**1** **Анализ постановки задачи проектирования ИС и исследование объекта автоматизации**

Информационная система предназначена для оптимизации рабочих процессов, бизнес-процессов; для строго регламентируемого учёта и хранения данных, сводя к минимуму при этом человеческий фактор в данном аспекте. Именно поэтому большое количество компаний заинтересованы в разных видах информационных системах. Причём компания может иметь и несколько информационных систем, каждая из которых может решать свои узконаправленные задачи. Такие системы внутри компании, как правило, связаны между собой интеграционным слоем для совместной работы с данными и оптимизации процессов.

**1.1 Классификация информационных систем с точки зрения проектирования и предназначения**

Информационная система (ИС) - это система, предназначенная для хранения, поиска и обработки информации.

ИС предназначена для своевременного обеспечения людей необходимой информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определённой предметной области, при этом результатом функционирования информационных систем является информационная продукция - документы, информационные массивы, базы данных и информационные услуги.

Информационные системы могут классифицироваться по:

* Архитектуре;
* По степени автоматизации;
* По характеру обработки данных;
* По сфере применения;
* По охвату задач.

Классификация по архитектуре подразумевает настольные и распределённые ИС. Распределённые информационные системы в свою очередь подразделяются с архитектурной точки зрения на:

* Файл-серверные;
* Клиент-серверные.

В файл-серверных ИС база данных находится на файловом сервере, а СУБД и клиентские приложения находятся на рабочих станциях. В клиент-серверных ИС база данных и СУБД находятся на сервере, а на рабочих станциях находятся только клиентские приложения.

В свою очередь, клиент-серверные ИС разделяют на двухзвенные и многозвенные. В двухзвенных ИС всего два типа «звеньев»: сервер базы данных, на котором находятся БД и СУБД (back-end), и рабочие станции, на которых находятся клиентские приложения (front-end). Клиентские приложения обращаются к СУБД напрямую. В многозвенных ИС добавляются промежуточные «звенья»: серверы приложений. Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями. Типичный пример применения трёхзвенной архитектуры - современные веб-приложения, использующие базы данных. В таких приложениях помимо звена СУБД и клиентского звена, выполняющегося в веб-браузере, имеется как минимум одно промежуточное звено - веб-сервер с соответствующим серверным программным обеспечением.

Информационные системы подразделяются по степени автоматизации на:

* Автоматизированные (информационные системы, в которых автоматизация может быть неполной, то есть требуется постоянное вмешательство персонала);
* Автоматические (информационные системы, в которых автоматизация является полной, то есть вмешательство персонала не требуется или требуется только эпизодически).

По характеру обработки данных ИС классифицируются на:

* Информационно-справочные или информационно-поисковые ИС, в которых нет сложных алгоритмов обработки данных, а целью системы является поиск и выдача информации в удобном виде;
* ИС обработки данных, или решающие ИС, в которых данные подвергаются обработке по сложным алгоритмам.

По сфере применения информационные системы подразделяются на:

* Экономическая ИС – это такая информационная система, которая предназначенная для выполнения функций управления на предприятии;
* Медицинская ИС – это такая информационная система, которая предназначенная для использования в лечебном или лечебно-профилактическом учреждении;
* Географическая ИС – это такая информационная система, которая обеспечивает сбор, хранение, обработку, доступ, отображение и распространение пространственно-координированных данных.

Информационные системы также подразделяются по охвату задач на:

* Персональная;
* Групповая;
* Корпоративная.

Персональная ИС предназначена для решения некоторого круга задач одного человека. Групповая ИС ориентирована на коллективное использование информации членами рабочей группы или подразделения. Корпоративная ИС автоматизирует все бизнес-процессы целого предприятия (компании или организации) или их значительную часть, достигая их полной информационной согласованности и прозрачности. Такие системы иногда называют информационными системами предприятия и системами комплексной автоматизации предприятия

**1.2 Этапы проектирования информационных систем**

Исходя из вышеперечисленных классификаций по различным признакам информационных систем, определим классификации по этим признакам для разрабатываемой ИС данного курсового проекта.

С точки зрения архитектуры разрабатываемая ИС является клиент-серверной, так как её архитектура подразумевает распределение задач и сетевой нагрузки между поставщиками услуг (микросервисов) и заказчиками услуг (клиентами).

По степени автоматизации информационная система автоматизированная, так система подразумевает неполную автоматизацию.

По характеру обработки данных ИС относится как к информационным системам обработки данных, так и к информационно справочным ИС, потому что система содержит алгоритмы обработки данных, а также может выдавать данные в удобном виде.

Так как разрабатываемая информационная система выполняет функции управления предприятием, она относится к экономическим ИС.

По охвату задач ИС относится к корпоративным информационным системам, так как она способна автоматизировать значительную часть процессов футбольного клуба, достигая их полной информационной согласованности и прозрачности.

Совокупность стадий и этапов, которые проходит ИС в своем развитии от момента принятия решения о создании системы до момента прекращения функционирования системы, называется жизненным циклом ИС.

Содержание жизненного цикла разработки ИС сводится к выполнению следующих стадий:

1. Планирование и анализ требований (предпроектная стадия) ─ системный анализ. Проводится исследование и анализ существующей информационной системы, определяются требования к создаваемой ИС, формируются технико-экономическое обоснование (ТЭО) и техническое задание (ТЗ) на разработку ИС;
2. Проектирование (техническое и логическое проектирование). В соответствии с требованиями формируются состав автоматизируемых функций (функциональная архитектура) и состав обеспечивающих подсистем (системная архитектура), проводится оформление технического проекта ИС;
3. Реализация (рабочее и физическое проектирование, кодирование). Разработка и настройка программ, формирование и наполнение баз данных, формулировка рабочих инструкций для персонала, оформление рабочего проекта;
4. Внедрение (опытная эксплуатация). Комплексная отладка подсистем ИС, обучение персонала, поэтапное внедрение ИС в эксплуатацию по подразделениям организации, оформление акта о приемо-сдаточных испытаниях ИС;
5. Эксплуатация ИС (сопровождение, модернизация). Сбор рекламаций и статистики о функционировании ИС, исправление недоработок и ошибок, оформление требований к модернизации ИС и ее выполнение (повторение стадий 2-5).

Сам жизненный цикл состоит из следующих стадий:

1. Системный анализ;
2. Анализ требований;
3. Проектирование;
4. Реализация;
5. Тестирование;
6. Сопровождение.

Все эти стадии представляют из себя связанный список последовательных задач, которые необходимо строго реализовывать одна за другой.

Системный анализ ИС начинается с описания и анализа функционирования рассматриваемого объекта в соответствии с требованиями (целями), которые предъявляются к нему. В результате этого этапа выявляются недостатки существующей ИС, на основе которых формулируется потребность в совершенствовании системы управления этим объектом, и ставится задача определения экономически обоснованной необходимости автоматизации определенных функций управления (создается технико-экономическое обоснование проекта ИС). После определения этой потребности возникает проблема выбора направлений совершенствования объекта на основе выбора программно-технических средств. Результаты оформляются в виде технического задания на проект, в котором отражаются технические условия и требования к ИС, а также ограничения на ресурсы проектирования. Требования к ИС определяются в терминах функций, реализуемых системой.

Анализ требований - предполагает сбор требований к программному обеспечению (ПО), их систематизацию, выявление взаимосвязей, а также документирование.

Проектирование информационной системы - это начальный этап разработки. В его рамках осуществляется совокупность работ по подготовке технического задания и разработке системы, результатом которых становится ее прототип. В рамках проектирования ИС также зачастую разрабатывается технический дизайн и функциональный дизайн системы. Технический дизайн системы – это техническое описание ИС, которому должны придерживаться все разработчики при реализации системы. Функциональный дизайн – это основной шаблон проектирования информационной системы, который значительно упрощает проектирование ИС. ФД описывает каждый программный модуль, а также гарантирует, что каждый такой модуль имеет единственную ответственность (первый принцип проектирования ПО SOLID, который называется «Single responsibility» - принцип единственной ответственности) и не оказывает побочного влияния на остальные программные модули, или это влияние сводится к минимуму (последний принцип SOLID - dependency inversion principle – принцип инверсии зависимостей).

Разработка (кодирование) ИС – это непосредственно сама реализация информационной системы согласно требованиям, уже составленным на этапе проектирования техническому заданию (ТЗ), техническому дизайну (ТД) и функциональному дизайну (ФД). Также в рамках непосредственной реализации ИС разрабатываются unit-тесты, UI-тесты, интеграционные тесты и так далее. Эти тесты позволяют на этапе разработки находить ошибки в реализации, расхождения с ТД и ФД.

Тестирование информационной системы подразумевает пользование системой перед её сдачей Заказчику с целью выявления ошибок для последующей отладки ИС. На стадии тестирования зачастую запускают ещё раз unit-тесты, UI-тесты, интеграционные тесты и т. п. для того чтобы убедиться, что система работает корректно согласно покрывающим её тестам.

Сопровождение информационной системы предназначено для оптимизации и устранения дефектов ИС после сдачи в эксплуатацию. В ходе сопровождения в информационную систему вносятся изменения, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повышения удобства использования и применимость ИС.

**1.3 Методологии ведения проектов по реализации ИС**

На сегодняшний день существует большое множество методологий ведения проектов по реализации ИС. Самые состоятельные и популярные из них – это Waterfall, Agile, Scrum. Данные методологии включают в себя все стадии, описанные выше, а именно системный анализ, анализ требований, проектирование, разработка (кодирование), тестирование и сопровождение.

**1.3.1 Методология Waterfall**

Методология Waterfall (в переводе означает водопад) ведёт реализацию информационной системы одним большим потоком без возвращения к предыдущим стадиям (Рисунок 1).

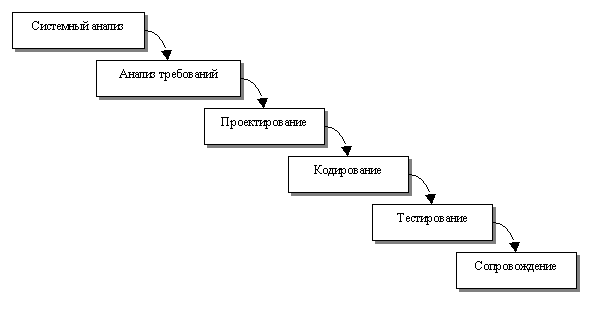


Рисунок 1 – Схема выполнения стадий по методологии Waterfall

Эта методология считается устаревшей, но до сих пор многие проекты по реализации ИС используют Waterfall. Данная методология не требует больших временных затрат на декомпозицию задач и ведение спринтов, что позволяет сразу же приступить к выполнению проекта. Но у данной методологии есть существенные, как я считаю, минусы, а именно – отсутствие гибкости при изменении требований; отсутствие возможности возвращения к предыдущим стадиям; отсутствие распараллеливания задач между несколькими командами; большой объём задач в огромном и единственном потоке. Почему же эту методологию всё ещё используют? Ответ простой – её используют в маленьких проектах, которые не имеют большие бюджеты и длительные сроки реализации. Проще говоря, Waterfall используют для проектов, которые нужно сделать «здесь и сейчас».

**1.3.2 Методология Agile**

Методология Agile подразумевает ведение спринтов (итераций) при реализации ИС. Спринт – это промежуток времени, за который должен быть сделан определённый набор задач. Каждый спринт (итерация) содержит в себе стадии жизненного цикла, а также может содержать и другие стадии, как например, планирование спринта и декомпозиция задач. Когда спринт завершается, сразу же начинается следующий спринт (Рисунок 2), и так будет продолжаться пока разработка ИС не завершится.

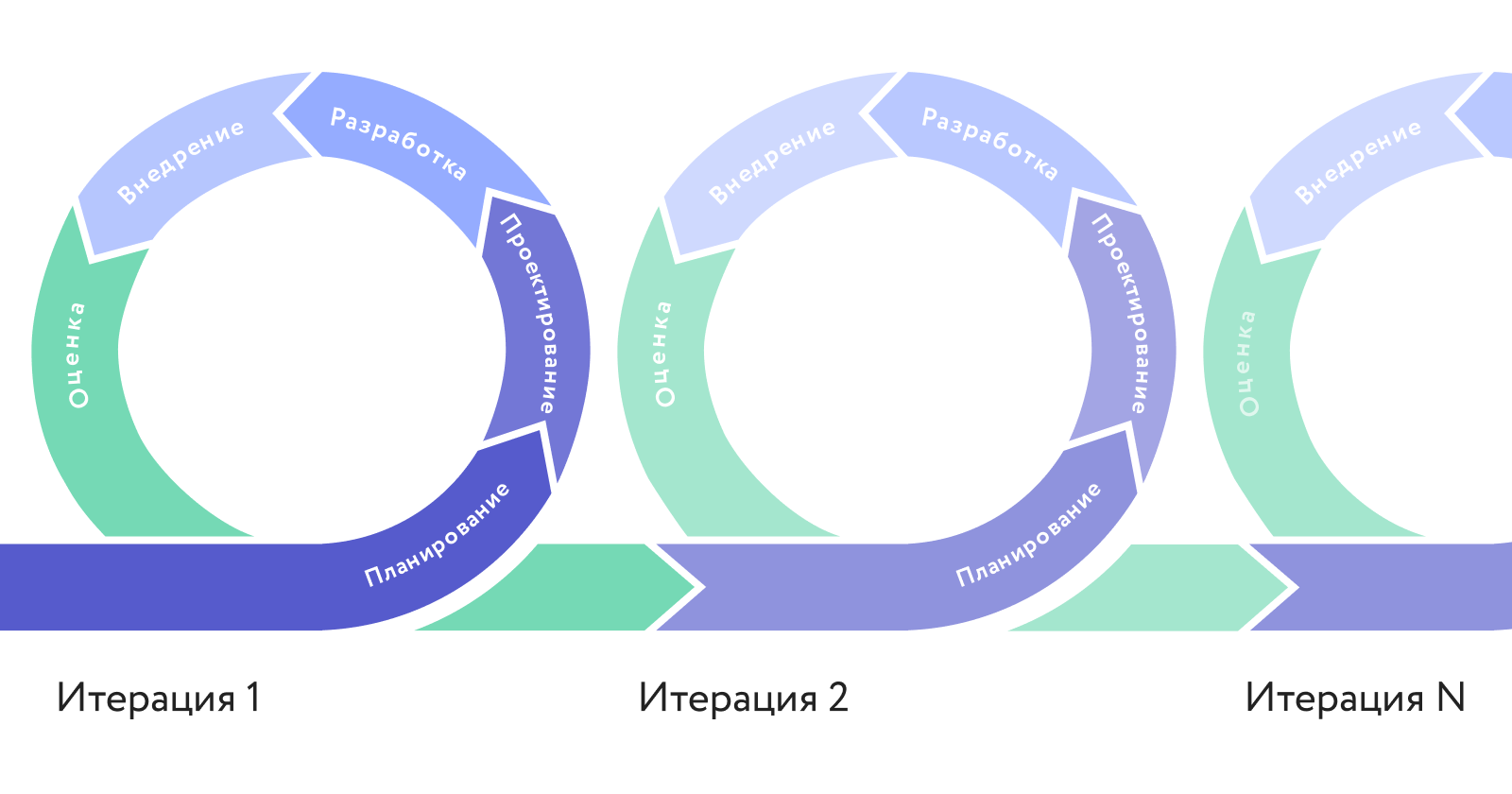


Рисунок 2 – Схема Agile

Данная методология имеет существенные плюсы – гибкость под изменяющиеся требования Заказчика; быстрота исправления ошибок в системе; регулярные поставки Заказчику, которые позволят ещё в процессе разработки пользоваться работающей частью ИС с целью выявления ошибок, корректировки требований, оптимизации и непосредственно работы, ведь уже готовую часть информационной системы можно использовать на production. Также можно сделать вывод, что такая система облегчит дальнейшее сопровождение после сдачи готовой ИС, так как пользователи в процессе разработки смогут оставлять свои отзывы, ошибки могут оперативно правится в следующем спринте. Тоже самое касается и оптимизации изменения требований. Минусом данной системы может оказаться большое количество итераций (спринтов) и даже больше, чем запланировано, так как у Заказчика появляется возможность что-то менять в ходе разработки.

**1.3.3 Методология Scrum**

SCRUM («схватка» - термин из регби) - минимально необходимый набор мероприятий, артефактов, ролей, на которых строится процесс SCRUM-разработки, позволяющий за фиксированные небольшие промежутки времени, называемые спринтами (sprints), предоставлять конечному пользователю работающий продукт с новыми бизнес-возможностями, для которых определён наибольший приоритет. Методология базируется на командной работе, по аналогии с тем, как в регби команда действует сообща, ради достижения общей цели. Возможности к реализации в очередном спринте определяются командой в начале спринта на совещании по планированию спринта Sprint Planning Meeting. Для оценки предстоящего объёма работ на спринте чаще всего используются относительные оценки, и практика покера планирования (Planning Poker).

В конце спринта Scrum-команда встречается на обзорном совещании результатов спринта (Sprint Review — старое название Demonstration) с заказчиком, и представляет ему инкремент бизнес-продукта (версия продукта с законченным набором функциональности, который уже можно отдавать заказчику и пользователю для использования), который она успела сделать за спринт. Цель Sprint Review — получение обратной связи от заказчика, чтобы понять, на чём нужно делать акцент в дальнейшем, и какой должен быть следующий инкремент бизнес-продукта.

Строго фиксированная небольшая длительность спринта (от 1 до 4 недель) снижает риски, и даёт возможность быстро получить обратную связь от заказчика, чтобы скорректировать видение продукта.

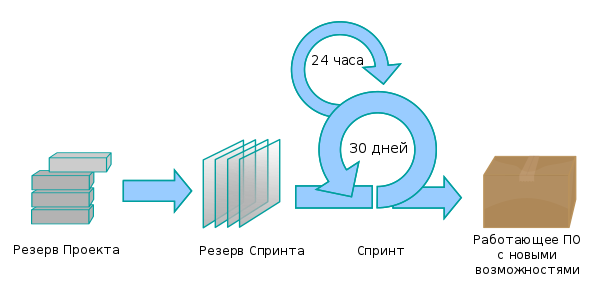


Рисунок 3 – Схема методологии Scrum

**1.4 Предпроектное обследование организации**

Выполним предпроектное обследование компании, изучив цели и задачи предприятия, его организационную структуру и основные бизнес-процессы.

**1.4.1 Описание организации**

«Локомотив» - советский и российский профессиональный футбольный клуб из Москвы. Выступает в Российской премьер-лиге. Основан 23 июля 1922 года. Один из старейших футбольных клубов России, ведущий свою историю с момента основания команды «Казанка» при Московско-Казанской железной дороге, затем неоднократно менявший своё название. Трёхкратный чемпион России, одиннадцатикратный обладатель Кубка страны: двукратный обладатель Кубка СССР и девятикратный обладатель Кубка России, трёхкратный обладатель Суперкубка России. Дважды полуфиналист Кубка обладателей кубков.

Миссия футбольного клуба – завоёвывание уверенных побед в Российской премьер-лиге.

**1.4.2 Стратегические цели и задачи организации**

Цель футбольного клуба «Локомотив» - усиление футбольной команды, 70% побед в Российской премьер-лиге, повышение количества разницы мячей (забитые мячи – пропущенные мячи), победа в Российской премьер-лиге.

Стратегические цели и направления действий по их достижению представлены на Рисунке 4.

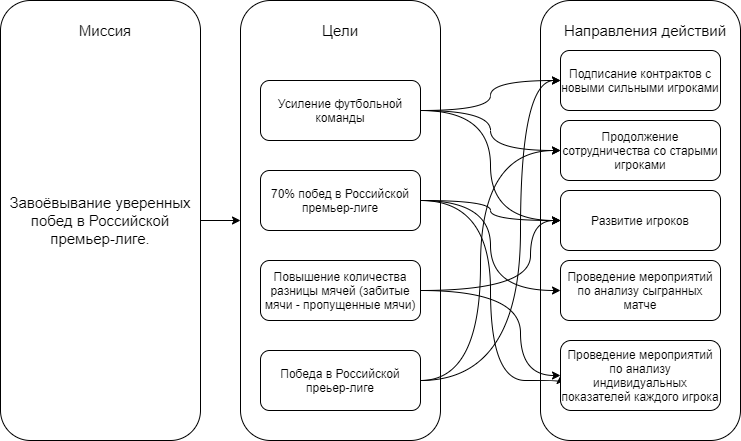


Рисунок 4 – Стратегические цели и направления действий по их достижению

**1.4.2 Организационная структура футбольного клуба**

Организационная структура футбольного клуба «Локомотив» представлена на Рисунке 5.

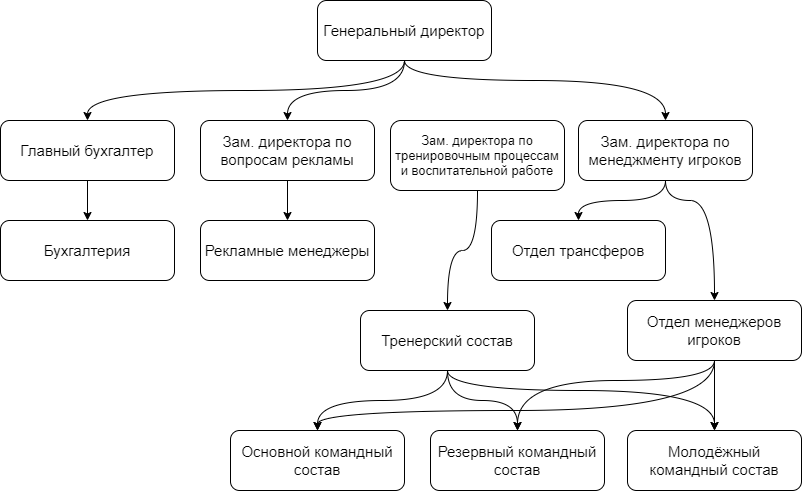


Рисунок 5 - Модель организационной структуры

Возглавляет футбольный клуб генеральный директор. В его подчинении находятся различные заместители: зам. директора по вопросам рекламы, зам. директора по тренировочным процессам и воспитательной работе, зам. директора по менеджменту игроков. Также в подчинении генерального директора находится главный бухгалтер.

В подчинении у главного бухгалтера находится отдел бухгалтерии, который выполняет все финансовые операции.

В подчинении у зам. директора по вопросам рекламы находятся рекламные менеджеры. Их задача – привлечение в клуб инвестиций от рекламодателей.

Тренерский состав подчиняется зам. директора по тренировочным процессам и воспитательной работе. Тренерский состав руководит всеми командными составами футбольного клуба, а именно основным составом, резервным составом и молодёжным составом.

В подчинении у зам. директора по менеджменту игроков находится отдел менеджеров, который решает различные организационные внутренние и внешние вопросы, касающиеся каждого игрока. Основной командный состав, резервный командный состав и молодёжный командный состав подчиняются помимо тренерского состава ещё и отделу менеджеров игроков.

**1.4.3 Анализ информационного обеспечения компании**

В своей деятельности отделы используют различное программное обеспечение, которое отвечает специфическим требованиям каждого отдела.

Бухгалтерия использует программу 1С: Бухгалтерия, 1С: Зарплата и Кадры, БухСофт: Зарплата и Кадры, Отдел кадров. На предприятии имеются 10 компьютеров снабженных выходом в интернет, 3 сканера и принтера.

**1.4.4 Основные бизнес-процессы в логистике**

К основным бизнес-процессам футбольного клуба относится:

* Процесс подписания контрактов;
* Участие в регулярных матчах;
* Ведение статистики матчей;
* Ведение индивидуальной статистики игроков;
* Ведение и анализ штрафов и дисквалификаций;
* Контроль положения в турнирной таблице.

**2 Разработка модели проектируемой ИС**

**2.1 Построение функциональной модели**

Методология IDEF0 представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель IDEF0 отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

IDEF0 может быть использована для моделирования широкого класса систем. Для новых систем применение IDEF0 имеет своей целью определение требований и указание функций для последующей разработки системы, отвечающей поставленным требованиям и реализующей выделенные функции. Применительно к уже существующим системам IDEF0 может быть использована для анализа функций, выполняемых системой, и отображения механизмов, посредством которых эти функции выполняются. Основу методологии IDEF0 составляет графический язык описания бизнес-процессов.

Модель бизнес-процессов футбольного клуба представлена на Рисунках 6-8.

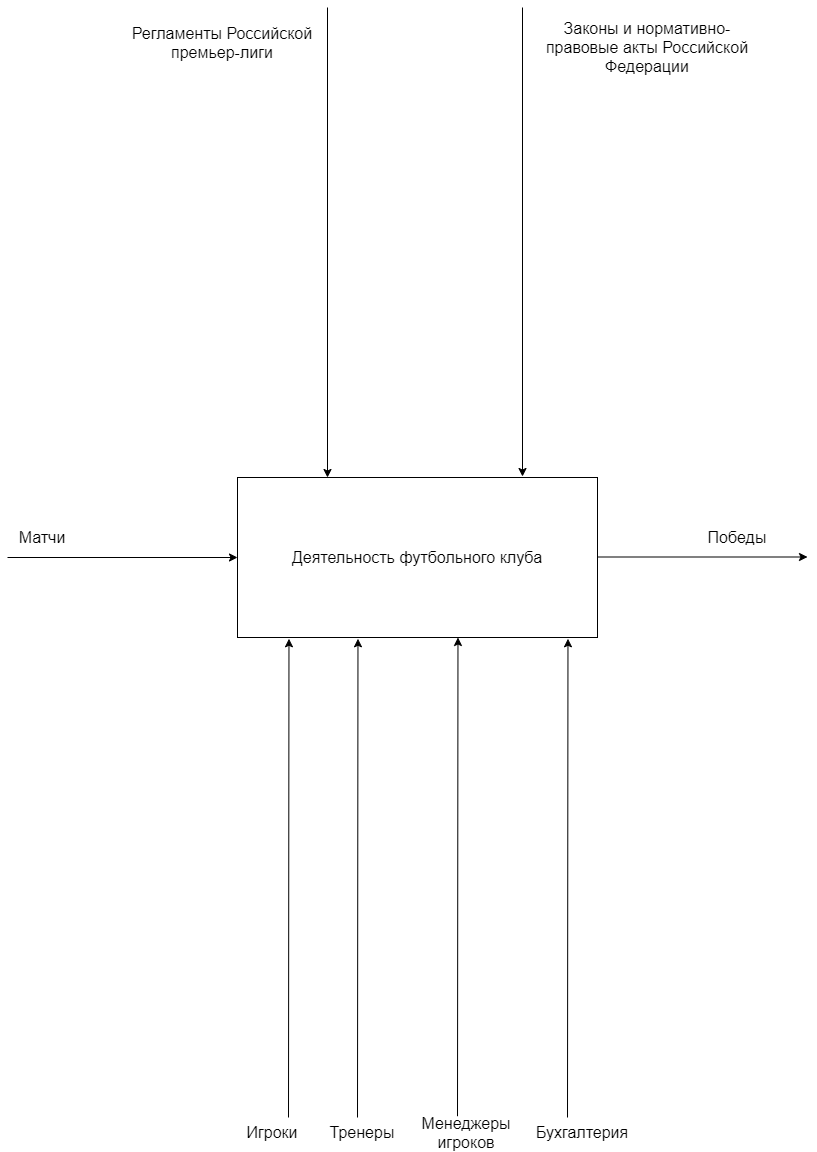


Рисунок 6 – Контекстная диаграмма

При декомпозиции главной бизнес-функции выделяются следующие функциональные блоки: оформление участия в Российской премьер-лиге, усиление команды, работа над максимальной результативность. в Российской премьер-лиге.

Диаграмма декомпозиции деятельности футбольного клуба показана на рисунке 7.

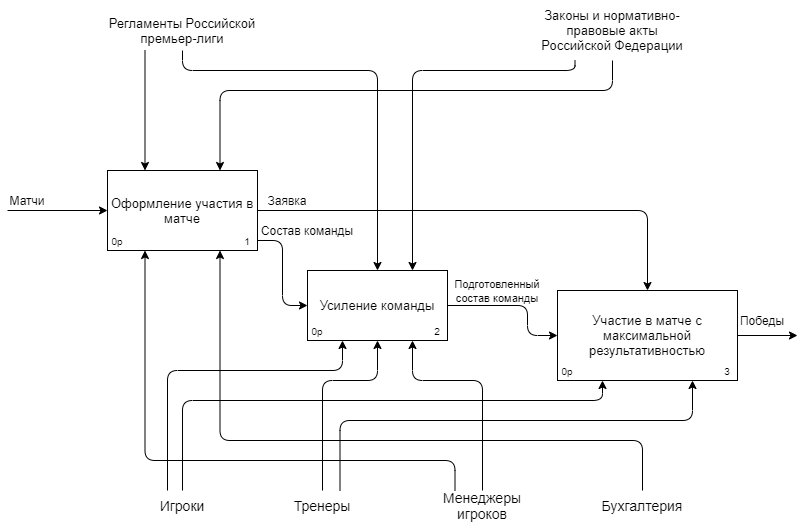


Рисунок 7 - Диаграмма декомпозиции первого уровня

Декомпозиция усиления команды представлена на Рисунке 8. При усилении команды могут проводиться специализированные предматчевые тренировки, которые способствуют подготовке командного состава к грядущему матчу. Оформление контракта со старым игроком, если этот контракт подходит к концу, способствует усилению команды. Оформление контракта с новым игроком также может усилить команду. После этого необходимо провести ряд мероприятий по интеграции этого игрока в команду. Анализ командного взаимодействия и индивидуальных показателей в предыдущих матчах помогает командному составу провести работу над ошибками и принять меры по их недопущению вновь.

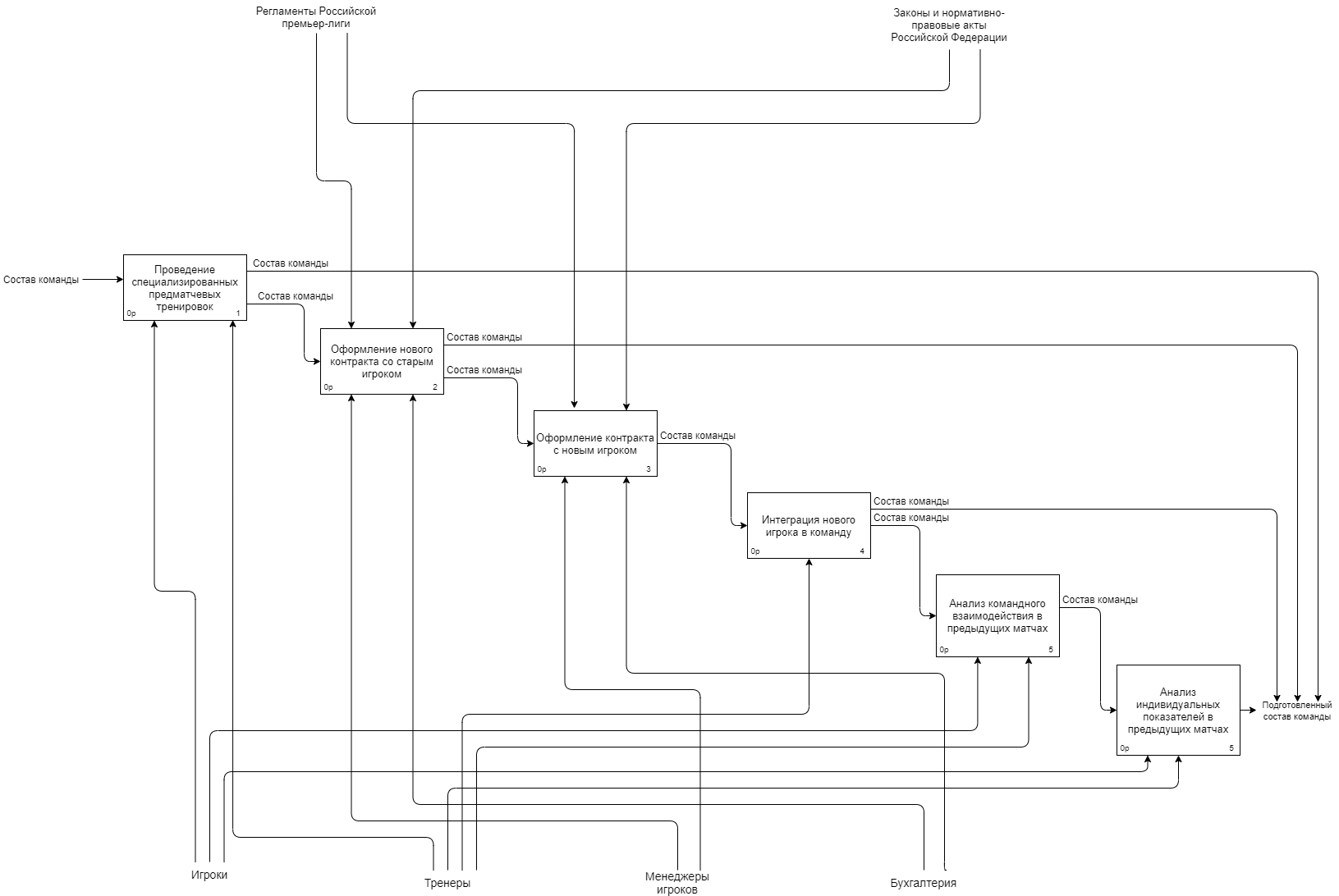


Рисунок 8 – Диаграмма декомпозиции второго уровня

Анализ бизнес процессов облегчает понимание работы футбольного клуба, и помогает в проектировании и разработке информационной системы.

**2.2 Построение модели потоков данных**

При построении функциональной модели системы альтернативой методологии IDEF0 является методология диаграмм потоков данных (Data Flow Diagrams, DFD). В отличие от IDEF0, предназначенной для проектирования систем вообще, DFD предназначена для проектирования информационных систем.

Основу методологии DFD составляет графический язык описания процессов.

Информационная модель системы в нотации DFD представляется в виде набора подсистем, потоки данных переносят информацию от одной подсистемы к другой.

Каждая из подсистем выполняет определенные преобразования входного потока данных и передает результаты обработки информации в виде потоков данных для других подсистем.

Модель потоков данных футбольного клуба представлена на Рисунке 8.

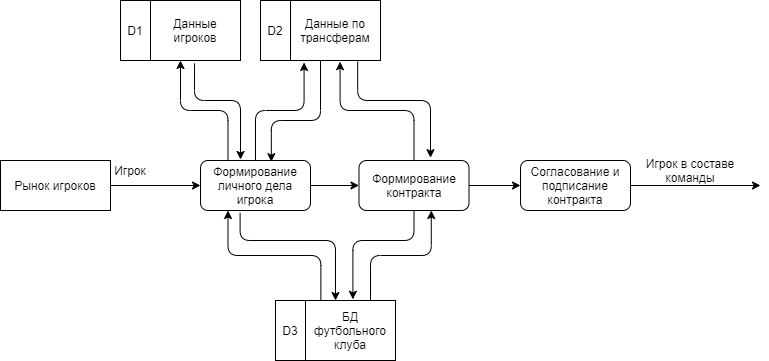


Рисунок 8 – Модель потоков данных

**2.3 Построение информационной модели системы**

Разработанная функциональная модель системы отвечает на вопросы «Что должна делать система?» и «За счет, каких действий, может быть, достигнут требуемый результат?». Эта модель также позволяет концептуально определить наборы данных, используемых в системе.

В то же время она не отвечает на вопрос: «Каким образом организованы данные в системе?». Для ответа на него необходимо построить информационную модель (запроектировать БД).

Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы "сущность-связь" (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи).

ERD непосредственно используются для проектирования реляционных баз данных.

ER диаграмма БД футбольного клуба представлена на Рисунке 9.

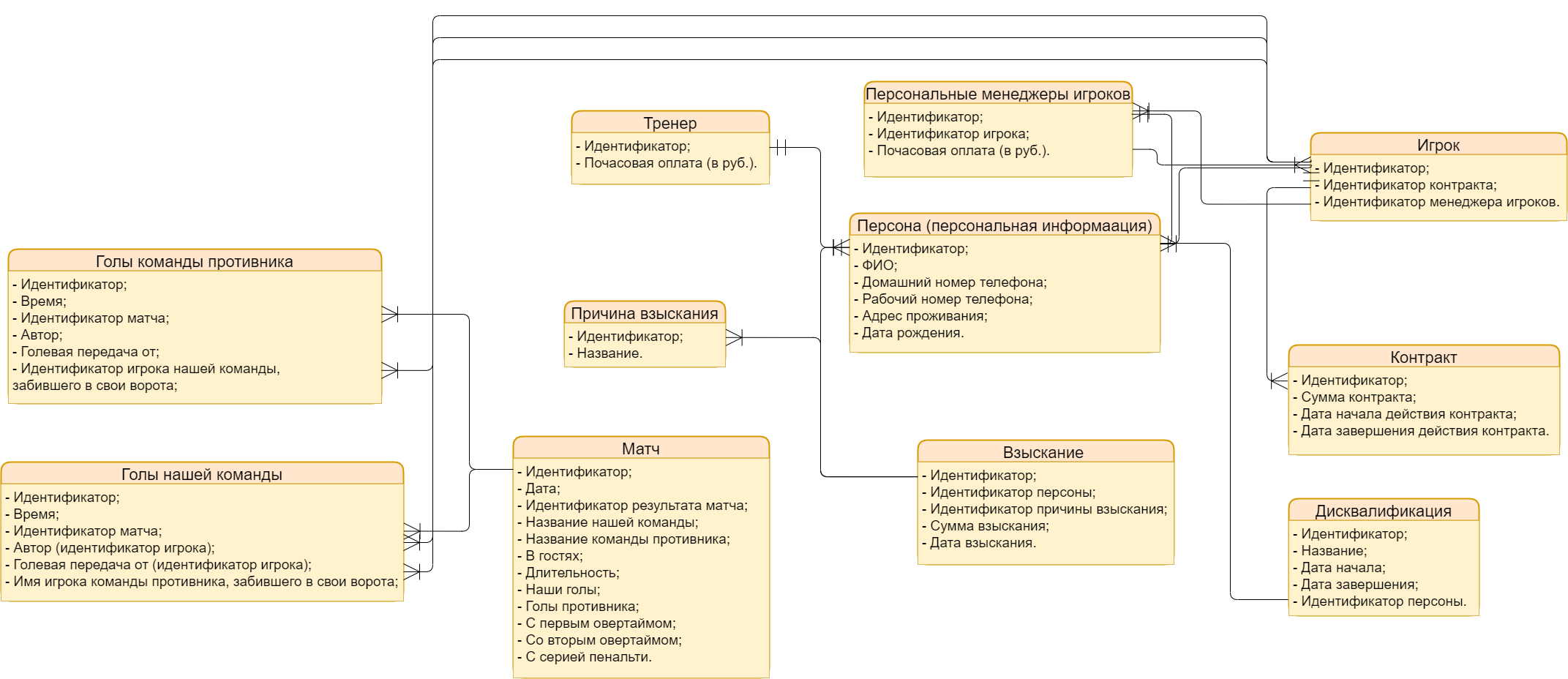


Рисунок 9 – ER- диаграмма БД футбольного клуба

**2.4 Разработка модели функционирования системы на основе диаграммы вариантов использования**

На первом этапе объектно-ориентированного процесса анализа и проектирования информационной системы, главной целью которого является представление совокупности функциональных требований к поведению проектируемой системы, разрабатывается модель вариантов использования.

Диаграмма вариантов использования (Use Case Diagram) является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и описывает функциональное назначение системы или что система будет делать в процессе своего функционирования.

Диаграмма Use Case представляет собой последовательность действий (функция ИС или прецедентов), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом или актером) и представляет собой граф специального вида, который является графической нотацией для представления конкретных вариантов использования, актеров, возможно, некоторых интерфейсов, и отношений между этими элементами.

Диаграмма вариантов использования ИС футбольного клуба представлена на Рисунке 10.

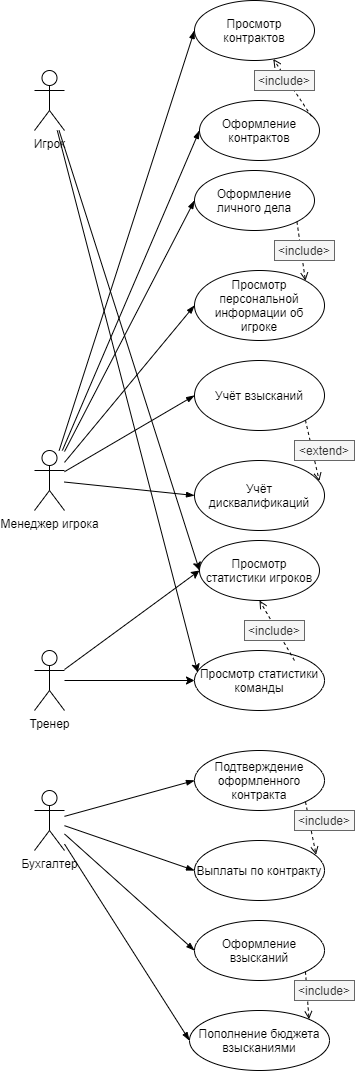


Рисунок 10 – Диаграмма вариантов использования

**2.5 Разработка моделей поведения системы**

Поведение системы - это процесс изменения состояний объектов системы и их переходов из одного состояния (деятельности) в другое. Диаграммы поведения позволяют описать алгоритмические и логические особенности реализации выполняемых системой вариантов использования, то есть определить алгоритм функционирования системы.

**2.5.1 Диаграмма активности системы**

Диаграмма активности (activity) позволяет моделировать жизненный цикл объекта в виде переходов из одного состояния (деятельности) в другое, т.е. отображают алгоритмы по преобразованию классов.

Диаграмма активности позволяет разделить функции по исполнителям с использованием разделов диаграмм. Разделы группируют действия относительно какой-либо характеристики.

Диаграмма активности ИС футбольного клуба представлена на Рисунке 11.

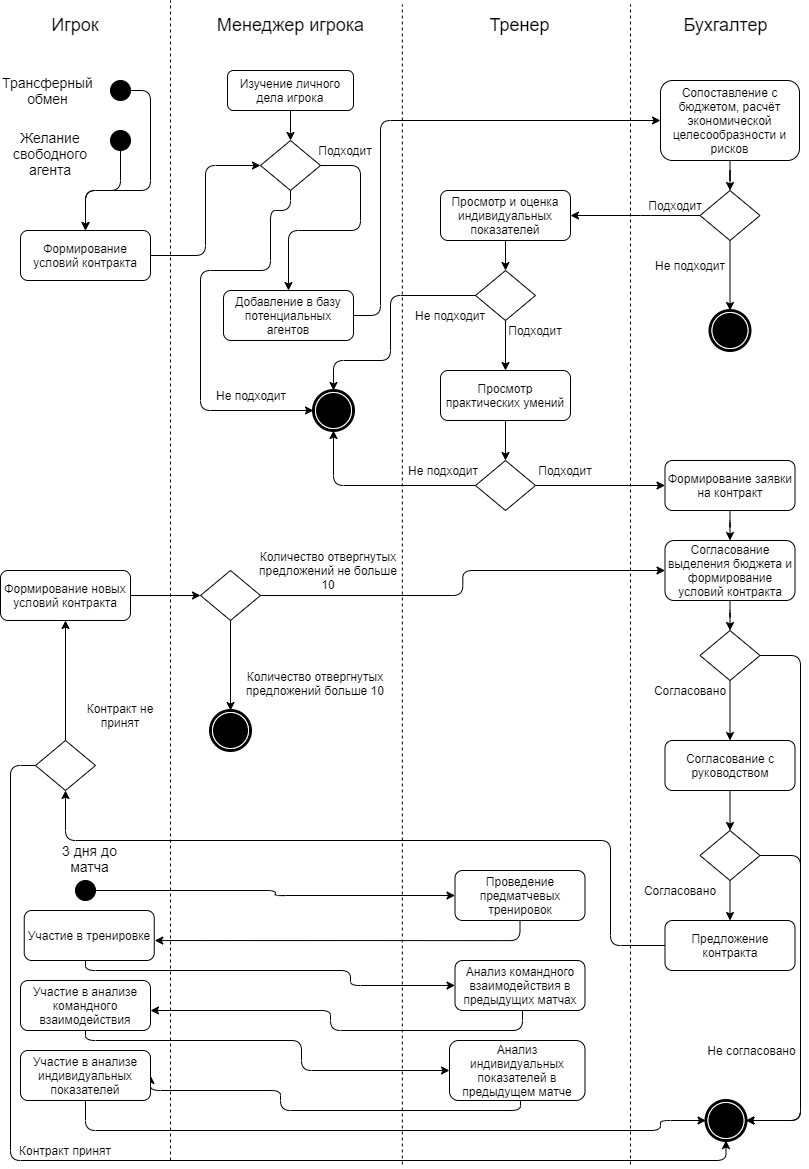


Рисунок 11 – Диаграмма активности

**4 Оценка экономических показателей проекта**

Необходимо расчитать и оценить экономические показатели и эффективность внедрения информационной системы в бизнес-процессы и рабочие процессы футбольного клуба. Для этого необходимо проанализировать план выполнения проекта.

**4.1 Планирование выполнения проекта**

Задачи, которые необходимо выполнить в ходе внедрения информационной системы, дата начала, продолжительность выполнения и дата завершения представлены в Таблице 1.

Таблица 1. – Задачи и ход выполнения проекта

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Задача** | **Дата начала** | | **Продолжительность (дни)** | | **Дата завершения** | |
| Анализ поставленной задачи, постановка задачи к проектируемой ИС | 8.02.2022 | | 7 | | 15.02.2021 | |
| Формирование требований к проектируемой ИС | 15.02.2022 | | 3 | | 18.02.2022 | |
| Выделение функций системы на основе разработки бизнес-модели (нотация IDEF0) | 18.02.2022 | | 3 | | 21.02.2022 | |
| Анализ информационных потоков ИС (на основе разработки модели потоков данных DFD) | 21.02.2022 | | 3 | | 24.02.2022 | |
| Моделирование данных (на основе разработки  ER- модели) | 24.02.2022 | | 1 | | 25.02.2022 | |
| Моделирование поведения системы на основе разработки моделей UML (диаграмма вариантов использования, активности, последовательности и состояния системы) | | 25.02.2022 | | 5 | | 01.03.2022 | |
| Моделирование структуры системы на основе разработки моделей UML (диаграмма классов, компонентов, размещения компонентов системы) | | 01.03.2022 | | 3 | | 04.03.2022 | |
| Разработка программной модели ИС | | 04.03.2022 | | 4 | | 08.03.2022 | |
| Разработка исходных модулей приложения | | 08.03.2022 | | 25 | | 02.04.2022 | |
| Тестирование и откладка ИС | | 02.04.2022 | | 15 | | 17.04.2022 | |
| Оценка эффективности проекта | | 17.04.2022 | | 3 | | 20.04.2022 | |

Исходя из задач и их продолжительности построим диаграмму Ганта для наглядного представления хода реализации проекта и внедрения информационной системы. Диаграмма Ганта - это популярный тип столбчатых диаграмм, а также гистограмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Он является одним из методов планирования проектов, используемый в различных приложениях по управлению проектами, например, в таких как YouTrack от компании JetBrains и Azure Devops Server (ранее называлась Team Foundation Server) от компании Microsoft.

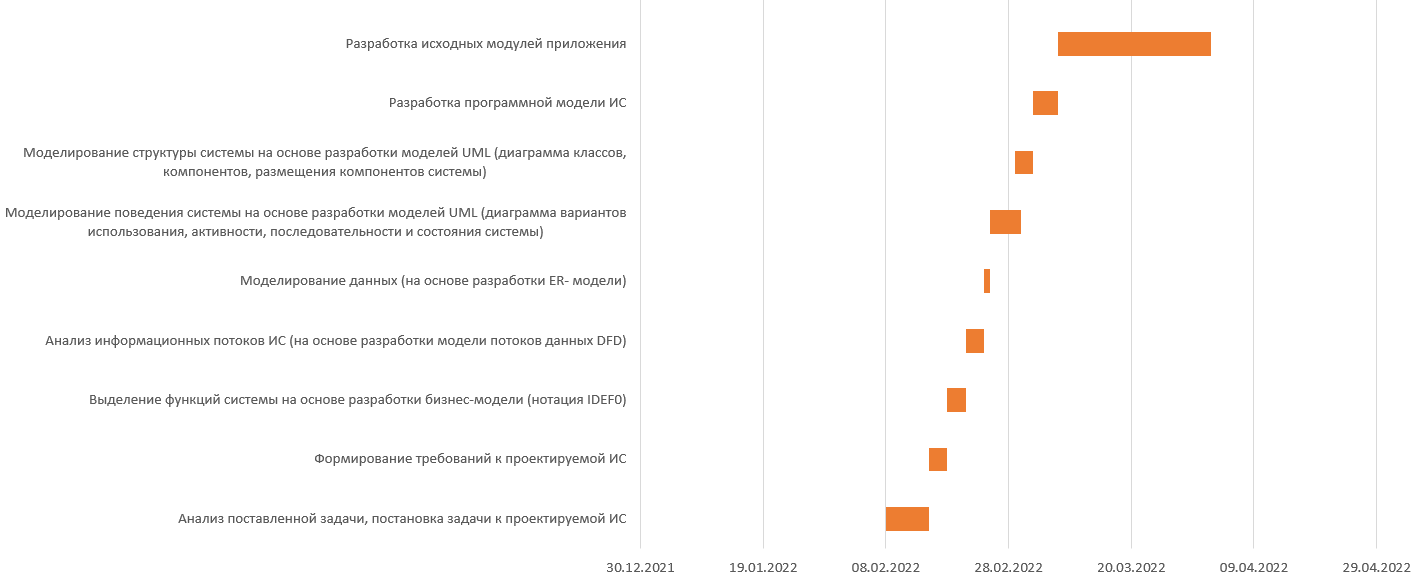


Рисунок N – Диаграмма Ганта

Рассчитаем текущие затраты до внедрения проекта.

Пусть в настоящее время эксплуатационными затратами является заработная плата 5 работников футбольного клуба.

Оклад составляет 35000 руб.

Премиальная оплата труда (30%) – 10500 руб.

Итого месячный фонд оплаты труда составляет 45500 руб.

Годовой ФОТ составляет 45500 \*12 = 546000 руб.

Для пятерых работников – 2730000 руб.

Единый социальный налог (30%) – 163800 руб. в год.

Для пятерых работников – 819000 руб.

Таким образом, годовые текущие затраты до внедрения проекта составляют:

2730000 + 819000 = 3549000 руб.

Рассчитаем текущие затраты после внедрения проекта.

После реализации проекта появится возможность сократить трёх сотрудников, но к текущим затратам также будут относиться затраты на электроэнергию всей системы.

Затраты на электроэнергию за год 7260 х 6,3 х 2,22 = 101538,36 руб. (7260 часов работы компьютеров, ИБП, охранных контроллеров и датчиков при общей мощности 6,3 кВт ч и стоимости 2,22 руб./кВт).

Таким образом, годовые текущие затраты после внедрения проекта составляют:

3549000 – 2129400 + 101538,36 = 1521138,36 руб.

Экономический эффект за год будет равен разнице затрат до и после внедрения проекта:

Э = 3549000 – 1521138,36 = 2027861,64

Рассчитаем экономические показатели.

Чистый доход от внедрения проекта рассчитывается по формуле (1):

ЧД = (1)

где T – горизонт расчета;

Rt – результаты (валовой доход или экономия), достигаемые на t шаге (рекомендуется величина расчетного шага t = 1 год);

Зt – суммарные инвестиционные и эксплуатационные затраты (с учетом налоговых выплат);

t – шаг расчета.

Горизонт расчета принимаем за 5 лет. Итого:

ЧД = (2027861,64 – 1521138,36) + 2027861,64 + 2027861,64 + 2027861,64 + 2027861,64 = 8618169,84 руб.

Рассчитаем срок окупаемости проекта. Срок окупаемости проекта Tок, год, найдем по формуле (2):

Ток = З / ЭN (2)

где N - максимальное количество лет, прошедших с начала эксплуатации программного продукта, в течение которых величина дохода от его использования не превысила величины капиталовложения при внедрении программного продукта;

Эn- величины приведенных (дисконтированных) годовых эффектов за j-й год, руб., прошедший с начала эксплуатации программного продукта, вычисленные по формуле при подстановке нормы дисконта E

Ток = 1521138,36 / 2027861,64 ≈ 3 квартала (0,75 года)

Таким образом, внедрение проекта будет эффективным, так как показатели эффективности имеют положительное значение, а срок окупаемости меньше двух лет, что говорит о его приемлемости.