

На правах рукописи



Тележкин Александр Михайлович

**Применение алгоритмических сетей для оценки
ресурсов в программных проектах**

Специальность 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации на соискание ученой степени кандидата
технических наук

Санкт-Петербург – 2015

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего образования Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики (Университет ИТМО)

Научные руководители:

Морозов Владимир Петрович

кандидат технических наук, доцент, старший научный сотрудник

Баранов Сергей Николаевич

доктор физико-математических наук, профессор

Официальные оппоненты:

Водяхо Александр Иванович

Доктор технических наук, профессор кафедры вычислительной техники, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина) (СПбГЭТУ «ЛЭТИ»), профессор кафедры вычислительной техники

Кознов Дмитрий Владимирович

Кандидат физико-математических наук, доцент кафедры системного программирования, Санкт-Петербургский государственный университет (СПбГУ), доцент кафедры системного программирования

Ведущая организация:

ПАО «Информационные телекоммуникационные технологии» (ПАО «Интелтех»)

Защита состоится « 8 » декабря 2015 года в 14:00 на заседании диссертационного совета Д 002.199.01 при Санкт-Петербургском институте информатики и автоматизации Российской академии наук по адресу: г. Санкт-Петербург, 14 линия В.О., д. 39.

С диссертацией можно ознакомиться в библиотеке и на сайте диссертационного совета Санкт-Петербургского института информатики и автоматизации Российской академии наук, <http://www.spiras.nw.ru/dissovvet/telegkin>

Автореферат разослан « ____ » _____ 2015 г.

Ученый секретарь
диссертационного совета
Д.002.199.01, к.т.н., доцент

Фаткиева Роза Равильевна

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Точная, надежная и своевременная оценка ресурсов, необходимых для выполнения проектов разработки программного обеспечения (ПО), является важным и нетривиальным процессом, требующим глубоких теоретических и практических (в том числе и плохо формализуемых) знаний. По данным экспертов, 70% программных проектов завершаются с существенно иными, чем запланированными, параметрами качества, бюджета и графика, либо не завершаются вообще. Одной из причин данной ситуации является некачественная оценка необходимых ресурсов при запуске и дальнейшем перепланировании проекта.

В свою очередь, некачественная оценка возникает вследствие неадекватной методики ее выполнения при недостаточности отводимых на нее сроков и неполноты исходной информации. При этом компании, разрабатывающие ПО, как правило, накапливают реальную информацию об уже выполненных ими проектах, но эти ценные данные не всегда анализируются и используются для оценки ресурсов в новом проекте. На этапе запуска проекта руководители выполняют оценку и планирование без учета этого опыта, часто повторяя ошибки совершенные ранее.

На данный момент существует большое количество моделей и методов для оценки ресурсов, необходимых для выполнения проектов разработки ПО, их можно разделить на две группы – алгоритмические и неалгоритмические. Алгоритмические модели (COCOMO и COCOMO II, SLIM, SEER-SEM, анализ функциональных точек и другие) характеризуются наличием некоего формального аппарата для подсчета необходимых ресурсов исходя из некоторых начальных данных о будущем программном изделии (ПИ) и команде его разработчиков. Основным преимуществом методов этого типа является обоснованность и повторяемость результата, получаемого в результате вычислений. Неалгоритмические модели (оценка по Паркинсону, экспертная оценка, оценка по аналогу и другие) характеризуются своими методиками, схемами и принципами подсчета, основанными на привлечении опыта и знаний эксперта, плохо поддающихся формализации. Но, несмотря на разнообразие методов и моделей оценки, до сих пор не существует единого общепринятого множества характеристик, используемого для оценки необходимых ресурсов.

В сложившейся ситуации является актуальным создание метода оценки ресурсов для выполнения проектов разработки ПИ, который бы работал в условиях неполноты и неточности исходных данных, а также совмещал бы преимущества алгоритмических и неалгоритмических методов. В качестве формального аппарата для метода предлагается использовать алгоритмические сети, т.к. их формализм, с одной стороны, можно рассматривать как полноценный язык высокого уровня, а с другой стороны, этот формализм ориентирован на конечного пользователя, не обладающего специальными знаниями в области программирования. Подходы на основе

алгоритмических сетей хорошо зарекомендовали себя в таких областях, как теория моделирования, моделирование экологических систем, экономика, параллельные вычисления. Таким образом, применение алгоритмических сетей к оценке ресурсов, необходимых для реализации планируемого программного проекта, является актуальным.

Цель работы: повышения точности и оперативности оценки ресурсов, необходимых программному проекту.

Решаемая научная задача: разработка на базе формализма алгоритмических сетей автоматизированного метода для обоснования оценок необходимых программному проекту ресурсов, который бы позволил увеличить точность и оперативность такой оценки. Метод должен быть ориентирован на эксперта, имеющего целостное представление о предметной области, но при этом не обладающего специальными знаниями в программировании.

В интересах достижения цели работы и решения научной задачи осуществлялись:

- сравнительный анализ существующих подходов (методы, модели, программные средства) оценки ресурсов, необходимых для разработки ПО в рамках программного проекта;
- разработка метода и технологии оценки необходимых ресурсов (метод гибких оценок) на базе формализма алгоритмических сетей;
- проектирование, разработка и апробация системы для создания исторических баз данных по выполненным разработкам, которая реализует метод гибких оценок.

Методы исследования. При выполнении диссертационного исследования использовались методы математического моделирования на основе алгоритмических сетей, теории распознавания образов, методы и модели теории принятия решений.

Основные положения, выносимые на защиту. На основе проведенных теоретических работ и их экспериментальной апробации на защиту выносятся следующие положения:

- 1) модель формирования базы выполненных проектов для поиска проектов-аналогов.
- 2) метод формирования пространства характеристик для оценки ресурсов, необходимых для выполнения проектов разработки программных изделий на основе алгоритмических сетей (метод гибких оценок).
- 3) модель программной системы для автоматизированного поиска ближайших проектов-аналогов по базе выполненных проектов для оценки необходимых ресурсов по принципу подобия.

Научная новизна работы.

Новизна первого результата состоит в разработке модели формирования базы выполненных проектов с учетом неопределенного характера информации о значениях характеристик, описывающих проекты,

неполноты исходных данных, слабо формализуемого опыта и высокой квалификация экспертов, проводящих оценку.

Особенность второго научного результата заключается в специфике использования формализма алгоритмических сетей для формирования пространства характеристик используемого для оценки ресурсов, необходимых для выполнения проектов разработки программных изделий (метод гибких оценок). Данный формализм используется для привлечения опыта экспертов, плохо поддающегося формализации, но при этом не требующего от самих экспертов специальных знаний в области программирования.

Отличительным признаком третьего научного результата выступают процедура и система для автоматизированного поиска проекта-аналога ранее выполнявшихся в компании для использования фактических данных этих проектов в обосновании оценок ресурсов нового проекта по принципу подобия. Особенностью процедуры поиска проекта-аналога является использование предварительно подготовленной (отфильтрованной) БД на основе количественных и качественных характеристик метрик. Помимо этого, для использования в системе было собрано и классифицировано множество, содержащее более 100 метрик, характеризующих программный проект, создаваемый продукт и процесс разработки, учет которых позволяет производить обоснованную оценку ресурсов, необходимых программному проекту.

Теоретическая и практическая ценность. Теоретическую ценность диссертационной работы составляют разработанная модель формирования базы выполненных проектов и метод гибких оценок, использующий класс алгоритмов распознавания, основанных на вычислении оценок (АВО), предложенный Ю.И. Журавлевым.

Практическая ценность диссертационной работы состоит в расширении области применения формализма алгоритмических сетей и в повышении точности и объективности оценки ресурсов, необходимых для выполнения проектов разработки программных изделий, при сокращении времени на получение такой оценки.

Реализация и внедрение результатов исследования

- Результаты, полученные в ходе исследования, нашли свое отражение в виде системы САМПО+ для поддержки создания баз выполненных проектов, которая является надстройкой над Microsoft Excel и реализована на языке Visual Basic for Applications (VBA).
- С 2009 по 2010 гг. система САМПО+ использовалась в процессе оценки ресурсов для проектов разработки ПО компании ООО «Эквиджен Сервисес» (в конце 2009 года компания претерпела изменения и стала именоваться «Ф-Лайн Софтвер»). С 2009 по 2010 гг. система САМПО+ использовалась в процессе преподавания курса «Проектирование информационных систем» в Санкт-

Петербуржском университете водных коммуникаций. С 2011 по 2014 гг. метод формирования пространства характеристик и множество метрических характеристик программных проектов использовались в НП «Объединение подземных строителей» для разработки и внедрения системы автоматизации документооборота «Эскулап».

Личный вклад автора. Научные результаты, представленные в диссертационной работе, получены соискателем либо самостоятельно, либо при непосредственном участии.

Публикация и апробация работы. По теме диссертации опубликовано 7 работ, в том числе 2 из них в изданиях, рекомендованных ВАК. Основные положения, выносимые на защиту, результаты исследования и выводы, содержащиеся в диссертационной работе, обсуждались в на научных семинарах в Санкт-петербургском институте информатики и автоматизации Российской академии наук, а также на конференциях: XI Санкт-Петербургская Международная конференция «Региональная информатика (РИ-2008)» (Санкт-Петербург, 2008), Четвертая всероссийская научно-практическая конференция «Имитационное моделирование. Теория и практика » (ИММОД-2009) (Санкт-Петербург, 2009), XII Санкт-Петербургская Международная конференция «Региональная информатика (РИ-2010)» (Санкт-Петербург, 2010), Юбилейная XIII Санкт-Петербургская Международная конференция «Региональная информатика (РИ-2012)» (Санкт-Петербург, 2012).

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, определены цели, задачи и методы проведенных исследований, отражена научная новизна, практическая и теоретическая значимость полученных результатов, а также представлена информация о реализации и внедрении результатов исследования.

В первой главе проведен анализ методов и программных средств для оценки ресурсов, кратко рассмотрен формализм алгоритмических сетей (ЯАС – язык алгоритмических сетей) и представлена постановка задачи.

Существует большое количество методов и моделей для оценки ресурсов как специфичных для области разработки ПО, так и подходящих для всех отраслей производства. Условно их можно разделить на алгоритмические и неалгоритмические методы оценки ресурсов.

В основе любого алгоритмического метода оценки ресурсов, необходимых программному проекту, лежит одна или несколько функций, описывающая зависимость характеристик проекта (оценочные значения размера продукта, различные отраслевые коэффициенты) от затрат на его выполнение. В качестве примеров алгоритмических методов оценки ресурсов ПО можно привести СОСОМО / СОСОМО II, SLIM (модель Путнэма), SEER-SEM и пр. К основным преимуществам использования алгоритмических методов можно отнести: повторяемость результата, обоснованность оценки и наличие формального аппарата для расчета

оценок. К недостаткам: модели отражают специфику той предметной области, в которой проводилась разработка и апробация данных методов, то есть они работают на «своем поле». Помимо этого в каждом из методов присутствуют настроечные коэффициенты, значения которых необходимо «подгонять» для более точной оценки.

Неалгоритмические методы основываются не на математических формулах, а на основополагающих принципах и схемах. Примеры неалгоритмических методов: оценка по Паркинсону, экспертная оценка, оценка по аналогу, Oracle AIM и др. К основным преимуществам неалгоритмических методов можно отнести возможность привлечения слабо формализуемых знания эксперта, к недостаткам – слабую обоснованность полученной оценки, а также необходимость согласования при наличии нескольких оценок.

Как правило, функцию для оценки ресурсов не выделяют в отдельную программу, так как она представляет один из шагов процесса разработки ПО и является частью ПО для управления проектами. Программные системы для управления проектами можно разделить на 3 группы по типу используемой технологии:

- 1) настольные системы (desktop). Примеры: Deltec Open Plan, семейство Artemis Views, Spider Project, Microsoft Project;
- 2) клиент-серверные (client-server). Примеры: Basecamp, JIRA;
- 3) веб-сервисы (web-based). Примеры: Wrike, Worksection.

Программные системы оценки ресурсов позволяют отслеживать выполнение проекта в течение всего его жизненного цикла, следить за установленными параметрами графика и бюджета; они обеспечивают расчет критического пути, построение диаграммы Ганта, но не отражают специфику той организации, в которой они применяются.

Учитывая эти факты, является целесообразным разработка метода, который бы объединял преимущества алгоритмических и неалгоритмических методов оценки ресурсов, то есть использовал бы некоторый формальный аппарат для подсчета ресурсов и привлекал слабо формализуемые знания эксперта, а также отражал бы специфику той организации, в которой применяется.

Вторая глава содержит описание модели формирования базы выполненных проектов для оценки необходимых программному проекту ресурсов при помощи поиска проекта-аналога (метод гибких оценок). Рассмотрены процедуры формирования исходного и уточненного множеств источников, характеристик и проектов.

Модель включает следующие этапы:

- 1) формирование *исходного множества источников*, в которых может (должна) содержаться информация о выполняемых (выполненных) проектах, их содержании, особенностях и ходе;
- 2) формирование и структуризация *исходного множества характеристик*, содержащих информацию о выполняемых

- (выполненных) проектах, их содержании, особенностях и ходе выполнения;
- 3) формирование *исходного множества выполнявшихся проектов*, обеспеченных источниками информации и перечня источников для каждого проекта;
 - 4) уточнение исходного множества характеристик, исходя из степени их обеспеченности проектной информацией и представления об их значимости для поиска проектов-аналогов – формирование *уточненного множества характеристик*;
 - 5) уточнение исходного множества проектов, исходя из степени их обеспеченности информацией уточненного множества характеристик – формирование *уточненного множества проектов*;
 - 6) уточнение исходного множества источников, исходя из степени их использования в уточненных множествах проектов и характеристик – формирование *уточненного множества источников*;
 - 7) проведение испытаний, подтверждающих достаточность уточненного множества характеристик – формирование *рекомендуемых множеств источников, характеристик и проектов*;

На рисунке 1 представлена модель формирования базы выполненных проектов (БВП).

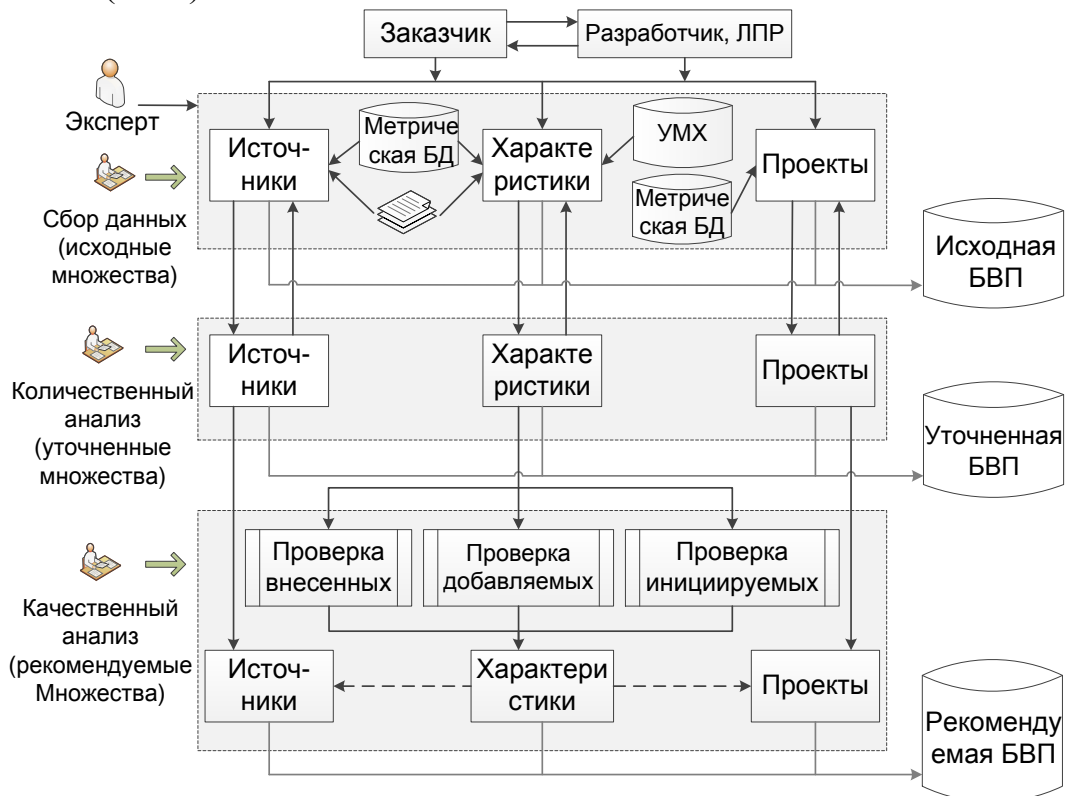


Рис. 1. Модель формирования базы выполненных проектов

Для того чтобы сформировать базу выполненных проектов, основанную на рекомендуемых множествах, необходимо выполнить 3 этапа анализа:

- вербальный — формирование, исходя из экспертного множества, исходных множеств источников, характеристик и проектов на основе анализа методической и специальной литературы, а также бизнес-процессов компании;
- количественный — формирование уточненных множеств источников, характеристик и проектов на основе анализа обеспеченности каждого из них проектной информацией и представления об их значимости для поиска проектов-аналогов;
- качественный — формирование рекомендуемых множеств источников, характеристик и проектов на основе экспериментов, подтверждающих достаточность сформированного в ходе предыдущих процессов множества характеристик, для решения задачи распознавания (под задачей распознавания в данном случае принимается поиск проекта-аналога).



Рис. 2. Обобщенный алгоритм формирования пространства характеристик

В **третье** главе рассматривается метод формирования пространства характеристик и этапы исследования функциональной пригодности сформированного пространства характеристик.

Функциональная пригодность базы данных – это уровень достижения целей, функций и назначения баз данных информацией, находящейся в базе данных.

После того как было сформировано уточненное (обеспеченное информацией) множество характеристик, формирование рекомендуемого множества характеристик предлагается рассматривать как последовательность следующих этапов исследования:

1) проверка функциональной пригодности информации относительно *внесённых* в базу выполненных проектов (то есть тех, которые уже находятся в базу выполненных проектов).

2) проверка функциональной пригодности информации относительно *добавляемых* в базу выполненных проектов (то есть тех, которые по завершении вносятся в базу выполненных проектов).

3) проверка функциональной пригодности информации относительно *иницируемых* проектов (то есть тех, для которых известны только прогнозные значения некоторых характеристик, определяемых экспертом).

На рисунке 2 представлена обобщенная схема алгоритма формирования пространства характеристик.

Информация, хранящаяся в базе исторических данных, является функционально пригодной относительно **выполненных** проектов, если

классификация проектов, полученная после применения процедур кластеризации *на всех* предметно значимых поисковых наборах, согласуется с мнением экспертов. В случае выявленного несогласия предпринимается расширение состава характеристик базы и повторное исследование на полноту.

Процедура проверка относительно **выполненных** проектов:

1. Формируется множество «минимальных» классификаторов (алгоритм формального создания классификаторов).
2. Выявляются кандидаты в предметно значимые классификаторы (экспертная оценка возможности предметной интерпретации того или иного элемента из ранее полученного множества классификаторов и правильность осуществляемой им классификации).
3. Формируется множество предметно значимых классификаторов (алгоритм экспертного создания классификаторов).
4. Определяется полнота полученных классификаторов (анализ мнений экспертов о достаточности полученного множества для решения задач, стоящих перед формируемой базой и, прежде всего, задачи поиска проектов-аналогов).
5. В зависимости от полученных результатов реализуются различные завершающие этапы:
 - если множество предметных классификаторов, по мнению экспертов, полно, формируется *исходное опорное множество характеристик* путем объединения множеств всех предметно значимых наборов (успешное завершение проверки полноты);
 - если множество предметных классификаторов, по мнению экспертов, не полно, осуществляется попытка его расширения в рамках уточненного множества характеристик с повторным анализом полноты (попытка добиться полноты в рамках исследуемого множества характеристик);
 - если множество предметных классификаторов, по мнению экспертов, не полно и попытка его расширения в рамках уточненного множества характеристик не привела к желаемому результату, осуществляется расширение уточненного множества характеристик, с последующим сбором информации о значениях и повторным исследованием функциональной пригодности информации (достижение полноты за счёт расширения исследуемого множества характеристик).

Информация, хранящаяся в базе исторических данных, является функционально пригодной относительно **добавляемого** проекта, если исследуемый проект может быть вписан в существующие классификаторы. Признаком того, что проект может быть вписан в классификатор, является наличие проекта-аналога, уже существующего в базе. Добавление проекта происходит либо путём включения в существующий класс, либо путём создания нового класса в существующем классификаторе. В случае если

проект не может быть вписан, предпринимается расширение состава характеристик базы и повторное исследование ее на полноту.

В рамках подхода, ориентированного на привлечение предметных знаний эксперта, с помощью класса алгоритмов распознавания, основанных на вычислении оценок (АВО), предложенного Ю.И. Журавлевым в 70-х годах и модифицированного для решения поставленной задачи поиска проектов-аналогов, формируется множество проектов (рекомендуемое множество), которые представляются эксперту в качестве возможных аналогов исследуемому проекту. При этом исключительно прерогативой эксперта является:

- *выбор проекта-аналога* исследуемому проекту (система формирует рекомендуемое множество проектов, а также сравнительную информацию о не совпавших входящих в классифицирующий ансамбль значениях характеристик);
- *уточнение значений* характеристик исследуемого проекта;
- *определение корректности проведенной классификации* в рамках выбранного классификатора с точки зрения эксперта (система предоставляет информацию обо всех не совпавших значениях характеристик);
- *подтверждение возможности расширения* выбранного классификатора путём включения в него нового класса (система предоставляет информацию обо всех не совпавших значениях характеристик).

Процедура проверки относительно **добавляемых** проектов:

1. Вводятся значения *всех* характеристик добавляемого проекта (эксперт).
2. Формируется множество проектов, рекомендуемых системой в качестве возможного аналога добавляемому проекту (алгоритм формирования рекомендуемого множества).
3. Анализируется информация о не совпавших значениях характеристик проектов рекомендуемого множества и добавляемого проекта (эксперт).
4. Выбирается проект, являющийся, по мнению эксперта, кандидатом в аналоги добавляемому (эксперт).
5. Уточняются или не уточняются значения характеристик исследуемого проекта в соответствии со значениями выбранного аналога (эксперт).
6. Определяется правильность отнесения добавляемого проекта к выбранному классу либо возможность расширения анализируемого классификатора на основе анализа значений прочих не совпавших характеристик (эксперт).
7. В зависимости от полученных результатов реализуются различные завершающие этапы:

- если, мнение экспертов положительное, но просмотрены не все добавляемые проекты, переход к следующему проекту и повторение ранее описанных этапов (1 – 7);
- если, мнение экспертов положительное и просмотрены все исследуемые проекты, успешное завершение проверки полноты;
- если, мнение экспертов отрицательное, и попыток решения проблемы за счет расширения множества предметных классификаторов не предпринималось, осуществляется расширение множества предметных классификаторов с повторным анализом его полноты;
- если, мнение экспертов отрицательное, и попытка решения проблемы за счет расширения множества предметных классификаторов не привела к желаемому результату, осуществляется расширение уточненного множества характеристик, сбор информации о значениях и повторное исследование функциональной пригодности информации относительно выполненных проектов.

Информация, хранящаяся в базе исторических данных, является функционально пригодной относительно **инициируемого** проекта, если иницируемому проекту может быть сопоставлен проект-аналог, выполнявшийся ранее. Если для какого-либо проекта проект-аналог не находится, то происходит расширение состава характеристик базы и повторное исследование на полноту.

Процедура проверки относительно **инициируемых** проектов соответствует процедуре проверки относительно добавляемых проектов, за исключением того, эксперт определяет множество характеристик, описывающих иницируемый проект и их значения.

Неопределённость информации, которой располагает пользователь на этапе инициации проекта, обусловила необходимость модификации исходного алгоритма АВО, с целью привлечения в процесс принятия решения дополнительных плохо формализуемых знаний об объекте исследования. В ходе модификации АВО принято решение:

- отказаться от использования весов характеристик;
- сформулировать решающее правило в форме, позволяющей пользователю самому принять окончательное решение о принадлежности исследуемого объекта к одному из определенных в качестве альтернатив классу;
- представлять пользователю информацию не только о характеристиках, по которым, произошло совпадение, но и характеристиках, по которым это совпадение не произошло. Что позволяет включить в процесс принятия решения плохо формализуемое представление пользователя о важности совпадения или несовпадения тех или иных значений ансамблей характеристик.

Опорное множество (S_i) – некоторое подмножество характеристик, которое используется для оценки степени близости проектов содержащихся в базе выполненных проектов и иницируемого.

Система опорных множеств (S_A) – совокупность опорных множеств, полученная в результате кластеризации: $S_A = \{S_1, S_2 \dots, S_z\}$.

Редукция опорного множества – переход от подмножества характеристик числом k к подмножеству характеристик числом $(k-n)$, где n — шаг редукции.

Начальное опорное множество (S_1) – опорное множество, из которого путем редукции формируются все опорные множества, в совокупности составляющие систему опорных множеств (S_A).

Правило (функция) близости – оценивает сходство иницируемого проекта (ω'), и соответствующего ему выполненного проекта в базе выполненных проектов (Ω_r). Возможные определения: проекты считаются похожими, если выполняется не менее чем δ неравенств вида $|\alpha_j - \beta_j| \leq \varepsilon_j, j = 1, \dots, k$. Здесь α_j, β_j — значение j -ой характеристики иницируемого и выполненного проектов, ε_j — порог близости характеристик, δ — порог близости сравниваемых проектов (минимальное число признаков, при совпадении которых проекты считаются похожими), k — общее число характеристик, по которым производится сравнение. Определение: полное совпадение значений всех характеристик выполненного (Ω_r) и иницируемого (ω') проектов в рамках используемого опорного множества ($\varepsilon_j = 0, \delta = k$).

Решающее правило — правило, на основании которого происходит соотнесение иницируемого проекта к одному из выполненных проектов в базе. Определение: $G_{S_A}(\omega', \Omega_i) - G_{S_A}(\omega', \Omega_j) > \eta_1, \forall j, j=1, \dots, m, i \neq j; \max [G_{S_A}(\omega', \Omega_1), G_{S_A}(\omega', \Omega_2), \dots, G_{S_A}(\omega', \Omega_m)]$. Здесь η_1, \dots — параметр АВО, $G_{S_A}(\omega', \Omega_r)$ — оценка для проекта Ω_r по системе опорных множеств S_A . Выбирается проект с максимальной оценкой.

Вычисление оценки класса по системе опорных множеств. Определение: для случая, когда система опорных множеств идентична системе всех подмножеств числом элементов k и используется принятое правило близости, в АВО существует аналитическая формула, заменяющая сложные переборные процедуры:

$$G_{S_A}(\omega', \Omega_q) = \frac{1}{z_q - z_{q-1}} \sum_{\omega_j \in \Omega_q} \gamma_{s_i \omega_j} C_{p(\omega_j, \omega')}^k \quad (1)$$

Здесь z_q, z_{q-1} — номер последнего проекта, представляющего текущий и предшествующий класс, соответственно; $\gamma_{s_i \omega_j}$ — коэффициент, характеризующий значимость в опорном множестве s_i проекта ω_j ; $p(\omega_j, \omega')$ — число выполненных равенств значений характеристик для пары (ω_j, ω') ; k — мощность рассматриваемых опорных множеств.

Определение: с учетом особенностей используемой системы опорных множеств, созданной посредством применения i шагов редукции, оценка проекта Ω_q по системе опорных множеств будет:

$$\Gamma s_A(\omega', \Omega_q) = \frac{1}{z_q - z_{q-1}} \sum_{i=1}^n \sum_{\omega_j \in \Omega_q} C_{p(\omega_j, \omega')}^k \quad (2)$$

В качестве примера реализации проверки функциональной пригодности информации относительно **инициируемых** проектов на рисунке 3 приведена алгоритмическая сеть, описывающая данную проверку.

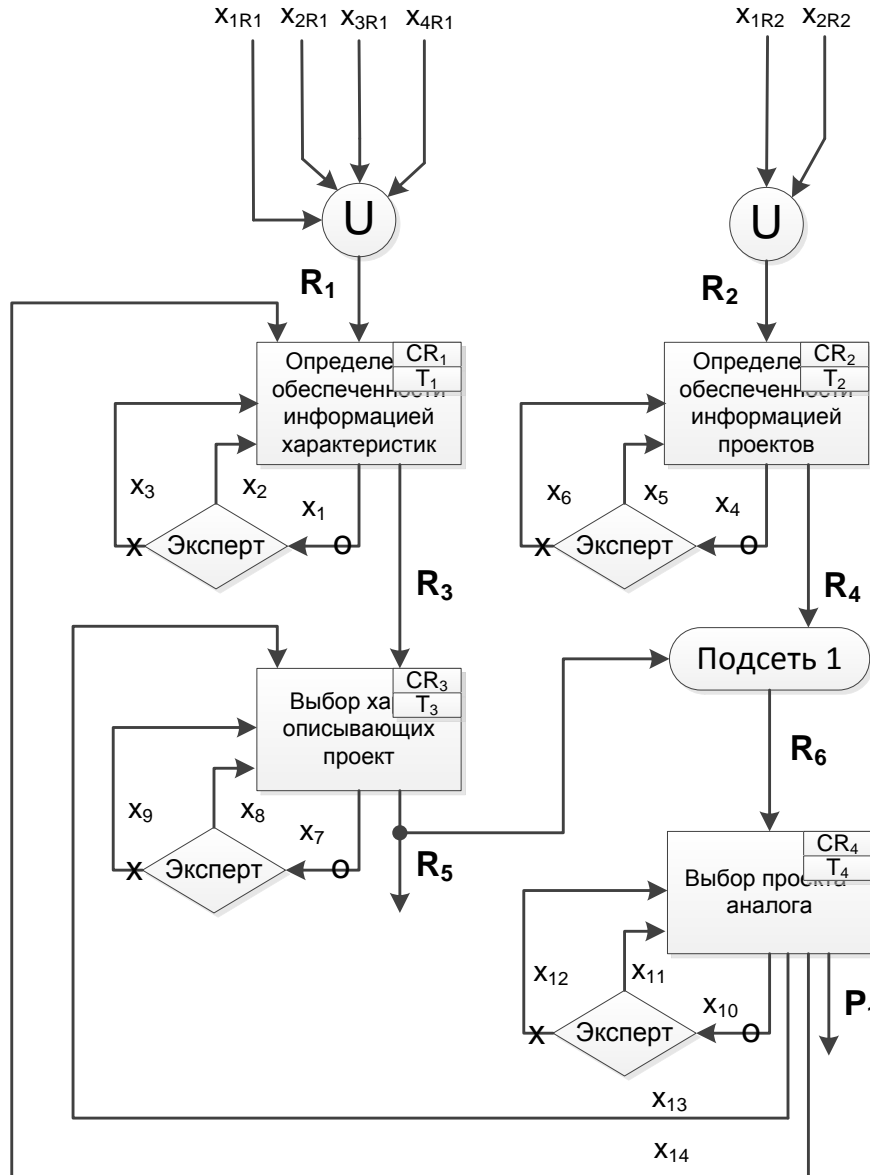


Рис. 3. Проверка функциональной пригодности информации относительно инициируемых проектов в формализме алгоритмических сетей

В данной сети переменная R_1 является исходным множеством характеристик, полученных путем объединения множеств $X_{1R1} \dots X_{4R1}$ (характеристики, добавляемые экспертом, полученные из метрической БД, полученные из документов процесса, а также универсальное множество характеристик). R_2 – исходное множество проектов, полученное путем объединения X_{1R2} и X_{2R2} (проекты, проекты добавленные экспертом, а также

полученные из метрической БД). R_3 и R_4 – уточненные множества характеристик и проектов соответственно (уточнение происходит на основе критерия обеспеченности информацией: x_3 и x_6 – характеристика и проект обеспеченные информацией, x_2 и x_5 – характеристика и проект не обеспеченные информацией). R_5 – ансамбль (множество) характеристик, который по мнению эксперта описывает иницируемый проектов (x_7 – анализируемая характеристика, x_8 и x_9 – характеристика, не попавшая и попавшая в ансамбль соответственно). В Подсети 1 выполняется процедура поиска проекта-аналога. R_6 – множества найденных проектов-аналогов. P_1 – проект-аналог, используемый в процедуре оценки (x_{10} – анализируемый проект-аналог, x_{11} и x_{12} – отвергнутый и выбранный в качестве аналога проект, x_{13} – возврат к процедуре расширения ансамбля характеристик, x_{14} – возврат к процедуре расширения уточненного множества характеристик).

В четвертой главе представлена модель системы поддержки создания баз выполненных проектов САМПО+, приведено описание и показана структура системы, описана технология работы в системе, а также представлены результаты применения и перспективы развития системы.

Система САМПО+ поддержки создания баз выполненных проектов компаний, разрабатывающих программное обеспечение, предназначена для снижения трудозатрат, связанных с их созданием. Она позволяет на основе полученной информации сформировать модель данных для дальнейшей разработки специализированной базы, а также проводить оценку выполнимости проектов, иницируемых в компании, для которой проводилось исследование. В этом качестве система САМПО+ служит инструментом, обеспечивающим снижения риска неудачного завершения иницируемых проектов.

Система ориентирована на пользователя, являющегося экспертом в области управления процессом разработки программных изделий, и обеспечивает привлечение его знаний, плохо поддающихся формализации. Данная особенность системы обуславливается использованием в ней методологии моделирования на основе алгоритмических сетей, а также методологии принятия решений на основе алгоритмических сетей (прозрачные технологии).

Важной особенностью системы САМПО+ является хранение прямых ссылок на документы, из которых взята информация, что связано с необходимостью обеспечения возможности контроля и повторного анализа вводимых значений.

Система представляет собой рабочую книгу Excel с набором из 30 рабочих листов, 9 модулей классов и 2 экранных форм. Особенность реализации системы состоит в том, что модулях кода листов находится код, отвечающий только за логику обработки элементов управления. Весь непосредственный функционал системы выведен в программные модули. Экранные формы используются для выбора режима сортировки элементов множеств: по уникальным номерам, по возрастанию и убыванию заполнения.

На рисунке 4 представлено главное меню системы.



Рис. 4. Меню системы САМПО+

Система САМПО+ обеспечивает три основных режима работы:

- режим формирования базы выполненных проектов организации (подрежимы: формирование источников, проектов, характеристик, значений и областей задания);
- режим исследования базы выполненных проектов организации (подрежимы: исследование источников, проектов, характеристик и функциональной пригодности БД);
- режим использования базы выполненных проектов организации (подрежимы: поиск зависимостей между характеристиками, поиск проекта-аналога (ручной), поиск проекта-аналога (автоматизированный), прогноз значений характеристик и оценка выполнимости проекта).

Переход между режимами и подрежимами системы осуществляется через нажатие кнопки «Завершение работы в режиме» (расположена в главном меню системы). В каждом режиме и подрежиме пользователь имеет возможность завершить работу в системе, нажав кнопку «Выход из системы» (расположена в главном меню системы).

На рисунке 4 приведен снимок экрана режима формирования базы выполненных проектов, в частности добавления характеристики в исходное множество.

Режим формирования множества характеристик проектов
(атрибуты характеристики)

Наименование характеристики:

Описание характеристики:

Тип характеристики:

Значение по умолчанию:

Документы-Источники:

Описание области задания:

Характеристика-родитель:

Характеристики-потомки:

Функция, реализующая вычисление:

Сохранить

Отменить

Характеристики базы данных	Выбранные характеристики	Уник. Номер (UN)
General characteristics		
Business characteristics		
Product characteristics		

Рисунок 4 – Режим формирования базы выполненных проектов. Добавление характеристики

В таблице 1 приведены расхождения по оценкам трудоёмкости и времени разработки (разница между расчетным и фактическим значениями). Расчеты производились по 4 наиболее представительным проектам базы выполненных проектов. В таблице представлены оценки полученные с использованием моделей СОСОМО и СОСОМО II со стандартными и усредненными коэффициентами, а также оценки полученные при помощи системы САМПО+.

Таблица 1. Сводная таблица по точности оценки

Метод оценки	Расхождение по трудоёмкости, %	Расхождение по времени разработки, %
СОСОМО со стандартными коэффициентами	18,2%	33,7%
СОСОМО со усредненными коэффициентами	10%	24,1%
СОСОМО II со стандартными коэффициентами	59,4%	147%
СОСОМО II со усредненными коэффициентами	4,1%	110,3%
САМПО+	6,5%	5%

ЗАКЛЮЧЕНИЕ

Полученные в диссертационном исследовании результаты представляют собой решение актуальной задачи разработки на базе формализма алгоритмических сетей автоматизированного метода для обоснования оценок необходимых программному проекту ресурсов, позволяющего увеличить точность и оперативность такой оценки.

В ходе исследования были получены следующие научные результаты:

- 1) модель формирования базы выполненных проектов для поиска проектов-аналогов.

- 2) метод формирования пространства характеристик для оценки ресурсов, необходимых для выполнения проектов разработки программных изделий на основе алгоритмических сетей (метод гибких оценок).
- 3) модель программной системы для автоматизированного поиска ближайших проектов-аналогов по базе выполненных проектов для оценки необходимых ресурсов по принципу подобия.

Результаты диссертационного исследования внедрены в ООО «Эксиджен Сервисес» (в конце 2009 года компания претерпела изменения и стала именоваться «Ф-Лайн Софтвер»), НП «Объединение подземных строителей», а также в учебном процессе Санкт-Петербургского университета водных коммуникаций.

Они также могут быть использованы в перспективных НИР, при разработке ОКР по созданию ПО и в учебном процессе при подготовке специалистов в области ПО.

СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в рецензируемых научных изданиях из перечня ВАК

1. Тележкин А.М. Система САМПО+ для создания и анализа исторической базы данных / Известия ВУЗов. Приборостроение. – 2014. № 11. – С. 58-62.
2. Баранов С.Н. Тележкин А.М. Метрическое обеспечение программных разработок / Труды СПИИРАН. - 2014. № 5(36). – С. 5-27

Публикации в других изданиях

1. Егоров А.Г., Морозов В.П., Тележкин А.М., Тубольцева В.В. Система автоматизации поиска проектов-аналогов для оценивания выполнимости проектов разработки программных изделий / Региональная информатика-2008 (РИ-2008) XI Санкт-Петербургская Международная Конференция, Санкт-Петербург, 22-24 октября 2008 года: Материалы конференции \ СПОИСУ. - СПб, 2008. С. 32 - 33.
2. Морозов В. П., Калугина Е. А., Тележкин А. М. Система поддержки создания баз исторических данных компаний, разрабатывающих программное обеспечение / Четвертая Всероссийская научно-практическая Конференция «Имитационное моделирование. Теория и практика» (ИММОД-2009), Санкт-Петербург, 21-23 октября 2009 года: <http://gpss.ru/immod09/doklad/99.pdf>
3. Морозов В.П., Тележкин А.М. Формирование пространства характеристик для определения ресурсов, обеспечивающих успешное завершение проекта, в системе САМПО+ / Региональная информатика-2010 (РИ-2010) XII Санкт-Петербургская Международная Конференция, Санкт-Петербург, 20-22 октября 2010 года: Материалы конференции \ СПОИСУ. - СПб, 2010. С. 199 - 200.
4. Тележкин А.М. Историческая база данных, как инструмент планирования проекта разработки программных изделий / Региональная информатика-2012 (РИ-2012) Юбилейная XIII Санкт-Петербургская Международная Конференция, Санкт-Петербург, 24-26

октября 2012 года: Материалы конференции \ СПОИСУ. - СПб, 2012. С. 52 - 53.

5. Тележкин А.М. Создание исторических баз данных при помощи системы САМПО+ / Региональная информатика-2012 (РИ-2012) Юбилейная XIII Санкт-Петербургская Международная Конференция, Санкт-Петербург, 24-26 октября 2012 года: Труды конференции \ СПОИСУ. - СПб, 2012. С. 84 - 90.