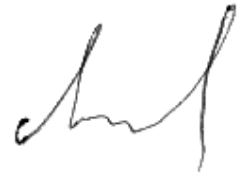


На правах рукописи



Лапшов Юрий Александрович

**СРЕДСТВА ПРОГРАММНО-КАРТОТЕЧНОГО  
УПРАВЛЕНИЯ ПОТОКАМИ РАБОТ В  
КОЛЛЕКТИВНОМ ПРОЕКТИРОВАНИИ  
АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

Специальность: 05.13.12 – Системы автоматизации  
проектирования (промышленность)

**Автореферат**

диссертации на соискание ученой степени  
кандидата технических наук

Ульяновск – 2015

Работа выполнена на кафедре Вычислительная техника Ульяновского государственного технического университета.

Научный руководитель – доктор технических наук, профессор,  
**Соснин Петр Иванович**

Официальные оппоненты:

**Тюгашев Андрей Александрович,**  
доктор технических наук, доцент,  
СГАУ, профессор кафедры «Программные  
системы»

**Липатова Светлана Валерьевна,**  
кандидат технических наук,  
УлГУ, доцент кафедры «Телекоммуникационные  
технологии и сети»

Ведущая организация – **Федеральное государственное бюджетное  
образовательное учреждение высшего  
профессионального образования  
«Волгоградский государственный технический  
университет»**

Защита состоится «24» июня 2015 г. в 15 часов 00 минут на заседании диссертационного совета Д 212.277.01 при Ульяновском государственном техническом университете по адресу: 432027, г. Ульяновск, ул. Северный Венец, 32 (ауд. 211, Главный корпус).

С диссертацией можно ознакомиться в библиотеке Ульяновского государственного технического университета. Также диссертация и автореферат размещены в Internet на сайте УлГТУ – <http://www.ulstu.ru/>

Автореферат разослан «\_\_\_» \_\_\_\_\_ 2015 г.

Ученый секретарь  
диссертационного совета,  
доктор технических наук, профессор



Смирнов Виталий Иванович

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### Актуальность работы

Основные проблемы расширяющейся компьютеризации всех сфер человеческой деятельности проявляются в разработках систем, интенсивно использующих программное обеспечение (Software Intensive Systems, SIS). К числу этих проблем относится чрезвычайно низкая степень успешности разработки SIS, которая за последние 20 лет с «трудом» приблизилась к 35% (всемирно признанные статистические исследования и отчеты корпорации Standish Group и других исследователей).

В отчетах, регистрирующих положение дел с успешностью разработок, фиксируются не только оценки успешности для разнообразных условий создания SIS, но и факторы, способствующие и препятствующие успеху разработок. В перечнях этих факторов (среди особо важных) устойчиво занимает «управление коллективной и персональной деятельностью проектировщиков», используемое в процессах проектирования SIS.

Необходимость переосмысления основ программной инженерии и внесения в них существенных изменений (а именно создание программных составляющих SIS, согласованных с остальными ее компонентами приводит к проблемам с успешностью) привела к инициативе SEMAT (Software Engineering Methods And Theory – Методы и Теория Программной Инженерии, доступно на сайте <http://www.semat.org>), в спецификации и осуществлении которой (начиная с 2009 года) приняли участие ведущие мировые ученые и специалисты в области программной инженерии (Ivar Jacobson, Watts S. Humphrey, Scott W. Ambler, Alistair Cockburn и другие).

В нормативных документах SEMAT «техника работы» (way-of-working), используемая командой проектировщиков, отмечена как очень важная сущность. В нормативах SEMAT «техника работы» определена как «адаптированный к специфике исполняемого проекта набор методов и инструментов, используемых командой проектировщиков в процессе проектирования», причем, адаптация подразумевает не только настройку методов и инструментов, выбранных командой, исполняющих проект, но и создание членами команды новых методов, оказавшихся им необходимыми.

Особое отношение к техникам работы не может не затрагивать вопросы управления коллективной и персональной деятельностью проектировщиков SIS, а значит и автоматизированных систем (АС), которые относятся к классу SIS. Вышесказанное содержит достаточно оснований, чтобы считать создание инновационных методов и средств управления коллективной и персональной деятельностью проектировщиков АС актуальным. К этому направлению инноваций относится и предлагаемый в диссертации комплекс средств программно-картотечного управления процессами проектирования автоматизированных систем.

То, что «техники работы» имеют алгоритмическую природу, привело в диссертации к идее представлять «техники работы» и их связанные совокупности (потoki работ) в виде программ на алгоритмическом языке, подобном естественному языку в его алгоритмическом употреблении для планирования персонального и коллективного поведения проектировщиков. Программное представление потоков работ и их составляющих открывает возможность для применения в их построении и использовании как методов и средств эмпирической программной инженерии, так и моделей опыта из других предметных областей, в первую очередь из арсенала Искусственного Интеллекта.

В документах SEMAT предлагается команде проектировщиков начинать разработку очередного проекта с адаптации к его специфике «ядра SEMAT», дополняя его «как пазлами» тем, что способствует реализации проекта. В диссертационном исследовании в

число таких пазлов, дополнительных к программному управлению, было решено включить средства визуально-картотечного управления, нашедшего свое материальное воплощение в различных версиях технологий **Kanban** и **Scrum**. Более того, интересы исследования было решено ограничить деятельностью проектировщиков в процессах концептуального проектирования АС, которые считаются наиболее проблематичными с позиций их влияния на успешность.

В диссертационной работе роль **области исследований** возложена на средства управление потоками работ в концептуальном проектировании АС.

**Направление исследований** в диссертации связано с инструментально-технологическими средствами программного управления потоками работ гибкой разработки программного обеспечения, которые специально введены в процесс разработки АС для координирования потоков проектных работ.

Функции **объекта исследований** в диссертации выполняют средства координирования потоков проектных работ, использующие гибкие методологии разработки.

Роль **предмета исследования** в диссертации возложена на методы и средства координирования потоков работ участника проектного процесса, такие, как управление очередями задач и прерываниями, а также – методы и средства координирования потоков работ команды проектировщиков, такие, как **Kanban**, **Scrum**.

**Целью исследования** является совершенствование процессов управления потоками работ, способствующее предотвращению ошибок, обусловленных человеческим фактором, и обеспечивающее сокращение непроизводительных затрат времени в оперативной индивидуальной и коллективной работе проектировщиков за счет включения в совокупность средств управления проектной деятельностью дополнительных программируемых составляющих

#### **Задачи диссертационного исследования:**

1. Проанализировать возможность и обосновать целесообразность включения в состав средств управления персональной и коллективной активностью проектировщиков АС дополнительных технологических средств программного управления работами проектировщиков, с позиций исполнения каждым из них роли «специализированного процессора».

2. Разработать специализированную систему псевдо-кодowego программирования, ориентированную на создание и исполнение программ персональных и коллективных действий проектировщиков, включающих действия по оперативному использованию доступного опыта и его моделей, способствующих повышению эффективности управления процессами проектирования.

3. Разработать программно-картотечную систему оперативного распределения проектных задач, планирования их параллельного исполнения, картотечной визуализации состояния, оценки результативности работ и анализа их исполнения для оптимизации очередных шагов проектирования.

4. Разработать методы и средства для создания и использования библиотек запрограммированных прецедентов, упрощающих создание средств программного управления проектной деятельностью, учитывающих специфику конкретной разрабатываемой АС.

5. Разработать методологическое обеспечение и систему инструментально технологических средств, обслуживающих программно-картотечное управление персональными и коллективными действиями проектировщиков.

6. Разработать прототип технологии программно-картотечного управления и провести его испытания в лабораторных условиях.

**На научную новизну претендуют:**

1. Программно-картотечная модель гибкого управления потоками работ, ориентированная на использование механизмов **Kanban** и **Scrum** в проектировании АС, специфику которой определяет визуализация очередей задач, учитывающая их распределение в коллективе проектировщиков и во времени, и открывающая возможность эффективного программируемого распараллеливания решений проектных задач в условиях управления их прерываниями;

2. Подмножество концептуально-алгоритмического языка, к особенностям которого относится определение его данных и операторов над семантической памятью, обеспечивающее оперативное программное управление очередями задач за счет полезных метрик планирования работ и продуктивности групп проектировщиков.

3. Совокупность методик, включающих методики отображения потоков работ и их исполнителей на вопросно-ответную память, оперативного планирования, картотечной визуализации и гибкого управления по образцам **Kanban** и **Scrum** с учетом детализации очередей, позволяющей программно управлять шагами их исполнения.

4. Библиотека псевдокодовых программ гибкого управления и программных моделей шаблонов типовых потоков работ, настроенная на повторное использование и включающая дополнительные разделы процедур и функций, написанных на языках Microsoft .NET Framework.

В число **методов исследований** включены методы: объектно-ориентированного анализа, проектирования и программирования; теории и практики автоматизированного проектирования; когнитивного анализа задач и принятия решений; массового обслуживания.

**Достоверность** полученных результатов обеспечивается использованием достоверных знаний, методов и средств из логики, прикладной информатики и программной инженерии. Практический вклад в достоверность подтверждается разработкой комплекса средств, обеспечивающего программно-картотечное управление процессами персональной и коллективной проектной деятельности в создании автоматизированных систем.

**На защиту выносятся следующие новые научные результаты:**

1. Программно-картотечная модель гибкого проектного управления и средства, позволяющие запланировано или оперативно вводить в управление потоками работ в проектировании АС, составляющие, которые программируют действия проектировщиков, исполняемые ими параллельно в корпоративной сети и/или псевдопараллельно на рабочих местах.

2. Моделирование визуализируемых очередей задач, с которыми оперативно взаимодействуют проектировщики, в виде ситуационно управляемых псевдокодовых программ.

3. Спецификации системы человеко-компьютерных прерываний, ориентированной на согласованное исполнение программ, управляющих концептуальным проектированием, интеллектуальными и компьютерными процессорами.

4. Способ комплексирования средств программного управления процессами решения проектных задач, их прерываний и картотечной визуализации.

**Практическую ценность** составляет разработанный набор средств, обеспечивающий реализацию координирования потоков работ в проектировании АС,

создана библиотека шаблонов потоков работ, разработан комплекс методик по координированию потоков работ в процессе проектирования АС.

### **Реализация и внедрение результатов работы**

Разработанные программные средства и комплекс методик их использования внедрены на ФНПЦ ОАО НПО «МАРС», проведен эксперимент по оценке эффективности использования предлагаемых средств.

Грант №15-07-04809 «Технологии и инструментарий программно-картотечного управления процессами в проектировании систем, интенсивно использующих ПО» и работы по Государственному заданию № 2014/232 на период 2014-2016 г.г.

Свидетельства о регистрации программы для ЭВМ: №2015610593 «Компилятор псевдокодовых программ», №2015610236 «Программный инструментарий контроля поручений»

### **Апробация работы**

Основные положения и результаты диссертационной работы докладывались и обсуждались на следующих конференциях:

8-th international conference «Interactive Systems: Problems of Human – Computer Interaction», региональная научно-техническая конференция «Информатика и вычислительная техника, 2009», всероссийская конференция «Проведение научных исследований в области обработки, хранения, передачи и защиты информации, 2009», российская школа-семинар «Информатика, моделирование, автоматизация проектирования», 2009, российская конференция «Информатика и вычислительная техника», 2010, Научно-техническая конференция УлГТУ, 2010, российская школа-семинар «Информатика, моделирование, автоматизация проектирования», 2010, российская конференция «Информатика и вычислительная техника», 2011, российская школа-семинар «Информатика, моделирование, автоматизация проектирования, 2011», 9-th international conference «Interactive Systems: Problems of Human – Computer Interaction», российская конференция «Информатика и вычислительная техника», 2012, российская школа-семинар «Информатика, моделирование, автоматизация проектирования», 2012, International conference «Open Semantic Technologies for Intelligent Systems» (OSTIS-2013), 10-th international conference «Interactive Systems: Problems of Human – Computer Interaction».

**Публикации.** По материалам диссертационной работы опубликовано 28 печатных работ.

**Структура и объем работы.** Диссертационная работа состоит из введения, четырех глав с выводами, заключения, библиографического списка использованной литературы (142 наименования) и двух приложений, общим объемом 246 страниц машинописного текста. Диссертация содержит 97 рисунков и 41 таблицу.

### **КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ**

Во **введении** дается обоснование актуальности работы, определены цели и задачи работы.

В **первой главе** диссертационной работы раскрыты вопросы, связанные с особенностями коллективного проектирования АС, человеко-компьютерным взаимодействием и дизайном, ориентированным на пользователя, дается обоснование необходимости управления потоками работ в проектировании АС. Обоснована необходимость учета особенностей коллективной и персональной деятельности во время реализации АС. Упомянуты вопросы профессиональной зрелости процессов разработки АС. Приведен обзор методов и средств координирования потоков проектных работ на ранних стадиях разработки АС.

Приведен анализ отчетов *Standish Group*, характеризующий крайне низкую успешность проектов *SIS*. Рассмотрен ряд факторов, способствующих достижению успеха, в том числе такие факторы, как подходящее планирование, реалистичные ожидания и управление проектом. Характеристики «успеха» проектной деятельности отражены в стандарте *ISO 9004:2009*.

В статистических исследованиях успешности разработок *SIS* конкретную разработку считают «успешной», если она завершена в соответствии с запланированными требованиями в рамках оговоренных финансовых затрат и времени.

Одним из направлений повышения успешности разработок программных систем и систем, интенсивно использующих программное обеспечение, является совершенствование профессиональной зрелости осуществляемых процессов. Уровень зрелости производственного процесса – это степень, до которой тот или иной процесс определен, управляем, измеряем, контролируем и эффективен. Уровень зрелости является интегральной характеристикой, включающей такие показатели, как степень определенности и степень управления проектным процессом. В степени управления процессом также принято выделять два значения «управляемый процесс» и «количественно управляемый процесс», вкладывая во второе значение оперативную управляемость по количественно измеряемым отклонениям характеристик процесса от плановых значений.

Рассмотрены особенности проектной деятельности с позиции управления процессом проектирования, представленные в различных стандартах проектного управления, в том числе стандартом *PMBOK*, который изначально и во всех своих версий (до *PMBOK 5:2012*) создавался как «свод» знаний о проектном управлении, или, как часто говорят, как «энциклопедия» об этой предметной области, *ГОСТ Р 54869–2011* «Проектный менеджмент. Требования к управлению проектом», *ISO 21500:2012* «Руководство по менеджменту проектирования», *ISO 9004:2009* «Менеджмент для достижения устойчивого успеха организации: Подход на основе менеджмента качества». Приводится обзор стандартизации знаний о проектном управлении с позиции стандарта *PMBOK*. Существуют также и другие стандарты проектного менеджмента, такие, как *PRINCE2*. Кроме того, существуют модели зрелости управления портфелями, программами и проектами, такие, как *P1M3*, *P2M3* и *P3M3*, набор моделей совершенствования процессов *CMMI*, в том числе модель *CMMI-DEV*, ориентированная на организации, занимающиеся разработкой программного обеспечения. Рассмотрена технология и инструментарий «*Rational Unified Process*» (*RUP*), созданные корпорацией *IBM*.

Рассмотрены гибкие методологии разработки (*Agile*-методологии), наиболее широкое распространение из которых получили методологии *Kanban* и *Scrum*, а также – их комбинации, чаще всего *Scrumban*.

Далее приведен обзор средств моделирования бизнес-процессов, наиболее полезным из которых является моделирование бизнес-процессов и их совокупностей в виде *потоков работ (workflows)*. Потоки работ представляют собой логически взаимосвязанные работы по выполнению проектных задач и других задач, связанных проектом. Для описания особенностей организации потока работ используются шаблоны управления потоками работ (*workflow patterns*), базовые из которых представлены в **Приложении 1** диссертационной работы.

Представлен обзор особенностей прерываний в человеческой деятельности, из которого следует, что прерывания вызывают такие негативные эффекты, связанные с человеческим фактором, как непроизводительные затраты времени и риск возникновения ошибок. На рисунке 1 представлена схема человеческого прерывания, на которой

отмечены временные интервалы, связанные с переключением внимания и ориентацией в прерванной задаче, являющиеся основными непроизводительными затратами времени.

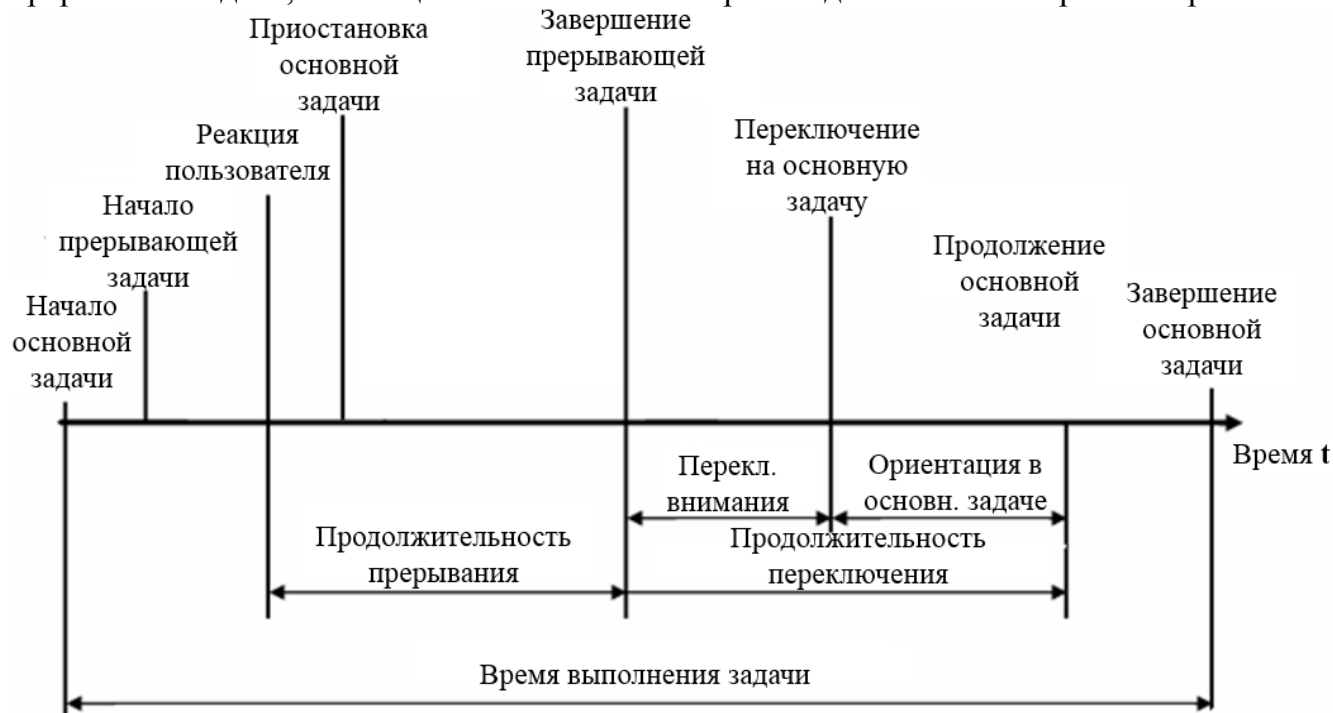


Рис. 1. Схема прерывания человеческой деятельности

Приведен ряд выводов, смысл которых заключается в том, что степень успешности разработок SIS в настоящее время недопустимо низка, что связано с отсутствием гарантийной предсказуемости ее результатов, которая определяется уровнем зрелости производственного процесса. Включение комплекса средств программно-картотечного управления (ПКУ) позволяет приблизиться к максимальному – эффективному уровню зрелости процесса проектирования.

Комплекс средств ПКУ предлагает подход к совершенствованию гибких методологий проектирования, таких, как **Kanban** и **Scrum** за счет включения в их картотечное управление дополнительных программируемых составляющих, управляющих прерываниями задач и распараллеливания их решений как в коллективных, так и в персональных действиях. Предлагаемое комплексирование получило название «Программно-картотечное управление» (ПКУ). Обосновывается решение о выборе инструментария **WIQA** (**Working In Questions and Answers**) для реализации средств ПКУ. На рисунке 2 представлены средства ПКУ, реализованные в составе среды **WIQA**. Комплекс **WIQA** разрабатывался для инструментально-технологического сопровождения процессов решения задач в концептуальном проектировании АС. Наиболее общим видом задач, с которым связаны интересы данной диссертационной работы, является задача **Z\*** разработки очередной АС из их семейства. За этой задачей стоят все остальные задачи проекта (как технологические, так предметные), отношения подчиненности между которыми фиксируются деревом задач.

Особое место в комплексе **WIQA** занимает та его часть, которая предназначена для отображения процесса решения задачи **Z\*** и его результата, то есть проекта АС на семантическую память вопросно-ответного типа, или другими словами, на **QA-память**. К особенностям **QA-памяти** относится то, что она специфицирована и материализована для обслуживания взаимодействий проектировщиков с доступным опытом.



Поток работ проекта отображается на память **WIQA** в виде вопросно-ответных моделей, на основе которых в процессе планирования происходит формирование фронта работ, включающего как задачи, так и сотрудников, ответственных за их выполнение. В процессе проектирования для каждого сотрудника формируются очереди задач, картотечное представление которых позволяет визуализировать данные очереди.

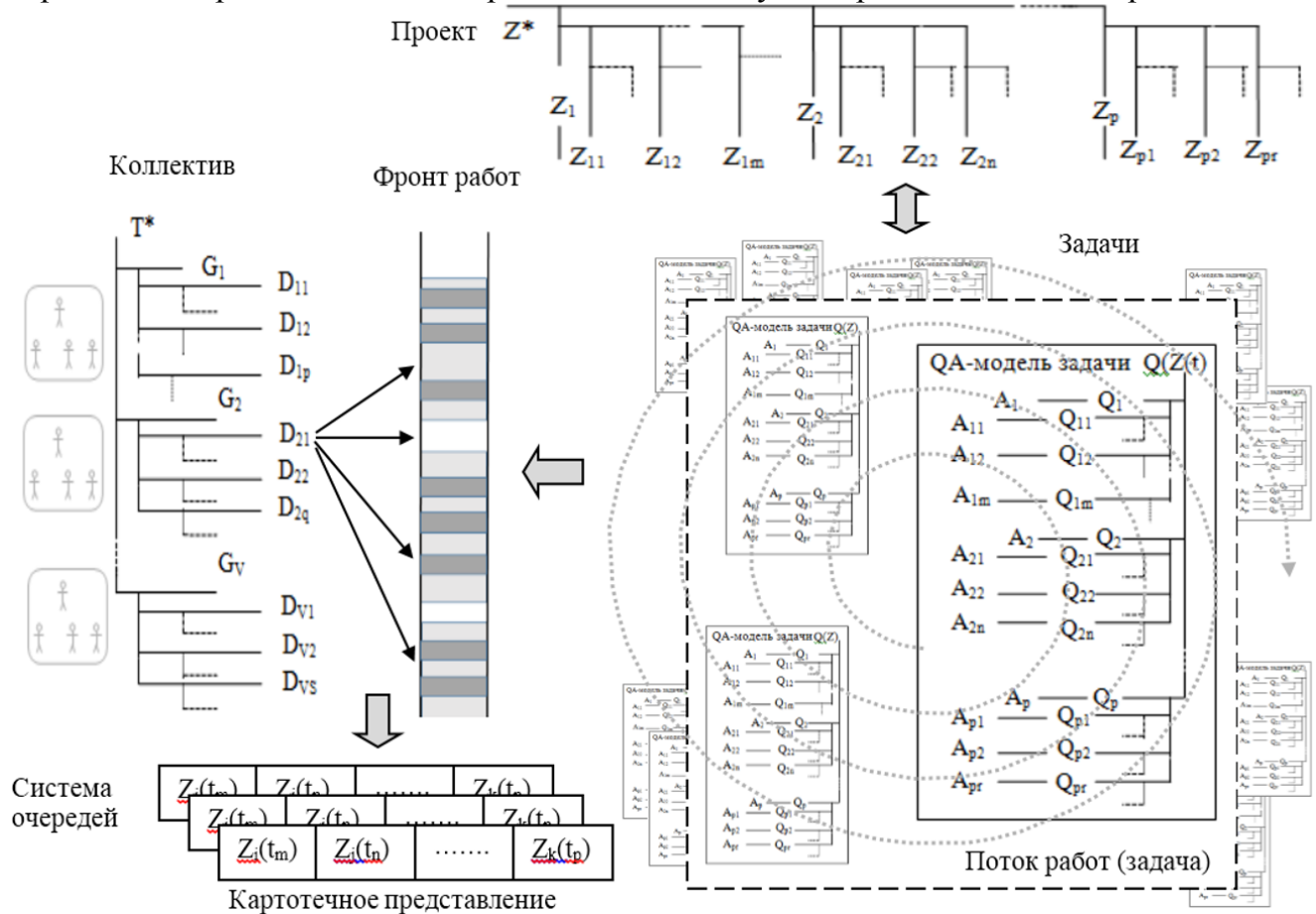


Рис. 2. Средства ПКУ в составе среды WIQA

Одной из специфик **WIQA** является поддержка псевдокодowego программирования, ориентированного на прецеденты. Такая возможность, во-первых, позволяет расширять функционал комплекса **WIQA**, а, во-вторых, позволяет строить на основе этого комплекса приложения.

В главе также приводится обобщенная постановка задачи **Z\*** диссертационного исследования:

**Z\*** 1. Разработать комплекс методов и средств управления потоками работ, способствующий предотвращению ошибок, обусловленных человеческим фактором, и обеспечивающий сокращение непроизводительных затрат времени в оперативной коллективной работе за счет включения в совокупность средств управления проектной деятельностью дополнительных программируемых составляющих;

2. Механизм дополнительных программируемых составляющих, предлагаемых комплексом средств следует ориентировать на участие в их реализации проектировщиков, выполняющих порученную им работу в соответствии с планами, представленными на языке, близком к естественному языку, в его алгоритмических условиях;

3. Для снижения затрат на разработку комплекса средств управления потоками проектных работ связать его информационно по входу и выходу с вопросно-ответной инструментальной средой **WIQA**.

Был проведен вопросно-ответный анализ **Z\***, обеспечивающий структуризацию результатов данного диссертационного исследования и распределение их по главам.

Также приводится мотивационно-целевой анализ, выделены мотивы и цели данной работы, а также эффекты, получаемые путем достижения этих целей. Комплекс средств ПКУ должен содержать инструментарий управления коллективной и персональной активностью проектировщиков, инструментарий работы с опытом проектирования, инструментарий контроля поручений, инструментарий визуализации очередей задач, а также набор средств, обеспечивающий параллельное и псевдопараллельное выполнение потоков работ и методики вычисления метрик, что позволяет обеспечить рациональное планирование проектных работ и управление очередями задач проектировщика, минимизировать негативное влияние человеческого фактора на процесс проектирования.

**Во второй главе** приведена формализация и спецификация процессов ПКУ, включающая архитектурную модель средств ПКУ, особенности отображения средств ПКУ на семантическую память **WIQA** и особенностей организации процесса ПКУ. Кроме того, в главе приводятся особенности программирования распараллеливания проектной деятельности как группы проектировщиков, так и проектировщика в работе над очередью задач как с позиции языка, так и с позиции алгоритмики.

Глава начинается с построения архитектурной модели средств ПКУ в среде **WIQA**. Архитектурная модель, включающая инструментарий планирования потока проектных работ и выполнения фронта работ проекта в соответствии с выбранной методологией проектирования, а также программные средства, обеспечивающие функционирование данного инструментария, представлена на рисунке 3. Среда **WIQA**, включающая в себя инструментарий ПКУ, содержит в **QA-памяти** иерархическое представление коллектива сотрудников, объединенных в группы и дерево проектных задач. Инструментарий организационной структуры объединяет коллектив и задачи, обеспечивая реализацию механизма распределения задач, зарегистрированных в дереве, между их исполнителями  $\{D_p\}$  в коллективе  $K(\{D_p\})$  с использованием процедуры назначения задач (средства работ с оргструктурой коллектива и деревом задач);

С организационной структурой взаимодействует средство управления поручениями, отвечающее за предварительное планирование сроков исполнения работ в процессах решения задач всех типов. С использованием средств **QA-программирования**, включающих редактор псевдокода, компилятор и интерпретатор псевдокодовых программ, реализованы такие средства ПКУ, взаимодействующие с **QA-памятью WIQA**, как механизмы формирования потоков работ, инструментарий **Kanban**, **Scrum** и управление прерываниями. Программные средства **Kanban** обеспечивает параллелизм в выполнении задач в коллективе проектировщиков за счет визуального средства работы с очередями проектных задач. Инструментарий **Scrum** обеспечивает рациональное оперативное планирование этапа проектной работы с использованием метрических характеристик выполнения предыдущих этапов. Средства управления прерываниями осуществляют поддержку псевдопараллельной работы каждого проектировщика с назначенными ему задачами.

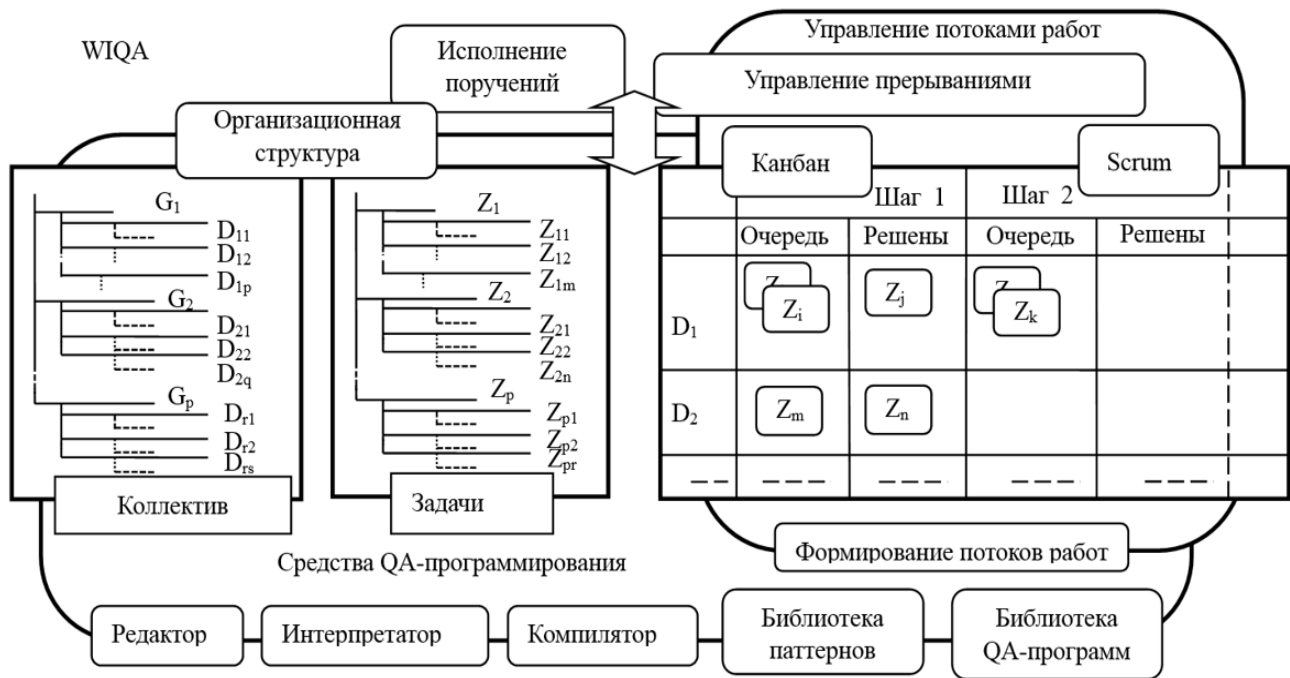


Рис. 3. Архитектурная модель средств ПКУ

Инструментарий обслуживает управляемое оперативное формирование, регистрацию, визуализацию и использование текущего состояния дерева задач  $T(\{Z_i\}, t)$ , включающего совокупность задач  $Z^D = \{Z_{mi}\}$  предметной области АС, которые в проектируемой АС будут решать ее пользователи, совокупность нормативных задач  $Z^N = \{Z_{ni}\}$  технологии, в рамках которой коллектив проектировщиков  $K(\{D_p\})$  создает АС, совокупность задач адаптации  $Z^A = \{Z_{aq}\}$ , решаемых проектировщиками для настройки задач типа  $Z^N$ , инвариантных к проблемной области АС, в их использовании при решении задач типа  $Z^D$  и совокупность задач управления  $Z^W = \{W^m(\{Z_{mi}\})\} \cup \{W^n(\{Z_{ni}\})\}$  и  $Z^G = \{Z_{gr}\}$ , решение которых обслуживает работу с задачами в потоках работ, а также согласованное управление в группах.

Задачи, представленные в дереве задач объединяются в **потоки работ** или их группы, каждому из которых соответствует задача типа  $Z^W$  или  $Z^G$ . Эта задача обеспечивает динамические отношения между задачами потока с помощью алгоритмов, реализованных на языке псевдокодowego программирования. Потоки работ предполагают определенную очередность выполнения задач, параллелизм и наборы условий, разрешающие выполнение той или иной задачи. Алгоритмическая конфигурация для каждого потока работ является специфической, но в то же время она описывается с использованием стандартных **шаблонов (паттернов) потоков работ**, представляющих собой частные случаи межзадачных отношений. Исследования в области шаблонов потоков работ насчитывают **42** шаблона управления. Эти шаблоны также являются программируемыми на языке  $L^{WIQA}$  и в состав средств ПКУ была включена библиотека шаблонов, которые можно использовать при построении алгоритма управления потоком работ. Псевдокодowe программы базовых шаблонов потоков работ представлены в **приложении 1** диссертационной работы.

В процессе выполнения проектных задач  $Z_i$ , отображенных на **QA-память WIQA**, проектировщик взаимодействует с компонентами ПКУ, используя средства взаимодействия, обеспечивающие интерактивный графический пользовательский интерфейс. На **QA-память WIQA** также отображены методики ПКУ и псевдокодowe программы инструментальных средств (Рисунок 4).



Рис. 4. Отображение средств ПКУ на семантическую память WIQA

В качестве **инструмента формализации** в диссертации используется **РБНФ**-грамматика, с использованием которой разработано около **200** правил, управляющих как отображениями на память, так и использованием результатов отображений.

Подсистема **Kanban** в системе ПКУ визуализирует доску **Kanban** для группы сотрудников **G** в виде таблицы **КТ**. Доска делится на набор колонок – шагов, соответствующих этапам выполнения проектного процесса.

$$КТ = G, \text{ Колонки};$$

Шаги представляют собой этапы проектного процесса, которые доска **Kanban** отображает в виде столбцов. Каждый шаг содержит в себе карточки, соответствующие задачам члена группы, которым может быть как подгруппа, так и сотрудник. Таким образом, карточки располагаются в соответствующих им ячейках **GD**.

$$\text{Колонки} = \{\text{Колонка}\};$$

$$\text{Колонка} = I, \{GD\};$$

**I** в данном случае характеризует индекс колонки, соответствующий шагу выполнения проектной работы.

Каждая ячейка содержит в себе набор карточек, соответствующих **очереди задач** проектировщика на определенном этапе работы над проектом

$$GD = \{\text{Карточка}\};$$

$$\text{Карточка} = Cr;$$

Карточка **Kanban** характеризуется определенным набором атрибутов. Карточка **Cr** является визуальным представлением задачи, поэтому одним из важнейших ее атрибутов является ссылка **Z<sub>ref</sub>** на задачу, которую она визуализирует. Для задачи карточка описывает ее имя **Desc**. Также карточка характеризуется цветом **C<sub>c</sub>**. Каждая карточка может быть отмечена флажком **Cr**.

$$Cr = Z_{ref}, Z_{desc}, Z_c, C_f;$$

Флажок может быть установлен или снят:

$$Cf = \text{Установлен} \mid \text{Снят};$$

Каждая карточка может быть раскрыта, т.е. представлена в увеличенном масштабе, отображая детальную информацию о задаче.

$$\begin{aligned} \text{Раскрытая карточка} &= Cr^O; \\ Cr^O &= Cr, \text{"раскрытие"}; \end{aligned}$$

Раскрытая карточка увеличивается в размере и отображает ряд дополнительной информации о задаче, кроме сведений, отображаемых карточкой. Карточка имеет две стороны – лицевую  $Cr^F$  и обратную  $Cr^B$ , а также – ряд элементов управления  $OCC$ , позволяющих перевернуть карточку на другую сторону, перейти к выполнению задачи и отметить задачу выполненной. Лицевая сторона карточки отображает ее детальное описание  $Desc_D$

$$\begin{aligned} Cr^O &= (Cr^F \mid Cr^B), Cf, OCC; \\ Cr^F &= Desc_D; \\ Cr^B &= Desc, \text{Автор}, \text{Дата исполнения}; \\ OCC &= \text{Перевернуть}, \text{К задаче}, \text{Выполнена}; \end{aligned}$$

**Scrum** представляет собой итеративный, инкрементальный процесс для разработки любого продукта или управления любой работой. Он производит потенциально готовый продукт после каждой итерации.

Итерация в **Scrum** именуется **спринтом (Sprint)**. Длительность такого спринта фиксирована. Объем задач проекта в терминологии **Scrum** называется **Бэклог проекта (Project Backlog)**. Набор задач для выполнения в течение времени спринта называется **бэклогом спринта (Sprint Backlog)**.

Итеративный процесс работы  $S_{proc}$  по методологии **Scrum** состоит из следующих основных частей:

- Формирование бэклога спринта ( $SB_f$ );
- Выполнение спринта ( $SB_w$ );
- Расчет метрик спринта и их анализ ( $SB_a$ ).

$$S_{proc} = \{SB_f, SB_w, SB_a\};$$

$SB_f$  заключается в формировании спринта  $SS$ , представляющего собой бэклог спринта  $SB$  и команду, выполняющую спринт  $ST$ .

$$SS = SB, ST;$$

Бэклог спринта в свою очередь представляет собой набор задач, каждая из которых имеет оценку трудозатратности  $Z_v$ , распределенный по приоритетам. Каким именно образом осуществляется формирование бэклога спринта, описано в п.3.3.1.

$$SB = \{SB_a\}, \{Z, Z_v\};$$

Кроме бэклога спринта, обратная связь также воздействует на формирование команды. Например, в процессе выполнения спринтов может выясниться, что проектировщики работают недостаточно эффективно из-за выполнения непривычных им ролей

Для формирования бэклога спринта используются метрики, полученные в результате анализа выполнения предыдущих спринтов. Таким образом, можно говорить о

возникновении обратной связи, соединяющей результаты метрик с новым бэклогом спринта.

$$ST = \{ \{ SB_a \}, ST \};$$

$SB_w$  представляет собой итеративный процесс выполнения работы над спринтом, в течение которого выполняются краткосрочные **24-часовые спринты**:

$$SB_w = \{ SB_t \};$$

Во время этого краткосрочного 24-часового спринта происходят следующие процессы:

- Ежедневный **Scrum** (Daily **Scrum**)  $D_{Sc}$ ;
- Выполнение проектных задач;

Таким образом,

$$SB_t = D_{Sc}, \{ (Z_i, \text{"выполнение"}) \};$$

В свою очередь, ежедневный **Scrum** представляет собой **15-минутное обсуждение**, в ходе которого происходит оценка проделанной работы и внесение изменений в диаграмму выгорания  $P_{ob}$ . Диаграмма выгорания представляет собой график, на котором по оси абсцисс отмечены дни – каждая отметка соответствует ежедневному **Scrum**, а по оси ординат отмечается оставшийся объем работы  $Sv - Z_{Vi}$  для  $i$ -го дня.

$$\begin{aligned} D_{Sc} &= (P_{ob}, \text{"отметка"}); \\ P_{ob} &= Sv, \{ Sv - Z_{Vi} \}; \\ \text{"отметка"} &= P_{ob}, Sv - Z_{V_n} \end{aligned}$$

Здесь  $Z_{Vi}$  представляет собой выполненный на текущий момент объем работ спринта.

После завершения времени выполнения спринта происходит вычисление метрик. К этим метрикам относятся:

- Скорость работы сотрудника (Individual Velocity, **IV**);
- Скорость работы команды (Team Velocity, **TV**);
- Оценка стоимости выполненной работы (Value Delivered, **VD**);
- Оценка планирования (On-time Delivery, **TD**).

Таким образом,

$$SB_a = (IV, TV, VD, TD), \text{"вычисление"};$$

Ряд метрик вычисляется автоматически по соответствующим им формулам. Метрика скорости работы сотрудника определяется как отношение трудоемкости выполненных им работ к длительности спринта.

$$IV = \frac{\sum_{i=0}^{nd} Z_{Vi}}{t},$$

где  $nd$  – количество выполненных задач проектировщика в спринте,

$Z_{Vi}$  – трудоемкость задачи  $i$ ,

$t$  – продолжительность спринта.

Оценка стоимости выполненной работы осуществляется совместно **стейкхолдерами** (stakeholder) проекта. Ему демонстрируют историю выполнения спринта и он ставит оценку  $M$  эффекта от использования результата выполнения каждой задачи спринта в денежном эквиваленте или в любой другой оценке для каждой задачи.

Таким образом,

$$VD = \{ Z, M \};$$

В главе приведено представление группы проектировщиков, выполняющих проектные задачи и проектировщика, работающего с очередью задач как имитации **СМО**, некоторые характеристики которой выступают в качестве метрик, описываемых в **Kanban**-методологии.

Также в главе раскрывается расширение языка псевдокодowego программирования  $L^{WQA}$ . Применение этого языка позволяет запрограммировать как особенности планирования параллельной проектной работы, так и особенности псевдопараллельного выполнения задачи проектировщиком, а точнее – его интеллектуальным процессором.

В третьей главе диссертации приводится сценарная структуризация средств ПКУ, систематизация задач управления проектом, описание методик решения задач проектного управления, в том числе задач вычисления метрических характеристик. Приводятся особенности экспериментирования в ПКУ.

Текст главы начинается со сценарной структуризации средств ПКУ, позволившей выделить **акторов** управления проектным процессом, а также – **набор задач ПКУ**.

К актерам в процессе ПКУ относятся следующие категории:

1. Руководитель проекта;
2. Проектировщики;
3. Другие стейкхолдеры (заказчики и др).

Если руководитель проекта и проектировщики принимают непосредственное участие в проектном процессе, то другие стейкхолдеры участвуют в метрической оценке проектного процесса.

Эти акторы решают следующие основные задачи ПКУ:

- Формирование потока работ проекта  $Z_F$ ;
- Выполнение проектной работы  $Z_W$ .

Задача формирования потока работ проекта, в свою очередь, включает в себя следующие подзадачи:

- Формирование набора задач проекта  $Z_{FZ}$ ;
- Формирование групп проектировщиков, ответственных за решение этих задач  $Z_{FG}$ ;
- Назначение задач проектировщикам и их группам  $Z_{FZa}$ ;

Задача выполнения проектной работы  $Z_W$  представляет собой итеративное выполнение следующих подзадач:

- Формирование бэклога этапа проектной работы  $Z_{WB}$ ;
- Выполнение этапа проектной работы  $Z_{WE}$ ;
- Формирование отчетов о выполненной работе  $Z_{WR}$ .

Задача формирования бэклога этапа проектной работы включает в себя задачу назначения поручений.

В свою очередь выполнение этапа проектной работы включает в себя:

- Выполнение проектных задач  $Z_{WEi}$ ;
- Вычисление метрик  $Z_{WEm}$ ;

Формирование отчетов о проделанной работе  $Z_{WRp}$ .

Для представления этих задач использованы **UML**-диаграммы вариантов использования. Общая диаграмма вариантов использования средств ПКУ представлена на рисунке 5. При более детальном рассмотрении данных задач было выделено **68** задач ПКУ (Рисунок 6).

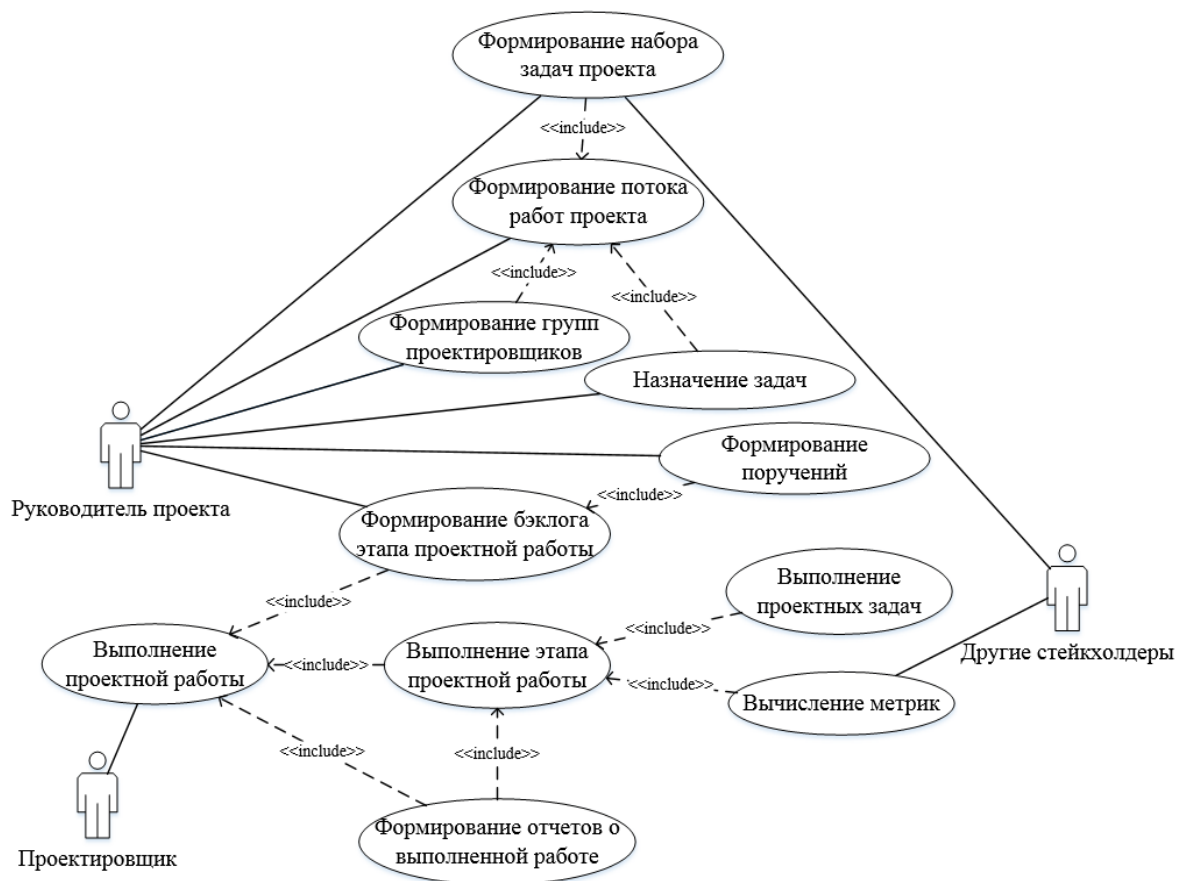


Рис. 5. Диаграмма вариантов использования, визуализирующая задачи ПКУ

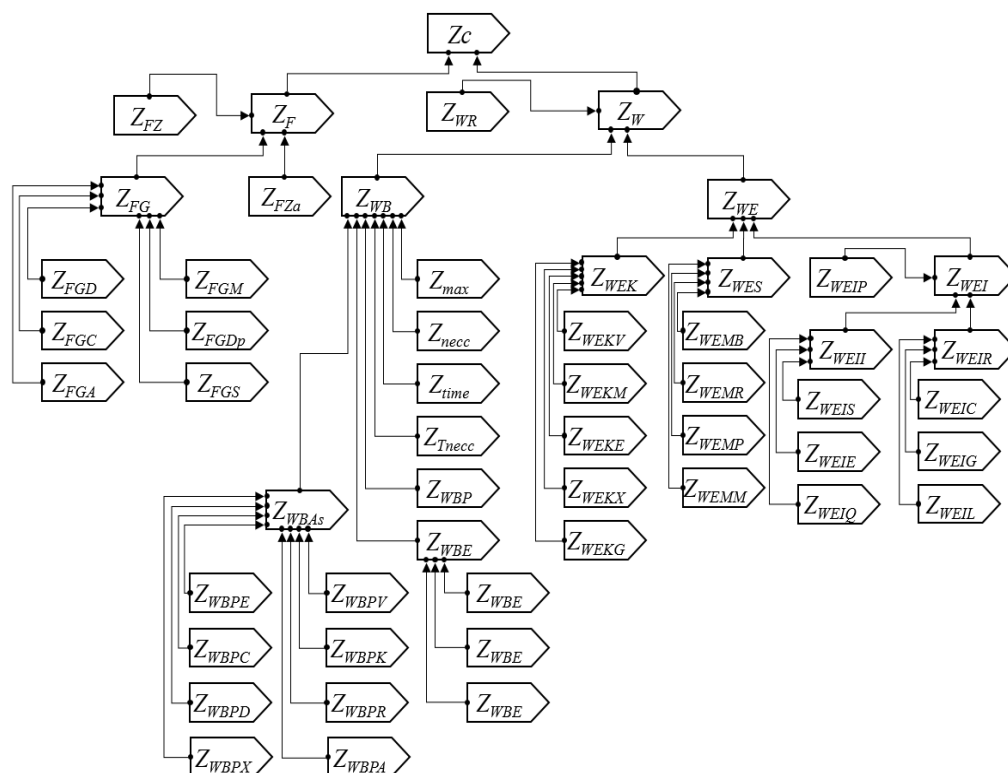


Рис. 6. Потoki работ управления проектированием

Эти задачи были систематизированы и представлены в виде набора потоков работ управления проектированием. Также были выделены артефакты, используемые и вырабатываемые в процессе выполнения работ ПКУ. В третьей главе также был



рассмотрен ряд методик выполнения работ программно-картотечного управления, связанных с поручениями, управлением очередями потоков работ, параллелизмом в процессе выполнения задач сотрудником, а также с вычислением метрических показателей проектного процесса.

В качестве примера методики можно взять методику создания поручения. Представим эту методику в виде последовательности шагов для их выполнения.

1. Просканировать отображения назначений задач на QA-дерево и обнаружить новые задачи;
2. Для обнаруженной задачи отобразить окно создания нового поручения;
3. Установить даты поручения;
4. Проверить по таблице Kanban, не возникает ли превышения числа выполняемых задач на данном этапе проектной работы;
5. Указать зависимости исполнения этого поручения от результатов выполнения других задач;
6. Установить исполнителя поручения;
7. Указать лиц, ответственных за контроль поручений;
8. Зарегистрировать поручения в структурах ПКУ;
  - 8.1. Зарегистрировать поручение в QA-представлении таблицы поручений;
  - 8.2. Сформировать карточку на доске Kanban;
    - 8.2.1. Зарегистрировать информацию о карточке Kanban в QA-представлении доски Kanban;
9. Конец методики.

Данная методика задействует сразу несколько компонентов системы ПКУ и использует результаты работы этих компонентов. Ряд шагов данной методики является автоматизированным с использованием языка  $L^{WIQA}$ . Далее представлен фрагмент псевдокодовой программы, отвечающей за формирование нового поручения и добавление его в базу данных поручений:

1. **&TaskTable& := QA\_GetTaskTable(&current\_project&)**
2. **&dbqaid&:=QA\_GetQAId(&current\_project&,"БД поручений")**
3. **&dbid& := OpenDB(&current\_project&, &dbqaid&)**
4. **&groups& := ExecuteSQL(&dbid&, "SELECT \* FROM Группы")**
5. **&subject& := ExecuteSQL (&dbid&, "SELECT \* FROM Субъект")**
6. **&mres& := AC ASSOCTASK(&groups&, &subject&, &TaskTable&)**
7. **&strc& := STRCMP (&mres&, "")**
8. **IF &strc& == 0 THEN GOTO &SF&**
9. **&query& := SPLITSTRING(&mres&, "??", 0)**
10. **ExecuteSQL(&dbid&, &query&)**

Другим примером методики является методика вычисления **Kanban**-метрики **CYCLE\_TIME** – среднего значения времени, которое задача проходит от момента ее постановки до завершения работы над задачей.

Для вычисления значения метрики **CYCLE\_TIME** – среднего значения времени, которое задача проходит от момента ее постановки до завершения работы над задачей – необходимо выполнить следующие действия:

- отобразить из таблицы контроля поручений выполненные задачи;
- для каждой задачи вычислить в днях разницу между началом ее постановкой и завершением работы над задачей и подсчитать среднее арифметическое этих разниц.

Выполнение такой работы осуществляется автоматически путем выполнения следующих шагов:

1. Получить идентификатор единицы QA-дерева, содержащего базу данных поручений;
2. Открыть базу данных и получить ее идентификатор;
3. Выполнить SQL-запрос, выбирающий выполненные поручения из таблицы;
4. Определить количество выбранных задач;
5. Определить счетчик дней в работе и проинициализировать нулем;
6. Для каждой задачи:
  - 6.1. Определить дату начала работы;
  - 6.2. Определить дату окончания работы;
  - 6.3. Добавить к счетчику дней в работе разницу в днях между определенными датами начала и окончания работы;
7. Разделить полученное количество дней на количество выполненных задач;
8. Конец методики.

Шаги данной методики реализованы на языке псевдокодowego программирования и предполагают только автоматическое их выполнение с использованием интерпретатора псевдокодowego языка программирования.

```

2. &dbid& := OpenDB (&current_project&, &dbqa&);
3. &tasks& := ExecuteSQL(&dbid&, "SELECT Дата_исполнения, Дата_поручения
FROM Задачи WHERE Статус = 1");
4. &ntasks& := DB_GETROWCOUNT(&tasks&);
5. &NCNT& := 0;
6. LABEL &CCT_L1&;
6.1. IF &NCNT& >= &ntasks& THEN GOTO &CCT_L2&;
6.2. &DSTART& := DB_GETCELLDATETIME(&tasks&, &NCNT&, 1);
6.3. &DEND& := DB_GETCELLDATETIME(&tasks&, &NCNT&, 0);
6.5. &NDAYS& := &NDAYS& + DT_GETDATEDIFFERENCE(&DEND&,
&DSTART&);
6.6. &NCNT& := &NCNT& + 1;
6.7. GOTO &CCT_L1&;
7. &NDAYS& := 0;
8. LABEL &CCT_L2&;
9. &CYCLE_TIME& := &NDAYS& / &NCNT&.
  
```

Далее процесс ПКУ был рассмотрен с позиции его *эмпиричности*. Было выделено два класса возможностей в экспериментировании в процессе ПКУ. Решая назначенную ему задачу в ее псевдокодовой представлении, проектировщик может прервать исполняемую им работу до или после любого оператора и провести концептуальный эксперимент или концептуальный анализ и оценивание, если он видит в этом необходимость. По ходу исполнения параллельных или псевдопараллельных работ имеется возможность для вычисления совокупности метрик, значения которых можно и следует использовать для воздействий на формирование групп проектировщиков, отбор задач в бэклоги и на выбор задач из их очередей.

В четвертой главе раскрыты вопросы реализации системы программно-картотечного управления потоками проектных работ, а также представлено описание проведения экспериментального исследования средств ПКУ с использованием их в реальной проектной работе на предприятии. Приведен пример программирования потока работ и очереди задач проектировщика, а также – программируемая настройка средств ПКУ на использование различных методологий проектирования.

Комплекс средств ПКУ реализован в двух версиях среды **WIQA** – многопользовательской сетевой **WIQA.Net** и однопользовательской **WIQA**. В большинстве своем компоненты ПКУ являются универсальными и используются в обеих средах за небольшими отличиями.

Рассмотрим реализацию ПКУ в среде **WIQA.Net**. В этом случае среда **WIQA** размещается на трех типах устройств:

- Сервер баз данных;
- Сервер **WIQA.Net**;
- Клиент **WIQA.Net**.

**Сервер баз данных** содержит **СУБД MS SQL**, в которую отображается память **WIQA** для ее хранения. **Сервер WIQA.Net** взаимодействует как с базой данных, так и с клиентом, таким образом, он является интерфейсом между клиентом **WIQA** и сервером баз данных.

Устройство **Клиент WIQA.Net** отвечает непосредственно за выполнение алгоритмов компонентов системы ПКУ. Клиент включает в себя приложение **WIQA клиент**, а также ряд дополнительных подключаемых библиотек, отвечающих за графический пользовательский интерфейс псевдокодовых компонентов ПКУ.

Приложение **WIQA-клиент** содержит следующие компоненты, связанные с функционированием ПКУ:

- Вопросно-ответный протокол;
- Демон отображения оргструктуры;
- Интерпретатор псевдокода.

Вопросно-ответный протокол применяется для создания, просмотра и редактирования данных, хранящихся в **QA-памяти WIQA**, в том числе – и задач проекта. Демон отображения оргструктуры осуществляет преобразование оргструктуры в **QA-формат** и отображение ее на **QA-память WIQA**. Интерпретатор псевдокода позволяет выполнить псевдокодовые алгоритмы ПКУ.

Теперь рассмотрим компоненты ПКУ, относящиеся к серверу **WIQA**. Эти компоненты представлены тремя типами: **QA-данные**, **QA-программы** (компоненты ПКУ, запрограммированные на языке псевдокода), а также данные, не имеющие вопросно-ответной структуры. Из используемых в подсистеме ПКУ к таким данным относятся данные оргструктуры.

К серверным компонентам ПКУ относятся:

- **QA-БД проекта** – вопросно-ответное представление проекта;
- **Оргструктура** – представление данных оргструктуры в памяти **WIQA**;
- **QA-отображение оргструктуры** – отображение, полученное с помощью демона отображения оргструктуры;
- **QA-БД поручений** – QA-представление таблиц поручений;
- **Псевдокоды поручений** – программные коды на языке **L<sup>WIQA</sup>**, отвечающие за работу подсистемы поручений;
- **QA-БД очередей** – QA-представление таблиц очередей задач и прерываний;
- **Псевдокоды очередей и прерываний** – псевдокодовая программная составляющая подсистемы управления очередями и прерываниями в ПКУ;
- **QA-БД Kanban** – QA-представление доски **Kanban**;
- **Псевдокоды Kanban** – алгоритмическое обеспечение **Kanban**;
- **QA-БД Scrum** – QA-представление данных **Scrum**;
- **Псевдокоды Kanban** – алгоритмическое обеспечение **Scrum**;
- **Псевдокоды метрик** – алгоритмы вычисления метрик **Kanban** и **Scrum**.

Перечисленные выше компоненты ПКУ в среде **WIQA.Net** можно обобщить в виде диаграммы размещения, представленной на рисунке 7.

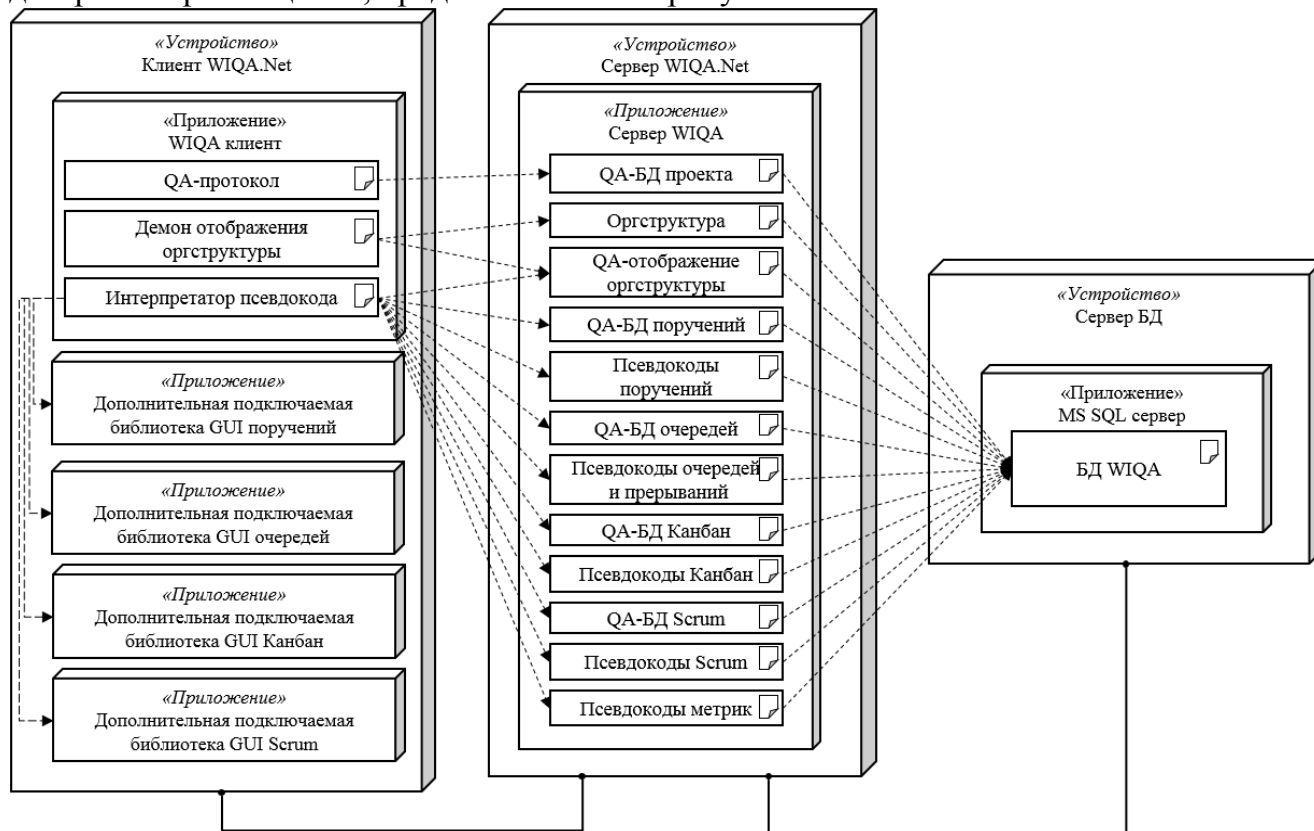


Рис. 7. Размещение компонентов ПКУ в WIQA.Net.

Среда **OwnWIQA** отличается отсутствием клиент-серверной функциональности, а также – организационной структуры, что потребовало для реализации комплекса ПКУ включить в него следующие составляющие:

- **QA-БД имитации оргструктуры** – вопросно-ответное представление оргструктуры для обеспечения функционирования средств ПКУ;
- **Псевдокоды имитации оргструктуры** – программные компоненты, обеспечивающие работу с данными, имитирующими оргструктуру;
- **Синхронизатор** – инструментарий, обеспечивающий репликацию БД **WIQA** для обеспечения взаимодействия средств ПКУ со средствами ПКУ других участников проекта.

Кроме этого, была реализована еще одна библиотека:

- **Дополнительная подключаемая библиотека GUI имитации оргструктуры.**

Большинство компонентов средств ПКУ реализованы как псевдокодовые процедуры. Взаимодействие с пользователем осуществляется путем дополнительных библиотек, реализующих диалоговые окна.

**В главе представлены экспериментальные исследования программно-картотечного управления.** В процессе проведения экспериментов оценивалась возможность применения разработанных средств ПКУ в условиях реального проектирования. Экспериментальное исследование проводилось на следующих видах проектных работ:

- Проект «Создание единого интегрированного справочника», выполняемый на предприятии ФНПЦ ОАО НПО «МАРС»;

- Проект «Образовательная Web-система», выполняемый на предприятии ООО «Центр технического творчества»;
- Проект, состоящий из работ над данным диссертационным исследованием;
- Проект, состоящий из задач студенческой олимпиады по программированию.

В данной главе раскрыт эксперимент, проведенный в ходе выполнения проекта «Создание единого интегрированного справочника». Остальные эксперименты рассмотрены в **Приложении 2** диссертационной работы.

В ходе выполнения данного эксперимента были запрограммированы механизмы параллельного выполнения потока работ проекта и очередей задач проектировщиков. Фрагментарно эти программы представлены на рисунке 8.

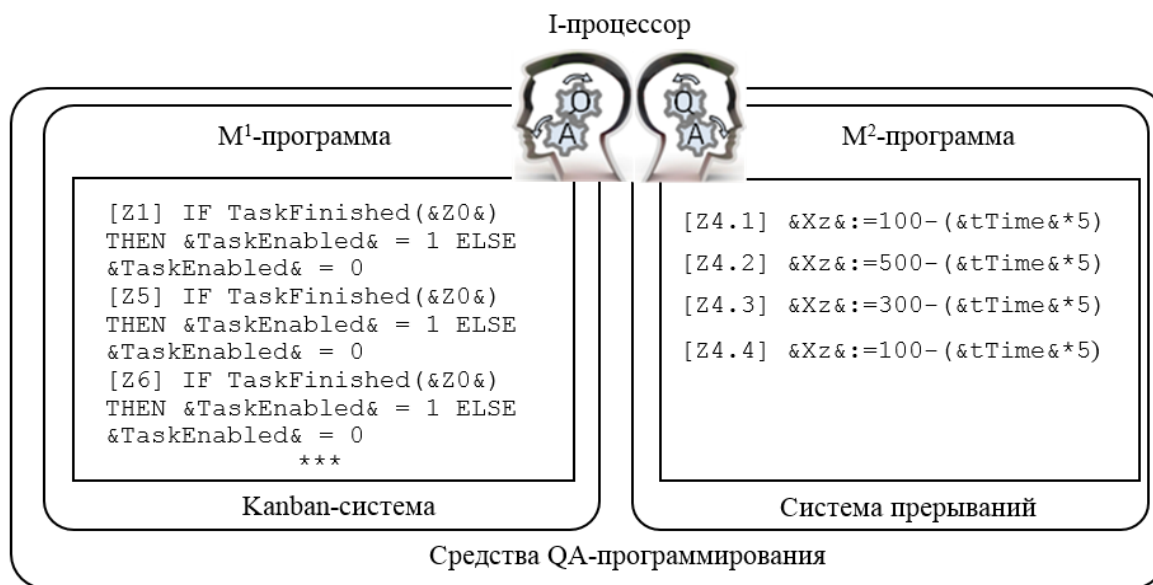
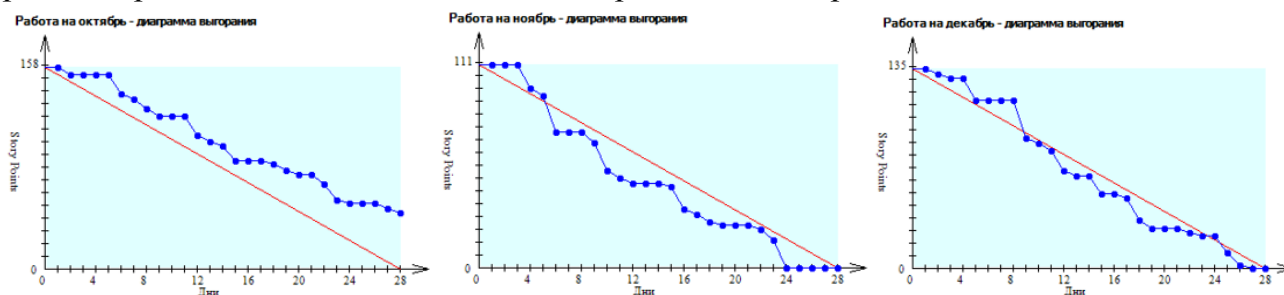


Рис. 8. Программирование параллелизма в выполнении проектных задач

В алгоритме параллельного выполнения потока работ были использованы конструкции, соответствующие таким шаблонам потоков работ, как последовательное выполнение задач (*Sequence*), распараллеливание (*Parallel split*), и синхронизация (*Synchronization*).

На рисунке 9 представлены три диаграммы выгорания спринтов, выполненных в рамках проекта «Создание единого интеграционного справочника».



Из этих диаграмм видно, что применение данной системы позволяет совершенствовать рациональность планирования этапа проектной работы. Самый первый спринт, выполненный в октябре 2014 года, по пятибалльной шкале был оценен группой разработчиков на 3 балла по рациональности планирования, ноябрьский спринт – на 4 балла, а спринт, выполненный в декабре – на 4,75 баллов. Оценка декабрьского спринта

была выполнена автоматически с использованием данных, полученных в результате выполнения первых двух спринтов.

Также в главе раскрыты особенности адаптации разработанных программных средств к применению методологий **Kanban**, **Scrum** и **Scrumban**, приведены применяемые алгоритмы программного управления

**В приложениях** раскрыты псевдокодовые реализации базовых шаблонов управления потоками работ, а также – раскрыты эксперименты, сделанные в ходе выполнения проекта «Образовательная Web-система», проекта, состоящего из работ над данным диссертационным исследованием и проекта, состоящего из задач студенческой олимпиады по программированию.

### **Основные результаты работы**

Подводя обобщающий итог диссертационному исследованию и практическим разработкам, реализованным на базе результатов исследований, можно утверждать следующее:

**Цель исследований**, повышение степени автоматизации коллективных и персональных действий проектировщиков в потоках работ концептуального проектирования АС, а также снижение ошибок проектирования, обусловленное автоматизацией и включением в средства управления возможностей экспериментирования с решениями задач в исполняемых потоках в реальном времени, достигнута.

Предложена, исследована и проверена совокупность средств и моделей, на базе которых осуществляется программно-картотечное управление потоками работ в проектировании АС.

### **Получены новые научные результаты:**

1. Программно-картотечная модель гибкого управления проектированием автоматизированных систем;
2. Представлено подмножество псевдокодowego алгоритмического языка, ориентированного на оперативное управление потоками работ, а также набор механизмов алгоритмического программирования параллелизма в выполнении проектных задач как на уровне потока работ проекта, так и на уровне очереди задач проектировщика;
3. Набор методик управления поручениями, формирования бэклога этапа проектной работы, вычисления метрических характеристик проектной работы, параллельного выполнения очередей задач проектировщиком и их группой;
4. Библиотека псевдокодовых программ и программных моделей паттернов потоков работ, облегчающая программно-картотечное управление коллективным проектированием АС.

### **Практическую ценность составляют:**

Разработанный комплекс средств, обеспечивающий программно-картотечное управление потоками работ в проектировании АС, а также – библиотека псевдо-кодовых реализаций паттернов потоков работ.

## **ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ**

### **Публикации в изданиях, рекомендованных ВАК РФ**

1. Ромодин, М.Ю. Псевдокодовое программирование в концептуальном проектировании баз данных / М.Ю. Ромодин, Ю.А. Лапшов, П.И. Соснин // Автоматизация процессов управления. – 2013. – №2(32). – С. 94-100.
2. Соснин, П.И. Инструментальные средства программного управления потоками работ в

проектировании автоматизированных систем / П.И. Соснин, Ю.А. Лапшов, В.А. Маклаев // Автоматизация процессов управления. – 2012. – №4(30). – С. 74-82.

3. Соснин, П.И. Псевдо-кодированное программное управление потоками работ в проектировании автоматизированных систем / П.И. Соснин, Ю.А. Лапшов, В.А. Маклаев // Автоматизация процессов управления. – 2012. – №3(29). – С. 42-49.

### **Монография**

4. Соснин, П.И. Программное управление потоками работ в компьютеризованных средах (монография) / П.И. Соснин, Ю.А. Лапшов, С.Н. Ларин, В.А. Маклаев // под ред. П.И. Соснина. – Ульяновск : УлГТУ, 2014. – 308 с.

### **Публикация, индексируемая в SCOPUS**

5. Sosnin, P.I. Programmable Managing of Workflows in Development of Software-Intensive Systems / P.I. Sosnin, Y.A. Lapshov, K.V. Svyatov // The 27th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, Volume: Part 1. – At Kaohsiung, Taiwan, 2014.

### **Свидетельства о государственной регистрации программы для ЭВМ**

6. Соснин, П.И. Компилятор псевдокодированных программ / П. И. Соснин, Ю.А. Лапшов, К.В. Святков // Свидетельство о государственной регистрации программы для ЭВМ №2015610593.
7. Соснин, П.И. Программный инструментальный контроль поручений / П.И. Соснин, Ю.А. Лапшов, С.Н. Ларин, В.А. Маклаев // Свидетельство о государственной регистрации программы для ЭВМ №2015610236.

### **Публикации в других изданиях**

8. Lapshov, Y.A. Program management of workflows in conceptual designing of automated systems / Y.A. Lapshov, V.A. Maklaev, P.I. Sosnin // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. научн.-техн. конф. (Минск, 21-23 февраля 2013г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013.
9. Lapshov, Y.A. Pseudo-Code Programming of Workflows in Conceptual Designing of Software Intensive System / Y.A. Lapshov, V.A. Maklaev, P.I. Sosnin // Interactive Systems: Problems of Human – Computer Interaction. – Collection of scientific papers. – Ulyanovsk: UlSTU, 2013. – С. 40-52.
10. Лапшов, Ю.А. Средства псевдокодированного моделирования и программирования проектных решений с использованием баз данных / Ю.А. Лапшов, В.А. Маклаев, М.Ю. Ромодин // Информатика и вычислительная техника: сборник научных трудов Т.1/ под ред. Н.Н. Войта. – Ульяновск: УлГТУ, 2012. – С. 427-439.
11. Лапшов, Ю.А. Учет сложности в управлении потоками работ в проектировании автоматизированных систем / Ю.А. Лапшов, В.А. Маклаев, П.И. Соснин // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н.Н. Войта. – Ульяновск : УлГТУ, 2012. – С. 15-23.
12. Лапшов, Ю.А. Процесс работы в коллективной среде проектирования в условиях прерываний как система массового обслуживания / Ю.А. Лапшов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н. Н. Войта. – Ульяновск : УлГТУ, 2011. – С. 277-282.
13. Лапшов, Ю.А. Библиотека паттернов потоков работ / Ю.А. Лапшов, В.А. Маклаев // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н.Н. Войта. – Ульяновск : УлГТУ, 2011. – С. 272-277.
14. Лапшов, Ю.А. Управление человеческими прерываниями в корпоративных средах моделирования / Ю.А. Лапшов // Информатика и вычислительная техника : сборник научных трудов / под ред. В.Н. Негоды. – Ульяновск, УлГТУ, 2010. – С. 321-326.
15. Лапшов, Ю.А. Объектно-ориентированное вопросно-ответное псевдокодированное программирование / Ю.А. Лапшов, В.А. Маклаев, П.И. Соснин // Информатика и

- вычислительная техника: сборник научных трудов / под ред. Н. Н. Войта. – Ульяновск: УлГТУ, 2011 – С. 357-362.
16. Лапшов, Ю.А. Псевдокодирование программирования управления прерываниями человека / Ю.А. Лапшов // Информатика и вычислительная техника: сборник научных трудов / под ред. Н. Н. Войта. – Ульяновск: УлГТУ, 2011 – С. 353-356.
  17. Лапшов, Ю.А. Реализация компилятора функций вопросно-ответных псевдокодовых программ / Ю.А. Лапшов // Информатика и вычислительная техника: сборник научных трудов / под ред. Н.Н. Войта. – Ульяновск: УлГТУ, 2011 – С. 363-367.
  18. Лапшов, Ю.А. Приоритеты задач в коллективной среде проектирования / Ю. А. Лапшов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н.Н. Войта. – Ульяновск : УлГТУ, 2011. – С. 272-276.
  19. Лапшов, Ю.А. Процесс работы в коллективной среде проектирования в условиях прерываний как система массового обслуживания / Ю.А. Лапшов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н.Н. Войта. – Ульяновск : УлГТУ, 2011. – С. 277-282.
  20. Lapshov, Y. Object-Oriented Question-and-Answer Pseudo-Code Programming / Y. Lapshov, V. Maklaev, P. Sosnin // Interactive Systems and Technologies: the Problems of Human-Computer Interaction. Volume III. – Collection of scientific papers. – Ulyanovsk: UlSTU, 2011. – С. 248-252.
  21. Lapshov, Y. Implementation of Compiler for Functions of Question-and-Answer Pseudo-Code Programs / Y. Lapshov, V. Maklaev // Interactive Systems and Technologies: the Problems of Human-Computer Interaction. Volume III. – Collection of scientific papers. – Ulyanovsk: UlSTU, 2011. – С. 275-279.
  22. Lapshov, Y.A. Human interruptions management in corporate modeling environment WIQA.Net / Y.A. Lapshov // Interactive Systems and Technologies: the Problems of Human-Computer Interaction. Volume III. – Collection of scientific papers. – Ulyanovsk: UlSTU, 2009. – С. 229-234.

## ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

|      |                                      |          |                                  |
|------|--------------------------------------|----------|----------------------------------|
| АС   | Автоматизированная система           | QA       | Question-Answer                  |
| ГОСТ | Государственный стандарт             | RUP      | Rational Unified Process         |
| ПКУ  | Программно-картотечное<br>Управление | SIS      | Software Intensive System        |
| СМО  | Система массового обслуживания       | WIQA.NET | Working In Questions and Answers |

Лапшов Юрий Александрович

Средства программно-картотечного управления потоками работ в коллективном проектировании автоматизированных систем

Автореферат

Подписано в печать 23.04.2015. Формат 60х84/16.

Усл. п. л. 1,4.

Тираж 100 экз. Заказ \_\_\_\_.

ИПК «Венец» УлГТУ. 432027. Ульяновск, Сев. Венец, 32