.

Внешний вид приложения составляют один активити и семь фрагментов в нем, каждый имеет собственный функционал и набор активных элементов.

- 1) Фрагмент «Приветствие»
- 2) Фрагмент «Выбор даты»
- 3) Фрагмент «Выбор номера»
- 4) Фрагмент «Выбор оплаты»
- 5) Фрагмент «Личные данные»
- 6) Фрагмент «Контакты»
- 7) Фрагмент «Успешное оформление»

3. Требования к функциям

- 1) Фрагмент «Приветствие»

Данный фрагмент будет представлять собой вид санатория с диалоговым окном будущей скидки. Так же будут две кнопки одна для перехода в следующий фрагмент «Выбора даты», другая для перехода к контактам санатория.

2) Фрагмент «Выбор даты»

Будет появляться окно с выбором конкретных дат проживания «день, месяц, год». Начальная и конечная даты.

3) Фрагмент «Выбор номера»

Будет представлен выбор номера, с указанием количества комнат, класса и ценой без учета скидки. Можно будет выбрать от одного до нескольких комнат с учетом поселения большого количества людей. Ниже будет указана общая сумма с учетом скидки и кнопка для перехода на следующий фрагмент.

4) Фрагмент «Выбор оплаты»

Будет представлено два метода оплаты. Онлайн картой и наличными через ближайшее отделение банка. Для оплаты наличными нужны реквизиты, которые вам сообщит менеджер по телефону, когда удостовериться в 100%-ом заселении. Оплата должна произвестись в течении 24-х часов, в противном случае путевка будет аннулирована. После оплаты клиент должен связаться с менеджером и сообщить номер квитанции. Это нужно во избежание издержек банка.

5) Фрагмент «Личные данные»

В данном фрагменте заказчик указывает свои данные, такие как ФИО, электронная почта, номер телефона и реквизиты карты, если оплата проходит безналичкой. Ниже будет кнопка для перехода на следующий фрагмент.

6) Фрагмент «Контакты»

Этот фрагмент будет представлять собой окно с указанием подробной информацией о санатории. Место расположения, почтовый адрес номер телефона по которому можно связаться с менеджером по продажам и электронный адрес. Также будет кнопка с переходом на карту навигатор

«Google maps». Для этого перехода нужно иметь данное приложение на своем мобильном устройстве. Эта ссылка делается для того чтобы было легче добраться до санатория, так как если клиент едет в первый раз, то это может доставить некоторые проблемы.

7) Фрагмент «Успешное оформление»

Будет текст с благодарностью оформления через наше приложение и две кнопки для перехода в главное меню и контакты.

3.2. Модель усовершенствованного бизнес-процесса «Сбыт»

Для более эффективной работы отдела продажи путевок необходима возможность оформления и оплаты онлайн, то есть через мобильное приложение. Несмотря на то, что многие функциональные возможности подсистемы управления «Сбыт», подсистемы «Продажи путевок» формализованы и автоматизированы, в системе отсутствует возможность онлайн оплаты.

Автоматизировать данную задачу можно с помощью:

- Электронные формы заявок.
- Формализация движения заявки, ее утверждения у нужных ролей, которые организуют все необходимые оповещения и обеспечит необходимый уровень безопасности и аудита.
- Отчетность (благодаря формализованному процессу), которая позволит более качественно принимать решения и анализировать текущую ситуацию.

Задача будет состоять в том, чтобы создать приложение, которое будет позволять подавать заявки через мобильное устройство и там же его оплачивать. Клиент должен пройти все этапы оформления заявки и на выходе будет создаваться отчет для заинтересованных лиц.

Для усовершенствованного бизнес процесса требуется новая модель организационно-логической сущности и алгоритм решения ее задач. Где наглядно будет показана работа данного процесса. [4, с. 50-58]

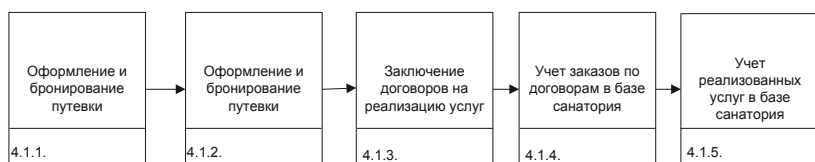


Рисунок 3.1 - Организационно-логическая сущность бизнес-процесса «Сбыт» (IDEF3)

Взаимосвязь бизнес-процессов рассматривается как взаимосвязь задач, входящих в состав этих бизнес-процессов. Основой информационных связей между задачами являются информационные массивы. Любая задача имеет входные информационные массивы и результирующие выходные информационные массивы. Кроме того, задача может иметь локальные информационные массивы, используемые только этой задачей. Выходные информационные массивы можно использовать для решения других задач или для принятия управленческих решений.

Решение задач любого бизнес-процесса или выполнение комплекса работ требует наличия:

2. Входных информационных массивов.
3. Алгоритма решения задач, выполнения работ.
4. Выходных информационных массивов.

В таблице 3.1. представлены структурные связи потоков бизнес-процесса «Сбыт» предприятия ЛПЧУП «Санаторий Ижминводы».

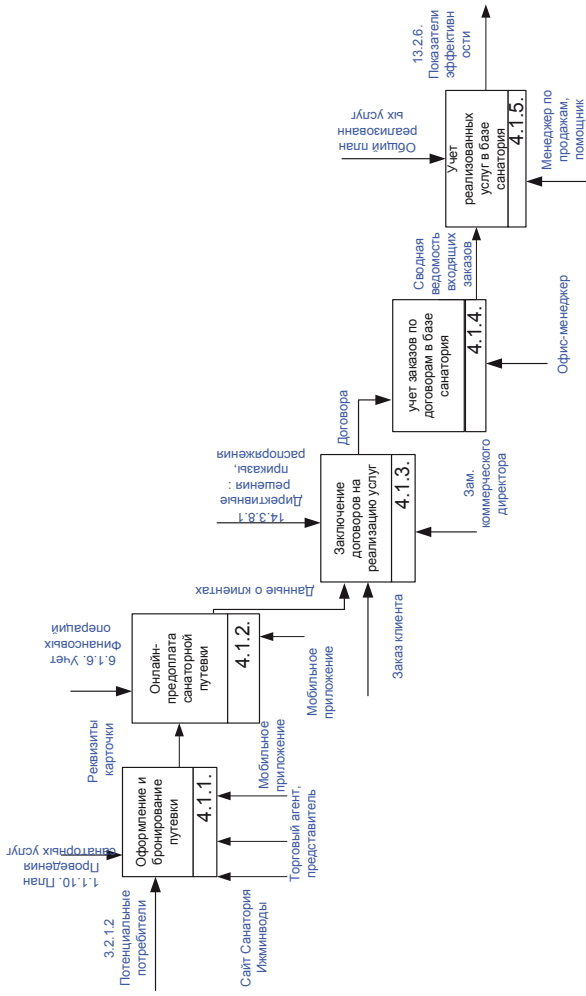


Рисунок 3.2 - Усовершенствованная схема информационных связей задач подсистемы «Сбыт» (IDEF0)

3.3. Разработка и программная реализация задач бизнес-процесса

«Сбыт»

3.3.1. Алгоритм решения задач

Для того чтобы определить процесс, необходимо его с чем-то отождествить, опознать или создать модель объекта, отображающую закономерности, присущие реальному объекту – оригиналу.

Применительно к процессам задача идентификации сводится к выбору простого и четко различимого средства идентификации в виде цифрового, графического или словесного обозначения, символа, цветовой пометки и т. п. на документах по процессу, дискетах и других носителях информации о процессах. Это позволяет быстро и однозначно опознать конкретный процесс в имеющейся совокупности процессов и определить порядок их выполнения

Дальнейшим шагом в идентификации процессов будет создание формализованных моделей, отражающих последовательные этапы и стадии выполнения процесса, их взаимосвязь и взаимодействие. Такие модели могут быть представлены в виде текстового описания, блок-схем, карт, графиков, алгоритмов, диаграмм и в виде их комбинаций. Модели должны быть как можно более простыми и понятными, но в то же время полными и исчерпывающими.

Идеальной же идентификацией процесса является создание его математической модели, устанавливающей взаимосвязь входных и выходных параметров и учитывающей все определяющие условия. [23, с.40-41]

Блок-схема представляет собой графическое описание потока процессов. Преимущества блок-схемы заключаются в том, что графическое представление объекта значительно легче понять, чем словесное его описание. Самый распространенный способ графического представления – использование различных символов для обозначения различных действий. Символы блок-схемы не нормируются, поэтому каждый автор выбирает их, как правило, по собственному усмотрению. Обычно в качестве символов используются наиболее простые геометрические фигуры.

В усовершенствованном бизнес-процессе «Сбыт» добавились две новые задачи:

- Оформление и бронирование путевки
- Онлайн-предоплата санаторной путевки

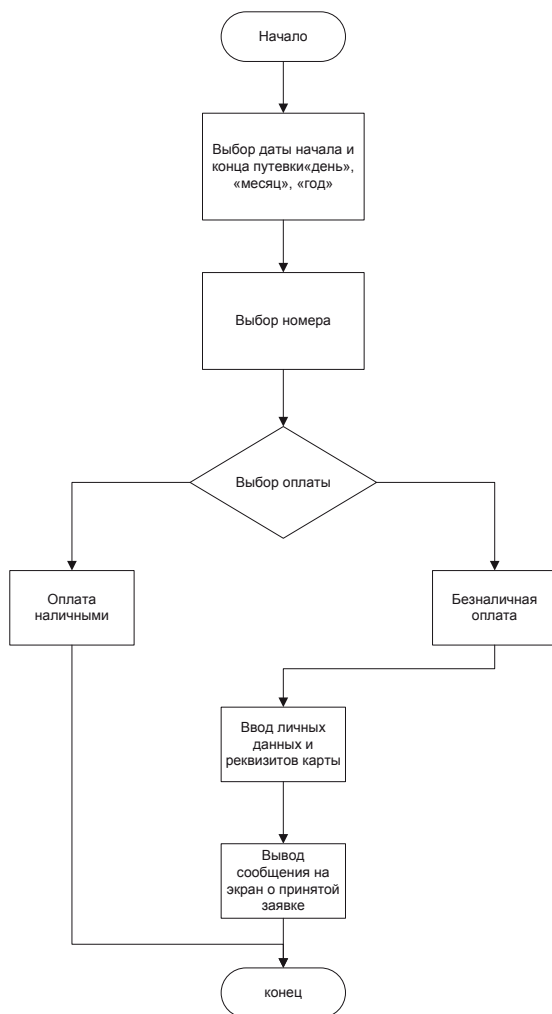


Рисунок 3.3 - Оформление и бронирование путевки

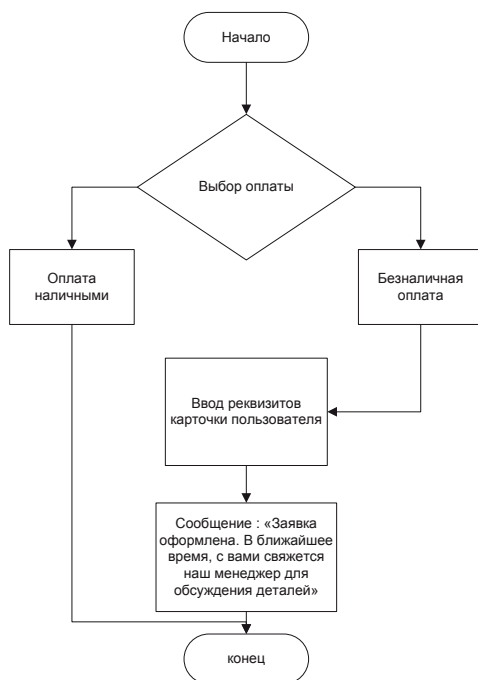


Рисунок 3.4 - Онлайн-предоплата санаторной путевки

3.3.2. Программная реализация задач бизнес-процесса «Сбыт»

1. Обязательные условия для работы приложения

Соединение с интернетом

2. Принцип работы приложения

Пользовательский интерфейс

Приложение имеет всего одно Activity с определенным набором Фрагментов (Fragments), которые в свою очередь находятся в ViewPager'е (стандартный элемент Android sdk который позволяет переключать объекты View).

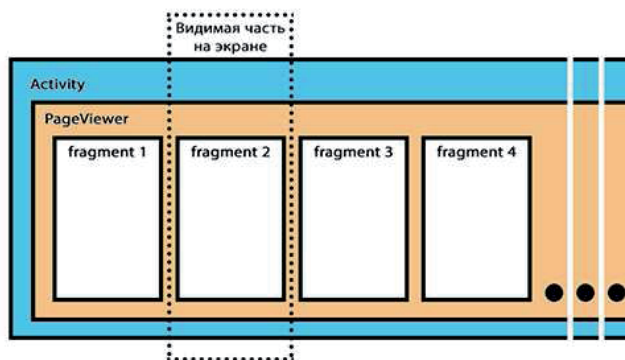


Рисунок 3.5 – Архитектура Android

Переключение между фрагментами происходит с помощью кнопок навигации внутри фрагментов. Обработчик прикосновений в PageViewer переопределен, для того что-бы отключить ненужный нам Swipe, потому что стандартной функций для ее отключения нет.

Каждый фрагмент имеет свой заранее заготовленный Слой (Layout).

Фрагменты, в имени которых встречается “Application” используются для оформления заявки.

Всего фрагментов 7:

- FragmentWelcome фрагмент приветствия
- FragmentApplicationTimePick фрагмент для выбора времени прибытия в санатории.
- FragmentApplicationServices фрагмент для выбора комнат.
- FragmentApplicationContacts фрагмент для указания контактов.
- FragmentApplicationPay фрагмент для выбора оплаты
- FragmentApplicationThankYou фрагмент сообщающий об успешном оформлении заявки.
- FragmentAbout фрагмент с контактами и схемой проезда.

3. Карта фрагментов



Рисунок 3.6 – Фрагментное описание мобильного приложения

4. Карта взаимодействия с сервером

Серверная часть приложения

Сервер имеет 2 метода для взаимодействия:

- Метод `getservices.php` для получения списка свободных комнат
- Метод `sendapplication.php` для регистрации заявки

В свою очередь, сервер подключен к базе данных 1С санатория через драйвер.

5. Листинг классов приложения

Классы программы разложены по соответствующим пакетам (папкам):

1) Activity

Находятся в `com.sanatoriy.activites`.

В приложении всего одно активити - `MainActivity.java`

`MainActivity.java`

Здесь мы создаем объекты необходимых фрагментов и кладем их в `ArrayList`, для того что бы обработать адаптером для `PageViewer`.

Так же с помощью здесь с помощью функции `setPage(Class<T> fragment)` мы переключаем отображаемый фрагмент, и обрабатываем кнопку “Назад” переопределенной функцией `finish()`.

2) Adapters

Находятся в `com.sanatoriy.adapters`.

Адаптеры нужны для адаптирования массивов объектов под некоторые `View`, такие как `ListView`, `GridView`, `PageViewer` и тд.

`ListViewApplicationAdapter.java`

Адаптер для списка комнат.

Особенности: Здесь мы используем свой `CheckBox` вместо стандартного, стандартный не подходит под необходимые требования. А именно, в элементе `ListView` обрабатывается прикосновение только по самому `CheckBox` вместо всего элемента `ListView`, что вызывает неудобства. Поэтому мы добавили свой `ImageView`, у которого есть 2 состояния: `Checked`, `Unchecked`.

ViewPagerAdapter.java

Адаптер для фрагментов в Activity

3) Fragments

Находятся в com.sanatoriy.fragments.

FragmentWelcome.java

Фрагмент приветствия, отображается при запуске приложения xml фрагмента находится в res/layout/fragment_welcome.xml

Внутри фрагмента 2 кнопки:

- Кнопка для перехода к фрагменту для оформления заявки.
- Кнопка для перехода к фрагменту контактов санатория.

При запуске приложения на экран отображается окно приветствия с указанием будущей скидкой при оформлении путевки. Так же будут две кнопки для перехода на фрагмент информации о санатории и кнопка для оформления путевки. Далее при нажатии кнопки оформления появится окно с выбором дат проживания в санатории.

FragmentApplicationTimePick.java

Фрагмент для выбора времени периода проживания. xml фрагмента находится в res/layout/fragment_application_timepick.xml. Здесь мы выбираем дату начала и конца прибытия в санатории. Исходя из введенной информации, будет формироваться список доступных, на этот промежуток времени, комнат.

Внутри фрагмента 3 кнопки:

- 2 кнопки для выбора даты
- Кнопка “Далее” для запроса доступных комнат и перехода к их списку



Рисунок 3.7 - FragmentWelcome



Рисунок 3.8 -
FragmentApplicationTimePick

FragmentApplicationServices.java

Фрагмент со списком доступных комнат. xml фрагмента находится в res/layout/fragment_application_services.xml. Здесь мы выбираем подходящие комнаты для бронирования.

Внутри фрагмента кнопка и список:

- Список комнат с элементами “checkbox”
- Кнопка “Далее” для перехода к фрагменту для заполнения контактных данных

FragmentApplicationContacts.java

Фрагмент для указания контактных данных xml фрагмента находится в res/layout/fragment_application_contacts.xml. Здесь мы вводим свои контактные данные.

Внутри фрагмента кнопка и 4 поля для ввода текста:

- Текстовые поля для ввода “имя”, ”фамилия”, ”email”, ”телефон”

- Кнопка “Далее” для перехода к фрагменту для указания платежной информации.

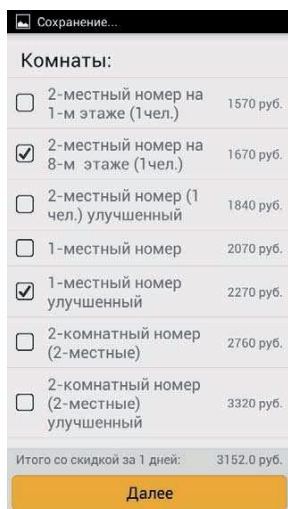


Рисунок 3.9 - FragmentApplicationServices

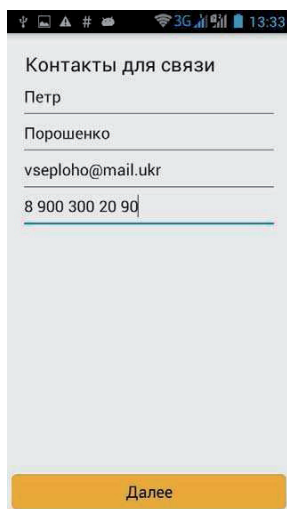


Рисунок 3.10 -

FragmentApplicationContacts

FragmentApplicationPay.java

Фрагмент для указания информации о методе оплаты. xml фрагмента находится в res/layout/fragment_application_pay.xml. Здесь мы выбираем метод оплаты.

На данный момент 2 варианта для оплаты:

- Оплата картой
- Оплата наличными в течении 24 часов

Фрагмент имеет 2 кнопки для выбора метода оплаты и в зависимости от выбранного метода отображаются необходимые для заполнения поля, а так же кнопка для отправления заявки на сервер.

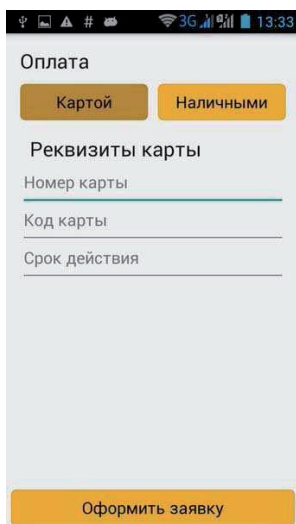


Рисунок 3.11 - FragmentApplicationPay
(карта)



Рисунок 3.12 - FragmentApplicationPay
(наличные)

FragmentApplicationThankYou.java

Фрагмент сообщающий об успешном оформлении заявки. xml фрагмента находится в res/layout/fragment_application_thankyou.xml.

Фрагмент отображается после успешного отправления заявки.

Внутри фрагмента 2 кнопки:

- Кнопка “О нас” для перехода к фрагменту с информацией о санатории “FragmentAbout”
- Кнопка “Главное меню” для перехода к первому фрагменту “FragmentWelcome”

FragmentAbout.java

Фрагмент с информацией о санатории. xml фрагмента находится в res/layout/fragment_about.xml. Внутри фрагмента одна кнопка “Как нас найти?” открывающая навигатор Google Maps до санатория.



Рисунок 3.13 -

FragmentApplicationThankYou

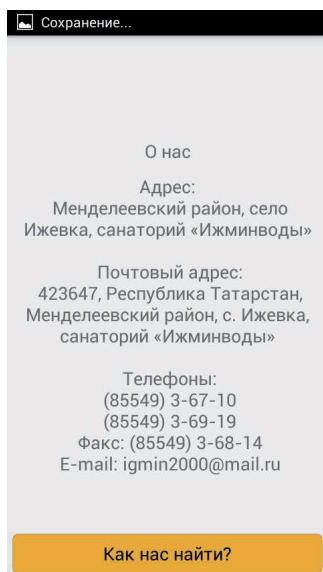


Рисунок 3.14 - FragmentAbout

Если вы производите онлайн-предоплату на ваш электронный адрес придет лист об оплате услуг (путевка) с логотипом санатория и данными об оплате.

ПУТЕВКА
об онлайн-предоплате курортно-санаторных услуг
№ _____

от " __ " _____ 20__ г.

Выдана путевка _____
(Ф.И.О)

В том, что он (она) оплатил(а) курортно-санаторные услуги стоимостью _____
(сумма)

оказанные: ему (ей), супруге (у), сыну (дочери), матери (отцу) _____, код услуги _____

Дата оплаты " __ " _____ 20__ г.

ФИО выдавшего справку _____

N телефона (____) _____



Тестирование

В процессе разработки приложения производилось поэтапное тестирование с целью выявления программных ошибок и несоответствий ТЗ (техническому заданию). Для этого нами были созданы эмуляторы смартфона и планшета с разными диагоналями экрана для разных версий Android. Тестируемый программный продукт последовательно запускался на этих эмуляторах, его поведение анализировалось, и при необходимости по результатам анализа вносились изменения в код. Для тестирования отдельных модулей работы с базой данных в текст программы были внесены специальные функции, позволяющие анализировать базу данных и, при подозрении на ошибку, выводящие сообщение в системный журнал. Они также известны как юнит-тесты. Например, при изменениях в базе данных проводилась проверка целостности базы данных (проверка на соответствие ключей – индексам), после чего при необходимости выводилось сообщение в системный log.

Были проведены приведенные ниже тесты.

1. Каждая активность была подвергнута юнит-тестированию с целью выявления ошибок, вызванных несоответствием ожидаемых и полученных параметров. Для этого для каждой активности был создан специальный юнит-класс, посылающий в активность различные верные и неверные параметры. При аномальном поведении активности или ее сбое, проводился анализ поведения и ошибка исправлялась.

2. В базу данных намеренно вносились недопустимые данные в соответствующие поля, которые могли быть неверно интерпретированы программой. Затем мною анализировалось поведение активности во время обработки недопустимых данных.

3. Приложение было запущено на устройствах, работающих под управлением разных версий Android с целью выявления особенностей работы приложения, запущенного в разных операционных системах.

4. После завершения цикла разработки, программный продукт тестировался на реальных устройствах. По результатам тестирования была добавлена виртуальная кнопка «О нас» для устройств, не имеющих аппаратных кнопок.

Возможные варианты развития программного приложения:

1. интеграция приложения со сторонними базами данных;
2. реализация двухкомпонентной базы данных (локальной базы данных и базы данных в облаке);
3. реализация возможности подгрузки пакетной информации;
4. обновление мест интереса с использованием удаленного сервера;
5. реализация приложения на базе IOS для пользователей продукции Apple.
6. реализация приложения для планшетов с более широким диалоговым окном.

3.4. Экономическое обоснование разработки программного обеспечения

3.4.1. Бюджет затрат на разработку программного обеспечения

Процесс разработки задач можно разделить на следующие этапы, представленные в таблице 3.2.

Таблица 3.2 - Этапы работ по проекту разработки программного обеспечения

№	Этапы работ по разработке ПО	№	Информационный продукт	Продолжи- тельность реализации задачи, час	Состав рабочей группы, должность	Трудоемкость, чел-час	
						По участникам	Всего
1	2	3	4	5	6	6	8
1.	Предпроектный анализ						
1.1	Анализ бизнес-процесса «Сбыт»						
1.1.1	Общая характеристика бизнес-процесса «Сбыт»	1.1.1.1	Пояснительная записка «бизнес-процесса «Сбыт»»	34	Руководитель IT-службы	4	46
					Программист	7	
					Директор по экономике	3	
					Экономист	32	
1.1.2	Построение IDEFO-диаграммы бизнес-процесса «Сбыт»	1.1.2.1	Диаграмма "IDEFO-диаграмма «бизнес-процесса «Сбыт»»	9	Руководитель IT-службы	2	17
					Программист	8	
					Директор по экономике	1	
					Экономист	6	
1.1.3	Построение IDEF3-диаграммы бизнес-процесса «Сбыт»	1.1.3.1	Диаграмма "IDEF3-диаграмма бизнес-процесса «Сбыт»»	7	Руководитель IT-службы	1	16
					Программист	8	
					Директор по экономике	1	
					Экономист	6	

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
1.1.4	Построение сетевого графа бизнес-процесса «Сбыт» и диаграммы Ганта	1.1.4.1	График Сетевой граф бизнес-процесса «Сбыт»	9	Руководитель IT-службы	1	19
		1.1.4.2	Диаграмма "Диаграмма Ганта бизнес-процесса «Сбыт»		Программист	8	
					Директор по экономике	2	
1.1.5	Построение WFD-диаграммы бизнес-задач по принципу "as is"	1.1.5.1	Диаграмма "WFD-диаграмма бизнес - задач по принципу "as is"	7	Экономист	8	
1.1.6	Формирование сводного отчета об эффективности реализации бизнес-процесса «Сбыт» и бизнес-задач	1.1.6.1	Сводный отчет "Эффективность реализации бизнес-процесса «Сбыт»	8	Руководитель IT-службы	1	16
					Программист	6	
					Директор по экономике	1	
					Экономист	8	
1.2	Анализ информационной системы бизнес-процесса «Сбыт»						
1.2.1	Характеристика действующей информационной системы. Определение состава подсистемы и задач	1.2.1.1	Отчет "Состав подсистем и задач ИС"	25	Руководитель IT-службы	4	28
					Программист	24	

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
1.2.2	Определение организационно-логических связей задач действующей информационной системы	1.2.2.1	Диаграмма "ОПС задач действующей ИС"	10	Руководитель IT-службы Программист	2 8	10
1.2.3	Идентификация структуры БД действующей информационной системы	1.2.3.1	Отчет "Структура БД действующей ИС"	6	Руководитель IT-службы Программист	1 8	9
1.2.4	Построение WFD-диаграммы реализации задач действующей информационной системы	1.2.4.1	Диаграмма "WFD-диаграмма реализации задач действующей ИС"	6	Руководитель IT-службы Программист Директор по экономике Экономист	1 6 1 3	11
1.2.5	Формирование сводного отчета об эффективности действующей ИС	1.2.5.1	Сводный отчет «Эффективность действующей ИС»	10	Руководитель IT-службы Программист Директор по экономике Экономист	1 6 1 5	13
	Итого				Руководитель IT-службы Программист Директор по экономике Экономист	20 97 11 76	204
2.	Реинжиниринг бизнес-процесса «Сбыт»						
2.1	Корректировка бизнес-процесса «Сбыт»						

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
2.1.1	Формирование общей концепции реинжиниринга бизнес-процесса «Сбыт»	2.1.1.1	Концепция реинжиниринга бизнес-процесса «Сбыт»	14	Руководитель IT-службы	2	25
					Программист	12	
					Директор по экономике	3	
					Экономист	8	
2.1.2	Построение скорректированной IDEFO-диаграммы бизнес-процесса «Сбыт» по принципу "to be"	2.1.2.1	Диаграмма "IDEFO-диаграмма бизнес-процесса «Сбыт» по принципу "to be"	6	Руководитель IT-службы	2	8
					Программист	3	
					Директор по экономике	1	
					Экономист	2	
2.1.3	Построение скорректированной IDEF3-диаграммы бизнес-процесса «Сбыт» по принципу "to be"	2.1.3.1	Диаграмма "IDEF3-диаграмма бизнес-процесса «Сбыт» по принципу "to be"	6	Руководитель IT-службы	1	8
					Программист	4	
					Директор по экономике	1	
					Экономист	2	
2.1.4	Построение скорректированного сетевого графа бизнес-процесса «Сбыт» и диаграммы Ганта	2.1.4.1	График "Сетевой граф бизнес-процесса «Сбыт» по принципу "to be"	5	Руководитель IT-службы	1	15
					Программист	4	

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
		2.1.4.2	Диаграмма "Диаграмма Ганта бизнес-процесса «Сбыт» по принципу "to be"		Директор по экономике	8	
					Экономист	2	
2.1.5	Построение скорректированной WFD-диаграммы бизнес-задач по принципу "to be"	2.1.5.1	Диаграмма "WFD-диаграмма бизнес-задач по принципу "to be"	5	Руководитель IT-службы	1	
					Программист	4	8
					Директор по экономике	2	
					Экономист	1	
2.1.6	Формирование проекта реинжиниринга бизнес-процесса «Сбыт»	2.1.6.1	Проект «Реинжиниринг бизнес-процесса «Сбыт»	9	Руководитель IT-службы	1	
					Программист	6	13
					Директор по экономике	1	
					Экономист	5	
2.2	Проектирование программного обеспечения задач с учетом реинжиниринга бизнес-процесса «Сбыт»						
2.2.1	Корректировка состава подсистем и задач информационной системы	2.2.1.1	Отчет "Скорректированный состав подсистем и задач"	5	Руководитель IT-службы	1	4
					Программист	3	
2.2.2	Корректировка ОИС задач информационной системы	2.2.2.1	Диаграмма "ОИС задач ИС"	5	Руководитель IT-службы	1	4
					Программист	3	
2.2.3	Корректировка структуры БД информационной системы	2.2.3.1	Отчет "Структура БД ИС"	4	Руководитель IT-службы	1	4
					Программист	3	

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
2.2.4	Корректировка WFD-диаграммы реализации задач информационной системы	2.2.4.1	Диаграмма "WFD-диаграмма реализации задач ИС"	2	Руководитель IT-службы Программист	1 3	4
2.2.5	Разработка и утверждение технического задания на проектирование задач	2.2.5.1	Техническое задание "Разработка ПО задач"	14	Руководитель IT-службы Программист Директор по экономике Экономист	2 16 2 10	30
	Итого				Руководитель IT-службы Программист Директор по экономике Экономист	11 39 18 30	98
3.	Разработка программного обеспечения задач						
3.1	Формирование программного кода задач	3.1.1.1	Реализованное ПО задач	30	Руководитель IT-службы Программист	4 32	36
3.2	Отладка ПО задач	3.2.1.1	Отчет об отладке ПО бизнес-задач	14	Руководитель IT-службы Программист	2 16	22
		3.2.1.2	Скорректированное ПО задач		Директор по экономике Экономист	1 3	
3.3	Разработка сопроводительной документации задач	3.3.1.1	Документ "Сопроводительная документация задач"	14	Руководитель IT-службы Программист Директор по экономике	1 16 1	23

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
3.4	Корректировка регламента реализации задач с учетом ПО	3.4.1.1	Документ "Регламент реализации задач с учетом ПО"	6	Экономист	5	
					Руководитель IT-службы	1	
					Программист	6	13
					Директор по экономике	1	
					Экономист	5	
	Итого				Руководитель IT-службы	8	
					Программист	70	94
					Директор по экономике	3	
					Экономист	13	
4.	Внедрение ПО задач						
4.1	Обучение персонала	4.1.1.1	Отчет "Обучение персонала"		Руководитель IT-службы	1	
				6	Программист	8	
					Руководитель службы "Наименование"	1	18
					Специалист "Наименование"	8	
4.2	Подготовка и ввод исходных данных ПО бизнес-задач	4.2.1.1	Отчет "Исходные данные для ПО БЗ"		Руководитель IT-службы	1	
		4.2.1.2	ПО с исходными данными	20	Программист	12	
					Руководитель службы "Наименование"	1	22
					Специалист "Наименование"	8	
4.3	Опытная эксплуатация	4.3.1.1	Акт "ПО бизнес-задач сланное в опытную эксплуатацию"	4	Руководитель IT-службы	1	162
					Программист	80	

Продолжение таблицы 3.2

1	2	3	4	5	6	7	8
		4.3.1.2	Отчет «Ошибки эксплуатации ПО бизнес-задач»		Руководитель службы "Наименование"	1	
		4.3.1.3	Скорректированное ПО		Специалист "Наименование"		
		4.3.1.4	Скорректированная сопроводительная документация			80	
4.4	Сдача ПО бизнес-процесса «Сбыт» в промышленную эксплуатацию		Акт «ПО бизнес-процесса «Сбыт» промышленную эксплуатацию»	98	Руководитель IT-службы		
					Программист		
					Руководитель службы "Наименование"		
					Специалист "Наименование"		
	Итого				Руководитель IT-службы	3	
					Программист	80	
					Руководитель службы "Наименование"	3	182
					Специалист "Наименование"	96	

Далее проанализируем продолжительность этапов разработки ПО с помощью диаграммы Ганта:

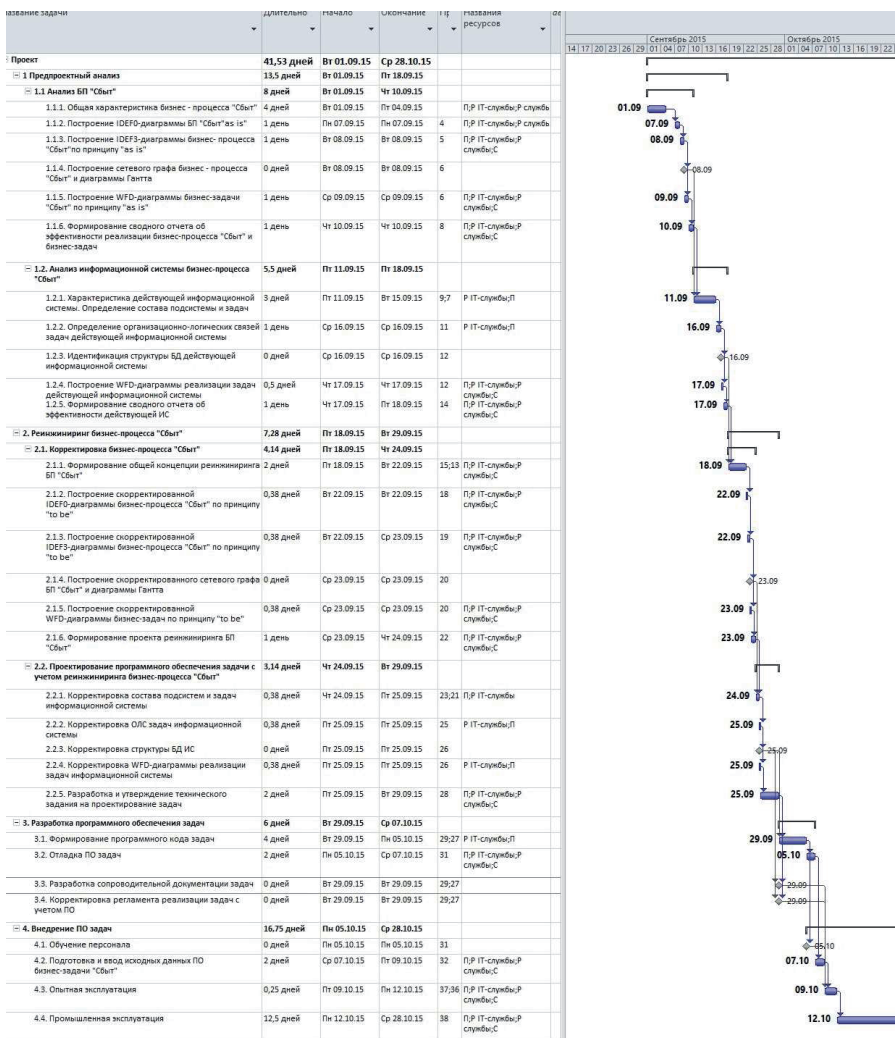


Рисунок 3.15 - Диаграмма Ганта этапов работ разработки программного обеспечения бизнес-процесса «Сбыт»

На диаграмме видно, что продолжительность этапов работ составляет в среднем 2 месяца. Бизнес-процесс охватывает оба месяца, следует для расчета бюджета затрат нужно все трудозатраты и другое разделить соразмерно продолжительности работы.

Для расчета критического пути решения задач нужно построить сетевой граф. Он представлен на следующих рисунках:

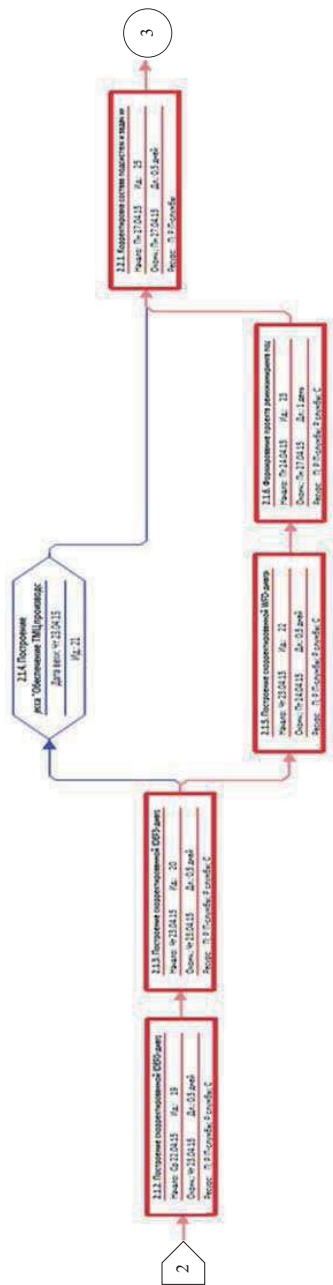


Рисунок 3.18 - Сетевой граф этапов работ разработки ПО (фрагмент 3)

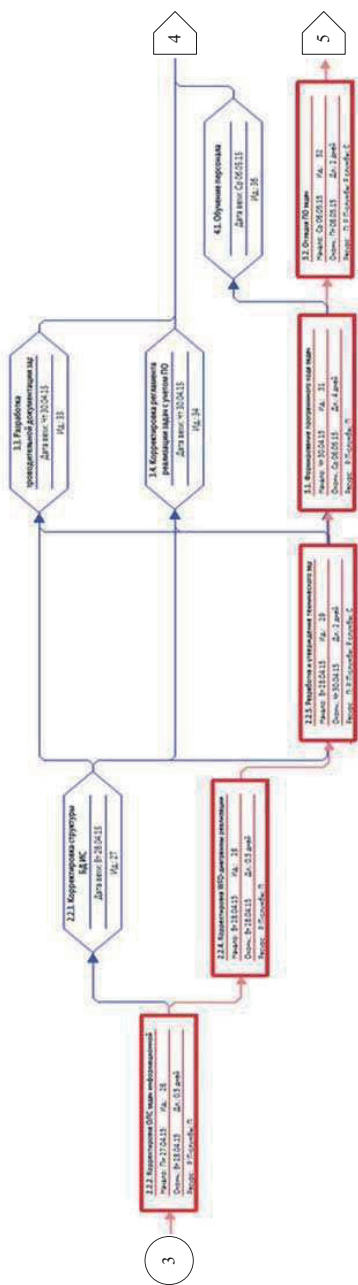


Рисунок 3.19 - Сетевой граф этапов работ разработки ПО (фрагмент 4)

Таким образом длительность всего проекта составляет ориентировочно 2 месяца. На основе этих данных можно рассчитать затраты на разработку ПО.

Расчет затрат на разработку программного обеспечения бизнес-процесса «Сбыт»

Для составления сметы затрат выделим основные статьи:

1. Материальные;
2. Трудовые;
3. Отчисления в социальные фонды;
4. Амортизация основных фондов;
5. Прочие.

В данном случае материальные затраты не учитываются, так как они составляют незначительную часть расходов для организации.

Трудовые затраты включают в оплату труда руководителя ИТ – службы, программиста, руководителя экономической службы и экономиста.

Таблица 3.3 - Расчет тарифных ставок

РСиС	Должностной оклад, руб.	Ставка руб./час
Руководитель ИТ-службы	35200,00	220,00
Программист	32000,00	200,00
Директор по экономике	33600,00	210,00
Экономист	24000,00	150,00

Составим расчет трудозатрат на разработку ПО решения бизнес – задач в разрезе проектных процессов и календарного плана.

Таблица 3.4 - Расчет трудозатрат в разрезе проектных процессов (руб.)

РСиС	1. Предпроектный анализ		2. Реинжиниринг ПП «Планирование закупок		3. Разработка ПО задач		4. Внедрение ПО задач "	
	чел-час	руб.	чел-час	руб.	чел-час	руб.	чел-час	руб.
Руководитель IT-службы	20	4 400	11	2 420	8	1 760	3	660
Программист	97	19 400	39	7 800	70	14 000	80	16 000
Директор по экономике	11	2310	18	3 780	3	630	3	630
Экономист	76	11 400	30	4 500	13	1 950	96	14 400
Итого	204	37 510	98	18 500	94	18 340	182	31 690

Таблица 3.5 - Расчет трудозатрат во временном разрезе (руб.)

РСиС	Периоды					
	0		1		2	
	чел-час	руб.	чел-час	руб.	чел-час	руб.
Руководитель IT-службы			31	6 820	11	2 420
Программист			136	27 200	150	30 000
Директор по экономике			29	6 090	6	1 260
Экономист			106	15 900	109	16 350
Итого			302	56 010	276	50 030

Отчисления в социальные фонды составляют 30% от фонда оплаты труда.

Расчет амортизации основных фондов приведен в таблице 3.12.

Средняя первоначальная стоимость комплекта:

1. Компьютерная техника: 60 000 рублей.

2. Орг. техника: 45 000 рублей.

Способ начисления амортизации: линейный. Срок службы: 5 лет. Норма амортизации: 20%.

Амортизационные начисления:

- в месяц

Компьютерная техника: $60\,000 * 0,2 = 12\,000$ рублей/месяц.

Орг. техника: $45\,000 * 0,2 = 9\,000$ рублей/месяц.

- в час

Компьютерная техника: $12\,000 / (20 * 8) = 75$ рублей/час.

Орг. техника: $9\,000 / (20 * 8) = 56,25$ рублей/час.

Таблица 3.6 - Расчет амортизации ОФ в разрезе проектных процессов

Наименование ОС	1. Предпроектный анализ		2. Реинжиниринг бизнес-процесса «Сбыт»		3. Разработка ПО задач		4. Внедрение ПО задач		Всего	
	час	руб.	час	руб.	час	руб.	час	руб.	час	руб.
Компьютерная техника	204	15 300	98	7 350	94	7 050	182	13 650	578	31 060
Орг. техника	204	11 475	98	5 512,5	94	5 287,5	182	10 237,5	578	32 512,5

Таблица 3.7 - Расчет амортизации ОФ во временном разрезе

Наименование ОС	Периоды						Всего
	0		1		2		
	час	руб.	час	руб.	час	руб.	
Компьютерная техника			298	22 350	276	20 700	руб. 43 050
Орг. Техника			298	16 762,5	276	15 525	32 287,5

На основе всех этих данных составим бюджет инвестиций на разработку ПО решения бизнес – задачи в разрезе этапов работ и календарного плана.

Таблица 3.8 - Бюджет инвестиций на разработку ПО решения бизнес - задач

Элементы затрат	1. Предпроектный анализ	2. Реинжиниринг бизнес-процесса «Сбыт»	3. Разработка ПО задач	4. Внедрение ПО задач	Всего
1	2	3	4	5	6
1. Материальные затраты	-	-	-	-	-
2. Заработная плата	37 510	18 500	18 340	31 690	106 040
3. Отчисления в соц. фонд	11 253	5 550	5 502	9 507	31 812
4. Амортизация:	26 775	12 862,5	12 337,5	23 887,5	75 862,5
здания и сооружения	-	-	-	-	-
компьютерная техника	15 300	7 350	7 050	13 650	43 350
орг. техника	11 475	5 512,5	5 287,5	10 237,5	32 512,5
прочее	-	-	-	-	-
5. Прочее:	2 980	1 923	1 450	2 690	9 043
материальные затраты	1 470	978	700	1 500	4 648
(в т.ч. бумага,	310	200	200	400	1 110
канцтовары,	400	245	200	400	1 245
расходные материалы и т.д.)	350	200	150	350	1 050
услуги связи (телефон, интернет)	450	300	200	450	1 400
Итого стоимость ПО	78 518	38 835,5	37 629,5	67 774,5	308 073

Таблица 3.9 - Размер инвестиций на разработку задач

Элементы затрат	Периоды			Всего
	0	1	2	
1. Материальные затраты				
2. Заработная плата		56 010	50 030	106 040
3. Отчисления в соц. фонд		16 803	15 009	31 812
4. Амортизация:		39 637,5	36 225	75 862,5
здания и сооружения		-	-	-
компьютерная техника		22 650	20 700	43 350
орг. техника		16 987,5	15 525	32 512,5
прочее		-	-	-
5. Прочее:		4 903	4 140	9 043
материальные затраты		2 448	2 200	4 648
(в т.ч. бумага,		510	600	1 110
канцтовары,		645	600	1 245
расходные материалы и т.д.)		550	500	1 050
услуги связи (телефон, интернет)		750	650	1 400
Итого стоимость ПО	0,00	161 894	146 179	308 073

Таким образом, размер инвестиций на разработку задач равен 308 073 руб.

3.4.2. Бюджет решения задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки»

Бюджет решения задачи без использования программного обеспечения

Процессно – продуктовый бюджет решения задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки» без использования ПО представлен в расчётных таблицах 3.10-3.11.

Таблица 3.10 - Себестоимость на одну единицу бизнес- продукта без использования программного обеспечения

Элементы затрат	Себестоимость бизнес - продукта без использования программного обеспечения, руб.
Бизнес - продукт «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки»	290,00
2. Заработная плата	60,94
3. Отчисления в соц.фонд	22,98
4. Амортизация:	26,11
здания и сооружения	-
компьютерная техника	4,11
орг. техника	9,00
прочее	-
5. Прочее:	67,87
материальные затраты	5,41
(в т.ч. бумага,	5,00
канцтовары,	1,18
расходные материалы и т.д.)	3,82
услуги связи (телефон, интернет)	5,76
Итого	494,82

Таким образом, стоимость решения бизнес – задачи на одну единицу бизнес – продукта без использования ПО равна 494,82 рубля.

Таблица 3.11 - Себестоимость бизнес – продукта без использования программного обеспечения

Элементы затрат	Себестоимость бизнес - продукта без использования программного обеспечения, руб.
Бизнес - продукт «Формирование рейтинга поставщиков»	290,00

Продолжение таблица 3.11

2. Заработная плата	1218,78
3. Отчисления в соц.фонд	459,59
4. Амортизация:	522,20
здания и сооружения	-
компьютерная техника	82,20
орг. техника	180,00
прочее	-
5. Прочее:	1357,4
материальные затраты	108,20
(в т.ч. бумага,	9,40
канцтовары,	23,60
расходные материалы и т.д.)	19,80
услуги связи (телефон, интернет)	115,2
Итого	4109,77

Таким образом, себестоимость бизнес – продукта задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки» составила 4109,77 рубля.

Бюджет решения задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки» с применением программного обеспечения

Процессно – продуктовый бюджет решения задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки» с использования ПО представлен в расчётной таблице 3.12-3.13.

Таблица 3.12 - Себестоимость на одну единицу бизнес – продукта с использованием программного обеспечения

Элементы затрат	Себестоимость бизнес - продукта с использованием программного обеспечения, руб.
Бизнес - продукт задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки»	290,00
2. Заработная плата	32,50
3. Отчисления в соц.фонд	9,75
4. Амортизация:	12,11
здания и сооружения	-
компьютерная техника	4,11
орг. техника	8,00

Продолжение таблицы 3.12

Прочее	-
5. Прочее:	4,00
материальные затраты	2,00
(в т.ч. бумага,	1,00
канцтовары,	1,00
расходные материалы и т.д.)	0,00
услуги связи (телефон, интернет)	0,00
Итого	379,39

Таким образом, стоимость решения бизнес – задачи на одну единицу бизнес – продукта с использованием ПО равна 379,39 рубля.

Таблица 3.13 - Себестоимость бизнес – продукта с использованием программного обеспечения

Элементы затрат	Себестоимость бизнес - продукта без использования программного обеспечения, руб.
1	2
Бизнес - продукт задач «Оформление и бронирование путевки » и «Онлайн-предоплата санаторной путевки»	290,00
2. Заработная плата	650,00
3. Отчисления в соц.фонд	195,00
4. Амортизация:	242,20
здания и сооружения	-
компьютерная техника	82,20
орг. техника	160,00
прочее	-
5. Прочее:	80,00
материальные затраты	40,00
(в т.ч. бумага,	20,00
канцтовары,	20,00
расходные материалы и т.д.)	0,00
услуги связи (телефон, интернет)	0,00
Итого	1699,20

Таким образом, себестоимость бизнес – продукта составила 1699,20 рублей.

Сравнительный анализ бюджетов решения бизнес – задачи и себестоимости бизнес – продуктов

Структуру изменения затрат с использованием ПО отражает таблица 3.14. и рисунок 3.21.

Таблица 3.14 - Структура изменения с использованием ПО

	Бюджет затрат без использования ПО	Бюджет затрат с использованием ПО	Структура изменения затрат с использованием ПО, руб.	Структура изменения затрат с использованием ПО, %
Трудовые затраты	1678,37	845,00	833,37	49,65
Амортизация ОФ	522,20	242,20	280,00	53,62
Прочие затраты	1357,40	80,00	1277,40	94,10
Входные бизнес-продукты	290,00	290,00	0	0,00
Итого	3847,97	1457,20	2390,77	62,13

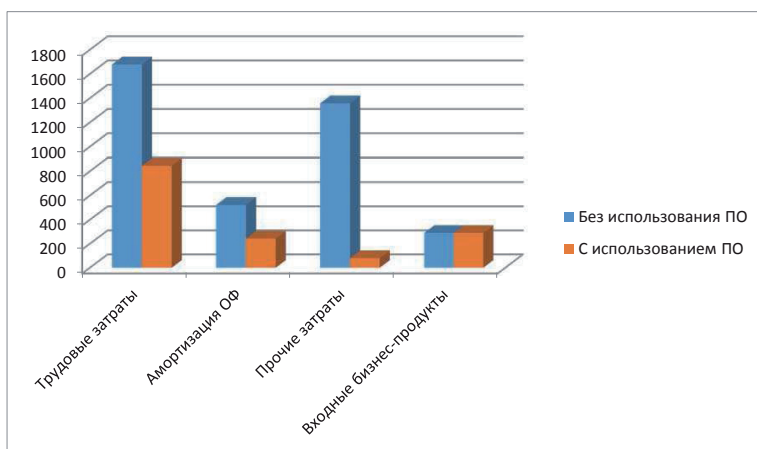


Рисунок 3.21 - Структура изменения затрат с использованием ПО

Таким образом, ежемесячная экономия от использования ПО составила 2475,03 рублей.

3.4.3. Расчет показателей экономической эффективности

Расчет экономической эффективности разработки ПО решения бизнес-задач произведем на основе метода «потоков платежей» («Cash Flow») с использованием разностного подхода.

Для оценки экономической эффективности определим чистый денежный поток за 1 год за вычетом стоимости единовременных вложений и с учетом

альтернативных упущенных возможностей. Для оценки упущенных возможностей примем в качестве ставки сравнения (дисконтирования) значение ставки рефинансирования ЦБ РФ.

Рассчитаем основные показатели, характеризующие экономическую эффективность и эффект данного проекта:

- 1) чистую текущую стоимость проекта;
- 2) внутреннюю норму доходности проекта;
- 3) индекс рентабельности инвестиций;
- 4) срок окупаемости инвестиций (простой и дисконтированный).

1. Чистая текущая стоимость проекта (ЧТС) рассчитывается по следующей формуле:

$$ЧТС = \sum_{t=0}^T \frac{ЧДП_t}{(1 + R/100)^t} - \sum_{t=0}^T \frac{KV_t}{(1 + R/100)^t}, \quad [14]$$

где ЧТС – чистая текущая стоимость проекта, руб.;

KV_t – проектные капиталовложения периода t , руб.;

t – номер планового периода (в качестве планового периода может быть принят месяц, квартал, полугодие, год);

T – горизонт планирования, лет;

R – ставка сравнения (дисконтирования), соответствующая плановому периоду, %;

$ЧДП_t$ – чистый денежный поток периода t , очищенный от капиталовложений, руб.

В случае если капиталовложения осуществляются одновременно в начале первого периода, то формула для расчета данного показателя примет следующий вид:

$$ЧТС = \sum_{t=0}^T \frac{ЧДП_t}{(1 + R/100)^t} - KV \quad [14]$$

2. Внутренний коэффициент окупаемости (внутренняя норма доходности) это такая ставка сравнения (дисконтирования), при которой чистая текущая стоимость проекта равна нулю, т.е.:

$$ВНД = R^* \left| \begin{array}{l} ЧТС = \sum_{t=0}^T \frac{ЧДП_t}{(1+R^*/100)^t} - \sum_{t=0}^T \frac{КV_t}{(1+R^*/100)^t} = 0 \end{array} \right| \quad [14]$$

Вышеприведенный способ является трудоемким, так как предполагает итерационный характер расчета данного показателя. В связи с этим можно воспользоваться альтернативной формулой:

$$ВНД = R^+ + \frac{ЧТС^+}{ЧТС^+ + |ЧТС^-|} \cdot (R^- - R^+), \quad [14]$$

где ВНД – внутренняя норма доходности (внутренний коэффициент окупаемости); ЧТС⁺, ЧТС⁻ – положительные и отрицательные значения показателя чистой текущей стоимости проекта, вычисленные соответственно при ставках дисконтирования R⁺, R⁻.

При расчете показателя ВНД по второй формуле с целью уменьшения погрешности необходимо, чтобы значения R⁺ и R⁻ были как можно ближе друг к другу (разница между ними не должна превышать 10-15%).

3. Индекс рентабельности

$$PI = \sum_{t=0}^T \frac{ЧДП_t}{(1+R/100)^t} / \sum_{t=0}^T \frac{КV_t}{(1+R/100)^t} \quad [14]$$

4. Дисконтированный срок окупаемости:

$$T_{ок.диск.} = m + \frac{\sum_{t=0}^T \frac{КV_t}{(1+R/100)^t} - \sum_{t=0}^m \frac{ЧДП_t}{(1+R/100)^t}}{\frac{ЧДП_{m+1}}{(1+R/100)^{m+1}}}, \quad [14]$$

где m – текущий номер периода, при котором должно выполняться условие

$$\sum_{t=0}^m \frac{ЧДП_t}{(1+R/100)^t} < \sum_{t=0}^T \frac{КV_t}{(1+R/100)^t} < \sum_{t=0}^{m+1} \frac{ЧДП_t}{(1+R/100)^t}; \quad [14]$$

5. Простой срок окупаемости.

$$T_{ок.пр.} = m + \frac{\sum_{t=0}^T КV_t - \sum_{t=0}^m ЧДП_t}{ЧДП_{m+1}}, \quad [14]$$

где m – текущий номер периода, при котором должно выполняться условие

$$\sum_{t=0}^m \times \ddot{A}I_t < \sum_{t=0}^T KV_t < \sum_{t=0}^{m+1} \times \ddot{A}I_t ;$$

Если величина прибыли (экономии) в каждом периоде планирования одинаковая, то простой срок окупаемости можно также вычислить по следующей формуле:

$$T_{ок.пр.} = \frac{\sum_{t=0}^T KV_t}{\Pi} \quad [14]$$

где Π – величина прибыли (экономии себестоимости продукции/услуг/работ) от использования информационной системы, руб.

Результаты расчетов показателей по вышеприведенным формулам приведены в таблице.



Рисунок 3.22 - Финансовый профиль

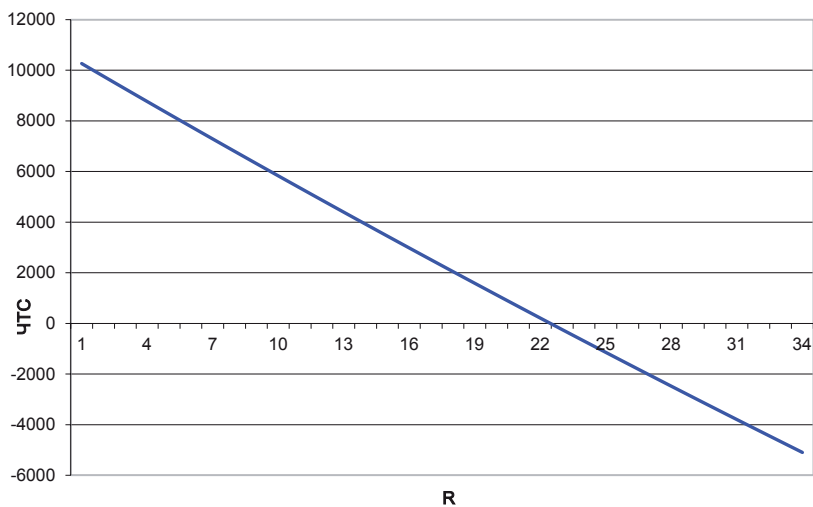


Рисунок 3.23 - График зависимости чистой текущей стоимости проекта от ставки дисконтирования

Экономическая эффективность капиталовложений в проект обеспечивается при выполнении следующей системы условий:

$$\begin{cases} ЧТС \geq 0; \\ ВНД \geq R; \\ T_{ок. диск} \leq T_{ок. норм.} \end{cases} \quad [15]$$

Норматив срока окупаемости устанавливается исходя из требований инвесторов в зависимости от объема инвестиций, амортизационного срока и т.д.

Для рассматриваемого проекта данная система условий удовлетворяется:

$$\begin{cases} (ЧТС = 6139,8) \geq 0; \\ (ВНД = 21,3\%) \geq (R = 8,25\%); \\ (T_{ок. диск} = 17,8) \leq (T_{ок. норм.} = 18). \end{cases} \quad [15]$$

Следовательно, рассмотренный проект с экономической точки зрения является эффективным и целесообразным для реализации.

Список использованных источников

1. Бочкарева И.И., Левина Г.Г. Бухгалтерский финансовый учет: учебник /И.И. Бочкарева, Г.Г. Левина; под ред. проф. Я.В. Соколова. – М.: Магистр, 2008. – С. 50-53.
2. Бизнес-реинжиниринг: Учебное пособие / Под ред. Н.В. Васильева - М.: Экмос, 2005. – С. 25-27.
3. Голощапов А.Л., Google Android. Создание приложений для смартфонов и планшетных ПК. – М.: БХВ-Петербург, 2003. – С. 30-35.
4. Залманова М.Е. Экономика материально-технического обеспечения.: учебник – Саратов, СГТУ, 2009. – С. 50-58.
5. Замков О.О., А.В. Толстопятенко, Ю.Н. Черемных. Математические методы в экономике: учебник/; под ред. А.В. Сидоровича. – 3-е изд. перераб. – М.: Дело и Сервис, 2001. – 112 с.
6. Леонов В.А. Планшеты и смартфоны на Android - М.: Сфера, 2014. – С. 110-111.
7. Майер Рето „ Программирование приложений для планшетных компьютеров и смартфонов, Эксмо, 2012. – С. 20-23.
8. Орлов А.И. Организационно-экономическое моделирование: теория принятия решений: учебник /М.:КНОРУС, 2011. – С. 90-92.
9. Проектирование экономических информационных систем: Учебник/ Г.Н. Смирнова, А.А. Сорокин, Ю. Ф.Тельнов; под ред. Ю.Ф. Тельнова. – М.: Финансы и статистика, 2010. – С. 50-51.
10. Пол Дейтел., Android для программистов: создаем приложения.- М. : Изд-во РДЛ, 2004. – С. 70-77.
11. Райзберг Б.А., Лобко А.Г. Программно-целевое планирование и управление. – М.: ИНФРА-М, 2006. – С. 420-431.
12. Рубцов С.В. Уточнение понятия «Бизнес-процесс» // Менеджмент в России и за рубежом. - 2001. - №6. – 110 с.

13. Ручкин К.А., Ручкина В.Н. Моделирование бизнес-процессов с помощью современных информационных технологий // Менеджмент в России и за рубежом. -2007.- №1. – С. 224-225.
14. Смирнов Ю.Н. О показателях эффективности решения бизнес-задач и их агрегирование для бизнес-процессов: Научно-практический журнал "Интеграл" - 2008. - №5. – С. 490-492.
15. Смирнов Ю.Н. Процессно-задачный инжиниринг бизнес-процессов и стандарт управления предприятием: Научно-практический журнал "Интеграл" - 2007. - № 5. – С. 110-119.
16. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем: Учебник/ Ю.Ф. Тельнова. – М.; Финансы и статистика, 2010. – 53 с.
17. Уткин Э.А. Бизнес-реинжиниринг. Обновление бизнеса. – М.: ЭКМОС, 2005. – С. 69-71.
18. Федюкин В.А. Управление процессами // Современный менеджмент. - 2006. - №5. – С. 78-91.
19. Хостинская Г.И. Реинжиниринг на предприятиях сферы услуг // Менеджмент в России и за рубежом. - 2001.- №6. – С. 461-466.
20. Черемных О.В. Реинжиниринг: в чем его польза? // Управление компаний. – 2005. - №6. – 120 с.
21. Шумилов, Ю.П. Менеджмент информационных ресурсов. - М.; ИРР. - 2005. – 355 с.
22. Экономика предприятия: учебник/; под ред. О.И.Волкова, - М.: Инфра-М, 2009. – С. 50-57.
23. Юдин Д.Б., Горяшко А.П. Математические методы оптимизации устройств и алгоритмов АСУ: учебник/;под ред. Ю.В.Астафьева, В.А. Шабалина. – М.:Радио и связь, 1982. – С. 40-41.
24. Экономика предприятия: учебник/; под ред. О.И.Волкова, - М.: Инфра-М, 2009. – С. 99-101.

25. Архитектура Android-приложений: Сайт [Электронный ресурс].- Режим доступа: <http://habrahabr.ru/post/140459/>. Дата обращения: 12.03.2017.
26. «ЛПЧУП Санаторий «Ижминводы»»: Официальный сайт [Электронный ресурс].- Режим доступа: <http://ижмин.пф>. Дата обращения: 21.09.2017.
27. Обмен данными с интернет-магазинами: Сайт [Электронный ресурс].- Режим доступа: <http://v8.1c.ru/small.biz/ishop/>. Дата обращения: 22.02.2017.
28. Android studio: Сайт [Электронный ресурс].- Режим доступа: <http://developer.android.com/sdk/index.html>. Дата обращения: 21.02.2017.
29. Developer Workflow: Сайт [Электронный ресурс].- Режим доступа: <http://developer.android.com/tools/workflow/index.html>. Дата обращения: 23.03.2017.
30. IOS: Официальный сайт [Электронный ресурс].- Режим доступа: <https://www.apple.com/ru/ios/>. Дата обращения: 21.02.2017.

