**Федеральное государственное бюджетное образовательное учреждение высшего образования «РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет: Управление

Кафедра: Информатики и информационных технологий

Направление подготовки: 09.03.03 Прикладная информатика

Направленность: Корпоративные информационные системы управления

**КОНТРОЛЬНАЯ РАБОТА**

По дисциплине: Программная инженерия

На тему: «Разработка информационной системы Гостиницы»

**АВТОР**

Студенты 2 курса группы ИБ-321

Очной формы обучения

Латышев Д.А, Павлов Н.А, Наумова А.П, Ластина А.И

**НАУЧНЫЙ РУКОВОДИТЕЛЬ**

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

Окулич Виктор Иванович

Нижний Новгород, 2022г.

**Оглавление**

[**Введение**](#_silxxyt616j8) **3**

[**1. Теоретическая часть**](#_30j0zll) **4**

[1.1. Описание этапов разработки](#_1fob9te) 4

[1.2. Роли исполнителей проекта и их назначение](#_3znysh7) 5

[1.3. Предметная область](#_tyjcwt) 8

[1.4 Описание процесса работы команды](#_1qkz4soitmts) 9

[1.4.1 Используемое программное обеспечение](#_3dy6vkm) 10

[1.4.1. Мастер ТЗ](#_1t3h5sf) 10

[1.4.2. ELMA365](#_4d34og8) 10

[1.4.3. BPWin](#_17dp8vu) 11

[1.4.4. Microsoft Visual Studio.](#_26in1rg) 11

[**2. Практическая часть**](#_vo1qvgnwl2t0) **13**

[2.1 Разработка схем в BPWin](#_8l61t14mmzgr) 13

[2.2 Разработка бизнес-процессов в ELMA 365](#_y18yxarxriwl) 14

[2.3 Разработка интерфейса](#_73kdtn76mgbl) 15

[2.4 Методика оценки стоимости COCOMO II](#_bzxdnb9ebrzm) 17

[**2.5 Оценка программного продукта методом функциональных точек**](#_tqjxh857c39y) **20**

[**Заключение**](#_kc0lleh6xwh4) **22**

[**Приложения**](#_e5qqlu8ys06w) **23**

# 

# Введение

В наше время информационные технологии играют важнейшую роль в жизни общества. Они помогают нам обрабатывать поток информации, который с каждым днём становится всё больше, упрощая нашу повседневную жизнь и ведение бизнеса.

Цифровизация и компьютеризация положительно сказывается на работе любых предприятий.

**В нашем проекте** мы рассмотрим проект по построению блока информационной системы Гостиницы, позволяющей автоматизировать основной блок задач данной организации с использованием различных CASE-средств: ELMA365, BPWin, Мастер ТЗ.

Контрольная работа направлена на закрепление знаний, полученный в рамках учебной дисциплины. В ходе выполнения контрольной работы необходимо решить следующие **задачи**:

* Изучение теоретических основ продуктов, которые будут использованы при выполнении контрольной работы;
* Рассмотрение бизнес-процессов, протекающих в рассматриваемой предметной области;
* Создание организационной структуры проектной организации;
* Изучение спецификации каждой из ролей, связанной с конкретным бизнес-процессом;
* Распределение состава команды по ролям, связанным с отдельными процессами;
* Выполнение участниками команды работ, соответствующих выделенной роли;
* Проведение оценки работы участников в ходе проекта.

# 1. Теоретическая часть

## 1.1. Описание этапов разработки

Жизненный цикл разработки ИТ-продукта делится на пять этапов:

1. **Планирование**

Грамотное планирование функциональности будущего продукта и анализ требований играют ключевую роль для всего проекта. За этот этап несет ответственность менеджер проекта, так как именно он отвечает за успех всего процесса разработки.  
До проектирования продукта планирование носит “грубый” характер, так как точный ход разработки на этом этапе узнать невозможно.  
После того, как UX/UI проектирование выполнено, можно составить точный план, как будет идти разработка, и какая функциональность будет в продукте.

1. **Разработка**

Разработчики следуют одной из методологий — для компании это в основном Agile. Эта методология предполагает гибкий итеративный подход — то есть разработчики действуют последовательно, разделяя проект на более мелкие задачи.  
Итерации в Agile называются спринтами, и в один спринт входят работы по всем направлениям: планирование, дизайн, разработка, тестирование.

1. **Дизайн**

После планирования наступает черед UX/UI дизайнеров — специалистов, которые проектируют пользовательские интерфейсы. Дизайнеры занимаются изучением поведения пользователей и выстраиванием понятного человеку интерфейса. Визуальный вид продукта — также результат работы дизайнеров.  
Вместе с ними работают системные архитекторы, которые решают, какую структуру будет иметь готовый продукт, и как он должен себя вести.

1. **Тестирование**

Специалисты по тестированию выполняют разные виды тестирования: модульное, интеграционное, тестирование интерфейса и другие виды в зависимости от цели. Эта категория специалистов должна прийти к конечному выводу, что в продукте нет ошибок и он готов к релизу.  
После этого продукт можно внедрять и интегрировать со сторонним программным обеспечением. Процесс разработки на этом не заканчивается — он продолжается, пока не будут внесены доработки.

1. **Поддержка**

Готовый продукт может нуждаться в дополнительной поддержке, будь то дополнительные вопросы по поводу работы продукта от клиентов, или необходимость внести изменения в уже заложенные функции, — специалисты службы поддержки всегда готовы прийти на помощь.  
Большинство проектов по разработке проходят все этапы вышеописанного жизненного цикла.  
Для примера мы подробнее рассмотрим процесс разработки продукта на текущем проекте компании.

## 1.2. Роли исполнителей проекта и их назначение

**Менеджер проекта (Павлов Никита)**

Когда требования клиента определены и правильно интерпретированы, в процесс разработки подключается менеджер проекта.  
Его основная задача заключается в управлении проектом, как следует из названия профессии.

Менеджер проекта выполняет огромное количество работ, начиная от разработки плана проекта, оценки рисков,  
контроля функциональных и стоимостных рамок и заканчивая ежедневной работой с командой на проекте.

Менеджер проекта распределяет задачи между ними, планирует ход работы, мотивирует команду, контролирует процесс и координирует общие действия. Также он несет ответственность за тайм-менеджмент, управление рисками и действия в случае непредвиденных ситуаций.

Менеджер проекта следит за тем, чтобы процесс разработки шел по плану.  
В гибких методологиях это также подразумевает то, что изменения, которые необходимо внести в проект, внедряются корректно, так что клиент может быстро и легко утвердить их, а команда не делает лишней работы.

**Аналитик (Латышев Данила)**

Эти ребята берут на себя весьма существенную часть работы, находясь на стыке бизнеса и технологий.

Роль бизнес-аналитика заключается в планировании оптимального процесса разработки, интерпретации требований заказчика, подготовке документации и осуществления эффективной коммуникации между клиентом и командой.

В типичные задачи бизнес-аналитика входит:

* Работа с представителями бизнеса и глубокое понимание их процессов;
* Самостоятельная идентификация и анализ проблем/узких мест в рабочих процессах;
* Сопоставление найденных проблем с потребностями бизнеса,которые они сами изъявили;
* Проектирование решения, которое удовлетворит все потребности и решит проблемы (зачастую, помимо проектирования IT решения, требуется также и предложение по реструктуризации деятельности);
* После того, как решение спроектировано (построены модели данных, описаны use-case, написаны спецификации на разработку), документация переходит к разработчику, а аналитик продолжает постоянную коммуникацию с разработчиком для более эффективной работы

**Дизайнер (Ластина Алина)**

Это тот человек, от которого идет большая часть креативности в проекте.  
Главная ответственность UI/UX дизайнера заключается в создании приятного интерфейса и отличного пользовательского опыта.

Дизайнер использует варфреймы, созданные клиентом или бизнес-аналитиком, чтобы создать дизайн интерфейса (UI) согласно действующим гайдлайнам и трендам.  
Он также планирует пользовательский опыт, который сделает продукт удобным для использования.

Клиенту удобно видеть модель приложения, а программистам прототип просто необходим, чтобы написать код.  
Это как дизайн-проект комнаты для профессионалов, которые будут её декорировать — необходимо видеть, что должно получиться в результате работы.

**Архитектор (Павлов Никита)**

Архитектор — придает законченную форму действиям команды, направляет внимание и пытается придать определенные рамки групповым обсуждениям и результатам совместной деятельности. В безнадежном проекте особенно важно иметь единое и четкое представление о проблеме и ее возможном решении.

**Программист (Латышев Данила)**

На основе анализа математических моделей и алгоритмов решения экономических и других задач разрабатывает программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку. Разрабатывает технологию решения задачи по всем этапам обработки информации. Осуществляет выбор языка программирования для описания алгоритмов и структур данных. Определяет информацию, подлежащую обработке средствами вычислительной техники, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и вывода, методы ее контроля.

**Тестировщик (Наумова Анастасия)**

Тестировщик ПО выполняет следующие должностные обязанности:

Разрабатывает планы, графики, методики и описания тестирования.  
Моделирует ситуации, которые могут возникнуть в условиях эксплуатации программного обеспечения.  
Выполняет тестирование программных продуктов.  
Выполняет нагрузочные тестирования.  
Составляет документацию для проведения функционального тестирования.

## 1.3. Предметная область

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста. Под контекстом здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Прежде чем приступать к работе, необходимо внимательно изучить выбранную предметную область, а в дальнейшем перейти к реализации базы данных по ней.

Гостиница дает возможность покупателям забронировать номера, ведет бюллетень по покупателям: туристам, командировочным. Для того, чтобы продать номер клиенту, необходимы следующие данные: фамилия, имя, отчество, дата рождения, паспортные данные, гражданство, адрес места жительства, время заселения и время отъезда, сумма к оплате за предоставленные услуги. При регистрации клиента, как покупателя, клиент должен предъявить паспортные данные или документы удостоверяющие личность, и оплатить стоимость номера. Гостиница хранит информацию о номерах в гостинице, которые заняты клиентами. В случае, если бронирование номеров гостиницы переносится на более позднее время или дату, изменения должны быть внесены в бюллетень покупателей, зарегистрированных в этот номер гостиницы, сообщены покупателю и должна быть предоплата за бронирование номера.

Базой данных может пользоваться администратор и покупатель номера, только у каждого разные права доступа к ней: администратор просматривает базу, вносит изменения, а покупатель может только просматривать. Следовательно, база данных должна быть защищена от несанкционированных действий злоумышленников паролем на вход.

Основной вид деятельности Гостиницы – предоставление мест для краткосрочного проживания, а также предоставление полного ассортимента продуктов питания и напитков, пригодных для непосредственного потребления. Вид экономической деятельности – предоставление услуг гостиницами с различными типами предприятий общественного питания (ресторанами, кафе, закусочными, барами и т.п.) (ОКВЭД код 55.11).

***Код ОКВЭД 55.11*** – предоставление услуг гостиницами с различными типами предприятий общественного питания (ресторанами, кафе, закусочными, барами и т.п.).

## 1.4 Описание процесса работы команды

Процесс разработки программного продукта начинается с поступления заказа нашей команде. В данном случае, это гостиница, которая хочет внедрить IT-технологии для автоматизации своей работы. Заказчик поставил нам задачу разработать интерфейс к уже имеющейся базе данных его предприятия, а также описал основной функционал программы, которая предназначена для рабочего персонала и конечных пользователей.

Далее аналитик нашей команды приступил к подсчету трудоемкости и стоимости создания проекта. После этого заказчик получает примерное представление того, как будет выглядеть законченный проект.

Дизайнер описал своё представление будущего приложения, основываясь на общепринятых практиках, заказчик в свою очередь согласился, и наша команда приступила к работе.

### 1.4.1 Используемое программное обеспечение

#### 1.4.1. Мастер ТЗ

Бесплатная программа «Мастер Технических Заданий» или же Мастер ТЗ создана для упрощения процесса создания технических заданий (ТЗ)

Программа не требует установки и представляет собой удобный пошаговый мастер создания технического задания.

Благодаря тому, что программа содержит готовые шаблоны и структуру, ТЗ можно создавать с минимальной потерей времени.

#### 1.4.2. ELMA365

ELMA365 – это система для автоматизации и роботизации бизнес-процессов и CRM. Благодаря Low-code технологии система ELMA365 позволяет создавать корпоративные приложения быстрее и с меньшим количеством программного кода. Она налаживает коммуникации, предотвращает потерю данных и предоставляет необходимые инструменты контроля выполнения работ. Таким образом, сотрудники могут заниматься более производительным трудом, повышая эффективность компании. Кроме того, система позволяет создавать свои решения для бизнеса, а также скачивать готовые из каталога Store365. ELMA365 не требует установки дополнительного программного обеспечения — вся работа осуществляется через браузер.

Возможности ELMA365:

* Low-code платформа для создания корпоративных приложений без программирования.
* Моделирование бизнес-процессов.
* Исполнение бизнес-процессов.
* Дизайнер для описания оргструктуры и бизнес-процессов.
* Работа с клиентами (CRM).
* Инструменты по управлению документооборотом.
* Каталог готовых решений, которые нетрудно настроить для пользовательского использования.
* Корпоративный мессенджер для совместной работы: лента, каналы и чаты, обмен файлами и постановка задач.
* Управление задачами.
* Хранилище файлов.
* Мобильный интерфейс.
* ChatDesk
* API для использования во внешних программных продуктах.

#### 1.4.3. BPWin

BPwin – это программное обеспечение, которое было выпущено разработчиками из ltd. Logic Works. Основное назначение утилиты – поддержка созданий систем контроля информации. Считается принадлежащим к средствам верхнего уровня. Последние требуются для работы с начальными этапами создания системы обработки информации.

Данное программное обеспечение напрямую связано с анализом и планированием. Главное назначение – расстановка задач, целей и приоритетов компании с визуальным отображением всех данных. Во время работы требуется управление графическим интерфейсом для построения графиков текущей модели. К перечню последних относят потоковые данные, схемы структур и диаграммы возможных решений.

#### 1.4.4. Microsoft Visual Studio.

**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода.

Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

# 2. Практическая часть

Процесс разработки программного продукта начинается с поступления заказа нашей команде. В данном случае, это Гостиница, который хочет внедрить IT-технологии для автоматизации своей работы.

В начале работы над проектом мы составили примерную схему процесса разработки ПО в приложении BPWin в IDEF0.

Далее аналитик нашей команды приступил к подсчету трудоемкости и стоимости создания проекта. После этого заказчик получает примерное представление того, как будет выглядеть законченный проект

В начале работы над проектом мы составили примерную схему процесса разработки ПО в приложении BPWin в IDEF0(Рис.1,Рис.2).

## 2.1 Разработка схем в BPWin

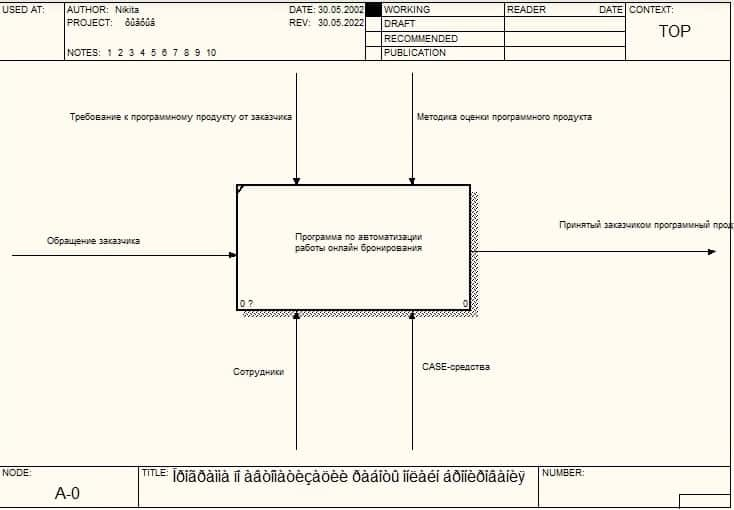


Рисунок 1 — Схема процесса разработки ПО.

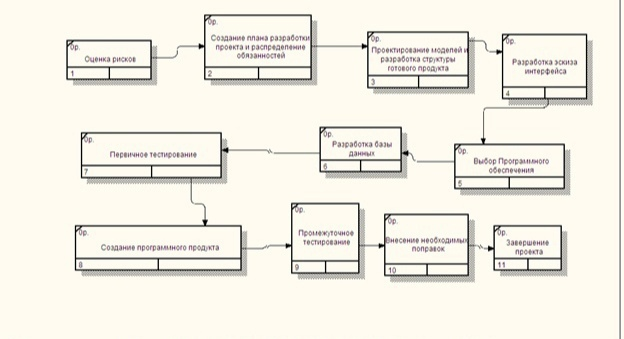
****

Рисунок 2 — Схема процесса разработки ПО.

## 2.2 Разработка бизнес-процессов в ELMA 365

В **ELMA 365** мы создали структуру компании, а также создали матрицу функциональной ответственности внутри команды, для того чтобы равномерно распределить все обязанности внутри команды.

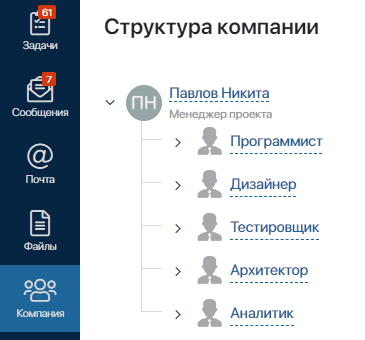


Рисунок 3 — Структура компании.

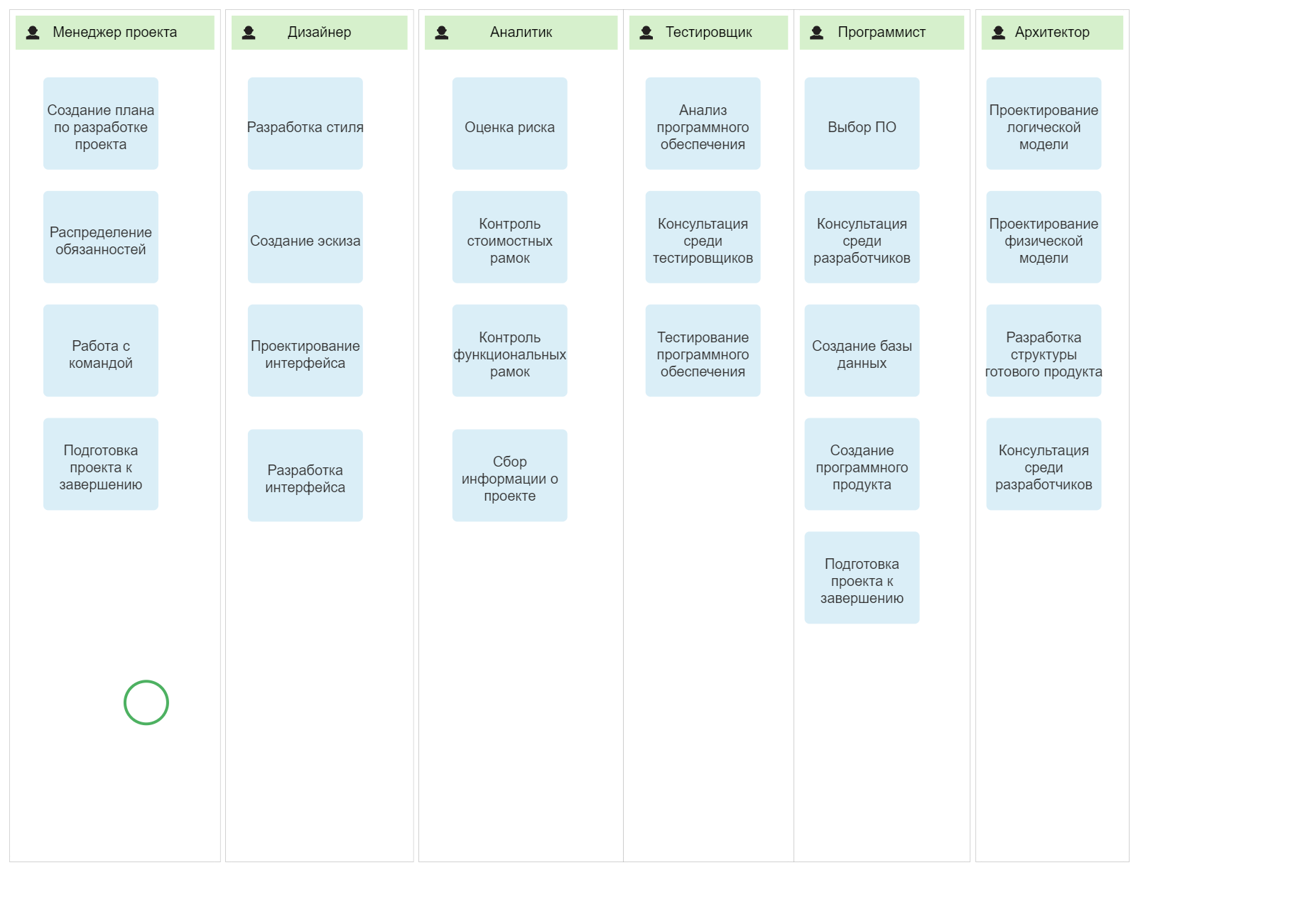


Рисунок 4 — Распределение зон ответственности.

## 2.3 Разработка интерфейса

После создания базы данных была начата разработка Windows-приложение «Гостиница» на языке программирования C# в интегрированной среде разработки (IDE) Visual Studio.

На главной форме мы видим 10 вкладок для выбора необходимых таблиц из БД, одну кнопку для загрузки данных из БД и три кнопки для взаимодействия с записями в базе и собственно элемент интерфейса, отвечающий за отображение данных (DataGridView) .

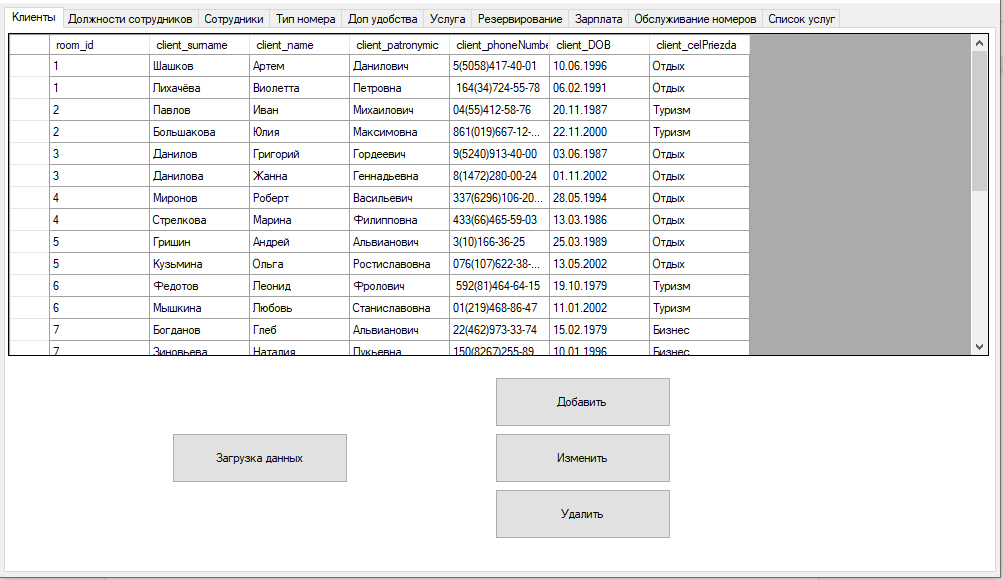


Рисунок 5 — Интерфейс приложения для просмотра забронированных номеров.

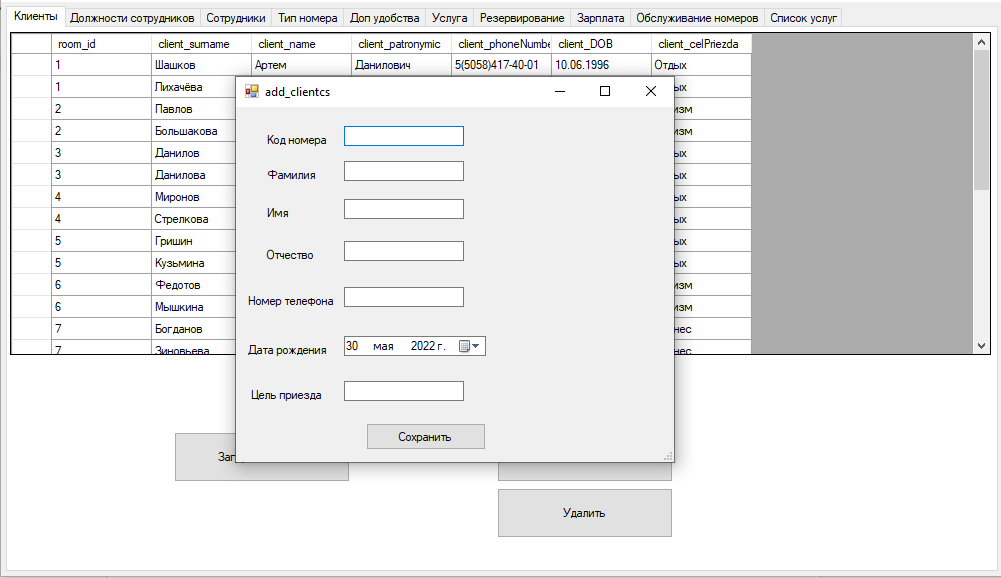


Рисунок 6 — Форма для редактирования бронирования.

## 

## 2.4 Методика оценки стоимости COCOMO II

**** Рисунок 7 - Сравнительные характеристики методов

Самым оптимальным на наш взгляд методом для оценки стоимости нашего проекта является метод COCOMO II, так как участие экспертов при оценке нужно лишь для калибровки параметров.

COnstructive COst MOdel (конструктивная модель стоимости) – это алгоритмическая модель оценки стоимости разработки программного обеспечения (ПО), разработанная Барри Боэмом (Barry Boehm). Модель использует простую формулу регрессии с параметрами, определенными из данных, собранных по ряду проектов.

COCOMO применим к трем классам проектов разработки ПО:

* Органический *(Organic mode)* – маленькие команды с хорошим опытом работы и не жесткими требованиями к разработке
* Полуразделенный вид *(Intermediate / Semi-detached mode)* – средние по размеру команды со смешанным опытом разработки и со смешанными требованиями (как жесткими, так и нет).
* Встроенный вид *(Intered / Embedded mode)* – разрабатываются с учетом множества жестких ограничений (по аппаратному, программному, операционному обеспечению и т.д.)

Формула оценки трудоемкости проекта в чел.×мес. имеет вид:

где

–факторы масштаба (ScaleFactors);

SIZE–объем программного продукта в тысячах строк исходного текста (KSLOC–KiloofSourceLineofCode);

–множители трудоемкости (EffortMultiplier);

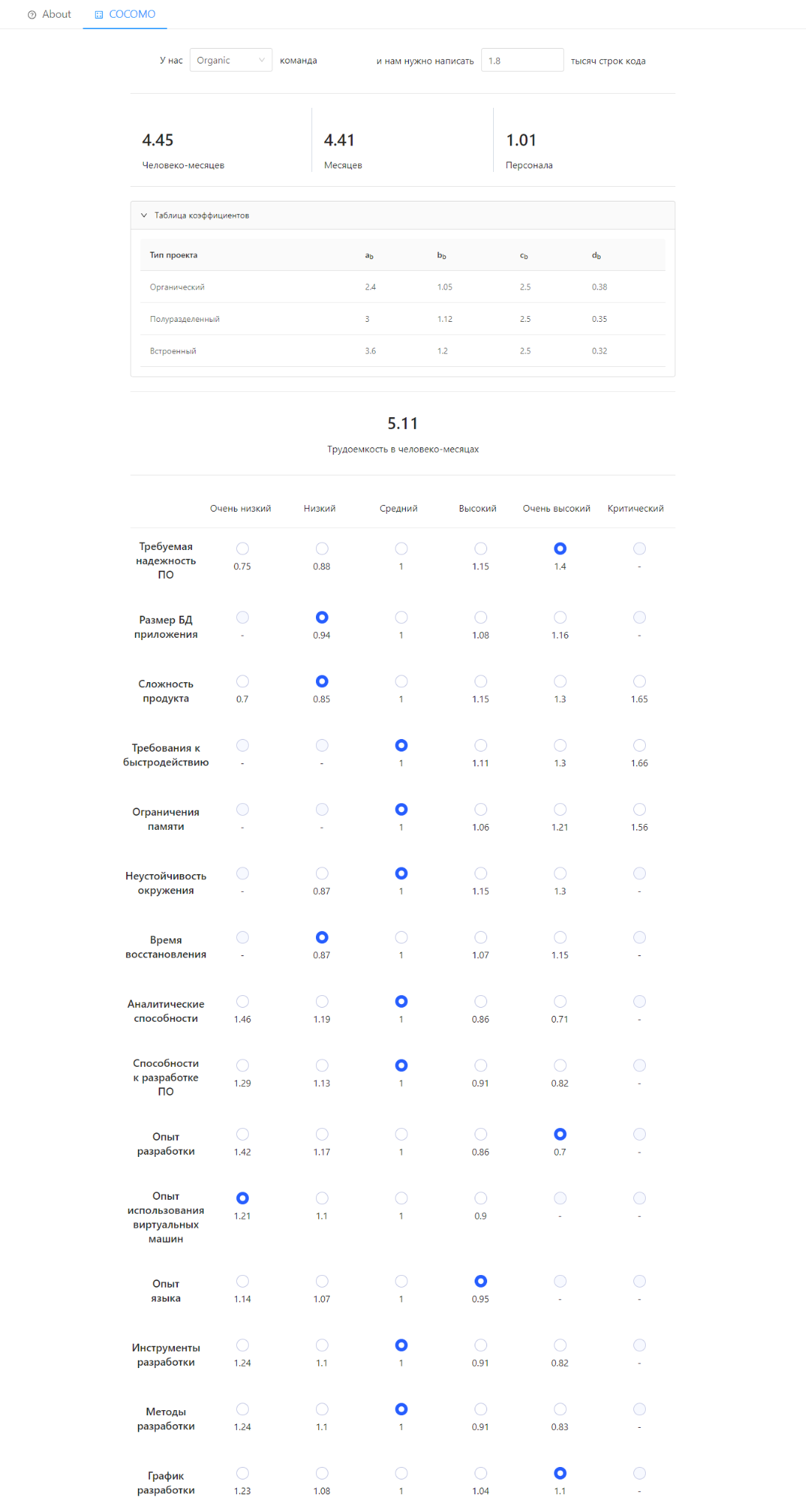
EAF(EffortAdjustmentFactor) –произведение выбранных множителей трудоемкости

Рисунок 8 — Расчеты COCOMO II методом.

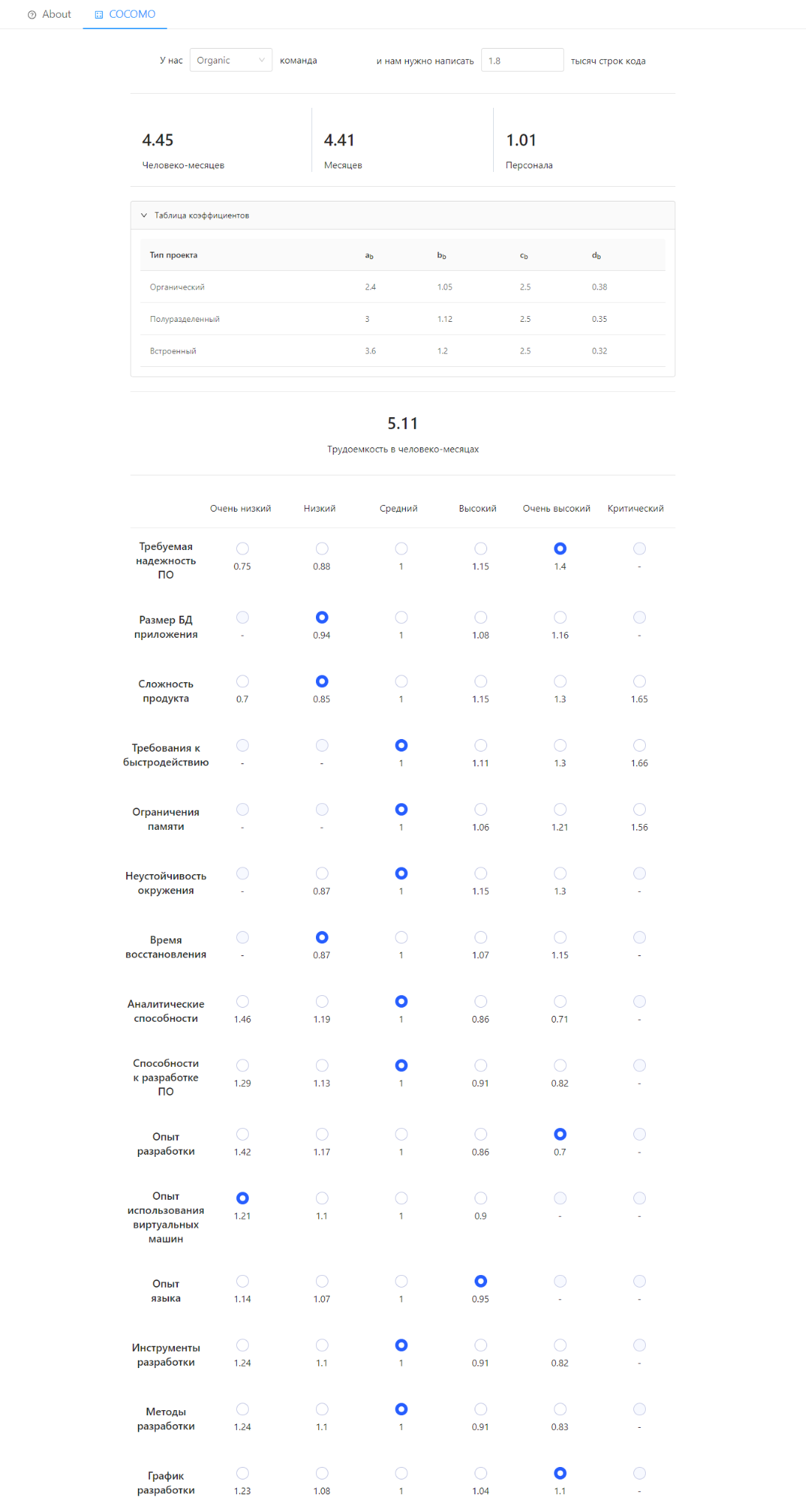


Рисунок 8 — продолжение рис.7 .

По методике COCOMO II проект будет выполнен 1 программистом

за 4.41 месяц.

## 

## 2.5 Оценка программного продукта методом функциональных точек

Оценка 14 системных характеристик:

| Номер | Системная характеристика | Описание | Оценка |
| --- | --- | --- | --- |
| 1 | Обмен данными | продукт обменивается данными по более, чем одному телекоммуникационному протоколу | 5 |
| 2 | Распределенная обработка данных | Распределенная обработка данных выполняется несколькими компонентами системы | 5 |
| 3 | Производительность | Пользовательские требования по производительность не установлены | 0 |
| 4 | Ограничения по аппаратным ресурсам | Нет ограничений | 0 |
| 5 | Транзакционная нагрузка | число транзакций велико и неравномерно, требуются специальные решения и инструменты | 5 |
| 6 | Интенсивность взаимодействия с пользователями | более 30% транзакций — интерактивные | 5 |
| 7 | Эргономика | Нет специальных требований | 0 |
| 8 | Интенсивность изменения данных пользователями | Изменения не интенсивные по восстановлению данных | 3 |
| 9 | Сложность обработки | Требования безопасности | 2 |
| 10 | Повторное использование | продукт разрабатывается как стандартный многоразовый компонент | 5 |
| 11 | Удобство инсталляции | Минимальные требования (Свободное место в памяти устройства) | 4 |
| 12 | Удобство администрирования | Программа имеет средние возможности самовосстановления | 3 |
| 13 | Портируемость | Система является распределенной и предполагает установку на различные «железо» и ОС | 3 |
| 14 | Гибкость | гибкая система запросов и построение произвольных отчетов, модель данных изменяется пользователем в интерактивном режиме | 5 |

Расчет суммарного эффекта 14 системных характеристик:

TDI = ∑DI = 45

Расчет значения фактора выравнивания:

VAF = (TDI\*0,01) +0,65 = (45\*0,01) +0,65 = 1,1

UFP = 1,1\*4 = 4,4

Расчет количества выровненных функциональных точек проекта разработки:

DFP = UFP\*VAF = 4,4\*1,1 = 4,84 чел\*мес

Вывод по результату оценки методом функциональных точек: требуемые трудозатраты на разработку программного продукта составляют 4,84 чел\*мес. А это значит, что разработка программного продукта будет выполнена группой из 4 человек за 27 дней.

# Заключение

В ходе данного проекта мы прошли все этапы работы начиная от распределения ролей в команде и заканчивая созданием готовой информационной системы.

Как результат нашей работы, потенциальный клиент может получит полностью готовое программное обеспечение, которое он может использовать в целях увеличения эффективности его предприятия

# Приложения

**Техническое задание**

**Техническое задание к программе**

**"Программное обеспечение для для сервиса онлайн бронирования номеров гостиницы "**

Содержание

1. Введение

1.1. Наименование программы

1.2. Назначение и область применения

2. Требования к программе

2.1. Требования к функциональным характеристикам

2.2. Требования к надежности

2.2.1. Требования к обеспечению надежного функционирования программы

2.2.2. Время восстановления после отказа

2.2.3. Отказы из-за некорректных действий оператора

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

3.2. Требования к квалификации и численности персонала

3.3. Требования к составу и параметрам технических средств

3.4. Требования к информационной и программной совместимости

3.4.1. Требования к информационным структурам и методам решения

3.4.2. Требования к исходным кодам и языкам программирования

3.4.3. Требования к программным средствам, используемым программой

3.4.4. Требования к защите информации и программ

3.5. Специальные требования

4. Требования к программной документации

4.1. Предварительный состав программной документации

5. Технико-экономические показатели

5.1. Экономические преимущества разработки

6. Стадии и этапы разработки

6.1. Стадии разработки

6.2. Этапы разработки

6.3. Содержание работ по этапам

7. Порядок контроля и приемки

7.1. Виды испытаний

7.2. Общие требования к приемке работы

1. Введение

1.1. Наименование программы

Наименование программы: "Программное обеспечение для сервиса онлайн бронирования номеров гостиницы "

1.2. Назначение и область применения

Автоматизация работы гостиницы

2. Требования к программе

2.1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

•​Хранение данных

•​Просмотр информации о сотрудниках

•​Просмотр информации об услугах

•​Просмотр информации о клиентах

Добавление, сохранение и удаление записей

•​Переход на предыдущую форму

2.2. Требования к надежности

Отказоустойчивость при повышенном количестве обращений

2.2.1 Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

а) организацией бесперебойного питания технических средств;

б) использованием лицензионного программного обеспечения;

в) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г.

Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

г) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов

2.2.2. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы,

не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

2.2.3. Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой.

Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям,

предъявляемым к техническим средствам в части условий их эксплуатации

3.2. Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и конечный пользователь программы — оператор.

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

а) задача поддержания работоспособности технических средств;

б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;

в) задача установки (инсталляции) программы.

г) задача создания резервных копий базы данных.

3.3. Требования к составу и параметрам технических средств

3.3.1. В состав технических средств должен входить персональный компьютер (ПЭВМ), выполняющий роль сервера, включающий в себя:

3.3.1.1. процессор Xeon E5 v3, не менее 2.5 Ghz;

3.3.1.2. оперативную память объемом, 128 Гб, не менее;

3.3.1.3. свободного пространства на жестком диске, 4тб, не менее;

3.3.1.4. операционную систему Windows 2019 Server или Windows 10;

3.4. Требования к информационной и программной совместимости

3.4.1. Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server. Используется многопоточный доступ к базе данных. Необходимо обеспечить одновременную работу с программой с той же базой данной модулей экспорта внешних данных.

3.4.2. Требования к исходным кодам и языкам программирования

Дополнительные требования не предъявляются

3.4.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 2019 Server или Windows 10 и Microsoft SQL Server 2014

3.4.4. Требования к защите информации и программ

Требования к защите информации и программ не предъявляются

3.5. Специальные требования

Специальные требования к данной программе не предъявляются

4. Требования к программной документации

4.1. Предварительный состав программной документации

Состав программной документации должен включать в себя:

4.1.1. техническое задание;

4.1.2. программу и методики испытаний;

4.1.3. руководство оператора;

5. Технико-экономические показатели

5.1. Экономические преимущества разработки

Ориентировочная экономическая эффективность не рассчитываются. Аналогия не проводится ввиду уникальности предъявляемых требований к разработке.

6. Стадии и этапы разработки

6.1. Стадии разработки

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

6.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы

6.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на неё;

5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение и методики испытаний;

2. проведение приемо-сдаточных испытаний;

3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

7. Порядок контроля и приемки

7.1. Виды испытаний

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний

7.2. Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.