

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Брянский государственный технический университет»

*На правах рукописи*

ВАЙНИЛОВИЧ ЮЛИЯ ВИКТОРОВНА

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ  
ОРГАНИЗАЦИОННЫМИ ПРОЦЕССАМИ IT-ПРОЕКТОВ НА ОСНОВЕ  
ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ

Научная специальность 05.13.10 – Управление в социальных  
и экономических системах

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель  
Захарова Алёна Александровна,  
доктор технических наук

Брянск – 2021

## ОГЛАВЛЕНИЕ

|                                                                                                                    |     |
|--------------------------------------------------------------------------------------------------------------------|-----|
| ВВЕДЕНИЕ.....                                                                                                      | 4   |
| ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ<br>УПРАВЛЕНИЯ ОРГАНИЗАЦИОННЫМИ СИСТЕМАМИ<br>ИТ-ПРОЕКТОВ.....         | 11  |
| 1.1 Процессы управления ИТ-проектом как организационной<br>системой.....                                           | 11  |
| 1.2 Обзор существующих методов.....                                                                                | 24  |
| 1.3 Обзор существующих средств.....                                                                                | 34  |
| 1.4 Постановка задачи .....                                                                                        | 36  |
| ГЛАВА 2. МЕТОДИЧЕСКИЕ ОСНОВЫ ПОВЫШЕНИЯ<br>ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ ОРГАНИЗАЦИОННЫМИ<br>СИСТЕМАМИ ИТ-ПРОЕКТОВ ..... | 45  |
| 2.1 Концепция повышения эффективности управления<br>организационными системами ИТ-проектов.....                    | 45  |
| 2.2 Метод повышения эффективности управления организационными<br>системами ИТ-проектов.....                        | 50  |
| 2.3 Алгоритм синтеза структуры и состава команд исполнителей .....                                                 | 61  |
| 2.4 Алгоритм назначения исполнителей на задачи проекта.....                                                        | 64  |
| 2.5 Методика многоуровневой оценки качества решения проектных<br>задач.....                                        | 68  |
| ГЛАВА 3. ПРОГРАММНЫЙ КОМПЛЕКС УПРАВЛЕНИЯ<br>ИТ-ПРОЕКТАМИ.....                                                      | 75  |
| 3.1 Обоснование выбора средств реализации программного комплекса..                                                 | 75  |
| 3.2 Архитектура программного комплекса .....                                                                       | 76  |
| 3.3 Проектирование основных компонентов программного комплекса...                                                  | 79  |
| 3.4 Технология управления ИТ-проектами с использованием<br>разработанного программного комплекса .....             | 85  |
| ГЛАВА 4. ТЕСТИРОВАНИЕ, АПРОБАЦИЯ И ВНЕДРЕНИЕ .....                                                                 | 89  |
| 4.1 Тестирование метода на учебном проекте .....                                                                   | 89  |
| 4.2 Тестирование метода при создании коммерческого программного<br>продукта .....                                  | 98  |
| 4.3 Тестирование программного обеспечения .....                                                                    | 106 |
| 4.4 Экспериментальная проверка результатов исследований.....                                                       | 110 |
| 4.5 Внедрение результатов исследования .....                                                                       | 123 |
| 4.6 Перспективы применения и развития .....                                                                        | 125 |

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| ЗАКЛЮЧЕНИЕ .....                                                                     | 130 |
| СПИСОК ЛИТЕРАТУРЫ.....                                                               | 134 |
| ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВО О ГОСУДАРСТВЕННОЙ<br>РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ .....  | 147 |
| ПРИЛОЖЕНИЕ Б. АКТЫ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ<br>ДИССЕРТАЦИОННОЙ РАБОТЫ .....             | 148 |
| ПРИЛОЖЕНИЕ В. ФРАГМЕНТЫ ИСХОДНОГО КОДА<br>РАЗРАБОТАННОГО ПРОГРАММНОГО КОМПЛЕКСА..... | 150 |

## ВВЕДЕНИЕ

### **Актуальность темы диссертации.**

Одним из путей повышения эффективности управления организационными системами учреждений образования и предприятий является реализация и внедрение ИТ-проектов. Успешная реализация ИТ-проектов зависит от эффективной организации процессов разработки программного обеспечения.

Рассматриваемые в данной работе организационные аспекты управления ИТ-проектами обеспечивают повышение эффективности работы участников команд исполнителей, что позволяет снизить трудоемкость решения проектных задач. Повышение эффективности управления предполагает снижение трудоемкости выполнения проекта за счет комплексного учета личностных и психологических качеств участников команд исполнителей, уровня и опыта владения технологиями и инструментами, опыта участия в аналогичных проектах, сработанности команды.

В соответствии с современными методологиями разработки программного обеспечения, в ИТ-проектах определяющее значение имеют люди и взаимоотношения. Поэтому ИТ-проект можно рассматривать как организационную систему, а задачу повышения эффективности управления ИТ-проектом рассматривать как задачу управления организационной системой.

Специфической особенностью управления ИТ-проектами в настоящее время является частое изменение условий внешней среды. Изменение состава и структуры задач ИТ-проекта, обусловленное изменением требований заказчика, приводит к необходимости оперативного принятия решений по изменению состава и структуры команд исполнителей, перераспределения исполнителей на задачи. При удаленном управлении ИТ-проектами с использованием современных информационно-коммуникационных технологий распределение задач ИТ-проекта между исполнителями и

организация контроля качества решения задач приобретает особую актуальность.

Современные методологии разработки программного обеспечения позволяют организовать работу над проектом при постоянном изменении условий внешней среды. При этом в процессе планирования и организации работ руководители IT-проектов сталкиваются со сложностями **комплексной** обработки информации для оперативного принятия решений по изменению параметров организационных процессов. Соответственно, создание математического и программного обеспечения для повышения эффективности управления организационными процессами IT-проектов является актуальной задачей.

Существующие подходы к управлению проектами позволяют выполнять определение основных этапов реализации проекта, разбиение каждого этапа на работы, определение последовательности работ. При этом в существующих универсальных системах управления процессами реализации проектов обработка информации для поддержки принятия решений по формированию состава задач, декомпозиции проектных задач на подзадачи, формированию состава и структуры команд, назначению исполнителей на задачи, оценка трудоемкости и продолжительности выполнения работ осуществляется руководителем проекта без учета всех специфических особенностей IT-проектов.

Существующие подходы рассматривают только **отдельные аспекты** задачи повышения эффективности управления IT-проектами: оценку личностных качеств разработчиков; формирование команды исполнителей; оценку трудоемкости проекта; распределение ресурсов проекта. Анализ существующих подходов показал отсутствие **комплексных подходов** к повышению эффективности управления организационными процессами IT-проектов, обеспечивающих комплексную обработку информации и поддержку принятия управленческих решений на следующих этапах проекта:

при отборе участников проектов, формировании команд исполнителей, планировании и контроле качества проектных работ.

Для исследования организационных процессов в IT-проектах предлагается использовать эволюционное моделирование, так как данный математический аппарат обеспечивает возможность решения **поисковых** задач, которые являются многомерными с большим объемом пространства поиска. Критерием эффективности управления организационными процессами IT-проектов является трудоемкость выполнения проекта, для расчета которой использована методика COSOMO II. Данная методика адаптирована к современным методологиям разработки ПО и учитывает большее число факторов, влияющих на выполнение IT-проекта, чем альтернативные методики оценки трудозатрат. Для повышения точности оценки поправочных коэффициентов методики COSOMO II использован метод парных сравнений Т. Саати.

**Объектом исследования** в диссертационной работе являются процессы управления организационными системами IT-проектов.

**Предметом исследования** являются математические методы и программные средства повышения эффективности управления организационными процессами IT-проектов.

**Целью диссертационной работы** является повышения эффективности управления организационными процессами IT-проектов за счет поддержки принятия решений при формировании команд исполнителей и распределения исполнителей на задачи на основе эволюционного моделирования.

Для достижения цели исследования необходимо решить следующие **задачи**:

1. Провести анализ существующих подходов к управлению организационными процессами IT-проектов и программных систем, предназначенных для управления проектами.

2. Разработать концепцию решения задачи повышения эффективности управления организационными процессами IT-проектов, реализующую

комплексную обработку информации о личностных и психологических качествах и профессиональных компетенциях участников команд исполнителей; комплексную поддержку принятия решений при управлении синтезом состава и структуры проектных задач и распределении участников проекта на задачи с использованием средств эволюционного моделирования.

3. Разработать метод повышения эффективности управления организационными процессами IT-проектов, основанный на использовании кластерного анализа для обработки информации об участниках проектов, интеграции результатов комплексной обработки информации о личностных и психологических качествах и профессиональных компетенциях участников команд исполнителей в операции эволюционного моделирования.

4. Разработать алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта путем применения эволюционного моделирования в сочетании с методикой COSOMO II, метода парных сравнений Т. Саати для определения поправочных коэффициентов методики COSOMO II при оценке трудоемкости проекта.

5. Разработать программный комплекс многоуровневого управления IT-проектами, реализующий концепцию, авторский метод и авторские алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта.

6. Разработать технологию повышения эффективности управления организационными процессами IT-проектов с применением предложенного метода и разработанного программного комплекса.

**Методы и средства исследования.** Для решения поставленных задач использованы положения теории организационных систем, методология управления проектами, методы кластерного анализа, методы принятия решений, методы эволюционного моделирования.

### **Научная новизна работы.**

1. Разработана концепция решения задачи повышения эффективности управления организационными процессами IT-проектов. Предложенная концепция **отличается** применением комплексного подхода к решению задачи повышения эффективности управления организационными процессами IT-проектов, обеспечивающего комплексную обработку информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей, комплексную поддержку принятия решений при управлении синтезом состава и структуры проектных задач и распределении участников проекта на задачи на основе эволюционного моделирования.

2. Предложен метод повышения эффективности управления организационными процессами IT-проектов, **отличающийся** интеграцией результатов комплексной обработки информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей в операции эволюционного моделирования для поддержки принятия решений при формировании состава команд исполнителей и назначения участников проекта на задачи.

3. Разработаны алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта путем адаптации средств эволюционного моделирования к решению задачи планирования выполнения работ в IT-проекте. Предложенные алгоритмы **отличаются** применением эволюционного моделирования в сочетании с методикой COSOMO II для расчета трудоемкости проектов и метода парных сравнений Т. Саати для оценки поправочных коэффициентов методики COSOMO II.

4. Разработана структура и состав программного комплекса, технология **многоуровневого** управления IT-проектами, обеспечивающая автоматизацию формирования иерархии задач, синтеза структуры и состава команд исполнителей и многоуровневый контроль качества решения задач.

**Теоретическая значимость** диссертации заключается в адаптации средств эволюционного моделирования к решению задачи повышения эффективности управления организационными процессами IT-проектов.

**Практическая значимость** полученных результатов состоит в следующем.

1. Использование разработанной концепции и метода для поддержки принятия управленческих решений в организационных системах IT-проектов.

2. Использование разработанных алгоритмов поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта для автоматизации организационных процессов на этапах планирования и организации работ при реализации IT-проектов.

3. Применение разработанного программного комплекса и технологии для принятия управленческих решений при управлении организационными процессами IT-проектов в организационных системах учреждений образования и предприятий.

4. Результаты работы внедрены в учебный процесс Белорусско-Российского университета при подготовке бакалавров по направлениям «Программная инженерия» и «Информатика и вычислительная техника».

5. Результаты работы внедрены в ООО «Стэпл Инк» для управления IT-проектами.

**Положения, выносимые на защиту:**

1. Концепция решения задачи повышения эффективности управления организационными процессами IT-проектов позволяет проводить комплексную обработку информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей, комплексную поддержку принятия решений при управлении синтезом состава и структуры проектных задач и распределении участников проекта на задачи на основе эволюционного моделирования.

2. Метод повышения эффективности управления организационными процессами IT-проектов обеспечивает математическую основу решения задачи повышения эффективности управления организационными процессами IT-проектов путем интеграции результатов комплексной обработки информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей в операции эволюционного моделирования для поддержки принятия решений при формировании состава команд исполнителей и назначения участников проекта на задачи.

3. Алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта на основе эволюционного моделирования в сочетании с методикой COSOMO II для расчета трудоемкости проектов и метода парных сравнений Т. Саати для определения поправочных коэффициентов методики COSOMO II обеспечивают автоматизацию организационных процессов на этапах планирования и организации работ при реализации IT-проектов.

4. Программный комплекс и технология **многоуровневого** управления IT-проектами обеспечивает реализацию алгоритмов поддержки принятия решений при формировании структуры и состава команд исполнителей, распределении исполнителей на задачи проекта.

**Личный вклад соискателя.** Лично автором разработаны концепция, метод, алгоритмы, выносимые на защиту, программное обеспечение и технология, реализующая предложенный метод и алгоритмы.

# ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ УПРАВЛЕНИЯ ОРГАНИЗАЦИОННЫМИ СИСТЕМАМИ ИТ-ПРОЕКТОВ

## 1.1 Процессы управления ИТ-проектом как организационной системой

### *Общее описание задачи реализации ИТ-проектов*

Жизненный цикл ИТ-проекта начинается при появлении идеи разработки программного обеспечения (ПО) и принятия решения о его реализации и заканчивается после полного вывода ПО из эксплуатации [34]. Существует несколько моделей жизненного цикла разработки ПО (каскадная или водопадная, V-модель, инкрементная, модель быстрой разработки приложений или RAD-модель, гибкая методология разработки Agile, итерационная модель, спиральная модель и др.) [18]. Модель жизненного цикла ПО определяет основные этапы разработки и их последовательность.

Комплекс задач управления организационными системами ИТ-проектов, соответствующий общей теории управления организационными системами [84, 86], представлен на рисунке 1.1.

Мониторинг и анализ процессов, которые необходимо автоматизировать, определяют исходное состояние системы, эффективность которой предполагается повысить в процессе реализации ИТ-проекта. Сравнение текущего состояния системы с состоянием, которое должно быть после автоматизации, позволяет заказчику или владельцу продукта определить эффективность функционирования существующей системы.

Прогноз результатов автоматизации процессов системы позволяет судить о динамике поведения системы в процессе реализации ИТ-проекта, и как система будет приближаться к представлению заказчика или владельца продукта по ходу автоматизации процессов в системе.

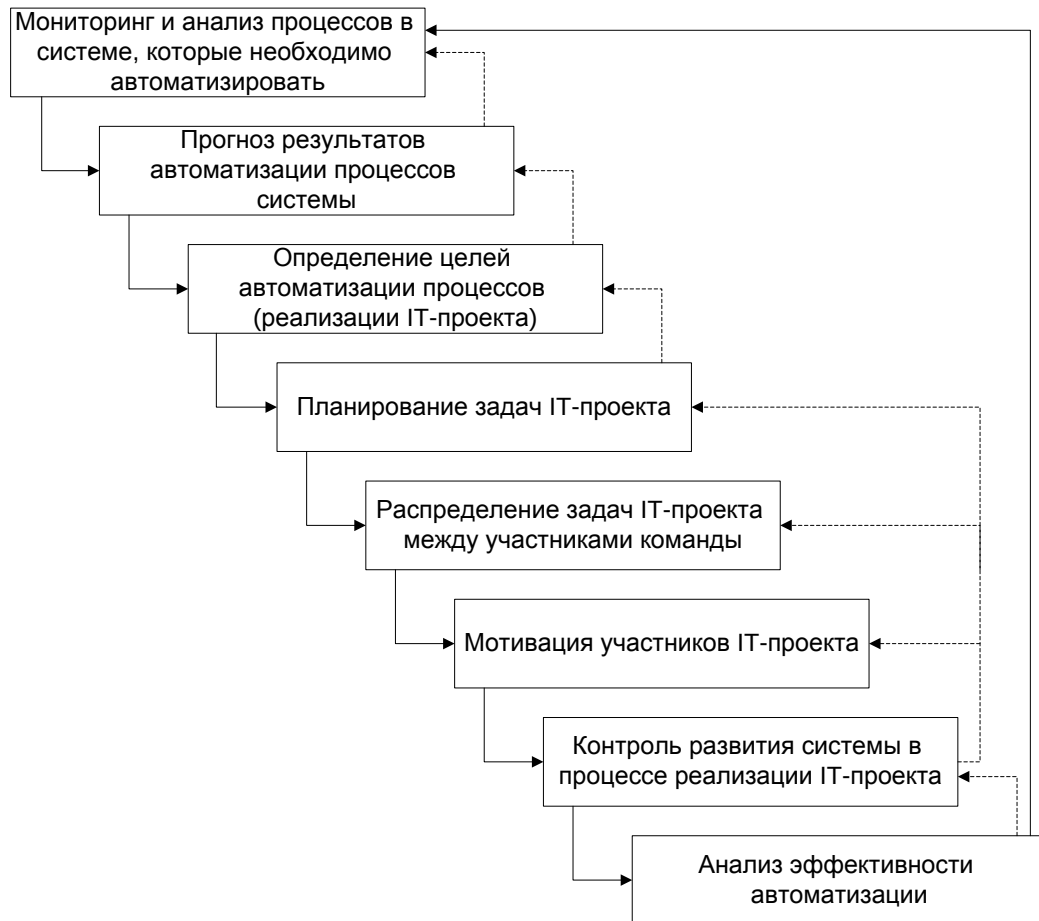


Рисунок 1.1 – Комплекс задач управления организационными системами IT-проектов

Независимо от выбранной модели жизненного цикла ПО, до начала реализации IT-проекта у заказчика (или владельца продукта) должно быть видение основных процессов «как есть» и «как должно быть» после внедрения ПО. В большинстве IT-проектов требования заказчика могут изменяться и уточняться на протяжении всего жизненного цикла. Для таких IT-проектов наиболее эффективно применение гибкой методологии разработки ПО Agile [67], обеспечивающей возможность еженедельного изменения плана работ и приоритетов задач в зависимости от изменения требований заказчика.

Целеполагание при автоматизации процессов системы (реализации IT-проекта) подразумевает формулировку общих целей развития системы в процессе автоматизации, определение критериев эффективности,

отражающих соответствие состояние системы до и после реализации IT-проекта целям развития системы.

На этапе планирования решения задач IT-проекта определяются приоритеты и последовательность решения задач IT-проекта для достижения или максимального приближения к целям автоматизации системы.

Распределение задач IT-проекта между участниками команды предполагает формирование состава и структуры команды исполнителей с учетом предыдущего опыта решения проектных задач, знаний, умений, навыков, личностных и психологических качеств участников команд исполнителей и назначение исполнителей на каждую задачу IT-проекта. Мотивация участников IT-проекта осуществляется путем учета оценки уровня профессиональных компетенций, что позволяет в дальнейшем участвовать в более значимых проектах

Контроль развития системы в процессе реализации IT-проекта предполагает оценку критериев эффективности состояния системы, связанных с реализацией и внедрением задач IT-проекта, отклонения критериев эффективности состояния системы от плановых. По мере поступления информации о влиянии результатов решения задач IT-проекта на показатели эффективности работы системы может потребоваться внесение в состав и структуру задач IT-проекта.

По мере реализации каждого этапа решения задач IT-проекта для успешной реализации следующих этапов необходим анализ эффективности автоматизации. Результаты анализа необходимы для принятия стратегических и тактических решений по автоматизации системы.

### ***Управление IT-проектом как организационной системой***

Рассмотрим постановку задачи управления IT-проектом как социально-экономической системой в соответствии с общей теорией управления организационными системами [84].

В соответствии с общей теорией управления организационными системами [84], структура управленческой деятельности включает управляемую систему и управляющий орган. Управляемой организационными системой является IT-проект (рисунок 1.2).

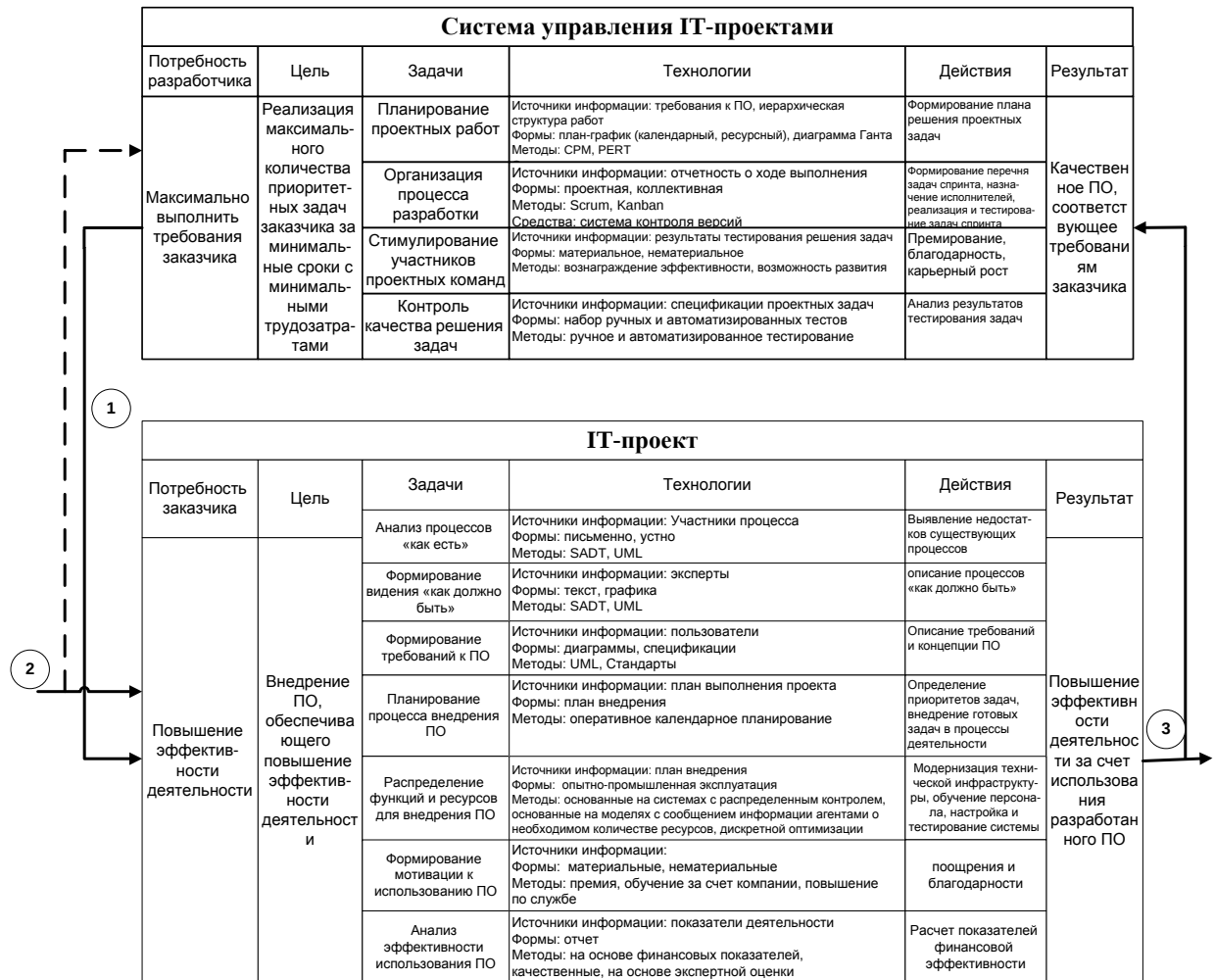


Рисунок 1.2 – Структура управленческой деятельности при реализации IT-проекта

Управляющим органом является система управления IT-проектами, цель которой – реализация максимального количества приоритетных задач заказчика в минимальные сроки с минимальными трудозатратами, что обеспечивает повышение эффективности управления организационными системами IT-проектов.

Управляемым объектом является организационная система IT-проектов. Состояние управляемой системы зависит от потребностей пользователей и от существующей организации процессов на этапе анализа процессов «как есть»; от возможных путей повышения эффективности процессов на стадии формирования видения «как должно быть»; от пожеланий заказчика, возможностей и потребностей разработчика на стадии формирования требований к программному обеспечению (ПО); от готовности пользователей к внедрению разрабатываемого ПО на стадии планирования процесса внедрения; от возможностей заказчика по модернизации технических средств, от квалификации персонала, от качества разработанного программного обеспечения на этапе распределения функций и ресурсов для внедрения ПО; от заинтересованности пользователей и заказчика в повышении эффективности деятельности на этапе формирования мотивации к использованию ПО; от степени соответствия результатов внедрения ПО ожиданиям заказчика на этапе анализа эффективности использования ПО.

С точки зрения руководителя проекта, состояние управляемой системы на этапе формирования требований к ПО зависит от полноты, непротиворечивости требований заказчика; на этапе планирования процесса внедрения – от структуры и состава команды исполнителей, структуры и состава задач проекта, распределения участников проектов на задачи; на этапе распределения функций и ресурсов для внедрения ПО – от многоуровневой оценки качества разработанного ПО.

Со стороны управляемой системы IT-проекта, состояние зависит от действий руководителя проекта, который на этапе формирования требований к ПО определяет структуру команды исполнителей; от действий участников команд исполнителей, которые на этапе планирования процесса внедрения выполняют выбор задач в спринт, определение времени решения каждой задачи, подбор исполнителей для каждой задачи, выбор способа реализации задачи. На этапе распределения функций и ресурсов для внедрения ПО

участники команд исполнителей выполняют контроль качества решения проектных задач, устранение ошибок и недоработок.

Задача управляющего органа состоит в планировании проектных работ, организации процесса разработки, стимулировании участников команд исполнителей, контроле качества решения задач, чтобы с учетом потребностей пользователей, пожеланий заказчика, возможностей и потребностей разработчика, качества разработанного программного обеспечения, степени соответствия результатов внедрения ПО ожиданиям заказчика обеспечить решение максимального количества задач заказчика за минимальные сроки с минимальными трудозатратами в зависимости от состояния управляемой системы (см. стрелку (3) на рисунке 1.2).

Воздействия внешней среды подразделяются на три группы:

- целенаправленные (см. стрелку (1) на рисунке 1.2): возможности и потребности разработчиков, качество разработанного программного обеспечения, степень соответствия результатов внедрения ПО ожиданиям заказчика, структуры и состава команды исполнителей, структуры и состава задач проекта, распределения участников проектов на задачи, оценки качества разработанного ПО;

- недетерминированные (см. стрелку (2) на рисунке 1.2): потребности пользователей, пожелания заказчика, заинтересованность пользователей и заказчика в повышении эффективности деятельности, полнота, непротиворечивость требований заказчика, выбор задач в спринт, определение времени решения каждой задачи, подбор исполнителей для каждой задачи, выбор способа реализации задачи;

- неконтролируемые управляющим органом (см. стрелку (2) на рисунке 1.2): существующая организация процессов в автоматизируемой системе, готовность пользователей к внедрению разрабатываемого ПО, возможности заказчика по модернизации технических средств, квалификация персонала заказчика.

Наиболее перспективной и распространенной методологией управления организационными системами IT-проектов, ориентированной на использование итерационного подхода к разработке проектов, динамического формирования и изменения требований, обеспечения их реализации в настоящее время является Agile. Методология Agile предполагает использование при реализации IT-проектов следующих основных **принципов** [62]:

*Принцип 1. Люди и взаимодействие важнее процессов и инструментов.* Существующая структура организации, для которой выполняется разработка ПО, в ряде случаев не обеспечивает эффективного функционирования и развития организации, не позволяет повышать эффективность процессов за счет реализации IT-проектов. Существующие регламентированные процедуры препятствуют внедрению новых программных продуктов и современных информационных технологий, организационные проблемы мешают эффективной работе команд при реализации IT-проектов. В соответствии с методологией Agile, потребности взаимодействия при реализации IT-проектов первичны, процессы и инструменты должны способствовать эффективной реализации и внедрению, а не мешать. Команды организационных систем IT-проектов должны иметь возможность изменять некоторые процессы, но обязаны им следовать.

В данном принципе методологии Agile не учитываются личностные и психологические качества участников организационных систем IT-проектов. При этом эффективность взаимодействия участников команд исполнителей во многом зависит от личностных и психологических качеств, обеспечивая повышение эффективности реализации IT-проектов за счет улучшения психологического климата в команде. Предлагается учитывать результаты психологического тестирования при формировании команд IT-проектов.

*Принцип 2. Сотрудничество с заказчиком важнее согласования условий контракта.*

IT-проект должен обеспечивать повышение эффективности работы клиента или заказчика. В методологии Agile основное внимание уделяется сотрудничеству с клиентом или заказчиком. Условия контракта должны обеспечивать эффективное взаимодействие с заказчиком, а не мешать этому процессу. Соответственно, разработчики должны быть готовы к изменению условий контракта при изменении потребностей клиентов или заказчика.

*Принцип 3. Готовность к изменениям важнее следования первоначальному плану.*

Гибкие подходы к реализации IT-проектов создавались для работы в условиях неопределенности и частых изменений требований заказчиков. Долгосрочное планирование, когда работы по реализации IT-проекта, ресурсы и задачи определяются на весь период разработки, не эффективно в условиях постоянного изменения условий контракта, потребностей клиентов или заказчика. В соответствии с методологией Agile, при изменении потребностей клиентов или заказчика должен изменяться состав и структура задач в IT-проекте. Горизонт детального планирования задач составляет от двух до четырех недель.

Изменение условий контракта и готовность к изменениям в IT-проекте, в соответствии со вторым и третьим принципом, обуславливает необходимость оперативного изменения состава и структуры команд IT-проектов, состава и структуры проектных задач и перераспределения участников команд исполнителей на задачи.

Для решения перечисленных задач предложен комплексный подход к повышению эффективности управления организационными системами IT-проектов, обеспечивающий обработку информации и поддержку принятия решений по изменению состава команд исполнителей, состава и структуры задач проекта и распределении участников IT-проектов на задачи при изменении потребностей клиентов или заказчиков. Предлагается применять

эволюционное моделирование в сочетании с методикой СОСОМО II, методом парных сравнений Т. Саати при формировании команд исполнителей, синтезе состава и структуры проектных задач и распределении участников команд исполнителей на задачи. Применение предложенного подхода обеспечивает повышение эффективности управления организационными системами ИТ-проектов за счет оперативного изменения состава команд и плана работ при изменении требований заказчиков в соответствии с принципами методологии Agile.

*Принцип 4. Работающий продукт важнее исчерпывающей документации.*

В соответствии с методологией Agile, первостепенное значение имеет качество разработанного ПО. В процессе реализации ИТ-проекта создается только рабочая документация, необходимая для эффективной работы команд над проектом. Документация, которая используется для коммуникации (технические задания, обоснования, бюджеты и т.п.) и не несет ценности клиентам или заказчику формируется после создания качественного программного продукта.

Для повышения качества программных продуктов, которые создаются в рамках организационных систем ИТ-проектов на основе принципов методологии Agile, предложена методика многоуровневого контроля качества решения проектных задач. Повышение качества решения проектных задач обеспечивает получение работающего продукта в соответствии с четвертым принципом методологии Agile и улучшает сотрудничество с клиентами и заказчиками в соответствии со вторым принципом методологии Agile.

Эффективность управления организационными системами ИТ-проектов определяется внутренними и внешними **условиями**:

- *Нормативно-правовые* условия реализации ИТ-проекта определяются нормативно-правовой базой системы, автоматизация которой выполняется в процессе реализации ИТ-проекта. При реализации ИТ-проекта необходимо

учитывать существующие регламенты, законодательные и иные нормативно-правовые акты и процедуры.

– Выполнение *финансовых условий* предполагает формирование бюджета ИТ-проекта. Использование гибких методологий разработки ПО предполагает определение бюджета и сроков реализации ИТ-проекта до запуска процесса разработки. Помимо оплаты труда участников команд исполнителей, могут потребоваться средства на создание и/или модернизацию инфраструктуры, необходимой для внедрения и функционирования результатов реализации ИТ-проекта. Кроме того, финансовые средства могут потребоваться для оплаты труда сотрудников, участвующих в формировании требований и оценке качества решения задач ИТ-проекта со стороны заказчика или владельца продукта.

– *Материально-технические условия* включают компьютеры, сервера, телекоммуникационное оборудование, программное обеспечение, необходимое для эффективной работы ПО, созданного в процессе реализации ИТ-проекта.

– Объекты материально технической базы образуют инфраструктуру ИТ-проекта. *Инфраструктурные условия* определяют способы ввода, обработки, передачи и вывода результатов обработки информации пользователям разработанного ПО.

– *Квалификация кадров* ИТ-проектов (участников команды исполнителей) является основой успешной реализации проекта. В процессе реализации проекта изменение требований заказчика может приводить к изменению кадрового состава участников команд исполнителей. При этом может возникать дефицит специалистов по некоторым технологиям, а квалификация команды определяется эффективностью системы мотивации участников ИТ-проекта.

– В современных условиях доступ к *информационным ресурсам* может быть организован практически повсеместно. При этом в процессе реализации

IT-проекта актуальной является задача защиты информационных ресурсов от несанкционированного доступа.

– При реализации IT-проектов команда разработчиков может быть территориально распределена и взаимодействовать удаленно. В ряде случаев архитектура IT-проекта может требовать распределенного хранения и обработки информации. Пользователи создаваемой системы также могут быть территориально распределены. Удаленная работа разработчиков, распределенное хранение, обработка и выдача информации пользователям определяет *условия организации работ* при реализации и внедрении IT-проектов, в процессе функционирования автоматизируемых систем.

В процессе управления организационными системами IT-проектов принимают участие три основных **субъекта**, каждый из которых представляет собой организационную систему:

- организация-заказчик или владелец продукта;
- команда, занимающаяся реализацией IT-проекта;
- команда управления организационной системой IT-проекта.

На некоторых стадиях реализации IT-проекта в процессе управления могут участвовать инвесторы и привлекаться сторонние специалисты к решению задач проекта и управления проектом.

Заказчик или владелец продукта определяет требования к разрабатываемой системе, перечень и приоритеты задач IT-проекта, контролирует выполнение требований, оценивает качество решения проектных задач и достижение целей автоматизации в процессе реализации IT-проекта. Использование гибких методологий разработки ПО позволяет заказчику или владельцу продукта изменять требования, перечень и приоритеты задач с определенной периодичностью (раз в неделю, раз в две недели, раз в месяц) в зависимости от результатов решения проектных задач в предыдущие периоды и достижения целей автоматизации.

В случае невыполнения требований, низкого качества решения задач заказчик или владелец продукта может не оплатить работы команды исполнителей, потребовать возврата денежных средств, если оплата была произведена. В случае отсутствия ожидаемого эффекта от автоматизации ИТ-проект может быть остановлен до завершения всех работ либо могут быть изменены требования заказчика, перечень и приоритеты задач.

Противоречивые требования, частично конфликтующие интересы субъектов управления, необходимость постоянных изменений требований к ИТ-проекту и перечня задач с учетом основных принципов гибких методологий разработки ПО обуславливают необходимость постоянной оперативной обработки информации для поддержки принятия управленческих решений по формированию состава и структуры команд исполнителей, состава и структуры проектных задач, распределению участников проекта на задачи, оценке качества решения проектных задач.

Показатели эффективности управления организационной системой ИТ-проекта определяются затратами времени и трудоемкостью реализации проекта, а также показателями эффективности автоматизируемой системы.

В процессе управления организационной системой ИТ-проекта участвуют специалисты разных профилей, основные из которых, входящие в состав команды управления организационной системой ИТ-проекта, представлены в таблице 1.1.

Каждый специалист принимает управленческие решения в рамках своих задач и несет ответственность за эффективность управленческих решений в рамках своих функциональных обязанностей. При этом в процессе анализа эффективности использования разработанного ПО принимают участие все специалисты, входящие в состав команды управления организационной системой ИТ-проекта. Кроме того, при возникновении в проекте сложностей и снижении эффективности решения задач управления (существенное изменение требований к ПО, отклонения от плана внедрения ПО, снижение показателей эффективности использования ПО ниже критических значений и

т.п.) для выявления причин проблем к решению задач управления организационной системой ИТ-проекта могут привлекаться специалисты, ответственные за решение любых задач в проекте, и, при необходимости, специалисты со стороны.

Таблица 1.1 - Основные специалисты, входящие в состав команды управления организационной системы ИТ-проекта

| Специалист                            | Анализ процессов «как есть» | Формирование видения «как должно быть» | Формирование требований к ПО | Планирование процесса внедрения ПО | Распределение функций и ресурсов при внедрении ПО | Формирование мотивации к использованию ПО | Анализ эффективности использования ПО |
|---------------------------------------|-----------------------------|----------------------------------------|------------------------------|------------------------------------|---------------------------------------------------|-------------------------------------------|---------------------------------------|
| Заказчик или владелец продукта        | +                           | +                                      |                              | +                                  | +                                                 | +                                         | +                                     |
| Бизнес-аналитик                       | +                           | +                                      | +                            |                                    | +                                                 |                                           | +                                     |
| Руководитель проекта или SCRUM-мастер |                             |                                        | +                            | +                                  | +                                                 |                                           | +                                     |
| Команда проекта                       |                             |                                        | +                            | +                                  | +                                                 |                                           |                                       |
| Специалист по качеству                |                             |                                        | +                            |                                    |                                                   |                                           | +                                     |
| Аналитик данных                       |                             |                                        |                              |                                    |                                                   |                                           | +                                     |

Небольшим компаниям, занимающимся разработкой ПО, экономически не целесообразно содержать в штате всех специалистов, которые необходимы для управления проектами. Привлечение специалистов со стороны к реализации ИТ-проектов приводит к увеличению сроков реализации проекта из-за необходимости изучения привлекаемыми специалистами документации по проекту, выстраивания эффективного взаимодействия с другими участниками команды исполнителей,

«погружения» новых специалистов в существующие текущие процессы реализации IT-проекта.

**Объектом** управленческой деятельности являются организационные процессы IT-проектов.

**Предметом** управленческой деятельности являются методы и средства организации процессов управления IT-проектами.

**Формы управления.** Структура системы управления IT-проектами может различаться для разных IT-компаний, однако чаще всего используется иерархическое или сетевое управление. По числу субъектов – коллективное управление, т.к. в процессе управления принимает участие большое количество специалистов. Управление IT-проектами преимущественно персонифицировано (многие решения субъективны), что обуславливает значительное влияние человеческого фактора на принимаемые решения.

**Средства.** Повышение результативности и эффективности организационных процессов IT-проектов может быть достигнуто за счет использования систем управления проектами. При этом не все системы управления проектами одинаково эффективны. Они обладают как преимуществами, так и недостатками, которые стоит принимать во внимание при выборе системы для реализации проекта.

## **1.2 Обзор существующих методов**

### ***Общая методология управления проектами***

Теоретические основы управления сложными проектами, широко исследовали В. Н. Бурков[21,22,23,24], В. И. Воропаев[31, 32], В. М. Губко [42], Д. А. Новиков [81, 82], Б. П. Титаренко [100], Д. В. Шапиро [108].

В. Н. Бурков развил теорию активных систем. В. И. Воропаев развили методологию управления проектами. Новиков Д. А. разработал методологию управления социальными и экономическими системами, которую адаптировал к различным предметным областям, в частности, к управлению

проектами. Б. П. Титаренко создал робастную технологию, предназначенную для поддержки принятия проектных решений в условиях неопределенности.

За рубежом вопросам методологии управления проектами посвящали свои труды Э. Голдратт [39], Т. ДеМарко, Т. Листер [48], Г. Гант, Клиффорд Ф. Грей [38], Г. Файоль (Henri Fayol) [102], Дж. Келли, Дж. Мартин [123], Лич Л [77].

Э. Голдратт сформулировал и обосновал теорию ограничений систем. Т. ДеМарко разработал методологию структурного анализа. Г. Гант предложил новый способ календарного планирования с использованием диаграмм. Г. Файоль определил пять функций менеджмента при управлении проектами.

### ***Существующие методологии управления разработкой программного обеспечения***

Значительный вклад в развитие теории управления разработкой программным обеспечением внесли Новиков Д. А. [84,85,86,87], Б. Боэм [16, 17], Ф. Брукс [19], А. Коберн [60], В. В. Липаев [73, 74, 75], У. Ройс [93], Р. Томсетт [101], У. Хамфри, Э. Йордан [54,55], М. Фаулер [103], Л. Константайн [66], Дж. Рамбо [27], Г. Буч [26], Ф. Крачтен [68].

Б. Боэмом была создана первая версия концепции RAD (Rapid Application Development), ставшей основой для создания гибкой, адаптивной системы разработки приложений, которую назвал «спиральная модель». Ф. Брукс одним из первых выполнил анализ деятельности по управлению ИТ-проектами в своей работе «Мифический человеко-месяц». Он определил основные типы команд разработки ПО, выделил командные роли и оценил применимость различных команд к различным типам проектов. А. Коберн предложил структурированный подход к выбору адекватной проектной методологии, предложил набор методологий семейства Crystal и определил область применения этих методологий. Э. Йордон выделил условия применения методологий управления «критическими» проектами. М. Фаулер выполнил сравнительный анализ адаптивных проектных методологий и

сформулировал рекомендации по процедуре их выбора. Новиков Д. А. исследовал модели оперативного управления проектами, позволяющие решать задачи управления с учетом моментов принятия решений, их содержания и согласованности. В том числе: модели дополнительных соглашений, сокращения продолжительности проекта, шкал оплаты, распределенного финансирования, типовых решений, точек контроля.

Исследования в области теории управления разработкой ПО проводит институт Карнеги Мелона (Carnegie Mellon University Software Engineering Institute, CMU SEI), а также ведущие компании-вендоры: IBM, HP, Microsoft, SAP, Oracle.

В современной IT-сфере наиболее распространенной методологией управления проектами является методология PMI PMBOK5 Американского института управления проектами – PMI [128]. Положительной стороной методологии является процессный подход. Весь процесс управления разделен на пять этапов, которые, в свою очередь, разделены на 47 процессов. Недостатком данной методологии являются большие затраты на планирование и разработку проектной документации [90].

Стандарт PRINCE2 [129] разработан по заказу Британского агентства Central Computer and Telecommunications Agency (CCTA), дает пошаговый алгоритм управления проектом и сфокусирован на достижении тех бизнес-задач, ради которых он был принят. Методология PRINCE2 более подходит для IT-индустрии, чем PMI PMBOK5.

Семейство методологий Agile [62,67] предназначено для управления проектами в области разработки программного обеспечения, базируется на принципах противоположных классическому подходу PMBOK.

Наиболее известной методологией данного семейства в настоящее время является SCRUM. Преимуществом использования методологии SCRUM является отсутствие предпроектного периода, начать работу над проектом можно как только соберутся все участники проекта. В настоящее время методология SCRUM набирает все большую популярность, так как при

правильном подходе позволяет решать задачи за короткий период времени, внедрять по ходу выполнения проекта новые элементы, заранее не предусмотренные проектом. Недостатком методологии является возможное затягивание времени выполнения проектов из-за большого числа итераций.

В настоящее время в IT-индустрии используется комплексный подход к управлению организационными системами IT-проектов, заключающийся в использовании нескольких методологий на разных стадиях реализации проекта.

### ***Существующие методы управления решением проектных задач***

Разработке моделей и методов решения задач планирования работ при управлении проектами посвящены работы Антоновой А.С. [7], Г.М. Адельсона-Вельского, В. М. Аньшина [6], И. В. Бурковой [25], С. Д. Бушуева, Л. Г. Голуба, С. И. Зуховицкого, Л.М. Лаврецкого, В. И. Либерзона, В. Д. Мазурова, И. И. Мазура, С. П. Никанорова, В. В. Познякова, М.Л. Разу, Я. А. Рекитара, А. С.Товба, Д. Филлипса, Б.И. Хацет, А.В. Цветкова, Г.Л. Ципеса, И.А. Горгидзе, С. Е. Ловецкого, А.Ю. Заложнева [6, 29, 100, 108] и др.

В работах В. И. Аверченкова, В. В. Емельянова, В. В. Курейчика, В. М. Курейчика, К. С. Мышенкова, А. Н. Полетайкина, I. Okada, E. Osaba, E. Sriprasert [1, 51] рассмотрены вопросы применения эвристических методов эволюционного моделирования к задаче планирования.

Антонова А.С. и Буркова И. В развивают новые подходы к планированию бизнес-процессов и проектных работ. Эффективность управления проектами повышается за счет создания новых методов планирования.

Наиболее популярными методами планирования являются метод сетевого планирования, методы теории расписаний, метод PERT, эвристические методы оптимизации [30]: искусственные иммунные системы, разбросанный поиск, гармоничный поиск, гравитационный поиск, метод

перекрестной энтропии, метод сглаживания, метод «Калибровка», генетический алгоритм, алгоритм имитации отжига [122], муравьиные алгоритмы.

Анализ исследований показывает, что методами планирования занимались многие исследователи. Однако, в соответствии с ежегодным аналитическим отчетом специалистов компании The Standish Group International [127], планирование занимает четвертое место среди причин, по которым ИТ-проекты не были закончены или имели существенные недостатки, что указывает на необходимость дальнейшего совершенствования методов планирования проектов.

### ***Существующие методы управления процессом синтеза состава и структуры команд исполнителей***

Одним из факторов успешной реализации ИТ-проекта является эффективность работы команды исполнителей. Характеристики эффективных команд, эффективность работы в команде в разное время исследовали Новиков Д. А., Губко М. В. [43, 83], К. Левин [72], Д. МакГрегор [118], Р. Лайкерт, К. Арджирис, Р. Блейк, Дж. Моутон, Р. Валтон, Дж. Катценбах, Д. Смит, Дж. Хэкман, Р. Шварц, Р. М. Белбин [11], И. Адизес [4].

Курт Левин основное внимание уделял исследованию организационного поведения в группах и групповой динамике, развивал вопросы социальной психологии, технологии разрешения конфликтов, теории лидерства. Д. МакГрегор и его коллеги изучали совершенствования менеджеров в промышленности. Р. Лайкерт определил характеристики эффективных команд, сфокусировав внимание на процессе и внутренней динамике команды. К. Арджирис исследовал влияние вида рабочей ситуации на личностное развитие индивида в организации, внес вклад в теорию организационного поведения, теорию организационного обучения, теорию интервенции, создал методы исследований, направленные на повышение

организационных компетенций. Р. Блейк и Дж. Моутон в своих работах связали стили управления и эффективность команд. Ими была разработана схема – управленческая решетка, включающая пять основных стилей руководства. Р. Валтон выделил и изучил три типа конфликтов, которые могут возникнуть между членами команды. Дж. Катценбах и Д. Смит предложили модель развития команды от рабочей группы до высокоэффективной команды. Р. Шварц разработал собственную модель групповой эффективности, опираясь на показатели эффективности групповой деятельности, предложенные Э. Сандстромом и Дж. Хэкманом. Р. М. Белбин в результате исследований выделил девять типов ролей, которые может исполнять человек в команде в зависимости от личных качеств. Для выявления склонности участника к той или иной командной роли, а, следовательно, и для формирования успешных команд он предложил использовать разработанный им тест.

Проблемы формирования и функционирования команд исследуются в социологии, психологии, менеджменте. В социологии этой проблеме посвящены работы А. И. Пригожина, А. В. Жуткина, О. Е. Артемова, в психологии – В. И. Корниенко, В. В. Авдеева, Ю. В. Синягина, М. Ю. Губиева, В. В. Исаева, К. Фоппеля, Р. Чалдини, Т. Кеннета, в менеджменте – в работах Г.И. Дзюбенко, А. М. Седых, В. Г. Куликова, В. Н. Михеева, С. А. Никоновой, Ю. В. Омеляненко, Остроуха Е. Н., П. Драккера, Л. Томпсона[8, 20, 40, 88, 91].

А. И. Пригожин занимается развитием теории организации, является автором теории синергичной организации.

А. В. Жуткин в своей работе показал многоаспектность проблемы формирования эффективных команд, выделили характеристики и стадии развития эффективных команд. Недостатком исследования Жуткина А. В. является отсутствие практических методик формирования эффективных команд.

О. Е. Артемовым в результате исследования выявлены наиболее значимые проблемы формирования эффективных команд, показана с точки зрения соционики необходимость учитывать психотипы работников при подборе состава команд.

Томас Кеннет – известный американский специалист в области психологии конфликта. Т. Кеннет выделил пять основных типов поведения людей в конфликтных ситуациях и разработал методику определения способов реагирования человека на конфликтные ситуации в коллективе.

Остроух Е.Н. для оптимального распределения программистов-претендентов в команде в целях улучшения качества реализации IT – проектов предлагает подход, основанный на комбинации классической модели распределения ресурсов, алгоритма колонии пчел, концепции командных ролей Белбина Р.М. и учете профессиональных качеств.

Проблемы формирования и функционирования команд для выполнения IT-проектов исследует в своей работе А. В. Будыльский.

А. В. Будыльский разработал методику распределения задач проекта между ресурсами и принятия решения о замене ресурса на основе кооперационных коэволюционных генетических алгоритмов. Критериями эффективности разработанной методики являются минимизация бюджета и длительности выполнения проекта. Разработанная методика позволяет снизить неопределенность информации экспертного характера. Недостатком предложенной методики является отсутствие ограничений на владение технологиями и опыта работы ресурсов проекта.

Оригинальные результаты исследований математических моделей формирования и функционирования команд изложены в работах Д. А. Новикова, А. Г. Чхартисвили, В. Holmstrom, J. Marshak, R. Radner[83, 85, 87, 106].

Необходимость проведения профессиональной диагностики человека для объективной оценки деловых качеств человека, выработке рекомендаций по совершенствованию его индивидуального стиля профессиональной

деятельности освещается в работах Адакина Е.Е., Скрипниковой Г. В., Бодрова В. А., Пряжникова Н. С., Шадрикова В. Д. [3, 12, 92, 107], методы диагностики профессиональных и личностно-психологических качеств, обоснование их применения рассматриваются в работе Барташева А.В., Алексеевой И. Ю., Майоровой Е. В. [10].

Исходя из анализа исследований задачи синтеза состава и структуры команд исполнителей, можно сделать вывод, что существующие научные работы не охватывают в полной мере все аспекты формирования команд IT-проектов.

### ***Существующие методы оценки трудоемкости выполнения IT-проектов***

Успех IT-проекта базируется на оценках таких факторов как затраты, времени выполнения, прогнозируемой выгоде. При неверной оценке времени выполнения может произойти перерасход ресурсов, сокращение границ проекта, экономия на выполнении некоторых видов работ (например, тестировании), чтобы выполнить проект в установленные сроки. В результате проект может получиться низкого качества, завершиться с нарушением сроков, иметь меньшее количество функций, чем было заявлено.

Оценкой трудоемкости выполнения проектов занимались А. Albrecht, С. Jones, Т. DeMarco, В.В. Boehm, S.N. Parkinson, L.Putnam, N. Fenton, S.L. Pfleeger, M. Shepperd [113, 115, 116, 119, 124, 126].

А. Альбрехт предложил метод функциональных точек для оценки труда программистов. Недостатком метода является то, что сформулированные требования не отражают сложности реализации. Кейперс Джонс (Capers Jones) модифицировал предложенный метод, который стал учитывать внутреннюю реализацию.

Том ДеМарко предложил метод, основанный на использовании «бэнг-метрики» (Bang Metric), близкой, по своему содержанию, к функциональным точкам, но оценки корректируются с учетом хронологических данных по

выполненным ранее проектам. Такой подход позволяет получить приближенные значения реальных, а не абстрактных, затрат ресурсов и времени.

Б. Боем (Barry Boehm) предложил методику для оценки трудоемкости разработки программного обеспечения COSOMO, которая эволюционировала в COSOMO II. Математическая модель, являющаяся основой COSOMO II адаптирована к современным методологиям разработки программного обеспечения. При оценке трудоемкости и проекта на основе методики COSOMO II для обработки статистических данных используется байесовский анализ, который дает лучшие результаты для программных проектов, характеризующихся неполнотой и неопределенностью, в отличие от многофакторного регрессионного анализа, являющегося основой методики COSOMO II.

Б. Боем назвал основные категории методов оценки трудоемкости в зависимости от способа получения исходных данных: метод экспертных оценок, метод аналогий, динамические методы, байесовские сети, нейронные сети, исследовательские и эмпирические методы, методы функциональных точек, COSOMO, методы алгоритмического моделирования, оценки с целью выиграть контракт.

Анализ существующих подходов к оценке трудоемкости IT-проектов показал отсутствие метода, позволяющего комплексно учитывать все специфические особенности IT-проектов. Для получения наиболее достоверных результатов необходимо применять сочетание методов.

### ***Эволюционное моделирование для оптимизации процессов в сложных системах***

Развитием эволюционного моделирования занимались Д. Холланд, де Джонг, N. Packard, A. Perelson, A. Фрейзер, В.М. Курейчик, В.В. Курейчик, А.П. Карпенко, Гладков Л. А.

Д. Холландом [120] заложены основы современных генетических алгоритмов. Де Джонг впервые описал стандартный генетический алгоритм [117].

N. Packard и A. Perelson являются авторами нового направления в эволюционном моделировании – искусственные иммунные системы [121]. N. Packard и A. Perelson использовали принципы функционирования человеческой иммунной системы для решения задач распознавания образов и оптимизации.

В.М. Курейчик, В.В. Курейчик исследуют эволюционное развитие сложных систем различной природы, развивают методы и подходы распределенного искусственного интеллекта и искусственной жизни [69, 70, 71].

А.П. Карпенко исследует методы и подходы гибридных систем, мета-оптимизации, развивает направление глобальной оптимизации. В статье «Паттерны мета-эвристических популяционных алгоритмов глобальной оптимизации» формализуются структуры алгоритмов непрерывной глобальной оптимизации. В работе «Прогнозирование эффективных стратегий алгоритма оптимизации на основе анализа информационного содержания ландшафта целевой функции» описывается задача настройки параметров алгоритма оптимизации с использованием методов ландшафтного анализа. [2, 56, 57, 58, 59].

Гладков Л. А. развивает гибридные методы и подходы на основе сочетания различных методов эволюционного моделирования, методов нечеткого управления, многоагентных систем, методов вычислительного интеллекта к решению задач многокритериальной оптимизации [35, 36].

Результаты обзора существующих подходов к решению задачи повышения эффективности управления проектами представлены на рисунке 1.3.

Анализ литературных источников позволяет сделать вывод о недостаточной проработанности задачи управления организационными

процессами IT-проектов, об отсутствии комплексных подходов к решению задачи повышения эффективности управления организационными процессами формирования команд IT-проектов, распределения исполнителей на задачи.

Большинство научных работ посвящено исследованию отдельных аспектов задачи управления проектами, без учета специфики организационных систем IT-проектов.

Соответственно, актуальной является задача комплексного повышения эффективности управления организационными процессами формирования команд и распределения участников команд на задачи IT-проектов.

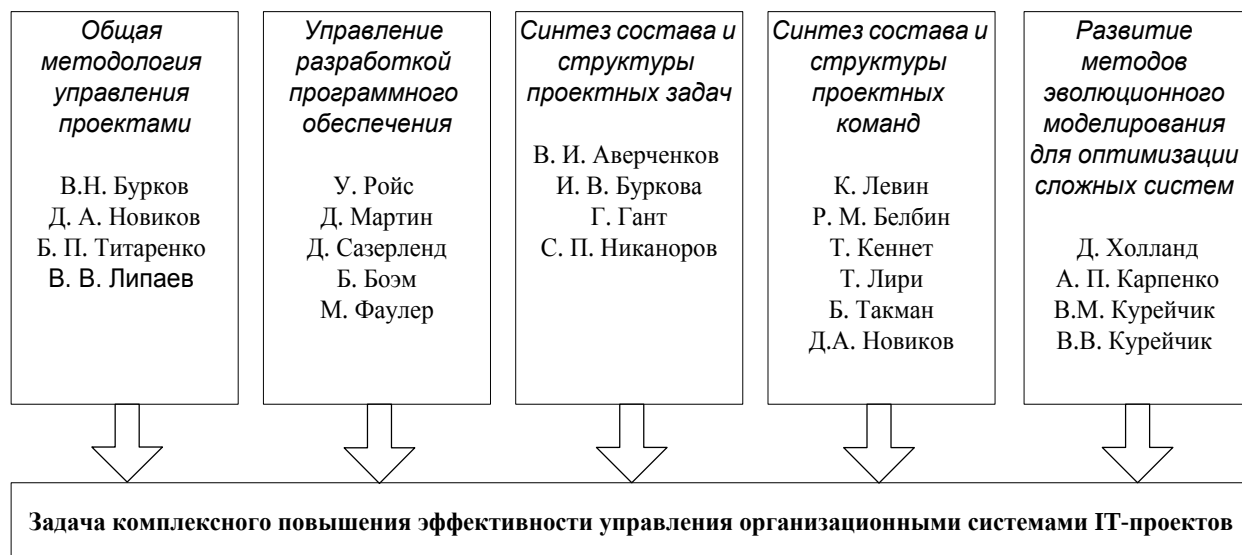


Рисунок 1.3 – Результаты обзора существующих подходов к решению задачи повышения эффективности управления проектами

### 1.3 Обзор существующих средств

Для повышения результативности и эффективности реализации сложных проектов руководителям проектов предназначены специализированные средства автоматизации процессов управления проектами [81].

Преимуществами систем управления проектами являются расчет длительности проекта и задач, построение диаграмм Ганта, распределение задачи между участниками проектов, контроль сроков, рисков, финансов.

К недостаткам систем управления проектами можно отнести риск усложнить простые проекты, высокую стоимость программ при низкой результативности, трату большого количества времени на составление планов вместо реальных действий.

На IT-рынке представлено множество программных продуктов по управлению проектами. Все они обладают базовым набором функций, таких как описание логической структуры проекта с указанием иерархии работ, расчет критического пути, контроль проектов, планирование ресурсов, анализ рисков, средства создания отчетов, поддержка мультипроектности. Однако реально на отечественном и российском рынках представлено небольшое количество программ, наиболее популярные из которых – Microsoft Project, Open Plan Professional, Spider Project, Primavera Project, Artemis Views.

MS Project. Входит в группу приложений MS Office [44, 99]. Наряду с системой управления персональной информацией MS Outlook и средством организации работы группы составляет комплекс программ рабочего места менеджера. На сегодняшний день является наиболее распространенной программой управления проектами.

Open Plan – продукт компании Welcom [104], профессиональная система управления проектами, которая позволяет облегчить нахождение эффективного распределения ресурсов и составления рабочего расписания, имеет мощные средства бюджетного и ресурсного планирования. По уровню автоматизации управления ресурсами и способам анализа рисков опережает MS Project[9].

Кроме Open Plan в семейство продуктов по управлению проектами компании Welcom входят также система управления бюджетом проектов Corba и Web-ориентированный инструмент для создания виртуального офиса

проекта WelcomHome. Важной особенностью Open Plan является возможность интеграции с различными системами предприятия и его открытая архитектура.

Российский пакет Spider Project. Основным отличием пакета Spider Project [94] от своих зарубежных аналогов является технология управления проектами, которая лежит в основе пакета.

Компания Primavera Systems, Inc. предлагает несколько продуктов для управления проектами. Пакет Sure Trek Project Manager [38] рекомендуется использовать для управления небольшими проектами. Для работы со сложными многоуровневыми проектами Primavera Systems, Inc выпустила программный продукт Primavera Project Planner (P3).

Artemis – продукт компании Artemis International, один из общепризнанных мировых лидеров на рынке программных решений для совместного управления ресурсами и проектами [94]. Artemis Views – целое семейство модулей, которые автоматизируют различные задачи управления проектами. Каждый модуль может использоваться как независимо, так и в комбинации с другими модулями.

Исходя из приведенного обзора можно сделать вывод, что рынок современного программного обеспечения для управления проектами представлен широким спектром продуктов, которые различаются степенью удовлетворения требований к управлению проектами.

Однако отсутствуют программные решения для многоуровневого управления проектами, а также специализированные решения для управления проектами, в комплексе учитывающие всю специфику IT-проектов.

#### **1.4 Постановка задачи**

Проведенный анализ существующих программных средств управления проектами в п. 1.3 показал, что на рынке информационных технологий

представлено большое количество систем управления проектами. При этом в существующих универсальных системах управления проектами не достаточно проработаны задачи учета таких особенностей IT-проектов как:

- необходимость учета личностных особенностей и психологических качеств участников при формировании команд разработчиков;
- иерархическую структуру задач в проекте, обуславливающую необходимость многоуровневого планирования работ и контроля качества решения задач;
- применение гибких методологий управления разработкой программного обеспечения обуславливает изменение требований к программному продукту на протяжении всего цикла разработки. Изменение требований к программному продукту приводит к необходимости оперативного изменения состава и структуры проектных задач, что обуславливает необходимость изменения состава и структуры команд исполнителей и перераспределения задач между участниками проекта.

Анализ существующих подходов к управлению IT-проектами позволил выявить следующие проблемы повышения эффективности управления IT-проектами (рисунок 1.4):

*При планировании проектных работ:*

- сложность определения трудоемкости проектных работ обусловлена большим количеством факторов, влияющих на длительность и трудоемкость решения каждой задачи и IT-проекта в целом;
- сложность модификации плана проектных работ при изменении требований заказчика, обусловленная необходимостью обработки большого количества информации о проектных задачах.

*При организации процесса разработки:*

- сложность синтеза рационального состава и структуры команд исполнителей при каждом изменении требований заказчика, обусловленная

необходимостью учета личностных и психологических качеств и результативности предыдущей работы в проекте;

– сложность рационального перераспределения задач между участниками команд исполнителей при изменении требований заказчика, обусловленная необходимостью обработки большого количества информации об участниках и задачах IT-проекта.

*При контроле качества проектных работ:*

– сложность контроля качества решения проектных задач, обусловленная отсутствием формализованной методики, обеспечивающей минимизацию ошибок в решении задач IT-проектов.

Эффективность управления организационными системами IT-проектов зависит от эффективности работы команды исполнителей, эффективности формирования состава и структуры задач, распределения задач между участниками проекта, эффективности контроля качества решения задач.

Существующие подходы рассматривают только **отдельные аспекты** задачи повышения эффективности управления организационными системами IT-проектами: формирование команд исполнителей; оценку трудоемкости проекта; распределение ресурсов проекта.

Анализ существующих подходов показал отсутствие **комплексных подходов** к повышению эффективности управления IT-проектами, обеспечивающих **комплексную обработку информации и поддержку принятия управленческих решений** на всех этапах проекта: при отборе участников проектов, формировании команд исполнителей, планировании и контроле качества проектных работ.

В связи с этим, актуальной является задача создания комплексного научного подхода, обеспечивающего решение научной задачи повышения эффективности управления организационными процессами IT-проектов.



Рисунок 1.4 – Проблемы повышения эффективности управления организационными системами IT-проектов и комплексный подход к их решению

Пусть руководитель проекта выбирает состав команды исполнителей  $team$  из множества  $TEAMS$  допустимых составов ( $team \in TEAMS$ ). В результате выбора состава команды исполнителей  $team \in TEAMS$ , исходя из потребностей проекта  $Req$  в используемых технологиях  $Tech_p$  и ограничении по длительности выполнения  $TDEV$  ( $Req = \langle Tech_p, TDEV \rangle$ ), реализуется результат  $team_0 \in TEAMS_0$ , где  $TEAMS_0$  – множество допустимых составов команд исполнителей. Допустимым является состав команды, в который каждый исполнитель входит один раз. Одновременно один исполнитель может входить только в одну команду.

При выборе состава команды исполнителей руководитель проекта действует согласно своим предпочтениям и тем, как состав команды  $Participants = \{Participant\}$  влияет на трудоемкость  $PM$  выполнения IT-проекта.

Соответственно, критерием повышения эффективности управления организационными системами IT-проектов является снижение трудоемкости решения задач проекта [115]

$$PM = \sum_{k=1}^N (PM_k^B \cdot \prod_{i=1}^6 EM_i) \rightarrow \min \quad (1.1)$$

при ограничении

$$TDEV = C \cdot (PM_{NS})^{D+0.2 \cdot 0.01 \cdot \sum_{j=1}^5 SF_j} \cdot \frac{SCED}{100} < T_{max} \quad (1.2)$$

где  $PM$  – трудоемкость проекта;  $EM_i$  – множители трудоемкости,

$$EM = \{PERS, PREX, RCPX, RUSE, PDIF, FCIL\},$$

где  $PERS$  – квалификация персонала;  $PREX$  – опыт персонала;  $RCPX$  – сложность и надежность продукта;  $RUSE$  – разработка для повторного использования;  $PDIF$  – сложность платформы разработки;  $FCIL$  – оборудование;  $PM_k^B$  – трудоемкость задачи проекта в чел.мес.;  $TDEV$  – длительность проекта;  $T_{max}$  – максимальное время выполнения проекта;  $C, D$  – коэффициенты, калибруемые по статистическим данным,  $C = 3,67$ ,  $D = 0,28$ ;  $PM_{NS}$  – трудоемкость проекта без учета множителя  $SCED$ , определяющего сжатие расписания;  $SF_j$  – значения факторов масштаба

$$SF = \{PREC, FLEX, RESL, TEAM, PMAT\},$$

где  $PREC$  – прецедентность, наличие опыта аналогичных разработок;  $FLEX$  – гибкость процесса разработки;  $RESL$  – архитектура и разрешение рисков;  $TEAM$  – сработанность команды;  $PMAT$  – зрелость процессов.

Исходными данными для решения задачи являются следующие компоненты IT-проектов: команда исполнителей, множество задач

$$IT = \langle Tasks, Participants \rangle$$

где  $Tasks$  – множество задач,  $Participants$  – команда исполнителей.

Участник команды исполнителей (*Participant*) характеризуется личностными и психологическими качествами (*PPQ*), перечнем (*Tech*) и уровнем владения технологиями (*LTech*), опытом участия в аналогичных проектах (*Ex*)

$$Participant = \langle PPQ, Tech, LTech, Ex \rangle.$$

Выбор состава команды руководителем проектов определяется правилом индивидуального рационального выбора  $P^{PM}(TEAM, Req, Participant) \subseteq TEAM$ , которое выделяет множество наиболее предпочтительных, с точки зрения руководителя проектов, составов команд.

Пусть руководитель проекта осуществляет декомпозицию задач на подзадачи, которые характеризуются типом (*Type*), технологией (*Tech*), приоритетом (*Ptask*)

$$Task = \langle Type, Tech, Ptask \rangle$$

и выбирает вариант распределения *distribution* исполнителей команды *team* на задачи из множества допустимых вариантов распределения *D*. В результате выбора варианта распределения исполнителей  $distribution \in D$  реализуется результат  $distribution_0 \in D_0$ , где  $D_0$  – множество допустимых вариантов распределения исполнителей.

При декомпозиции задач на подзадачи и выборе варианта распределения исполнителей на задачи руководитель проекта действует согласно своим предпочтениям и тем, как состав задач и вариант распределения на задачи проекта влияет на трудоемкость *PM* выполнения IT-проекта в целом. Трудоемкость выполнения IT-проекта зависит от состава задач  $Tasks = \{Task\}$  и характеристик исполнителей проектов  $Participants = \{Participant\}$ .

Выбор состава задач и варианта распределения исполнителей руководителем проектов определяется правилом индивидуального рационального выбора  $P^{PM}(D, Tasks, Participants) \subseteq D$ , которое выделяет множество наиболее предпочтительных, с точки зрения руководителя проектов, вариантов состава задач и распределения исполнителей на задачи.

Модель принятия решения руководителем проектов описывается кортежем  $\{TEAMS, TEAMS_0, D, D_0, Participants, Tasks, PM\}$ .

Структура модели принятия решения руководителем проекта представлена на рисунке 1.5.

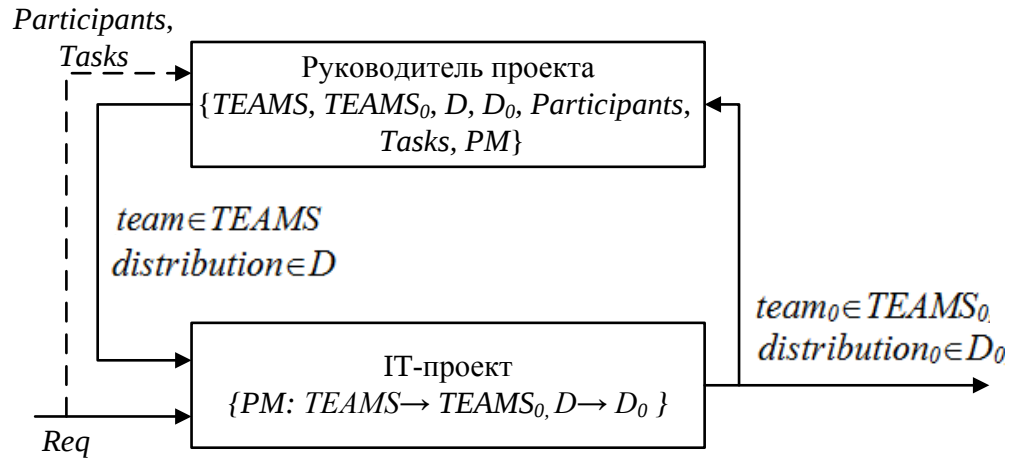


Рисунок 1.5 - Структура модели принятия решения руководителем проекта

Обозначим  $u = (team, distribution) \in U = TEAMS \otimes D$  – вектор управления. В качестве математического аппарата выбрано эволюционное моделирование, которое позволяет получить несколько вариантов рациональных составов команд и распределений исполнителей на задачи за меньшее количество итераций по сравнению с другими математическими методами. Целевая функция руководителя проектов рассчитывается на основе эволюционного моделирования при выборе управления  $u \in U$  и равна  $PM(u) = EvModel(Tasks, Participants) \rightarrow min$ .

Эффективность управленческого решения руководителя проектов без учета результатов эволюционного моделирования рассчитывается по формуле

$$K_m(u) = \frac{1}{PM_m(u)},$$

где  $PM_m(u)$  – трудоемкость выполнения проекта без учета результатов эволюционного моделирования. Эффективность управленческого решения

руководителя проектов с учетом результатов эволюционного моделирования оценивается по формуле

$$K_{EvM}(u) = \frac{1}{PM_{EvM}(u)},$$

где  $PM_{EvM}(u)$  – трудоемкость выполнения проекта с учетом результатов эволюционного моделирования.

Соответственно, задача повышения эффективности управления организационной системой ИТ-проекта заключается в поиске допустимого рационального управления, обеспечивающего

$$K_{em}(u) \rightarrow \max_{u \in U}.$$

Для повышения эффективности управления организационными процессами ИТ-проектов необходимо описать метод, содержащий математический аппарат решения задачи повышения эффективности управления организационными процессами ИТ-проектов, разработать алгоритмы поддержки принятия решений позволяющие автоматизировать организационные процессы ИТ-проектов на этапах планирования и организации работ при реализации ИТ-проектов, разработать программный комплекс и технологию управления организационными системами ИТ-проектов, обеспечивающие реализацию предложенных алгоритмов.

На рисунке 1.6 представлена структурная схема управления организационной системой ИТ-проектов на основе математических и программных средств, которые предлагается реализовать.

Выделенные на рисунке 1.6 блоки отражают компоненты математических и программных средств, реализующих повышение эффективности управления ИТ-проектом.



1 – новые требования заказчика; 2 – вариант распределения исполнителей на задачи;  
3 – результаты решения задач проекта.

Рисунок 1.6 – Структурная схема управления организационной системой ИТ-проекта

Таким образом, по итогам аналитического обзора предложена схема организационной системы ИТ-проекта, формализованы задачи исследования.

## **ГЛАВА 2. МЕТОДИЧЕСКИЕ ОСНОВЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ ОРГАНИЗАЦИОННЫМИ СИСТЕМАМИ ИТ-ПРОЕКТОВ**

### **2.1 Концепция повышения эффективности управления организационными системами ИТ-проектов**

Концепция повышения эффективности управления организационными процессами ИТ-проектов основана на модификации и дополнении отдельных аспектов методологии управления разработкой программного обеспечения Agile; **развитии и комплексном применении** современных подходов к оценке личностных и психологических качеств, методологии структурного анализа, методов формирования команд исполнителей, оценки трудозатрат, распределения ресурсов на задачи проекта для повышения эффективности управления организационными системами ИТ-проектов путем адаптации средств эволюционного моделирования для решения поставленной задачи диссертационного исследования.

Предложенная концепция (рисунок 2.1) основана на следующих принципах:

1) комплексный подход к повышению эффективности управления организационными системами ИТ-проектов с учетом результатов обработки информации о личностных и психологических качествах участников проектов с изменением параметров алгоритмов эволюционного моделирования в сочетании с методикой COCOMO II и методом Т. Саати при изменении параметров системы и внешней среды;

2) комплексная обработка информации и принятие управленческих решений при формировании команд исполнителей и распределении участников проекта на задачи путем адаптации алгоритмов эволюционного моделирования в сочетании с методикой COCOMO II и методом Т. Саати к решению задачи управления организационными процессами ИТ-проектов;

3) комплексная поддержка принятия решений на каждом шаге реализации IT-проекта при изменении системных (состава и структуры участников) и внешних (состава и структуры задач) факторов путем выбора параметров алгоритмов эволюционного моделирования.

Повышение эффективности управления организационными системами IT-проектов, в соответствии с предложенной концепцией, осуществляется путем применения комплексного подхода к управлению организационными процессами IT-проектов, включающего формирование команд исполнителей и распределения участников IT-проекта на задачи с использованием эволюционного моделирования, многоуровневый контроль качества решения проектных задач.

**Отличительными особенностями** представленной концепции повышения эффективности управления организационными системами IT-проектов являются:

1) комплексный подход к повышению эффективности управления организационными системами IT-проектов с применением модифицированных алгоритмов эволюционного моделирования для поддержки принятия решений в процессе синтеза состава и структуры команд исполнителей с учетом личностных и психологических качеств участников проекта, синтеза структуры и состава проектных задач и распределения участников проектов на задачи с учетом результатов многоуровневой оценки качества решения проектных задач;

2) поддержка принятия решения при управлении организационными процессами IT-проектов на этапе формирования команд исполнителей и распределение участников IT-проектов на задачи путем комплексной обработки информации о личностных и психологических качествах исполнителей проектов, трудоемкости и сложности решения проектных задач для сокращения количества итераций при использовании эволюционного моделирования, позволяющая обеспечить оперативное

Состав задач каждого этапа, обеспечивающих повышение эффективности управления организационными процессами ИТ-проектов, представлен в таблице 2.1.

Таблица 2.1 – Состав задач, обеспечивающих повышение эффективности управления организационными системами IT-проектов

| № этапа (шага) | Цель                                                                                                               | Задачи организации процесса                                                                                                                                                                                                                                           | Задачи управления                                                                                                                                                | Результат                                                                                                                                                 |
|----------------|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1              | 2                                                                                                                  | 3                                                                                                                                                                                                                                                                     | 4                                                                                                                                                                | 5                                                                                                                                                         |
| Этап 1         | Повышение эффективности управления формированием команд исполнителей                                               |                                                                                                                                                                                                                                                                       |                                                                                                                                                                  |                                                                                                                                                           |
| Шаг 1.1        | Комплексная оценка ЛПК участников проектов                                                                         | 1) Организация прохождения комплекса психологических тестов<br>2) Оценка результатов прохождения тестов                                                                                                                                                               | Предварительная обработка информации о результатах анализа ЛПК для разделения участников проекта на группы                                                       | Повышение эффективности управления организационным и процессами IT-проектов за счет сокращения количества конфликтных ситуаций при реализации проектов    |
| Шаг 1.2        | Комплексная оценка результатов решения задач каждым участником команды исполнителей проекта                        | Обработка информации о результатах многоуровневого контроля качества решения проектных задач                                                                                                                                                                          | Поддержка принятия решений по разделение участников проектов на группы по результатам предварительной обработки информации о результатах решения проектных задач | Повышение мотивации участников проекта, позволяющее сократить сроки решения проектных задач за счет повышения производительности труда                    |
| Шаг 1.3        | Ускорение работы алгоритмов эволюционного моделирования в процессе синтеза состава и структуры команд исполнителей | 1)Разделение участников проекта на группы с использованием кластерного анализа на основе комплексной обработки информации о ЛПК<br>2)Расчет трудоемкости проектов на основе методики СОСОМО II<br>3)Применение метода Т.Саати для ранжирования проектов по приоритету | Предварительная обработка информации для поддержки принятия решений по составу и структуре команд исполнителей                                                   | Повышение эффективности работы алгоритмов эволюционного моделирования за счет сокращения объема обработки информации при формировании команд исполнителей |

Продолжение таблицы 2.1

| 1       | 2                                                                                                                   | 3                                                                                                                                                                                                                | 4                                                                                                                                                                                                                               | 5                                                                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Шаг 1.4 | Формирование рационального состава и структуры команд исполнителей                                                  | Применение эволюционного моделирования для синтеза структуры и состава команды исполнителей                                                                                                                      | 1) Синтез состава и структуры команд исполнителей на основе предварительной комплексной обработки информации об участниках команд исполнителей<br>2) Оценка состава и структуры команды исполнителей лицом, принимающим решения | Сокращение времени решения проектных задач за счет рационального подбора состава и структуры команд исполнителей                                                      |
| Этап 2  | Повышение эффективности управления организационным процессом распределения участников команд исполнителей на задачи |                                                                                                                                                                                                                  |                                                                                                                                                                                                                                 |                                                                                                                                                                       |
| Шаг 2.1 | Формирование рационального состава проектных задач                                                                  | 1) Декомпозиция задачи проекта на подзадачи на основе методологии SADT<br>2) Ранжирование задач по уровню сложности с использованием метода Т.Саати;                                                             | Синтез состава и структуры проектных задач                                                                                                                                                                                      | Сокращение трудоемкости проекта                                                                                                                                       |
| Шаг 2.2 | Формирование рационального распределения участников команды исполнителей между задачами проекта                     | 3) Расчет трудоемкости задач проекта и проекта в целом на основе методики СОСОМО II.<br>4) Рациональное распределение задач проектов между участниками команд исполнителей на основе эволюционного моделирования | Оценка распределения участников команд исполнителей лицом, принимающим решение, с учетом результатов расчета трудоемкости проекта                                                                                               |                                                                                                                                                                       |
| Этап 3  | Повышение эффективности многоуровневого контроля качества решения проектных задач                                   |                                                                                                                                                                                                                  |                                                                                                                                                                                                                                 |                                                                                                                                                                       |
| Шаг 3.1 | Повышение качества решений проектных задач                                                                          | Построение многоуровневой системы контроля качества готовых решений                                                                                                                                              | Оценка результатов контроля качества решения проектных задач для поддержки принятия решений по структуре и составу участников команд исполнителей, распределению участников проектов на задачи                                  | Повышение эффективности решения проектных задач за счет сокращения количества ошибок в программных решениях, более раннего выявления ошибок в решении проектных задач |

Для решения задач формирования команд исполнителей и распределения участников команд исполнителей на задачи при управлении организационными системами IT-проектов разработаны метод и алгоритмы повышения эффективности на основе эволюционного моделирования.

## **2.2 Метод повышения эффективности управления организационными системами IT-проектов**

Метод включает следующие этапы повышения эффективности управления IT-проектами:

1 организация процесса управления формированием команд исполнителей на основе эволюционного моделирования в сочетании с методикой COSOMO II и методом парных сравнений Т. Саати с учетом ЛПК и профессиональных компетенций участников команд исполнителей;

2 организация процесса управления формирования перечня задач и распределения исполнителей на задачи на основе эволюционного моделирования в сочетании с методикой COSOMO II и методом парных сравнений Т. Саати с профессиональных компетенций участников команд исполнителей;

3 организация процесса многоуровневого контроля качества решения проектных задач.

Решение задачи повышения эффективности организационного процесса формирования команды исполнителей предполагает обработку информации о личностных и психологических качествах участников команды исполнителей с учетом их личностных и психологических качеств, уровня владения технологиями; применение эволюционного моделирования в сочетании с методикой COSOMO II и методом Т. Саати для формирования рационального состава команд исполнителей.

Рациональным считается состав команды, который руководитель проекта как лицо, принимающее решения, считает наиболее эффективным с

точки зрения реализации проекта. В процессе эволюционного моделирования формируется несколько вариантов составов команд исполнителей, из которых руководитель проекта выбирает рациональный с его точки зрения.

**Организация процесса управления формированием команд исполнителей** включает следующие шаги.

Шаг 1. Обработка информации об участниках команд исполнителей. Результатом данного шага является набор параметров, характеризующих личностные и психологические качества участников проектов, предпочитаемую и поддерживающую роли в команде [11], уровень владения технологиями, опыт применения технологий при реализации проектов.

*Исследование личностно-психологических особенностей участников команд исполнителей* проводится на основе обработки информации о результатах психологических тестов. Набор тестов включает тесты Р.М. Белбина, Т.Кеннета [40]. Результаты тестов позволяют определить склонность вести себя, принимать участие и взаимодействовать с другими определенным образом.

$$PPQ = \{ppq | ppq = (ppq_1, ppq_2, \dots, ppq_n), n = NPPQ\}$$

где  $NPPQ$  – количество тестов.

Каждый тест описывается набором признаков, состоящих из множества вопросов тестовых заданий, множества ответов объекта исследования, множества результатов обработки ответов, интерпретации результатов теста

$$PT = \langle Q, X, Def_X, Y, I \rangle$$

где  $Q$  – множество вопросов тестовых заданий,

$$Q_{pt} = \{q | q = (q_1, q_2, \dots, q_n), n = NQ\},$$

где  $NQ$  – количество вопросов в тесте;  $X$  – множество ответов объекта исследования

$$X = \{x | x \in Def_X, i = 1, \dots, NQ\},$$

где  $Def_X$  – область определения вариантов ответа;  $Y$  – множество результатов обработки ответов;  $I$  – интерпретация результатов теста.

Обработка информации о результатах психологического тестирования проводится методом кластерного анализа  $k$ -средних [79, 97] на этапе формирования команд исполнителей. В результате все претенденты на включение в команду исполнителей должны быть разделены на кластеры (группы) с учетом их личностных и психологических качеств.

*Обработка информации о результатах решения проектных задач для оценки уровня владения и опыта работы с технологиями и инструментами.* Оценка формируется на основании результатов участия в предыдущих проектах.

Перечень технологий  $Tech_{pp}$ , которыми владеет участник команды исполнителей формируется из множества  $T$  технологий и инструментов применяемых для разработки IT-проектов

$$Tech = \{tech | tech \in T\}.$$

Для каждой технологии из множества  $T$  оценивается уровень владения и опыт разработок участника проектов

$$LTech = \langle LT, Ex \rangle,$$

где  $LT$  – оценка уровня владения технологией;  $Ex$  – опыт участия в проектах с использованием данной технологии.

Для системных аналитиков, проектировщиков баз данных, программистов уровень владения технологией вычисляется для каждого завершенного проекта по формуле

$$LT = \frac{\sum_{i=1}^{NTasks} quality_i + \sum_{i=1}^{NTasks} speed_i}{NTasks} \cdot complexity, \quad (2.1)$$

где  $quality_i$  – качество решения  $i$ -й задачи проекта,

$$quality = 1 - \frac{ConfirmedErrors}{FindErrors}; \quad (2.2)$$

где  $ConfirmedErrors$  - число подтвержденных ошибок;  $FindErrors$  – число найденных ошибок;  $speed$  – скорость решения задачи проекта, находится как отношение планового срока решения задачи к фактическому сроку решения

$$speed = \frac{TPlanned}{TActual}; \quad (2.3)$$

где  $T_{Planned}$  – плановое время решения задачи;  $T_{Actual}$  – фактическое время решения задачи;  $N_{Tasks}$  – число задач проекта, решенных участником;  $complexity$  – сложность задачи.

Сложность задачи определяется методом парных сравнений Т. Саати.

Для тестировщиков уровень владения технологиями оценивается по формуле

$$quality = 1 - \frac{NoErrors}{FindErrors}; \quad (2.4)$$

где  $NoErrors$  - число неподтвержденных ошибок.

Опыт работы с технологиями и инструментами участника проектов  $Ex$  определяется по правилу 10000 часов[34]. Для этого находится суммарная длительность всех проектов с использованием оцениваемой технологии и в соответствии с таблицей 2.2 определяется уровень владения.

Таблица 2.2 – Оценка опыта работы с технологиями и инструментами

| Супер-низкий | Очень низкий | Низкий    | Нормальный | Высокий   | Очень высокий | Супер-высокий |
|--------------|--------------|-----------|------------|-----------|---------------|---------------|
| Менее 1500   | 1500 - 3000  | 3001-4500 | 4501-6000  | 6001-7500 | 7501-9000     | 10000 и более |

Таким образом, в результате прохождения первого шага каждый участник проектов характеризуется набором параметров

$$Participant = \langle PPQ, Tech, LTech, Ex \rangle.$$

Шаг 2. Оценка параметров проекта.

Проводится оценка объема программного кода проекта одним из следующих способов:

– с использованием экспертной оценки. Рекомендуется использовать для получения каждой оценки не менее трех независимых экспертов, усредняя их показания;

– руководителем проектов на основании собственного опыта или статистических данных.

Методом парных сравнений Т. Саати определяется уровень поправочных коэффициентов методики СОСОМО II: гибкость процесса разработки, архитектура и разрешение рисков, зрелость процессов, сложность и надежность продукта, разработка для повторного использования, сложность платформы разработки, оборудование.

Шаг 3. Управление процессом формирования структуры и состава команд на основе эволюционного моделирования.

Рассмотрим IT-проект, как объект, состоящий из таких компонент как множество задач и команда исполнителей

$$IT = \langle Tasks, Team \rangle$$

где  $Tasks$  – множество задач,  $Team$  – команда исполнителей.

Компонент  $Team$  имеет сложную структуру. Его можно декомпозировать следующим образом

$$Team = \langle Role, TP \rangle$$

где  $Role$  – множество командных ролей;  $TP$  – множество участников команды исполнителей.

$$Role = \{role | role = (role_1, role_2, \dots, role_m), m = NR\}$$

где  $NR$  – количество ролей в команде.

$$TP = \{tp | tp = (tp_1, tp_2, \dots, tp_n), n = NTP\}$$

где  $NTP$  – количество участников команды исполнителей.

Каждая задача проекта из множества  $Tasks$  задается набором характеристик

$$Task = \langle T, Type, Tech, Priority, complexity \rangle,$$

где  $complexity$  – сложность задачи;  $T$  – время выполнения задачи

$$T = \{t | t \in N\},$$

$Type$  – тип задачи

$$Type = \{type | type \in Def\},$$

где  $Def$  – область определения типа задачи;

$Tech$  – применяемая технология

$$Tech = \{tech | tech \in Def_T\},$$

где  $Def_T$  – область определения типа технологии;

$Priority$  – приоритет задачи.

Задача управления организационным процессом формирования состава команды исполнителей проекта состоит в том, чтобы сформировать состав, при котором трудоемкость проекта и срок выполнения проекта минимизируются. При этом выполняются следующие условия:

- участники команды исполнителей совместимы по личностно-психологическим качествам;
- команда исполнителей является сбалансированной с точки зрения теории командных ролей Р. М. Белбина.

Таким образом, сформулирована задача оптимизации: сформировать такую команду исполнителей *Participants* из множества возможных, при которой

$$PM(Team) \rightarrow \min$$

при условии

$$TDEV(Team) < TDEV_{max}.$$

Структура команды исполнителей задается руководителем проектов. В соответствии со структурой руководитель проектов формирует один или несколько вариантов состава команд исполнителей.

Для улучшения состава команды используется алгоритм эволюционного моделирования, описанный в п. 2.3.

Состав команды исполнителей формируется в соответствии с правилами:

- участники принадлежат разным кластерам, сформированным по результатам кластерного анализа на шаге 1;
- в команде представлено максимальное количество командных ролей.

Для расчета значения поправочного коэффициента «Квалификация персонала» проводится нормирование уровня владения технологиями каждого участника команды исполнителей

$$LTech_{tp}^{norm} = \frac{LTech_{tp}}{LTech_{max}} \quad (2.5)$$

Уровень владения технологиями командой определяется по формуле

$$LTech_{general} = \frac{\sum LTech_{tp}^{norm}}{N} \quad (2.6)$$

где  $N$  – количество участников команды исполнителей.

Значение поправочного коэффициента «Квалификация персонала» выбирается в соответствии с таблицей 2.3.

Таблица 2.3 – Значения поправочного коэффициента «Квалификация персонала» методики СОСОМО II

| Супернизкий  | Очень низкий  | Низкий            | Нормальный        | Высокий           | Очень высокий     | Супер-высокий |
|--------------|---------------|-------------------|-------------------|-------------------|-------------------|---------------|
| $\leq 0.143$ | 0.144...0.286 | 0.287...<br>0.429 | 0.430...<br>0.571 | 0.572...<br>0.714 | 0.715...<br>0.857 | $> 0.857$     |

Значения поправочных коэффициентов «Опыт персонала» и «Наличие опыта аналогичных разработок» рассчитываются как среднее значение по команде.

Значение поправочного коэффициента «Сработанность команды» рассчитывается в соответствии с таблицей 2.4. Для этого вычисляется наибольший процент участников команды, принимавших участие в разработке одного проекта.

Таблица 2.4 – Значения поправочного коэффициента «Сработанность команды» методики СОСОМО II

| Очень низкий | Низкий | Средний | Высокий | Очень высокий | Критический |
|--------------|--------|---------|---------|---------------|-------------|
| 20%          | 40%    | 60%     | 75%     | 90%           | 100%        |

Схема управления организационным процессом формирования состава команды исполнителей представлена на рисунке 2.2. В выделенном блоке используется авторский алгоритм синтеза структуры и состава команд исполнителей IT-проекта.



Рисунок 2.2 – Схема управления организационным процессом формирования структуры и состава команд исполнителей IT-проектов

Особенности предложенной схемы процесса управления формированием структуры и состава команд исполнителей:

1) **Комплексная обработка информации об участниках команд исполнителей** на основе исследования личностно-психологических качеств с использованием методов кластерного анализа, знаний, умений и навыков участников команд исполнителей с использованием методов обработки результатов решения проектных задач, описанных в п. 2.5.

Предварительная обработка информации обеспечивает ускорение работы алгоритмов эволюционного моделирования.

2) Синтез рационального состава и структуры команд исполнителей на основе эволюционного моделирования с формированием системы ограничений, обусловленных спецификой IT-проектов, в процессе настройки параметров алгоритмов эволюционного моделирования в сочетании с использованием методики COSOMO II и метода парных сравнений Т. Саати для оценки трудоемкости проекта.

Предложенная схема процесса управления формированием структуры и состава команд IT-проектов обеспечивает комплексную обработку информации о кандидатах на участие в проектах и повышение эффективности управленческих решений при формировании команд исполнителей лицом, принимающим решения.

**Процесс управления распределением участников команд исполнителей IT-проекта на задачи** предполагает определение структуры и состава проектных задач с учетом рационального распределения участников команды исполнителей на задачи.

Предложенная схема организационного процесса управления распределением исполнителей IT-проектов на задачи (рисунок 2.3) обеспечивает комплексную оценку целесообразности декомпозиции задач проекта на подзадачи с точки зрения трудоемкости IT-проектов в целом, что позволяет автоматизировать процесс управления распределением исполнителей IT-проектов на задачи с использованием модулей управления организационным процессом формирования структуры и состава задач проектов, управления организационным процессом распределения исполнителей IT-проекта на задачи, расчета трудоемкости проектов.

В выделенном блоке используется авторский алгоритм распределения исполнителей IT-проекта на задачи.

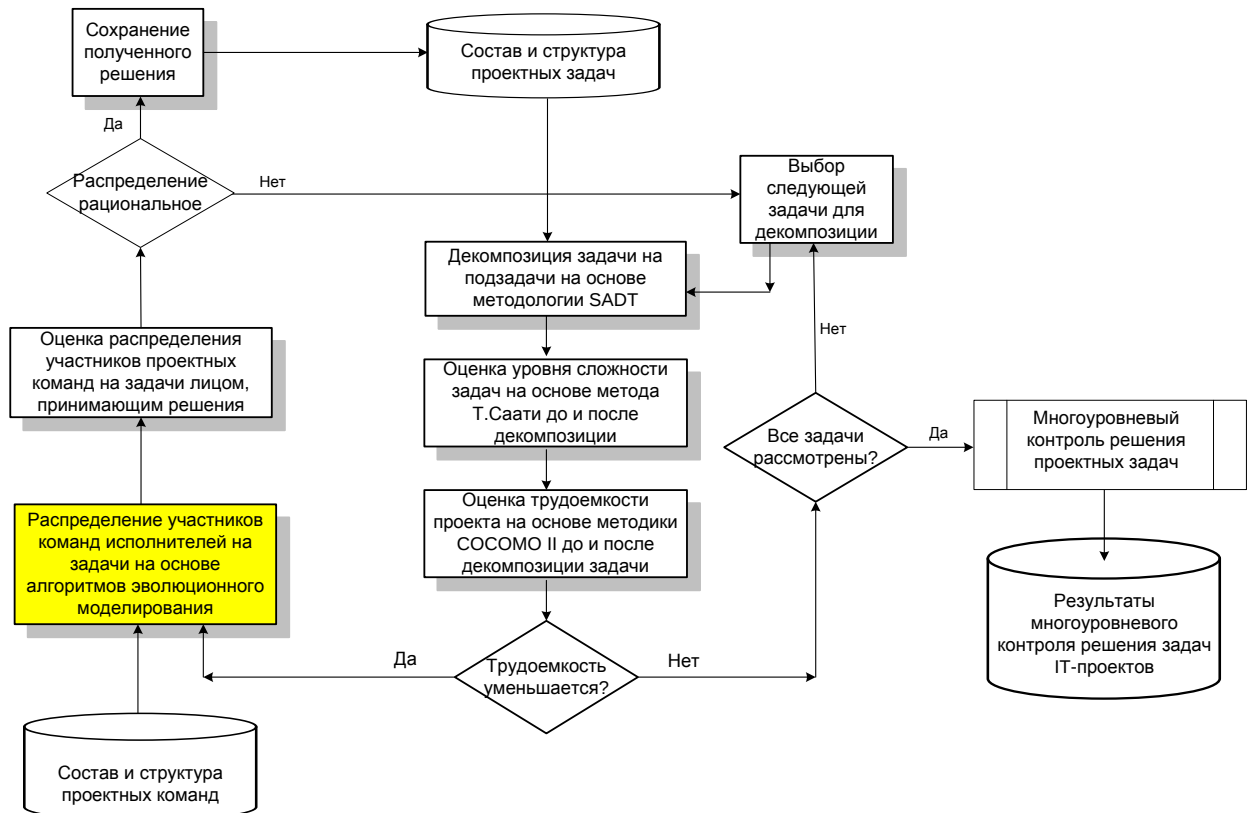


Рисунок 2.3 – Схема управления организационным процессом распределения исполнителей ИТ-проектов на задачи

**Особенности предложенной схемы процесса управления распределением участников команд исполнителей на задачи:**

1) Комплексная обработка информации в процессе синтеза состава и структуры проектных задач с использованием методологии SADT для декомпозиции задачи на подзадачи, методики COCOMO II для оценки трудоемкости проекта в сочетании с методом парных сравнений Т. Саати для оценки сложности задач. Предложенная процедура обработки информации обеспечивает комплексную оценку целесообразности декомпозиции задач проекта на подзадачи с точки зрения трудоемкости ИТ-проектов в целом. Полученные результаты являются исходными данными для алгоритмов эволюционного моделирования, использующихся при распределении участников команд ИТ-проектов на задачи.

2) Распределение участников команд проектов на задачи с использованием алгоритмов эволюционного моделирования с

формированием системы ограничений, обусловленных спецификой IT-проектов, в процессе настройки параметров алгоритмов эволюционного моделирования. Предложенная схема процесса управления распределением участников команд исполнителей на задачи обеспечивает комплексную обработку информации и повышение эффективности управленческих решений при формировании состава и структуры проектных задач, назначении участников команд IT-проектов на задачи.

***Шаги формирования структуры и состава проектных задач и распределение участников команд исполнителей на задачи***

Как было показано выше, каждая задача проекта характеризуется следующими параметрами: трудоемкость, тип, используемая технология, приоритет

$$Task = \langle T, Type, Tech, Priority, complexity \rangle.$$

Трудоемкость каждой задачи проекта определяется с использованием методики COSOMO II. Объем программного кода определяется методом экспертных оценок либо на основе статистических данных руководителем проектов. Тип и используемая технология определяются руководителем проекта. Приоритет задачам присваивает заказчик.

Выбирается задача с наивысшим приоритетом. Выполняется ее декомпозиция на подзадачи

$$TaskPriority = \{task_1, task_2, \dots, task_n\}.$$

На каждую подзадачу назначается исполнитель из числа участников команды исполнителей

$$Distribution = \{tp_1, tp_2 \dots, tp_n\}.$$

Рациональным считается распределение исполнителей, при котором

$$PM_{TaskPriority} (Distribution) \rightarrow \min$$

при условии

$$TDEV_{TaskPriority} (Distribution) < TDEV_{max}.$$

Руководителем проектов формируется один или несколько вариантов распределения исполнителей на подзадачи.

Для улучшения первоначального варианта распределения применяется алгоритм эволюционного моделирования, описанный в п. 2.4.

Руководителем проекта оценивается целесообразность декомпозиции задачи на подзадачи с точки зрения трудоемкости проекта в целом.

Последовательная декомпозиция задач прекращается, если выполнено одно из следующих условий:

- все задачи станут элементарными;
- дальнейшая декомпозиция каждой задачи станет нецелесообразной с точки зрения трудоемкости;
- все участники проектной команды распределены на задачи.

### **2.3 Алгоритм синтеза структуры и состава команд исполнителей**

Алгоритм синтеза структуры и состава команд исполнителей состоит из следующих шагов (рисунок 2.6).

Подготовительный шаг. Настройка параметров алгоритма: размер популяции *population*, число поколений  $N_{step}$ , номер итерации  $i=1$ .

Формирование структуры команды исполнителей руководителем проектов (блок Б1, рис. 2.6). Первоначальное создание одной или нескольких команд исполнителей руководителем проектов (блок Б2, рис. 2.6).

Шаг 1. Формирование начальной популяции команд исполнителей (блок Б3, рис. 2.6).

Шаг 2. Вычисление функции приспособленности каждой команды по формулам (1.1), (1.2) (блок Б4, рис. 2.6).

Шаг 3. Операция селекции (блок Б5, рис. 2.6). В операции селекции используется турнирный отбор, Проводится два раунда турнирного отбора. В каждом раунде случайным образом отбирается заданное количество команд исполнителей. Родителем становится команда исполнителей с наилучшей

функцией приспособленности. Результатом шага является две команды исполнителей  $s^1$  и  $s^2$ .

Шаг 4. Операция скрещивания (блок Б6, рис. 2.6). Для формирования состава новых команд используется односточный кроссинговер (рисунок 2.4). Случайным образом из промежутка  $[1, N]$  (где  $N$  – количество участников команды исполнителей) выбирается точка разрыва. Команды  $s^1$  и  $s^2$  по этой точке делятся на две части и меняются соответствующими частями.

Осуществляется проверка допустимости состава команды исполнителей. Допустимым считается решение, в котором каждый участник входит в состав команды только один раз.

Результатом шага 4 являются две новые команды  $s^{ch1}$  и  $s^{ch2}$ , полученные путем применения операции скрещивания к исходным командам  $s^1$  и  $s^2$ .

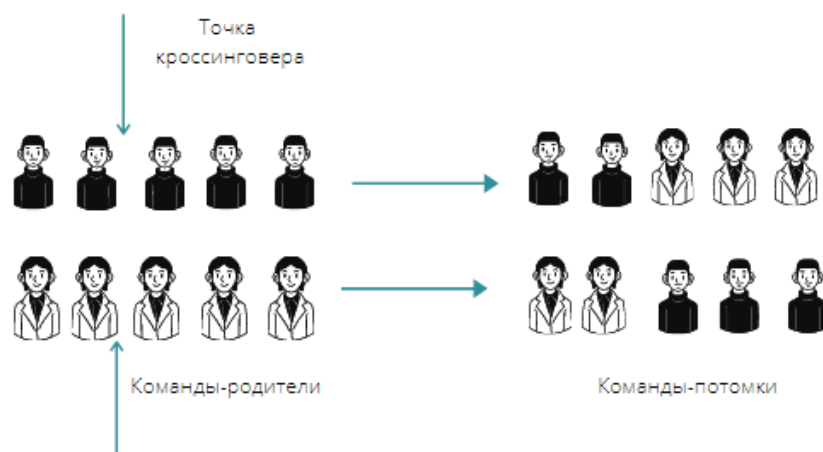


Рисунок 2.4 – Операция скрещивания в задаче формирования команды исполнителей

Шаг 5. Операция мутации (блок Б7, рис. 2.6). Мутация в рамках решаемой задачи заключается в удалении случайным образом выбранного участника команды и присоединении случайно выбранного нового участника (рисунок 2.5).

Осуществляется проверка допустимости состава команды исполнителей. Допустимым считается решение, в котором каждый участник входит в состав команды только один раз.

Результатом шага 5 являются две новые команды  $s^{M1}$  и  $s^{M2}$ , полученные путем применения оператора мутации к исходным командам  $s^1$  и  $s^2$ .

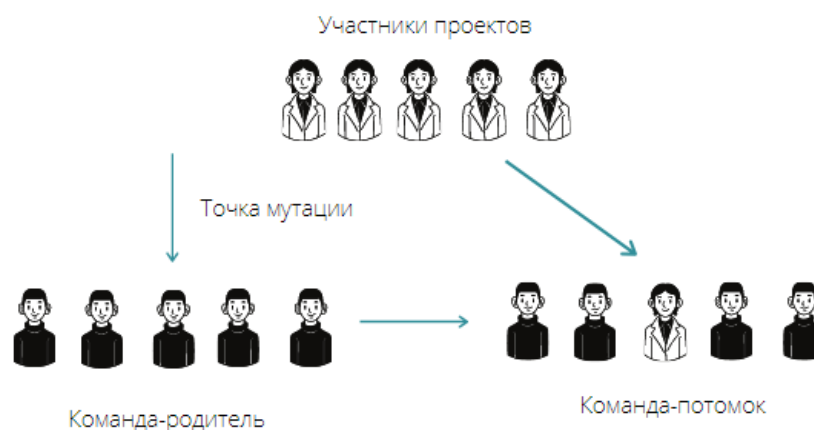


Рисунок 2.5 – Операция мутации в задаче формирования команды исполнителей

Шаг 6. Сокращение числа команд исполнителей в промежуточной популяции (блок Б8, рис. 2.6). Применяется стратегия элитарного отбора. Промежуточная популяция включает исходные команды и команды, полученные в результате скрещивания и мутации. Выбираются *population* команд с наилучшими показателями функции приспособленности.

Проверяются условия:

- если  $i < N_{step}$ , то полагаем  $i = i + 1$  и переходим к шагу 2;
- если  $i = N_{step}$ , то переходим к шагу 7;
- если наилучшее значение функции приспособленности не улучшилось, переходим к шагу 7.

Шаг 7. Окончание работы эволюционного моделирования. В качестве решения задачи выбирается команда исполнителей с наилучшим значением функции приспособленности.

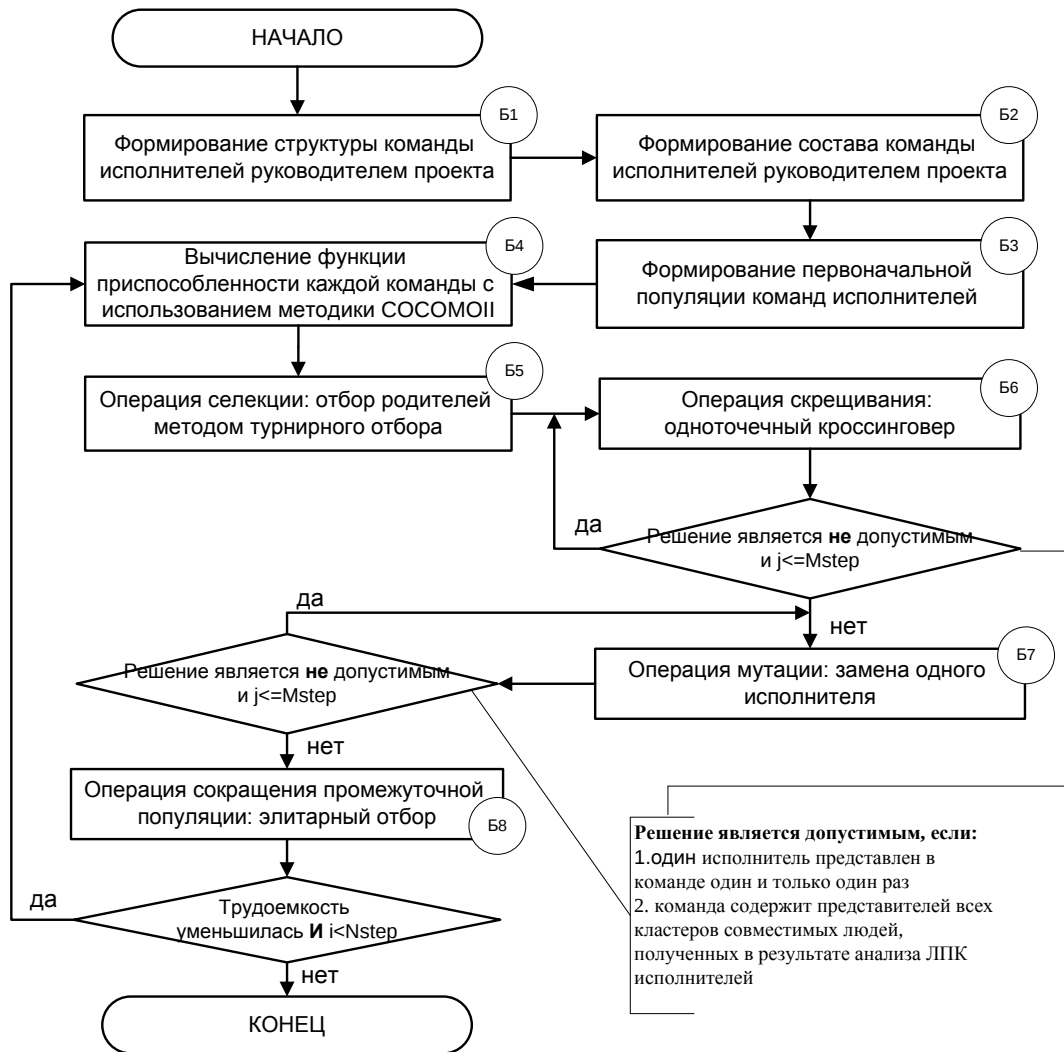


Рисунок 2.6 – Схема алгоритма синтеза состава команды исполнителей на основе эволюционного моделирования

## 2.4 Алгоритм назначения исполнителей на задачи проекта

Алгоритм назначения исполнителей на задачи состоит из следующих шагов (рисунок 2.9).

Подготовительный шаг. Задаются параметры эволюционного моделирования: размер популяции *population*, число поколений  $N_{step}$ , номер итерации  $i=1$ , количество вариантов распределения в турнирном отборе.

Руководителем проектов формируется один или несколько вариантов распределения исполнителей на задачи с учетом ограничений:

- один исполнитель выполняет одну задачу;
- одна задача выполняется одним исполнителем.

Варианты распределения, отличающиеся только порядком следования исполнителей, считаются различными.

Считается, что исполнитель, стоящий в варианте распределения на первом месте назначен на выполнение первой задачи в выборке задач и т. д.

Шаг 1. Вычисляется функция приспособленности для каждого варианта распределения.

В поправочном коэффициенте методики СОСОМО II «Квалификация команды» учитывается уровень владения исполнителем только той технологией, которая необходима для выполнения задачи, на которую назначен исполнитель.

Шаг 2. Операция селекции. Используется турнирный отбор. Проводится два раунда турнирного отбора. В каждом раунде случайным образом отбирается заданное количество вариантов распределения исполнителей на задачи. Родителем становится вариант распределения с наилучшей функцией приспособленности. Результатом этого шага будут два варианта распределения исполнителей на задачи  $s^1$  и  $s^2$ .

Шаг 3. Операция скрещивания. Используется кроссинговер порядка (рисунок 2.7). Случайным образом выбираются две точки разрыва. Генотип первого родителя делится на три секции по точкам разрыва. Секция, находящаяся между точками разрыва, полностью копируется в генотип потомка в те же самые абсолютные позиции.

Формируется блок заполнения из генотипа второго родителя из исполнителей, которых нет в скопированной секции потомка. Исполнители добавляются последовательно, начиная со второй точки разрыва. Когда будет достигнут конец генотипа второго родителя, процесс заполнения продолжается с первой позиции.

Первый исполнитель блока заполнения помещается в первую позицию генотипа потомка. Следующие исполнители занимают свободные позиции в относительном порядке.

Второй потомок формируется аналогичным образом для тех же родителей, роли которых поменялись.

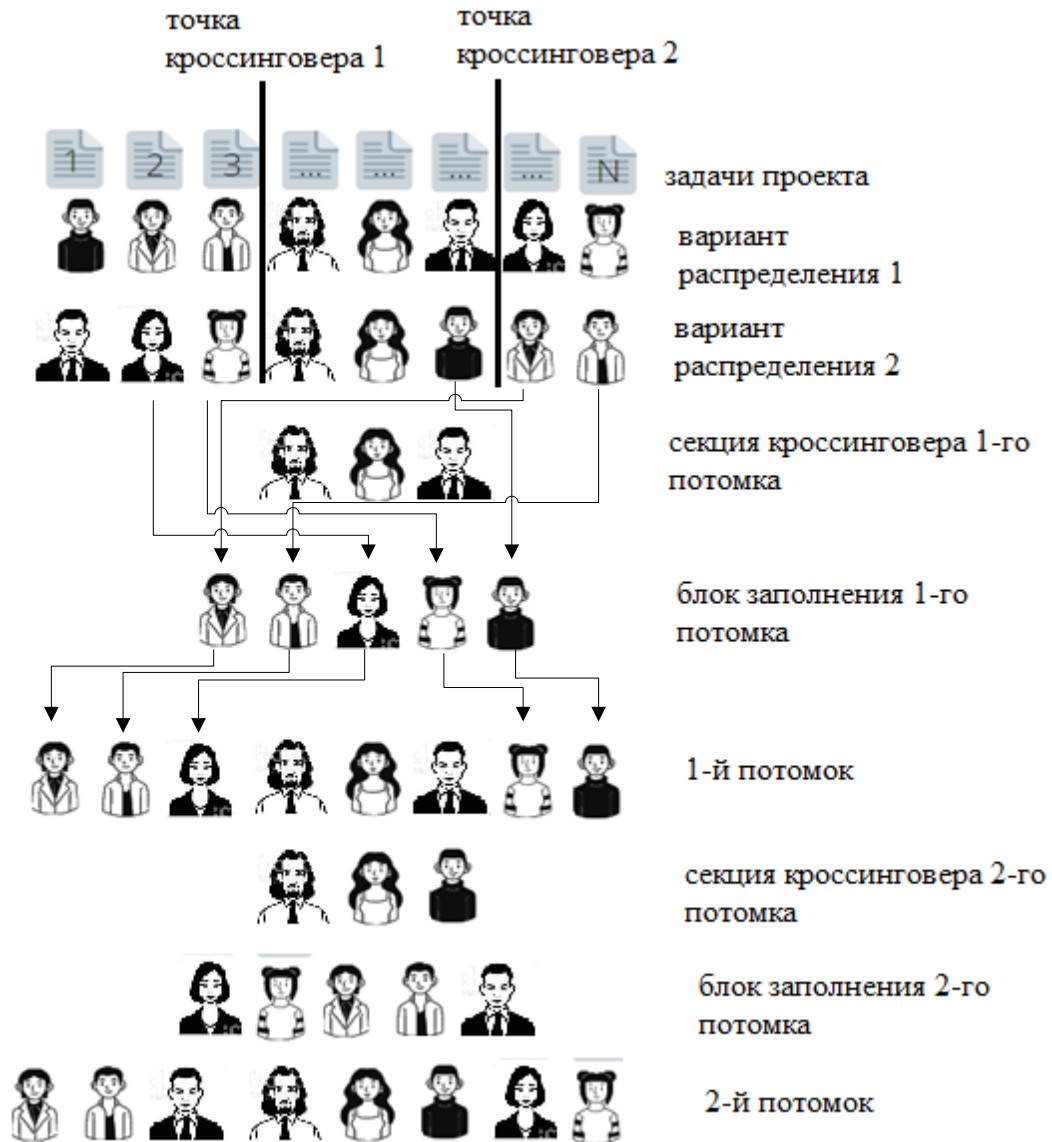


Рисунок 2.7 – Операция скрещивания при распределении исполнителей на задачи

Результатом шага 3 являются два варианта распределения  $s^{ch1}$  и  $s^{ch2}$ .

Шаг 4. Операция мутации. Используется точечная мутация, которая в рамках решаемой задачи заключается в перестановке для двух исполнителей задач проекта (рисунок 2.8).

Результатом шага 4 являются новые варианты распределения  $s^{M1}$  и  $s^{M2}$ .

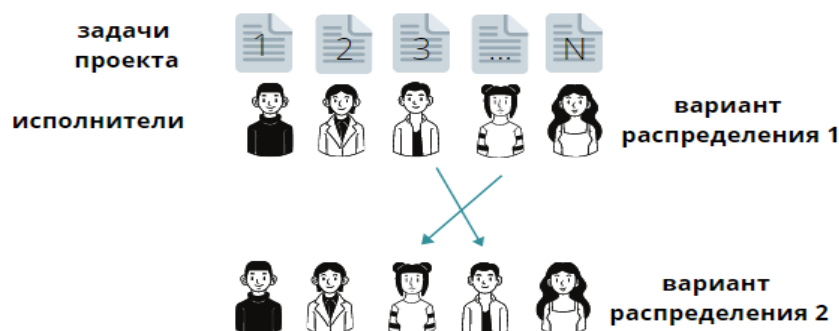


Рисунок 2.8 – Операция мутации при распределении исполнителей на задачи

Шаг 5. Сокращение популяции вариантов распределения. Применяется стратегия элитарного отбора. Промежуточная популяция включает исходные варианты распределения и варианты распределения, полученные в результате скрещивания и мутации. Выбирается *population* вариантов распределения с лучшими значениями функции приспособленности.

Проверяются условия:

- если  $i < N_{step}$ , то  $i = i + 1$  и выполняется переход к шагу 2;
- если  $i = N_{step}$ , то выполняется переход к шагу 6;
- если наилучшее значение функции приспособленности не улучшилось, то выполняется переход к шагу 6.

Шаг 6. Окончание работы эволюционного моделирования. В качестве решения задачи выбирается вариант распределения исполнителей на задачи с наилучшим значением функции приспособленности из последней популяции.

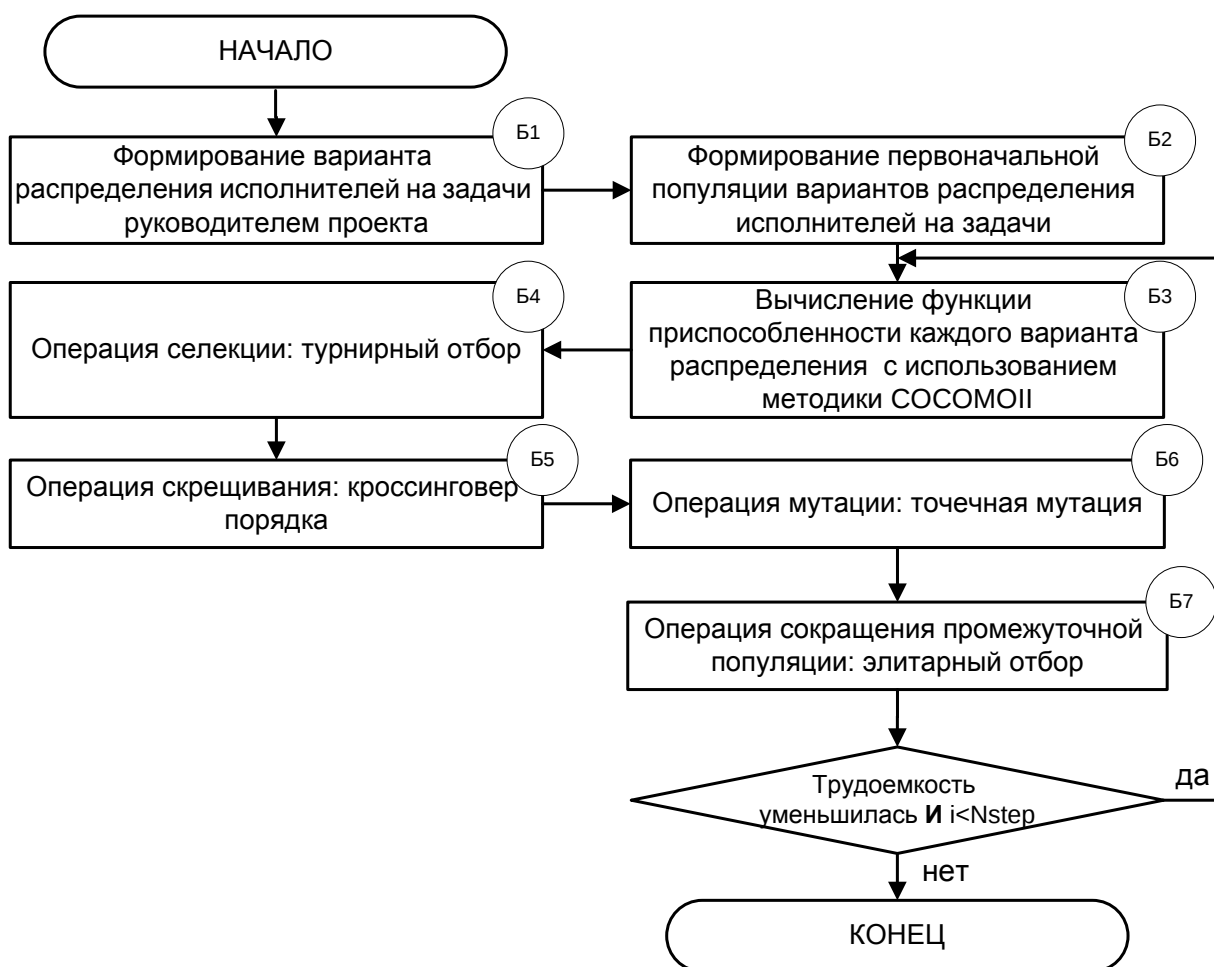


Рисунок 2.9 – Схема алгоритма распределения исполнителей на задачи проекта на основе эволюционного моделирования

## 2.5 Методика многоуровневой оценки качества решения проектных задач

Целью многоуровневого контроля качества решения задач ИТ-проекта является сокращение сроков решения задач и минимизация ошибок, которые заказчик или владелец продукта и пользователи обнаруживают в разработанном ПО [73].

В результате декомпозиции ИТ-проекта образуется иерархия задач, качество решения которых необходимо оценивать для определения эффективности работы каждого участника ИТ-проекта. Оценка качества решения проектных задач необходима для повышения мотивации участников команд ИТ-проекта.

Методика многоуровневого контроля качества решения проектных задач (рисунок 2.10) включает следующие этапы:

Этап 1. Определение приоритетов задач на каждом уровне декомпозиции с использованием метода анализа иерархий Т. Саати.

Шаг 1.1. Многоуровневая оценка приоритетов задач на основе матриц парных сравнений. Приоритеты задач верхнего уровня иерархии определяются заказчиком или владельцем продукта. Приоритеты задач каждого следующего уровня иерархии формируются руководителем проекта с учетом приоритетов задач более высокого уровня и парных сравнений важности задач руководителем проекта или SCRUM-мастером.

Шаг 1.2. Проверка согласованности приоритетов задач на каждом уровне иерархии.



Рисунок 2.10 – Схема методики многоуровневого контроля качества решения задач IT-проекта

Этап 2. Многоуровневая оценка качества решения задач (разработчиков оценивают тестировщики, разработчики оценивают проектировщиков,

проектировщики оценивают бизнес-аналитиков, тестировщиков оценивают заказчики) Разбивка задачи на 3 составляющие (BA, SA, DEV)

Структурная схема этапа 2 представлена на рисунке 2.11.

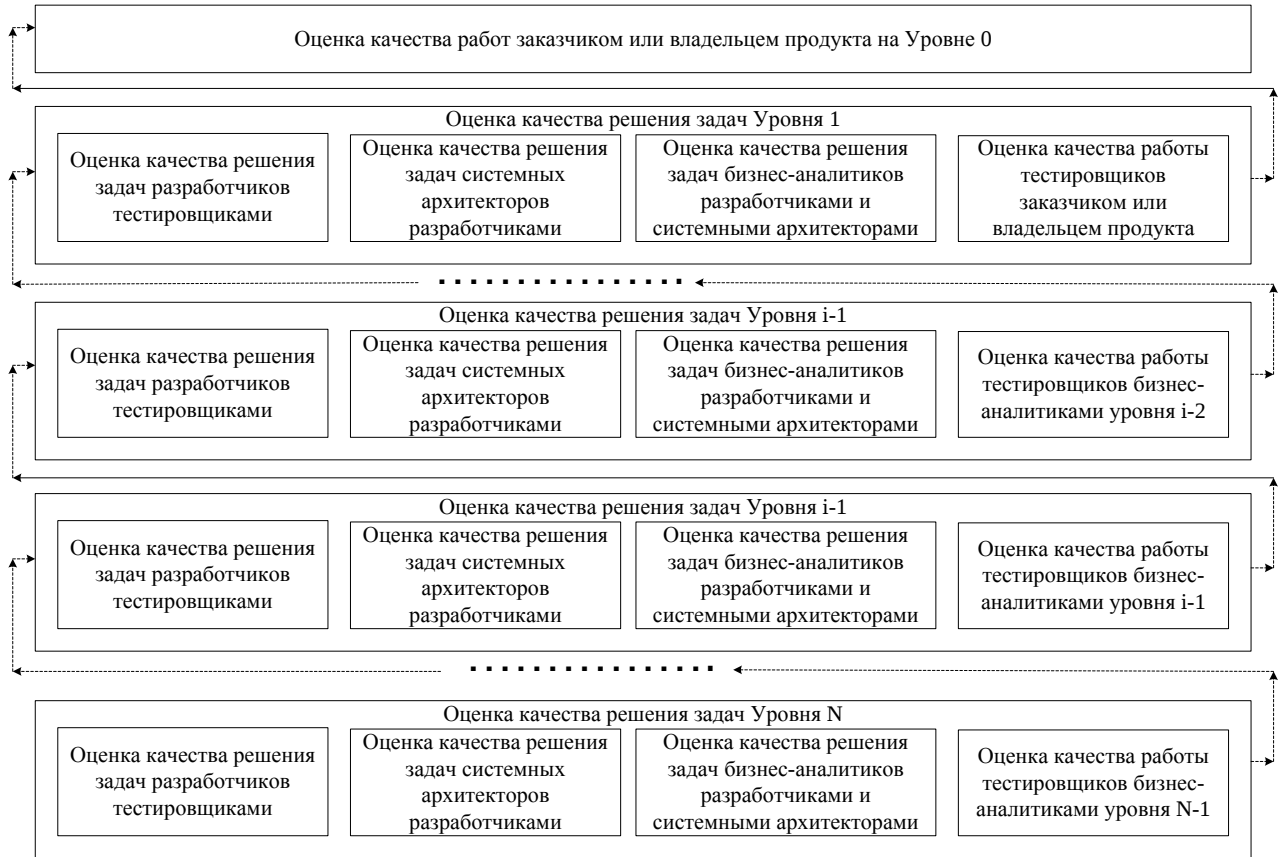


Рисунок 2.11 – Структурная схема многоуровневой оценки качества решения задач IT-проектов

Шаг 2.1. Оценка тестировщиками результатов решения задач разработчиками. Для каждой решенной задачи проекта качество оценивается по формуле:

$$Level_{dev} = (\sum quality_{dev} + \sum speed_{dev}) \cdot complexity_{dev} \cdot priority,$$

где  $quality_{dev}$  – качество решения задачи проекта разработчиками

$$quality_{dev} = 1 - \frac{ConfirmedErrors_{dev}}{\sum FindErrors_{dev}};$$

где  $ConfirmedErrors_{dev}$  - число подтвержденных ошибок разработчика в решении задачи;  $FindErrors_{dev}$  - число найденных тестировщиками ошибок

разработчика в решении задачи;  $speed_{dev}$  – скорость решения задачи проекта, находится как отношение планового времени решения задачи разработчиком к фактическому времени решения

$$speed_{dev} = \frac{TimePlanned_{dev}}{TimeActual_{dev}};$$

где  $TimePlanned_{dev}$  – плановое время решения задачи разработчиком;  $TimeActual_{dev}$  – фактическое время решения задачи разработчиком;  $complexity_{dev}$  – сложность задачи разработчика;  $priority$  – приоритет задачи.

Сложность задачи определяется методом парных сравнений Саати.

Шаг 2.2. Оценка разработчиками результатов решения задач системными архитекторами. Для каждой решенной задачи проекта качество оценивается по формуле:

$$Level_{SA} = (\sum quality_{SA} + \sum speed_{SA}) \cdot complexity_{SA} \cdot priority,$$

где  $quality_{SA}$  – качество решения задачи проекта разработчиками

$$quality_{SA} = 1 - \frac{ConfirmedErrors_{SA}}{\sum FindErrors_{SA}};$$

где  $ConfirmedErrors_{SA}$  – число подтвержденных ошибок системного архитектора;  $FindErrors_{SA}$  – число найденных разработчиками ошибок системного архитектора в решении задачи;  $speed_{SA}$  – скорость решения задачи проекта, находится как отношение планового времени решения задачи системным архитектором к фактическому времени решения

$$speed_{SA} = \frac{TimePlanned_{SA}}{TimeActual_{SA}};$$

где  $TimePlanned_{SA}$  – плановое время решения задачи системным архитектором;  $TimeActual_{SA}$  – фактическое время решения задачи системным архитектором;  $complexity_{SA}$  – сложность задачи системного архитектора.

Шаг 2.3. Оценка системными архитекторами результатов решения задач системными аналитиками. Для каждой решенной задачи проекта качество оценивается по формуле:

$$Level_{BA} = (\sum quality_{BA} + \sum speed_{BA}) \cdot complexity_{BA} \cdot priority,$$

где  $quality_{dev}$  – качество решения задачи проекта разработчиками

$$quality_{dev} = 1 - \frac{ConfirmedErrors_{BAdev}}{\sum FindErrors_{BAdev}} - \frac{ConfirmedErrors_{BAAsa}}{\sum FindErrors_{BAAsa}};$$

где  $ConfirmedErrors_{BAdev}$  - число подтвержденных ошибок бизнес-аналитика, найденных разработчиками;  $FindErrors_{BAdev}$  - число найденных разработчиками ошибок в решении задачи бизнес-аналитиком;  $ConfirmedErrors_{BAAsa}$  - число подтвержденных ошибок бизнес-аналитика, найденных системным архитектором;  $FindErrors_{BAAsa}$  - число найденных системным архитектором ошибок в решении задачи бизнес-аналитиком;  $speed_{BA}$  – скорость решения задачи проекта, находится как отношение планового времени решения задачи бизнес-аналитиком к фактическому времени решения

$$speed_{BA} = \frac{TimePlanned_{BA}}{TimeActual_{BA}};$$

где  $TimePlanned_{BA}$  – плановое время решения задачи бизнес-аналитиком;  $TimeActual_{BA}$  - фактическое время решения задачи бизнес-аналитиком;  $complexity_{BA}$  – сложность задачи бизнес-аналитика.

Шаг 2.4. Оценка бизнес-аналитиками более высокого уровня результатов решения задач тестировщиками более низкого уровня. Результаты реализации проекта оцениваются заказчиком или владельцем продукта. Для каждой решенной задачи проекта качество работы тестировщиков оценивается по формуле:

$$quality_{tester} = 1 - \frac{NoConfirmedErrors}{FindErrors},$$

где  $NoConfirmedErrors$  - число неподтвержденных ошибок.

Этап 3. Формирование исходных данных для поддержки принятия решений по составу и структуре задач следующего спринта и распределению участников IT-проекта на задачи.

## ВЫВОДЫ

В ходе диссертационного исследования для повышения эффективности управления организационными системами IT-проектов разработаны авторские решения:

- новая концепция решения задачи повышения эффективности управления организационными процессами IT-проектов, позволяющая выполнять комплексную обработку информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей; комплексную поддержку принятия решений при управлении синтезом состава и структуры проектных задач и распределении участников проекта на задачи с использованием средств эволюционного моделирования.

- новый метод повышения эффективности управления организационными процессами IT-проектов обеспечивающий поддержку принятия решений при формировании состава команд исполнителей и назначения участников проекта на задачи,

- новые алгоритмы поддержки принятия решений по формированию структуры и состава команд исполнителей и распределения исполнителей на задачи проекта путем адаптации средств эволюционного моделирования к решению задачи планирования выполнения работ в IT-проекте.

Применение алгоритмов эволюционного моделирования в сочетании с методикой СОСОМО II для расчета трудоемкости проекта и метода парных сравнений Т. Саати для определения значений поправочных коэффициентов методики СОСОМО II позволяют повысить эффективность управления организационными системами IT-проектов при формировании структуры и

состава команды исполнителей и распределения участников команд исполнителей на задачи проекта.

Ограничением предложенного метода является участие исполнителя одновременно только в одном проекте.

## ГЛАВА 3. ПРОГРАММНЫЙ КОМПЛЕКС УПРАВЛЕНИЯ ИТ-ПРОЕКТАМИ

### 3.1 Обоснование выбора средств реализации программного комплекса

При разработке программного комплекса использовалась система контроля версий Git [105]. Она бесплатна, имеет открытый исходный код, имеет возможность развёртывания локального сервера или использования удалённого сервера. В качестве удаленного сервера использовался GitHub.com. Бесплатного аккаунта достаточно для выполнения поставленных задач.

Программный модуль серверной части разработан на языке Java [45, 78] с использованием Spring Boot, Spring Data и Spring Security [64].

Проект Spring Boot упрощает создание самостоятельных приложений промышленного уровня на основе Spring, которые остается лишь запустить на выполнение. В состав Spring Boot входят готовые конфигурации для различных типов приложений Spring, организованные в стартовые пакеты. Например, пакет web-starter содержит предварительно сконфигурированный и легко настраиваемый контекст веб-приложений, а также обеспечивает стандартную поддержку встраиваемых контейнеров сервлетов Tomcat 7+, Jetty 8+ и Undertow 1.3.

Библиотека DevExtreme — свободный для некоммерческого использования набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения [33, 63, 89].

Достоинства использования библиотеки компонентов:

- уменьшение количества времени, затрачиваемого на разработку;

- адаптивность. Позволяет создавать адаптивные сайты. Дизайн сайта будет корректно отображаться на экранах устройств разных размеров вне зависимости от их диагонали;
- кроссбраузерность. Сайты, сделанные с использованием библиотеки, будут одинаково отображаться во всех современных браузерах;
- легкость в использовании и быстрота в освоении;
- понятный код. Позволяет писать качественный и понятный код, который легко поймет другой разработчик, что значительно упрощает разработку в команде.

Angular – открытая платформа, нацеленная прежде всего, на применение для разработки одностраничных веб-приложений, поддерживая командой разработчиков Google. Языком программирования в Angular принято использовать TypeScript, который является транслируемым в JavaScript. Предоставляет функциональность двухстороннего связывания и реактивного программирования, позволяет удобно организовать работу с событиями, асинхронно программировать [49].

В качестве реляционной базы данных использовалась PostgreSQL в связи со свободной лицензией, кроссплатформенностью и поддержкой стандарта SQL 11 [47].

### **3.2 Архитектура программного комплекса**

Программный комплекс имеет модульную архитектуру (рисунок 3.1) [26, 27, 103].

Модуль обработки информации о личностно-психологических качествах участников проектов M1 распределяет претендентов на участие в проектах на четыре кластера на основе результатов психологического тестирования методом кластерного анализа  $k$ -средних. Количество кластеров выбрано по количеству групп командных ролей – лидерование, деятельность, мышление,

социализация. Данные о результатах психологических тестов загружаются в программный комплекс из программы, реализующей прохождение и обработку результатов тестов М. Белбина и К. Томаса.

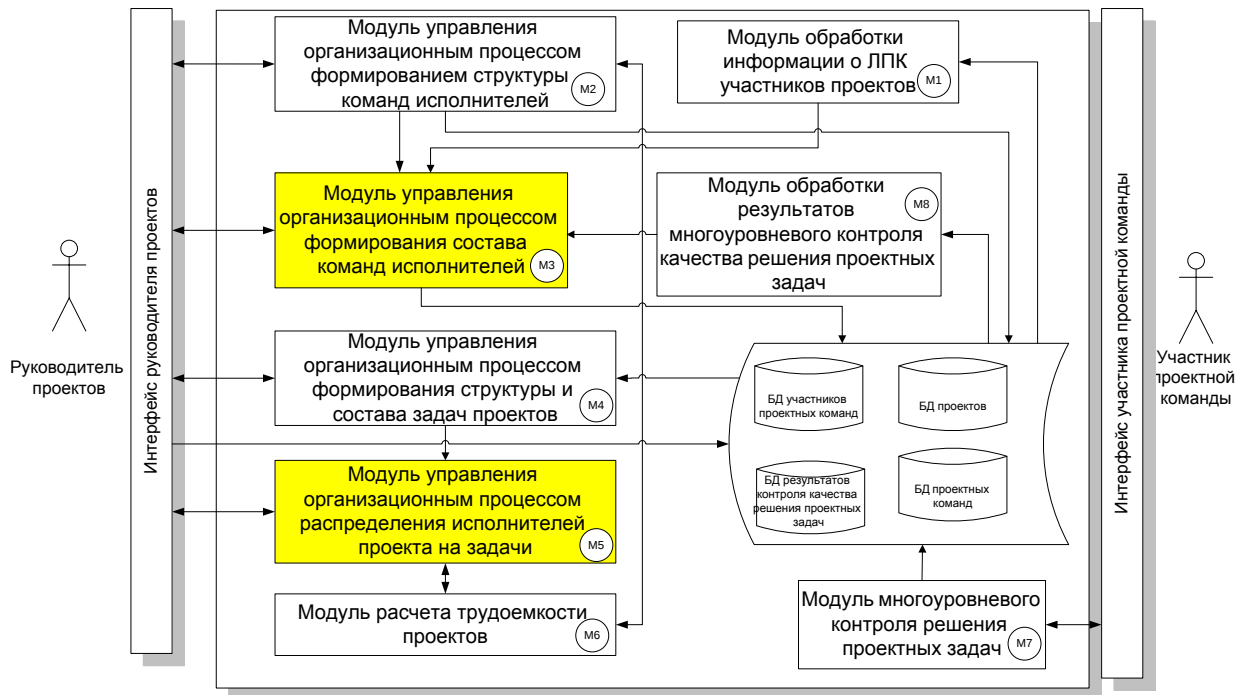


Рисунок 3.1 – Архитектура программного комплекса

В выделенных на рисунке 3.1 компонентах программного комплекса реализованы авторские алгоритмы повышения эффективности управления организационными процессами ИТ-проектов.

Модуль управления организационным процессом формирования структуры команд исполнителей M2 предназначен для создания рациональной структуры команды исполнителей руководителем ИТ-проекта на основе предварительной оценки трудоемкости с использованием методики COSOMO II и информации об используемых технологиях.

Модуль управления организационным процессом формирования состава команд исполнителей M3 предназначен для поддержки принятия решений в организационных системах ИТ-проектов при решении задачи формирования команд исполнителей.

В модулях управления организационными процессами формирования структуры и состава реализован авторский алгоритм синтеза структуры и состава команд исполнителей на основе эволюционного моделирования. При разработке алгоритма учтены специфические особенности IT-проектов, в процессе эволюционного моделирования использована методика COCOMO II и метод парных сравнений Т. Саати для оценки трудоемкости проекта (модуль М6).

Модуль управления организационным процессом формирования структуры и состава задач проектов М4 предназначен для поддержки принятия решений в организационных системах IT-проектов в процессе декомпозиции задач проекта на подзадачи.

Модуль управления организационным процессом распределения исполнителей IT-проекта на задачи М5 обеспечивает автоматизацию назначения исполнителей на задачи проекта в соответствии с авторским алгоритмом. Предложенный алгоритм обеспечивает поддержку принятия решений в процессе синтеза состава и структуры проектных задач с использованием методологии IDEF0 для декомпозиции каждой задачи проекта на подзадачи, методики COCOMO II для оценки трудоемкости проекта в сочетании с методом парных сравнений Т. Саати.

Модуль обработки результатов многоуровневого контроля качества решения проектных задач М8 обеспечивает получение и обработку информации о результатах решения проектных задач для оценки профессиональных компетенций исполнителей проектов. Результаты контроля качества решения проектных задач используются для принятия решений в задачах управления организационным процессом формирования состава и структуры команд исполнителей и назначения исполнителей IT-проектов на задачи. Поступление данных о результатах решения проектных задач в систему обеспечивает модуль М7.

Таким образом, решена задача создания программного комплекса и алгоритмов управления организационными процессами IT-проектов.

### 3.3 Проектирование основных компонентов программного комплекса

В программном комплексе реализованы следующие основные функции, обеспечивающие автоматизацию методов и алгоритмов, представленных в главе 2:

- *обработка информации о личностных и психологических качествах исполнителей* с использованием метода кластерного анализа для уменьшения количества итераций эволюционного моделирования (модуль М1 рис. 3.1). Данные психологического тестирования поступают из внешнего модуля. Результатом выполнения данной функции является разделение участников проектов на пять кластеров в зависимости от модели поведения в команде, определяемой на основании теста Т. Кеннета и предпочитаемой командной роли, . определяемой на основании теста Р. М. Белбина.

- *анализ результатов многоуровневого контроля качества решения проектных задач* для расчета поправочных коэффициентов методики СОСОМО II (модуль М8 рис. 3.1). Данные поступают из базы данных «БД участников проектных команд». Результатом выполнения данной функции является оценка уровня владения технологиями и инструментами исполнителем.

- *создание команды исполнителей* с применением эволюционного моделирования в сочетании с методикой СОСОМО II для расчета трудоемкости проектов, метода парных сравнений Т.Саати для оценки поправочных коэффициентов методики СОСОМО II и интеграцией результатов комплексной обработки информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей в операции эволюционного моделирования. Количество строк программного кода и поправочные коэффициенты проекта для методики СОСОМО II поступают из базы данных «БД проектов».

Информация об участниках проектов, необходимая для расчета поправочных коэффициентов команды, поступает из базы данных «БД участников проектных команд». Результатом выполнения данной функции является рациональный состав команды исполнителей.

– *назначение исполнителей на задачи* с применением эволюционного моделирования в сочетании с методикой COSOMO II для расчета трудоемкости проектов и метода парных сравнений Т. Саати для оценки поправочных коэффициентов методики COSOMO II. Данные об участниках команды исполнителей поступают из базы данных «БД проектных команд». Результатом выполнения данной функции является рациональный вариант распределения исполнителей на задачи.

Помимо основных функций на диаграмме вариантов использования представлены вспомогательные функции, обеспечивающие выполнение основных:

– *регистрация исполнителя* (модуль М2 рис. 3.1) для ввода данных об исполнителе проектов. Для оценки уровня знания технологий исполнитель решает тестовые задачи. Полученные данные сохраняются в базе данных «БД результатов контроля качества решения проектных задач» и в дальнейшем используются при формировании команд исполнителей.

– *регистрация проекта* (модуль М4 рис. 3.1) для ввода данных о проекте: наименование, краткое описание, приоритет, дата начала выполнения, планируемая дата завершения. Данные сохраняются в базе данных «БД проектов».

– *создание команды исполнителей* (модуль М3 рис. 3.1) для ввода данных о структуре команды исполнителей проекта. Данные сохраняются в базе данных «БД проектных команд».

– *просмотр информации об исполнителе* (модуль М2 рис. 3.1) для предоставления руководителю проектов информации об уровне владения инструментами и технологиями участником проектов для расчета уровня

владения технологиями и инструментами командой. Данные о совместном участии исполнителей в проектах используется для расчета коэффициента сработанности команды методики СОСОМО II в модуле М6. Данные поступают из базы данных «БД проектных команд».

– *просмотр результатов формирования команд на основе эволюционного моделирования* (модуль М3 рис. 3.1) для оценки структуры и состава команды исполнителей руководителем проекта. Результатом является предоставление руководителю проекта информации об оптимальных составах команд исполнителей, полученных по результатам синтеза состава команд на основе эволюционного моделирования. В случае если несколько команд имеют одинаковую трудоемкость, выбор состава команды осуществляется руководителем проекта.

– *формирование списка задач проекта* (модуль М4 рис. 3.1) для декомпозиции задач проекта на подзадачи. Исходные данные поступают из базы данных «БД проектов» и уточняются с учетом новых требований заказчика. Результатом является многоуровневый список задач.

– *назначение исполнителей на задачи* (модуль М5 рис. 3.1) для формирования первоначального варианта распределения исполнителей на задачи руководителем проектов; для просмотра результатов эволюционного моделирования распределения исполнителей на задачи для оценки целесообразности декомпозиции задачи на подзадачи с точки зрения уменьшения трудоемкости выполнения проекта в целом.

– *вывод списка задач* (модуль М4 рис. 3.1) проекта исполнителю. Результатом является предоставление информации об используемых технологиях, сроках исполнения, дефектах.

– *просмотр результатов контроля качества решения задач* (модуль М8 рис. 3.1) проекта руководителем проекта и исполнителем. Результатом является предоставление информации о фактических сроках исполнения, количестве дефектов, фактической трудоемкости.



Рисунок 3.2 – Диаграмма вариантов использования программного комплекса

В таблице 3.1 представлены пользовательские истории для каждого варианта использования разработанного программного комплекса.

Таблица 3.1 – Пользовательские истории

| Актер                | Действие                | Цель                                                                                                                                             |
|----------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                    | 2                       | 3                                                                                                                                                |
| руководитель проекта | регистрация IT-проекта  | управление организационной системой IT-проекта на всех этапах выполнения на основе эволюционного моделирования                                   |
|                      | регистрация исполнителя | сбор исходных данных для поддержки принятия решений при формировании структуры и состава проектных команд и распределения исполнителей на задачи |

Продолжение таблицы 3.1

| 1                    | 2                                                                                           | 3                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| руководитель проекта | обработка информации о личностных и психологических качествах участников проектов           | использование результатов для принятия решения при формировании структуры и состава проектных команд и распределения исполнителей на задачи |
|                      | создание команд исполнителей                                                                | выполнение проектов с наименьшей трудоемкостью                                                                                              |
|                      | просмотр результатов эволюционного моделирования                                            | принятие решения о структуре и составе команд исполнителей                                                                                  |
|                      | формирование списка задач проекта                                                           | управление организационными процессами IT-проектов на этапе планирования проектных работ                                                    |
|                      | назначение исполнителей на задачи                                                           | получение рационального варианта распределения исполнителей на задачи                                                                       |
|                      | осуществление анализа результатов многоуровневого контроля качества решения проектных задач | расчет поправочных коэффициентов методики COCOMO II                                                                                         |
|                      | просмотр информации об исполнителях                                                         | получение информации об уровне владения инструментами и технологиями участником проектов                                                    |
|                      | просмотр списка задач                                                                       | получение информации об используемых технологиях, сроках исполнения, дефектах                                                               |
|                      | просмотр результатов контроля качества решения задач                                        | получение информации о фактических сроках исполнения, количестве дефектов, фактической трудоемкости                                         |
| Исполнитель проекта  | просмотр списка задач                                                                       | получение информации об используемых технологиях, сроках исполнения, дефектах                                                               |
|                      | просмотр результатов контроля качества решения задач                                        | получение информации о фактических сроках исполнения, количестве дефектов, фактической трудоемкости                                         |

Последовательность действий руководителя проектов при формировании структуры и состава проектных команд представлена на рисунке 3.3.

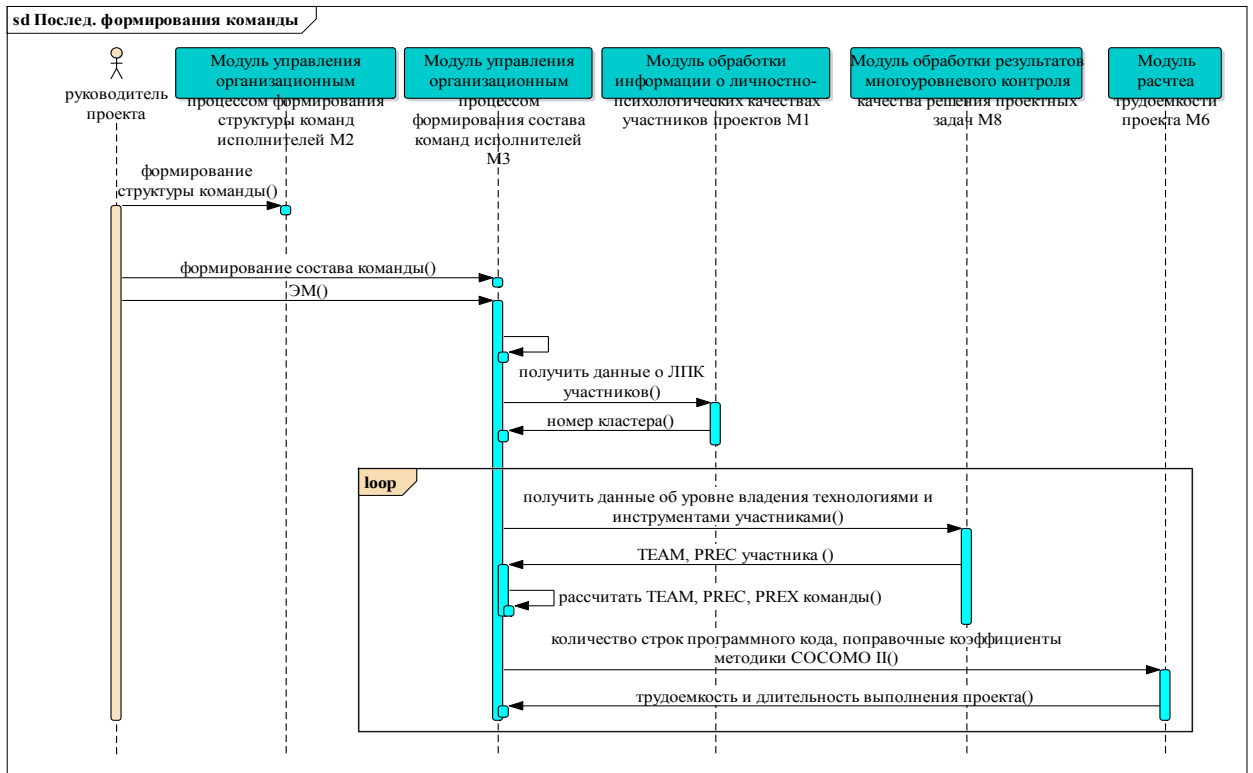


Рисунок 3.3 – Диаграмма последовательности варианта использования «Создание команды исполнителей»

Последовательность действий руководителя проектов при распределении исполнителей на задачи представлена на рисунке 3.4.

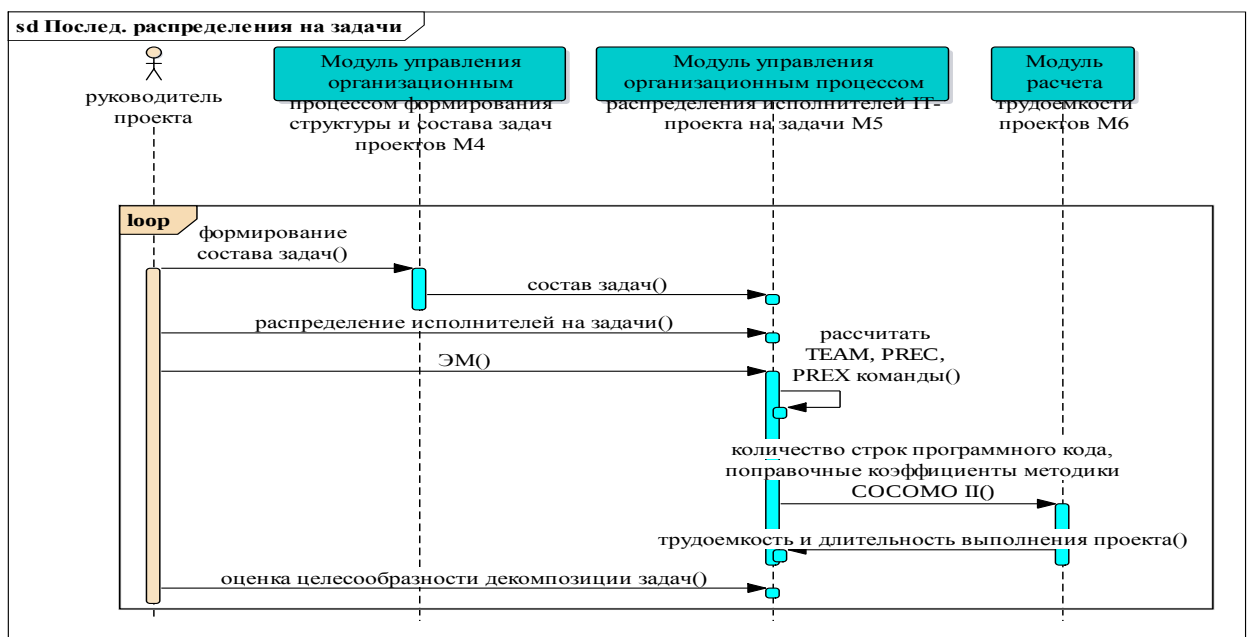


Рисунок 3.4 – Диаграмма последовательности варианта использования «Назначение исполнителей на задачи»

### **3.4 Технология управления IT-проектами с использованием разработанного программного комплекса**

Разработанная технология определяет порядок использования основных модулей программного комплекса для автоматизации основных этапов метода управления организационными процессами IT-проектов (рисунок 3.2).

Технологические этапы повышения эффективности управления организационным процессам IT-проектов.

*Этап 1.* Комплексная обработка информации о личностных и психологических качествах исполнителей IT-проектов (модуль M1), результатах решения проектных задач исполнителями IT-проектов (модуль M8) для сокращения количества итераций при решении задачи формирования команд исполнителей.

Данные о результатах психологических тестов загружаются в программный комплекс из программы, реализующей прохождение и обработку результатов тестов М. Белбина, К. Томаса. По результатам тестов осуществляется разделение участников команд исполнителей на группы методом k-средних в зависимости от их личностных и психологических качеств.

Оценка уровня владения технологиями и инструментами, опыта работы с технологиями и инструментами осуществляется на основании результатов участия в предыдущих проектах по результатам оценки качества решения проектных задач. Для этого используется модуль обработки результатов многоуровневого контроля качества решения проектных задач (модуль M8). Исходные данные для оценки уровня владения технологиями и инструментами участниками команд исполнителей поступают в модуль M7 в процессе выполнения проектов.

*Этап 2.* Формирование исходных данных для применения методики СОСОМО II.

*Шаг 2.1.* Оценка размера программного кода проекта. Размер программного кода определяется руководителем проекта с привлечением ведущих разработчиков на основе опыта решения аналогичных задач реализованных ранее IT-проектов. Выходными данными являются количество строк программного кода.

*Шаг 2.2.* Получение значений поправочных коэффициентов (гибкость процесса разработки, архитектура и разрешение рисков, зрелость процессов, сложность и надежность продукта, разработка для повторного использования, сложность платформы разработки, оборудование) с использованием метода парных сравнений Т. Саати. Выходными данными этапа является значения поправочных коэффициентов методики COSOMO II.

*Этап 3.* Формирование состава и структуры команды исполнителей.

*Шаг 3.1.* Принятие решения руководителем проектов о структуре команды исполнителей (модуль M2).

*Шаг 3.2.* Формирование руководителем проектов первоначального состава команды исполнителей (модуль M3).

*Шаг 3.3.* Улучшение состава команды исполнителей на основе эволюционного моделирования (модуль M3).

*Шаг 3.4.* Оценка руководителем проектов структуры и состава команды исполнителей. Принятие решения о рациональной структуре и составе команды исполнителей.

*Этап 4.* Синтез структуры и состава задач IT-проекта. Декомпозиция наиболее приоритетной задачи на подзадачи на основе методологии SADT (модуль M4).

*Этап 5.* Распределение исполнителей на задачи проекта

*Шаг 5.1.* Первоначальное распределение исполнителей на задачи руководителем проектов (модуль M5).

*Шаг 5.2.* Улучшение варианта распределения исполнителей на задачи проекта с применением эволюционного моделирования (модуль M5).

*Шаг 5.3.* Оценка трудоемкости проекта (модуль М6). Принятие решений по назначению исполнителей на задачи ИТ-проекта на основе результатов эволюционного моделирования.

*Этап 6.* В процессе выполнения проекта осуществляется многоуровневый контроль качества решения проектных задач, на основании которого формируется оценка квалификации и опыта разработки участников команды исполнителей (модуль М7).

Этапы 2-6 периодически повторяются по мере изменения требований заказчика, обеспечивая повышение скорости обработки информации для поддержки принятия решений руководителем ИТ-проекта.

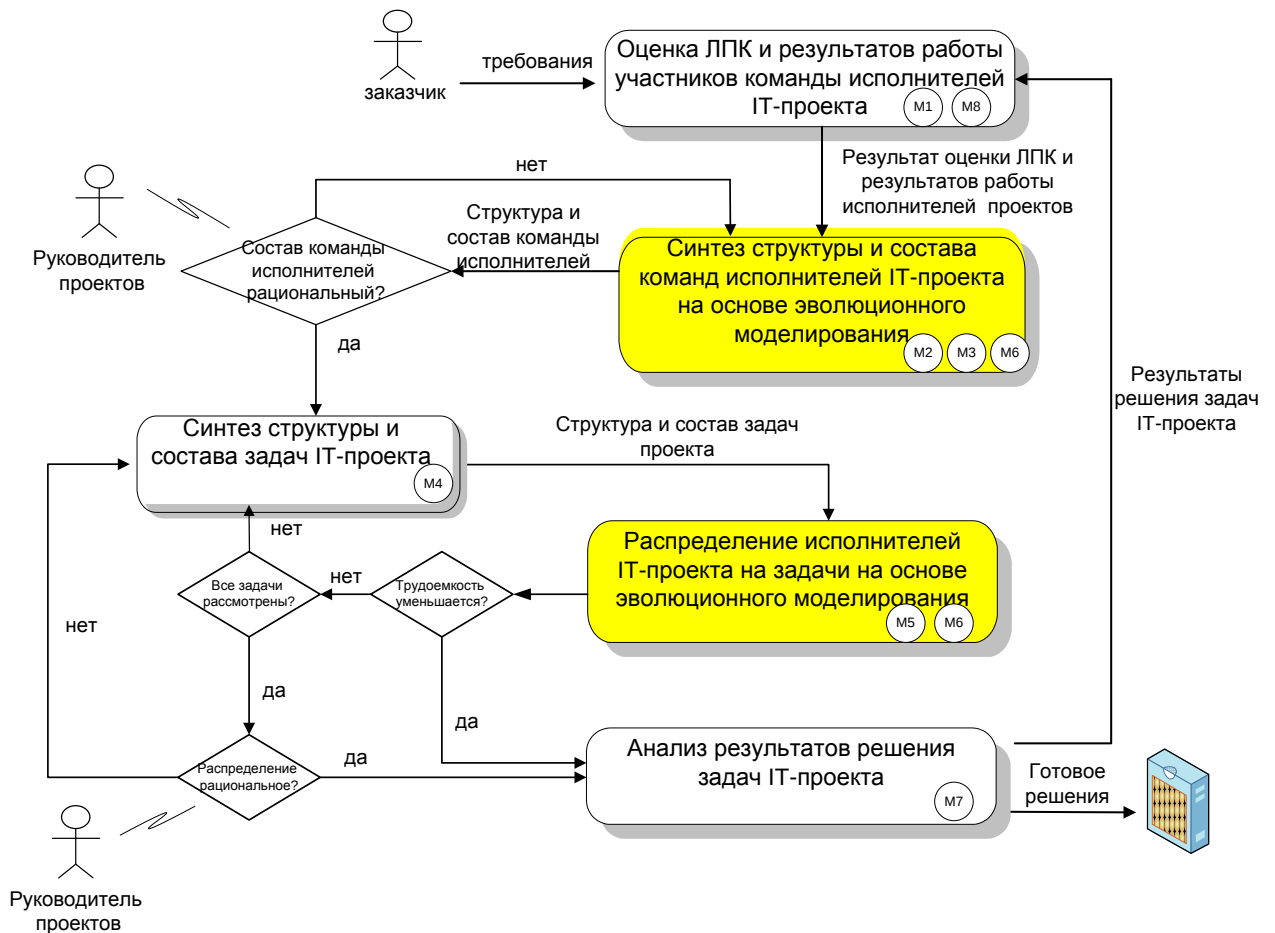


Рисунок 3.2 – Технология управления организационными процессами ИТ-проектов с использованием разработанного программного комплекса

## ВЫВОДЫ

В программном комплексе реализованы основные этапы авторского метода повышения эффективности управления организационными системами IT-проектов и авторские алгоритмы эволюционного моделирования в сочетании с методикой СОСОМО II и методом парных сравнений для поддержки принятия решений руководителем проектов на этапе формирования команд исполнителей и распределения исполнителей на задачи.

Представлена архитектура программного комплекса многоуровневого управления организационными системами IT-проектов, обеспечивающая реализацию авторских алгоритмов поддержки принятия решений при формировании структуры и состава команд исполнителей, распределении исполнителей на задачи проекта.

Представлены основные варианты использования программного комплекса руководителем проектов и участниками команд исполнителей.

Представлена последовательность применения разработанного программного комплекса для повышения эффективности формирования команд исполнителей и распределения исполнителей на задачи.

Представлена технология управления организационными системами IT-проектов с использованием разработанного программного комплекса, обеспечивающая принятие управленческих решений при управлении организационными процессами IT-проектов в социально-экономических системах учреждений образования и предприятий.

## ГЛАВА 4. ТЕСТИРОВАНИЕ, АПРОБАЦИЯ И ВНЕДРЕНИЕ

### 4.1 Тестирование метода на учебном проекте

Для выполнения тестирования предложенного метода повышения эффективности управления организационными системами IT-проектов выбран учебный проект для организации управления процессом обучения IT-специалистов.

Целью проекта является повышение эффективности подготовки IT-специалистов за счет автоматизации основных процессов обучения.

Для достижения поставленной цели решены следующие задачи:

- автоматизация построения индивидуальных образовательных траекторий IT-специалистов;
- автоматизация построения командных образовательных траекторий IT-специалистов;
- автоматизация компиляции учебных задач;
- автоматизация функций обучающегося: реализация возможности просмотра учебных курсов, последовательного прохождения учебных курсов;
- учет и анализ результативности обучения;
- формирование рейтинга обучающегося;
- автоматизация функций преподавателя: создание учебных курсов, создание учебных задач, создание контрольных работ и тестовых заданий.

Участники проекта – 10 студентов специальности «Автоматизированные системы обработки информации».

По результатам получено свидетельство о регистрации компьютерных программ в Роспатенте, участниками проекта опубликовано три статьи в журналах из перечня ВАК РФ [13, 14, 109], три тезиса докладов [110, 111, 112]. Результаты работы внедрены в образовательный процесс Белорусско-Российского университета.

На участие в проекте претендовали 50 студентов.

Для определения индивидуальной предрасположенности к конфликтному взаимодействию участники проектов прошли тест Т. Кеннета, для определения командной роли – тест Р. М. Белбина.

Данные психологического тестирования участников проектов представлены в таблице 4.1.

Таблица 4.1 – Данные психологического тестирования

| Участ-<br>ник | Результат теста<br>Кеннета | Результат теста<br>Белбина | Участ-<br>ник | Результат теста<br>Кеннета | Результат теста<br>Белбина |
|---------------|----------------------------|----------------------------|---------------|----------------------------|----------------------------|
| 1             | соперничество              | исполнитель                | 26            | приспособление             | разведчик                  |
| 2             | сотрудничество             | Мыслитель                  | 27            | компромисс                 | формирователь              |
| 3             | компромисс                 | Оценщик                    | 28            | компромисс                 | исполнитель                |
| 4             | соперничество              | координатор                | 29            | соперничество              | исполнитель                |
| 5             | приспособление             | координатор                | 30            | сотрудничество             | формирователь              |
| 6             | компромисс                 | Разведчик                  | 31            | сотрудничество             | координатор                |
| 7             | избегание                  | исполнитель                | 32            | компромисс                 | мыслитель                  |
| 8             | приспособление             | исполнитель                | 33            | соперничество              | исполнитель                |
| 9             | сотрудничество             | исполнитель                | 34            | соперничество              | исполнитель                |
| 10            | соперничество              | Мыслитель                  | 35            | избегание                  | разведчик                  |
| 11            | соперничество              | Мыслитель                  | 36            | сотрудничество             | разведчик                  |
| 12            | соперничество              | Мыслитель                  | 37            | компромисс                 | исполнитель                |
| 13            | соперничество              | председатель               | 38            | приспособление             | координатор                |
| 14            | компромисс                 | формирователь              | 39            | избегание                  | исполнитель                |
| 15            | соперничество              | председатель               | 40            | избегание                  | исполнитель                |
| 16            | компромисс                 | Разведчик                  | 41            | сотрудничество             | разведчик                  |
| 17            | сотрудничество             | координатор                | 42            | соперничество              | координатор                |
| 18            | соперничество              | председатель               | 43            | избегание                  | мыслитель                  |
| 19            | сотрудничество             | председатель               | 44            | соперничество              | оценщик                    |
| 20            | избегание                  | исполнитель                | 45            | сотрудничество             | оценщик                    |
| 21            | компромисс                 | исполнитель                | 46            | соперничество              | мыслитель                  |
| 22            | соперничество              | председатель               | 47            | соперничество              | координатор                |
| 23            | сотрудничество             | исполнитель                | 48            | приспособление             | формирователь              |
| 24            | соперничество              | председатель               | 49            | компромисс                 | исполнитель                |
| 25            | компромисс                 | исполнитель                | 50            | сотрудничество             | председатель               |

На основе матрицы парных сравнений Т. Саати руководителем проекта оценены модели поведения по Т. Кеннету и командные роли по М. Р. Белбину с точки зрения важности для успешной реализации проекта. Используя полученную матрицу весов результаты тестов переведены в количественную форму (таблица 4.2).

Для ускорения работы эволюционного моделирования за счет сокращения количества итераций, участники проектов по результатам психологического тестирования разделены на пять групп с использованием метода кластерного анализа k-средних. Результат представлен в таблице 4.2.

Таблица 4.2 – Результаты кластерного анализа

| Участ-<br>ник | Результат<br>теста<br>Кеннета | Результат<br>теста<br>Белбина | Номер<br>кластера | Участ-<br>ник | Результат<br>теста<br>Кеннета | Результат<br>теста<br>Белбина | Номер<br>кластера |
|---------------|-------------------------------|-------------------------------|-------------------|---------------|-------------------------------|-------------------------------|-------------------|
| 1             | 0,05                          | 0,08                          | 4                 | 26            | 0,09                          | 0,11                          | 5                 |
| 2             | 0,34                          | 0,14                          | 2                 | 27            | 0,36                          | 0,2                           | 3                 |
| 3             | 0,36                          | 0,05                          | 3                 | 28            | 0,36                          | 0,08                          | 3                 |
| 4             | 0,05                          | 0,05                          | 4                 | 29            | 0,05                          | 0,08                          | 4                 |
| 5             | 0,09                          | 0,05                          | 5                 | 30            | 0,34                          | 0,2                           | 2                 |
| 6             | 0,36                          | 0,11                          | 3                 | 31            | 0,34                          | 0,05                          | 2                 |
| 7             | 0,15                          | 0,08                          | 5                 | 32            | 0,36                          | 0,14                          | 3                 |
| 8             | 0,09                          | 0,08                          | 5                 | 33            | 0,05                          | 0,08                          | 4                 |
| 9             | 0,34                          | 0,08                          | 2                 | 34            | 0,05                          | 0,08                          | 4                 |
| 10            | 0,05                          | 0,14                          | 1                 | 35            | 0,15                          | 0,11                          | 3                 |
| 11            | 0,05                          | 0,14                          | 1                 | 36            | 0,34                          | 0,11                          | 2                 |
| 12            | 0,05                          | 0,14                          | 1                 | 37            | 0,36                          | 0,08                          | 3                 |
| 13            | 0,05                          | 0,27                          | 1                 | 38            | 0,09                          | 0,05                          | 5                 |
| 14            | 0,36                          | 0,2                           | 3                 | 39            | 0,15                          | 0,08                          | 5                 |
| 15            | 0,05                          | 0,27                          | 1                 | 40            | 0,15                          | 0,08                          | 5                 |
| 16            | 0,36                          | 0,11                          | 3                 | 41            | 0,34                          | 0,11                          | 2                 |
| 17            | 0,34                          | 0,05                          | 2                 | 42            | 0,05                          | 0,05                          | 4                 |
| 18            | 0,05                          | 0,27                          | 1                 | 43            | 0,15                          | 0,14                          | 2                 |
| 19            | 0,34                          | 0,27                          | 2                 | 44            | 0,05                          | 0,05                          | 4                 |
| 20            | 0,15                          | 0,08                          | 5                 | 45            | 0,34                          | 0,05                          | 2                 |
| 21            | 0,36                          | 0,08                          | 3                 | 46            | 0,05                          | 0,14                          | 1                 |
| 22            | 0,05                          | 0,27                          | 1                 | 47            | 0,05                          | 0,05                          | 4                 |
| 23            | 0,34                          | 0,08                          | 2                 | 48            | 0,09                          | 0,2                           | 5                 |
| 24            | 0,05                          | 0,27                          | 1                 | 49            | 0,36                          | 0,08                          | 3                 |
| 25            | 0,36                          | 0,08                          | 3                 | 50            | 0,34                          | 0,27                          | 2                 |

Методом парных сравнений Т. Саати оценены уровни параметров проекта и определены значения поправочных коэффициентов проекта по методике COSOMO II. Результаты представленные в таблице 4.3.

Таблица 4.3 – Поправочные коэффициенты проекта

| Параметр                                | Уровень    | Значение |
|-----------------------------------------|------------|----------|
| количество строк программного кода      |            | 130000   |
| гибкость процесса разработки            | Средний    | 3,04     |
| архитектура и разрешение рисков         | Средний    | 4,24     |
| зрелость процессов                      | Средний    | 4,68     |
| сложность и надежность продукта         | нормальный | 1,00     |
| разработка для повторного использования | нормальный | 1,00     |
| сложность платформы разработки          | нормальный | 1,00     |
| Оборудование                            | нормальный | 1,00     |

Руководителем проекта первоначально сформированы команды исполнителей, представленные в таблице 4.4.

Таблица 4.4 – Составы команд, сформированные руководителем проекта

| участники \ номер команды | 1  | 2  | 3  | 4  | 5  | 6  |
|---------------------------|----|----|----|----|----|----|
| 1                         | 23 | 21 | 26 | 6  | 6  | 25 |
| 2                         | 17 | 3  | 46 | 39 | 17 | 33 |
| 3                         | 18 | 9  | 27 | 19 | 29 | 35 |
| 4                         | 24 | 7  | 34 | 14 | 31 | 5  |
| 5                         | 49 | 49 | 6  | 12 | 47 | 24 |

На основании анализа результатов качества решения проектных задач, данных из таблиц 2.2-2.4 и расчетов по формулам 2.1-2.2 получены значения поправочных коэффициентов команды разработчиков методики СОСОМО II и трудоемкость выполнения проекта (таблица 4.5).

Таблица 4.5 – Поправочные коэффициенты команды разработчиков

| номер команды                        | 1   | 2   | 3   | 4   | 5   | 6   |
|--------------------------------------|-----|-----|-----|-----|-----|-----|
| квалификация персонала               | 4   | 4   | 3   | 4   | 4   | 4   |
| опыт персонала                       | 4   | 4   | 3   | 4   | 3   | 4   |
| сработанность команды                | 1   | 1   | 4   | 4   | 1   | 4   |
| наличие опыта аналогичных разработок | 3   | 4   | 1   | 2   | 2   | 5   |
| трудоемкость проекта                 | 575 | 541 | 850 | 520 | 745 | 434 |

Таким образом, получены исходные данные для применения эволюционного моделирования.

В соответствии с предложенным методом повышения эффективности управления организационными системами ИТ-проектов, на основании данных таблиц 4.3, 4.5 и составов команд, сформированных руководителем проектов (таблица 4.4) получены результаты пяти итераций применения эволюционного моделирования, представленные в таблицах 4.6-4.10.

Таблица 4.6 – Результаты первой итерации эволюционной моделирования

| номер<br>команды<br>участники | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1                             | 23  | 21  | 26  | 6   | 6   | 25  | 6   | 19  | 2   | 25  |
| 2                             | 17  | 3   | 46  | 39  | 17  | 33  | 39  | 14  | 39  | 33  |
| 3                             | 18  | 9   | 27  | 19  | 29  | 35  | 35  | 12  | 19  | 35  |
| 4                             | 24  | 7   | 34  | 14  | 31  | 5   | 5   | 25  | 14  | 5   |
| 5                             | 49  | 49  | 6   | 12  | 47  | 24  | 24  | 33  | 12  | 23  |
| PERS                          | 4   | 4   | 3   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| PREX                          | 4   | 4   | 3   | 4   | 3   | 4   | 3   | 4   | 4   | 4   |
| TEAM                          | 1   | 1   | 4   | 4   | 1   | 4   | 4   | 4   | 4   | 4   |
| PREC                          | 3   | 4   | 1   | 2   | 2   | 5   | 2   | 4   | 2   | 5   |
| PM                            | 575 | 541 | 850 | 520 | 745 | 434 | 635 | 461 | 520 | 434 |

Таблица 4.7 – Результаты второй итерации эволюционного моделирования

| номер<br>команды<br>участники | 2   | 4   | 6   | 8   | 9   | 10  | 11  | 12  | 13  | 14  |
|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1                             | 21  | 6   | 25  | 19  | 2   | 25  | 25  | 19  | 25  | 19  |
| 2                             | 3   | 39  | 33  | 14  | 39  | 33  | 14  | 33  | 40  | 14  |
| 3                             | 9   | 19  | 35  | 12  | 19  | 35  | 12  | 35  | 35  | 12  |
| 4                             | 7   | 14  | 5   | 25  | 14  | 5   | 25  | 5   | 5   | 3   |
| 5                             | 49  | 12  | 24  | 33  | 12  | 23  | 33  | 23  | 23  | 33  |
| PERS                          | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 5   | 4   |
| PREX                          | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| TEAM                          | 1   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| PREC                          | 4   | 2   | 5   | 4   | 2   | 5   | 4   | 3   | 5   | 4   |
| PM                            | 542 | 521 | 435 | 462 | 521 | 435 | 462 | 490 | 361 | 462 |

Таблица 4.8 – Результаты третьей итерации эволюционного моделирования

| номер<br>команды<br>участники | 6   | 8   | 10  | 11  | 13  | 14  | 15  | 16  | 17  |
|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1                             | 25  | 19  | 25  | 25  | 25  | 19  | 25  | 25  | 25  |
| 2                             | 33  | 14  | 33  | 14  | 40  | 14  | 19  | 33  | 40  |
| 3                             | 35  | 12  | 35  | 12  | 35  | 12  | 35  | 4   | 35  |
| 4                             | 5   | 25  | 5   | 25  | 5   | 3   | 5   | 5   | 11  |
| 5                             | 24  | 33  | 23  | 33  | 23  | 33  | 24  | 23  | 23  |
| PERS                          | 4   | 4   | 4   | 4   | 5   | 4   | 4   | 4   | 5   |
| PREX                          | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 5   |
| TEAM                          | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| PREC                          | 5   | 4   | 5   | 4   | 5   | 4   | 5   | 5   | 4   |
| PM                            | 435 | 462 | 435 | 462 | 361 | 462 | 435 | 435 | 333 |

Таблица 4.9 – Результаты четвертой итерации эволюционной моделирования

| номер<br>команды<br>участники | 6   | 10  | 13  | 15  | 16  | 17  | 18  | 19  | 20  |
|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1                             | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 18  | 25  |
| 2                             | 33  | 33  | 40  | 19  | 33  | 40  | 40  | 40  | 33  |
| 3                             | 35  | 35  | 35  | 35  | 4   | 35  | 48  | 35  | 4   |
| 4                             | 5   | 5   | 5   | 5   | 5   | 11  | 5   | 11  | 28  |
| 5                             | 24  | 23  | 23  | 24  | 23  | 23  | 23  | 23  | 23  |
| PERS                          | 4   | 4   | 5   | 4   | 4   | 5   | 5   | 5   | 4   |
| PREX                          | 4   | 4   | 4   | 4   | 4   | 5   | 4   | 5   | 4   |
| TEAM                          | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| PREC                          | 5   | 5   | 5   | 5   | 5   | 4   | 5   | 4   | 5   |
| PM.                           | 435 | 435 | 361 | 435 | 435 | 333 | 361 | 333 | 435 |

Таблица 4.10 – Результаты пятой итерации эволюционной моделирования

| номер<br>команды<br>участники | 6  | 10 | 16 | 17 | 19 | 20 | 21 | 22 | 23 |
|-------------------------------|----|----|----|----|----|----|----|----|----|
| 1                             | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| 1                             | 25 | 25 | 25 | 25 | 18 | 25 | 25 | 18 | 25 |
| 2                             | 33 | 33 | 33 | 40 | 40 | 33 | 40 | 27 | 40 |
| 3                             | 35 | 35 | 4  | 35 | 35 | 4  | 35 | 35 | 35 |
| 4                             | 5  | 5  | 5  | 11 | 11 | 28 | 34 | 11 | 5  |
| 5                             | 24 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 14 |

Продолжение таблицы 4.10

| 1    | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| PERS | 4   | 4   | 4   | 5   | 5   | 4   | 4   | 5   | 4   |
| PREX | 4   | 4   | 4   | 5   | 5   | 4   | 4   | 5   | 4   |
| TEAM | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| PREC | 5   | 5   | 5   | 4   | 4   | 5   | 3   | 4   | 4   |
| PM   | 435 | 435 | 435 | 333 | 333 | 435 | 490 | 333 | 462 |

Так как после пятой итерации трудоемкость выполнения проекта не изменилась, эволюционное моделирование завершило работу.

По результатам применения эволюционного моделирования сформированы три команды с наименьшей трудоемкостью выполнения проекта.

Для выполнения проекта руководителем выбрана команда исполнителей №17.

Сокращение трудоемкости в результате применения эволюционного моделирования составило 23% по сравнению с лучшим решением руководителя проекта №6.

На первом этапе планирования работ в результате декомпозиции выделено пять задач, для реализации которых используется три технологии. Путем сравнения с аналогичными завершенными задачами руководителем проекта оценено количество строк программного кода.

Руководителем проекта первоначально сформированы варианты распределения исполнителей на задачи, представленные в таблице 4.11.

Таблица 4.11 – Вариант распределения исполнителей на задачи  
руководителем проекта

| номер задачи                                | 1  | 2  | 3  | 4  | 5  |
|---------------------------------------------|----|----|----|----|----|
| номер исполнителя варианта распределения №1 | 35 | 23 | 25 | 40 | 11 |
| номер исполнителя варианта распределения №2 | 25 | 40 | 35 | 11 | 23 |

В соответствии с предложенным методом повышения эффективности управления организационными системами IT-проектов, на основании данных

таблицы 4.3, состава команды №17 (таблица 4.4) получены результаты трех итераций применения эволюционного моделирования для распределения участников команды исполнителей на задачи, представленные в таблицах 4.12-4.14.

На первой итерации вариант распределения №3 получен операцией скрещивания вариантов распределения №1 и №2, №4 и №5 – операцией мутации вариантов распределения соответственно №1 и №2. Операция сокращения не использовалась, так как максимальный размер популяции задан равным 5.

Таблица 4.12 – Результаты первой итерации

| номер задачи \ номер распределения | 1    | 2    | 3    | 4    | 5    |
|------------------------------------|------|------|------|------|------|
| 1                                  | 25   | 35   | 35   | 23   | 35   |
| 2                                  | 40   | 23   | 40   | 40   | 11   |
| 3                                  | 35   | 25   | 25   | 35   | 25   |
| 4                                  | 11   | 40   | 11   | 11   | 40   |
| 5                                  | 23   | 11   | 23   | 25   | 23   |
| PERS                               | 6    | 6    | 6    | 6    | 6    |
| PREX                               | 5    | 5    | 4    | 6    | 3    |
| PM                                 | 0,44 | 0,44 | 0,51 | 0,38 | 0,62 |

Результаты второй итерации представлены в таблице 4.13.

Таблица 4.13 – Результаты второй итерации

| номер задачи \ номер распределения | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
|------------------------------------|------|------|------|------|------|------|------|
| 1                                  | 25   | 35   | 35   | 23   | 35   | 23   | 11   |
| 2                                  | 40   | 23   | 40   | 40   | 11   | 25   | 23   |
| 3                                  | 35   | 25   | 25   | 35   | 25   | 35   | 25   |
| 4                                  | 11   | 40   | 11   | 11   | 40   | 11   | 40   |
| 5                                  | 23   | 11   | 23   | 25   | 23   | 40   | 35   |
| PERS                               | 6    | 6    | 6    | 6    | 6    | 6    | 6    |
| PREX                               | 5    | 5    | 4    | 6    | 3    | 6    | 5    |
| PM                                 | 0,44 | 0,44 | 0,51 | 0,38 | 0,62 | 0,38 | 0,44 |

Результат операции сокращения промежуточной популяции представлен в таблице 4.14.

Таблица 4.14 – Результат второй итерации

| номер задачи \ номер распределения | 1    | 2    | 4    | 6    | 7    |
|------------------------------------|------|------|------|------|------|
| 1                                  | 25   | 35   | 23   | 23   | 11   |
| 2                                  | 40   | 23   | 40   | 25   | 23   |
| 3                                  | 35   | 25   | 35   | 35   | 25   |
| 4                                  | 11   | 40   | 11   | 11   | 40   |
| 5                                  | 23   | 11   | 25   | 40   | 35   |
| PERS                               | 6    | 6    | 6    | 6    | 6    |
| PREX                               | 5    | 5    | 6    | 6    | 5    |
| PM                                 | 0,44 | 0,44 | 0,38 | 0,38 | 0,44 |

Результаты третьей итерации эволюционного моделирования представлены в таблицах 4.13-4.14.

Таблица 4.13 – Результат применения операции мутации на третьей итерации

| номер задачи \ номер распределения | 1    | 2    | 4    | 6    | 7    | 8    | 9    |
|------------------------------------|------|------|------|------|------|------|------|
| 1                                  | 25   | 35   | 23   | 23   | 11   | 23   | 40   |
| 2                                  | 40   | 23   | 40   | 25   | 23   | 40   | 25   |
| 3                                  | 35   | 25   | 35   | 35   | 25   | 35   | 35   |
| 4                                  | 11   | 40   | 11   | 11   | 40   | 25   | 11   |
| 5                                  | 23   | 11   | 25   | 40   | 35   | 11   | 23   |
| PERS                               | 6    | 6    | 6    | 6    | 6    | 6    | 6    |
| PREX                               | 5    | 5    | 6    | 6    | 5    | 6    | 6    |
| PM                                 | 0,44 | 0,44 | 0,38 | 0,38 | 0,44 | 0,38 | 0,38 |

Таблица 4.13 – Результат третьей итерации

| номер задачи \ номер распределения | 2    | 4    | 6    | 7    | 8    | 9    |
|------------------------------------|------|------|------|------|------|------|
| 1                                  | 35   | 23   | 23   | 11   | 23   | 40   |
| 2                                  | 23   | 40   | 25   | 23   | 40   | 25   |
| 3                                  | 25   | 35   | 35   | 25   | 35   | 35   |
| 4                                  | 40   | 11   | 11   | 40   | 25   | 11   |
| 5                                  | 11   | 25   | 40   | 35   | 11   | 23   |
| PERS                               | 6    | 6    | 6    | 6    | 6    | 6    |
| PREX                               | 5    | 6    | 6    | 5    | 6    | 6    |
| PM.                                | 0,44 | 0,38 | 0,38 | 0,44 | 0,38 | 0,38 |

Так как наименьшее значение трудоемкости не улучшилось, эволюционное моделирование закончило работу.

Руководителем проекта выбран вариант распределения №6.

Сокращение трудоемкости решения задач в результате применения эволюционного моделирования составило 13% по сравнению с решением руководителя проекта.

#### **4.2 Тестирование метода при создании коммерческого программного продукта**

Для выполнения тестирования предложенного метода повышения эффективности управления организационными системами ИТ-проектов в ООО «Стэпл Инк» выбран ИТ-проект «Теплосеть Controlling». Цель создания системы – повышение эффективности процессов контроля ведения документации при проектировании, монтаже и эксплуатации теплотрасс посредством автоматизации учета.

Задачи, которые необходимо решить для достижения цели:

– создание базы данных организаций, осуществляющих поставку материалов, проектирование, монтаж и эксплуатацию теплосетей;

– создание автоматизированной информационной системы хранения документации по каждому объекту с возможностью формирования журналов, актов и протоколов в электронном виде;

– автоматизация расчета показателей эффективности работы предприятий теплоэнергетического комплекса РБ.

На участие в проекте претендовали 50 разработчиков компании.

Для определения индивидуальной предрасположенности к конфликтному взаимодействию участники проектов прошли тест Т. Кеннета, для определения командной роли – тест Р. М. Белбина.

Данные психологического тестирования участников проектов представлены в таблице 4.14.

Методом парных сравнений Т. Саати результаты тестов переведены в количественную форму. В зависимости от ситуации, складывающейся в проекте, руководителем проекта выбираются критерии сравнения стилей поведения и командных ролей.

Таблица 4.14 – Данные психологического тестирования

| Участник | Результат теста Кеннета | Результат теста Белбина | Участник | Результат теста Кеннета | Результат теста Белбина |
|----------|-------------------------|-------------------------|----------|-------------------------|-------------------------|
| 1        | 2                       | 3                       | 4        | 5                       | 6                       |
| 1        | сотрудничество          | Формирователь           | 26       | избегание               | разведчик               |
| 2        | приспособление          | Формирователь           | 27       | сотрудничество          | координатор             |
| 3        | избегание               | Исследователь           | 28       | компромисс              | исследователь           |
| 4        | сотрудничество          | Формирователь           | 29       | соперничество           | председатель            |
| 5        | приспособление          | Мыслитель               | 30       | избегание               | координатор             |
| 6        | сотрудничество          | Разведчик               | 31       | приспособление          | исследователь           |
| 7        | соперничество           | Оценщик                 | 32       | соперничество           | оценщик                 |
| 8        | приспособление          | Мыслитель               | 33       | сотрудничество          | координатор             |
| 9        | избегание               | формирователь           | 34       | сотрудничество          | исследователь           |
| 10       | сотрудничество          | Разведчик               | 35       | компромисс              | исследователь           |
| 11       | сотрудничество          | Мыслитель               | 36       | соперничество           | формирователь           |
| 12       | соперничество           | исследователь           | 37       | компромисс              | формирователь           |
| 13       | избегание               | Разведчик               | 38       | избегание               | оценщик                 |
| 14       | компромисс              | исследователь           | 39       | соперничество           | председатель            |
| 15       | соперничество           | формирователь           | 40       | избегание               | исследователь           |
| 16       | сотрудничество          | координатор             | 41       | сотрудничество          | исследователь           |
| 17       | соперничество           | Разведчик               | 42       | сотрудничество          | исследователь           |
| 18       | соперничество           | Разведчик               | 43       | соперничество           | формирователь           |

Продолжение таблицы 4.14

| 1  | 2              | 3             | 4  | 5              | 6             |
|----|----------------|---------------|----|----------------|---------------|
| 19 | соперничество  | исследователь | 44 | компромисс     | координатор   |
| 20 | компромисс     | исследователь | 45 | избегание      | исследователь |
| 21 | сотрудничество | Разведчик     | 46 | приспособление | оценщик       |
| 22 | избегание      | председатель  | 47 | соперничество  | оценщик       |
| 23 | соперничество  | исследователь | 48 | приспособление | разведчик     |
| 24 | компромисс     | Мыслитель     | 49 | приспособление | мыслитель     |
| 25 | соперничество  | председатель  | 50 | компромисс     | председатель  |

На основе матрицы парных сравнений Т. Саати руководителем проекта оценены модели поведения по Т. Кеннету и командные роли по М. Р. Белбину с точки зрения важности для успешной реализации проекта. Используя полученную матрицу весов результаты тестов переведены в количественную форму (таблица 4.15).

Для ускорения работы эволюционного моделирования за счет сокращения количества итераций, участники проектов по результатам психологического тестирования разделены на пять групп с использованием метода кластерного анализа к-средних. Результат представлен в таблице 4.15.

Таблица 4.15 – Результаты кластерного анализа

| Участ-<br>ник | Результат<br>теста<br>Кеннета | Результат<br>т теста<br>Белбина | Номер<br>кластера | Участ-<br>ник | Результат<br>теста<br>Кеннета | Результат<br>т теста<br>Белбина | Номер<br>кластера |
|---------------|-------------------------------|---------------------------------|-------------------|---------------|-------------------------------|---------------------------------|-------------------|
| 1             | 2                             | 3                               | 4                 | 5             | 6                             | 7                               | 8                 |
| 1             | 0,41                          | 0,19                            | 5                 | 26            | 0,16                          | 0,09                            | 3                 |
| 2             | 0,1                           | 0,19                            | 2                 | 27            | 0,41                          | 0,04                            | 5                 |
| 3             | 0,16                          | 0,09                            | 3                 | 28            | 0,29                          | 0,09                            | 5                 |
| 4             | 0,41                          | 0,19                            | 5                 | 29            | 0,04                          | 0,3                             | 1                 |
| 5             | 0,1                           | 0,17                            | 2                 | 30            | 0,16                          | 0,04                            | 3                 |
| 6             | 0,41                          | 0,09                            | 5                 | 31            | 0,1                           | 0,09                            | 4                 |
| 7             | 0,04                          | 0,06                            | 1                 | 32            | 0,04                          | 0,06                            | 1                 |
| 8             | 0,1                           | 0,17                            | 2                 | 33            | 0,41                          | 0,04                            | 5                 |
| 9             | 0,16                          | 0,19                            | 3                 | 34            | 0,41                          | 0,09                            | 5                 |
| 10            | 0,41                          | 0,09                            | 5                 | 35            | 0,29                          | 0,09                            | 5                 |
| 11            | 0,41                          | 0,17                            | 5                 | 36            | 0,04                          | 0,19                            | 1                 |
| 12            | 0,04                          | 0,09                            | 1                 | 37            | 0,29                          | 0,19                            | 2                 |
| 13            | 0,16                          | 0,09                            | 3                 | 38            | 0,16                          | 0,06                            | 3                 |
| 14            | 0,29                          | 0,09                            | 5                 | 39            | 0,04                          | 0,3                             | 1                 |
| 15            | 0,04                          | 0,19                            | 1                 | 40            | 0,16                          | 0,09                            | 3                 |

Продолжение таблицы 4.15

| 1  | 2    | 3    | 4 | 5  | 6    | 7    | 8 |
|----|------|------|---|----|------|------|---|
| 16 | 0,41 | 0,04 | 5 | 41 | 0,41 | 0,09 | 5 |
| 17 | 0,04 | 0,09 | 1 | 42 | 0,41 | 0,09 | 5 |
| 18 | 0,04 | 0,09 | 1 | 43 | 0,04 | 0,19 | 1 |
| 19 | 0,04 | 0,09 | 1 | 44 | 0,29 | 0,04 | 4 |
| 20 | 0,29 | 0,09 | 5 | 45 | 0,16 | 0,09 | 3 |
| 21 | 0,41 | 0,09 | 5 | 46 | 0,1  | 0,06 | 4 |
| 22 | 0,16 | 0,3  | 3 | 47 | 0,04 | 0,06 | 1 |
| 23 | 0,04 | 0,09 | 1 | 48 | 0,1  | 0,09 | 4 |
| 24 | 0,29 | 0,17 | 1 | 49 | 0,1  | 0,17 | 2 |
| 25 | 0,04 | 0,3  | 1 | 50 | 0,29 | 0,3  | 2 |

Методом парных сравнений Т. Саати получены оценки уровней параметров проекта и значения поправочных коэффициентов проекта по методике COCOMO II. Результаты представленные в таблице 4.16.

Таблица 4.16 – Поправочные коэффициенты проекта

| Параметр                                | Уровень    | Значение  |
|-----------------------------------------|------------|-----------|
| количество строк программного кода      |            | 1 000 000 |
| гибкость процесса разработки            | высокий    | 2,03      |
| архитектура и разрешение рисков         | средний    | 4,24      |
| зрелость процессов                      | высокий    | 3,12      |
| сложность и надежность продукта         | высокий    | 1,33      |
| разработка для повторного использования | нормальный | 1,00      |
| сложность платформы разработки          | нормальный | 1,00      |
| Оборудование                            | высокий    | 0,87      |

Руководителем проекта первоначально сформированы команды исполнителей, представленные в таблице 4.17.

Таблица 4.17 – Составы команд, сформированные руководителем проекта

| номер команды \ участники | 1  | 2  | 3  | 4  | 5  | 6  |
|---------------------------|----|----|----|----|----|----|
| 1                         | 39 | 4  | 20 | 30 | 10 | 14 |
| 2                         | 36 | 21 | 7  | 25 | 31 | 13 |
| 3                         | 29 | 10 | 31 | 46 | 14 | 30 |
| 4                         | 23 | 48 | 26 | 21 | 30 | 23 |
| 5                         | 5  | 44 | 21 | 7  | 44 | 9  |

На основании анализа результатов качества решения проектных задач, данных из таблиц 2.2-2.4 и расчетов по формулам 2.1-2.2 получены значения поправочных коэффициентов команд разработчиков методики СОСОМО II и трудоемкость выполнения проекта, представленные в таблице 4.18.

Таблица 4.18 – Поправочные коэффициенты команд разработчиков

| номер команды                        | 1    | 2    | 3    | 4    | 5    | 6    |
|--------------------------------------|------|------|------|------|------|------|
| квалификация персонала               | 4    | 4    | 4    | 4    | 4    | 4    |
| опыт персонала                       | 4    | 4    | 4    | 4    | 4    | 4    |
| сработанность команды                | 2    | 4    | 3    | 3    | 3    | 3    |
| наличие опыта аналогичных разработок | 1    | 3    | 3    | 5    | 4    | 4    |
| трудоемкость проекта                 | 2631 | 1968 | 2107 | 1806 | 1951 | 1951 |

Таким образом, получены исходные данные для применения эволюционного моделирования.

В соответствии с предложенным методом повышения эффективности управления организационными системами IT-проектов, на основании данных таблиц 4.16, 4.18 и составов команд, сформированных руководителем проектов (таблица 4.17) получены результаты пяти итераций применения эволюционного моделирования, представленные в таблицах 4.19-4.23.

В результате селективного отбора отобраны команды №4 и №5. В результате применения операции скрещивания к ним получены команды №7 и №8, при применении операции мутации получены команды №9 и №10.

В результате сокращения промежуточной популяции следующее поколение составят команды №2, №4, №5, №6, №7, №9.



Так как трудоемкость не изменилась, эволюционное моделирование завершило работу.

По результатам применения эволюционного моделирования сформированы восемь команд с наименьшей трудоемкостью выполнения проекта.

Сокращение трудоемкости в результате применения эволюционного моделирования составило 26% по сравнению с лучшим решением руководителя проекта.

Для реализации проекта руководителем выбрана команда №14.

На первом этапе планирования работ в результате декомпозиции выделено пять задач, для реализации которых используется три технологии. Путем сравнения задач с выполненными ранее, руководителем проекта определено количество строк программного кода.

Руководителем проекта первоначально сформированы варианты распределения исполнителей на задачи, представленные в таблице 4.21.

4.21 – Вариант распределения исполнителей на задачи  
руководителем проекта

| номер задачи                                | 1  | 2  | 3  | 4  | 5  |
|---------------------------------------------|----|----|----|----|----|
| номер исполнителя варианта распределения №1 | 25 | 16 | 40 | 46 | 21 |
| номер исполнителя варианта распределения №2 | 25 | 40 | 46 | 16 | 21 |

В соответствии с предложенным методом повышения эффективности управления организационными системами ИТ-проектов, на основании данных таблицы 4.16, состава команды №14 (таблица 4.20) получены результаты трех итераций применения эволюционного моделирования для распределения исполнителей на задачи, представленные в таблицах 4.22-4.24.

На первой итерации вариант распределения №3 получен операцией скрещивания вариантов распределения №1 и №2, №4 и №5 – операцией мутации вариантов распределения №1 и №2 соответственно. Операция

сокращения не использовалась, так как максимальный размер популяции задан равным 5.

Таблица 4.22 – Результаты первой итерации

| номер задачи \ номер распределения | 1  | 2  | 3  | 4  | 5  |
|------------------------------------|----|----|----|----|----|
| 1                                  | 25 | 21 | 25 | 21 | 40 |
| 2                                  | 16 | 40 | 16 | 16 | 25 |
| 3                                  | 40 | 46 | 40 | 40 | 46 |
| 4                                  | 46 | 16 | 46 | 46 | 16 |
| 5                                  | 21 | 25 | 21 | 25 | 21 |
| PERS                               | 5  | 5  | 5  | 5  | 5  |
| PREX                               | 4  | 5  | 5  | 6  | 4  |
| PM                                 | 51 | 45 | 45 | 38 | 51 |

Таблица 4.23 – Результат применения операций скрещивания и мутации на второй итерации

| номер задачи \ номер распределения | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
|------------------------------------|----|----|----|----|----|----|----|
| 1                                  | 25 | 21 | 25 | 21 | 40 | 16 | 21 |
| 2                                  | 16 | 40 | 16 | 16 | 25 | 40 | 16 |
| 3                                  | 40 | 46 | 40 | 40 | 46 | 46 | 25 |
| 4                                  | 46 | 16 | 46 | 46 | 16 | 21 | 46 |
| 5                                  | 21 | 25 | 21 | 25 | 21 | 25 | 40 |
| PERS                               | 5  | 5  | 5  | 5  | 5  | 5  | 5  |
| PREX                               | 4  | 5  | 5  | 6  | 4  | 4  | 6  |
| PM                                 | 51 | 45 | 45 | 38 | 51 | 45 | 38 |

Результат операции сокращения промежуточной популяции представлен в таблице 4.24.

Таблица 4.24 – Результат второй итерации

| номер задачи \ номер распределения | 2  | 3  | 4  | 6  | 7  |
|------------------------------------|----|----|----|----|----|
| 1                                  | 21 | 25 | 21 | 16 | 21 |
| 2                                  | 40 | 16 | 16 | 40 | 16 |
| 3                                  | 46 | 40 | 40 | 46 | 25 |
| 4                                  | 16 | 46 | 46 | 21 | 46 |
| 5                                  | 25 | 21 | 25 | 25 | 40 |
| PERS                               | 5  | 5  | 5  | 5  | 5  |
| PREX                               | 5  | 5  | 6  | 4  | 6  |
| PM                                 | 45 | 45 | 38 | 45 | 38 |

Так как наименьшее значение трудоемкости не улучшилось, эволюционное моделирование закончило работу.

Руководителем проекта выбран вариант распределения №4.

Сокращение трудоемкости выполнения задач в результате применения эволюционного моделирования составило 26% по сравнению с решением руководителя проекта.

### 4.3 Тестирование программного обеспечения

Для управления организационными системами IT-проектов с использованием разработанного программного комплекса необходимо зарегистрировать в системе участников проектов. Данную работу осуществляет руководитель проектов.

Руководитель проектов вводит в систему данные о проектах, подлежащих реализации (рисунок 4.1).

**Создать новый проект**

Название проекта:\*

Автоматизированная система оценки работы сварщиков и качества сварочных работ

Описание:\*

ПО для повышения эффективности контроля работы сварщиков и качества сварочных работ

Размер команды:\*

4

Дата начала:

1/24/2020, 11:09 PM

Приоритет(1...7):\*

7

Дата окончания:

6/24/2020, 11:09 PM

Сложность(1...5):\*

5

Крайний срок голосования:

6/1/2020, 11:09 PM

Крайний срок формирования группы:

6/2/2020, 11:09 PM

Крайний срок создания плана:

6/10/2020, 11:09 PM

**СОХРАНИТЬ** **ОТМЕНА**

Рисунок 4.1 – Форма создания нового проекта

Для распределения участников проектов в команды исполнителей руководитель проектов создает одну или несколько команд исполнителей (рисунок 4.2).

**Создание новой проектной группы**

Имя проектной группы:\*

Крайний срок тестирования:

| <input type="checkbox"/> | Имя       | Фамилия     | Логин         | Почта                 | Краткий срок тестир... |
|--------------------------|-----------|-------------|---------------|-----------------------|------------------------|
| <input type="checkbox"/> | Виктор    | участник 47 | Victor_mar    | Victor_mar@yahoo...   | 2019-05-31T09:32...    |
| <input type="checkbox"/> | Юлия      | участник 6  | julia_ivanova | julia_ivanova@mai...  | 2019-05-31T09:32...    |
| <input type="checkbox"/> | Андрей    | участник 15 | andrew_am     | andrew_am@gmail...    | 2019-05-31T09:32...    |
| <input type="checkbox"/> | Александр | участник 29 | alex_petrov   | alexasha@gmail.com    | 2019-05-31T09:32...    |
| <input type="checkbox"/> | Василий   | участник 39 | vasya_petrov  | vasya_petrov@gmail... | 2019-05-31T09:32...    |
| <input type="checkbox"/> | Виктор    | участник 25 | Victor_mar    | Victor_mar@yahoo...   | 2019-05-31T09:32...    |

**СОЗДАТЬ НОВОГО УЧАСТНИКА** **СОХРАНИТЬ** **ОТМЕНА**

Рисунок 4.2 – Форма создания новой группы

В программном комплексе предусмотрена возможность просматривать созданные команды(рисунок 4.3), вручную вносить изменения в состав команд.

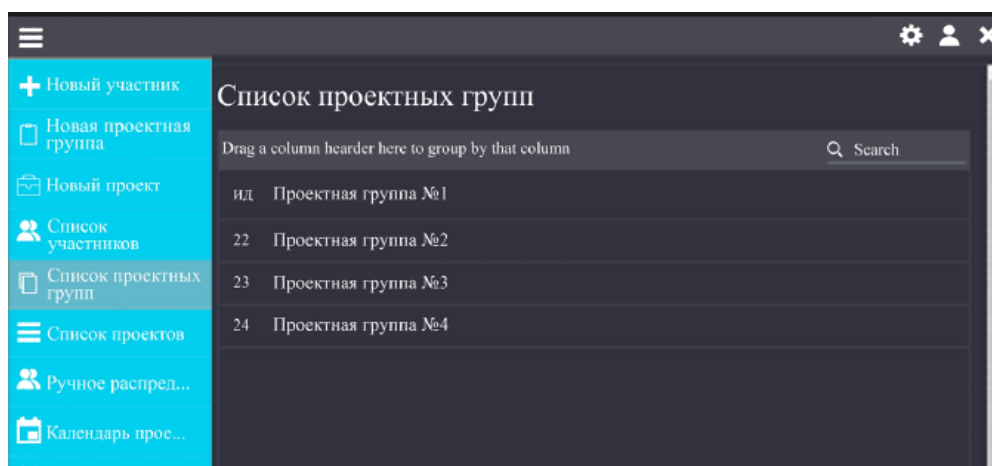


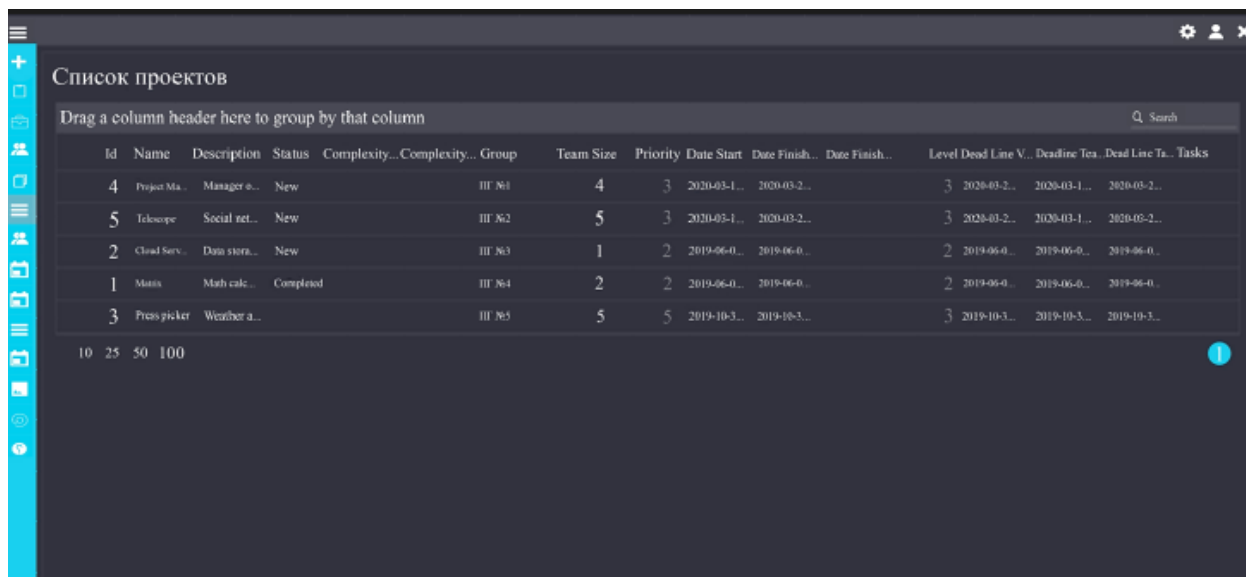
Рисунок 4.3 – Форма списка групп

Для анализа результатов тестирования личностных и психологических качеств участников проектов методом  $k$ -средних кластерного анализа предназначена форма, представленная на рисунке 4.4.

| Участники проектов |             |             |             |             |  |
|--------------------|-------------|-------------|-------------|-------------|--|
| Кластер №1         | Кластер №2  | Кластер №3  | Кластер №4  | Кластер №5  |  |
| участник 6         | участник 3  | участник 12 | участник 1  | участник 2  |  |
| участник 7         | участник 13 | участник 14 | участник 4  | участник 5  |  |
| участник 10        | участник 22 | участник 16 | участник 11 | участник 8  |  |
| участник 15        | участник 26 | участник 19 | участник 24 | участник 9  |  |
| участник 17        | участник 30 | участник 20 | участник 32 | участник 38 |  |
| участник 18        | участник 31 | участник 23 | участник 37 | участник 46 |  |
| участник 21        | участник 40 | участник 27 | участник 50 | участник 49 |  |
| участник 25        | участник 45 | участник 28 |             |             |  |
| участник 29        | участник 48 | участник 33 |             |             |  |
| участник 36        |             | участник 34 |             |             |  |
| участник 39        |             | участник 35 |             |             |  |
| участник 43        |             | участник 41 |             |             |  |
| участник 47        |             | участник 42 |             |             |  |
|                    |             | участник 44 |             |             |  |

Рисунок 4.4 – Форма с результатами кластерного анализа

Для управления формированием состава задач проекта предназначена форма 4.5. Данная форма поддерживает сортировку и поиск проектов. Двойным щелчком мыши на название проекта можно перейти в дополнительное окно для просмотра детальной информации.



Список проектов

Drag a column header here to group by that column

Search

| Id | Name          | Description   | Status    | Complexity... | Complexity... | Group  | Team Size | Priority | Date Start   | Date Finish... | Date Finish... | Level | Dead Line V... | Deadline Tea... | Dead Line Ta... | Tasks |
|----|---------------|---------------|-----------|---------------|---------------|--------|-----------|----------|--------------|----------------|----------------|-------|----------------|-----------------|-----------------|-------|
| 4  | Project Ma... | Manager o...  | New       |               |               | III №1 | 4         | 3        | 2020-03-1... | 2020-03-2...   |                | 3     | 2020-03-2...   | 2020-03-1...    | 2020-03-2...    |       |
| 5  | Telescope     | Social net... | New       |               |               | III №2 | 5         | 3        | 2020-03-1... | 2020-03-2...   |                | 3     | 2020-03-2...   | 2020-03-1...    | 2020-03-2...    |       |
| 2  | Cloud Serv... | Data stora... | New       |               |               | III №3 | 1         | 2        | 2019-06-0... | 2019-06-0...   |                | 2     | 2019-06-0...   | 2019-06-0...    | 2019-06-0...    |       |
| 1  | Matrix        | Math calc...  | Completed |               |               | III №4 | 2         | 2        | 2019-06-0... | 2019-06-0...   |                | 2     | 2019-06-0...   | 2019-06-0...    | 2019-06-0...    |       |
| 3  | Press picker  | Weather a...  |           |               |               | III №5 | 5         | 5        | 2019-10-3... | 2019-10-3...   |                | 3     | 2019-10-3...   | 2019-10-3...    | 2019-10-3...    |       |

10 25 50 100

Рисунок 4.5 – Форма работы с проектами

Используя меню из левой части экрана можно перейти на страницу синтеза структуры и состава задач проекта, который представлен на рисунке 4.6. Детальную информацию о задаче можно увидеть выполнив двойной клик на мыши на названии задачи. Форма календарного плана поддерживает контекстное меню в котором можно: создать новую задачу, просмотреть информацию о задаче, удалить задачу.

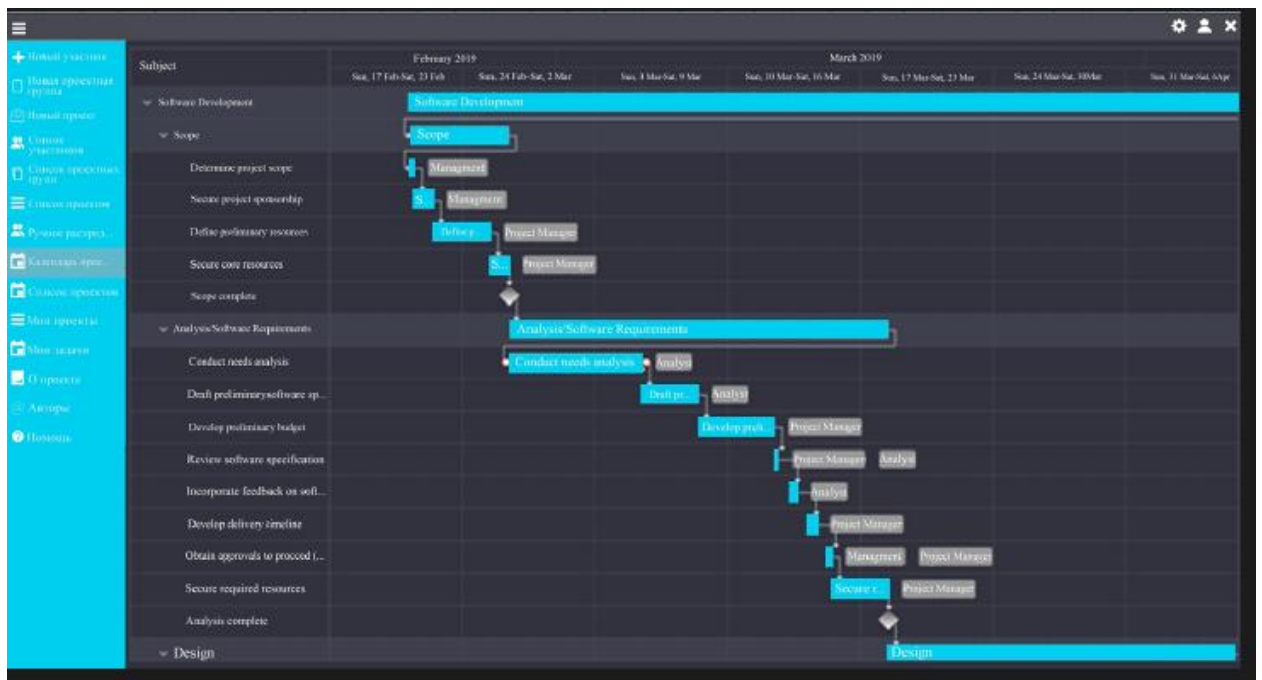


Рисунок 4.6 – Форма календарного плана проекта

## 4.4 Экспериментальная проверка результатов исследований

4.4.1 Параметры модели. Параметрами, которые исследуются в процессе эксперимента, являются:

- по результатам психологического тестирования:
  - стиль поведения по Т. Кеннету: соперничество, сотрудничество, компромисс, избегание, приспособление;
  - командная роль по Р.М. Белбину: формирователь, исследователь, мыслитель, разведчик, оценщик, координатор, председатель, миротворец, доводчик;
- по результатам решения проектных задач:
  - сложность задачи, задается числом от 0 до 1, определяется методом парных сравнений Т. Саати;
  - число найденных дефектов решения задачи, разработчиками оцениваются результаты решения задач системными архитекторами, разработчиков оценивают тестировщики, тестировщиков оценивают

системные аналитики, системных аналитиков – системные архитекторы, системных архитекторов – разработчики;

- плановое время решения задач проекта;
- фактическое время решения задач проекта;
- количество часов работы с данной технологией; определяется как накопительная сумма часов, на протяжении которых участник работал с данной технологией;
- наибольший процент участников команды, принимавших участие в разработке одного проекта;
- структура команды исполнителей; определяется руководителем проекта.

В процессе работы алгоритма формирования состава команды исполнителей на основе эволюционного моделирования определяются следующие значения:

- уровень владения технологиями командой, рассчитывается по формуле (2.6);
- скорость решения задачи проекта, рассчитывается по формуле (2.3);
- опыт работы с технологиями и инструментами участника проектов  $E_x$  определяется по правилу 10000 часов и выбирается из таблицы 2.2.
- объем программного кода проекта; определяется руководителем проектов на основе статистических данных;
- поправочные коэффициенты проекта методики COCOMO II: FLEX – гибкость процесса разработки, RESL – архитектура и разрешение рисков, PMAT – зрелость процессов, RCPX – сложность и надежность продукта, RUSE – разработка для повторного использования, PDIF – сложность платформы разработки, FCIL – оборудование; определяются методом парных сравнений Т. Саати, сравниваются уровни поправочных коэффициентов методики COCOMO II.

– поправочный коэффициент команды исполнителей PERS «Квалификация персонала» определяется как среднее значение участников команды исполнителей и выбирается из таблицы 2.3;

– поправочный коэффициент команды исполнителей TEAM «Сработанность команды» определяется как наибольший процент участников команды, принимавших участие в разработке одного проекта и выбирается по таблице 2.4;

– поправочный коэффициент команды исполнителей PREX «Опыт персонала» определяется как среднее значение участников команды исполнителей;

– поправочный коэффициент команды исполнителей PREC «Прецедентность, наличие опыта аналогичных разработок» определяется как среднее значение участников команды исполнителей;

#### 4.4.2 Последовательность проведения эксперимента

Шаг 1. Выгрузка данных психологического тестирования. Пример представлен в таблице 4.14.

Шаг 2. Кластерный анализ результатов. Методом парных сравнений Т. Саати результаты тестов переведены в количественную форму. Результаты применения кластерного анализа представлены на рисунке 4.4.

Шаг 3. Оценка размера программного кода и поправочных коэффициентов методики СОСОМО II методом парных сравнений Т. Саати. Результаты представлены на рисунке 4.7.

Шаг 4. Формирование структуры и состава команд исполнителей руководителем проекта. Формирование первоначального состава команд исполнителей представлено на пользовательской форме на рисунке 4.2.

| Параметр                                | Значение  | Уровень    |
|-----------------------------------------|-----------|------------|
| Количество строк программного кода      | 1 000 000 |            |
| гибкость процесса разработки            | 2,03      | высокий    |
| архитектура и разрешение рисков         | 4,24      | средний    |
| зрелость процессов                      | 3,12      | высокий    |
| сложность и надежность продукта         | 1,33      | высокий    |
| разработка для повторного использования | 1,00      | нормальный |
| сложность платформы разработки          | 1,00      | нормальный |
| Оборудование                            | 0,87      | высокий    |

Рисунок 4.7 – Результаты оценки поправочных коэффициентов методики COCOMO II

Шаг 5. Улучшение состава команды исполнителей на основе эволюционного моделирования. Результаты представлены на рисунке 4.8.

| Id | Username     | Name    | Surname    | Email        | Password   | Roles | Testing D... |
|----|--------------|---------|------------|--------------|------------|-------|--------------|
| 8  | julia_iva... | Юлия    | участик 18 | julia_iva... | adasdas    |       | 2019-05...   |
| 13 | andrew_...   | Андрей  | участик 10 | andrew_...   | adasdas    |       | 2019-05...   |
| 15 | vasya_p...   | Василий | участик 39 | vasya_p...   | adasdas    |       | 2019-05...   |
| 33 | Victor_m...  | Виктор  | участик 47 | Victor_m...  | \$2a\$10\$ |       | 2019-05...   |

Рисунок 4.8 – Результаты эволюционного моделирования

Шаг 6. Распределение исполнителей на задачи руководителем проектов.

Формирование первоначального распределения исполнителей на задачи представлено на пользовательской форме на рисунке 4.9.

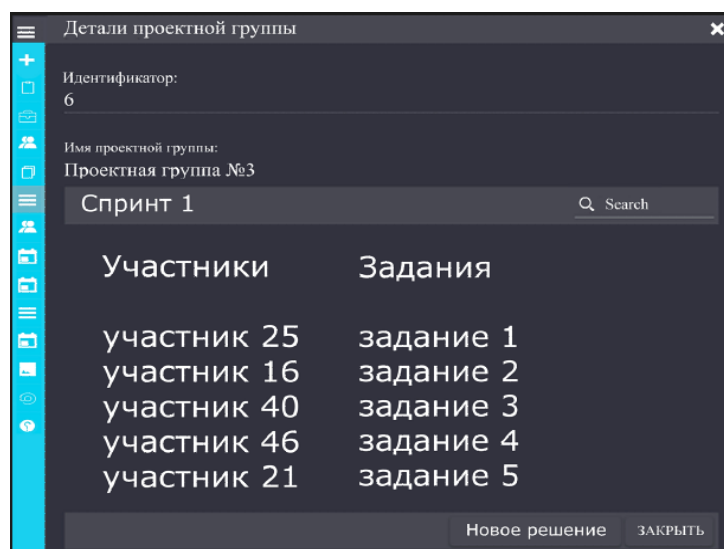


Рисунок 4.9 - Первоначальное распределение исполнителей на задачи

Шаг 7. Улучшение распределения исполнителей на задачи на основе эволюционного моделирования. Результат представлен на рисунке 4.10.

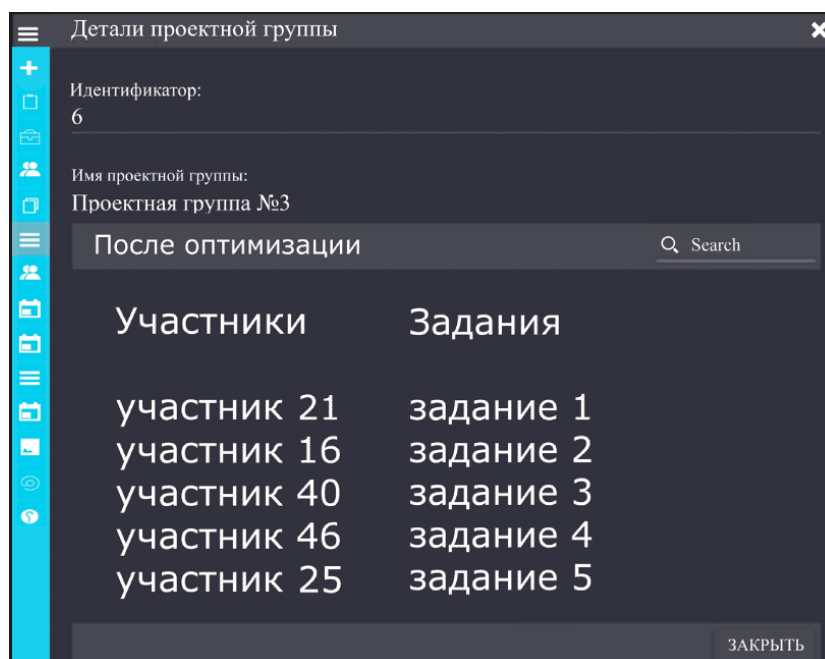


Рисунок 4.10 – Результаты эволюционного моделирования по распределению участников на задачи

#### 4.4.3 Результаты применения метода, средств и технологий в учебном процессе

В результате применения метода, средств и технологий, созданных в рамках диссертационной работы, студентами Белорусско-Российского университета реализованы следующие проекты:

- Программа для организации управления процессом обучения IT-специалистов [80]. Участники проекта – 10 студентов специальности «Автоматизированные системы обработки информации». По результатам получено свидетельство о регистрации компьютерных программ в Роспатенте (свидетельство №2019662022).

- Программа для оптимизации траектории подготовки IT-специалистов на основе алгоритмов роевого интеллекта. Участники проекта – 4 студента направления подготовки «Программная инженерия». По результатам получено свидетельство о регистрации компьютерных программ в Роспатенте (свидетельство №2019660959).

- Программа для оценки личностных особенностей и психологических качеств IT специалистов. Участники проекта – 4 студента направления подготовки «Программная инженерия». По результатам получено свидетельство о регистрации компьютерных программ в Роспатенте (свидетельство №2019661028).

- Программа по обучению олимпиадному программированию на основе алгоритмов роевого интеллекта. Участники проекта – 7 студентов направлений подготовки «Информатика и вычислительная техника», «Биотехнические системы и технологии». По результатам получено свидетельство о регистрации компьютерных программ в Роспатенте (свидетельство №2020612506).

- Обучающая компьютерная игра «Один день». Программа предназначена для обучения детей дошкольного и младшего школьного возраста основам электробезопасности и энергосбережения. Участники

проекта – 5 студентов специальности «Автоматизированные системы обработки информации». Проект удостоен диплома первой степени на международном конкурсе «Цифровой ветер 2018» (<http://bru.by/news/post?id=25258>).

- Программно-информационная система для любителей активного отдыха. Участники проекта – 4 студента направления подготовки «Программная инженерия».

- Программно-информационная система для авто- и мотолюбителей. Участники проекта – 5 студентов направления подготовки «Программная инженерия».

- Программно-информационная система для любителей животных. Участники проекта – 4 студента направления подготовки «Программная инженерия».

- Программно-информационная система «Автошкола». Участники проекта – 4 студента направления подготовки «Программная инженерия». Проект занял 3-е место на областном командном турнире по программированию «CodingFest-3».

- Программно-информационная система для организации мероприятий. Участники проекта – 4 студента направления подготовки «Программная инженерия».

- Web-приложение «Достопримечательности Беларуси». Участники проекта – 6 студентов направления подготовки «Программная инженерия».

- «Программно-информационная система учета работы сотрудников. Участники проекта – 4 студента направления подготовки «Программная инженерия».

- Web-приложение «Здоровый образ жизни». Участники проекта – 5 студентов направления подготовки «Программная инженерия».

– Программно-информационная система «Оказание услуг населению». Участники проекта – 5 студентов направления подготовки «Программная инженерия».

– Web-приложение «Оказание услуг кейтеринга». Участники проекта – 5 студентов направления подготовки «Программная инженерия».

Результаты повышения эффективности управления учебными ИТ-проектами при использовании разработанного метода представлены в таблице 4.25.

Таблица 4.25 – Результаты повышения эффективности управления учебными ИТ-проектами при использовании разработанного метода

| Номер проекта | Начальная трудоемкость, $PM_1$ , чел.-дни. | Расчетная трудоемкость, $PM_2$ , чел.-дни. | $\Delta PM_{12}$ , % | Фактическая трудоемкость, $PM_3$ , чел.-дни. | $\Delta PM_{23}$ , % |
|---------------|--------------------------------------------|--------------------------------------------|----------------------|----------------------------------------------|----------------------|
| 1             | 575                                        | 490                                        | 14,78                | 480                                          | 2,04                 |
| 2             | 541                                        | 435                                        | 19,59                | 440                                          | 1,15                 |
| 3             | 580                                        | 462                                        | 20,34                | 450                                          | 2,60                 |
| 4             | 520                                        | 435                                        | 16,35                | 400                                          | 8,05                 |
| 5             | 745                                        | 570                                        | 23,49                | 585                                          | 2,63                 |
| 6             | 434                                        | 333                                        | 23,27                | 345                                          | 3,60                 |
| 7             | 810                                        | 527                                        | 34,94                | 500                                          | 5,12                 |
| 8             | 480                                        | 380                                        | 20,83                | 400                                          | 5,26                 |
| 9             | 489                                        | 390                                        | 20,25                | 375                                          | 3,85                 |
| 10            | 573                                        | 437                                        | 23,73                | 420                                          | 3,89                 |
| 11            | 679                                        | 416                                        | 38,73                | 440                                          | 5,77                 |
| 12            | 762                                        | 534                                        | 29,92                | 520                                          | 2,62                 |
| 13            | 960                                        | 487                                        | 49,27                | 470                                          | 3,49                 |
| 14            | 487                                        | 394                                        | 19,10                | 410                                          | 4,06                 |
| 15            | 439                                        | 327                                        | 25,51                | 300                                          | 8,26                 |

В среднем трудоемкость уменьшилась на 25,3%.

Для проверки соответствия результатов вычислительного эксперимента результатам производственного эксперимента используется  $t$ -критерий Стьюдента [61] для зависимых выборок. Отклонение результатов эксперимента от нормального распределения проверяется с использованием критерия Шапиро-Уилка [41]. Для расчетов используется пакет STATISTICA [15].

На рисунках 4.13-4.14 представлены результаты проверки нормальности распределения расчетной и фактической трудоемкости. Для критерия Шапиро-Уилка при  $n=15$  и  $p=0.05$  табличное значение  $p$ -квантиля равно 0,874. Поскольку  $W_{теор}=0,972>0,874$ ,  $W_{факт}=0,9967>0,918$ , гипотеза о нормальности выборок принимается.

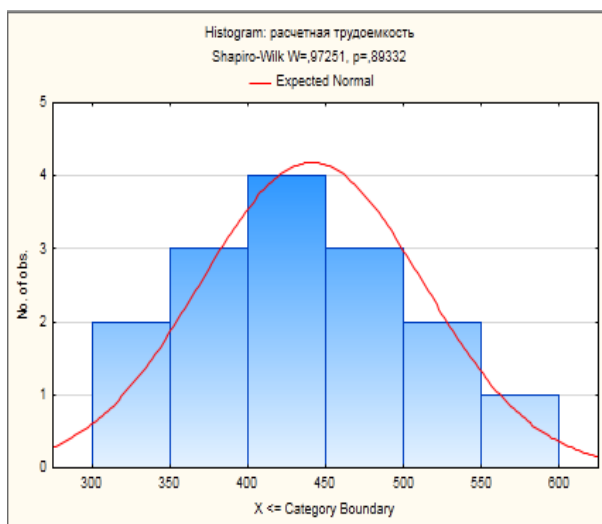


Рисунок 4.13 - Гистограмма распределения теоретической трудоемкости

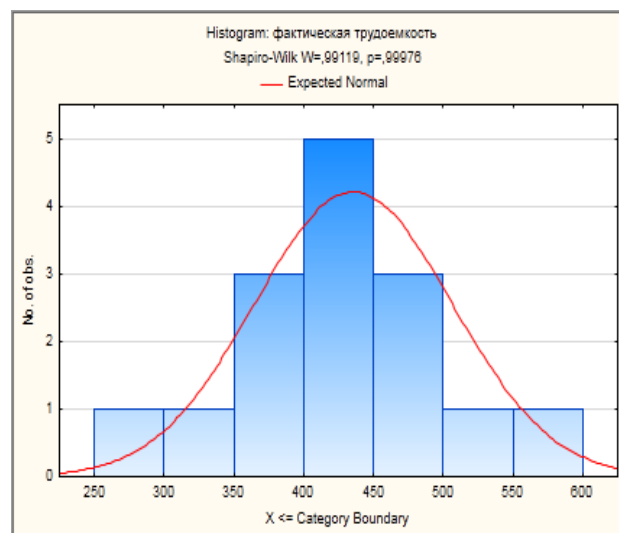


Рисунок 4.14 - Гистограмма распределения фактической трудоемкости

В таблице 4.26 представлены результаты проверки соответствия результатов вычислительного эксперимента результатам производственного эксперимента по  $t$ -критерию Стьюдента в пакете STATISTICA. Проверялась гипотеза о близости средних значений теоретических и экспериментальных значений трудоемкости. Таблица содержат следующие столбцы: *Mean* – средние значения трудоемкости для каждой из сравниваемых групп; *Std. Dv.* – стандартные отклонения для каждой группы; *N* – число наблюдений; *Diff.* – средняя разница трудоемкости; *Std. dv. diff.* – стандартное отклонение для средней разницы; *t* – значение  $t$ -критерия; *df* – число степеней свободы; *p* – вероятность ошибочно отвергнуть нулевую гипотезу о том, что средние величины трудоемкости в сравниваемых группах не различаются.

Таблица 4.26 – Результаты  $t$ -теста для значений теоретической и фактической трудоемкостей

| сравниваемые группы      | Mean     | Std. Dv  | N  | Diff.    | Std. Dv. Diff | t        | df | p        |
|--------------------------|----------|----------|----|----------|---------------|----------|----|----------|
| расчетная трудоемкость   | 441.1333 | 71.48813 |    |          |               |          |    |          |
| фактическая трудоемкость | 435.6667 | 71.06000 | 15 | 5.466667 | 19.09325      | 1.108889 | 14 | 0.286164 |

Поскольку  $p > 0,05$ , можно заключить, что средние значения трудоемкости, полученные теоретически и экспериментально существенно не различаются. Следовательно, гипотеза о близости средних значений теоретических и экспериментальных значений трудоемкости для учебных проектов принимается, модель является адекватной.

#### 4.4.4 Результаты применения метода, средств и технологий в коммерческих IT-проектах

Научные исследования выполнялись в рамках госбюджетной научно-исследовательской работы (№ госрегистрации 20150716) «Инновации в подготовке специалистов по направлению «Разработка программно-информационных систем». В 2020 году в БелИСА сдан заключительный отчет по данной теме [53].

В настоящее время исследования продолжаются в рамках госбюджетной научно-исследовательской работы (№ госрегистрации 20210581) «Методы, модели и интеллектуальные технологии исследования и проектирования информационно-аналитических процессов, ресурсов и систем».

Концепция, метод, алгоритмы, программные средства и технология, разработанные в рамках диссертационной работы, использованы при реализации следующих IT-проектов:

– IT-проект «Разработка автоматизированной системы обработки информации бизнес-процессов СЗАО «Могилевский вагоностроительный завод». Работы выполнялись в рамках хоздоговора №ХД11108 между Белорусско-Российским университетом и СЗАО «Могилевский

вагоностроительный завод». В проекте участвовало 25 исполнителей. Работы по договору успешно завершены и результаты сданы заказчику.

– IT-проект «Разработка автоматизированной информационно-аналитической системы формирования отчетов о выпуске, отгрузке продукции дивизионом ППУ и МК, о поставках на склад и отгрузке в цеха сырья и материалов». Работы выполнялись в рамках хоздоговора ХД1251 между Белорусско-Российским университетом и ООО «СМИТ-Ярцево». В проекте участвовало 8 исполнителей. Работы по договору успешно завершены и результаты сданы заказчику.

– IT-проект «Программный комплекс цеха тонкого металла ООО «СМИТ-Ярцево». Работы выполнялись в рамках хоздоговора ХД1387 между Белорусско-Российским университетом и ООО «СМИТ-Ярцево». В проекте участвовало 10 исполнителей. Работы по договору успешно завершены и результаты сданы заказчику.

– IT-проект «Разработка функциональной и информационной модели бизнес-процесса планирования продаж ЗАО «Завод полимерных труб». Работы выполнялись в рамках хоздоговора ХД1767 между Белорусско-Российским университетом и ЗАО «Завод полимерных труб». В проекте участвовало 10 исполнителей. Работы по договору успешно завершены и результаты сданы заказчику.

– IT-проект «Теплосеть Controlling». Данный проект выполнен в рамках договором о научно-техническом сотрудничестве между ООО «Степл Инк» и Белорусско-Российским университетом. Результаты внедрены в МГКУП «Теплоэнергетика». В проекте участвовало 9 человек. Результаты докладывались на двух международных конгрессах по теплоэнергетике в г. Минск, Беларусь.

– Автоматизированная система оценки работы сварщиков и качества сварочных работ. Участники проекта 5 студентов направлений подготовки «Информатика и вычислительная техника» и «Программная инженерия»,

один магистрант специальности «Системный анализ, управление и обработка информации». Результаты выполнения проекта в настоящее время проходят апробацию в ПО «Белоруснефть». Автоматизированная система доступна по ссылке [www.weldingcontrol.by](http://www.weldingcontrol.by).

– IT-проект по созданию информационного сайта инженерного факультета заочного образования Белорусско-Российского университета, сайт доступен по ссылке <http://demo.zao.bru.by>. В проекте участвовало 5 человек.

Остальные 9 проектов были успешно реализованы в ООО «Стэпл Икн», для которых применялся разработанный метод, программный комплекс и технология. Договором предусмотрено неразглашение информации, поэтому представлены только результаты в таблице 4.27.

Таблица 4.27 – Результаты повышения эффективности управления коммерческими IT-проектами при использовании разработанного метода

| Номер проекта | Начальная трудоемкость, $PM_1$ , чел.-дни. | Расчетная трудоемкость, $PM_2$ , чел.-дни. | $\Delta PM_{12}$ , % | Фактическая трудоемкость, $PM_3$ , чел.-дни. | $\Delta PM_{23}$ , % |
|---------------|--------------------------------------------|--------------------------------------------|----------------------|----------------------------------------------|----------------------|
| 1             | 2631                                       | 2107                                       | 19,92                | 2080                                         | 1,28                 |
| 2             | 1968                                       | 1706                                       | 13,31                | 1850                                         | 8,44                 |
| 3             | 2107                                       | 1780                                       | 15,52                | 1900                                         | 6,74                 |
| 4             | 1806                                       | 1564                                       | 13,40                | 1520                                         | 2,81                 |
| 5             | 1951                                       | 1687                                       | 13,53                | 1640                                         | 2,79                 |
| 6             | 2380                                       | 1850                                       | 22,27                | 2010                                         | 8,65                 |
| 7             | 2223                                       | 1744                                       | 21,55                | 1765                                         | 1,20                 |
| 8             | 2414                                       | 1822                                       | 24,52                | 1790                                         | 1,76                 |
| 9             | 2617                                       | 1806                                       | 30,99                | 1760                                         | 2,55                 |
| 10            | 2435                                       | 1687                                       | 30,72                | 1730                                         | 2,55                 |
| 11            | 2126                                       | 1576                                       | 25,87                | 1470                                         | 6,73                 |
| 12            | 1822                                       | 1462                                       | 19,76                | 1450                                         | 0,82                 |
| 13            | 1687                                       | 1211                                       | 28,22                | 1320                                         | 9,00                 |
| 14            | 2618                                       | 1951                                       | 25,48                | 2010                                         | 3,02                 |
| 15            | 2107                                       | 1413                                       | 32,94                | 1380                                         | 2,34                 |
| 16            | 9810                                       | 7750                                       | 22,02                | 7900                                         | 3,27                 |

Трудоемкость коммерческих проектов в среднем снизилась на 22.5%.

Проверка адекватности модели проводилась на пятнадцати проектах.

На рисунках 4.16-4.17 представлены результаты проверки нормальности распределения расчетной и фактической трудоемкости коммерческих проектов. Для критерия Шапиро-Уилка при  $n=15$  и  $p=0.05$  табличное значение  $p$ -квантиля равно 0,874. Поскольку  $W_{теор}=0,980>0,874$ ,  $W_{факт}=0,952>0,918$ , гипотеза о нормальности выборок принимается.

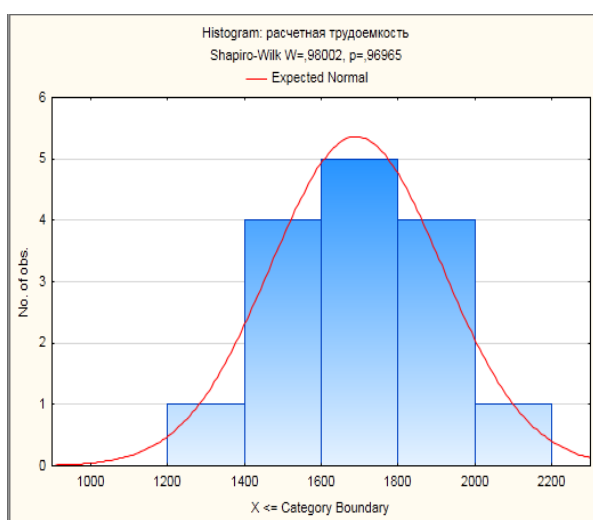


Рисунок 4.16 - Гистограмма распределения теоретической трудоемкости

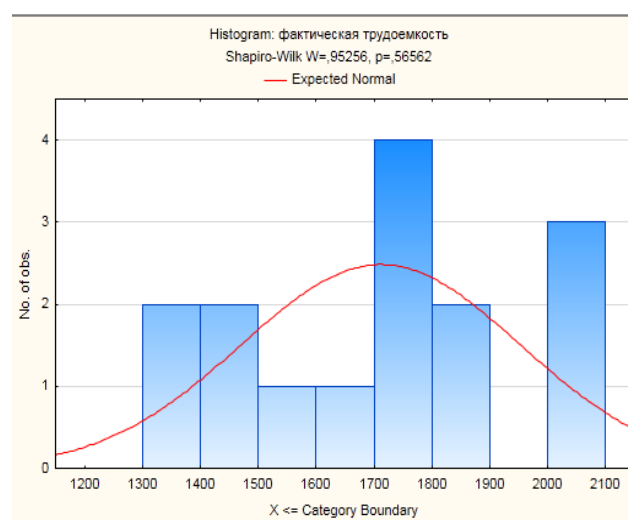


Рисунок 4.17 - Гистограмма распределения фактической трудоемкости

В таблице 4.28 представлены результаты проверки соответствия результатов вычислительного эксперимента результатам производственного эксперимента по  $t$ -критерию Стьюдента. Проверялась гипотеза о близости средних значений теоретических и экспериментальных значений трудоемкости коммерческих проектов.

Таблица 4.28 – Результаты  $t$ -теста для значений теоретической и фактической трудоемкостей коммерческих проектов

| сравниваемые группы      | Mean     | Std. Dv  | N  | Diff.    | Std. Dv. Diff | t        | df | p        |
|--------------------------|----------|----------|----|----------|---------------|----------|----|----------|
| расчетная трудоемкость   | 441.1333 | 71.48813 |    |          |               |          |    |          |
| фактическая трудоемкость | 435.6667 | 71.06000 | 15 | 5.466667 | 19.09325      | 1.108889 | 14 | 0.286164 |

Поскольку  $p > 0,05$ , можно заключить, что средние значения трудоемкости коммерческих проектов, полученные теоретически и экспериментально существенно не различаются. Следовательно, гипотеза о близости средних значений теоретических и экспериментальных значений трудоемкости для коммерческих проектов принимается, модель является адекватной.

#### **4.5 Внедрение результатов исследования**

4.5.1 Внедрение результатов исследования в учебный процесс Белорусско-Российского университета. Разработка использована в образовательном процессе кафедры «Программное обеспечение информационных технологий».

Программный комплекс многоуровневого управления IT-проектами используется в процессе выполнения выпускных квалификационных работ по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и направления подготовки 09.03.04 «Программная инженерия». В процессе курсового проектирования программный комплекс используется по следующим дисциплинам:

- «Базы и банки данных» специальности 1 – 53 01 02 «Автоматизированные системы обработки информации».
- «Программирование», «Объектно-ориентированное программирование», «Базы данных», «Средства взаимодействия человека с вычислительными системами» направления подготовки 09.03.01 «Информатика и вычислительная техника» и направления подготовки 09.03.04 «Программная инженерия».
- «Технологии разработки программного обеспечения», «Проектирование программного обеспечения», «Архитектура программных систем» направления подготовки 09.03.04 «Программная инженерия».

Программный комплекс многоуровневого управления IT-проектами обеспечивает повышение эффективности образовательного процесса за счет автоматизации формирования команд и распределения участников проекта на задачи с использованием эволюционного моделирования.

Работа выполнена по договору о сотрудничестве между Белорусско-Российским университетом (г.Могилев) и ООО «Стэпл Инк» (г.Минск).

4.5.2 Внедрение результатов исследования в ООО «Стэпл Инк». Программный комплекс использован в процессе управления учебными, аутсорсинговыми проектами и стартапами. Программный комплекс управления IT-проектами обеспечивает автоматизацию обработки информации о результатах психологического тестирования участников проектов, поддержку принятия решений при формировании проектных команд и при декомпозиции проекта на задачи, распределение участников проектной команды на задачи проекта на основе алгоритмов эволюционного моделирования, многоуровневый контроль качества решения задач. Разработанный программный комплекс включает следующие программные модули:

- модуль многоуровневого контроля решения проектных задач и модуль обработки результатов многоуровневого контроля решения проектных задач обеспечивают снижение трудоемкости выполнения IT-проектов за счет автоматизации сбора, хранения и обработки данных о результатах решения проектных задач;

- модуль обработки информации о ЛПК участников проектов, модули формирования структуры и состава проектных команд обеспечивают ускорение подбора рационального состава проектной команды за счет автоматизированной обработки результатов психологического тестирования с использованием кластерного анализа, автоматизации применения алгоритмов эволюционного моделирования для формирования проектной команды;

– модуль распределения участников проектных команд на задачи и модуль формирования состава и структуры задач проекта позволяют ускорить распределение участников проектной команды на задачи за счет автоматизации применения эволюционного моделирования для распределения участников проектной команды на задачи проекта.

Применение разработанного программного комплекса в ООО «Стэпл Инк» позволяет сократить трудоемкость выполнения IT-проектов на 20-30% по сравнению с ручным подбором участников проектных команд и планирования проектных работ.

Работа выполнена по договору о сотрудничестве между Белорусско-Российским университетом (г.Могилев) и ООО «Стэпл Инк» (г.Минск).

#### **4.6 Перспективы применения и развития**

Перспективы применения:

1. Выполнение учебных проектов командами студентов. Разработанный метод обеспечивает реализацию процессов проектно-организованного обучения IT-специалистов в рамках программ подготовки бакалавров. Перспективным является применение предложенного метода и алгоритмов при реализации командами студентов небольших IT-проектов по заказам предприятий и организаций в рамках курсового и дипломного проектирования. Подобный подход позволяет преподавателям повышать квалификацию в качестве руководителей IT-проектов, а студентам осваивать новые технологии разработки программного обеспечения.

2. Выполнение внутренних проектов в IT-компаниях для повышения квалификации сотрудников. Для сохранения численности сотрудников в период сокращения количества проектов часть сотрудников аутсорсинговых IT-компаний занимаются повышением квалификации, выполняя внутренние проекты. Предложенная концепция, метод и алгоритмы могут использоваться для сокращения сроков реализации внутренних IT-проектов.

3. Формирование команд разработчиков в IT-проектах в аутсорсинговых компаниях. При появлении коммерческого проекта в аутсорсинговой IT-компании актуальной является задача формирования команды проекта из числа исполнителей, занимающихся повышением квалификации и реализацией внутренних IT-проектов. Разработанный метод и алгоритмы могут использоваться руководителем IT-проекта для формирования рационального состава команды исполнителей проекта.

4. Формирование команд разработчиков и распределение исполнителей на задачи в IT-компаниях с продуктовой моделью. В IT-компаниях с продуктовой моделью основной задачей является создание нового программного продукта или новой версии существующего программного продукта с минимальными трудозатратами в минимальные сроки. Разработанный метод и алгоритмы позволяют снизить трудозатраты и сократить сроки реализации нового программного продукта или создания новой версии существующего программного продукта за счет формирования рационального состава команд исполнителей и распределения исполнителей на задачи.

5. В стартапах одной из основных задач является создание программного продукта в минимальные сроки для опережения конкурентов и максимально быстрого запуска первой версии проекта. Спецификой стартапов в настоящее время является возможность выбора исполнителей из большого числа претендентов. Разработанный метод и алгоритмы обеспечивают существенное сокращение времени рационального выбора состава команды исполнителей в стартап. Рациональный выбор состава команды и распределение исполнителей на задачи с использованием разработанного метода и алгоритмов позволит получить конкурентные преимущества за счет сокращения трудозатрат и сроков реализации стартапа.

### Перспективы развития:

1. Оценка влияния ЛПК и профессиональных компетенций исполнителей на эффективность управления в любых типах проектов. Подбор методов, методик и тестов, обеспечивающих анализ ЛПК с учетом специфики различных типов проектов, позволит оценить влияние ЛПК на эффективность управления проектом. Критерии повышения эффективности будут определяться специфическими особенностями проекта. Комплексная оценка влияния ЛПК и профессиональных компетенций исполнителей на эффективность управления различными типами проектов позволит выбирать рациональный состав команд и распределять исполнителей на задачи с учетом требований к ЛПК, профессиональным компетенциям с учетом специфики различных типов проектов.

2. Оценка трудоемкости любых типов проектов с интеграцией результатов оценки в операции эволюционного моделирования для повышения эффективности. Трудоемкость является одним из ключевых критериев эффективности управления для любого проекта. Снижение трудоемкости за счет формирования рационального состава команд и распределения исполнителей на задачи с использованием методов и средств эволюционного моделирования позволит сократить расходы на реализацию и сроки реализации различных типов проектов.

3. Рациональный выбор параметров алгоритмов эволюционного моделирования при формировании команд и распределении исполнителей на задачи в IT-проектах. Структурно-параметрическая адаптация алгоритмов эволюционного моделирования обеспечит нахождение рационального состава команды исполнителей и распределении исполнителей на задачи в IT-проектах при минимальном количестве итераций эволюционного моделирования.

4. Развитие предложенного метода для повышения эффективности управления разными типами проектов. Решение данной задачи предполагает комплексную оценку ЛПК и профессиональных компетенций с учетом

специфики различных типов проектов, оценку трудоемкости, адаптацию методов и средств эволюционного моделирования к решению задачи рационального выбора состава команд исполнителей и распределения исполнителей на задачи для разных типов проектов.

5. Модификация предложенного метода для реализации возможности участия одного исполнителя одновременно в нескольких проектах, частичной занятости исполнителя в проекте. Решение поставленной задачи предполагает замену эволюционного моделирования другим математическим аппаратом.

## ВЫВОДЫ

Эффективность предложенного метода и алгоритмов подтверждена апробацией на учебных и коммерческих IT-проектах.

Тестирование на учебном проекте для организации управления процессом обучения IT-специалистов показало сокращение трудоемкости в результате применения эволюционного моделирования на 23% по сравнению с лучшим решением руководителя проекта.

Тестирование метода при создании промышленного программного продукта в ООО «Стэпл Инк» «Теплосеть Controlling» показало сокращение трудоемкости в результате применения эволюционного моделирования на 26% по сравнению с лучшим решением руководителя проекта.

Проведена апробация на 15 учебных проектах. По результатам реализации опубликованы четыре работы, получены четыре свидетельства о регистрации, один проект получил диплом первой степени на международном конкурсе «Цифровой ветер 2018», один проект занял 3-е место на областном командном турнире по программированию «CodingFest-3».

Проведена апробация на 15 коммерческих проектах. Результаты докладывались на двух международных конгрессах по теплоэнергетике в г. Минск, Беларусь.

Представлены перспективы применения в образовательном процессе и IT-компаниях.

Описаны перспективы развития результатов диссертационной работы для повышения эффективности управления проектами различных типов.

## ЗАКЛЮЧЕНИЕ

Решена актуальная народнохозяйственная задача повышения эффективности управления организационными процессами ИТ-проектов за счет предложенных авторских решений – концепции, метода, алгоритмов и программного комплекса.

1. Проведен анализ существующих подходов к управлению организационными системами ИТ-проектов и средств автоматизации управления проектами. На основе анализа выявлены следующие недостатки управления организационными процессами ИТ-проектов: отсутствуют средства **комплексной** оценки сочетания профессиональных компетенций с личностными и психологическими качествами участников ИТ-проектов, оценка трудоемкости выполнения работ проекта, назначение исполнителей выполняется без учета специфики ИТ-проектов, отсутствует многоуровневый контроль качества решений задач ИТ-проекта. Обоснована актуальность создания математического и программного обеспечения повышения эффективности управления организационными процессами ИТ-проектов.

2. Разработана концепция повышения эффективности управления организационными процессами ИТ-проектов. Предложенная концепция **отличается** применением комплексного подхода к решению задачи повышения эффективности управления организационными процессами ИТ-проектов. Разработанная концепция позволяет проводить комплексную обработку информации о личностных и психологических качествах и профессиональных компетенций участников команд исполнителей, комплексную поддержку принятия решений при управлении синтезом состава и структуры проектных задач и распределении участников проекта на задачи на основе эволюционного моделирования.

3. Предложен метод повышения эффективности управления организационными процессами ИТ-проектов, **отличающийся** интеграцией результатов комплексной обработки информации о личностных и

психологических качествах и профессиональных компетенций участников команд исполнителей в операции эволюционного моделирования для поддержки принятия решений при формировании состава команд исполнителей и назначения участников проекта на задачи.

4. Разработаны алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта путем адаптации средств эволюционного моделирования к решению задачи планирования выполнения работ в IT-проекте. Предложенные алгоритмы **отличаются** применением эволюционного моделирования в сочетании с методикой СОСОМО II и методом парных сравнений Т.Саати. Алгоритмы поддержки принятия решений при формировании структуры и состава команд исполнителей и распределения исполнителей на задачи проекта обеспечивают автоматизацию организационных процессов на этапах планирования и организации работ при реализации IT-проектов.

5. Разработана структура и состав программного комплекса и технология многоуровневого управления IT-проектами, реализующие автоматизированное формирование иерархии задач, синтеза структуры и состава команд исполнителей и многоуровневый контроль качества решения задач.

6. Результаты исследования прошли экспериментальную проверку в группе студентов по специальности «Программная инженерия» Белорусско-Российского университета. Теоретический расчет на основе предложенного метода показал сокращение трудоемкости выполнения проектов в среднем на 25,3%. Эксперимент в образовательном процессе Белорусско-Российского университета подтвердил соответствие результатов теоретического расчета результатам, полученным на практике.

7. Результаты исследований использовались для формирования системы проектно-организованного обучения студентов по направлениям подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная

инженерия» Белорусско-Российского университета (акт о внедрении результатов научно-исследовательской работы в образовательный процесс).

8. Выполнена апробация разработанной концепции, метода, алгоритмов и технологии при реализации коммерческих IT-проектов в рамках 4 хоздоговоров с СЗАО «Могилевский вагоностроительный завод» (г.Могилев, Республика Беларусь), ООО «СМИТ-Ярцево» (г.Ярцево, Смоленская область, Российская Федерация), ЗАО «Завод полимерных труб» (г.Могилев, Республика Беларусь). В настоящее время результаты диссертационной работы используются при выполнении хоздоговора с РУП «Производственное объединение «Белоруснефть».

9. Внедрение разработанной технологии для управления организационными процессами IT-проектов в ООО «Стэпл Инк» (г.Минск, Республика Беларусь) позволило снизить трудоемкость выполнения проекта на 20-30% за счет повышения эффективности управления IT-проектами на основе эволюционного моделирования, что подтверждено актом внедрения.

10. Результаты диссертационных исследований могут применяться в высших и средних специальных учебных заведениях, занимающихся подготовкой IT-специалистов при реализации проектно-организованного обучения, в аутсорсинговых IT-компаниях – для повышения квалификации сотрудников при реализации внутренних IT-проектов, в коммерческих проектах – для повышения эффективности формирования команд исполнителей и распределения исполнителей на задачи, в IT-компаниях с продуктовой линейкой программных продуктов – для сокращения сроков реализации стартапов.

Развитие результатов диссертационных исследований может выполняться в следующих направлениях: оценка влияния ЛПК и профессиональных компетенций исполнителей на эффективность управления в любых типах проектов; оценка трудоемкости любых типов проектов с интеграцией результатов оценки в операции эволюционного моделирования для повышения эффективности управления проектами; рациональный выбор

параметров алгоритмов эволюционного моделирования при формировании команд и распределении исполнителей на задачи в IT-проектах; развитие предложенного метода для повышения эффективности управления разными типами проектов; модификация предложенного метода для реализации возможности участия одного исполнителя одновременно в нескольких проектах, частичной занятости исполнителя в проекте.

## СПИСОК ЛИТЕРАТУРЫ

1. Аверченков, В. И. Эволюционное моделирование и его применение: монография. 2-е изд. / В. И. Аверченков, П. В. Козаков. - М.: Флинта, 2011. - 200 с.
2. Агасиев, Т. А. Прогнозирование эффективных стратегий алгоритма оптимизации на основе анализа информационного содержания ландшафта целевой функции / Т. А. Агасиев, А. П. Карпенко // Системы компьютерной математики и их приложения. – 2018. – № 19. – С. 3-8.
3. Адакин Е.Е., Скрипникова Г.В. Оценка профессиональных квалификаций работников // Фундаментальные исследования. 2016. № 7-2. С. 282–286.
4. Адизес И. К. Управление жизненным циклом корпорации. - СПб: Питер, 2007.- 383 с.
5. Андреев, А. М. Программное обеспечение для рационального распределения задач IT-проектов между исполнителями / Ю. В. Вайнилович, А. М. Андреев, С. М. Емельяненко // Междисциплинарность научных исследований как фактор инновационного развития: сборник статей Международной научно-практической конференции. г. Иркутск, 2020. - С. 19-21.
6. Аньшин В.М., Ильина О. Н. Исследование методологии оценки и анализ зрелости управления портфелями проектов в российских компаниях. М:ИНФРА-М, 2018. 200 с.
7. Антонова, А. С. Применение мультиагентного и эволюционного моделирования при планировании работ / А. С. Антонова, К. А. Аксенов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2013. – № 6 (186). – С. 126–136.
8. Артемов, О. В. Управление внутренней интеграцией команды // Управление интеллектуальным капиталом. Материалы международной конференции – Владимир: Изд-воВлГУ, 2007. – С 91-95

9. Бабенчук С.П. Анализ программного обеспечения управления проектами // Программные продукты и системы. 2011. №2. С 45-49.
10. Батаршев А. В., Алексеева И. Ю., Майорова Е. В. Диагностика профессионально важных качеств. — СПб. : Питер, 2007.
11. Белбин Р.М. Типы ролей в командах менеджеров; [пер.с англ.]. М.: НИРРО, 2003. - 232 с.
12. Бодров В.А. Психология профессиональной деятельности. М.: Институт психологии СО РАН, 2006. 623 с.
13. Борисов В.В. Программный комплекс управления подготовкой IT-специалистов SkillsForYou / В.В. Борисов, С.П. Янукович, Т.В. Мрочек, Д.С. Ореховский // Программные продукты и системы. 2020. — № 2. С.177-185. DOI: 10.15827/0236-235X.130.177-185.
14. Борисов В.В. Интеллектуальное управление подготовкой к олимпиадам по программированию на основе алгоритмов речевого интеллекта / В.В. Борисов, С.П. Янукович, Т.В. Мрочек, М.В. Воробьев, А.Ю. Миранков // Искусственный интеллект и принятие решений, 2021. — № 1. с.62-74. DOI: 10.14357/20718594210106
15. Боровиков В.П. Программа Statistica для студентов и инженеров / В.П. Боровиков. - М.: Компьютер пресс, 2000. - 301 с.
16. Боэм Б. Инженерное проектирование программного обеспечения: Пер с англ. — М.: Радио и Связь, 1985. — 240 с.
17. Боэм Б., Браун Дж., Каспар Х., Характеристики качества программного обеспечения. — М.: "Мир", 1981. — 208 с.
18. Бруде Э. Технология разработки программного обеспечения. — СПб.: Питер, 2004. 655 с.
19. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. — Пер. с англ. — СПб.: Символ-Плюс, 1999. — 304 с.
20. Будыльский, А. В. Управление проектами разработки программного обеспечения с использованием агентных технологий / А. В. Будыльский,

- И. Ю. Квятковская // Прикаспийский журнал: управление и высокие технологии - 2013 - №3(23) - С. 119-128.
21. Бурков В.Н., Коргин Н. А., Новиков Д. А. Как управлять проектами. М. : Либроком, 2009. 264 с.
  22. Бурков В.Н., Новиков Д. А. Теория активных систем: состояние и перспективы. М. : Синтег, 1999. 128 с.
  23. Бурков В. Н. Основы математической теории активных систем. – М.: Наука, 1977.
  24. Бурков В. Н., Данев Б., Еналеев А. К. и др. Большие системы: моделирование организационных механизмов. – М.: Наука, 1989.
  25. Буркова И.В. Метод сетевого программирования в задачах нелинейной оптимизации. – «Автоматика и телемеханика», журнал. 2009. №10. С. 15-21.
  26. Буч Г. Объектно-ориентированное проектирование. – Киев, Москва, “Диалектика”, АО “ИВК” — 516 с.
  27. Буч Г. Рамбо. Д. Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. ДМК, 2000. – 432 с.
  28. Вайнилович, Ю.В. Программный комплекс многоуровневого управления ИТ-проектами / Ю. В. Вайнилович, К. В. Захарченков // Инновации, 2019. – № 8(250). – С. 88–96.
  29. Васильев Д.К., Заложнев А.Ю., Новиков Д.А., Цветков А.В. Типовые решения в управлении проектами. – М.: ИПУ РАН, 2003.
  30. Воробич Н.П. Программные методы и средства планирования и управления проектами // ВЕСТНИК КРАСГАУ – 2009 - №10(37) – с. 6-11
  31. Воропаев, В.И. Обобщенные стохастические сетевые модели для управления комплексными проектами (часть 1) / В.И.Воропаев, Я.Д.Гельруд // Управление проектами и программами. –2008. –№1. –С. 2–13.

32. Воропаев, В.И. Управление проектами для стейкхолдеров / В.И.Воропаев, Я.Д.Гельруд, О.А.Клименко. – М.: LAP LAMBERT Academic Publishing, 2015. – 278 с.
33. Вуд, Кит Расширение библиотеки jQuery / Кит Вуд. – Москва : ДМК Пресс, 2013. - 256 с.
34. Гецци К., Джазайери М. Мандриоли Д. Основы инженерии программного обеспечения. – СПб.: БХВ-Петербург, 2005. 832 с.
35. Гладков, Л. А. Решение задач поиска и оптимизации проектных решений на основе гибридного подхода / Л. А. Гладков, Н. В. Гладкова, С. Н. Лейба // Известия ЮФУ. Технические науки. – 2019. – № 4(206). – С. 164-174. – DOI 10.23683/2311-3103-2019-4-164-174.
36. Гладков, Л. А. Интегрированный подход к решению задач оптимизации на основе методов эволюционного проектирования и мультиагентных технологий / Л. А. Гладков, Н. В. Гладкова // Информатика, вычислительная техника и инженерное образование. – 2019. – № 1(34). – С. 1-8.
37. Гладуэлл М. Гении и аутсайдеры: Почему одним всё, а другим ничего?; [пер. с англ. О. Галкин]. М.: Манн, Иванов и Фербер, 2016. 224 с.
38. Грей Клиффорд Ф., Ларсон Эрик У. Управление проектами. Практическое руководство: Пер. с англ. – М.: Изд. «Дело и сервис», 2003. 528 с.
39. Голдратт Э. Критическая цепь. Пер. с англ. М: ТОС Центр, 2006. 272 с.
40. Гришина Н.В. Психология конфликта. 2-е изд. – СПб.: Питер, 2008. — 544 с.
41. ГОСТ Р ИСО 5479-2002. Национальный стандарт Российской Федерации. Статистические методы. Проверка отклонения распределения вероятностей от нормального распределения. Введен 2002-07-01
42. Губко М.В. Механизмы управления организационными системами с коалиционным взаимодействием участников. – М.: ИПУ РАН, 2003.

43. Губко М.В., Новиков Д.А. Теория игр в управлении организационными системами. – М.: Синтег, 2002
44. Гультяев, А.К. MS Project 2007 Управление проектами / А.К. Гультяев. – СПб.: Корона-Век, 2012. - 480 с.
45. Гупта, А. Java EE 7. Основы / Арун Гупта. – М.: Вильямс, 2014. – 336 с.
46. Гуцыкова С.В. Метод экспертных оценок. Теория и практика / С.В. Гуцыкова. - М.: Институт психологии РАН, 2012. - 144 с.
47. Дейт, К. Дж. Введение в системы баз данных – Introduction to Database Systems / К. Дж. Дейт. – 8-е изд. – М. : Вильямс, 2006. – 1328 с.
48. ДеМарко Т., Листер Т. Человеческий фактор: успешные проекты и команды. Издательство: Символ-Плюс, 2005 г. – 256 с.
49. Дронов В. А. Angular 4. Быстрая разработка сверхдинамических Web-сайтов на TypeScript и PHP. – СПб.:БХВ\_Петербург, 2018. – 448 с.
50. Емельяненко, С. М. Программное обеспечение для рационального формирования команд IT–проектов / Ю. В. Вайнилович, А. М. Андреев, С. М. Емельяненко // Теоретические и практические основы научного прогресса в современном обществе: сборник статей Международной научно-практической конференции, г. Уфа, 2020 – С. 11-14.
51. Емельянов В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.М. Курейчик, В.В. Курейчик – М.: Книга по Требованию, 2013. – 430 с.
52. Захарченков К.В., Вайнилович Ю.В. Методика многоуровневого управления учебными IT-проектами // Энергетика, информатика, инновации – 2018 (инновационные технологии и оборудование в промышленности, управление инновациями, экономика и менеджмент, научные исследования в области физической культуры, спорта и общественных наук): сб. трудов VIII междунар. науч.-техн. конф. Смоленск: «Универсум», 2018. Т 3. С. 18–21.
53. Инновации в подготовке специалистов по направлению «Разработка программно-информационных систем» [Электронный ресурс]: отчёт о

- НИР (заключ.) / Белорусско-Российский университет; К.В. Захарченков, Т.В. Мрочек, Ю. В. Вайнилович и др. . — Могилев, 2019. — 49 с. — Библиогр.: с. 38-44. — № ГР 20150716. — Инв. № 89511.
54. Йордон Э. Структурное проектирование и конструирование программ./ Пер. с англ. Фролова В.В., Теплицкого Л.А.; Под ред. Королева Л.Н.- М.: Мир, 1979.- 415 с.
  55. Йордон, Э. Управление сложными Интернет-проектами / Э. Йордон. - М.: Лори, 2014. - 344 с.
  56. Карпенко А. П. Современные алгоритмы оптимизации. Алгоритмы, вдохновленные природой / А. П. Карпенко. – Москва: Московский Государственный Технический Университет (МГТУ) имени Н.Э. Баумана – 2017. – 447 с.
  57. Карпенко, А. П. Выразительные средства популяционных алгоритмов глобальной оптимизации / А. П. Карпенко // Системы компьютерной математики и их приложения. – 2018. – № 19. – С. 283-292.
  58. Карпенко, А. П. Методы повышения эффективности популяционных алгоритмов глобальной оптимизации / А. П. Карпенко // Перспективные направления развития отечественных информационных технологий : материалы V межрегиональной научно-практической конференции, Севастополь, 24–28 сентября 2019 года / Севастопольский государственный университет; Санкт-Петербургский институт информатики и автоматизации РАН. – Севастополь: Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет", 2019. – С. 87-88.
  59. Карпенко, А. П. Паттерны мета-эвристических популяционных алгоритмов глобальной оптимизации / А. П. Карпенко // Нечеткие системы, мягкие вычисления и интеллектуальные технологии НСМВИТ-2020 : Труды VIII Международной научно-практической конференции.

В 2-х томах, Смоленск, 29 июня – 01 2020 года. – Смоленск: Универсум, 2020. – С. 5-15.

60. Коберн А. Современные методы описания функциональных требований к системам. М: Лори, 2002 г. – 264 с.
61. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников / А.И. Кобзарь. - М.: ФИЗМАТЛИТ, 2012. - 816 с.
62. Кон М. Agile: оценка и планирование проектов. – М.: ООО «Альпина Диджитал», 2018. – 460 с.
63. Кожемякин, А. А. HTML и CSS в примерах. Создание Web-страниц / А. А. Кожемякина. – Москва : «Альтекс», 2004. – 416 с.
64. Козмина, Ю. В. Spring 5 для профессионалов / Ю. В. Козмина. – Москва : «Вильямс», 2019. – 1120 с.
65. Коновальчук Е. В., Новиков Д. А. Модели и методы оперативного управления проектами. М.:ИПУ РАН, 2004. 63 с.
66. Константайн Л. Человеческий фактор в программировании. Символ-Плюс, 2004 г. – 384 с.
67. Коул, Р. Блистательный Agile. Гибкое управление проектами с помощью Agile, Scrum и Kanban / Р. Коул, Э. Скотчер. - СПб.: Питер, 2015. - 376 с.
68. Крачтен Ф. Введение в Rational Unified Process. Вильямс, 2002 г. – 240 с
69. Курейчик, В.М. Генетические алгоритмы и их применение / В.М. Курейчик. – 2-е изд. перераб. и доп. – Таганрог: Изд-во ТРТУ, 2002. – 242 с. 172 24.
70. Курейчик, В.М. Исследование динамических операторов в эволюционном моделировании / В.М. Курейчик, Л.А. Зинченко, И.В. Хабарова // Перспективные информационные технологии и интеллектуальные системы. – 2001. – №3(7). - С. 65–70. 25.
71. Курейчик, В.М. Эволюционные вычисления: Генетическое и эволюционное программирование / В.М. Курейчик, С.И. Родзин // Новости искусственного интеллекта. – 2003. - №5.

72. Левин, Курт., Динамическая психология : Избр. тр. / Курт Левин; Под общ. ред. Д.А. Леонтьева и Е.Ю. Патяевой; [Сост., пер. с нем. и англ. яз. и науч. ред. Д.А. Деонтьев, Е.Ю. Патяева]. – М. : Смысл, 2001. - 572 с.
73. Липаев В. В. Качество программного обеспечения // Финансы и статистика, М., 1983
74. Липаев В. В. Проектирование программных средств. М.: Высш. шк., 1990. – 330 с.
75. Липаев В.В. Техничко-экономическое обоснование проектов сложных программных средств / В.В. Липаев. - М. : СИНТЕГ, 2004. - 284 с.
76. Литвак Б.Г., Экспертные технологии в управлении. 2-е изд., испр. и дополн. -М.: Дело, 2004. - 400 с.
77. Лич, Л. Вовремя и в рамках бюджета: Управление проектами по методу критической цепи / Л. Лич. - М.: Альпина Паблишер, 2016. - 352 с.
78. Лонг, Д. «Java в облаке» / Д. Лонг. — СПб.: «Питер», 2019. — 624 с.
79. Мандель И. Д. Кластерный анализ. — М.: Финансы и статистика, 1988. — 176 с.
80. Метод управления процессом обучения промышленному программированию на основе алгоритмов роевого интеллекта / В. В. Борисов, С. П. Янукович, К. В. Захарченков, Ю. В. Вайнилович // Cloud of Science. – 2020. – Т. 7. – № 1. – С. 189-206.
81. Новиков, Д. А. Управление проектами: организационные механизмы./ Д. А. Новиков – М.: ПМСОФТ, 2007. – 140 с.
82. Новиков, Д.А. Курс теории активных систем / Д.А. Новиков, С.Н. Петрачков. — М: СИНТЕГ, 1999. — 104 с.
83. Новиков Д.А. Математические модели формирования и функционирования команд. – М.: Издательство физико-математической литературы, 2008. – 184 с.
84. Новиков Д.А. Теория управления организационными системами. – М.: Физматлит, 2005. 584 с.

85. Новиков Д.А., Цветков А.В. Механизмы стимулирования в многоэлементных организационных системах. М.: Апостроф, 2000.
86. Новиков, Д. А. Структура теории управления социально-экономическими системами. // Управление большими системами. 2009. №24. с. 216-237.
87. Новиков Д.А., Чхартишвили А.Г. Рефлексивные игры. М.: Синтег, 2003.
88. Остроух, Е. Н. Решение задачи моделирования бизнес-процессов IT-компаний с использованием метода роевого интеллекта / Е. Н. Остроух, Ю. О. Чернышов, Д. Н. Климова // Международный научно-исследовательский журнал, 2017. – Ч. 3. №9. – С.53-57
89. Низамутдинов, М. Тактика защиты и нападения на Web-приложения / М. Низамутдинов. - М.: БХВ-Петербург, 2005. - 109 с.
90. Павлов, А.Н. Управление проектами на основе стандарта PMI PMBOK.Изложение методологии и опыт применения / А.Н. Павлов. - М.: Бином. Лаборатория знаний, 2012. - 208 с.
91. Прихожий, А. А. Метод оценки квалификации и оптимизация состава профессиональных групп программистов / А.А. Прихожий А.М. Ждановский // Системный анализ и прикладная информатика, 2018. – №2. – С.4–11.
92. Пряжников Н.С. Профессиональное самоопределение: теория и практика. М.: Академия, 2008. 320 с.
93. Ройс У. Управление проектами по созданию программного обеспечения. Унифицированный подход. М.: Лори, 2002. – 434 с
94. Романова М.В. Анализ программного обеспечения и выявление проблем при внедрении информационной системы управления проектами // Экономический анализ: теория и практика. 2007. №11(92). С 50-61
95. Саати Т. Л. Принятие решений. Метод анализа иерархий. — М.: Радио и связь, 1993. — 320 с.
96. Саати, Т.Л. Принятие решений при зависимостях и обратных связях : аналитические сети. М.: URSS : ЛКИ, 2008. – 357 с

97. Савченко Т.Н. Применение методов кластерного анализа для анализа данных психологических исследований // прикладная юридическая психология №4 2008. 100-111 с.
98. Саймон Д. Алгоритмы эволюционной оптимизации. — М: ДМК Пресс, 2020. 940 с.
99. Сингаевская Г. И. Управление проектами в Microsoft Project 2007. – М.: «Диалектика», 2008. 800 с.
100. Титаренко Б.П. Робастные технологии управления в ситуации неопределенности.//Вестник МГАДА, Серия Экономика, 2012, №6, с.119-121
101. Томсетт Р. Радикальное управление ИТ-проектами. Лори, 2005 г. 294 стр.
102. Файоль А., Эмерсон Г., Тейлор Ф., Форд. Г. Управление это наука и искусство. М: Республика, 1992. 349 с.
103. Фаулер М. Архитектура корпоративных программных приложений. Вильямс, 2004 г. – 533 с.
104. Хелдман К. Профессиональное управление проектами. Пер. с англ. – М.: Бином, 2013. 517 с.
105. Чакон С., Штрауб Б. Git для профессионального программиста. – СПб.: Питер, 2016. – 496 с.
106. Чхартишвили А.Г. Теоретико-игровые модели информационного управления. М.: ПМСОФТ, 2004.
107. Шадриков В.Д. Профессиональные способности. М.: Университетская книга, 2010. 320 с.
108. Шапиро В. Д. и др. Управление проектами. – СПб.: ДваТри, 1996 г. – 610 с.
109. Янукович С.П. Концепция повышения эффективности управления образовательными системами на основе адаптивных алгоритмов роевого интеллекта / С.П. Янукович, В.В. Борисов, К.В. Захарченков //

Информационные технологии, 2020. –№ 12. с.706-716. DOI: 10.17587/it.26.706-716.

110. Янукович, С. П. Контроль самостоятельной работы студентов IT-специальностей посредством автоматизации / С. П. Янукович, Т. В. Мрочек, Д. С. Ореховский // Материалы, оборудование и ресурсосберегающие технологии: материалы Международной научно-технической конференции, Могилев, 25-26 апреля 2019 г. / редкол.: М. Е. Лустенков (гл. ред.) [и др.]. - Могилев: Белорусско-Российский университет, 2019. - С. 398.
111. Янукович, С.П. Цикл автоматизированного информационного управления подготовкой IT-специалистов / С.П. Янукович, Т.В. Мрочек, Д.С. Ореховский // ЭНЕРГЕТИКА, ИНФОРМАТИКА, ИННОВАЦИИ – 2018 (инновационные технологии и оборудование в промышленности, управление инновациями, экономика и менеджмент, научные исследования в области физической культуры, спорта и общественных наук). Сб. трудов VIII -ой Межд. науч.-техн. конф. В 3 т. Т 3. – Смоленск «Универсум», 2018. – С. 53-57
112. Янукович, С.П. Анализ личностных и психологических качеств обучающихся для повышения эффективности управления подготовкой IT-специалистов / С.П. Янукович, В.П. Шуть, Т.Е. Титов // ЭНЕРГЕТИКА, ИНФОРМАТИКА, ИННОВАЦИИ – 2019 (инновационные технологии и оборудование в промышленности, управление инновациями, экономика и менеджмент, научные исследования в области физической культуры, спорта и общественных наук). Сб. трудов IX -ой Межд. науч.-техн. конф. В 2 т. Т 2. – Смоленск «Универсум», 2019. – С. 370-373.
113. A. J. Albrecht, J. E. Gaffney. «Software function, source lines of codes, and development effort prediction: a software science validation», IEEE Trans Software Eng. SE-9, 1983, pp.639-648.

114. Boehm B., "A Spiral Model of Software Development and Enhancement", Proc. Int'l Workshop Software Process and Software Environments, ACM Press, 1985
115. Boehm B., et al. «Software cost estimation with COCOMO II». Englewood Cliffs, NJ: Prentice–Hall, 2000.
116. C. Jones. «Applied Software Measurement, Assuring Productivity and Quality»
117. De Jong, K. Analysis of behavior of class of genetic adaptive systems: Ph D. Thesis / K. De Jong. - University of Michigan, 1975.
118. Douglas McGregor. The Human Side of Enterprise. Penguin, 1987. P. 47.
119. G. Parkinson. «Parkinson's Law and Other Studies in Administration, Houghton-Mifflin»
120. Holland, J. Adaptation in Natural and Artificial Systems. An Introductory Analysis with Application to Biology, Control and Artificial Intelligence / J. Holland. - The University of Michigan Press, 1975. - 210 p.
121. J. D. Farmer, N. Packard and A. Perelson, “The Immune System, Adaptation and Machine Learning,” Physica D, Vol. 2, No. 1-3, 1986, pp. 187-204. doi:10.1016/0167-2789(86)90240-X
122. Kirkpatrick S., Gelatt Jr C.D., Vecchi M.P. Optimization by simulated annealing // Science, 1983. no. 220. P. 671-680.
123. Martin, James Rapid Application Development, New York : Macmillan Pub. Co., – 1991
124. L. Putnam. «A general empirical solution to the macro software sizing and estimating problem», IEEE Trans. Soft. Eng., July 1978, pp. 345-361.
125. M. Shepperd, C. Schofield. “Estimating software project effort using analogy”, IEEE Trans. Soft. Eng. SE-23:12, 1997, pp. 736-743
126. N. Fenton. S. Pfleeger. «Software Metrics: A Rigorous and Practical Approach».
127. The CHAOS Manifesto. The Standish Group International, 2015. 48 p.

128. A Guide To The Project Management Body Of Knowledge (PMBOK® Guide) – Fifth Edition, 2013, Project Management Institute, PA, USA.
129. PRINCE2®:2009 Manual – Managing Successful Projects With PRINCE2® – 2009 Edition, 2012, London, The Stationery Office (TSO).

**ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВО О ГОСУДАРСТВЕННОЙ  
РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

**№ 2020619104**

**Программа многоуровневого управления IT-проектами на  
основе эволюционного моделирования**

Правообладатели: *Вайнилович Юлия Викторовна (ВУ), Андреев  
Андрей Михайлович (ВУ), Емельяненко Станислав Михайлович  
(ВУ), Захарова Алёна Александровна (РУ), Захарченков  
Константин Васильевич (ВУ)*

Авторы: *Вайнилович Юлия Викторовна (ВУ), Андреев Андрей  
Михайлович (ВУ), Емельяненко Станислав Михайлович (ВУ),  
Захарова Алёна Александровна (РУ), Захарченков Константин  
Васильевич (ВУ)*

Заявка № 2020618213

Дата поступления 24 июля 2020 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 11 августа 2020 г.



Руководитель Федеральной службы  
по интеллектуальной собственности

Г.П. Ивлиев

## ПРИЛОЖЕНИЕ Б. АКТЫ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ

УТВЕРЖДАЮ

Проректор по научной работе

В.М. Пашкевич

« 18 »

М.П.

АКТ

о внедрении результатов НИР в образовательный процесс

Настоящий акт составлен об использовании в образовательном процессе разработки «Программный комплекс многоуровневого управления IT-проектами на основе эволюционного моделирования».

Разработка использована в образовательном процессе кафедры «Программное обеспечение информационных технологий».

Программный комплекс многоуровневого управления IT-проектами используется в процессе выполнения выпускных квалификационных работ по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и направления подготовки 09.03.04 «Программная инженерия». В процессе курсового проектирования программный комплекс используется по следующим дисциплинам:

1) «Базы и банки данных» специальности 1 – 53 01 02 «Автоматизированные системы обработки информации».

2) «Программирование», «Объектно-ориентированное программирование», «Базы данных», «Средства взаимодействия человека с вычислительными системами» направления подготовки 09.03.01 «Информатика и вычислительная техника» и направления подготовки 09.03.04 «Программная инженерия».

3) «Технологии разработки программного обеспечения», «Проектирование программного обеспечения», «Архитектура программных систем» направления подготовки 09.03.04 «Программная инженерия».

Программный комплекс многоуровневого управления IT-проектами обеспечивает повышение эффективности образовательного процесса за счет автоматизации формирования команд и распределения участников проекта на задачи с использованием эволюционного моделирования.

Работа выполнена по договору о сотрудничестве между Белорусско-Российским университетом (г.Могилев) и ООО «Стэпл Инк» (г.Минск).

Описание объекта внедрения прилагается и является неотъемлемой частью Акта.

Зам. проректора по НИЧ

В.Н. Усик

Заведующий кафедрой

В.В. Кутузов

Сотрудники, использовавшие разработку:

К.В. Захарченков

Т.В. Мрочек

Ю.В. Вайнилович

О.В. Сергиенко

Визы:

Начальник патентно-информационного отдела

И.В. Брискина

№ регистрации 31/2020 Дата 18.10.2020г

## УТВЕРЖДАЮ

Директор ООО «Стэпл Инк»,

кандидат технических наук

К.А. Шалыжин

2020г.

М.П.

## АКТ

о внедрении результатов НИР в ООО «Стэпл Инк»

Настоящий акт составлен о внедрении в ООО «Стэпл Инк» разработки «Программный комплекс многоуровневого управления IT-проектами». Программный комплекс использован в процессе управления учебными, аутсорсинговыми проектами и стартапами. Программный комплекс управления IT-проектами обеспечивает автоматизацию обработки информации о результатах психологического тестирования участников проектов, поддержку принятия решений при формировании проектных команд и при декомпозиции проекта на задачи, распределение участников проектной команды на задачи проекта на основе алгоритмов эволюционного моделирования, многоуровневый контроль качества решения задач. Разработанный программный комплекс включает следующие программные модули:

- модуль многоуровневого контроля решения проектных задач и модуль обработки результатов многоуровневого контроля решения проектных задач обеспечивают снижение трудоемкости и длительности выполнения IT-проектов за счет автоматизации сбора, хранения и обработки данных о результатах решения проектных задач;

- модуль обработки информации о ЛПК участников проектов, модули формирования структуры и состава проектных команд обеспечивают ускорение подбора рационального состава проектной команды за счет автоматизированной обработки результатов психологического тестирования с использованием кластерного анализа, автоматизации применения алгоритмов эволюционного моделирования для формирования проектной команды;

- модуль распределения участников проектных команд на задачи и модуль формирования состава и структуры задач проекта позволяют ускорить распределение участников проектной команды на задачи за счет автоматизации применения эволюционного моделирования для распределения участников проектной команды на задачи проекта.

Применение разработанного программного комплекса в ООО «Стэпл Инк» позволяет сократить трудоемкость выполнения IT-проектов на 20-30% по сравнению с ручным подбором участников проектных команд и планированием проектных работ.

Работа выполнена по договору о сотрудничестве между Белорусско-Российским университетом (г.Могилев) и ООО «Стэпл Инк» (г.Минск).

Разработчики:

|                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Старший преподаватель кафедры «Автоматизированные системы управления» Белорусско-Российского университета</p> <p> Ю.В. Вайнилович</p> | <p>Студент гр. АСОИР-161 Белорусско-Российского университета</p> <p> С.М. Емельяненко</p> <p>Студент гр. ПИР-161 Белорусско-Российского университета</p> <p> А.М. Андреев</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## ПРИЛОЖЕНИЕ В. ФРАГМЕНТЫ ИСХОДНОГО КОДА РАЗРАБОТАННОГО ПРОГРАММНОГО КОМПЛЕКСА

```
package by.bru.manager.service;
```

```
import by.bru.manager.converter.StudentConverter;
import by.bru.manager.domain.Election;
import by.bru.manager.domain.Student;
import by.bru.manager.domain.Vote;
import by.bru.manager.dto.ElectionDto;
import by.bru.manager.dto.StudentDto;
import by.bru.manager.repository.ElectionRepository;
import by.bru.manager.repository.ProjectRepository;
import by.bru.manager.repository.StudentRepository;
import by.bru.manager.repository.VoteRepository;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.*;
```

```
@Service
```

```
@FieldDefaults(level = AccessLevel.PRIVATE)
```

```
public class ElectionService extends AbstractService<Election, ElectionRepository> {
```

```
    @Autowired
```

```
    StudentConverter studentConverter;
```

```
    @Autowired
```

```
    UserService userService;
```

```
    @Autowired
```

```
    StudentService studentService;
```

```
    @Autowired
```

```
    StudentRepository studentRepository;
```

```
    @Autowired
```

```
    VoteRepository voteRepository;
```

```
    @Autowired
```

```

ProjectRepository projectRepository;

public ElectionService(ElectionRepository repository) {
    super(repository);
}

public ElectionDto getResultsByProjectId(Long id, Long voterId) {
    Election election = repository.findByProjectId(id).orElseThrow(() -> new
RuntimeException(""));
    ElectionDto dto = new ElectionDto();
    List<Vote> votes = election.getVotes();
    Map<StudentDto, Integer> results = new HashMap<>();
    StudentDto voter = null;
    try {
        voter = studentService.getStudentById(voterId);
    } catch (Exception e) {
    }
    for (Vote vote : votes) {
        StudentDto candidate = studentConverter.toDto(vote.getCandidate());
        Integer votesForCandidate = results.get(candidate);
        if (votesForCandidate != null) {
            votesForCandidate++;
            results.put(candidate, votesForCandidate);
        } else {
            results.put(candidate, 1);
        }
        if (voter != null) {
            StudentDto currentVoter = studentConverter.toDto(vote.getVoter());
            if (voter.equals(currentVoter)) {
                dto.setMyCandidate(studentConverter.toDto(vote.getCandidate()));
            }
        }
    }
    dto.setIsFinished(election.getIsFinished());
    dto.setResults(results);
    return dto;
}

```

```

    }

    public ElectionDto vote(Long id, Long voterId, Long candidateId) {
        Optional<Election> myBeElection = repository.findById(id);
        Election election;
        if (myBeElection.isPresent()) {
            election = myBeElection.get();
            if (!election.getIsFinished()) {
                Student voter = studentRepository.findById(voterId).orElseThrow(() -> new
RuntimeException(""));
                Student candidate = studentRepository.findById(candidateId).orElseThrow(() ->
new RuntimeException(""));
                Vote vote = new Vote();
                vote.setCandidate(candidate);
                vote.setVoter(voter);
                List<Vote> votes = election.getVotes();
                votes.add(vote);
                voteRepository.save(vote);
                repository.save(election);
            }
        } else {
            election = new Election();
            election.setIsFinished(false);
            election.setProject(projectRepository.findById(id).orElseThrow(() -> new
RuntimeException(""));
            Student voter = studentRepository.findById(voterId).orElseThrow(() -> new
RuntimeException(""));
            Student candidate = studentRepository.findById(candidateId).orElseThrow(() ->
new RuntimeException(""));
            Vote vote = new Vote();
            vote.setCandidate(candidate);
            vote.setVoter(voter);
            List<Vote> votes = new ArrayList<>();
            votes.add(vote);
            voteRepository.save(vote);
            repository.save(election);
        }
    }

```

```

    }
    return getResultsByProjectId(id, voterId);
}
}

package by.bru.manager.service;

import by.bru.manager.StudentCluster;
import by.bru.manager.converter.GroupConverter;
import by.bru.manager.converter.StudentConverter;
import by.bru.manager.domain.Group;
import by.bru.manager.domain.Project;
import by.bru.manager.domain.Role;
import by.bru.manager.domain.Student;
import by.bru.manager.dto.GroupDto;
import by.bru.manager.dto.StudentDto;
import by.bru.manager.enums.UserRole;
import by.bru.manager.exceptions.UserAlreadyExistsException;
import by.bru.manager.repository.*;
import by.bru.manager.util.ClusterAnalyser;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import javax.transaction.Transactional;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

@Service
@FieldDefaults(level = AccessLevel.PRIVATE)
public class GroupService extends AbstractService<Group, GroupRepository> {
    @Autowired
    GroupConverter groupConverter;
    @Autowired

```

```

StudentService studentService;
@Autowired
UserRepository userRepository;
@Autowired
RoleRepository roleRepository;
@Autowired
StudentRepository studentRepository;
@Autowired
StudentConverter studentConverter;
@Autowired
ClusterAnalyser clusterAnalyser;
@Autowired
ProjectRepository projectRepository;
public GroupService(GroupRepository repository) {
    super(repository);
}
@Transactional
public void createGroup(GroupDto groupDto) {
    Group group = groupConverter.fromDto(groupDto);
    List<Student> students = new ArrayList<>();
    if (group.getStudents() != null) {
        for (StudentDto studentDto : groupDto.getStudents()) {
            students.add(studentRepository.findById(studentDto.getId()).orElseThrow(() ->
                new RuntimeException("")))
        }
        group.setStudents(students);
    }
    repository.save(group);
}

public List<GroupDto> getGroups() {
    List<Group> groups = repository.findAll();
    List<GroupDto> groupDtos = new ArrayList<>();
    for (Group group : groups) {
        groupDtos.add(groupConverter.toDto(group));
    }
}

```

```

    }
    return groupDtos;
}

public void addNewStudent(Long id, StudentDto studentDto) {
    if (userRepository.existsByUsername(studentDto.getUsername())) {
        throw new UserAlreadyExistsException("Username is already taken");
    }
    if (userRepository.existsByEmail(studentDto.getEmail())) {
        throw new UserAlreadyExistsException("Email is already in use");
    }
    Set<Role> roles = new HashSet<>();
    roles.add(roleRepository.findByName(UserRole.ROLE_STUDENT).orElseThrow(()
-> new RuntimeException("")))");
    Student student = studentService.createStudent(studentDto);
    student.setRoles(roles);
    Group group = repository.findById(id).orElseThrow(() -> new
RuntimeException(""));
    List<Student> students = group.getStudents();
    students.add(student);
    repository.save(group);
}

public GroupDto getGroupById(Long id) {
    return groupConverter.toDto(repository.findById(id).orElseThrow(() -> new
RuntimeException(""));
}

public void addStudent(Long groupId, Long studentId) {
    if (!userRepository.existsById(studentId)) {
        throw new RuntimeException("");
    }
    Student student = studentRepository.findById(studentId).get();
    Group group = repository.findById(groupId).orElseThrow(() -> new
RuntimeException(""));
    List<Student> students = group.getStudents();
    students.add(student);
    repository.save(group);
}

```

```

    }

    public List<StudentDto> getGroupStudents(Long id) {
        List<Student> students = repository.findById(id).orElseThrow(() ->
            new RuntimeException("")).getStudents();
        List<StudentDto> studentDtos = new ArrayList<>();
        for (Student student : students) {
            studentDtos.add(studentConverter.toDto(student));
        }
        return studentDtos;
    }

    public StudentCluster getGroupCluster(Long id) {
        Group group = repository.findById(id).orElseThrow(() -> new
        RuntimeException(""));
        return clusterAnalyser.buildKMeansClusterer(group.getStudents());
    }

    public void assignProjectToGroup(Long groupId, Long projectId) {
        Group group = repository.findById(groupId).orElseThrow(() -> new
        RuntimeException(""));
        Project project = projectRepository.findById(projectId).orElseThrow(() -> new
        RuntimeException(""));
        project.setGroup(group);
        projectRepository.save(project);
    }
}

```

```
package by.bru.manager.service;
```

```

import by.bru.manager.StudentCluster;
import by.bru.manager.converter.ProjectConverter;
import by.bru.manager.converter.StudentConverter;
import by.bru.manager.domain.*;
import by.bru.manager.dto.ProjectDto;
import by.bru.manager.dto.StudentDto;
import by.bru.manager.enums.ProjectRoleType;
import by.bru.manager.repository.*;

```

```

import by.bru.manager.util.ClusterAnalyser;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

@Service
@FieldDefaults(level = AccessLevel.PRIVATE)
public class ProjectService extends AbstractService<Project, ProjectRepository> {
    @Autowired
    ClusterAnalyser clusterAnalyser;
    @Autowired
    ProjectConverter projectConverter;
    @Autowired
    GroupRepository groupRepository;
    @Autowired
    SwapRepository swapRepository;
    @Autowired
    StudentConverter studentConverter;
    @Autowired
    ElectionRepository electionRepository;
    @Autowired
    ProjectRoleRepository projectRoleRepository;
    public ProjectService(ProjectRepository repository) {
        super(repository);
    }
    public List<ProjectDto> allocateAutomatically(Long id) {
        Group group = groupRepository.findById(id).orElseThrow(() -> new
RuntimeException(""));
        //TODO sort by priority
        List<Project> projects = repository.findAllByGroupId(group.getId());

```

```
List<Student> students = group.getStudents();
StudentCluster studentCluster = clusterAnalyser.buildKMeansClusterer(students);
```

```
List<Student> clStudents = new ArrayList<>();
clStudents.addAll(studentCluster.getFirstCusterStudents());
clStudents.addAll(studentCluster.getSecondCusterStudents());
clStudents.addAll(studentCluster.getThirdCusterStudents());
clStudents.addAll(studentCluster.getFourthCusterStudents());
int projectCount = projects.size();
int i = 0;
int prN = 0;
for (; i < projectCount; i++) {
    Project project = projects.get(prN);
    List<Student> prStudents = project.getStudents();
    List<ProjectRole> projectRoles = project.getProjectRoles();
    if (project.getTeamSize() > prStudents.size()) {
        prStudents.add(clStudents.get(i));
        project.setStudents(prStudents);
        ProjectRole projectRole = new ProjectRole();
        projectRole.setStudent(clStudents.get(i));
        projectRole.setName(ProjectRoleType.ARCHITECT);
        projectRoles.add(projectRole);
        project.setProjectRoles(projectRoles);
    }
    prN++;
}
prN = 0;
projectCount += projects.size();
for (; i < projectCount; i++) {
    Project project = projects.get(prN);
    List<Student> prStudents = project.getStudents();
    List<ProjectRole> projectRoles = project.getProjectRoles();
    if (project.getTeamSize() > prStudents.size()) {
        prStudents.add(clStudents.get(i));
        project.setStudents(prStudents);
```

```

        ProjectRole projectRole = new ProjectRole();
        projectRole.setStudent(clStudents.get(i));
        projectRole.setName(ProjectRoleType.DEVELOPER);
        projectRoles.add(projectRole);
        project.setProjectRoles(projectRoles);
    }
    prN++;
}
prN = 0;
projectCount += projects.size();
for (; i < projectCount; i++) {
    Project project = projects.get(prN);
    List<Student> prStudents = project.getStudents();
    List<ProjectRole> projectRoles = project.getProjectRoles();
    if (project.getTeamSize() > prStudents.size()) {
        prStudents.add(clStudents.get(i));
        project.setStudents(prStudents);
        ProjectRole projectRole = new ProjectRole();
        projectRole.setStudent(clStudents.get(i));
        projectRole.setName(ProjectRoleType.TESTER);
        projectRoles.add(projectRole);
        project.setProjectRoles(projectRoles);
    }
    prN++;
}
prN = 0;
projectCount += projects.size();
for (; i < projectCount; i++) {
    Project project = projects.get(prN);
    List<Student> prStudents = project.getStudents();
    List<ProjectRole> projectRoles = project.getProjectRoles();
    if (project.getTeamSize() > prStudents.size()) {
        prStudents.add(clStudents.get(i));
        project.setStudents(prStudents);
        ProjectRole projectRole = new ProjectRole();

```

```

        projectRole.setStudent(clStudents.get(i));
        projectRole.setName(ProjectRoleType.DESIGNER);
        projectRoles.add(projectRole);
        project.setProjectRoles(projectRoles);
    }
    prN++;
}
for (; i < clStudents.size(); i++) {
    for (Project project : projects) {
        List<Student> prStudents = project.getStudents();
        List<ProjectRole> projectRoles = project.getProjectRoles();
        if (project.getTeamSize() > prStudents.size()) {
            prStudents.add(clStudents.get(i));
            project.setStudents(prStudents);
            ProjectRole projectRole = new ProjectRole();
            projectRole.setStudent(clStudents.get(i));
            projectRole.setName(ProjectRoleType.MEMBER);
            projectRoles.add(projectRole);
            project.setProjectRoles(projectRoles);
        } else {
            continue;
        }
    }
}

repository.saveAll(projects);
return projects.stream().map(project ->
projectConverter.toDto(project)).collect(Collectors.toList());
}

public List<ProjectDto> getProjects() {
    List<Project> projects = repository.findAll();
    List<ProjectDto> projectDtos = new ArrayList<>();
    for (Project project : projects) {
        projectDtos.add(projectConverter.toDto(project));
    }
}

```

```

    }
    return projectDtos;
}

public void editProjectById(Long id, ProjectDto dto) {
    Project project = repository.findById(id).orElseThrow(() ->
        new RuntimeException(""));
    Project updated = projectConverter.fromDto(dto);
    repository.save(updated);
}

public ProjectDto getProjectById(Long id) {
    return projectConverter.toDto(repository.findById(id).orElseThrow(() ->
        new RuntimeException("")));
}

public void createProject(ProjectDto projectDto) {
    Election election = new Election();
    election.setIsFinished(false);
    election.setProject(repository.save(projectConverter.fromDto(projectDto)));
    election.setVotes(new ArrayList<>());
    electionRepository.save(election);
}

public Swap swapStudents(Long projectFromId, Long studentFromId, Long
projectToId, Long studentToId) {
    Project projectFrom = repository.findById(projectFromId).orElseThrow(() -> new
RuntimeException(""));
    Project projectTo = repository.findById(projectToId).orElseThrow(() -> new
RuntimeException(""));
    Student studentFrom = projectFrom.getStudents().stream().filter(s ->
s.getId().equals(studentFromId))
        .findFirst().orElseThrow(() -> new RuntimeException(""));
    Student studentTo = projectTo.getStudents().stream().filter(s ->
s.getId().equals(studentToId))
        .findFirst().orElseThrow(() -> new RuntimeException(""));
    Optional<Swap> maybeSwapFromStudentA = swapRepository

```

```

.findByStudentFromIdAndProjectFromIdAndStudentToIdAndProjectToId(studentFromId,
projectFromId, studentToId, projectToId);
    if (!maybeSwapFromStudentA.isPresent()) {
        Swap swap = new Swap();
        swap.setProjectFrom(projectFrom);
        swap.setProjectTo(projectTo);
        swap.setStudentFrom(studentFrom);
        swap.setStudentTo(studentTo);
        swapRepository.save(swap);
        return swap;
    }
    Optional<Swap> maybeSwapFromStudentB = swapRepository
        .findByStudentFromIdAndProjectFromIdAndStudentToIdAndProjectToId(studentToId,
projectToId, studentFromId, projectFromId);
    if (!maybeSwapFromStudentB.isPresent()) {
        Swap swap = new Swap();
        swap.setProjectFrom(projectTo);
        swap.setProjectTo(projectFrom);
        swap.setStudentFrom(studentTo);
        swap.setStudentTo(studentFrom);
        swapRepository.save(swap);
    }
    projectFrom.getStudents().remove(studentFrom);
    projectFrom.getStudents().add(studentTo);
    projectTo.getStudents().remove(studentTo);
    projectTo.getStudents().add(studentFrom);
    repository.save(projectFrom);
    repository.save(projectTo);
    return null;
}

public void changeStudentsByTeacher(Long projectFromId, Long studentFromId, Long
projectToId, Long studentToId) {
    Project projectFrom = repository.findById(projectFromId).orElseThrow(() -> new
RuntimeException(""));

```

```

        Project projectTo = repository.findById(projectToId).orElseThrow(() -> new
RuntimeException(""));
        Student studentFrom = projectFrom.getStudents().stream().filter(s ->
s.getId().equals(studentFromId))
            .findFirst().orElseThrow(() -> new RuntimeException(""));
        Student studentTo = projectTo.getStudents().stream().filter(s ->
s.getId().equals(studentToId))
            .findFirst().orElseThrow(() -> new RuntimeException(""));
        projectFrom.getStudents().remove(studentFrom);
        projectFrom.getStudents().add(studentTo);
        projectTo.getStudents().remove(studentTo);
        projectTo.getStudents().add(studentFrom);
        repository.save(projectFrom);
        repository.save(projectTo);
    }

    public List<StudentDto> getStudentsByProjectId(Long id) {
        List<Student> students = repository.findById(id).orElseThrow(() -> new
RuntimeException("")).getStudents();
        List<StudentDto> studentDtos = new ArrayList<>();
        for (Student student : students) {
            studentDtos.add(studentConverter.toDto(student));
        }
        return studentDtos;
    }
}

package by.bru.manager.service;

import by.bru.manager.converter.ProjectConverter;
import by.bru.manager.converter.StudentConverter;
import by.bru.manager.domain.Project;
import by.bru.manager.domain.Student;
import by.bru.manager.dto.ProjectDto;
import by.bru.manager.dto.StudentDto;
import by.bru.manager.repository.StudentRepository;
import lombok.AccessLevel;

```

```

import lombok.experimental.FieldDefaults;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
import java.util.stream.Collectors;

@Service
@FieldDefaults(level = AccessLevel.PRIVATE)
public class StudentService extends AbstractService<Student, StudentRepository> {
    @Autowired
    StudentConverter studentConverter;
    @Autowired
    ProjectConverter projectConverter;
    public StudentService(StudentRepository repository) {
        super(repository);
    }
    public Student createStudent(StudentDto studentDto) {
        return repository.save(studentConverter.fromDto(studentDto));
    }
    public List<StudentDto> getStudents() {
        return
repository.findAll().stream().map(StudentDto::new).collect(Collectors.toList());
    }
    public StudentDto getStudentById(Long id) {
        return studentConverter.toDto(repository.findById(id).orElseThrow(() ->
            new RuntimeException(""));
    }
    public List<ProjectDto> getProjectsByStudentId(Long id) {
        List<Project> projects = repository.findById(id).orElseThrow(() -> new
RuntimeException("")).getProjects();
        return
            projects.stream().map(p
projectConverter.toDto(p)).collect(Collectors.toList());
    }
}

```

```
package by.bru.manager.service;
```

```
import by.bru.manager.converter.TaskConverter;
import by.bru.manager.domain.Project;
import by.bru.manager.domain.Task;
import by.bru.manager.dto.TaskDto;
import by.bru.manager.repository.ProjectRepository;
import by.bru.manager.repository.TaskRepository;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;
```

```
@Service
```

```
@FieldDefaults(level = AccessLevel.PRIVATE)
```

```
public class TaskService extends AbstractService<Task, TaskRepository> {
```

```
    @Autowired
```

```
    TaskConverter converter;
```

```
    @Autowired
```

```
    ProjectRepository projectRepository;
```

```
    public TaskService(TaskRepository repository) {
```

```
        super(repository);
```

```
    }
```

```
    public List<TaskDto> getTasks() {
```

```
        return repository.findAll().stream().map(task
```

```
->
```

```
converter.toDto(task)).collect(Collectors.toList());
```

```
    }
```

```
    public TaskDto getTaskById(Long id) {
```

```
        Optional<Task> task = repository.findById(id);
```

```
        if (!task.isPresent()) {
```

```
            throw new RuntimeException("");
```

```
        }
```

```

        return converter.toDto(task.get());
    }

    public void editTaskById(Long id, TaskDto taskDto) {
        Optional<Task> task = repository.findById(id);
        if (!task.isPresent()) {
            throw new RuntimeException("");
        }
        Task editedTask = converter.fromDto(taskDto);
        repository.save(editedTask);
    }

    public void deleteTaskById(Long id) {
        repository.deleteById(id);
    }

    public List<TaskDto> getTasksByProject(Long id) {
        return repository.findAllByProjectId(id).stream().map(task ->
converter.toDto(task)).collect(Collectors.toList());
    }

    public void createTasksByProject(Long id, TaskDto taskDto) {
        Project project = projectRepository.findById(id).orElseThrow(() -> new
RuntimeException(""));
        Task task = converter.fromDto(taskDto);
        task.setProject(project);
        repository.save(task);
    }
}

package by.bru.manager.service;

import by.bru.manager.domain.Team;
import by.bru.manager.repository.TeamRepository;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;

```

```

import org.springframework.stereotype.Service;

@Service
@FieldDefaults(level = AccessLevel.PRIVATE)
public class TeamService extends AbstractService<Team, TeamRepository> {

    public TeamService(TeamRepository repository) {
        super(repository);
    }

}

package by.bru.manager.service;

import by.bru.manager.domain.User;
import by.bru.manager.repository.UserRepository;
import lombok.AccessLevel;
import lombok.experimental.FieldDefaults;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
@FieldDefaults(level = AccessLevel.PRIVATE)
public class UserService extends AbstractService<User, UserRepository> {

    public UserService(UserRepository repository) {
        super(repository);
    }

    public List<User> getUsers() {
        return repository.findAll();
    }

}

package by.bru.manager.util;

import by.bru.manager.StudentCluster;
import by.bru.manager.domain.Student;
import org.springframework.stereotype.Component;

```

```

import weka.clusterers.SimpleKMeans;
import weka.core.Attribute;
import weka.core.DenseInstance;
import weka.core.Instance;
import weka.core.Instances;
import java.util.ArrayList;
import java.util.List;

@Component
public class ClusterAnalyser {
    private static final String FILENAME = "Just path to file";
    public StudentCluster buildKMeansClusterer(List<Student> students) {
        SimpleKMeans kmeans = new SimpleKMeans();
        StudentCluster studentCluster = new StudentCluster();
        kmeans.setSeed(10);
        //important parameter to set: preserve order, number of cluster.
        kmeans.setPreserveInstancesOrder(true);
        try {
            kmeans.setNumClusters(4);
            Instances data = new Instances("Test results", getAttributes(students), 0);
            for (Student student : students) {
                double test1res = student.getTests().get(0).getResult();
                double test2res = student.getTests().get(1).getResult();
                double test3res = student.getTests().get(2).getResult();
                double test4res = student.getTests().get(3).getResult();
                Instance data1 = new DenseInstance(4);
                data1.setValue(0, test1res);
                data1.setValue(1, test2res);
                data1.setValue(2, test3res);
                data1.setValue(3, test4res);
                data.add(data1);
            }

            kmeans.buildClusterer(data);
            // This array returns the cluster number (starting with 0) for each instance

```

```

        // The array has as many elements as the number of instances
        int[] assignments = kmeans.getAssignments();
        int i = 0;
        for (int clusterNum : assignments) {
            System.out.printf("Instance %d -> Cluster %d \n", i, clusterNum);
            studentCluster.addStudentToCluster(students.get(i), clusterNum);
            i++;
        }
    } catch (Exception e) {
        throw new RuntimeException(e.getMessage());
    }
    return studentCluster;
}

private ArrayList<Attribute> getAttributes(List<Student> students) {
    ArrayList<Attribute> attributes = new ArrayList<>();
    Attribute test1 = new Attribute("test1");
    Attribute test2 = new Attribute("test2");
    Attribute test3 = new Attribute("test3");
    Attribute test4 = new Attribute("test4");
    attributes.add(test1);
    attributes.add(test2);
    attributes.add(test3);
    attributes.add(test4);
    return attributes;
}
}

package by.bru.manager;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class ProjectManagerApplication {
    public static void main(String[] args) {
        SpringApplication.run(ProjectManagerApplication.class, args);    }}

```