ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 7](#_Toc295397031)

[1 СПЕЦИАЛЬНАЯ ЧАСТЬ 9](#_Toc295397032)

[1.1 Концепция проекта 9](#_Toc295397033)

[1.2 Сценарии использования 54](#_Toc295397034)

[1.3 Логический дизайн 101](#_Toc295397035)

[1.4 Физический дизайн 118](#_Toc295397036)

[1.5 План пилотного внедрения 125](#_Toc295397037)

[2 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 133](#_Toc295397038)

[2.1 Аннотация 133](#_Toc295397039)

[2.2 Назначение разработки 135](#_Toc295397040)

[2.3 Условия выполнения программы 136](#_Toc295397041)

[2.4 Выполнение программы 137](#_Toc295397042)

[2.5 Сообщения оператору 141](#_Toc295397043)

[3 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ 146](#_Toc295397044)

[3.1 Анализ достоинств интерфейса пользователя разработанной программы 146](#_Toc295397045)

[3.2 Рекомендации по организации рабочего места пользователя 148](#_Toc295397046)

[3.3 Рекомендации по организации режима труда и отдыха пользователя 165](#_Toc295397047)

[4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ 168](#_Toc295397048)

[4.1 Сетевое планирование 168](#_Toc295397049)

[4.2 Смета затрат на проведение работ 185](#_Toc295397050)

[4.3 Анализ технико-экономической эффективности 190](#_Toc295397051)

[ЗАКЛЮЧЕНИЕ 193](#_Toc295397052)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 194](#_Toc295397053)

[ПРИЛОЖЕНИЯ 197](#_Toc295397054)

[ПРИЛОЖЕНИЕ А. Пример файла конфигурации   
системы сборки и тестирования программного   
обеспечения 197](#_Toc295397055)

[ПРИЛОЖЕНИЕ Б. Пример файла конфигурации поведения веб-интерфейса 199](#_Toc295397056)

# ВВЕДЕНИЕ

Современный бизнес, основанный на разработке ПО, требует в короткие сроки разрабатывать программное обеспечение, удовлетворяющее требованиям заказчика. Также из-за роста сфер применения и ответственности функций, выполняемых программами, требуется гарантировать высокое качество программных продуктов. Вследствие этого с каждым годом растет спрос на высококвалифицированных специалистов в области разработки ПО.

С ростом применения гибких методологий разработки ПО, таких как Agile, Scrum, Kanban, всё большая роль отводится интеграционному и нагрузочному тестированию, при том, что сами тесты выступают в роли критерия приемки текущей выпускаемой версии программного продукта.

С другой стороны, непосредственный процесс написания программного кода, строится с применением методик экстремального программирования (XP), включающих в свой состав так называемый подход разработки через тестирования (TDD, test-driven development). Соответственно в данном случае критерием качественной и правильной реализации конкретной функциональности в разрабатываемом программном продукте является проходящий модульный тест.

Одной из задач современного образования, целью которого является подготовка высококвалифицированного специалиста в области разработки ПО, можно считать подготовку студентов к работе в вышеозначенных условиях. Поскольку, по большей части, студенческие работы, особенно на младших курсах, не подразумевают создание крупных программных систем, то можно рассматривать некий сдвиг модульного тестирования в область приемочных тестов для всего программного продукта.

Становится очевидной актуальность использования модульного тестирования как способа не только приемки, но и оценки работ студентов, поскольку гибко варьируя сложность самих тестов можно определять степень удовлетворения конкретного программного продукта заданным требованиям.

Все это требует создания не только системы непосредственно запуска модульных тестов, но и тесного интегрирования подобной системы в учебный процесс, связи ее с существующими инструментами поддержания этого процесса, предоставления преподавателям мощных инструментов для управления системой.

В связи с этим, создание программной системы модульного тестирования студенческих работ с возможностью гибкого конфигурирования и автоматизации бизнес-процессов преподавателей, связанных с проверкой работ студентов, является актуальной и нетривиальной задачей.

# СПЕЦИАЛЬНАЯ ЧАСТЬ

Специальная часть содержит весь процесс разработки программного проекта: от анализа проблем заказчика до спецификации готового решения.

## Концепция проекта

Концепции проекта описывает идеи и решения, выработанные на первой фазе проектирования. Цель данной фазы – представить соглашение, достигнутое разработчиком и заказчиком, связанное с общим видением проекта [1].

### Бизнес проблемы

Изначально поставленная заказчиком проблема состояла в том, что остро стояла проблема нехватки времени на проверку большого числа работ студентов, выполненных в рамках обучения дисциплинам цикла «Программирование», а также отстутствовали механизмы четкого указание критериев, предъявляемых к программному коду работ.

### Описание текущей ситуации

На текущий момент в рамках цикла «Программирование» проверка студенческих семестровых и других контрольных работ, представляющих собой небольшие программы на языке C++, происходит вручную. Преподаватель вынужден запускать каждую программу на определенных наборах входных значений, чтобы убедиться в корректности ее работы, после чего произвести общую оценку кода на соответствие поставленной задаче и выставить оценки, руководствуясь определенными требования к функциональности работ, вытекающими из формулировки задания. Взаимодействие со студентами, включающее в себя выдачу задания и прием готовых программ осуществляется посредством системы управления учебными курсами Moodle [13], в результате чего перед началом проверки заданий преподаватель скачивает их к себе на компьютер и в ручном режиме выполняет построение проектов, чтобы убедится в отсутствии ошибок компиляции.

Бизнес-процессы Заказчика (преподавателя) подразумевают выполнение большого объема рутинных операций, которые можно и нужно автоматизировать, дабы избежать перерасхода времени на повторяющиеся действия. Кроме того, со стороны студентов происходит ожидание оценки результатов своего труда, чего можно избежать, если процесс проверки и выставления оценки также сделать автоматическим, задавая критерии оценки извне.

Ниже описывается текущая ситуация у заказчика, которая вызвала необходимость создания данного проекта. Описание выполнено с помощью таблицы с раскрытием сути проблемы заказчика (таблица 1.1).

Таблица 1.1 – Описание текущей ситуации

|  |  |
| --- | --- |
| Свойство | Значение |
| Проблема | Необходимость выполнения рутинных операций по скачиванию, упорядочиванию, построению работ студентов, ручному выполнения программ и оценке правильности кода |
| Воздействует на | Студентов и преподавателей |
| Результатом чего является | Перерасход времени преподавателя, существует риск ошибки, связанный с пропущенным заданием или специфичными настройками на машине преподавателя |
| Выигрыш от | Система модульного тестирования для приема и оценки заданий по дисциплинам цикла «Программирование» |

Продолжение таблицы 1.1

|  |  |
| --- | --- |
| Свойство | Значение |
| Успешное  решение должно | Сократить время ожидания для студента и практически нивелировать временные затраты преподавателя на выполнение нужных действий, уменьшить вероятность ошибки, формализовать требования к программному коду работ и решений задач |

### Описание будущей ситуации

Проектируемая система будет обеспечивать:

1. Автоматическое получение исходных кодов работ студентов из системы управления учебными курсами Moodle 2.0, на основе которой создан веб-сайт item74.ru.
2. Автоматическую сборку студенческих работ и запуск модульных тестов, написанных преподавателем.
3. Создание отчетов о результатах сборки и тестирование студенческих работ.
4. Возможность отправки уведомлений об изменения состояния тестирования по электронной почте.

Система будет иметь интуитивно понятный интерфейс пользователя.

### Бизнес-анализ

Это выгода полностью основана на бизнес-целях заказчика и ожиданиях от реализации данного проекта [2]. Бизнес-анализ разрабатываемого проекта для учебного процесса проводится с помощью нижеследующей таблицы 1.2.

Таблица 1.2 – Бизнес-анализ проекта для учебного процесса

|  |  |
| --- | --- |
| Свойство | Значение |
| Для кого | Преподавателя и обучающихся |
| Кто | Желает получить максимально автоматизированную систему с минимальным вмешательством, сводящимся к указанию условий задания и написанию модульных тестов для него |
| Что (название продукта) | Система модульного тестирования для приема и оценки заданий по дисциплинам цикла «Программирование» |
| Как | Обеспечивает полностью автоматически процесс проверки работ на удовлетворения критериям, заданным в виде модульных тестов, интегрируется с платформой Moodle 2.0, на которой основан сайт Item74 и позволяет получать сводный отчет об обработанных работах |
| В отличие от | TeamCity, CruiseControl, Jenkins |
| Наш продукт | Система нацелена на тесную интеграцию с Moodle 2.0, более легковесна и проста в настройке. Позволяет работать без организации репозиториев с использованием VCS. Позволяет собирать проект вместе с кодом модульных тестов задаваемых извне. Содержит настройки, специфичные для поддерживаемого языка – C++ |

### Сравнение отечественных и передовых зарубежных технологий и решений

В этом пункте приведен подробный анализ аналогов проектируемой системы. Согласно требованиям к функциональности разрабатываемого проектного решения ближайшим его аналогом являются системы непрерывной интеграции (Continuous Integration).

При выборе средств и их сравнении учитываются следующие общие критерии:

1. Надежность – отсутствие отказов и сбоев в работе программ, а также простота исправления дефектов и ошибок.
2. Документированность – качество документации и степень ее полноты, также учитывается документированность исходного кода, если он открыт.
3. Модульность и возможность расширения функционала – модульность архитектуры программной системы; насколько сложно изменить программу для удовлетворения новых требований, реализации нового функционала; учитывается наличие дополнительных плагинов.
4. Кроссплатформенность (особенно важно для клиентской части системы).
5. Возможность интеграции с другими средствами и системами – важный критерий использования программных средств для построения системного решения.
6. Популярность – насколько программный продукт популярен среди разработчиков, современен, учитывается развитие функционала продукта с расчетом на будущее.
7. Стоимость – стоимость покупки программного продукта, но сюда также можно отнести стоимость внедрения и интеграции, доработки под поставленные требования.

Системы непрерывной интеграции. Непрерывная интеграция – это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем [3]. Соответственно под системой непрерывной интеграции понимается такая система, которая обеспечивает поддержание процесса автоматической сборки проекта и выявления интеграционных проблем, путем запуска модульных или интеграционных тестов.

Ниже приведены наиболее известные из существующих на данный момент систем непрерывной интеграции:

1. TeamCity.
2. CruiseControl.
3. Jenkins.
4. Apache Gump.
5. Microsoft Team Foundation Server.

TeamCity (www.jetbrains.com/teamcity) – система непрерывной интеграции от компании JetBrains, написанная на языке Java. Является проприетарным программным обеспечением [36].

В набор стандартных функциональных возможностей входит:

1. Поддержка программных продуктов, написанных на языках Java, Ruby или на платформе .NET.
2. Наличие расширений для интегрированных сред разработки: Eclipse, IntelliJ IDEA, Visual Studio.
3. Интегрированные средства для оценки покрытия кода модульными тестами, поиск дублирования в коде.
4. Встроенная поддержка распределенной работы с использованием grid-технологий для параллельной сборки и запуска тестов в разными параметрами окружения.
5. Интеграция с системами отслеживания ошибок.
6. Администрирование системы через веб-интерфейс.
7. Гибкое управление правами доступа к системе для пользователей на основе ролей и групповых политик безопасности.
8. Кроссплатформенность.

TeamCity поддерживает следующие системы управления версиями для получения исходного кода проектов:

* SVN;
* Perforce;
* CVS;
* Git;
* Mercurial;
* SourceGear Vault;
* Microsoft Visual SourceSafe;
* Team Foundation Server.

TeamCity поддерживает создание отчетов в разметке XML-документа, используя различные форматы: Ant, NUnit, Surefire, PMD и другие.

Система непрерывной интеграции TeamCity поддерживает написание декларативных расширений, используя язык разметки XML, содержащих дополнительную информацию о собираемых проектах, которая может использоваться непосредственно самой системой. Также возможно создание расширений (плагинов) на языке Java, используя OpenAPI, дающий возможность интеграции TeamCity с системами сборки, программами оповещения разработчиков или средами разработки.

CruiseControl (cruisecontrol.sourceforge.net) – система непрерывной интеграции, написанная на языке Java, и являющаяся программным обеспечением с открытым исходных кодом [37].

Возможности системы CruiseControl:

1. Поддержка программных продуктов, написанных на языках Java, Ruby или на платформе .NET.
2. Поддержка средств сборки программного обеспечения Ant, NAnt, Maven, PHing, Rake.
3. Отправка уведомлений по электронной почте.
4. Администрирование системы через веб-интерфейс.
5. Кроссплатформенность.

CruiseControl позволяет получать исходный код из следующих систем управления версиями:

* SVN;
* Git;
* Mercurial;
* CVS;
* Team Foundation Server.

CruiseControl поддерживает создание расширений, а также генерацию отчетов, представлямых либо в виду отдельной HTML-страницы, либо отображаемых на так называемой «приборной панели» (dashboard) веб-интерфейса системы.

Jenkins(jenkins-ci.org) – система непрерывной интеграции, написанная на языке Java, и являющаяся программным обеспечением с открытым исходных кодом, распространяемым под лицензией MIT [4].

Возможности системы Jenkins:

1. Поддержка программных продуктов, написанных на языке Java.
2. Поддержка средств сборки программного обеспечения Ant и Maven.
3. Администрирование системы через веб-интерфейс.
4. Кроссплатформенность.
5. Взаимодействие с системами управления версиями SVN, CVS и Git.
6. Запуск сборки и тестирования проектов по расписанию, или по возникновению определенного события в системе управления версиями.

Одной из особенностей системы непрерывной интеграции Jenkins является наличие двоичных сборок системы под множество целевых операционных систем.

Jenkins позволяет создавать плагины на языке Java, призванные расширить функциональность системы в части интеграции ее с системами отслеживания ошибок или инструментами сборки.

**Выводы по обзору систем непрерывной интеграции.** В целом, функциональные возможности, рассмотренных систем непрерывной интеграции схожи, однако существуют и различия.

Так, например, Jenkins обладает более бедными возможностями из всех приведенных в обзоре систем, а TeamCity и CruiseControl в целом одинаковы по своей функциональности.

К особенностям TeamCity можно отнести распределение процесса сборки и тестирования по многим серверам, а также, несмотря на проприетарность, возможность создания расширений на языке Java.

#### Общие выводы по обзору существующих решений среди систем непрерывной интеграции. Если обобщить все выводы, то рассмотренные решения имеют следующие существенные недостатки при использовании их в качестве основы для построения системы, удовлетворяющей предъявляемым заказчиком бизнес-требованиям:

1. Невозможность адаптации под бизнес-процессы заказчика.
   * все системы ориентированы на использование систем управления версиями, в качестве репозиториев для хранения исходного кода;
   * все системы рассчитаны на сборку одного проекта на одной конкретной конфигурации или нескольких конфигурациях;
   * вышеприведенные системы не поддерживают язык C++ и библиотеку UnitTest++.
2. Ориентация на комплексную поддержку процесса разработки ПО для крупных программных продуктов.
3. Избыточность функциональных возможностей, тяжеловесность.
4. Сложность настройки.

Таким образом, создание системы, которая бы отвечала всем требованиям заказчика, на основе рассмотренных решений невозможно, а модернизация и доработка систем непрерывной интеграции с открытым исходным кодом затруднена.

### Концепция решения

Концепция решения содержит общее описания подхода проектной команды к удовлетворению потребностей пользователя. Она включает в себя понимание потребностей пользователя, круга пользователей и заинтересованных лиц, описание возможностей и функций будущей системы.

Концепция решения обеспечивает команду ограниченными, но достаточными деталями для выработки в дальнейшем законченного решения, при этом она включает в себя анализ рисков проекта, анализ осуществимости проекта, удобство использования готового продукта, а также анализ будущих эксплуатационных характеристик.

**Анализ факторов риска проекта.** Анализ факторов риска проводится   
для своевременного принятия мер по уменьшению их вероятности или уменьшения последствий [17]. Они могут быть связаны:

* с недостаточным уровнем знаний и навыков разработчиков   
  (таблица 1.3);
* с серьезной болезнью разработчика (таблица 1.4);
* с выходом из строя оборудования для разработки и тестирования проекта (таблица 1.5);
* с непреодолимой сложностью в настройке интегрируемых средств (таблица 1.6);
* с возникновением проблем с обучением пользователей (таблица 1.7);
* с изменением требований к проекту (таблица 1.8);
* с выявлением ошибок, сделанных на ранних этапах проектирования (таблица 1.9).

Таблица 1.3 – Риск срыва заданных сроков дипломного проектирования

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Срыв заданных сроков дипломного проектирования |
| Возможные последствия | Отсутствие готового проекта на защите диплома |
| Вероятность | Средняя |
| Меры по предотвращению | Учитывать возможные задержки этапа разработки, предусмотреть резерв времени |
| Меры по минимизации | Перераспределение времени, отведенного на разработку дипломного проекта |

Таблица 1.4 – Риск болезни разработчика

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Болезнь разработчика |
| Возможные последствия | Замедление разработки программного продукта |
| Вероятность | Низкая |
| Меры по предотвращению | Профилактика, отдых |
| Меры по минимизации | Лечение |

Таблица 1.5 – Риск выхода из строя оборудования

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Выход из строя оборудования для разработки  и тестирования проекта |
| Возможные последствия | Замедление или невозможность разработки программного продукта |
| Вероятность | Средняя |
| Меры по предотвращению | Иметь возможность развернуть тестирование  на другом оборудовании |
| Меры по минимизации | Ремонт или замена оборудования |

Таблица 1.6 – Риск возникновения ошибок в интегрируемых средствах

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Непреодолимая сложность в настройке или найденные критические ошибки интегрируемых средств |
| Возможные последствия | Невозможность дальнейшей разработки программного продукта |
| Вероятность | Средняя |
| Меры по предотвращению | Детальное изучение выбираемых интегрируемых средств, поиск альтернатив для каждого средства |
| Меры по минимизации | Замена проблемного средства альтернативным, изменение функциональности программного продукта |

Таблица 1.7 – Риск возникновения проблем с обучением пользователей

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Проблемы с обучением пользователей |
| Возможные последствия | Неудовлетворенность пользователей предоставляемым продуктом, увеличение времени на внедрение |
| Вероятность | Средняя |
| Меры по предотвращению | Создание как можно более интуитивно понятного пользователям интерфейса |
| Меры по минимизации | Проведение мероприятий по обучению пользователей |

Таблица 1.8 – Риск изменения требований к проекту

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Изменение требований к проекту |
| Возможные последствия | Увеличение времени разработки для реализации новых требований |
| Вероятность | Средняя |
| Меры по предотвращению | Описание четких границ проекта в концепции, согласование с заказчиком на протяжении всего времени разработки |
| Меры по минимизации | Реализация новых требований в последующих версиях |

Таблица 1.9 – Риск возникновения ошибок, сделанных на ранних этапах проектирования

|  |  |
| --- | --- |
| Свойство | Значение |
| Фактор риска | Выявление ошибок, сделанных на ранних этапах проектирования |
| Возможные последствия | Увеличение времени разработки для исправления ошибок, возможность не реализации законченного проекта в срок |
| Вероятность | Средняя |
| Меры по предотвращению | Тщательная проработка концепции и архитектуры проекта |
| Меры по минимизации | Использование гибкой архитектуры в проекте |

**Заинтересованные лица.** Ниже описаны лица, заинтересованные в разрабатываемой системе: заказчик проекта (таблица 1.10) и студенты (таблица 1.11).

Таблица 1.10 – Описание заинтересованного лица: заказчик

|  |  |
| --- | --- |
| Свойство | Значение |
| Представитель | Павел Леонидович Цытович |
| Описание | Преподаватель курса «Технология программирования» |
| Тип | Специалист в области управления проектами |
| Ответственность | Постановка требований, описание требуемой функциональности |
| Критерий успеха | Снижение затрат времени и трудоемкости по проверке и оцениванию семестровых работ студентов и ответов на рейтинговые задания |
| Вовлеченность | Рецензирование требований, рецензирование архитектурных и технических решений, тестирование программного продукта, администрирование системы |
| Поставляемые  артефакты (документы) | Концепция проекта, сценарии использования, логический дизайн, физический дизайн, спецификация тестирования, план пилотного внедрения |
| Комментарии/  Проблемы | Возможно изменение поставленных требований  во время разработки проекта |

Таблица 1.11 – Описание заинтересованного лица: студенты

|  |  |
| --- | --- |
| Свойство | Значение |
| Представители | Студенты |
| Описание | Студенты кафедры ЭВМ, изучающие дисциплины цикла «Программирование» |
| Тип | Обучающиеся |

Продолжение таблицы 1.11

|  |  |
| --- | --- |
| Свойство | Значение |
| Ответственность | Загрузка своих семестровых работ и решений к рейтинговым задачам на сайт item74.ru |
| Критерий успеха | Снижение времени ожидания проверки своей работы или задания, получение детальной информации о недостатках своего решения |
| Вовлеченность | Тестирование программного продукта |
| Поставляемые  артефакты | Руководство пользователя |
| Комментарии/  Проблемы | Необходимо обучение терминологии и принципам модульного тестирования |

**Профили пользователей.** Профили пользователей создаваемой системы выделены на основе ролей пользователей, каждая из которых связана с определенным набором прав, которые пользователь имеет в системе. Пользователь может иметь любое количество ролей из представленных в системе. Можно выделить следующие роли пользователей:

* администратор системы (таблица 1.12);
* студент (таблица 1.13).

Таблица 1.12 – Профиль пользователя «Администратор»

|  |  |
| --- | --- |
| Свойство | Значение |
| Представитель | Администратор |
| Описание | Пользователь, имеющий полный контроль над системой |
| Тип | Профессионал |
| Ответственность | Администрирование, обслуживание, поддержание работоспособности системы |

Продолжение таблицы 1.12

|  |  |
| --- | --- |
| Свойство | Значение |
| Критерий успеха | Уменьшение времени на обслуживание системы, безотказная работа системы. |
| Вовлеченность | Управление системой, защита от вредоносных действий, написание модульных тестов, составление семестровых и рейтинговых заданий |
| Поставляемые артефакты | Техническая документация, руководство по настройке и администрированию системы |
| Комментарии/  Проблемы | Пользователь должен иметь полное представление  о том, как работает система |

Таблица 1.13 – Профиль пользователя «Студент»

|  |  |
| --- | --- |
| Свойство | Значение |
| Представитель | Студент |
| Описание | Студент, зарегистрированный на портале item74.ru и подписанный на один из курсов цикла «Программирование» |
| Тип | Обучающийся |
| Ответственность | Закачивает проект выполненного семестрового или контрольного задания |
| Критерий успеха | Уменьшение времени необходимого для получения оценки, получение информации о допущенных ошибках |
| Вовлеченность | Использование штатных средств, предоставляемых Moodle 2.0 для закачивания архивов с файлами исходных кодов или самих файлов исходных кодов |
| Поставляемые  артефакты | Руководство пользователя |

Продолжение таблицы 1.13

|  |  |
| --- | --- |
| Свойство | Значение |
| Комментарии/  Проблемы | Пользователь должен иметь навыки работы с системой поддержки учебного процесса на базе Moodle 2.0 |

**Сценарии использования.** Пункт сценарии использования в концепции проекта отражает видение пользователем системы, ее основной функциональности. Сценарий использования определяет последовательность действий пользователя для получения некоторого результата, связанного с решением задач в рамках своей предметной области.

**Сценарий «Авторизация пользователя».** Описание сценария авторизации пользователя в системе приведено ниже в таблице 1.14.

Таблица 1.14 – Сценарий авторизации пользователя

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Авторизация в системе | Проверяет логин и пароль пользователя на соответствие указанным в файле конфигурации. Заносит информацию об авторизации пользователя в файл лога. |

**Сценарий «Просмотр списка проектов».** Описание сценария просмотра списка проектов приведено ниже в таблице 1.15.

Таблица 1.15 – Сценарий просмотра списка проектов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе, для каждого из которых указывается краткая информация |

**Сценарий «Создание проекта».** Описание сценария создания нового проекта приведено ниже в таблице 1.16.

Таблица 1.16 – Сценарий создания нового проекта

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе |
| Нажатие на кнопку «Добавить проект» | Открывает форму для ввода информации |
| Ввод информации о проекте | Проверяет корректность вводимых данных |
| Подтверждение введенной информации | Внесение информации в базу данных, регистрация проекта в системе, создание конвейера, обслуживающего проект |

**Сценарий «Просмотр информации о проекте».** Описание сценария просмотра информации о проекте приведено ниже в таблице 1.17.

Таблица 1.17 – Сценарий просмотра информации о проекте

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе |
| Нажатие на название проекта в списке | Открывает форму для просмотра информации  о проекте |

**Сценарий «Изменение проекта».** Описание сценария изменения проекта приведено ниже в таблице 1.18.

Таблица 1.18 – Сценарий изменения проекта

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе |
| Нажатие на кнопку «Редактировать» под названием проекта в списке | Открывает форму для изменения информации  о проекте |

Продолжение таблицы 1.18

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Ввод новой информации об проекте | Проверяет корректность вводимых данных |
| Подтверждение введенной информации | Внесение информации в базу данных |

**Сценарий «Удаление проекта».** Описание сценария удаления проекта приведено ниже в таблице 1.19.

Таблица 1.19 – Сценарий удаления проекта

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе |
| Нажатие на кнопку «Удалить» под названием проекта в списке | Запрашивает подтверждение удаления проекта |
| Подтверждение удаления проекта | Удаление информации из базы данных |

**Сценарий «Просмотр списка источников исходных кодов».** Описание сценария просмотра списка источников исходных кодов приведено ниже в таблице 1.20.

Таблица 1.20 – Сценарий просмотра списка источников исходных кодов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Источники исходных кодов» | Отображает список источников исходных кодов, зарегистрированных в системе, для каждого из которых указывается краткая информация |

**Сценарий «Создание источника исходных кодов».** Описание сценария создания нового источника исходных кодов приведено ниже в таблице 1.21.

Таблица 1.21 – Сценарий создания нового источника исходных кодов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Источники исходных кодов» | Отображает список источников исходных кодов, зарегистрированных в системе, для каждого из которых указывается краткая информация |
| Нажатие на кнопку «Добавить источник исходных кодов» | Открывает форму для ввода информации |
| Ввод информации об источнике исходных кодов | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных, регистрация источника исходных кодов в системе |

**Сценарий «Просмотр информации об источнике исходных кодов».** Описание сценария просмотра информации об источнике исходных кодов приведено ниже в таблице 1.22.

Таблица 1.22 – Сценарий просмотра информации об источнике исходных кодов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Источники исходных кодов» | Отображает список источников исходных кодов, зарегистрированных в системе, для каждого из которых указывается краткая информация |
| Нажатие на название источника исходных кодов в списке | Открывает форму для просмотра информации  об источнике исходных кодов |

**Сценарий «Изменение источника исходных кодов».** Описание сценария изменения источника исходных кодов приведено ниже в таблице 1.23.

Таблица 1.23 – Сценарий изменения источника исходных кодов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Источники исходных кодов» | Отображает список источников исходных кодов, зарегистрированных в системе, для каждого из которых указывается краткая информация |
| Нажатие на кнопку «Редактировать» под названием источника исходных кодов в списке | Открывает форму для изменения информации  об источнике исходных кодов |
| Ввод новой информации об источнике исходных кодов | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных |

**Сценарий «Удаление источника исходных кодов».** Описание сценария удаления источника исходных кодов приведено ниже в таблице 1.24.

Таблица 1.24 – Сценарий удаления источника исходных кодов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Источники исходных кодов» | Отображает список источников исходных кодов, зарегистрированных в системе, для каждого из которых указывается краткая информация |

Продолжение таблицы 1.24

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Нажатие на кнопку «Удалить» под названием источника исходных кодов в списке | Запрашивает подтверждение удаления источника исходных кодов |
| Подтверждение удаления источника исходных кодов | Удаление информации из базы данных |

**Сценарий «Просмотр списка конфигураций отчетов».** Описание сценария просмотра списка конфигураций отчетов приведено ниже в таблице 1.25.

Таблица 1.25 – Сценарий просмотра списка конфигураций отчетов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации отчетов» | Отображает список конфигураций отчетов, зарегистрированных в системе, для каждой из которых указывается краткая информация |

**Сценарий «Создание конфигурации отчетов».** Описание сценария создания новой конфигурации отчетов приведено ниже в таблице 1.26.

Таблица 1.26 – Сценарий создания новой конфигурации отчетов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации отчетов» | Отображает список конфигураций отчетов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Добавить конфигурацию отчетов» | Открывает форму для ввода информации |
| Ввод информации о конфигурации отчетов | Проверяет корректность вводимых данных |

Продолжение таблицы 1.26

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных, регистрация конфигурации отчетов в системе |

**Сценарий «Просмотр информации о конфигурации отчетов».** Описание сценария просмотра информации о конфигурации отчетов приведено ниже в таблице 1.27.

Таблица 1.27 – Сценарий просмотра информации о конфигурации отчетов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации отчетов» | Отображает список конфигураций отчетов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на название конфигурации отчетов в списке | Открывает форму для просмотра информации  о конфигурации отчетов |

**Сценарий «Изменение конфигурации отчетов».** Описание сценария изменения конфигурации отчетов приведено ниже в таблице 1.28.

Таблица 1.28 – Сценарий изменения конфигурации отчетов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации отчетов» | Отображает список конфигураций отчетов, зарегистрированных в системе, для каждой из которых указывается краткая информация |

Продолжение таблицы 1.28

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Нажатие на кнопку «Редактировать» под названием конфигурации отчетов в списке | Открывает форму для изменения информации  о конфигурации отчетов |
| Ввод новой информации о конфигурации отчетов | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных |

**Сценарий «Удаление конфигурации отчетов».** Описание сценария удаления конфигурации отчетов приведено ниже в таблице 1.29.

Таблица 1.29 – Сценарий удаления конфигурации отчетов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации отчетов» | Отображает список конфигураций отчетов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Удалить» под названием конфигурации отчетов в списке | Запрашивает подтверждение удаления конфигурации отчетов |
| Подтверждение удаления конфигурации отчетов | Удаление информации из базы данных |

**Сценарий «Просмотр списка конфигураций сборки».** Описание сценария просмотра списка конфигураций сборки приведено ниже в таблице 1.30.

Таблица 1.30 – Сценарий просмотра списка конфигураций сборки

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации сборки» | Отображает список конфигураций сборки, зарегистрированных в системе, для каждой из которых указывается краткая информация |

**Сценарий «Создание конфигурации сборки».** Описание сценария создания новой конфигурации сборки приведено ниже в таблице 1.31.

Таблица 1.31 – Сценарий создания новой конфигурации сборки

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации сборки» | Отображает список конфигураций сборки, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Добавить конфигурацию сборки» | Открывает форму для ввода информации |
| Ввод информации о конфигурации сборки | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных, регистрация конфигурации сборки в системе |

**Сценарий «Просмотр информации о конфигурации сборки».** Описание сценария просмотра информации о конфигурации сборки приведено ниже в таблице 1.32.

Таблица 1.32 – Сценарий просмотра информации о конфигурации сборки

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации сборки» | Отображает список конфигураций сборки, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на название конфигурации сборки в списке | Открывает форму для просмотра информации  о конфигурации сборки |

**Сценарий «Изменение конфигурации сборки».** Описание сценария изменения конфигурации сборки приведено ниже в таблице 1.33.

Таблица 1.33 – Сценарий изменения конфигурации сборки

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации сборки» | Отображает список конфигураций сборки, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Редактировать» под названием конфигурации сборки в списке | Открывает форму для изменения информации  о конфигурации сборки |
| Ввод новой информации о конфигурации сборки | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных |

**Сценарий «Удаление конфигурации сборки».** Описание сценария удаления конфигурации сборки приведено ниже в таблице 1.34.

Таблица 1.34 – Сценарий удаления конфигурации сборки

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации сборки» | Отображает список конфигураций сборки, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Удалить» под названием конфигурации сборки в списке | Запрашивает подтверждение удаления конфигурации сборки |
| Подтверждение удаления конфигурации сборки | Удаление информации из базы данных |

**Сценарий «Просмотр списка конфигураций запуска модульных тестов».** Описание сценария просмотра списка конфигураций запуска модульных тестов приведено ниже в таблице 1.35.

Таблица 1.35 – Сценарий просмотра списка конфигураций запуска модульных тестов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации запуска модульных тестов» | Отображает список конфигураций запуска модульных тестов, зарегистрированных в системе, для каждой из которых указывается краткая информация |

**Сценарий «Создание конфигурации запуска модульных тестов».** Описание сценария создания новой конфигурации запуска модульных тестов приведено ниже в таблице 1.36.

Таблица 1.36 – Сценарий создания новой конфигурации запуска модульных тестов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации запуска модульных тестов» | Отображает список конфигураций запуска модульных тестов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Добавить конфигурацию запуска модульных тестов» | Открывает форму для ввода информации |
| Ввод информации о конфигурации запуска модульных тестов | Проверяет корректность вводимых данных |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных, регистрация конфигурации запуска модульных тестов в системе |

**Сценарий «Просмотр информации о конфигурации запуска модульных тестов».** Описание сценария просмотра информации о конфигурации запуска модульных тестов приведено ниже в таблице 1.37.

Таблица 1.37 – Сценарий просмотра информации о конфигурации запуска модульных тестов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации запуска модульных тестов» | Отображает список конфигураций запуска модульных тестов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на название конфигурации запуска модульных тестов в списке | Открывает форму для просмотра информации  о конфигурации запуска модульных тестов |

**Сценарий «Изменение конфигурации запуска модульных тестов».** Описание сценария изменения конфигурации запуска модульных тестов приведено ниже в таблице 1.38.

Таблица 1.38 – Сценарий изменения конфигурации запуска модульных тестов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации запуска модульных тестов» | Отображает список конфигураций запуска модульных тестов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Редактировать» под названием конфигурации запуска модульных тестов в списке | Открывает форму для изменения информации  о конфигурации запуска модульных тестов |
| Ввод новой информации о конфигурации запуска модульных тестов | Проверяет корректность вводимых данных |

Продолжение таблицы 1.38

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбора типа компонента поставщика | Отображение списка возможных параметров для выбранного типа компонента поставщика с описанием и значением по умолчанию для каждого параметра |
| Подтверждение введенной информации | Внесение информации в базу данных |

**Сценарий «Удаление конфигурации запуска модульных тестов».** Описание сценария удаления конфигурации запуска модульных тестов приведено ниже в таблице 1.39.

Таблица 1.39 – Сценарий удаления конфигурации запуска модульных тестов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Конфигурации запуска модульных тестов» | Отображает список конфигураций запуска модульных тестов, зарегистрированных в системе, для каждой из которых указывается краткая информация |
| Нажатие на кнопку «Удалить» под названием конфигурации запуска модульных тестов в списке | Запрашивает подтверждение удаления конфигурации запуска модульных тестов |
| Подтверждение удаления конфигурации запуска модульных тестов | Удаление информации из базы данных |

**Сценарий «Скачивание файла отчета».** Описание сценария скачивания файла отчета приведено ниже в таблице 1.40.

Таблица 1.40 – Сценарий скачивания файла отчета

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Проекты» | Отображает список проектов, зарегистрированных в системе |
| Нажатие на название проекта в списке | Открывает форму для просмотра информации  о проекте |
| Нажатие на название файла отчета в списке «Отчеты» | Отправляет файл отчета для скачивания |

**Сценарий «Просмотр реестра компонентов».** Описание сценария просмотра реестра компонентов приведено ниже в таблице 1.41.

Таблица 1.41 – Сценарий просмотра реестра компонентов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Компоненты» | Отображает список компонентов, зарегистрированных в системе, сгруппированных по типу компонента, с описание и возможными параметрами для каждого компонента |

**Сценарий «Повторная сборка реестра компонентов».** Описание сценария повторной сборки реестра компонентов приведено ниже в таблице 1.42.

Таблица 1.42 – Сценарий повторной сборки реестра компонентов

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Выбор раздела «Компоненты» | Отображает список компонентов, зарегистрированных в системе |
| Нажатие кнопки «Пересобрать реестр компонентов» | Очистка реестра компонентов и регистрация доступных компонентов, отображение список компонентов, зарегистрированных в системе |

**Сценарий «Выход из системы».** Описание сценария выхода из системы приведено ниже в таблице 1.43.

Таблица 1.43 – Сценарий выхода из системы

|  |  |
| --- | --- |
| Действие пользователя | Реакция системы |
| Нажатие кнопки «Выход из системы» на панели управления | Завершает сеанс работы текущего пользователя с системой, отображает форму авторизации |

**Описание требований. Бизнес-требования.** Бизнес-требования описывают ожидания пользователей и заинтересованных лиц от программного продукта.

Система должна дать преподавателям возможность осуществлять проверку семестровых и контрольных работ, а также решений студентов к рейтинговым задачам в автоматическом режиме получая исходный код из файлов, которые учащиеся закачивают на сайт item74.ru, основанный на системе управления учебными курсами Moodle 2.0. Также система должно иметь возможность дальнейшего расширения для работы не только с Moodle 2.0, но и с другими возможными источниками исходного кода.

Преподаватель должен иметь возможность указывать файлы модульных тестов, содержащих тестовые сценарии, согласно которым осуществляется проверка студенческих работ. Помимо этого, преподавателю необходимо иметь возможность указывать параметры и способ сборки проектов студентов, иметь возможность изменять систему для работы с различными библиотеками модульного тестирования и языками программирования. Минимальным требованием является возможность сборки проектов, написанных на языке C++ компилятором Microsoft VisualC++ и поддержка библиотеки модульного тестирования UnitTest++.

Система должна обеспечивать гибкую настройку компонентов и основного рабочего процесса, состоящего из получения исходного кода, его сборки, тестирования и составления отчета.

Преподаватели должны иметь возможность легко обучить студентов пользоваться системой.

**Требования к операционной среде (системе).** Требования к операционной среде (системе) определяют дополнительные системные требования, которым должно соответствовать проектируемое решение.

Системные требования:

* полная поддержка протокола TCP/IP;
* операционная система семейства Windows NT (5.1 или выше);
* Ruby 1.9.2.

Конфигурация основного сервера:

* процессор – x86-совместимый (семейства Intel Pentium 4 и AMD Athlon XP или выше);
* оперативная память – не менее 2 Гбайт;
* дисковая подсистема – не менее 60 Гбайт.

Производительность сетевых интерфейсов между основным сервером и сервером, на котором размещен item74.ru, не менее 1 Мбит/с.

**Требования к производительности.** Система должна поддерживать до 50 проектов. Отклик системы в пределах 2–10 секунд. Доступность – 99%.

**Требования к среде эксплуатации.** Для работы с системой необходим веб-браузер, поддерживающий технологии HTML5, JavaScript.

**Требования к документации.** Документация к системе должна содержать руководство оператора, выполненное по ГОСТ 19.505-79 [18].

### Границы проекта

Данный пункт описывает границы проектного решения, которые определяются диапазоном возможностей и функций программного продукта, путем анализа того, что лежит вне решений проекта; выработкой стратегии выпуска версий программного продукта и критериев приемки.

Описание функций программного продукта. Функции программного продукта:

1. Авторизация в системе.
2. Автоматизация процессов проверки семестровых работ и решений рейтинговых задач, ранжирования работ, информирования студентов о допущенных ошибках.
3. Предоставление инструментов для гибкой настройки и администрирования системы.
4. Уведомление преподавателей о наступлении важных событий, таких как прохождение модульных тестов, создание отчетов, невозможность выполнить получение исходного кода и др.
5. Возможность получения исходного кода тестируемых работ из системы управления учебными курсами Moodle 2.0 или из директории в файловой системе.
6. Хранение и предоставление доступа к отчетам о результате сборки и пройденных тестах для каждой студенческой работы.
7. Возможность изменения функциональности программного продукта за счет написания собственных компонентов, которые позволяют изменять тип источника исходных кодов для тестирования, способ сборки из исходных кодов, поддерживаемую библиотеку модульного тестирования и формат выходного отчета.

**Описание границ проектного решения.** Клиентское программное обеспечение будет реализовано по технологии HTML5 и AJAX и будет исполняться в браузере.

Серверная часть будет представлять собой приложение на языке Ruby, предоставляющее администратору веб-интерфейс для управления и конфигурирования.

Для целей взаимодействия с системой управления учебными курсами Moodle 2.0 будет создан веб-сервис, выполненный на языке PHP, являющийся модулем к данной системе управления курсами и регистрирующийся в ней через установочный скрипт.

Система содержит средства для расширения функциональности пользователем. Поставляется минимально необходимый набор функций (см. выше), остальные задачи решаются сами пользователем или разработчиком, по взаимной договоренности.

**Стратегия выпуска версий программного продукта.** В ходе разработки выпускается несколько тестовых версий, обладающих частичной функциональностью и призванных смоделировать один или несколько прецедентов работы системы.

После первого этапа разработки выпускается версия 0.9, которая обладает необходимой функциональностью. Эта версия проверяется на соответствие критериям приемки. После необходимых доработок, выпускается версия 1.0, обладающая всей необходимой функциональностью и отвечающая всем задокументированным требованиям заказчика.

**Критерии приемки.** По окончании разработки продукта проводятся тестовые испытания согласно документу «Спецификация тестирования», в ходе которых проверяются все требования, описанные в пунктах «Описание требований» и «Сценарии использования». Если все требования выполнены, система считается сданной. Если какое-либо требование не выполнено, разработчик дорабатывает программный продукт [].

### Стратегия архитектуры и технических решений

Описание возможных архитектурных решений. Проектируемая система будет построена на основе микроядерной архитектуры [14], и будет реализовывать смешанную модель программной системы – как наиболее подходящую для гибких программных систем [].

Возможная архитектура разрабатываемой Системы модульного тестирования студенческих работ представлена ниже (рисунок 1.1).

Архитектура проектируемой системы

Рисунок 1.1 – Архитектура проектируемой системы

Отдельные структурные модули разрабатываемой системы и технические решения можно разделить на несколько типов в зависимости от их назначения (рисунок 1.2).

Типы модулей и решений для проектируемой системы

Рисунок 1.2 – Типы модулей и решений для проектируемой системы

Взаимодействие модулей системы будет осуществляться на основе событийно-ориентированного подхода. Так же пользователям будет предоставлена возможность изменять поведение системы с помощью конфигурационных файлов и параметризации вызовов к компонентам. Описание выбранных возможных технических решений для реализации структурных модулей системы приведено ниже.

Описание возможных технических решений. Данный пункт описывает технологии, которые будет применяться для реализации архитектурных решений, показанных выше. Здесь приводится высокоуровневое описание выбранной технологии, которое описывает ключевые элементы используемой технологии.

Технические средства серверной части системы. В серверной части разрабатываемой системы планируется использовать представленные ниже технические решения (рисунок 1.3).

Компоненты серверной части системы

Рисунок 1.3 – Технические средства серверной части системы

WEBrick (www.webrick.org) – библиотека, написанная на языке про-граммирования Ruby и предоставляющая набор базовых HTTP-сервисов, включающих в себя простой веб-сервер, позволяющий размещать   
Ruby-приложения [5]. WEBrick является кроссплатформенным программным обеспечением с открытым исходным кодом и распространяется по лицензии Ruby.

Функции HTTP-сервера WEBrick:

1. Обслуживание статических запросов, индексных файлов, автоматическое создание списка файлов.
2. Модульность, фильтры, в том числе сжатие (gzip), HTTP-аутентификация.
3. Совместимость с интерфейсом Rack, для размещения веб-приложений, написанных на языке Ruby.

В разрабатываемой системе WEBrick используется в качестве веб-сервера обслуживающего запросы к статическим файлам и размещающего веб-приложение для администрирования, написанное с использование фреймворка Sinatra.

Sinatra (www.sinatrarb.com) – бесплатный и открытый программный каркас написанный на языке Ruby, предоставляющий возможность быстрого создания веб-приложений, с использование предметно-ориентированного   
языка (DSL) [6]. Sinatra является кроссплатформенным программным обеспечением с открытым исходным кодов и распространяется под лицензией MIT.

Sinatra не следует типичному паттерну MVC (Model-View-Controller) и предоставляет разработчикам свободу выбора способов организации модели и языка разметки для описания веб-страниц.

Фреймворк Sinatra представлен монолитным gem'ом, состоящим из одного основного файла с кодом и двух вспомогательных, и включает в себя механизмы помощников (helpers) для написания расширений к базовой функциональности, а также фильтров, позволяющих изменить стандартное поведение обработчиков.

Как и большинство программных каркасов для написания веб-приложений на языке Ruby Sinatra использует сервисы, предоставляемые нижележащим веб-сервером (в нашем случае – WEBrick) через единый интерфейс Rack.

Sinatra позволяет создавать приложения используя два различных подхода: инкапсуляцию обработчиком маршрутов и помощников в отдельные классы-модули, унаследованные от базового класса Application, или с помощью определения обработчиком маршрутов в глобальном пространстве имен. В разрабатываемой системе будет использоваться именно второй подход, поскольку, вследствие небольшого размера веб-приложения, а также заключения специфичного кода ядра в отдельные модули (пространства имен и примеси языка Ruby), будет исключена возможность возникновения конфликтов.

DataMapper (datamapper.org) – библиотека для организации объектно-реляционного отображения, написанная на языке программирования Ruby, и разработанная с целью устранения недостатков присутствующих в библиотеке ActiveRecord. DataMapper является кроссплатформенным программным обеспечением с открытым исходным кодов и распространяется под лицензией MIT.

DataMapper не ориентирован на работу только с SQL-совместимыми источниками данным и может взаимодействовать, при наличии необходимого драйвера, и с NoSQL базами данных, такими как CouchDB, Apache Solr и другими.,

Некоторые основные возможности DataMapper:

1. Работу с несколькими репозиториями (источниками данных).
2. Гибкое управление привязкой доменных сущностей к физической структуре и ассоциациями (один-к-одному, один-ко-многим, многие-ко-многим).
3. Ленивая загрузка специально помеченных полей.
4. Каскадные запросы и отложенное выполнение запросов до того момента, когда данные, получаемые в запроса не становятся необходимы.
5. Ранняя загрузка дочерних объектов во избежание проблемы N+1 запросов.
6. Позволяет хранить в базе данных поля сложных типов, такие как массивы или хеши Ruby (ассоциативные массивы, сортированные по ключу), используя их представление в виде JSON или YAML.

В разрабатываемой системе DataMapper используется для получения, сохранения и представления объектов предметной области в связке с драйвером для баз данных SQLite.

YAML (www.yaml.org) – человекочитаемый формат сериализации данных, концептуально близкий к языкам разметки, но ориентированный на удобство ввода-вывода типичных структур данных многих языков программирования [7]. YAML обладает минимальным синтаксисом, особенно по сравнению с XML.

YAML создан для реализации следующих требований:

* краткость и понятность;
* выразительность и расширяемость;
* использование структур данных, родных для языков программирования;
* использование цельной модели данных (нет исключений – нет беспорядка).

В разрабатываемой системе YAML используется как формат для файла конфигурации, а также как формат для представления полей сущностей, хранящих сложные структуры данных, хранимых в БД, и которые не участвуют в критериях для выборки.

Технические средства пользовательского интерфейса системы. Технические средства, используемые для организации пользовательского интерфейса, описаны ниже (рисунок 1.4).

Технические средства пользовательского интерфейса системы

Рисунок 1.4 – Технические средства пользовательского интерфейса системы

AJAX (www.javascript.ru/ajax) – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером [8]. Базируется на двух основных принципах: использовании технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, и DHTML для динамического изменения содержания страницы.

AJAX позволяет перенести задачи связанные с работой пользовательского интерфейса с сервера на клиентские машины.

jQuery (jquery.com) – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML [9]. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с AJAX. jQuery позволяет отделить поведение от структуры HTML, согласуясь с принципом ненавязчивого JavaScript. jQuery является кроссплатформенным программным обеспечением с открытым исходным кодов и распространяется под двойной лицензией MIT/GNU GPL.

Возможности библиотеки:

* движок кроссбраузерных CSS-селекторов Sizzle;
* переход по дереву DOM;
* работа с событиями DOM-дерева;
* визуальные эффекты;
* работа с AJAX;
* расширяемость за счет плагинов.

В разрабатываемой системе jQuery будет использоваться для организации ненавязчивого JavaScript путем определения поведения интерактивных элементов страницы через атрибуты HTML-тегов, которые указывают на необходимый набор обработчиков DOM-событий. Также jQuery будет использоваться для взаимодействия с сервером по технологии AJAX.

jQueryUI (jqueryui.com) – библиотека JavaScript с открытым исходным кодом для создания насыщенного пользовательского интерфейса в веб-приложениях, часть проекта jQuery [10]. Построена поверх главной библиотеки jQuery и предоставляет разработчику упрощенный доступ к её функциям взаимодействия, анимации и эффектов, а также набор виджетов. jQuery UI является кроссплатформенным программным обеспечением с открытым исходным кодов и распространяется под двойной лицензией MIT/GNU GPL.

В разрабатываемой системе jQuery UI будет использоваться для организации некоторых элементов веб-интерфейса администратора, в частности всплывающих модальных диалогов (popup), полей ввода с автодополнениям и полей для ввода даты с выбором из календаря.

HAML (haml-lang.org) – язык разметки для упрощенной генерации XHTML. HAML нацелен на минимальность синтаксиса и четкое представление структуры результирующего XHTML-документа [11]. HAML является кроссплатформенным программным обеспечением с открытым исходным кодов и распространяется под лицензией MIT.

В разрабатываемой системе HAML используется для описания разметки файлов шаблонов страниц веб-интерфейса администратора.

СУБД, применяемые для хранения данных в системе. СУБД, применяемые для хранения данных в системе, описаны ниже (рисунок 1.5).

Организация хранения данных

Рисунок 1.5 – Технические средства пользовательского интерфейса системы

SQLite (www.sqlite.org) – легковесная файл-серверная реляционная база данных [12].

Является кроссплатформенным программным обеспечением с открытым исходным кодом (передан в общественное достояние).

Поскольку SQLite не использует парадигму клиент-сервер, то движок SQLite не является работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа.

Преимущества:

1. Удобство – отсутствие проблем в разворачивании и настройке. Возможность переносимости данных при разворачивании одного и того же экземпляра системы на разным компьютерах.
2. Скорость – в качестве протокола обмена используются вызовы функций библиотеки SQLite.

В разрабатываемой системе SQLite используется для хранения системных данных – сущностей предметной области.

PStore (www.ruby-doc.org/stdlib/libdoc/pstore/rdoc/classes/PStore.html) – высокопроизводительная база данных, являющаяся частью стандартной библиотеки языка Ruby версии 1.9.2. PStore является нереляционной базой данных – она хранит пары ключ/значение как массивы байтов, используя хеш-таблицы, соблюдая правило единственности значения для одного ключа. Являясь частью стандартной библиотеки Ruby, PStore является программным обеспечением с открытым исходным кодов и распространяется под лицензией Ruby.

Преимущества:

1. Безопасность – изоляция от системных баз данных.
2. Удобство – возможность быстрого резервного копирования, смены или обнуления БД, возможность хранить любые произвольные объекты языка Ruby.
3. Скорость – все данные однотипны, и как следствие, высокая скорость чтения данных.

В разрабатываемой системе PStore используется для хранение произвольных рабочих данных компонентов, для расширения их возможностей.

Данные компонентов обслуживаются отдельной СУБД для снижения нагрузки на БД, предназначенную для хранения системных данных и организации «песочницы», в которой работают компоненты.

## Сценарии использования

Сценарии использования описывают множество действий (активностей), которые будут реализовывать проектное решение и поддерживать его. Эти действия описываются в контексте того, что ожидает пользователь от предложенного решения, и каким системным и прикладным требованиям они будут удовлетворять. Сценарии использования служат для документирования функциональных требований к программным системам.

### Словарь предметной области

Составление словаря предметной области позволит говорить с заказчиком на одном языке, оперируя его терминами и понятиями. Кроме того, словарь предметной области служит хорошей основой для идентификации акторов, прецедентов и объектов будущей системы.

В таблице 1.44 приводятся все ключевые термины и определения, которые используются в предметной области заказчика.

Таблица 1.44 – Словарь предметной области

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Программное обеспечение | Совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ. Также, это совокупность программ, процедур и правил, а также документации, относящихся  к функционированию системы обработки данных. |

Продолжение таблицы 1.44

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Разработка ПО | Процесс, направленный на создание и поддержание работоспособности, качества и надежности программного обеспечения, используя технологии, методологию и практики из информатики, управления проектами, математики, инженерии и других областей знания. |
| Проектирование ПО | Процесс создания проекта программного обеспечения (ПО), а также дисциплина, изучающая методы проектирования. Проектирование подразумевает выработку свойств системы на основе анализа постановки задачи, а именно: моделей предметной области, требований к ПО, а также опыта проектировщика.  Модель предметной области накладывает ограничения на бизнес-логику и структуры данных. |
| Проект | Совокупность настроек компонентов определенного типа для получения исходных кодов, сборки цели, запуска модульных тестов и создания файлов отчетов, а также интервала обновления ревизии проекта. Исходные коды для конкретного программного обеспечения являются целью проекта |
| Конвейер проекта | Процесс, состоящий из четырех шагов: получение исходных кодов, сборка из исходных кодов, запуск модульных тестов, создание файлов отчетов. |
| Цель (цель тестирования) | В узком смысле – студенческая семестровая работа или решение рейтингового задания; в более широком смысле – любое программное обеспечение, которое может быть протестирована в системе с определенной конфигурацией проекта |

Продолжение таблицы 1.44

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Компонент | Программный модуль, реализующий механизм изменяемости функциональности системы и представляющий собой директорию с файлами классов языка Ruby, отвечающих определенной конвенции имен, описанной в документации |
| Отчет | Файл, содержащий информацию о ревизии проекта, целях тестирования для данной ревизии, а также информацию о пройденных и проваленных тестах для каждой цели. |
| Пользователь | Лицо, которое использует действующую систему для выполнения конкретной функции. |
| Аутентификация | Подтверждение подлинности пользователя, которое состоит во вводе вашего пользовательского идентификатора – «логина» (англ. login – регистрационное имя пользователя) и пароля – некой конфиденциальной информации, знание которой обеспечивает владение определенным ресурсом. Получив введенный пользователем логин и пароль, компьютер сравнивает их со значением, которое хранится в специальном хранилище и, в случае совпадения, пропускает пользователя в систему. |
| Авторизация | Процесс предоставления пользователю прав на выполнение некоторых действий.  Процесс подтверждения (проверки) прав пользователей на выполнение некоторых действий. |

Продолжение таблицы 1.44

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Администратор | Сотрудник, должностные обязанности которого подразумевают обеспечение штатной работы парка компьютерной техники, сети и самой программной системы. Занимается настройкой и конфигурированием данной системы.  В разрабатываемой программной системе данная роль означает, что пользователь имеет все права (максимальный уровень доступа) для работы в системе. В том числе дается доступ к средству конфигурирования системы. |
| Заказчик | Лицо (физическое или юридическое), заинтересованное в выполнении исполнителем (командой разработчиков) какого-либо объема работ над программным проектом и оформляющее в связи с этим заказ. |
| Ошибка (баг, bug) | Ошибка в программе или системе, которая выдает неожиданный или неправильный результат. Большинство багов возникают из-за ошибок, сделанных разработчиками программы в её исходном коде, либо в её дизайне. «Баг» проявляет себя на стадии работы программы, в отличие, например, от ошибок проектирования или синтаксических ошибок. |
| Ревизия | Версия проекта, которая однозначно идентифицирует набор целей, которые были собраны и протестированы в рамках данного проекта |
| Конфигурационный файл | Файл описывающий параметры работы системы и компонентов в человекочитаемом формате |

Продолжение таблицы 1.44

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Сборка проекта и сборка по расписанию | Сборки по расписанию (англ. daily build – русск. ежедневная сборка), они, как правило, проводятся каждой ночью в автоматическом режиме – ночные сборки (чтобы к началу рабочего дня были готовы результаты тестирования). Для различения дополнительно вводится система нумерации сборок – обычно, каждая сборка нумеруется натуральным числом, которое увеличивается с каждой новой сборкой. Исходные тексты и другие исходные данные при взятии их из репозитория системы контроля версий помечаются номером сборки. Благодаря этому, точно такая же сборка может быть точно воспроизведена в будущем – достаточно взять исходные данные по нужной метке и запустить процесс снова. Это даёт возможность повторно выпускать даже очень старые версии программы с небольшими исправлениями. |
| Тестирование ПО | Процесс исследования программного обеспечения (ПО) с целью получения информации о качестве продукта. |
| Функциональное тестирование | Тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает. |

Продолжение таблицы 1.44

|  |  |
| --- | --- |
| Понятие | Описание понятия или расшифровка термина |
| Модульное тестирование или  юнит-тестирование (англ. unit testing) | Процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже написанных и оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок. Цель модульного тестирования – изолировать отдельные части программы и показать, что по отдельности эти части работоспособны. |

### Модель прецедентов

В данном разделе описаны сценарии использования, которые разрабатывались на основе информации, полученной от заказчика.

Акторы. Акторы представляют собой сущности (не путать с понятием сущности в контексте предметной области проекта), внешние по отношению   
к системе. Выявление акторов позволяет построить сценарии использования для будущего проектного решения. Акторы, то есть пользователи нашего проектного решения, которые инициируют выполнение сценариев, внешние объекты или существующие подсистемы, с которыми взаимодействует разрабатываемое проектное решение [], представлены ниже (таблица 1.45).

Таблица 1.45 – Акторы системы

|  |  |  |
| --- | --- | --- |
| Наименование актора | Тип актора | Описательная характеристика |
| Администратор | первичный актор, пользователь | Администратор имеет все права для работы в системе. В том числе ему дается доступ к средствам конфигурирования системы и ее веб-интерфейсу. |
| Системная СУБД | вторичный актор, внешняя система | Система управления базой данных, хранящей информацию о проектах, конфигурациях конвейера и параметрах компонентов. |
| СУБД компонентов | вторичный актор, внешняя система | Система управления базой данных, хранящей служебную рабочую информацию различных компонентов системы. |

Диаграмма прецедентов. Диаграмма прецедентов раскрывает суть взаимодействия акторов с разрабатываемой системой и служит основой для разработки сценариев использования будущего проектного решения. Она помогает выявить активности, которые необходимо будет реализовать в программном продукте []. Также, модель прецедентов можно построить и более подробно, показав взаимодействие пользователей и отдельных компонентов системы. В этом случае перечень активностей получается более точным и подробным. При проектировании построена модель прецедентов с точки зрения пользователей системы.

На рисунке 1.6 показана диаграмма прецедентов верхнего уровня   
для Системы сборки и тестирования программного обеспечения. На рисунках   
1.7-1.13 отражены более подробно отдельные прецеденты с диаграммы верхнего уровня.

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UCMain.emf

Рисунок 1.6 – Диаграмма прецедентов Системы сборки и тестирования программного обеспечения (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC2.emf

Рисунок 1.7 – Диаграмма прецедентов для прецедента UC2: «Управление проектами» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC3.emf

Рисунок 1.8 – Диаграмма прецедентов для прецедента UC3: «Управление источниками исходных кодов» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC4.emf

Рисунок 1.9 – Диаграмма прецедентов для прецедента UC4: «Управление конфигурациями сборки» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC5.emf

Рисунок 1.10 – Диаграмма прецедентов для прецедента UC5: «Управление конфигурациями запуска тестов» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC6.emf

Рисунок 1.11 – Диаграмма прецедентов для прецедента UC6: «Управление конфигурациями отчетов» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC7.emf

Рисунок 1.12 – Диаграмма прецедентов для прецедента UC7: «Работа с отчетами» (программа StarUML)

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\UC8.emf

Рисунок 1.13 – Диаграмма прецедентов для прецедента UC8: «Управление компонентами» (программа StarUML)

### Сценарии

Далее подробно описываются сценарии использования, которые лежат в основе прецедентной модели, которые были отражены в концепции проекта в более общем виде и графически показаны на диаграмме прецедентов выше. Такое подробное описание позволяет при разработке системы опираться на этот план []. Таким образом, сценарии использования в этом пункте отражают работу системы с точки зрения программистов.

**Авторизация пользователя.** Сценарий авторизации пользователя отражен ниже в таблице 1.46.

Таблица 1.46 – Сценарий авторизации пользователя

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Авторизация пользователя |
| Идентификатор прецедента | UC1 |
| Версия | 1 |
| Список акторов | Администратор |
| Предусловия | Пользователь перешел по базовому адресу системы. |
| Действия | Пользователь вводит в загруженной форме авторизации корректные логин и пароль и нажимает кнопку «Войти». Система идентифицирует пользователя – проверяет введенные логин и пароль администратора в конфигурационном файле. Заносит информацию об открытой сессии пользователя в файла системного журнала. |
| Постусловия | Пользователем введены верные логин и пароль. Служебная информация об открытой сессии успешно добавлена в файл журнала. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Просмотр списка проектов.** Сценарий просмотра списка проектов отражен ниже в таблице 1.47.

Таблица 1.47 – Сценарий просмотра списка проектов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка проектов |
| Идентификатор прецедента | UC2.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Проекты». |
| Действия | Система подгружает информацию о всех проектах из системной БД. В веб-интерфейсе отображается список с информацией о проектах. |
| Постусловия | Система успешно загрузила информацию о проектах из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Создание проекта.** Сценарий создания проекта отражен ниже   
в таблице 1.48.

Таблица 1.48 – Сценарий создания проекта

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Создание проекта |
| Идентификатор | UC2.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка проектов». Пользователь выбрал действие «Добавить проект» на инструментальной панели. |
| Действия | Пользователю открывается форма для создания нового проекта. Пользователь вводит в форме всю необходимую информацию о проекте: название, описание, интервал обновления, адрес электронной почты ответственного. Добавляет к проекту конфигурации шагов конвейера. Если вся информация введена корректно, то пользователь может нажать кнопку «Создать» и система занесет данные о новом проекте в системную БД, для проекта создается задача в расписании и запускается конвейер. |
| Постусловия | Введенная информация о новом проекте корректна и успешно занесена в системную БД, для проекта создана задача и запущен конвейер. |
| Включение | «Просмотр списка проектов» |
| Расширение | – |
| Обобщение | – |

**Просмотр информации о проекте.** Сценарий просмотра информации о проекте отражен ниже в таблице 1.49.

Таблица 1.49 – Сценарий просмотра информации о проекте

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр информации о проекте |
| Идентификатор прецедента | UC2.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка проектов». Пользователь совершил нажатие на названии проекта в списке. |
| Действия | В веб-интерфейсе открывается форма для отображения информации о проекте. В форме отображается вся информация о проекте: название, описание, интервал обновления, адрес электронной почты ответственного, список конфигураций шагов конвейера для данного проекта, а также список созданных отчетов. Для связанных с проектов конфигураций шагов конвейера также отображается служебная информация, взятая из СУБД компонентов. |
| Постусловия | – |
| Включение | «Просмотр списка проектов» |
| Расширение | – |
| Обобщение | – |

**Изменение проекта.** Сценарий изменения проекта отражен ниже в таблице 1.50.

Таблица 1.50 – Сценарий изменения проекта

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Изменение проекта |
| Идентификатор прецедента | UC2.4 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка проектов». В панели действий проекта в списке пользователь выбрал действие «Изменить». |
| Действия | В веб-интерфейсе открывается форма для редактирования информации о проекте. В форме отображается вся информация о проекте, которую можно изменять: название, описание, интервал обновления, адрес электронной почты ответственного. Можно также изменять привязку заранее созданных конфигураций шагов конвейера к редактируемому проекту. Если вся измененная новая информация введена корректно, то пользователь может нажать кнопку «Изменить», и система занесет измененные данные о проекте в системную БД. |
| Постусловия | Измененная информация о проекте корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка проектов» |
| Расширение | – |
| Обобщение | – |

**Удаление проекта.** Сценарий удаления проекта отражен ниже в таблице 1.51.

Таблица 1.51 – Сценарий удаления проекта

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Удаление проекта |
| Идентификатор прецедента | UC2.5 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка проектов». В панели действий проекта в списке пользователь выбрал действие «Удалить». |
| Действия | В веб-интерфейсе открывается диалоговое окно с запросом на подтверждение удаления выбранного проекта. Если пользователь подтверждает удаление, то система удаляет проект из системной БД. Задача конвейера для данного проекта удаляется из памяти. |
| Постусловия | Проект успешно удален из системной БД. Задача конвейера для данного проекта успешно удалена из памяти. |
| Включение | «Просмотр списка проектов» |
| Расширение | – |
| Обобщение | – |

**Просмотр списка источников исходных кодов.** Сценарий просмотра списка источников исходных кодов отражен ниже в таблице 1.52.

Таблица 1.52 – Сценарий просмотра списка источников исходных кодов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка источников исходных кодов |
| Идентификатор прецедента | UC3.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Источники исходных кодов». |
| Действия | Система подгружает информацию о всех источниках исходных кодов из системной БД. В веб-интерфейсе отображается список с информацией об источниках исходных кодов. |
| Постусловия | Система успешно загрузила информацию об источниках исходных кодов из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Создание источника исходных кодов.** Сценарий создания источника исходных кодов отражен ниже в таблице 1.53.

Таблица 1.53 – Сценарий создания источника исходных кодов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Создание источника исходных кодов |
| Идентификатор | UC3.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка источников исходных кодов». Пользователь выбрал действие «Добавить источник исходных кодов» на инструментальной панели. |
| Действия | Пользователю открывается форма для создания нового источника исходных кодов. Пользователь вводит в форме всю необходимую информацию об источнике исходных кодов: название, описание, тип компонента и значения параметров инициализации компонента. Если вся информация введена корректно, то пользователь может нажать кнопку «Создать» и система занесет данные о новом источнике исходных кодов в системную БД. |
| Постусловия | Введенная информация о новом источнике исходных кодов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка источников исходных кодов» |
| Расширение | – |
| Обобщение | – |

**Просмотр информации об источнике исходных кодов.** Сценарий просмотра информации об источнике исходных кодов отражен ниже в таблице 1.54.

Таблица 1.54 – Сценарий просмотра информации об источнике исходных кодов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр информации об источнике исходных кодов |
| Идентификатор прецедента | UC3.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка источников исходных кодов». Пользователь совершил нажатие на названии источника исходных кодов в списке. |
| Действия | В веб-интерфейсе открывается форма для отображения информации об источнике исходных кодов. В форме отображается вся информация об источнике исходных кодов: название, описание, тип компонента и значения параметров инициализации компонента. |
| Постусловия | – |
| Включение | «Просмотр списка источников исходных кодов» |
| Расширение | – |
| Обобщение | – |

**Изменение источника исходных кодов.** Сценарий изменения источника исходных кодов отражен ниже в таблице 1.55.

Таблица 1.55 – Сценарий изменения источника исходных кодов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Изменение источника исходных кодов |
| Идентификатор прецедента | UC3.4 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка источников исходных кодов». В панели действий для источника исходных кодов в списке пользователь выбрал действие «Изменить». |
| Действия | В веб-интерфейсе открывается форма для редактирования информации об источнике исходных кодов. В форме отображается вся информация об источнике исходных кодов, которую можно изменять название, описание, тип компонента и значения параметров инициализации компонента. Если вся измененная новая информация введена корректно, то пользователь может нажать кнопку «Изменить», и система занесет измененные данные об источнике исходных кодов в системную БД. |
| Постусловия | Измененная информация об источнике исходных кодов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка источников исходных кодов» |
| Расширение | – |
| Обобщение | – |

**Удаление источника исходных кодов.** Сценарий удаления источника исходных кодов отражен ниже в таблице 1.56.

Таблица 1.56 – Сценарий удаления источника исходных кодов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Удаление источника исходных кодов |
| Идентификатор прецедента | UC3.5 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка источников исходных кодов». В панели действий для источника исходных кодов в списке пользователь выбрал действие «Удалить». |
| Действия | В веб-интерфейсе открывается диалоговое окно с запросом на подтверждение удаления выбранного источника исходных кодов. Если пользователь подтверждает удаление, то система удаляет источник исходных кодов из системной БД. |
| Постусловия | Источник исходных кодов успешно удален из системной БД. |
| Включение | «Просмотр списка источников исходных кодов» |
| Расширение | – |
| Обобщение | – |

**Просмотр списка конфигураций сборки.** Сценарий просмотра списка конфигураций сборки отражен ниже в таблице 1.57.

Таблица 1.57 – Сценарий просмотра списка конфигураций сборки

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка конфигураций сборки |
| Идентификатор прецедента | UC4.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Конфигурации сборки». |
| Действия | Система подгружает информацию о всех конфигурациях сборки из системной БД. В веб-интерфейсе отображается список с информацией о конфигурациях сборки. |
| Постусловия | Система успешно загрузила информацию о конфигурациях сборки из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Создание конфигурации сборки.** Сценарий создания конфигурации сборки отражен ниже в таблице 1.58.

Таблица 1.58 – Сценарий создания конфигурации сборки

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Создание конфигурации сборки |
| Идентификатор | UC4.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций сборки». Пользователь выбрал действие «Добавить конфигурацию сборки» на инструментальной панели. |
| Действия | Пользователю открывается форма для создания новой конфигурации сборки. Пользователь вводит в форме всю необходимую информацию о конфигурации сборки: название, описание, тип компонента и значения параметров инициализации компонента. Если вся информация введена корректно, то пользователь может нажать кнопку «Создать» и система занесет данные о новой конфигурации сборки в системную БД. |
| Постусловия | Введенная информация о новой конфигурации сборки корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций сборки» |
| Расширение | – |
| Обобщение | – |

**Просмотр информации о конфигурации сборки.** Сценарий просмотра информации о конфигурации сборки отражен ниже в таблице 1.59.

Таблица 1.59 – Сценарий просмотра информации о конфигурации сборки

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр информации о конфигурации сборки |
| Идентификатор прецедента | UC4.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций сборки». Пользователь совершил нажатие на названии конфигурации сборки в списке. |
| Действия | В веб-интерфейсе открывается форма для отображения информации о конфигурации сборки. В форме отображается вся информация о конфигурации сборки: название, описание, тип компонента и значения параметров инициализации компонента. |
| Постусловия | – |
| Включение | «Просмотр списка конфигураций сборки» |
| Расширение | – |
| Обобщение | – |

**Изменение конфигурации сборки.** Сценарий изменения конфигурации сборки отражен ниже в таблице 1.60.

Таблица 1.60 – Сценарий изменения конфигурации сборки

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Изменение конфигурации сборки |
| Идентификатор прецедента | UC4.4 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций сборки». В панели действий для конфигурации сборки в списке пользователь выбрал действие «Изменить». |
| Действия | В веб-интерфейсе открывается форма для редактирования информации о конфигурации сборки. В форме отображается вся информация о конфигурации сборки, которую можно изменять название, описание, тип компонента и значения параметров инициализации компонента. Если вся измененная новая информация введена корректно, то пользователь может нажать кнопку «Изменить», и система занесет измененные данные о конфигурации сборки в системную БД. |
| Постусловия | Измененная информация о конфигурации сборки корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций сборки» |
| Расширение | – |
| Обобщение | – |

**Удаление конфигурации сборки.** Сценарий удаления конфигурации сборки отражен ниже в таблице 1.61.

Таблица 1.61 – Сценарий удаления конфигурации сборки

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Удаление конфигурации сборки |
| Идентификатор прецедента | UC4.5 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций сборки». В панели действий для конфигурации сборки в списке пользователь выбрал действие «Удалить». |
| Действия | В веб-интерфейсе открывается диалоговое окно с запросом на подтверждение удаления выбранной конфигурации сборки. Если пользователь подтверждает удаление, то система удаляет конфигурацию сборки из системной БД. |
| Постусловия | Конфигурация сборки успешно удалена из системной БД. |
| Включение | «Просмотр списка конфигураций сборки» |
| Расширение | – |
| Обобщение | – |

**Просмотр списка конфигураций запуска тестов.** Сценарий просмотра списка конфигураций запуска тестов отражен ниже в таблице 1.62.

Таблица 1.62 – Сценарий просмотра списка конфигураций запуска тестов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка конфигураций запуска тестов |
| Идентификатор прецедента | UC5.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Конфигурации запуска тестов». |
| Действия | Система подгружает информацию о всех конфигурациях запуска тестов из системной БД. В веб-интерфейсе отображается список с информацией о конфигурациях запуска тестов. |
| Постусловия | Система успешно загрузила информацию о конфигурациях запуска тестов из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Создание конфигурации запуска тестов.** Сценарий создания конфигурации запуска тестов отражен ниже в таблице 1.63.

Таблица 1.63 – Сценарий создания конфигурации запуска тестов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Создание конфигурации запуска тестов |
| Идентификатор | UC5.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций запуска тестов». Пользователь выбрал действие «Добавить конфигурацию запуска тестов» на инструментальной панели. |
| Действия | Пользователю открывается форма для создания новой конфигурации запуска тестов. Пользователь вводит в форме всю необходимую информацию о конфигурации запуска тестов: название, описание, тип компонента и значения параметров инициализации компонента. Если вся информация введена корректно, то пользователь может нажать кнопку «Создать» и система занесет данные о новой конфигурации запуска тестов в системную БД. |
| Постусловия | Введенная информация о новой конфигурации запуска тестов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций запуска тестов» |
| Расширение | – |
| Обобщение | – |

**Просмотр информации о конфигурации запуска тестов.** Сценарий просмотра информации о конфигурации запуска тестов отражен ниже   
в таблице 1.64.

Таблица 1.64 – Сценарий просмотра информации о конфигурации запуска тестов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр информации о конфигурации запуска тестов |
| Идентификатор прецедента | UC5.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций запуска тестов». Пользователь совершил нажатие на названии конфигурации запуска тестов в списке. |
| Действия | В веб-интерфейсе открывается форма для отображения информации о конфигурации запуска тестов. В форме отображается вся информация о конфигурации запуска тестов: название, описание, тип компонента и значения параметров инициализации компонента. |
| Постусловия | – |
| Включение | «Просмотр списка конфигураций запуска тестов» |
| Расширение | – |
| Обобщение | – |

**Изменение конфигурации запуска тестов.** Сценарий изменения конфигурации запуска тестов отражен ниже в таблице 1.65.

Таблица 1.65 – Сценарий изменения конфигурации запуска тестов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Изменение конфигурации запуска тестов |
| Идентификатор прецедента | UC5.4 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций запуска тестов». В панели действий для конфигурации запуска тестов в списке пользователь выбрал действие «Изменить». |
| Действия | В веб-интерфейсе открывается форма для редактирования информации о конфигурации запуска тестов. В форме отображается вся информация о конфигурации запуска тестов, которую можно изменять название, описание, тип компонента и значения параметров инициализации компонента. Если вся измененная новая информация введена корректно, то пользователь может нажать кнопку «Изменить», и система занесет измененные данные о конфигурации запуска тестов в системную БД. |
| Постусловия | Измененная информация о конфигурации запуска тестов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций запуска тестов» |
| Расширение | – |
| Обобщение | – |

**Удаление конфигурации запуска тестов.** Сценарий удаления конфигурации запуска тестов отражен ниже в таблице 1.66.

Таблица 1.66 – Сценарий удаления конфигурации запуска тестов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Удаление конфигурации запуска тестов |
| Идентификатор прецедента | UC5.5 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций запуска тестов». В панели действий для конфигурации запуска тестов в списке пользователь выбрал действие «Удалить». |
| Действия | В веб-интерфейсе открывается диалоговое окно с запросом на подтверждение удаления выбранной конфигурации запуска тестов. Если пользователь подтверждает удаление, то система удаляет конфигурацию запуска тестов из системной БД. |
| Постусловия | Конфигурация запуска тестов успешно удалена из системной БД. |
| Включение | «Просмотр списка конфигураций запуска тестов» |
| Расширение | – |
| Обобщение | – |

**Просмотр списка конфигураций отчетов.** Сценарий просмотра списка конфигураций отчетов отражен ниже в таблице 1.67.

Таблица 1.67 – Сценарий просмотра списка конфигураций отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка конфигураций отчетов |
| Идентификатор прецедента | UC6.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Конфигурации отчетов». |
| Действия | Система подгружает информацию о всех конфигурациях отчетов из системной БД. В веб-интерфейсе отображается список с информацией о конфигурациях отчетов. |
| Постусловия | Система успешно загрузила информацию о конфигурациях отчетов из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Создание конфигурации отчетов.** Сценарий создания конфигурации отчетов отражен ниже в таблице 1.68.

Таблица 1.68 – Сценарий создания конфигурации отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Создание конфигурации отчетов |
| Идентификатор | UC6.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций отчетов». Пользователь выбрал действие «Добавить конфигурацию отчетов» на инструментальной панели. |
| Действия | Пользователю открывается форма для создания новой конфигурации отчетов. Пользователь вводит в форме всю необходимую информацию о конфигурации отчетов: название, описание, тип компонента и значения параметров инициализации компонента. Если вся информация введена корректно, то пользователь может нажать кнопку «Создать» и система занесет данные о новой конфигурации отчетов в системную БД. |
| Постусловия | Введенная информация о новой конфигурации отчетов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций отчетов» |
| Расширение | – |
| Обобщение | – |

**Просмотр информации о конфигурации отчетов.** Сценарий просмотра информации о конфигурации отчетов отражен ниже в таблице 1.69.

Таблица 1.69 – Сценарий просмотра информации о конфигурации отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр информации о конфигурации отчетов |
| Идентификатор прецедента | UC6.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций отчетов». Пользователь совершил нажатие на названии конфигурации отчетов в списке. |
| Действия | В веб-интерфейсе открывается форма для отображения информации о конфигурации отчетов. В форме отображается вся информация о конфигурации отчетов: название, описание, тип компонента и значения параметров инициализации компонента. |
| Постусловия | – |
| Включение | «Просмотр списка конфигураций отчетов» |
| Расширение | – |
| Обобщение | – |

**Изменение конфигурации отчетов.** Сценарий изменения конфигурации отчетов отражен ниже в таблице 1.70.

Таблица 1.70 – Сценарий изменения конфигурации отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Изменение конфигурации отчетов |
| Идентификатор прецедента | UC6.4 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД, СУБД компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций отчетов». В панели действий для конфигурации отчетов в списке пользователь выбрал действие «Изменить». |
| Действия | В веб-интерфейсе открывается форма для редактирования информации о конфигурации отчетов. В форме отображается вся информация о конфигурации отчетов, которую можно изменять название, описание, тип компонента и значения параметров инициализации компонента. Если вся измененная новая информация введена корректно, то пользователь может нажать кнопку «Изменить», и система занесет измененные данные о конфигурации отчетов в системную БД. |
| Постусловия | Измененная информация о конфигурации отчетов корректна и успешно занесена в системную БД. |
| Включение | «Просмотр списка конфигураций отчетов» |
| Расширение | – |
| Обобщение | – |

**Удаление конфигурации отчетов.** Сценарий удаления конфигурации отчетов отражен ниже в таблице 1.71.

Таблица 1.71 – Сценарий удаления конфигурации отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Удаление конфигурации отчетов |
| Идентификатор прецедента | UC6.5 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка конфигураций отчетов». В панели действий для конфигурации отчетов в списке пользователь выбрал действие «Удалить». |
| Действия | В веб-интерфейсе открывается диалоговое окно с запросом на подтверждение удаления выбранной конфигурации отчетов. Если пользователь подтверждает удаление, то система удаляет конфигурацию отчетов из системной БД. |
| Постусловия | Конфигурация отчетов успешно удалена из системной БД. |
| Включение | «Просмотр списка конфигураций отчетов» |
| Расширение | – |
| Обобщение | – |

**Просмотр списка отчетов.** Сценарий просмотра списка отчетов отражен ниже в таблице 1.72.

Таблица 1.72 – Сценарий просмотра списка отчетов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка отчетов |
| Идентификатор прецедента | UC7.1 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр информации о проекте». В веб-интерфейсе пользователь нажимает на кнопку «Показать отчеты». |
| Действия | Система подгружает информацию о всех отчетах, детали которого просматриваются, из системной БД. В веб-интерфейсе отображается список отчетов с указанием даты создания для каждого отчета. |
| Постусловия | Система успешно загрузила информацию об отчетах из БД. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Скачивание файла отчета.** Сценарий скачивания файла отчета отражен ниже в таблице 1.73.

Таблица 1.73 – Сценарий скачивания файла отчета

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Скачивание файла отчета |
| Идентификатор прецедента | UC7.2 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка отчетов». Пользователь нажимает на кнопку «Скачать» рядом с названием отчета в списке. |
| Действия | Система подгружает файл отчета с диска и отправляет его пользователю через веб-браузер. |
| Постусловия | – |
| Включение | «Просмотр списка отчетов» |
| Расширение | – |
| Обобщение | – |

**Просмотр файла отчета.** Сценарий просмотра файла отчета отражен ниже в таблице 1.74.

Таблица 1.74 – Сценарий просмотра файла отчета

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр файла отчета |
| Идентификатор прецедента | UC7.3 |
| Версия | 1 |
| Список акторов | Администратор, Системная СУБД |
| Предусловия | Выполнен сценарий «Просмотр списка отчетов». Пользователь нажимает на название отчета в списке. |
| Действия | Система подгружает файл отчета с диска и отображает его содержимое в веб-интерфейсе. |
| Постусловия | – |
| Включение | «Просмотр списка отчетов» |
| Расширение | «Скачивание файла отчета» |
| Обобщение | – |

**Просмотр списка компонентов.** Сценарий просмотра списка компонентов отражен ниже в таблице 1.75.

Таблица 1.75 – Сценарий просмотра списка компонентов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Просмотр списка компонентов |
| Идентификатор прецедента | UC8.1 |
| Версия | 1 |
| Список акторов | Администратор, Реестр компонентов |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в системном меню «Обзор» пункт «Компоненты». |
| Действия | Система подгружает информацию о всех компонентах из реестра компонентов. В веб-интерфейсе отображается список компонентов с указанием имени, описания и информации о параметрах инициализации для каждого. |
| Постусловия | Система успешно загрузила информацию о компонентах из реестра компонентов. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

**Сборка реестра компонентов.** Сценарий сборки реестра компонентов отражен ниже в таблице 1.76.

Таблица 1.76 – Сценарий сборки реестра компонентов

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Сборка реестра компонентов |
| Идентификатор прецедента | UC8.2 |
| Версия | 1 |
| Список акторов | Администратор, Реестр компонентов |
| Предусловия | Выполнен сценарий «Просмотр списка компонентов». В веб-интерфейсе пользователь нажимает на кнопку «Пересобрать реестр компонентов». |
| Действия | Система выполняет очистку реестра компонентов и повторный сбор информации о компонентах и их регистрацию в системе. В веб-интерфейсе отображается обновленный список компонентов с указанием имени, описания и информации о параметрах инициализации для каждого. |
| Постусловия | Система успешно выполнила очистку реестра сбор информации о компонентах. |
| Включение | «Просмотр списка компонентов» |
| Расширение | – |
| Обобщение | – |

**Выход из системы.** Сценарий выхода из системы ниже в таблице 1.77.

Таблица 1.77 – Сценарий выхода из системы

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание | Выход из системы |
| Идентификатор прецедента | UC9 |
| Версия | 1 |
| Список акторов | Администратор |
| Предусловия | Выполнен сценарий «Авторизация пользователя». Пользователь выбрал в меню системы пункт «Выход из системы», либо время бездействия пользователя превысило определенное значение. |
| Действия | Система закрывает текущую сессию для данного пользователя. |
| Постусловия | Текущая сессия работы с системой успешно завершена. |
| Включение | – |
| Расширение | – |
| Обобщение | – |

## Логический дизайн

Логический дизайн идентифицирует и определяет объекты предметной области проектного решения, их атрибуты, поведение и связи (отношения). Целью логического дизайна является превращение сценариев использования в абстрактную модель, показывающую кооперацию логических компонент реализующих проектное решение. В логическом дизайне будет описана логическая структура системы сборки и тестирования программного обеспечения.

### Объекты проектного решения

Поскольку структура системы сборки и тестирования программного обеспечения довольно объемна, то для упрощения изложения опустим некоторые вспомогательные и несущественные классы и приведем выборочный перечень объектов логического дизайна. Основное внимание уделим сущностям предметной области, с которыми работает система. Все классы можно разбить на следующие группы:

* объекты, связанные с управлением проектами и конфигурациями конвейера;
* объекты, связанные с организацией работы конвейера;
* объекты, связанные с конкретными компонентами;
* вспомогательные объекты, используемые компонентами для выполнения своих задач.

В логическом дизайне не будут рассмотрены объекты веб-интерфейса пользователя, поскольку они зависят от технологии реализации интерфейса.

Управление проектами и конфигурациями конвейера**.** Главной сущностью здесь будет являться проект, связанный с объектами, хранящими в себе конфигурации компонентов, определенного типа, для осуществления работы отдельных шагов конвейера. Основная цель: реализовать логическую структуру, позволяющую легко изменять поведение конвейера для определенного проекта.

Project**.** Класс Project представлен ниже (рисунок 1.14, таблица 1.78).

ProjectDiagram

Рисунок 1.14 – Класс Project (программа StarUML)

Таблица 1.78 – Класс Project

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Сущность, объединяющая в себе конфигурации компонентов для выполнения шагов конвейера. Представляет собой задачу сборки и тестирования, используя данные конфигурации. |
| Описание атрибута: id | Уникальный числовой идентификатор проекта. Обязательный атрибут. |
| Описание атрибута: name | Осмысленное имя проекта. Обязательный атрибут. |
| Описание атрибута: description | Описание проекта. |
| Описание атрибута: maintainer\_email | Адрес электронной почты ответственного за проект. |

Продолжение таблицы 1.78

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание атрибута:  interval | Интервал проверки (в секундах) на наличие новых исходных кодов с помощью компонента поставщика исходных кодов. Обязательный атрибут. |
| Описание атрибута:  revision | Строковый токен, являющийся идентификатором текущей ревизии совокупности исходных кодов, полученных для данного проекта. |
| Описание атрибута:  source | Объект источника исходных кодов, связанный с данным проектом. Подробнее см. Source. |
| Описание атрибута:  report\_configuration | Объект конфигурации отчетов, связанный с данным проектом. Для установки и получения значения атрибута используются методы report\_config и report\_config=. Подробнее см. ReportConfiguration. |
| Описание атрибута:  reports | Множество объектов отчетов о сборке и тестировании, созданных для данного проекта. Подробнее см. Report. |
| Описание ответственности: to\_text | Возвращает текстовое представление информации о проекте. |

**Source.** Класс Source представлен ниже (рисунок 1.15, таблица 1.79).

SourceDiagram

Рисунок 1.15 – Класс Source (программа StarUML)

Таблица 1.79 – Класс Source

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой конкретный источник исходных кодов. Хранит в себе тип компонента, реализующего логику получения исходных кодов, а также набор значений для его параметров конфигурации |
| Описание атрибута: id | Уникальный числовой идентификатор источника исходных кодов. Обязательный атрибут. |
| Описание атрибута: name | Осмысленное имя источника исходных кодов. Обязательный атрибут. |
| Описание атрибута: type | Тип компонента для получения исходных кодов. Обязательный атрибут. |
| Описание атрибута: params | Набор пар «ключ-значение», представляющих собой набор параметров конфигурации конкретного компонента. Обязательный атрибут. |
| Описание атрибута:  projects | Множество проектов, использующих данных источник исходных кодов. Подробнее см. Project. |
| Описание ответственности: to\_text | Возвращает текстовое представление информации об источнике исходных кодов. |

**ReportConfiguration.** Класс ReportConfiguration представлен ниже (рисунок 1.16, таблица 1.80).

ReportConfigurationDiagram

Рисунок 1.16 – Класс ReportConfiguration (программа StarUML)

Таблица 1.80 – Класс ReportConfiguration

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой конкретный источник исходных кодов. Хранит в себе тип компонента, реализующего логику получения исходных кодов, а также набор значений для его параметров конфигурации |
| Описание атрибута: id | Уникальный числовой идентификатор конфигурации отчетов. Обязательный атрибут. |
| Описание атрибута: name | Осмысленное имя конфигурации отчетов. Обязательный атрибут. |
| Описание атрибута: type | Тип компонента для создания отчетов. Обязательный атрибут. |
| Описание атрибута: params | Набор пар «ключ-значение», представляющих собой набор параметров конфигурации конкретного компонента. Обязательный атрибут. |
| Описание атрибута: reports | Множество объектов отчетов созданных с помощью данной конфигурации. Подробнее см. Report. |

Продолжение таблицы 1.80

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание атрибута:  projects | Множество проектов, использующих данную конфигурацию отчетов. Подробнее см. Project. |
| Описание ответственности: to\_text | Возвращает текстовое представление информации о конфигурации отчетов. |

**Report.** Класс Report представлен ниже (рисунок 1.17, таблица 1.81).

ReportDiagram

Рисунок 1.17 – Класс Report (программа StarUML)

Таблица 1.81 – Класс Report

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой сущность, хранящую сводную информацию об отчете. |
| Описание атрибута: id | Уникальный числовой идентификатор отчета. Обязательный атрибут. |
| Описание атрибута: filename | Имя файла отчета. Обязательный атрибут. |
| Описание атрибута: path | Путь к файлу отчета в файловой системе. Обязательный атрибут. |
| Описание атрибута: date | Дата и время создания отчета. Обязательный атрибут. |

Продолжение таблицы 1.81

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание атрибута: project | Объект проекта, которому принадлежит данный отчет. Подробнее см. Project. |
| Описание атрибута:  report\_configuration | Объект конфигурации отчетов, с использованием конфигурации которого создан данный отчет. Для установки и получения значения атрибута используются методы report\_config и report\_config=. Подробнее см. ReportConfiguration. |

**Организация работы конвейера.** Главной сущностью здесь является непосредственно конвейер, который осуществляет вызовы к компонентам и их инициализацию с использованием конфигураций, указанных в проекте. Помимо основного конвейера существует несколько частных конвейеров, реализующих вспомогательную логику для каждого конкретного шага основного конвейера.

**Pipeline.** Класс Pipeline представлен ниже (рисунок 1.18, таблица 1.82).

PipelineDiagram

Рисунок 1.18 – Класс Pipeline (программа StarUML)

Таблица 1.82 – Класс Pipeline

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой абстрактный конвейер. Предоставляет потомках механизмы создания уникальных идентификаторов конвейеров. |
| Описание атрибута: max\_id | Текущее максимальное значение идентификатора конвейера. Обязательный атрибут. |
| Описание атрибута: id | Уникальный числовой идентификатор конвейера. Обязательный атрибут. |
| Описание ответственности: initialize | Создает новый экземпляр конвейера с указанным идентификатором. |
| Описание ответственности: act! | Выполняет один цикл работы конвейера. |
| Описание ответственности: log\_me | Логирует действия конвейера, описанные в виде текстового сообщения. |
| Описание ответственности: get\_id | Формирует уникальный идентификатор конвейера. |

**CommonPipeline.** Класс CommonPipeline представлен ниже (рисунок 1.19, таблица 1.83).

CommonPipelineDiagram

Рисунок 1.19 – Класс CommonPipeline (программа StarUML)

Таблица 1.83 – Класс CommonPipeline

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой основной конвейер, состоящий из шагов получения исходных кодов, сборки, запуска модульных тестов и формирования файла отчета. Наследуется от Pipeline. Осуществляет управление жизненным циклом конвейеров отвечающих за выполнение конкретных шагов. |
| Описание атрибута: project | Проект, для которого выполняется данный конвейер. Обязательный атрибут. Подробнее см. Project. |
| Описание ответственности: act! | Выполняет один цикл работы конвейера, состоящий из четырех шагов. |

**Scheduler.** Класс Scheduler представлен ниже (рисунок 1.20, таблица 1.84).

SchedulerDiagram

Рисунок 1.20 – Класс Scheduler (программа StarUML)

Таблица 1.84 – Класс Scheduler

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой главный объект расписания, управляющий созданием и запуском задач периодического запуска конвейеров. Предоставляет интерфейс для выполнения массовых операций над задачами. |
| Описание атрибута: default | Объект расписания по умолчанию. Обязательный атрибут. |
| Описание атрибута: tasks | Множество задач для данного расписания. Обязательный атрибут. Подробнее см. Task. |
| Описание атрибута: working | Флаг, указывающий, запущены ли задачи данного расписания. Обязательный атрибут. |
| Описание ответственности: default | Возвращает расписание по умолчанию. |
| Описание ответственности: initialize | Создает новый экземпляр класса расписания. |
| Описание ответственности: start | Запускает все задачи для расписания. |
| Описание ответственности: stop | Останавливает выполнение задач расписания. |
| Описание ответственности: spawn\_task | Создает новую задачу для указанного проекта. |
| Описание ответственности: working? | Возвращает статус расписания – запущены его задачи или нет. |

**Task.** Класс Task представлен ниже (рисунок 1.21, таблица 1.85).

TaskDiagram

Рисунок 1.21 – Класс Task (программа StarUML)

Таблица 1.85 – Класс Task

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой задачу периодического запуска конвейера для конкретного проекта в отдельном потоке выполнения. Управляет жизненным циклом объекта основного конвейера. |
| Описание атрибута: project\_id | Идентификатор проекта, для которого выполняется конвейер в данной задаче. Обязательный атрибут. |
| Описание атрибута: interval | Интервал (частота) запуска конвейера в потоке выполнения задачи. Обязательный атрибут. |
| Описание атрибута: loop | Объект потока, в котором выполняется ожидание времени, указанного в интервале, и запуск конвейера. Обязательный атрибут. |
| Описание ответственности: initialize | Создает новый экземпляр задачи для проекта с указанным идентификатором и с указанным интервалом запуска конвейера. |
| Описание ответственности: kill! | Немедленно завершает выполнение задачи. |
| Описание ответственности: start | Запускает задачу на выполнение. |

**Компоненты.** Каждый компонент в системе сборки и тестирования представляет собой, как минимум, одинкласс, являющийся наследником класса, специфичного для компонента определенного типа. Ниже приведено описание классов, используемых компонентом, предназначенным для получения исходных кодов тестируемых программных продуктов из системы управления.

**MoodleSourceProvider.** Класс MoodleSourceProvider представлен ниже (рисунок 1.22, таблица 1.86).

**MoodleSourceProviderDiagram**

Рисунок 1.22 – Класс MoodleSourceProvider (программа StarUML)

Таблица 1.86 – Класс MoodleSourceProvider

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой главный класс компонента, предназначенного для получения исходных кодов из системы управления курсами Moodle 2.0. Является точкой входа в процесс работы компонента. |
| Описание атрибута: host | URL системы Moodle 2.0. Обязательный атрибут. |
| Описание атрибута: token | Строка авторизации для пользователя веб-сервиса, зарегистрированного в системе Moodle 2.0. Обязательный атрибут. |

Продолжение таблицы 1.86

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание атрибута: assignment\_id | Уникальный идентификатор задания в системе Moodle 2.0, решения которого, будет являться целями тестирования в проекте, для которого создается экземпляр данного компонента. Обязательный атрибут. |
| Описание атрибута: szip\_path | Путь к исполняемому файла архиватора 7-zip. Обязательный атрибут. |
| Описание атрибута: client | Экземпляр служебного класса клиента для веб-сервиса Moodle 2.0. Обязательный атрибут. Подробнее см. MoodleServiceClient. |
| Описание атрибута: unpacker | Экземпляр служебного класса распаковщика архивов. Обязательный атрибут. Подробнее см. Unpacker. |
| Описание атрибута: temp\_dir | Путь к директории, предназначенной для хранения временных файлов компонента. |
| Описание ответственности: initialize | Создает новый экземпляр класса компонента с использованием хеша конфигурационных параметров. |
| Описание ответственности: pull | Получает исходные коды целей тестирования из системы Moodle 2.0, используя параметры конфигурации с которыми был создан компонент. |
| Описание ответственности: name | Возвращает человекочитаемое имя компонента. |
| Описание ответственности: params\_definition | Возвращает множество хешей, содержащих описание параметров конфигурации данного компонента. |

**MoodleServiceClient.** Класс MoodleServiceClient представлен ниже (рисунок 1.23, таблица 1.87).

MoodleServiceClientDiagram

Рисунок 1.23 – Класс MoodleServiceClient (программа StarUML)

Таблица 1.87 – Класс MoodleServiceClient

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой клиент для веб-сервиса Moodle 2.0, позволяющего получать файлы решений к заданиям, доступным в учебных курсах системы. |
| Описание атрибута: client | Объект клиента для веб-сервисов для обмена информацией по нестандартной версии протокола REST, реализованному в Moodle 2.0. Обязательный атрибут. |
| Описание ответственности: initialize | Создает новый экземпляр клиента для работы с веб-сервисом Moodle 2.0 для работы с системой развернутой на определенном URL с использованием указанного токена авторизации. |
| Описание ответственности: get\_submissions | Получает список загруженных файлов для конкретного задания с указанием краткой информации по каждому из файлов. |
| Описание ответственности: get\_submission\_file | Получает содержимое файла, загруженного в систему Moodle 2.0 в качестве решения к заданию, по его уникальному идентификатору. |

**Unpacker.** Класс Unpacker представлен ниже (рисунок 1.24, таблица 1.88).

UnpackerDiagram

Рисунок 1.24 – Класс Unpacker (программа StarUML)

Таблица 1.88 – Класс Unpacker

|  |  |
| --- | --- |
| Свойство | Значение |
| Описание объекта | Представляет собой обвертку для работы с архиватором 7-zip, предназначенным для извлечения исходного кода из архивов, загруженных из системы Moodle 2.0. |
| Описание атрибута: szip\_path | Путь к исполняемому файлу архиватора 7-zip. Обязательный атрибут. |
| Описание ответственности: initialize | Создает новый экземпляр распаковщика. |
| Описание ответственности: unpack | Выполняет распаковку содержимого из архива с использованием архиватора 7-zip. |

### Структурные отношения объектов проектного решения

Данный раздел описывает структурные отношения между классами и интерфейсами проектного решения, описание которых приведено в предыдущем разделе. Графически эти структурные отношения показываются на диаграммах классов.

#### Структурные отношения между базовыми классами компонентной системы. Ниже представлены структурные отношения классов компонентной системы между базовым классом компонента и классами компонентов конкретных базовых типов, а также классами, предназначенными для передачи рабочих конвейера между компонентами (рисунок 1.25).

C:\Users\Eskat0n\Documents\Диплом\Диаграммы и графика\ComponentsOverallDiagram.emf

Рисунок 1.25 – Структурные отношения между базовыми классами компонентной системы (программа StarUML)

#### Структурные отношения между классами организации работы конвейера. Ниже представлены структурные отношения классов, предназначенных для организации работы конвейера (рисунок 1.26).

PipelineOverallDiagram

Рисунок 1.26 – Структурные отношения классов организации работы конвейера (программа StarUML)

## Физический дизайн

Физический дизайн показывает реальное воплощение логического дизайна в проектном решении. Путем анализа логический дизайн разбивается на компоненты, которые уже существуют или должны быть разработаны. Таким образом, обеспечивается повторное использование уже существующих решений и разработка решений для повторного использования.

### Инфраструктура проектного решения

Часть «Инфраструктура проектного решения» показывает решения, принятые разработчиками, по организации взаимодействия компонент в системе между собой на основе определенных технологий, механизмов и протоколов.

Общие вопросы размещения и взаимосвязи компонент. Размещение модулей системы. Непосредственно система, а также набор ПО, необходимого для ее работы размещается на виртуальной машине под управлением Microsoft Windows Server 2003. Система управления учебными курсами Moodle 2.0, а также веб-сервис, реализованный в виде расширения к ней, размещены на виртуальной машине, находящейся на хостинг-площадке провайдера под управлением Gentoo Linux.

Размещение модулей системы на виртуальных машинах представлено ниже на схеме (рисунок 1.27).

Размещение - ВМ

Рисунок 1.27 – Схема распределения модулей системы по виртуальным машинам серверов

**Взаимодействие шагов конвейера.** На рисунке 1.28 в виде схемы представлено взаимодействие шагов конвейера сборки и тестирования программного обеспечения. В левой части схемы можно увидеть внешние данные, необходимые для функционирования каждого шага конвейера, которые представлены в правой части. Для каждого шага приведены тип компонента, который осуществляет логику работы данного шага, а также результат работы (промежуточные данные), являющийся в тоже время и входными данными для следующего шага. Исключением является последний шаг создания файла отчета, выходные данные которого представляют собой непосредственно результат работы всего конвейера для данного проекта.

C:\Users\Eskat0n\Dropbox\Диплом\Презентация\Диаграмма.emf

Рисунок 1.28 – Схема взаимодействия шагов конвейера сборки и тестирования

**Взаимодействие клиентской и серверной части.** Это взаимодействие реализовано с помощью протокола HTTP, используемого для получения разметки и стилей страниц веб-интерфейса администратора. Также для организации динамического интерфейса пользователя используется технология AJAX, позволяющая организовать динамическое обновление данных представляемых на странице, без ее полной перезагрузки.

**Взаимодействие с Moodle 2.0.** Взаимодействие с системой управления учебными курсами Moodle 2.0 осуществляется с использованием механизма веб-сервисов.

Обмен данными между клиентом и сервером осуществляется с использованием POST-запросов протокола HTTP, которые содержат имя и параметры вызываемой функции, и ответов в формате XML, в который сериализуются данные, передаваемые сервером.

Официальная терминология документации к Moodle 2.0 называет подобный механизм обмена REST, однако такая его реализация не соответствует сдандарту REST (Representational State Transfer) [15], поскольку использует общий URL для вызова всех функций сервиса и единственный тип запроса (POST).

Для публикации функций сервиса в вебе и организации сериализации/десериализации входных и выходных значений используется Moodle API, позволяющий описывать конвенции имен для параметров и форматы передаваемых данных [38].

Безопасность данных, аутентификация и авторизация. В системе сборки и тестирования программного обеспечения предусмотрена лишь одна роль пользователя – администратор системы. Следовательно администратор имеет полноту прав доступа к системе, согласно указанному в профилях пользователей, приведенных выше в концепции проекта.

Технология единого входа (англ. Single Sign On) – технология, при использовании которой пользователь переходит из одного раздела системы в другой без повторной аутентификации [16]. В разработанной системе технология реализована с помощью единого метода-фильтра авторизации. При аутентификации на странице логина, пользователю выдается ключ, связанный с его сессией в системе. После этого при переходе между страницами, пользователь передает серверной части этот ключ и система проверяет, действителен ли он. Если это так, то серверная часть веб-приложения для управления системой получает информацию о пользователе и считает его авторизованным, если нет – перенаправляет веб-браузер пользователя на страницу авторизации.

Данные учетной записи администратора хранятся в файле конфигурации системы. Пароль хранится в открытом виде.

Для управления доступом к веб-сервису Moodle 2.0 используются механизмы авторизации и аутентификации, встроенные в эту систему управления учебными курсами. Механизм их работы заключается в том, что для администратор систему Moodle 2.0 самостоятельно создает необходимого пользователя, наделяет его правами на доступ к определенным веб-севрисам, при этом генерируется специальный токен авторизации, представляющий собой текстовую строку, которая передается клиенту и используется для получения доступа к веб-сервисам. Передачи логина или пароля пользователя системы Moodle 2.0 при взаимодействии клиента и веб-сервиса в открытом или зашифрованном видео не осуществляется.

Таким образом выполняется требование заказчика о необходимости безопасной передачи данных между системой сборки и тестирования программного обеспечения и системой управления учебными курсами Moodle 2.0.

Конфигурирование и настройка системы. В данном пункте описываются средства конфигурирования и настройки системы. Основным средством настройки и администрирования проектируемой системы являются файлы конфигурации.

Пример файла конфигурации для ССИТПО приведен в приложении А. В конфигурации описаны следующие основные системные настройки:

1. Тип окружения, в котором развернута система, определяющий ее целевое использование – разработка системы, ее тестирование или реальное применение.
2. Настройки веб-интерфейса системы.
3. Настройки безопасности системы.
4. Настройки отправки оповещений по электронной почте.
5. Настройки временных директорий.
6. Настройки журналирования системных событий.
7. Глобальные значения конфигурационных параметров для отдельных компонентов.

Файл конфигурации, представляющий собой программный код на языке Ruby, предназначен для более тонкого конфигурирования поведения веб-интерфейса системы и подключения к базе данных и приведен в приложении Б.

### Развертывание проектного решения

В данном пункте описываются решения по развертыванию программной системы. Принятие этих решений позволит разработчикам и пользователям согласовать все необходимые процедуры, связанные с началом эксплуатации разработанной системы.

Установка программной системы, реализующей проектное решение. Для установки модулей системы необходим набор серверов следующих конфигураций:

1. Сервер, имеющий доступ к сети интернет, для запуска виртуальной машины, на котором будет размещена ССИТПО, а также база данных, используемая системой.
2. Виртуальный сервер, имеющий доступ к сети интернет, для размещения системы управления учебными курсами Moodle 2.0.

Инфраструктура для развертывания проектного решения представлена ниже на схеме (рисунок 1.29).

Инфраструктура

Рисунок 1.29 – Инфраструктура для развертывания проектного решения

На виртуальную машину V1 устанавливается ОС Microsoft Windows 2003, библиотеки .NET Framework 4.0, среда выполнения языка Ruby 1.9.2, компилятор VC++, копируются бинарные и заголовочные файлы библиотеки UnitTest++.

На виртуальном сервере, предназначенном для размещения Moodle 2.0, необходимо иметь установленный веб-сервер Apache, базу данных MySQL, среду выполнения программного кода на языке PHP.

Для установки системы ССИТПО на виртуальный сервер V1, необходимо выполнить распаковку содержимого архива с системой в одну из директорий в файловой системе и запустить установочную подпрограмму install.rb, которая выполнит необходимые начальные конфигурационные работы и разрешит необходимые зависимости.

Для установки веб-сервиса, выполненного в виде расширения для системы Moodle 2.0, необходимо запустить скрипт deploy.sh при использовании виртуального сервера под управлением ОС Linux, или скрипт deploy.bat при использовании виртуального сервера под управлением ОС семейства Windows.

После настройки конфигурационных файлов система тестирования работ студентов готова к работе. Для доступа к веб-интерфейсу системы на машине администратора должен быть установлен веб-браузер и, при необходимости, средства удаленного доступа к виртуальных машинам.

## План пилотного внедрения

Документ «План пилотного внедрения» описывает, какие аспекты проектного решения будут определены как пилотные и обеспечивает необходимое руководство, чтобы пилотное внедрение проекта было успешным. В этом документе отражаются все этапы пилотной эксплуатации реализации проектного решения.

### Общие положения

Обзор плана пилотного внедрения. Внедрение пилотного решения выполняется на кафедре ЭВМ ЮУрГУ в качестве средства проверки студенческих работ по дисциплинам цикла «Программирование».

В плане пилотного внедрения приводятся цели и ожидания от внедрения со стороны заказчика и конечных пользователей, а также со стороны разработчика, на их основе, а также на основе функциональных требований к проектируемой системе, формулируются критерии успешного внедрения. Затем приводится описание подразделения, в котором выполняется внедрение и профили пользователей. Прорабатываются вопросы, связанные с обучением будущих пользователей пилотной версии системы. Разрабатываются этапы пилотного внедрения.

Цели и ожидания от пилотного внедрения. В пилотном внедрении со стороны заказчика и конечных пользователей участвуют следующие заинтересованные лица:

* преподаватель курсов «Программирование на языке С/С++», «Программная инженерия», «Тестирование программного обеспечения» и руководитель дипломного проектирования кафедры ЭВМ;
* студенты 2–4 курса, изучающие данные дисциплины и выполняющие семестровые работы, курсовые проекты и решающие рейтинговые задачи по разработке ПО.

Ожидания преподавателей и студентов:

1. Пилотное решение позволяет преподавателям экономить время при проверке семестровых работ и решений рейтинговых задач.
2. Пилотное решение позволяет преподавателям формализовать процесс проверки, стандартизировать и упорядочить требования, предъявляемые к работам студентов.
3. Пилотное решение предоставляет студентам механизмы наглядного представления допущенных ошибок в виде детальных отчетов о тестировании.
4. Пилотное решение позволяет прозрачно интегрировать модульное тестирование в процесс обучения разработке программного обеспечения, дает возможность студентам использовать разработку через тестирование в своей учебной деятельности.

Ожидания команды разработчиков внедрения системы:

1. Тестирование проектируемой системы с участием заказчика и конечных пользователей:
   * функциональное тестирование разработанной ССИТПО и её взаимодействия с Moodle 2.0 и внешними средствами;
   * тестирование удобства использования веб-интерфейса   
     (юзабилити-тестирование);
   * тестирование надежности, производительности пилотного решения на развернутой инфраструктуре.
2. Проверка по результатам тестирования соответствия требованиям заказчика:
   * соответствие функциональности пилотного решения требованиям к функциям системы, описанных в концепции проекта;
   * соответствие разработанного веб-интерфейса требованиям к интерфейсу пользователя;
   * соответствие спроектированной инфраструктуры для развертывания системы требованиям к производительности, надежности и среде эксплуатации.
3. Выявление в ходе пилотного внедрения дополнительных требований заказчика к системе (выявленные в ходе пилотного внедрения требования описаны ниже в итогах пилотного внедрения).

Критерии успешного пилотного внедрения проекта:

1. Пилотное решение позволило преподавателю:
   * экономить время на проверку работ студентов;
   * формализовать требования, которым должны удовлетворять работы студентов, в виде модульных тестов;
   * отслеживать ход сдачи семестровых работ и решений к рейтинговым задачам.
2. Пилотное решение предоставило студентам:
   * возможность получать в краткие сроки результат проверки своих работ;
   * доступ к детальной информации об ошибках допущенных в своих работах.
3. Преподаватель использует отчеты, создаваемые системой, для информирования студентов о допущенных ими ошибках.

Границы пилотного внедрения. Пилотное решение внедряется в рамках кафедры ЭВМ ЮУрГУ для использования в учебном процессе при преподавании дисциплин «Программирование на языке С/С++», «Программная инженерия», «Тестирование программного обеспечения». Пилотное решение не ориентируется на сборку и тестирование иных проектов, кроме учебных. В рамках пилотного решения внедряется только микроядро системы сборки и тестирования программного обеспечения, базовый набор компонентов для получения исходного кода из файловой системы или из системы управления учебными курсами Moodle 2.0, компоненты для сборки и тестирования работ на языке C++, а также компонент для создания отчетов, представляемых в виде HTML-таблицы.

### Процедуры пилотного внедрения

**Описание подразделения.** Пилотное решение внедряется на кафедре ЭВМ Приборостроительного факультета ЮУрГУ. Кафедра готовит инженеров по специальности «Вычислительные машины, комплексы, системы и сети» – специалистов в области создания компьютерных систем, разработки интегрированных программно-аппаратных комплексов. Основные области знаний студентов: технология разработки, эксплуатации и сопровождения технических и программных средств вычислительной техники; основные направления развития информационных технологий; современные средства вычислительной техники, коммуникаций и связи; сети и сетевые технологии.

В контексте пилотного решения наиболее важно преподавание дисплин, связанных с созданием программных систем и особенностями управления проектами в области разработки ПО – «Программирование на языке С/С++», «Программная инженерия», «Тестирование программного обеспечения». Пилотное решение должно учитывать особенности курсового и дипломного проектирования, выполняемого в рамках кафедры ЭВМ, для адаптации и оптимального использования пилотного решения в учебном процессе в рамках указанных курсов.

При внедрении пилотного решения используется сетевая инфрастуктура кафедры, поэтому необходимо настроить взаимодействие пилотного решения с ней, то есть интегрировать решение в процессе внедрения в сеть кафедры.

**Профили пользователей.** В данном пилотном решении используются профили заинтересованных лиц: преподавателей и студентов, и профили пользователей: администратор, студент. Описание этих профилей приведено выше в концепции проекта (см. пункты «Заинтересованные лица» и «Профили пользователей» подраздела 1.1.6 «Концепция решения»).

**Процедура подготовки и обучения пользователей****.** Обучение преподавателей проводится в виде индивидуальных консультаций. В процессе обучения использованию ССИТПО особое внимание уделяется изучению возможностей по управлению проектами и конфигурациями компонентов: процедурам регистрации студентов в системе, созданию новых проектов и конфигураций, получению отчетов и служебных писем, просмотру и отправке файлов журналов разработчикам. Также демонстрируются возможности применения модульного тестирования в учебном процессе.

Отдельного обучения студентов работе не требуется и может проводиться в рамках ознакомления с возможностями системы управления учебными курсами Moodle 2.0, поскольку установка расширения веб-сервиса, не влияет на внешних вид интерфейса системы Moodle 2.0 и не вносит изменений в ее поведение со стороны обычного пользователя.

Этапы выполнения пилотного внедрения. Ниже описан четкий перечень этапов, согласно которым осуществляется пилотное внедрение (таблица 1.89).

Таблица 1.89 – Этапы выполнения пилотного внедрения

|  |  |  |
| --- | --- | --- |
| Наименование этапа | Критерий  выполнения этапа | Требования для выполнения этапа |
| Анализ особенностей учебного процесса и подразделения | собраны необходимые данные для разработки плана пилотного внедрения | консультации с заказчиком, конечными пользователями |
| Разработка плана пилотного внедрения | утверждение плана пилотного внедрения | консультации с заказчиком |
| Подготовка инфраструктуры для развертывания | инфраструктура развернута и готова для установки системы | предоставленное заказчиком необходимое аппаратное и программное лицензионное обеспечение |
| Установка системы и адаптация при необходимости | система установлена и работоспособна | – |
| Интеграция инфраструктуры системы в сеть кафедры ЭВМ | система получила доступ к требуемым сетевым ресурсам | взаимодействие с системными администраторами кафедры |
| Подготовка обучающих курсов | курсы разработаны и согласованы с заказчиком | консультации с заказчиком |
| Обучение преподавателей | проведены запланированные занятия | дисплейный класс, мультимедийное оборудование |
| Обучение студентов | проведены запланированные занятия | дисплейный класс, мультимедийное оборудование |

Продолжение таблицы 1.89

|  |  |  |
| --- | --- | --- |
| Наименование этапа | Критерий  выполнения этапа | Требования для выполнения этапа |
| Эксплуатация пилотного решения, проведение консультаций и тех. поддержка | проведены необходимые консультации, запланированные учебные занятия с использованием системы успешно проведены | – |
| Сбор статистики по тестированию и использованию системы | данные анкетирования обработаны и составлен отчет о результатах пилотного внедрения | проведение анкетирования заказчика и конечных пользователей |
| Составление акта пилотного внедрения | подписание акта пилотного внедрения | – |

### Итоги пилотного внедрения

В ходе пилотного внедрения выявлены следующие изменения в требованиях заказчика:

1. Необходимость проведения статического анализа исходного кода работ студентов с целью определения сходственных участков для отслеживания плагиата.
2. Необходимость указания метаданных для каждого модульного теста, содержащих информацию о «весе» теста среди других. Данная информация должна использования для автоматического выставления оценки для каждой работы.
3. Возможность уведомления студентов об оценке, поставленной за работу, с указанием ошибок для конкретного студента.
4. Улучшения в веб-интерфейсе администратора система с целью добавления возможности изменения значений настроек, указанных в конфигурационном файле.

При внедрении и использовании пилотного решения в учебном процессе возникал ряд технических проблем. Так канал доступа в Интернет через сеть кафедры был недостаточно надежен и производителен для удаленной работы администратора и получения исходных кодов из системы Moodle 2.0. Проблемы с доступностью системы возникали также из-за периодических проблем с энергообеспечением серверов, связанных с отключением электроэнергии.

Пилотное решение было успешно внедрено на кафедре ЭВМ и используется в учебном процессе. В итоге пилотное внедрение позволило оценить возможности использования модульных тестов как критериев при приемке и оценке работ студентов по дисциплинам цикла «Программирование».

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

Технологический раздел дипломного проекта представлен в виде руководства оператора программной системы в соответствии с ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению» из ЕСПД.

В данном руководстве пользователя будет описана одна из основных частей дипломного проекта – «Система сборки и тестирования программного обеспечения».

## Аннотация

Целью данного дипломного проекта является разработка Системы модульного тестирования студенческих работ, которая будет должна дать преподавателям дисциплин цикла «Технология программирования» возможность использовать модульные тесты для формирования требований к решениям заданий, выдаваемых студентам, быстро и эффективно проверять функциональные возможности студенческих решений, использовать результаты тестирования как основу для выставления оценок. Система также позволяет информировать студентов о результатах тестирования, хранить историю проведенных тестов, что положительно сказывается на общей организации учебного процесса, делает его более открытым для студентов, позволяет избежать предвзятого отношения к человеческому фактору при проверке. Кроме того система должна обеспечивать гибкую настройку своих подсистем (компонентов) и их взаимодействия с помощью удобного и понятного интерфейса, а также интеграцию с системой управления учебными курсами Moodle для получения студенческих работ, выложенных в качестве ответов на задания, опубликованные на сайте на базе Moodle.

Система выполнена в виде трех взаимодействующих частей:

1. Серверная часть, включающая в себя следующие подсистемы (наборы компонентов во внутренней терминологии):
   * компоненты получения исходного кода студенческих работ (в их число входит компонент для получения исходного кода из системы Moodle);
   * компоненты сборки файлов исходных кодов в исполняемые файлы;
   * компоненты запуска модульных тестов и сбора результатов во внутренние представления;
   * компоненты генерации файлов отчетов, о результатах процесса получения, сборки и тестирования;
   * хранилище данных, содержащее как и промежуточные данные для каждого компонента, так и данные необходимые для интеграции компонентов друг с другом;
   * средства гибкого конфигурирования и настройки взаимодействия компонентов системы.
2. Расширение для системы управления учебными курсами Moodle, предоставляющее доступ к выложенным студентами, в качестве решений к заданиям, файлам через механизм веб-сервисов.
3. Веб-интерфейс администратора, предоставляющий возможность взаимодействовать с серверной частью системы и конфигурировать ее.

Таким образом, разрабатываемая программная система имеет широкую планируемую область применения, дает возможность максимальной адаптации основного конвейера под каждый из разрабатываемых проектов. Одной из уникальных возможностей создаваемой системы является ее модульность: динамическое конфигурирование компонентов, использование соглашений имен и возможность замены одного компонента другим «на лету» позволяют наращивать или изменять функциональность без внесения изменений в ядро системы – простым копированием файлов с кодом компонентов в специальные директории.

## Назначение разработки

Основной частью создаваемой в рамках дипломного проекта «Системы модульного тестирования студенческих работ» является «Система сборки и тестирования программного обеспечения» (ССИТПО).

ССИТПО предназначена для управления каждым шагом работы основного конвейера работы системы, состоящего из последовательных операций получения исходных кодов студенческой работы, сборки исполняемого файла из исходных кодов вместе с файлом (файлами) модульных тестов, запуска модульных тестов из получившегося на предыдущем шаге исполняемого файла и создании файла (файлов) отчетов.

Логика, исполняемая на каждом шаге работы конвейера, содержится в компонентах системы, управление которыми осуществляется через конфигурационные файлы и веб-интерфейс ССИТПО.

ССИТПО предназначена для:

* управления проектами: создание, редактирование и удаление проектов;
* управления источниками исходных кодов: создание, редактирование и удаление источников исходных кодов;
* управления конфигурациями отчетов: создание, редактирование и удаление конфигураций отчетов;
* управления механизмами построениями (сборщиками): создание, редактирование и удаление сборщиков;
* управления механизмами запуска и анализа результатов модульных тестов (тестерами): создание, редактирование и удаление тестеров;
* управления реестром зарегистрированных в системе компонентов: обзор компонентов, перезагрузка компонентов без перезапуска всей системы;
* управления общими параметрами конфигурации системы, работой с файловой системой и внешним доступом к системе.

## Условия выполнения программы

Для установки, настройки и использования программной системы пользователь должен обладать профессиональными навыками работы на компьютере, уметь настраивать систему управления учебными курсами Moodle, уметь работать с файлами конфигурации в формате YAML.

Серверная часть системы выполнена в виде консольного приложения, написанного на языке Ruby. Приложение не требует особых операций по установке – все специфичные установочные операции автоматизированы и пользователю предлагается лишь сделать правильный выбор в спорных случаях, возникших при установке.

Минимальные системные требования для успешной установки и работы серверной части системы:

* среда выполнения языка Ruby версии 1.9.2;
* инструменты разработчика для языка Ruby (Ruby Developer Tools);
* пакеты языка Ruby: nokogiri, sinatra, datamapper, haml, json, do-sqlite3;
* процессор не ниже Intel Pentium 4 или эквивалентный ему по производительноси процессор другой архитектуры;
* 1 Гб оперативной памяти или более;
* наличие доступа к сети Интернет со скоростью не менее 256 кб/с при условии удаленного администрирования системы и/или использования компонентов получения исходного кода из удаленных источников.

Для работы идущих в базовой поставке компонентов необходимы также:

* операционная система Windows XP SP3 или более высокой версии;
* компилятор Microsoft Visual C++;
* установленный .NET Framework версии 2.0 или выше;
* архиватор 7zip.

Рекомендуемые системные требования для работы с веб-интерфейсом администратора:

* веб-браузер, поддерживающий стандарты ECMAScript 1.8.1 и CSS 3 (Mozilla Firefox 3.5, Google Chrome 5, Safari 4, Internet Explorer 9, Opera 10.50);
* процессор не ниже Intel Pentium 3 или эквивалентный ему по производительноси процессор другой архитектуры;
* 1 Гб оперативной памяти или более;
* наличие доступа к сети Интернет со скоростью не менее 128 кб/с при условии удаленного администрирования системы.

Расширение для системы управления учебными курсами Moodle предназначено для работы с Moodle версии 2.0 или выше.

## Выполнение программы

**Начало работы и авторизация в системе.** Пользователю для загрузки веб-интерфейса и начала работы с Системой сборки и модульного тестирования проектов необходимо через веб-браузер перейти по адресу http://localhost:9009 (в случае настроек по умолчанию и системе, установленной на компьютере пользователя). В браузере появится окно авторизации (внешний вид окна зависит от используемого браузера), в котором необходимо ввести логин и пароль администратора и нажать на кнопку входа (рисунок 2.1).

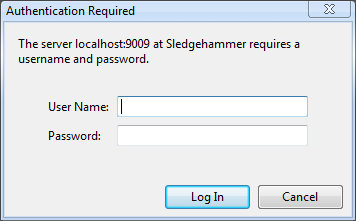


Рисунок 2.1 – Главная страница сайта информационной системы

Система идентифицирует пользователя – проверяет введенные логин   
и пароль пользователя на соответствие указанным в файле конфигурации, в случае успешной проверки система создает для данного пользователя новую сессию и загружает для него стартовую страницу Системы сборки и модульного тестирования программного обеспечения.

**Стартовая страница.** На стартовой странице Системы сопровождения разработки ПО (рисунок 2.2) отображаются:

* ссылки для перехода на страницы отображения различных типов элементов системы (ссылки группы «Обзор»): «Проекты», «Источники исходных кодов»;
* ссылку для перехода на страницу обзора компонентов, зарегистрированных в системе («Компоненты»);
* ссылки для перехода на страницы, содержащие формы добавления проекта, источника исходных кодов или конфигурации отчетов;
* список проектов, созданных в системе – для каждого проекта отображается его название, а также кнопки для редактирования информации о проекте, и удаления проекта.

По умолчанию адрес стартовой страницы совпадает с адресом страницы «Обзор проектов».

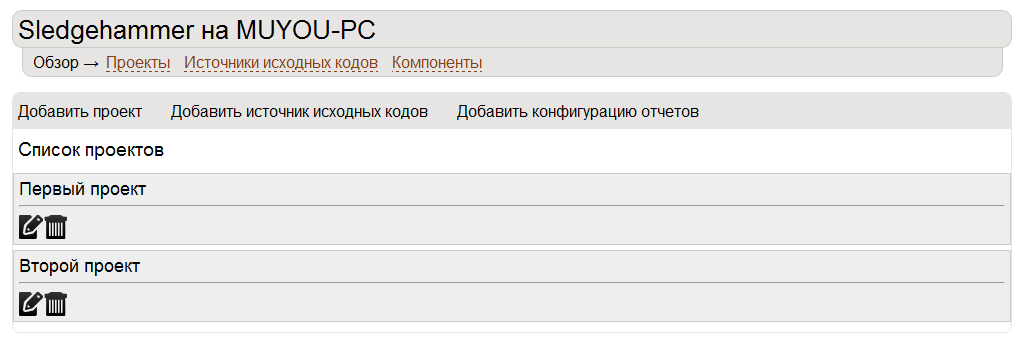


Рисунок 2.2 – Стартовая страница Системы сборки и модульного тестирования программного обеспечения

**Обзор компонентов.** На странице обзора компонентов отображается список зарегистрированных компонентов, сгруппированных по типу компонента (рисунок 2.3). Для каждого компонента приводится его имя, описание (если указано), тип класса Ruby, реализующего функциональность компонента, а также список параметров, которые могут быть переданы компоненту при его инициализации (с описанием каждого параметра). Набор элементов управления в целом аналогичен таковому присутствующему на главной странице, за исключением ссылки «Пересобрать реестр компонентов». При нажатии на нее происходит перерегистрация и повторный сбор информации о всех доступных компонентах. Это может полезно, если пользователю необходимо произвести регистрацию в системе нового компонента без ее перезапуска.

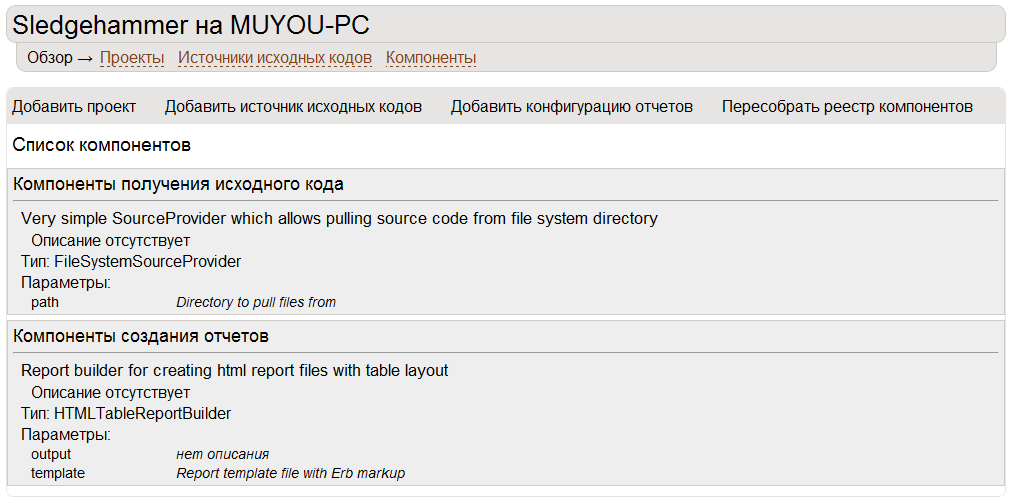


Рисунок 2.3 – Страница обзора компонентов

**Добавление нового проекта.** При переходе по ссылке «Добавить проект» пользователь попадает на страницу, содержащую форму добавления нового проекта (рисунок 2.4). Пользователю требуется в форме ввести необходимую информацию: название нового проекта, интервал его обновления (если данное поле оставить пустым, то интервал будет установлен в значение 60 секунд), выбрать из выпадающих список источник исходных кодов и конфигурацию отчетов (должны быть созданы через веб-интерфейс системы ранее). По желанию можно ввести описание проекта.

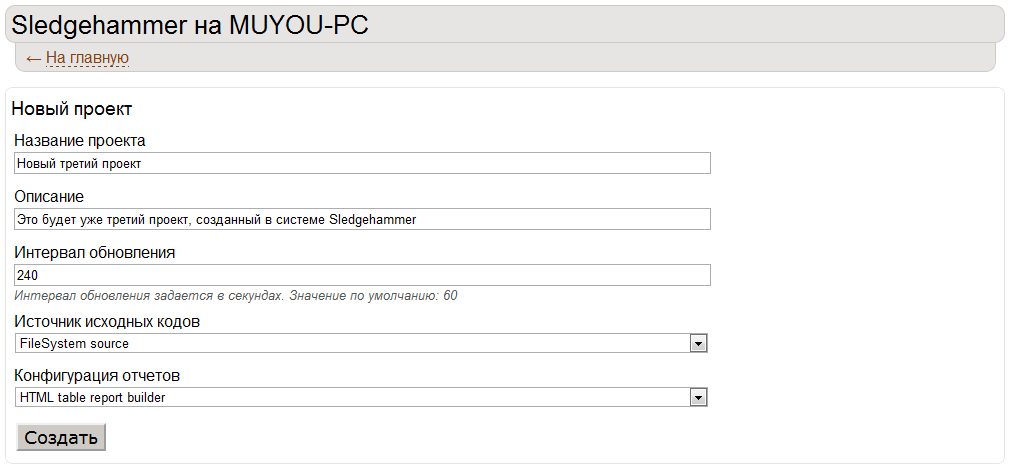
****

Рисунок 2.4 – Страница добавления нового проекта

Если вся информация введена корректно, то пользователь может нажать кнопку «Создать», и данные о новом проекте будут занесены в систему.

**Добавление нового источника исходных кодов.** При переходе по ссылке «Добавить источник исходных кодов» пользователь попадает на страницу, содержащую форму добавления нового источника исходных кодов (рисунок 2.5). На форме пользователь должен указать имя нового источника, из выпадающего списка выбрать тип компонента, который будет обеспечивать собственно логику получения исходных кодов. Также пользователь может указать дополнительные параметры, предназначенные для передачи в компонент, при его инициализации. Каждый параметр задается в форме «имя параметра – значение параметра». Число параметров произвольно, однако действительными являются только те параметры, имена которых указаны в выбранном компоненте.

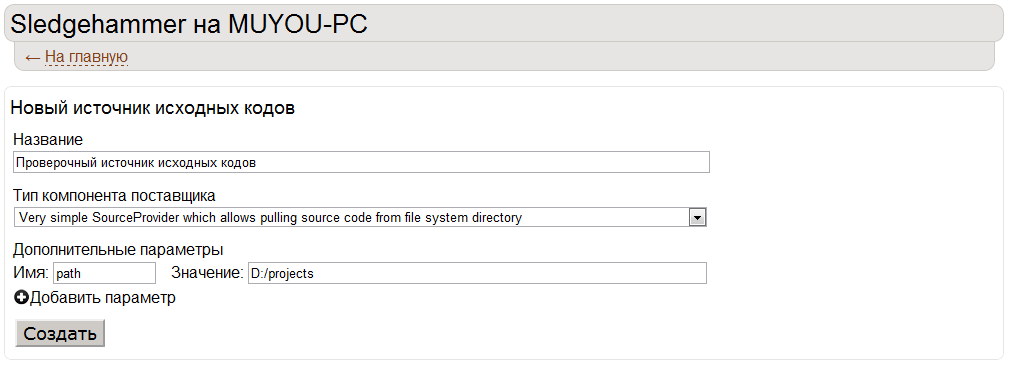


Рисунок 2.5 – Страница добавления нового источника исходных кодов

Если вся информация введена корректно, то пользователь может нажать кнопку «Создать», и данные о новом источнике исходных кодов будут занесены в систему.

Аналогичным образом построена работа и с страницей добавления новой конфигурации отчетов.

## Сообщения оператору

В процессе работы с веб-интерфейсом пользователь может столкнуться с рядом ошибок и технических неисправностей. Система сообщает о возникновении ошибки или неисправности пользователю с помощью отображения специальной страницы, поясняющей причины возникновения ошибки. На рисунке 2.6 приведен пример страницы с сообщением об ошибке: «Не найдено», которая отображается в случае, если пользователь перешел на адресу, который не поддерживается системой.

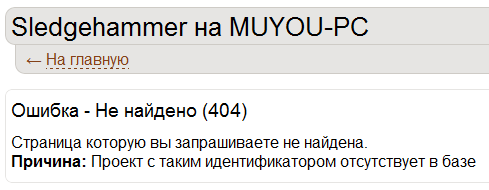


Рисунок 2.6 – Окно с сообщением об ошибке «Не найдено»

Страница с сообщением об ошибке, помимо общего для всех страниц заголовка, содержащего имя экземпляра системы, включает в себя также:

* заголовок, в котором отображено краткое название типа ошибки;
* текст с сообщением об ошибке, в котором приведено описание ошибки;
* текстовое описание возможной причины возникновения ошибки, которое поможет избежать повторения данной ошибки в дальнейшем;
* ссылку «На главную», при нажатии на которую пользователь перенаправляется на главную страницу веб-интерфейса системы, отображающую список всех созданных проектов.

Основные выводимые ошибки бывают двух типов:

1. «*Не найдено*». Такая ошибка возникает в случае попытки перехода по адресу, который не поддерживается системой или в случае обращения к странице деталей или редактирования элемента системы (проекта, поставщика исходных кодов и т.д.), который не существует в базе данных. Текст сообщения об такой ошибке: «Страница которую вы запрашиваете не найдена» (см. рисунок 2.6). В случае возникновении такой ошибки, пользователь должен проверить правильность введенного адреса, если он вводил его вручную или же проверить (например, через отображение полного списка элементов системы) наличие определенного элемента в базе данных. Сделать это можно перейдя по ссылке на главную страницу и переключившись в режим отображения нужного типа элементов. Ошибка может возникнуть из-за того, что пользователь обновил страницу с деталями элемента после того, как в другом окне браузера удалил его.
2. «*Внутренняя ошибка сервера*». Такая ошибка возникает в случае аварийного завершения выполнения операции на сервере. В описании причины ошибки уточняется при каком действии произошла ошибка на сервере. Причинами ошибок могут быть, например, недоступность базы данных (на рисунке 2.7 приведен пример ошибки, возникшей из-за недоступности основной БД), ошибка при работе с файловой системой (файлами конфигурации, временными файлами) (на рисунке 2.8 приведен пример ошибки при неправильно указанном пути до файла конфигурации), и так далее. При постоянном возникновении внутренних ошибок сервера о них необходимо сообщить администратору программной системы, сообщив условия при которых возникают ошибки, а также тип возникающих ошибок.

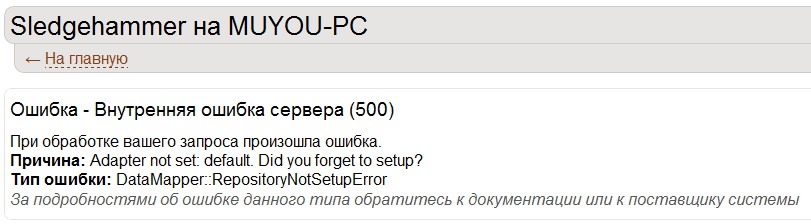


Рисунок 2.7 – Страница с сообщением об ошибке «Внутренняя ошибка сервера» из-за недоступности основной БД

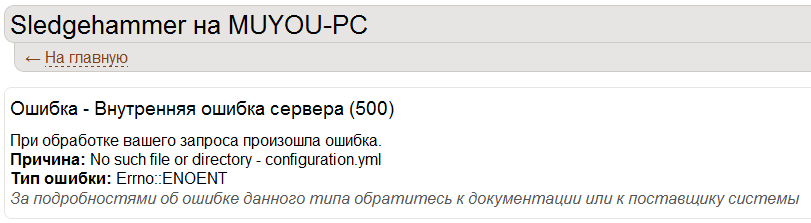


Рисунок 2.8 – Страница с сообщением об ошибке «Внутренняя ошибка сервера» из-за неправильно указанного пути к файлу конфигурации

Кроме сообщений об ошибках, которые аварийно прерывают выполнение какого-либо действия в системе, пользователю также сообщается об общем некорректном состоянии системы, которое приводит к полной невозможности ее правильного функционирования. При наличии такого состояния при переходе на любой адрес веб-интерфейса пользователю отображается системная страница, приведенная на рисунке 2.9. На данной странице отображается список критических ошибок, а также список предупреждений, которые, возможно, могут привести к некорректной работе.

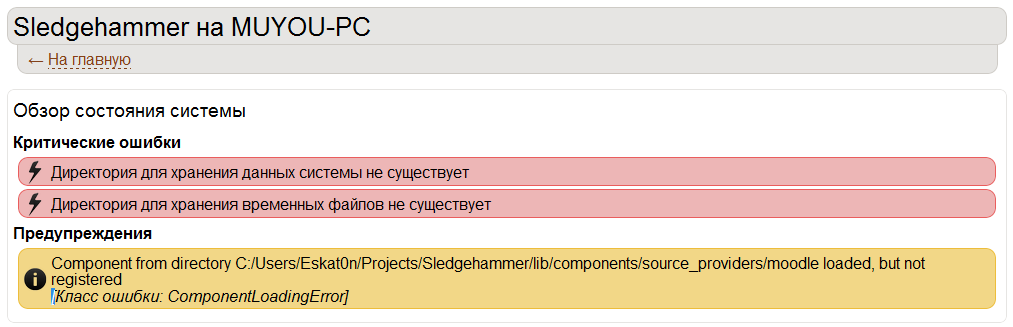


Рисунок 2.9 – Страница с обзором состояния системы

Пользователю также сообщается о некорректности вводимых данных, на страницах, которые содержат формы для ввода. Проверка корректности осуществляется самим веб-интерфейсом. В случае ввода некорректных данных пользователем под поле ввода с некорректными данными появится сообщение, выделенное красным цветом и содержащее подсказку о некорректных данных.

Чтобы успешно завершить необходимое действие, пользователь должен ввести все данные правильно. На рисунке 2.10 приведен пример ввода некорректных данных в форме создания нового проекта.

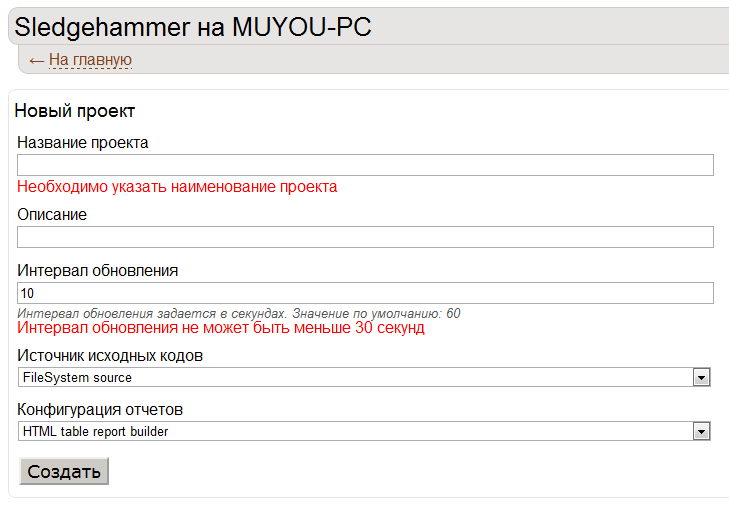


Рисунок 2.10 – Сообщения при вводе некорректных данных в форме создания нового проекта

# БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

Целью дипломного проектирования является разработка программной системы, предназначенной для оценки студенческих работ по дисциплинам цикла «Программирование» с помощью модульного тестирования. Ее особенностями является интеграция с веб-порталом поддержки образовательного процесса item74.ru (или любым другим сайтом, созданным на базе системы управления учебными курсами Moodle 2.0), а также возможность гибкой конфигурации параметров получения исходных кодов, сборки, тестирования и формирования отчетов через удобный веб-интерфейс.

## Анализ достоинств интерфейса пользователя разработанной программы

Интерфейс разрабатываемой системы предназначен для выполнения администраторских функций – управления параметрами проектов, источников исходных кодов и конфигураций отчетов, поэтому во главу угла при его проектировании ставилась максимальная функциональность и информативность. При этом необходимо, чтобы интерфейс был нацелен на повышение производительности оператора, должен быть интуитивно-понятен и не требовать существенных затрат времени на обучение его использованию, позволяя уделить больше времени планированию конфигурации компонентов системы.

Поскольку система ориентирована на пользователя с профессиональными навыками работы на компьютере, то интуитивность интерфейса обеспечивается его схожестью с интерфейсами тех программ, к которым привык пользователь: системами управления проектами, системами непрерывной интеграции, интерфейсами оболочек для работы с базами данных и др.

Немалую роль играет и цветовая гамма интерфейса. В нем отсутствуют яркие цвета, отвлекающие внимание пользователя от органов управления и представленной на экране информации. При этом, с целью привлечения внимания пользователя, сообщения об ошибках выделены красным или оранжевым цветами мягких оттенков, чтобы ненавязчиво выполнять сигнализирующие функции.

Любой программе свойственны особенные черты и предоставляемые возможности. Поэтому помимо интуитивности интерфейс должен быть понятным для пользователя, чтобы пользователь мог легко найти требуемую функциональность. Это отчасти достигается логической группировкой элементов управления, расположенных на страницах веб-интерфейса, в отдельные визуальные блоки; отчасти – сокрытием избыточной информации, контекстным характером использования функций, вынесенные в органы управления.

Перечислим достоинства интерфейса разрабатываемой нами системы модульного тестирования студенческих работ:

* использование графического интерфейса: используется удобный   
  веб-интерфейс с панелью меню и отдельными страницами-формами для ввода информации;
* наличие подсказок о назначении полей ввода, действиях совершаемых на формах;
* наличие справок о работе программы в целом и ее отдельных частей: предусмотрены краткие подсказки для пользователя рядом с элементами управления, а также доступно подробное руководство пользователя;
* использование наглядных способов предъявления информации пользователю: списковая форма – для вывода информации о проектах, источниках исходного кода, конфигурациях отчетов, компонентах;
* отдельные страницы – для ввода/вывода информации о выбранных элементах;
* блокировка ошибочных действий пользователя: в большинстве форм для ввода информации предусмотрена проверка вводимой информации, индикация о некорректности данных;
* наличие пояснений ошибок для пользователя: информация обо всех ошибках системы, вызванных как действиями самого пользователя, так и неполадками в программном или аппаратном обеспечении, выводится на экран, при этом предусмотрена возможность обращения к справке, содержащей информацию о возникшей ошибке;
* наличие сигнализации об аварийных режимах работы системы: на отдельной странице выводится информация о неполадках, при этом уточняются причины аварии в виде списка, для каждой причины отображаются ссылки на соответствующие разделы справки, содержащие инструкции по устранению аварии.

## Рекомендации по организации рабочего места пользователя

Пользователи разрабатываемой системы вынуждены работать с ПЭВМ. Рассмотрим основные нормативные документы и приведем некоторые рекомендации по организации рабочего места пользователя.

### Рекомендации по выбору помещения для размещения рабочего места

При работе с вычислительной техникой решающим фактором, обеспечивающим высокий уровень работоспособности, является правильно спроектированное помещение и освещение.

СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [24] предъявляет следующие требования к помещению для работы с ПЭВМ.

Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при наличии расчетов, обосновывающих соответствие нормам естественного освещения и безопасность их деятельности для здоровья работающих.

Естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток.

Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и другие.

Не допускается размещение мест пользователей ПЭВМ в цокольных   
и подвальных помещениях.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электроннолучевой трубки (ЭЛТ) должна составлять не менее 6 м2, в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) – 4,5 м2.

При использовании ПВЭМ с ВДТ на базе ЭЛТ (без вспомогательных устройств – принтер, сканер и другие), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы   
 менее 4-х часов в день допускается минимальная площадь 4,5 м2 на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка – 0,7-0,8; для стен – 0,5-0,6; для пола –   
0,3-0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ.

В помещении должно удовлетворять требованиям пожарной безопасности. Обязательно наличие огнетушителя, термодатчиков. В больших помещениях желательно наличие плана эвакуации.

### Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ

Согласно СанПиН 2.2.2/2.4.1340-03 [24] в производственных помещениях, в которых работа с использованием ПЭВМ является вспомогательной, температура, относительная влажность и скорость движения воздуха на рабочих местах должны соответствовать действующим санитарным нормам микроклимата производственных помещений.

В производственных помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а и 1б в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям указанных нормативов (таблица 3.1).

Таблица 3.1 – Гигиенические требования к микроклимату производственных помещений (СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений») [25]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Период года | Категория работ по уровням энергозатрат, Вт | Темпера-тура воздуха, °С | Температура поверх-ностей, °С | Относи-тельная влажность воздуха, % | Скорость движения воз-  духа, м/с |
| Холодный | 1а (до 139) | 22-24 | 21-25 | 60-40 | 0,1 |
| 1б (140-174) | 21-23 | 20-24 | 60-40 | 0,1 |
| Теплый | 1а (до 139) | 23-25 | 22-26 | 60-40 | 0,1 |
| 1б (140-174) | 22-24 | 21-25 | 60-40 | 0,1 |

Так как разрабатываемая система может применяться в учебном процессе,   
то необходимо отметить, что согласно СанПиН 2.2.2/2.4.1340-03 [24] в помещениях всех типов образовательных и культурно-развлекательных учреждений для детей и подростков, где расположены ПЭВМ, должны обеспечиваться оптимальные параметры микроклимата (таблица 3.2).

Таблица 3.2 – Оптимальные параметры микроклимата во всех типах учебных и дошкольных помещений с использованием ПЭВМ (СанПиН 2.2.4.548-96) [25]

|  |  |  |  |
| --- | --- | --- | --- |
| Температура, С° | Относительная влажность, % | Абсолютная влажность, г/м3 | Скорость движения воздуха, м/с, не менее |
| 19 | 62 | 10 | 0,1 |
| 20 | 58 | 10 | 0,1 |
| 21 | 55 | 10 | 0,1 |

В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

Уровни положительных и отрицательных аэроионов в воздухе помещений, где расположены ПЭВМ, должны соответствовать действующим санитарно-эпидемиологическим нормативам.

Содержание вредных химических веществ в воздухе производственных помещений, в которых работа с использованием ПЭВМ является вспомогательной, не должно превышать предельно допустимых концентраций вредных веществ в воздухе рабочей зоны в соответствии с действующими гигиеническими нормативами (ГН 2.1.6.1338-03 «Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест» [26]).

Содержание вредных химических веществ в производственных помещениях, в которых работа с использованием ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и другие), не должно превышать предельно допустимых концентраций загрязняющих веществ в атмосферном воздухе населенных мест в соответствии с действующими гигиеническими нормативами.

Содержание вредных химических веществ в воздухе помещений, предназначенных для использования ПЭВМ во всех типах образовательных учреждений, не должно превышать предельно допустимых среднесуточных концентраций для атмосферного воздуха в соответствии с действующими санитарно-эпидемиологическими нормативами.

Показатели концентрации аэроионов должны поддерживаться оптимальными по СаНПиН 2.2.4.1294-03 «Санитарно-гигиенические нормы допустимых уровней ионизации воздуха» [27] (таблица 3.3).

Таблица 3.3 – Значения нормируемых показателей концентраций аэроионов и коэффициента униполярности (СаНПиН 2.2.4.1294-03 [27])

|  |  |  |  |
| --- | --- | --- | --- |
| Нормируемые показатели | Концентрация аэроионов, ион/см3 | | Коэффициент униполярности |
| положительной полярности | отрицательной полярности |
| Минимально допустимые, не менее | 400 | 600 | от 0,4 до 1,0 включ. |
| Оптимальные | от 1500 до 3000 | от 30 000 до 50 000 |
| Максимально допустимые, не более | 5000 | 50 000 |

В помещении обеспечиваются оптимальные параметры микроклимата для категории работ 1а. Проводится ежедневная влажная уборка и проветривание.

### Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ

Основными источниками шума в помещении являются персональные компьютеры, устройства ввода/вывода (принтер, сканер, копир).

Шум оказывает различное действие в зависимости от уровня, характера, продолжительности и индивидуальных способностей человека. Шум, даже если он не велик (50-60 дБ), создает значительную нагрузку на нервную систему человека, оказывая на него психологическое воздействие.

Постоянные воздействия шума вызывают повышенную утомленность, головную боль, головокружение, бессонницу, замедление реакции и другие. Современные системы охлаждения и дисководы имеют низкий уровень шума, а использование лазерных принтеров позволяет добиться ещё лучших результатов.

В соответствие с нормами СанПиН 2.2.2/2.4.1340-03 [24] в производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарно-эпидемиологическими нормативами.

В помещениях всех образовательных и культурно-развлекательных учреждений для детей и подростков, где расположены ПЭВМ, уровни шума не должны превышать допустимых значений, установленных для жилых и общественных зданий.

В помещениях всех типов образовательных и культурно-развлекательных учреждений, в которых эксплуатируются ПЭВМ, уровень вибрации не должен превышать допустимых значений для жилых и общественных зданий в соответствии с действующими санитарно-эпидемиологическими нормативами.

Допустимые уровни шума для «Творческой деятельности, научной деятельности, программирования, преподавания и обучения» в октавных полосах согласно СанПиН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» [28] приведены ниже (таблица 3.4).

Таблица 3.4 – Допустимые уровни звукового давления ПЭВМ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Уровни звукового давления, дБ, в октавных полосах со среднеметрическими частотами, Гц | | | | | | | | | Уровни звука и эквивалентные уровни звука, дБА |
| 31 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 | – |
| 93 | 79 | 70 | 68 | 58 | 55 | 52 | 52 | 49 | 60 |

Реальный уровень шума в помещении должен не превышать указанной величины. Для снижения уровня шума, проникающего в производственное помещение извне, оно должно оборудоваться звукопоглощающими облицовками,   
а также различными звукопоглощающими устройствами (перегородки, кожухи, прокладки), уплотнением по периметру притворов окон, дверей.

При выполнении работ с использованием ПЭВМ в производственных помещениях уровень вибрации не должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип «в») в соответствии с действующими санитарно-эпидемиологическими нормативами СанПиН 2.2.4/2.1.8.566-96 «Производственная вибрация. Вибрация в помещениях жилых и общественных зданий» [29].

Шумящее оборудование, уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ.

### Требования к освещению на рабочих местах, оборудованных ПЭВМ

При работе с вычислительной техникой решающим фактором, обеспечивающим высокий уровень работоспособности, является правильно спроектированное освещение. Работа программиста требует большой зрительной нагрузки, поэтому помимо естественного, применяется искусственное освещение.

Согласно СанПиН 2.2.2/2.4.1340-03 [24] рабочие столы следует размещать таким образом, чтобы ВДТ были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева. Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Следует ограничивать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и другие), находящихся в поле зрения, должна быть не более 200 кд/м2.

Следует ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и другие) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного  
 и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м2 и яркость потолка не должна превышать 200 кд/м2.

Показатель ослепленности для источников общего искусственного освещения в производственных помещениях должен быть не более 20. Показатель дискомфорта в административно-общественных помещениях не более 40, в дошкольных и учебных помещениях не более 15.

Яркость светильников общего освещения в зоне углов излучения   
от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м2, защитный угол светильников должен быть   
не менее 40 градусов. Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее 40 градусов. Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1-5:1, а между рабочими поверхностями и стенами и оборудованием 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пуско-регулирующими аппаратами (ЭПРА). Применение светильников без рассеивателей и экранирующих решеток не допускается. При отсутствии светильников   
с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя   
при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору. Коэффициент пульсации не должен превышать 5%. Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

### Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ

Источниками электромагнитного излучения являются мониторы и блоки питания. Для защиты от электромагнитного излучения последних предусмотрено экранирование и заземление блоков питания (экранов блоков).

СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [24] предъявляет нижеприведенные требования и нормы:

* мощность экспозиционной дозы мягкого рентгеновского излучения   
  в любой точке на расстоянии 0,05 м от экрана и корпуса ВДТ (на электроннолучевой трубке) при любых положениях регулировочных устройств не должна превышать 1 мкЗв/ч (100 мкР/ч);
* конструкция ВДТ должна предусматривать регулирование яркости   
  и контрастности.

Остальные требования к видео-дисплейным устройствам (ВДУ) описаны ниже (таблица 3.5, таблица 3.6таблица ).

Таблица 3.5 – Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей

|  |  |  |
| --- | --- | --- |
|  | Диапазон частот, Гц | ВДУ |
| Напряженность электрического поля, В/м | от 5 до 2×103 | 25,0 |
| от 2×103 до 400×103 | 2,5 |
| Плотность магнитного потока, нТл | от 5 до 2×103 | 250,0 |
| от 2×103 до 400×103 | 25,0 |
| Напряженность электростатического поля, кВ/м | | 15,0 |

Таблица 3.6 – Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах

|  |  |
| --- | --- |
| Параметры | Допустимые значения |
| Яркость белого поля, кд/м2, не менее | 35 |
| Неравномерность яркости рабочего  поля, не более | ±20% |
| Контрастность (для монохромного режима), не менее | 3:1 |
| Временная нестабильность изображения (мелькания) | Не должна фиксироваться |
| Пространственная нестабильность изображения (дрожание), не более | 2 × 10-4L, где L – проектное расстояние наблюдения, мм |

Заметим, что современные мониторы удовлетворяют жестким требованиям на уровни электромагнитного излучения. На поверхность их экрана нанесено антибликовое и антиотражающее покрытие (ARAG). При применении подобных мониторов использование защитных экранов необязательно.

### Общие требования к организации рабочих мест пользователей ПЭВМ

Рабочее место оператора ЭВМ проектируется согласно СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [24].

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных производственных факторов должны размещаться в изолированных кабинах с организованным воздухообменом.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5-2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5-0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы   
с ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки   
от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680-800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь пространство для ног высотой   
не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен –   
не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей (размер ВДТ и ПЭВМ, клавиатуры, и другие), характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики.

Конструкция рабочего стула должна обеспечивать:

* ширину и глубину поверхности сиденья не менее 400 мм;
* поверхность сиденья с закругленным передним краем;
* регулировку высоты поверхности сиденья в пределах 400-550 мм и углам наклона вперед до 15°, и назад до 5°;
* высоту опорной поверхности спинки 300±20 мм, ширину – не менее   
  380 мм и радиус кривизны горизонтальной плоскости – 400 мм;
* угол наклона спинки в вертикальной плоскости в пределах ±30°;
* регулировку расстояния спинки от переднего края сиденья в пределах

260-400 мм;

* стационарные или съемные подлокотники длиной не менее 250 мм и шириной – 50-70 мм;
* регулировку подлокотников по высоте над сиденьем в пределах 230 ±30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии   
100-300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

### Требования к организации и оборудованию рабочих мест с ПЭВМ для обучающихся в общеобразовательных учреждениях и учреждениях начального и высшего профессионального образования

Поскольку разрабатываемая система предназначена в том числе для использования в учебном процессе в учреждении высшего профессионального образования, то необходимо отметить, что согласно СанПиН 2.2.2/2.4.1340-03 [24]   
для данной категории пользователей на организацию и оборудование рабочих мест с ПЭВМ накладываются более жесткие требования, чем для взрослых пользователей.

Помещения для занятий оборудуются одноместными столами, предназначенными для работы с ПЭВМ.

Конструкция одноместного стола для работы с ПЭВМ должна предусматривать:

* две раздельные поверхности: одна горизонтальная для размещения ПЭВМ с плавной регулировкой по высоте в пределах 520-760 мм и вторая – для клавиатуры с плавной регулировкой по высоте и углу наклона от 0 до 15 градусов с надежной фиксацией в оптимальном рабочем положении (12-15 градусов);
* ширину поверхностей для ВДТ и клавиатуры не менее 750 мм (ширина обеих поверхностей должна быть одинаковой) и глубину   
  не менее 550 мм;
* опору поверхностей для ПЭВМ или ВДТ и для клавиатуры на стояк, в котором должны находится провода электропитания и кабель локальной сети.
* основание стояка следует совмещать с подставкой для ног;
* отсутствие ящиков;
* увеличение ширины поверхностей до 1200 мм при оснащении рабочего места принтером.

Высота края стола, обращенного к работающему с ПЭВМ, и высота пространства для ног должны соответствовать росту обучающихся в обуви.

При наличии высокого стола и стула, несоответствующего росту обучающихся, следует использовать регулируемую по высоте подставку для ног.

Линия взора должна быть перпендикулярна центру экрана и оптимальное ее отклонение от перпендикуляра, проходящего через центр экрана в вертикальной плоскости, не должно превышать ±5 градусов, допустимое ±10 градусов.

Рабочее место с ПЭВМ оборудуют стулом, основные размеры которого должны соответствовать росту обучающихся в обуви.

### Электробезопасность

По степени опасности поражения электрическим током согласно правилам устройства электроустановок рабочее помещение относится к классу помещений с повышенной опасностью, так как имеется возможность одновременного прикосновения человека к имеющим соединения с землей металлоконструкциям здания и с металлическими корпусами электрооборудования с другой.

Если физический доступ к токоведущим частям оборудования затруднен, то основной причиной возникновения данного опасного фактора может являться прикосновение к металлическим нетоковедущим частям (например, корпусу ПЭВМ), которые могут оказаться под напряжением в результате повреждения изоляции. В соответствии с правилами электробезопасности, в служебном помещении должен осуществляться постоянный контроль состояния электропроводки, предохранительных щитов, шнуров, с помощью которых включаются в электросеть компьютеры, осветительные приборы, другие электроприборы.

Для предотвращения образования и защиты от статического электричества в помещениях с ПК необходимо использовать аэроионизаторы и увлажнители воздуха. В отделке помещений следует отдавать предпочтение антистатическим материалам. Полы должны иметь антистатическое покрытие.

Согласно ГОСТ 12.1.019-79 ССБТ «Электробезопасность. Общие требования и номенклатура видов защиты» [30] основными требованиями от поражения электрическим током являются:

* обеспечение недоступности токоведущих частей, находящихся под напряжением, для случайного прикосновения;
* устранение опасности поражения при появлении напряжения на корпусе и других частях электрооборудования путем зануления корпусов электрических приборов (ГОСТ 12.1.030-81 ССБТ «Электробезопасность. Защитное заземление, зануление» [31]);
* применение предупреждающей сигнализации и знаков безопасности.

### Пожарная безопасность

Наиболее вероятными возможными причинами пожара в помещении могут быть различные повреждения электросети, прежде всего короткие замыкания. Согласно ГОСТ 12.1.004-91 ССБТ «Пожарная безопасность. Общие требования» [32] могут быть применены следующие мероприятия по пожарной безопасности:

* применение быстродействующих устройств защитного отключения возможных источников возгорания (электрооборудования);
* применение средств пожарной сигнализации;
* эвакуация людей. Число эвакуационных выходов из здания должно быть не менее двух. Эвакуационные выходы должны располагаться рассредоточено. Ширина участков путей эвакуации должна быть не менее 1 м,   
  а дверей на путях эвакуации – не менее 0,8 м. Высота прохода –   
  не менее 2 м;
* разработка мероприятий на случай возникновения пожара.

В начальной стадии пожара для тушения электропроводки (под напряжением до 100 В) можно использовать порошковые огнетушители.

## Рекомендации по организации режима труда и отдыха пользователя

Режимы труда и отдыха при работе с ПЭВМ и ВДТ должны организовываться в зависимости от вида и категории трудовой деятельности.

По виду трудовой деятельности работу оператора можно отнести к группе «А» – работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом.

Для видов трудовой деятельности устанавливается три категории тяжести и напряженности работы с ВДТ и ПЭВМ.

Для группы А категории определяются по суммарному числу считываемых знаков за рабочую смену, но не более 60 000 знаков за смену: 1 категория –   
до 20 000 знаков; 2 категория – до 40 000 знаков; 3 категория – до 60 000 знаков.

Продолжительность обеденного перерыва определяется действующим законодательством о труде и правилами внутреннего трудового распорядка предприятия.

Для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении рабочей смены должны устанавливаться регламентированные перерывы. Время регламентированных перерывов в течении рабочей смены следует устанавливать в зависимости от ее продолжительности, вида и категории трудовой деятельности. Продолжительность непрерывной работы без регламентированного перерыва не должна превышать двух часов. При восьмичасовой рабочей смене и работе на видео-дисплейном терминале (ВДТ) и ПЭВМ регламентированные перерывы следует устанавливать:

* для I категории работ через 2 ч. от начала рабочей смены и через 2 ч.   
  после обеденного перерыва продолжительностью 15 мин. каждый;
* для II категории работ через 2 ч. от начала рабочей смены и   
  через 1,5-2 ч. после обеденного перерыва продолжительностью 15 мин. каждый или продолжительностью 10 мин через каждый час работы;
* для III категории работ через 2 ч от начала рабочей смены и   
  через 1,5-2,0 ч. после обеденного перерыва продолжительностью 20 мин. каждый или продолжительностью 15 мин. через каждый час работы.

Во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления целесообразно выполнять комплексы упражнений для глаз,  
 для улучшения мозгового кровообращения, для снятия утомления с плечевого пояса и рук, а также общего воздействия.

В случаях возникновения у пользователя зрительного дискомфорта и других неблагоприятных субъективных ощущений, несмотря на соблюдение санитарно-гигиенических, эргономических требований, режимов труда и отдыха следует применять индивидуальный подход в ограничении времени работ с ВДТ   
и ПЭВМ. Коррекцию длительности перерывов для отдыха или проводить смену деятельности на другую, не связанную с использованием ВДТ и ПЭВМ [33].

# ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

В данном разделе проводится сетевое планирование дипломного проекта, расчет затрат на разработку проекта и анализ его технико-экономической эффективности.

## Сетевое планирование

Целью дипломного проекта является создание Системы модульного тестирования для приема и оценки заданий по дисциплинам цикла «Программирование».

Для составления плана работ по выполнению дипломного проектирования будем использовать методы сетевого планирования, разработанные в начале   
50-х годов, широко и успешно применяемые для оптимизации планирования и управления сложными и разветвленными комплексами работ, требующими участия большого числа исполнителей и затрат ограниченных ресурсов. Эти методы позволяют отобразить логическую связь узловых событий и работ, обеспечивающих достижение поставленной задачи. С помощью методов сетевого планирования легко найти критический путь и сосредоточить внимание на важных работах, от которых в большей степени зависит выполнение всей работы в срок. Сетевое планирование также позволяет определить величину трудоемкости работ, необходимую для расчета заработной платы [].

До начала построения сетевого графика нужно составить перечень работ, необходимый для реализации дипломного проекта:

1. Получение и анализ задания на разработку программной системы.
2. Разработка сетевого графика.
3. Подбор литературы и электронных источников информации.
4. Поиск существующих решений в данной области.
5. Анализ литературы и электронных источников информации (выявление аналогов и эффективных программных решений).
6. Анализ существующих решений.
7. Анализ предметной области и бизнес-логики процессов заказчика.
8. Выбор стека технологий и протоколов.
9. Создание прототипов интерфейса для модуля интеграции с платформой Moodle 2.0.
10. Разработка описания протокола взаимодействия модуля интеграции и сервисной части программной системы.
11. Документирование протокола взаимодействия диаграммами UML.
12. Анализ функций пилотной версии программной системы.
13. Ранжирование реализуемых пилотной версией функций по затратам на их разработку и интеграцию.
14. Выявление эффективных программных решений требуемых функций в существующих аналогах.
15. Описание доменных сущностей, составление первой версии диаграммы классов.
16. Настройка и установка среды разработки и тестирования.
17. Разработка пилотной версии ПС.
18. Проведение интеграционного тестирования пилотной версии ПС.
19. Проведение нагрузочного тестирования пилотной версии ПС.
20. Расчет показателей эффективности пилотной версии.
21. Встреча с заказчиком, проработка предложений по модернизации пилотной версии.
22. Составление финальной версии диаграммы классов.
23. Разработка финальной версии ПС.
24. Проведение интеграционного тестирования финальной версии ПС.
25. Проведение нагрузочного тестирования финальной версии ПС.
26. Расчет показателей эффективности.
27. Подготовка пакетов для развертывания системы и миграции данных.
28. Составление пользовательской документации.
29. Оформление ПЗ.
30. Сдача заказчику.

Работы из перечня будут распределены между 4 работниками: руководителем, двумя инженерами-программистами и лаборантом.

### Расчет ожидаемой продолжительности выполнения работ

Перечень работ, продолжительность выполнения, параметры и вероятностные характеристики приведены ниже (таблица 4.1).

На основании минимальной продолжительности работы *tmin* и максимальной продолжительности работы *tmax* для каждой работы были рассчитаны следующие показатели:

Ожидаемая продолжительность работы *tож*, рассчитанная по формуле (1).

tij ожид = 0,6tij min + 0,4tij max.

(1)

Среднеквадратическое отклонение  рассчитано по формуле (2).

δij=0,2(tmax-tmin).

(2)

Дисперсия  рассчитана по формуле (3).

Dij=0,04(tmax-tmin)2.

(3)

Таблица 4.1 – Перечень работ, параметры и вероятностные характеристики

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Продолжи-тельность, дн. | | | Исполни-тели, чел. | | | Среднеквадратичное  оклонение, дн. | Дисперсия, дн2 |
| мин. | макс | ожид. | руковод. | инженер | лаборант |
| 0-1 | Получение и анализ задания на разработку программной системы | 2 | 3 | 2 | 1 | 2 | 1 | 0,2 | 0,04 |
| 1-2 | Разработка сетевого графика | 3 | 4 | 3 | 1 | 2 | 0 | 0,2 | 0,04 |
| 2-3 | Подбор литературы и электронных источников информации | 3 | 4 | 3 | 0 | 1 | 1 | 0,2 | 0,04 |
| 2-4 | Поиск существующих решений в данной области | 1 | 2 | 1 | 0 | 2 | 1 | 0,2 | 0,04 |
| 3-5 | Анализ литературы и электронных источников информации (выявление аналогов и эффективных программных решений) | 3 | 5 | 4 | 1 | 2 | 0 | 0,4 | 0,16 |
| 4-5 | Анализ существующих решений | 2 | 4 | 3 | 0 | 2 | 0 | 0,4 | 0,16 |
| 5-6 | Анализ предметной области и бизнес-логики процессов заказчика | 2 | 3 | 2 | 1 | 2 | 0 | 0,2 | 0,04 |
| 6-7 | Выбор стека технологий и протоколов | 1 | 3 | 2 | 1 | 2 | 0 | 0,4 | 0,16 |
| 6-9 | Создание прототипов интерфейса для модуля интеграции с платформой Moodle 2.0 | 2 | 4 | 3 | 0 | 1 | 1 | 0,4 | 0,16 |

Продолжение таблицы 4.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Продолжи-тельность, дн. | | | Исполни-тели, чел. | | | Среднеквадратичное  оклонение, дн. | Дисперсия, дн2 |
| мин. | макс | ожид. | руковод. | инженер | лаборант |
| 7-8 | Разработка описания протокола взаимодействия модуля интеграции и сервисной части программной системы | 5 | 6 | 5 | 0 | 1 | 1 | 0,2 | 0,04 |
| 8-9 | Документирование протокола взаимодействия диаграммами UML | 2 | 3 | 2 | 0 | 1 | 1 | 0,2 | 0,04 |
| 9-10 | Анализ функций пилотной версии программной системы | 2 | 4 | 3 | 1 | 1 | 0 | 0,4 | 0,16 |
| 10-11 | Ранжирование реализуемых пилотной версией функций по затратам на их разработку и интеграцию | 1 | 3 | 2 | 0 | 1 | 1 | 0,4 | 0,16 |
| 11-13 | Выявление эффективных программных решений требуемых функций в существующих аналогах | 1 | 2 | 1 | 0 | 2 | 1 | 0,2 | 0,04 |
| 10-12 | Описание доменных сущностей, составление первой версии диаграммы классов | 2 | 4 | 3 | 0 | 1 | 1 | 0,4 | 0,16 |

Продолжение таблицы 4.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Продолжи-тельность, дн. | | | Исполни-тели, чел. | | | Среднеквадратичное  оклонение, дн. | Дисперсия, дн2 |
| мин. | макс. | ожид. | руковод. | инженер | лаборант |
| 12-13 | Настройка и установка среды разработки и тестирования | 1 | 2 | 1 | 0 | 1 | 1 | 0,2 | 0,04 |
| 13-14 | Разработка пилотной версии ПС | 10 | 13 | 11 | 0 | 2 | 1 | 0,6 | 0,36 |
| 14-16 | Проведение интеграционного тестирования пилотной версии ПС | 2 | 5 | 3 | 0 | 1 | 1 | 0,6 | 0,36 |
| 14-15 | Проведение нагрузочного тестирования пилотной версии ПС | 2 | 4 | 3 | 0 | 1 | 1 | 0,4 | 0,16 |
| 15-16 | Расчет показателей эффективности пилотной версии | 1 | 2 | 1 | 0 | 1 | 1 | 0,2 | 0,04 |
| 16-17 | Встреча с заказчиком, проработка предложений по модернизации пилотной версии | 2 | 3 | 2 | 1 | 2 | 0 | 0,2 | 0,04 |
| 17-18 | Составление финальной версии диаграммы классов | 1 | 3 | 2 | 0 | 2 | 1 | 0,4 | 0,16 |
| 18-19 | Разработка финальной версии ПС | 23 | 25 | 24 | 0 | 2 | 1 | 0,4 | 0,16 |
| 19-21 | Проведение интеграционного тестирования финальной версии ПС | 2 | 5 | 3 | 0 | 1 | 1 | 0,6 | 0,36 |
| 19-20 | Проведение нагрузочного тестирования финальной версии ПС | 2 | 5 | 3 | 0 | 1 | 1 | 0,6 | 0,36 |

Продолжение таблицы 4.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Продолжи-тельность, дн. | | | Исполни-тели, чел. | | | Среднеквадратичное  оклонение, дн. | Дисперсия, дн2 |
| мин. | макс | ожид. | руковод. | инженер | лаборант |
| 20-21 | Расчет показателей эффективности | 1 | 3 | 2 | 0 | 1 | 1 | 0,4 | 0,16 |
| 21-23 | Подготовка пакетов для развертывания системы и миграции данных | 1 | 2 | 1 | 0 | 1 | 1 | 0,2 | 0,04 |
| 21-22 | Составление пользовательской документации | 2 | 3 | 2 | 0 | 1 | 1 | 0,2 | 0,04 |
| 22-23 | Оформление ПЗ | 4 | 5 | 4 | 0 | 2 | 1 | 0,2 | 0,04 |
| 20-21 | Расчет показателей эффективности | 1 | 3 | 2 | 0 | 1 | 1 | 0,4 | 0,16 |
| 21-23 | Подготовка пакетов для развертывания системы и миграции данных | 1 | 2 | 1 | 0 | 1 | 1 | 0,2 | 0,04 |
| 21-22 | Составление пользовательской документации | 2 | 3 | 2 | 0 | 1 | 1 | 0,2 | 0,04 |
| 22-23 | Оформление ПЗ | 4 | 5 | 4 | 0 | 2 | 1 | 0,2 | 0,04 |

Сетевой график, полученный в результате сетевого планирования, представлен ниже на рисунках 4.1-4.3.

|  |  |
| --- | --- |
|  | Рисунок 4.1 – Сетевой график работ. Часть 1 |
|  | Рисунок 4.2 – Сетевой график работ. Часть 2 |
|  | Рисунок 4.3 – Сетевой график работ. Часть 3 |

### Расчёт параметров событий сетевого графика

Ранний срок свершения исходного события СГ принимается равным нулю. Ранний срок свершения данного промежуточного события рассчитывается путем сравнения сумм, состоящих из раннего срока свершения события, непосредственно предшествующего данному. Так как данное событие не может свершиться, пока не закончится последнее из предшествующих ему работ, в качестве раннего срока свершения события принимается большая из сравниваемых сумм.

Рассчитанныйтаким способом ранний срок свершения завершающего события всего СГ принимается в качестве его же позднего срока свершения. Это означает, что завершающее событие СГ никаким резервом времени не располагает.

Поздний срок свершения данного промежуточного события определяется при просмотре СГ в обратном направлении. Для этого сопоставляются разности между поздним сроком свершения события, непосредственно следующего за данным, и продолжительности работы, соединяющей соответствующее событие   
с данным. Так как ни одна из непосредственно следующих за данным событием работ не может начаться, пока не свершится само данное событие, поздний срок свершения равен минимуму из подсчитанных разностей.

Резерв времени образуется у тех событий, для которых поздний срок свершения больше раннего, и он равен их разности. Если же эти сроки равны, событие резервом времени не располагает и, следовательно, лежит на критическом пути.

Период времени, выделенный для дипломного проектирования и преддипломной практики – с 01.03.2011 по 31.05.2011, то есть 93 дня.

Рассчитанные параметры событий сетевого графика ниже (таблица 4.2).

Таблица 4.2 – Параметры событий сетевого графика

В днях

|  |  |  |  |
| --- | --- | --- | --- |
| Номер события | Сроки свершения | | Резерв времени |
| ранний | поздний |
| 0 | 0 | 0 | 0 |
| 1 | 2 | 2 | 0 |
| 2 | 5 | 5 | 0 |
| 3 | 8 | 8 | 0 |
| 4 | 6 | 9 | 3 |
| 5 | 12 | 12 | 0 |
| 6 | 14 | 14 | 0 |
| 7 | 16 | 16 | 0 |
| 8 | 21 | 21 | 0 |
| 9 | 23 | 23 | 0 |
| 10 | 26 | 26 | 0 |
| 11 | 28 | 29 | 1 |
| 12 | 29 | 29 | 0 |
| 13 | 30 | 30 | 0 |
| 14 | 41 | 41 | 0 |
| 15 | 44 | 44 | 0 |
| 16 | 45 | 45 | 0 |
| 17 | 47 | 47 | 0 |
| 18 | 49 | 49 | 0 |
| 19 | 73 | 73 | 0 |
| 20 | 76 | 76 | 0 |
| 21 | 78 | 78 | 0 |
| 22 | 80 | 80 | 0 |

Продолжение таблицы 4.2

В днях

|  |  |  |  |
| --- | --- | --- | --- |
| Номер события | Сроки свершения | | Резерв времени |
| ранний | поздний |
| 23 | 84 | 84 | 0 |
| 24 | 85 | 85 | 0 |

### Расчёт параметров работ сетевого графика

Ранний срок начала работы совпадает с ранним сроком свершения ее начального события.

Поздний срок начала работы получаем путем вычитания из позднего срока свершения ее конечного события ее ожидаемую продолжительность.

Ранний срок окончания работы образуется прибавлением ее продолжи­тельности к раннему сроку свершения ее начального события. Поздний срок окончания работы совпадает с поздним сроком свершения ее конечного события.

Резервами времени обладают работы, не лежащие на критическом пути. Полный резерв времени работы образуется вычитанием из позднего срока свершения ее конечного события, раннего срока свершения ее начального события и ее ожидаемой продолжительности.

Частный резерв времени работы первого рода равен разности поздних сроков свершения ее конечного и начального событий за вычетом ее ожидаемой продолжительности.

Частный резерв времени работы второго рода равен разности ранних сроков свершения ее конечного и начального событий за вычетом ее ожидаемой продолжительности.

Свободный резерв времени работы образуется вычитанием из раннего срока свершения ее конечного события позднего срока свершения ее начального события и ее ожидаемой продолжительности.

Для работ, лежащих на критическом пути, никаких резервов времени нет и, следовательно, коэффициент напряженности (КН) таких работ равен 1. Если работа не лежит на критическом пути, она располагает резервами времени и ее коэффициент напряженности меньше 1. Его величина подсчитывается как отношение суммы продолжительностей отрезков максимального пути, проходящего через данную работу, не совпадающих с критическим путем, к сумме продолжительностей отрезков критического пути, не совпадающих с максимальным путем, проходящим через эту работу.

В зависимости от коэффициента напряженности все работы попадают   
в одну из трех зон напряженности:

* критическую, кнij более 0,8;
* промежуточную, кнij более 0,5 и не более 0,8;
* резервную, кнij менее 0,5.

Результаты расчетов показаны ниже (таблица 4.3).

Таблица 4.3 – Параметры событий сетевого графика

В днях

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Ожидаемая продолжительность | Сроки начала | | Сроки окончания | | Резервы времени | | | | Коэффициент напряженности |
| ранний | поздний | ранний | поздний | полный | частный 1 рода | частный 2 рода | свободный |
| 0-1 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 1,00 |
| 1-2 | 3 | 2 | 2 | 5 | 5 | 0 | 0 | 0 | 0 | 1,00 |
| 2-3 | 3 | 5 | 5 | 8 | 8 | 0 | 0 | 0 | 0 | 1,00 |
| 2-4 | 1 | 5 | 8 | 6 | 9 | 3 | 3 | 0 | 0 | 0,57 |
| 3-5 | 4 | 8 | 8 | 12 | 12 | 0 | 0 | 0 | 0 | 1,00 |
| 4-5 | 3 | 6 | 9 | 9 | 12 | 3 | 0 | 3 | 0 | 0,57 |
| 5-6 | 2 | 12 | 12 | 14 | 14 | 0 | 0 | 0 | 0 | 1,00 |
| 6-7 | 2 | 14 | 14 | 16 | 16 | 0 | 0 | 0 | 0 | 1,00 |
| 6-9 | 3 | 14 | 20 | 17 | 23 | 6 | 6 | 6 | 6 | 0,33 |
| 7-8 | 5 | 16 | 16 | 21 | 21 | 0 | 0 | 0 | 0 | 1,00 |
| 8-9 | 2 | 21 | 21 | 23 | 23 | 0 | 0 | 0 | 0 | 1,00 |
| 9-10 | 3 | 23 | 23 | 26 | 26 | 0 | 0 | 0 | 0 | 1,00 |
| 10-11 | 2 | 26 | 27 | 28 | 29 | 1 | 1 | 0 | 0 | 0,75 |

Продолжение таблицы 4.3

В днях

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код работы | Ожидаемая продолжительность | Сроки начала | | Сроки окончания | | Резервы времени | | | | Коэффициент напряженности |
| ранний | поздний | ранний | поздний | полный | частный 1 рода | частный 2 рода | свободный |
| 11-13 | 1 | 28 | 29 | 29 | 30 | 1 | 0 | 1 | 0 | 0,75 |
| 10-12 | 3 | 26 | 26 | 29 | 29 | 0 | 0 | 0 | 0 | 1,00 |
| 12-13 | 1 | 29 | 29 | 30 | 30 | 0 | 0 | 0 | 0 | 1,00 |
| 13-14 | 11 | 30 | 30 | 41 | 41 | 0 | 0 | 0 | 0 | 1,00 |
| 14-16 | 3 | 41 | 42 | 44 | 45 | 1 | 1 | 1 | 1 | 0,75 |
| 14-15 | 3 | 41 | 41 | 44 | 44 | 0 | 0 | 0 | 0 | 1,00 |
| 15-16 | 1 | 44 | 44 | 45 | 45 | 0 | 0 | 0 | 0 | 1,00 |
| 16-17 | 2 | 45 | 45 | 47 | 47 | 0 | 0 | 0 | 0 | 1,00 |
| 17-18 | 2 | 47 | 47 | 49 | 49 | 0 | 0 | 0 | 0 | 1,00 |
| 18-19 | 9 | 49 | 49 | 73 | 73 | 0 | 0 | 0 | 0 | 1,00 |
| 19-21 | 3 | 73 | 75 | 76 | 78 | 2 | 2 | 2 | 2 | 0,60 |
| 19-20 | 3 | 73 | 73 | 76 | 76 | 0 | 0 | 0 | 0 | 1,00 |
| 20-21 | 2 | 61 | 61 | 78 | 78 | 0 | 0 | 0 | 0 | 1,00 |
| 21-23 | 1 | 63 | 68 | 64 | 69 | 5 | 5 | 5 | 5 | 0,17 |
| 21-22 | 2 | 63 | 63 | 65 | 65 | 0 | 0 | 0 | 0 | 1,00 |
| 22-23 | 4 | 65 | 65 | 69 | 69 | 0 | 0 | 0 | 0 | 1,00 |
| 23-24 | 1 | 69 | 69 | 70 | 70 | 0 | 0 | 0 | 0 | 1,00 |

### Расчёт параметров сетевого графика в целом

Количество событий сетевого графика nС равно 45. Количество работ сетевого графика nР равно 55. Коэффициент сложности kС сетевого графика равен отношению количества работ к количеству событий в сетевом графике по формуле (4) и составляет 1,22.

.

(4)

Критический путь в СГ проходит через события и работы, не обладающие резервами времени и имеет, следовательно, максимальную продолжительность, равную сроку свершения завершающего события: Lкр = 120 дн.

Продолжительность критического пути соответствует математическому ожиданию срока свершения завершающего события, равного сумме ожидаемых продолжительностей работ, составляющих критический путь. Дисперсия срока наступления завершающего события определяется в соответствии с центральной предельной теоремой теории вероятностей как сумма дисперсий работ критического пути, а вероятность свершения завершающего события в срок, равный продолжительности критического пути, равна р(tсв  tкр)=0,5. Если директивный срок установлен меньше продолжительности критического пути, вероятность свершения события к директивному сроку меньше 0,5 и может быть рассчитана с помощью функции распределения нормального отклонения (функции Лапласа) (*Ф*(u)+0,5). Нормальное отклонение "u" равно разности между директивным сроком и продолжительностью критического пути, отнесенной к среднеквадратическому отклонению продолжительности критического пути.

Среднеквадратическое отклонение продолжительности критического пути определяется по формуле (5):

δкр = ,

(5)

где – среднеквадратичное отклонение продолжительности критического пути, дисперсия срока наступления завершающего события, равная сумме дисперсий работ критического пути, дн2.

Вероятность свершения завершающего события в зависимости от директивного срока приведена ниже (таблица 4.4).

Таблица 4.4 – Параметры сетевого графика в целом

|  |  |
| --- | --- |
| Наименование показателя | Значение |
|
| Критический срок, дн. | 85,00 |
| Среднеквадратическое отклонение | 1,57 |
| Директивный срок, дн. | 93,00 |
| Нормальное отклонение | 5,10 |
| Вероятность свершения | 0,99 |

## Смета затрат на проведение работ

Смета затрат на проведение работ включает следующие статьи:

* материалы;
* командировочные расходы;
* контрагентские расходы;
* покупные изделия и специальное оборудование;
* заработная плата;
* начисления на заработную плату;
* накладные расходы.

### Материалы

При разработке программного обеспечения расходы на материалы состоят в основном из канцелярских расходов. Затраты на расходные материалы приведены ниже (таблица 4.5).

Таблица 4.5 –Материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Ед. измерения | Цена за ед., руб. | Количество | Стоимость, руб. |
| Листы А4 | упаковка (500 шт.) | 140,00 | 1 | 140,00 |
| Листы А3 | шт. | 2,00 | 10 | 20,00 |
| Листы А1 | шт. | 10,00 | 8 | 80,00 |
| Ручка шариковая | шт. | 10,00 | 3 | 30,00 |
| Обложка для диплома | шт. | 115,00 | 2 | 230,00 |
| Карандаш простой | шт. | 10,00 | 1 | 10,00 |
| Файлы формата А4 | шт. | 1,50 | 10 | 15,00 |
| CD-R диски | шт. | 20,00 | 2 | 40,00 |
| Итого | | | | 565,00 |

### Командировочные расходы

Поскольку каждому из группы разработки (руководитель, 2 инженера и лаборант) необходимо каждый день ездить на работу и обратно в течение всего срока проектирования, необходимо учесть данные транспортные расходы. Общее количество поездок рассчитывается из продолжительности всех работ в днях   
для каждого сотрудника. Затраты на командировочные расходы приведены ниже (таблица 4.6).

Таблица 4.6 – Командировочные расходы

В рублях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование затрат | Ед. измерения | Цена за единицу | Количество | Сумма |
| Расходы на проезд | 1 поездка | 15,00 | 294 | 4410,00 |
| Итого | | | | 4410,00 |

### Контрагентские расходы

К контрагентским расходам относятся все расходы по оплате услуг сторонним организациям, например, создание рекламы, упаковки, ксерокопирование материалов, печать на плоттере и топу подобное. Аренда машинного времени также относится к этой статье расходов. Аренда рассчитывается исходя из продолжительности всех работ, для которых необходимо применение ЭВМ, и количества работников. Расчет стоимости контрагентских расходов приведен ниже (таблица 4.7).

Таблица 4.7 – Контрагентские расходы

В рублях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование затрат | Единица измерения | Коли-чество | Цена за единицу | Сумма |
| Аренда компьютеров | руб./(комп.\*час) | 20 | 2074,00 | 41 480,00 |
| Печать формата А4 | руб./лист | 1 | 500,00 | 500,00 |
| Печать формата А3 | руб./лист | 2 | 10,00 | 20,00 |
| Вывод графического материала | руб./лист | 50 | 8,00 | 400,00 |
| Итого | | | | 42 400,00 |

### Заработная плата

Рассчитаем заработную плату сотрудников в зависимости от занимаемой должности и количества отработанных дней, для чего найдем стоимость одного дня, равную отношению оклада работника к количеству дней в месяце, примем равным 24 дням (таблица 4.8).

Рассчет суммы, необходимой на социальные отчисления, приведет в таблице 4.9.

Таблица 4.8 – Заработная плата

В рублях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность | Оклад | Среднедневная зарплата | Занятость, дней | Зарплата |
| Руководитель | 35 000 | 1750,00 | 19 | 33 250,00 |
| Инженер | 20 000 | 1000,00 | 162 | 162 000,00 |
| Лаборант | 10 000 | 500,00 | 83 | 41 500,00 |
| Итого | | | | 236 750,00 |
| Премиальный фонд | | | | 23 675,00 |
| Итого (ФОТ) | | | | 260 425,00 |

Таблица 4.9 – Отчисления на социальные нужды

В рублях

|  |  |  |
| --- | --- | --- |
| Наименование отчисления | Процент отчисления | Сумма |
| Социальные отчисления | 34% | 88 544,50 |

### Покупные изделия

Предполагается, что необходимое офисное ПО уже установлена на арендуемых компьютерах. Все используемое для разработки специализированное ПО является бесплатным и свободно-распространяемым. Таким образом, стоимость покупных изделий равна 0 рублей.

### Накладные расходы

Накладные расходы включают в себя все расходы на обслуживание, и составляют 10% от суммы, полученной в пунктах с по (таблица 4.10).

Таблица 4.10 – Накладные расходы

В рублях

|  |  |
| --- | --- |
| Наименование затрат | Сумма |
| Сумма по первым пяти пунктам | 396 344,50 |
| Накладные расходы (10%) | 39 634,45 |

### Смета затрат

Таким образом, после определения накладных расходов легко можем найти итоговую сумму – себестоимость проведенной работы (таблица 4.11).

Таблица 4.11 – Смета затрат

В рублях

|  |  |
| --- | --- |
| Наименование статьи затрат | Сумма |
| Материалы | 565,00 |
| Командировочные расходы | 4 410,00 |
| Контрагентские расходы | 42 400,00 |
| Покупные изделия | 0,00 |
| Заработная плата | 348 969,50 |
| Накладные расходы | 39 634,45 |
| Итого | 395 890,00 |

## Анализ технико-экономической эффективности

В разделе концепции дипломного проекта был представлен подробный обзор существующих решений для организации процесса непрерывной интеграции, сборки проектов и запуска модульных тестов. В этом пункте рассмотрим технико-экономическую эффективность нашей разработки и существующих аналогов.

На сегодняшний день существует две большие группы решений – со свободно распространяемым исходным кодом и являющихся собственническим ПО.

В первую группу выходят системы непрерывной интеграции: CruiseControl, Apache Continuum, Jenkins – прикладные программы, разработанные с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) следить и поддерживать проекты в состоянии постоянной стабильной работы, путем их непрерывной сборки и запуском модульных тестов в автоматическом режиме при наличии изменений в исходном коде проекта.

Все перечисленные выше средства выполняют только часть функций, требуемых заказчиком, например, сборка или запуск модульных тестов, что ограничивает их самостоятельное использование и требует дополнительных программных средств для полноценной работы. При этом в некоторых случаях добавление требуемых функций (как то интеграция с системой поддержки образовательного процесса Moodle и др.) трудноосуществимо. Наше решение представляет программную систему "под ключ", которая позволяет после непродолжительной настройки поддерживать процессы получения кода, сборки, запуска тестов и формирования отчетов, предоставляет удобные гибкие средства конфигурирования как всей системы в целом, так и каждого средства в отдельности.

К продуктам второй группы (имеющим проприетарную лицензию), относятся такие системы как:

* средства IBM Rational автоматизации процессов жизненного цикла ПО.
* Team Foundation Server – продукт корпорации Microsoft, представляющий собой комплексное решение, объединяющее в себе систему управления версиями, сбор данных, построение отчетов, отслеживание статусов и изменений по проекту, и предназначенное для совместной работы над проектами по разработке программного обеспечения.
* TeamCity – разработка компании JetBrains, представляющая собой централизованную систему управления построением приложений и непрерывной интеграции, для организации эффективной коллективной работы над кодом приложений.

Даже примерную общую стоимость коммерческих продуктов оценить сложно, поскольку нет единой методики определения цены для заказчика. Все системы подразумевают не только покупку лицензии и установку ПО, но и его последующую настройку, адаптацию под требования заказчика, которая значительно осложняется закрытостью исходных кодов этих систем.

Для примера приведем стоимость решений от IBM Rational по прайс-листам (на 28 апреля 2011 года) компании «Интерфейс» (таблица 4.12) [], которая является официальным партнером компании IBM в России.

Таблица 4.12 – Стоимость коммерческих аналогов

В рублях

|  |  |
| --- | --- |
| Наименование продукта | Стоимость |
| IBM Rational Suite | 332 996 |
| IBM Rational Team Unifying Platform | 142 534 |

При этом все проприетарные решения данные решения не включают в себя ни поддержку необходимых заказчику библиотек модульного тестирования, ни возможность интеграции с платформой Moodle, что приводит к необходимости прибегать с обходным решениям при их адаптации к конкретным нуждам. Кроме того разрабатываемая система изначально ориентирована на применение в учебном процессе для решения основной бизнес-проблемы заказчика – необходимость в эффективном по времени и надежном способе оценки студенческих работ. Все это делает разработку такой системы в некотором роде уникальной.

Использование бесплатного ПО при разработке системы значительно снижает стоимость создаваемой системы, делает ее разработку экономически выгодной. Так основной статьей расходов является заработная плата работников, следовательно, стоимость разработки фактически зависит только от размеров окладов работников, и становится очевидной невыгодность заказа разработки такой системы сторонним организациям.

# ЗАКЛЮЧЕНИЕ

В ходе дипломного проектирования было проведено анкетирование заказчика, выполнен анализ бизнес-проблем и разработана концепция проекта, включающая в себя перечень требований к разрабатываемой системе.

Были рассмотрены существующие программные решения и выбраны   
технологии реализации проектного решения.

Также разработан интуитивно понятный пользовательский веб-интерфейс, позволяющий управлять сущностями и конфигурациями конвейера получения исходных кодов студенческих работ, их сборкой, тестированием и созданием отчетов. Разработанная компонентная модель организации конвейера позволяет гибко настраивать весь процесс под конкретные нужды заказчика и минимизирует возможные будущие затраты на изменение функциональности системы. Была выполнена интеграция системы с системой управления учебными курсами   
Moodle 2.0, используя технологию веб-сервисов, для получения исходных кодов работ студентов из закачиваемых ими решений к заданиям, создаваемым преподавателем. Для проектного решения была спроектирована и реализована серверная инфраструктура, на которой и была развернута система.

В результате разработанная система была внедрена на кафедре ЭВМ ЮУрГУ и успешно используется в учебном процессе для проверки и выставления оценок за семестровые и контрольные работы, решения к рейтинговым задачам.

Пояснительная записка отражает все этапы разработки программного комплекса, начиная от концепции проекта и заканчивая планом пилотного внедрения. Кроме того, в пояснительной записке приведено руководство оператора, описаны рекомендации по организации рабочего места пользователя и режима труда и отдыха, а также приведено сетевое планирование и технико-экономическое обоснование разработки.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

##### Цытович, П.Л. Программная инженерия: учебное пособие. – Челябинск: Изд-во ЮУрГУ, 2007. – Ч. 1. – 106 с.

##### Кролл, П. Rational Unified Process – это легко. Руководство по RUP / П. Кролл, Ф. Кратчен; пер. с англ. С. Лунин. – М.: КУДИЦ-ОБРАЗ, 2004. – 432 с.

##### Википедия. Непрерывная интеграция. – http://ru.wikipedia.org/wiki/ Непрерывная\_интеграция.

##### Википедия. Hudson. – http://ru.wikipedia.org/wiki/ Hudson.

##### Википедия. WEBrick. – http://ru.wikipedia.org/wiki/WEBrick.

##### Википедия. Sinatra. – http://ru.wikipedia.org/wiki/Sinatra.

##### Википедия. YAML. – http://ru.wikipedia.org/wiki/YAML.

##### Википедия. AJAX. – http://ru.wikipedia.org/wiki/AJAX.

##### Википедия. jQuery. – http://ru.wikipedia.org/wiki/Jquery.

##### Википедия. jQuery UI. – http://ru.wikipedia.org/wiki/JQuery\_UI.

##### Википедия. HAML. – http://ru.wikipedia.org/wiki/Haml.

##### Википедия. SQLite. – http://ru.wikipedia.org/wiki/SQLite.

##### Википедия. Moodle. – http://ru.wikipedia.org/wiki/Moodle.

##### Википедия. Микроядро. – http://ru.wikipedia.org/wiki/Микроядро.

##### Википедия. REST. – http://ru.wikipedia.org/wiki/REST.

##### Википедия. Технология единого входа. – http://ru.wikiepedia.org/wiki/ Технология\_единого\_входа.

##### Шафер, Ф. Управление программными проектами. Достижение оптимального качества при минимуме затрат: Пер. с англ. / Ф. Шафер, Р. Фатрелл. – М.: Издательский дом «Вильямс», 2003. – 1136 с.

##### ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению. – М.: Изд-во стандартов, 1980. – 2 с.

##### Леффингуэлл, Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уиндриг; пер. с англ. Н. Орехова. – М.: Издательский дом «Вильямс», 2002. – 448 с.

##### Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер; пер. с англ. Н. Орехова. – М.: Издательский дом «Вильямс», 2007. – 544 с.

##### Буч, Г. Объектно-ориентированное проектирование с примерами применения / Г. Буч; пер. с англ. Д. Клюшин. – М.: Конкорд, 1992.

##### Буч, Г. Язык UML: Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; пер. с англ. Н. Мухин. – М.: ДМК, 2000.

##### Кратчен, Ф. Введение в Rational Unified Process. 2-е изд. / Ф. Кратчен; пер. с англ. С. Лунин. – М.: Издательский дом «Вильямс», 2002. – 240 с.

##### СанПин 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрава России, 2003. – 35 с.

##### СанПин 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрава России, 1996. – 38 с.

##### ГН 2.1.6.1338-03. Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест. – М.: Информационно-издательский центр Минздрава России, 2003. – 30 с.

##### СанПиН 2.2.4.1294-03. Санитарно-гигиенические нормы допустимых уровней ионизации воздуха. – М.: Информационно-издательский центр Минздрава России, 2003. – 41 с.

##### СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. – М.: Информационно-издательский центр Минздрава России, 1996. – 34 с.

##### СанПиН 2.2.4/2.1.8.566-96. Производственная вибрация. Вибрация в помещениях жилых и общественных зданий. – М.: Информационно-издательский центр Минздрава России, 1996. – 27 с.

##### ГОСТ 12.1.019-79. Электробезопасность. Общие требования и номенклатура видов защиты. – М.: Изд-во стандартов, 1979. – 29 с.

##### ГОСТ 12.1.030-81. Электробезопасность. Защитное заземление. Зануление. – М.: Изд-во стандартов, 1991. – 35 с.

##### ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования. – М.: Изд-во стандартов, 1991. – 28 с.

##### Методические указания по дипломному проектированию для студентов приборостроительного факультета. (Раздел «Безопасность жизнедеятельности») / сост. Н. В. Глотова, И. С. Окраинская; под ред. А. И. Сидорова. – Челябинск: Изд-во ЮУрГУ, 1999. – 16 с.

##### Зинкевич, В.С. Сетевые методы планирования и управления: Методические указания к курсовому проекту для студентов приборостроительного факультета / сост. В.С. Зинкевич, Л.А. Баев, Н.П. Мешковой. – Челябинск: Изд-во ЮУрГУ, 1998. – 22 с.

##### Сайт компании «Интерфейс». – http://www.interface.ru/home.asp.

##### Wikipedia. TeamCity. – http://en.wikipedia.org/wiki/TeamCity.

##### Wikipedia. CruiseControl. – http://en.wikipedia.org/wiki/CruiseControl.

##### Moodle Docs. External services description. – http://docs.moodle.org/dev/External\_services\_description.

# ПРИЛОЖЕНИЯ

# ПРИЛОЖЕНИЕ А Пример файла конфигурации системы сборки и тестирования программного обеспечения

Листинг А.1 – Пример файла конфигурации системы сборки и тестирования программного обеспечения

---

environment: "production"

web:

enable: yes

display\_warnings: yes

publish: no

security:

user:

login: "admin"

password: "admin"

email:

smtp:

host: "127.0.0.1"

port: 25

tempdir: "temp"

logging:

web: yes

system: yes

path: "logs"

filename: "log\_%starttime"

console: yes

Окончание приложения А

Продолжение листинга А.1

developer\_email: muyou.prj@gmail.com

admin\_email: "example@example.com"

local\_codepage: "CP1251"

components:

VisualStudio10Builder:

lib\_path: "C:/Users/Eskat0n/Lib/UnitTest++/UnitTest++/Debug"

include\_path: "C:/Users/Eskat0n/Lib/UnitTest++/UnitTest++/src"

lib: "UnitTest++.vsnet2010.lib"

HTMLTableReportBuilder:

output: "reports"

MoodleSourceProvider:

host: "http://localhost/moodle"

token: "3cf6d1bac901bedd2e11752227519784"

szip\_path: "C:/Program Files/7-Zip/7z.exe"

# ПРИЛОЖЕНИЕ Б Пример файла конфигурации поведения веб-интерфейса

Листинг Б.1 – Пример файла конфигурации поведения веб-интерфейса

# coding: utf-8

#

# Overall configuration for Sledgehammer

COMPUTER\_NAME = ENV['COMPUTERNAME']

INSTANCE\_NAME = "Sledgehammer на #{COMPUTER\_NAME}"

ENVIRONMENT = SHR::Config.environment

# Sinatra's configuration for Sledgehammer web interface

set :environment, ENVIRONMENT

enable :sessions

disable :run

set :display\_warnings, SHR::Config.display\_warnings?

set :logging, (SHR::Config.log\_web? and SHR::Config.log\_console?)

disable :dump\_errors if settings.environment == :production

set :root, File.dirname(\_\_FILE\_\_)

set :public, ->() { File.join(root, "static") }

set :views, ->() { File.join(root, "views") }

set :server, :webrick

Окончание приложения Б

Продолжение листинга Б.1

set :bind, (SHR::Config.publish? ? '0.0.0.0' : 'localhost')

set :port, 9009

# Security configuration

use Rack::Auth::Basic, 'Sledgehammer' do |username, password|

user = SHR::Config.user

[username, password] == [user['login'], user['password']]

end unless settings.environment == :development

require\_relative 'config.db.rb'