

<b>Содержание/Оглавление</b>	
<b>Реферат</b> .....	4
<b>Термины и сокращения</b> .....	5
<b>Введение</b> .....	6
<b>1. Предпроектное исследование</b> .....	7
1.1. Основные этапы жизненного цикла программного продукта .....	7
1.2. Группы CASE-средств .....	7
1.3. Постановка задачи.....	15
<b>2. Техническое задание на ALM систему ДРПО ЗАО «Элвис - Неотек».</b> .....	17
2.1. Общие сведения.....	17
2.1.1. Полное наименование работы и ее условное обозначение. ....	17
2.1.2. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы .....	17
2.2. Назначение и цели «Внедрения ALM Системы» .....	17
2.2.1. Назначение «Внедрения ALM Системы» .....	17
2.2.2. Цели и эффекты выполнения работы .....	17
2.2.3. Общие принципы «Внедрения ALM Системы».....	18
2.3. Характеристика объектов автоматизации .....	18
2.3.1. Объекты автоматизации .....	18
2.3.2. Общее описание автоматизируемых сервисов .....	19
2.4. Требования к системе .....	21
2.4.1. Требования к сервисам .....	21
2.4.2. Требования к этапам «Внедрения ALM Системы» .....	22
2.5. Состав и содержание работ по созданию «ALM Системы» .....	23
2.5.1. Перечень стадий и этапов работ по созданию системы .....	23
2.6. Порядок контроля и приемки подсистемы.....	24
2.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу подсистемы в действие .....	25
2.8. Требования к документированию.....	26
2.9. Другие требования к «Внедрению ALM Системы».....	27
<b>3. Концептуальный этап проектирования</b> .....	28
3.1. Анализ существующих программных продуктов и систем.....	28
3.2. Выбор языка программирования .....	34
3.3. Основные принципы разработки .....	35
<b>4. Технический этап проектирования</b> .....	36
4.1. Выбор архитектуры системы .....	37
4.2. Выбор платформы виртуализации .....	38
4.3. Планирование уровня данных .....	39
4.4. Планирование уровня приложений .....	39
4.5. Планирование клиентского уровня .....	40
4.5.1. Подсистема вывода информации.....	40
4.5.2. Подсистема выполнения сборки приложений .....	41
4.6. Расчет необходимого аппаратного обеспечения ALM Системы .....	41
4.7. Выбор аппаратного обеспечения ALM Системы .....	43
4.8. Проектирование развертывания ALM Системы .....	45
4.9. Обеспечение отказоустойчивости системы .....	46

4.9.2.	Проектирование подсистемы резервного копирования .....	48
4.10.	Планирование работ по адаптации TFS к процессу ДРПО .....	48
5.	Исследовательская часть .....	50
5.1.	Постановка задачи .....	50
5.2.	Методика исследования .....	50
5.3.	Результат .....	50
6.	Рабочее проектирование .....	52
6.1.	Переработка шаблона проекта .....	52
6.1.1.	Элемент Bug .....	53
6.1.2.	Элемент Task .....	53
6.1.3.	Иерархия элементов .....	53
6.2.	Модификация пользовательского интерфейса .....	54
6.3.	Создание пользовательского обработчика событий .....	54
7.	Организационно-экономическая часть .....	56
7.1.	Введение .....	56
7.2.	Оценка трудоемкости и сроков разработки модулей интеграции .....	57
7.2.1.	Стадии разработки и состав работ .....	58
7.2.2.	Расчет трудоемкости разработки программного продукта .....	60
7.3.	Расчет себестоимости модулей интеграции .....	69
7.4.	Расчет себестоимости приобретаемой части ALM системы .....	73
7.5.	Заключение организационно-экономической части .....	74
8.	Требования безопасности при разработке модулей интеграции ALM Системы .....	76
8.1.	Введение .....	76
8.2.	Требования безопасности при отработке программного продукта .....	77
8.2.1.	Требования к персональным электронно-вычислительным машинам .....	77
8.2.2.	Требования к помещению для работы с ПЭВМ .....	79
8.2.3.	Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ .....	82
8.2.4.	Требования к освещению на рабочих местах, оборудованных ПЭВМ .....	83
8.2.5.	Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ .....	86
8.2.7.	Общие требования к организации рабочих мест пользователей ПЭВМ .....	86
8.2.8.	Требования к организации и оборудованию рабочих мест с ПЭВМ для обучающихся в общеобразовательных учреждениях и учреждениях начального и высшего профессионального образования .....	88
8.2.9.	Требования к электроснабжению, электробезопасности на рабочих местах, оборудованных ПЭВМ .....	90
8.2.10.	Пожаробезопасность .....	91
8.3.	Типовой расчет виброизоляции для системы кондиционирования .....	92
8.4.	Утилизация носителей информации .....	93
9.	Заключение .....	97
10.	Список используемой литературы .....	98
11.	Приложения .....	99

11.1.	Регламент работы с требованиями до внедрения ALM Системы.....	99
11.2.	Регламент работы с сущностью “Bug” .....	101
11.3.	Листинг программ подсистемы резервного копирования.....	106
11.3.1.	Модуль Backup.ps1 .....	106
11.3.2.	Модуль VMBackup.ps1 .....	109
11.3.3.	Модуль DBBackupTransfer.ps1 .....	110
11.4.	Описание рабочего элемента «Bug».....	111
11.5.	Описание рабочего элемента «Task» .....	122
11.6.	Описание компонента CommonProcessConfig .....	130
11.7.	Модуль визуализации ID .....	131
11.8.	Модуль WorkItemChangedEventHandler .....	131

## Реферат

Отчет \_\_\_\_ с., \_\_\_\_ рис., \_\_\_\_ табл., \_\_\_\_ источников, \_\_\_\_ прил.

**Перечень ключевых слов:** *жизненный цикл, разработка, тестирование, стадии разработки, бизнес-процесс, автоматизация, поддержка, внедрение.*

В работе рассмотрена архитектура системы управления жизненным циклом программного обеспечения. Разработана и реализована архитектура модулей интеграции системы в бизнес-процесс предприятия. В системе доработаны алгоритмы и состояния рабочих элементов в соответствии с принятым процессом на предприятии.

Для разработки подсистем и внедрения системы использовались языки платформы Microsoft .NET и расширяемый язык разметки XML.

Система прошла тестирование и успешно внедрена в Департаменте разработки программного обеспечения ЗАО «Элвис-Неотек». Это позволило повысить эффективность работы сотрудников департамента и увеличить качество выпускаемого ПО.

## **Термины и сокращения**

**ALM Система** - система комплексной поддержки и управления жизненным циклом программных продуктов.

**ЗАО «Элвис-Неотек»** — Компания, заказчик разрабатываемой ALM системы.

**Департамент Разработки Программного Обеспечения (ДРПО)** - структурное подразделение Компании, где будет внедряться система.

**Итерация** — фиксированный временной промежуток, по окончании которого команда разработчиков способна предоставить готовое решение поставленных задач. В рамках итерации список задач не может быть изменен.

**Product Owner**

**Triage-группа**

**Milestone**

**PM (Product Manager)** – менеджер проектов.

**API**

**TFS**

**SharePoint**

**MSDN**

**OC**

**IIS**

## **Введение**

В современном мире разработка ПО состоит не только из написания исходных кодов и дизайна пользовательских интерфейсов, но и из иных процессов, им предшествующих и за ними следующих, и это известно уже очень давно. Много лет существуют и программные продукты, предназначенные для поддержки и автоматизации данных процессов. Однако полноценное и целенаправленное применение этих систем в подавляющем большинстве проектов, связанных с разработкой приложений, началось относительно недавно — еще три-пять лет назад, из всех возможных инструментов создания ПО, самыми популярными были интегрированные средства разработки приложений (IDE) и средства проектирования баз данных.

В наше время ситуация резко меняется — начинается активное внедрение средств управления и поддержки жизненного цикла приложений (Application Lifecycle Management - ALM), и происходит это в компаниях и в отделах разработки различных размеров и для весьма разнообразных проектов. Нередко внедрение идет стихийно, начавшись с открытого средства контроля версий или с простейшего средства генерации документации, но бывает и так, что в качестве корпоративного стандарта принимается линейка продуктов какого-то одного производителя в расчете на более тесную взаимную интеграцию продуктов для поддержки различных этапов жизненного цикла приложений.

Одной из таких компаний является компания ЗАО «ЭЛВИС-НеоТек». В связи с требованиями по повышению качества выпускаемых программных продуктов, улучшению поддержки пользователей выпускаемого ПО, повышению эффективности работы разработчиков, появилась задача по внедрению интегрированной системы управления и поддержки жизненного цикла ПО.

# **1. Предпроектное исследование**

## **1.1. Основные этапы жизненного цикла программного продукта**

Что бы ни представляла собой методология разработки при реализации какого-либо проекта, в нем всегда в различном виде существуют такие этапы, как определение требований, проектирование приложений и данных, создание приложений, их тестирование и внедрение и поддержка, которые, в зависимости от применяемой методологии, могут повторяться один или несколько раз. Кроме того, редкий проект обходится без документирования созданного продукта.

Стоимость различных этапов жизненного цикла может значительно различаться для разных типов проектов и различных методологий. Например, в случае коробочного продукта наибольшие затраты приходятся на тестирование продукта в условиях применения разнообразных конфигураций аппаратного и программного обеспечения, тогда как в случае нестандартного заказного проекта основная часть средств расходуется на управление требованиями.

## **1.2. Группы CASE-средств**

Ниже приведены группы средств, применяемых на различных этапах разработки программного продукта(CASE-средства):

- Средства управления требованиями
- Средства моделирования бизнес-процессов, приложений и данных
- Средства разработки приложений
- Средства тестирования и оптимизации приложений
- Средства управления коллективной работой и контроля версий
- Средства внедрения приложений

Управление требованиями — одна из самых важных составных частей процесса разработки. Полнота, точность и корректность формулировки того, что именно нужно заказчику проекта, являются залогом успеха проекта и

минимизируют работы, связанные с исправлениями уже готового или почти готового продукта.

В ряде стран документ, содержащий формулировки требований, служит обязательным приложением к договору на разработку продукта. В зависимости от методологии выполнения проекта, формулировка требований или их модификация может производиться однократно или многократно. Иногда требования формулируются в виде текстового документа, например технического задания. Однако в последнее время все большую популярность приобретают средства автоматизации управления требованиями, позволяющие не только генерировать документ согласно стандартам, принятым в той или иной стране, но и автоматизировать реализацию требований на уровне модели и даже кода приложения.

Из наиболее часто применяющихся в нашей стране средств управления требованиями следует назвать **Rational Requisite Pro** (IBM, [www.ibm.com](http://www.ibm.com)) и **Borland CaliberRM** (Borland, [www.borland.com](http://www.borland.com)). Первый из этих продуктов довольно давно продвигается в России несколькими наиболее активными партнерами IBM.

Что же умеют делать все современные средства управления требованиями? Как правило, они позволяют классифицировать требования по категориям и подкатегориям, осуществлять их детализацию и генерировать различные документы, содержащие требования к продукту, в соответствии со стандартами, которые приняты в той или иной стране (иногда для реализации подобной возможности требуется создать соответствующие шаблоны). Однако основная функциональность, ради которой приобретаются указанные средства, — это возможность интеграции с инструментами, предназначенными для других этапов жизненного цикла приложений. Так, Borland CaliberRM интегрируется со многими средствами разработки Borland и Microsoft, вплоть до встраивания списка требований в среду разработки и генерации заготовок кода при переносе пиктограммы требования мышью в редактор кода, а Rational Requisite Pro — со средством генерации проектной



документации Rational SoDA и со средством объектно-ориентированного моделирования Rational Rose (в частности, этот продукт позволяет генерировать требования на основе прецедентов, описанных в модели).

Моделирование и проектирование данных и приложений основывается на сформулированных требованиях и является весьма важной частью процесса создания готового продукта. Инструменты для поддержки данного этапа жизненного цикла приложений можно условно разделить на средства моделирования бизнес-процессов, средства проектирования данных и средства объектно-ориентированного моделирования. Отметим, однако, что сегодня многие компании производят все три категории инструментов, интегрирующихся между собой (например, позволяющих сгенерировать модель данных и модели бизнес-процессов или синхронизировать их между собой), либо реализуют функциональность нескольких разнотипных средств моделирования в одном продукте, поэтому применительно к указанной категории инструментов имеет смысл говорить о линейках продуктов различных производителей.

Ниже перечислены наиболее известные на российском и мировом рынках продукты и линейки продуктов, предназначенных для моделирования и проектирования.

AllFusion Modelling Suite (Computer Associates, [www.cai.com](http://www.cai.com)) состоит из следующих продуктов:

- AllFusion Process Modeler — основной инструмент для визуального моделирования бизнес-процессов;
- AllFusion ERwin Data Modeler — средство проектирования и документирования баз данных, хранилищ данных и витрин данных;
- AllFusion Data Model Validator — инструмент для проверки структуры баз данных и создаваемых моделей, позволяющий выявлять недочеты и ошибки проектирования;
- AllFusion Model Manager — средство обеспечения для совместной

работы группы проектировщиков над одним проектом с помощью AllFusion ERwin Data Modeler и/или AllFusion Process Modeler;

- AllFusion Component Modeler — средство объектно-ориентированного проектирования приложений, интегрирующееся с AllFusion Process Modeler.

В России эта линейка продуктов весьма популярна, а AllFusion ERwin Data Modeler является одним из самых распространенных средств проектирования данных, главным образом благодаря поддержке широкого спектра СУБД.

Oracle Designer представляет собой инструмент, позволяющий проектировать данные, моделировать бизнес-процессы, создавать диаграммы потоков данных и функциональные модели, а также реализовывать их в виде серверных объектов. Этот продукт главным образом предназначен для применения совместно с СУБД Oracle и поддерживает все особенности данной СУБД, хотя с его помощью можно осуществлять и обратное проектирование для СУБД других производителей.

Sybase PowerDesigner — это инструмент, в состав которого входят средства создания моделей и объектно-ориентированного моделирования. Помимо серверных СУБД производства Sybase, PowerDesigner способен работать с любыми ODBC-источниками, генерировать код клиентских приложений для PowerBuilder, а также классы Java и компоненты JavaBeans. Возможно и обратное проектирование диаграмм классов из исходных текстов Java. Набор функциональных возможностей продукта варьируется в зависимости от редакции. Отметим, что PowerDesigner весьма популярен на российском рынке, и отнюдь не только среди пользователей СУБД и средств разработки Sybase.

Visio (Microsoft, [www.microsoft.com](http://www.microsoft.com)) представляет собой универсальное средство моделирования данных и приложений, поддерживающее и создание моделей данных, и объектно-ориентированное моделирование приложений. Как и подавляющее большинство средств проектирования данных, Visio позволяет производить прямое и обратное проектирование данных,

поддерживает все ODBC- и OLE DB-источники данных и особенности серверных СУБД всех ведущих производителей. Помимо средств проектирования данных, Visio включает средства объектно-ориентированного моделирования и генерации кода приложений (главным образом для платформы Microsoft .NET).

Rational Rose (IBM) — одно из самых популярных средств объектно-ориентированного UML-моделирования приложений. Данный продукт позволяет решать практически любые задачи в проектировании информационных систем — от анализа бизнес-процессов и моделирования данных до генерации кода на различных языках программирования, а также обладает средствами интеграции с другими инструментами Rational, в частности с Requisite Pro.

Rational XDE Professional (IBM) — инструмент UML-моделирования, встраиваемый в среды разработки Microsoft Visual Studio .NET и IBM WebSphere Studio Application Developer. Этот продукт дает возможность осуществлять визуальное проектирование на основе диаграмм UML и по окончании процесса проектирования генерировать код на выбранном языке программирования, а также проводить двустороннюю синхронизацию кода и модели.

SparxSystems Enterprise Architect — инструмент, основное назначение которого - предоставить возможность моделирования и проектирования приложений при помощи UML-диаграмм. Однако, поскольку, метамодель заложенная в основу UML позволяет описать практически любые модели, используемые в разработке ПО, то ЕА расширяем и предоставляет возможности моделирования бизнес-процессов, баз данных и т.п.

Из вышесказанного следует, что основной тенденцией развития средств моделирования в настоящее время является активное предложение средств интеграции их между собой и с инструментами поддержки других этапов жизненного цикла приложений, как-то: средств разработки, средств управления требованиями и средств управления изменениями.

Происходившее в последние годы слияние компаний, специализирующихся на производстве подобных инструментов, позволило создать линейки продуктов, в своей совокупности реализующих все или почти все задачи, которые могут возникнуть на этапе проектирования приложений.

Средства разработки приложений обычно классифицируют на основе их платформы, для которой они предназначены. В соответствии с этой классификацией можно выделить следующие категории средств разработки: средства создания Java-приложений, средства создания нативных Windows-приложений (использующих Win32 API ), средства создания .NET-приложений (использующих Microsoft .NET фреймворк), инструменты создания приложений для операционных систем, применяющихся в мобильных устройствах, средства создания приложений для различных версий UNIX/Linux и других платформ.

В ряду самых важных составляющих современных средств разработки приложений необходимо выделить средства повышения производительности труда разработчиков, средства поддержки коллективной работы, возможности, связанные с повторным использованием моделей и кода, поддержку средств и стандартов создания распределенных приложений и их интеграции, поддержку разработки приложений для мобильных устройств и, разумеется, средства интеграции с инструментами поддержки других этапов жизненного цикла приложений (в частности, со средствами проектирования и моделирования приложений и данных, управления требованиями, тестирования и оптимизации).

Тестирование — абсолютно необходимая составляющая процесса создания качественного программного обеспечения. Именно на этом этапе проверяется, удовлетворяет ли приложение сформулированным требованиям, а затем в продукт вносятся изменения, устраняющие выявленные при тестировании недостатки.

Средства тестирования приложений можно с некоторой долей условности разделить на несколько категорий: средства управления

процессом тестирования и генерации планов тестирования, средства управления найденными дефектами (Bug tracking), средства тестирования баз данных, средства тестирования пользовательского интерфейса, средства тестирования производительности, средства тестирования безопасности приложений, средства регрессионного тестирования (создающие испытательные сценарии на основе записанных действий пользователя и затем многократно их повторяющие), средства непрерывной интеграции и ряд других категорий.

Отдельно хотелось бы отметить средства управления тестовыми средами. Ввиду все возрастающего количества различных программных платформ, возникает необходимость тестирования разрабатываемого ПО на весьма большом множестве сред, и зачастую на подготовку и разворачивание тестовой среды уходит большая часть ресурсов, предназначенных для тестирования приложений. Средства управления тестовыми средами позволяют автоматизировать большинство процессов, связанных с развёртыванием и повторным использованием тестовых сред(создание шаблонов сред, установка ОС, резервное копирование, удаление, простая модификация).

Так же как и для средств моделирования, основной тенденцией здесь является тесная интеграция со средствами разработки и управления требованиями.

В данной категории достаточно прочные позиции занимают инструменты с открытым исходным кодом.

Средства управления коллективной работой применяются на различных этапах создания приложений — начиная с формулировки требований и заканчивая внедрением готового продукта. Условно их можно разделить на средства управления задачами и проектами, средства контроля версий кода, моделей и документов, средства организации документооборота внутри команды разработчиков.

Из средств контроля версий самыми распространенными считаются и

Microsoft Visual SourceSafe, и его приемник — Microsoft Team Foundation Server, однако именно среди этой категории продуктов немалую популярность имеют и средства категории Open Source: Subversion, Git, Mercurial, Bazaar.

Среди средств управления проектами в первую очередь нужно отметить Microsoft Project Professional и Microsoft Project Server, которые являются сейчас безусловными лидерами рынка средств управления проектами по разработке и внедрению ПО, а также продукты компании SAP и IBM.

Современные средства управления коллективной работой и контроля версий характеризуются, как правило, наличием централизованного репозитория (нередко хранящего не только сведения о проекте и его составных частях, но и сами составные части проекта — код, шаблоны и схемы, медиа ресурсы, модели), средства поддержки репозитариев других производителей, а также наличие средств интеграции со средствами поддержки других этапов жизненного цикла приложений.

К средствам внедрения приложений относятся инструменты для создания дистрибутивов и систем доставки приложения конечным пользователям (web-сервисы, позволяющие установить приложение на конечное устройство без участия физического носителя информации — дисков, флэш памяти и др.)

Из компаний, лидирующих на рынке средств создания дистрибутивов приложений, следует упомянуть InstallShield, Wise, Microsoft, Oracle. Продуктами этих компаний пользуются многие крупнейшие производители программного обеспечения, а специализированные (либо облегченные) версии их средств создания инсталляционных приложений нередко включаются в состав средств разработки приложений.

Итак, можно сделать вывод, что рынок инструментов создания приложений отнюдь не ограничивается собственно средствами разработки. Более того, они перестают играть доминирующую роль в процессе реализации проектов, связанных с разработкой приложений. Объясняется это

прежде всего появлением новых технологий создания приложений, выдвигающих на первый план моделирование и проектирование приложений и определение поведения приложений непосредственно в моделях, а также с развитием компонентно-ориентированного подхода, позволяющего реализовывать приложения на основе готовых компонентов и бизнес-объектов при минимальном объеме кодирования.

### **1.3. Постановка задачи.**

Проблемы:

1. В Департаменте Разработки Программного Обеспечения ЗАО «Элвис-Неотек» имеется разрозненный набор программно-аппаратных средств поддержки создания программных продуктов.
2. Существует не стандартизированный, но устоявшийся бизнес-процесс разработки ПО, близкий к методологии Agile.
3. Персонал компании тратит большое количество ресурсов на интеграцию средств поддержки разработки ПО между собой для того, чтобы они поддерживали существующий бизнес-процесс.
4. Сильно менять процесс нет возможности.

Требуется:

Спроектировать ALM систему Компании и спланировать ее внедрение, интегрирующее ее в существующий бизнес-процесс.

Особенности:

- Компания занимается разработкой программных систем, в основе которых лежат программные решения корпорации Microsoft. Т.е. для работы данных систем требуется определенный набор ПО, выпущенного Microsoft.
- Компания является золотым партнером Microsoft, что позволяет ей использовать множество программных решений корпорации бесплатно в соответствии с условиями партнерского соглашения.
- Поддержкой и дальнейшим развитием ALM системы должны заниматься сотрудники ДРПО.

## 1.4. Диаграмма целей

На основании поставленной задачи была сформирована диаграмма целей(рис.1.1)

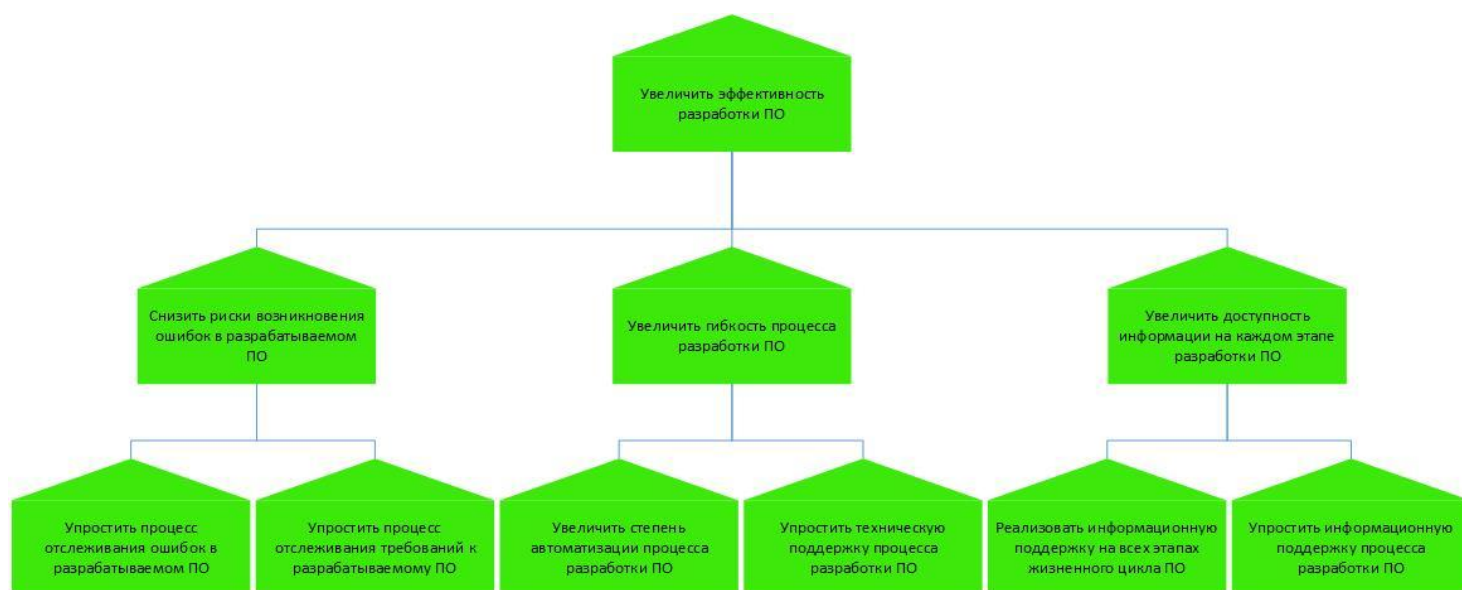


Рис. 1.1



## **2. Техническое задание на ALM систему ДРПО ЗАО «Элвис - Неотек».**

### **2.1. Общие сведения.**

#### **2.1.1. Полное наименование работы и ее условное обозначение.**

Полное наименование работы: Проектирование и внедрение системы комплексной поддержки и управления жизненным циклом программных продуктов Департамента Разработки Программного Обеспечения ЗАО «Элвис-Неотек» с использованием программных продуктов Microsoft и интеграция данной системы с существующими с бизнес-процессами предприятия.

Условное обозначение работы: «Внедрение ALM Системы»

#### **2.1.2. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы**

Разработчик:

ДРПО ЗАО «Элвис-Неотек»

Заказчик:

ДРПО ЗАО «Элвис-Неотек»

Настоящий документ содержит необходимые исходные данные для разработки соответствующих разделов технического проекта.

### **2.2. Назначение и цели «Внедрения ALM Системы»**

#### **2.2.1. Назначение «Внедрения ALM Системы»**

«Внедрение ALM Системы» предназначено для координации потока персонала, процессов и информации в итерационном цикле операций доставки программного обеспечения.

#### **2.2.2. Цели и эффекты выполнения работы**

Целью «Внедрения ALM Системы» является создание в рамках ДРПО единого технического решения, позволяющего Компании компоновать более качественные продукты и службы, повышая коэффициент успешности проектов разработки и контролируя затраты, связанные с ошибками программного обеспечения.

Положительные эффекты от «Внедрения ALM Системы» выражены в повышении конкурентоспособности продуктов Компании за счет:

- снижения высоких затрат и рисков неэффективной разработки программного обеспечения;
- применения интегрированного подхода к управлению жизненным циклом ПО, предусматривающий совместную работу и оптимизацию;
- унификации инфраструктуры с помощью единой, расширяемой интегрированной платформы для всего жизненного цикла программного обеспечения;
- увеличения текущих инвестиций Компании в инфраструктуру разработки программного обеспечения, позволяя всем заинтересованным лицам (существующим и потенциальным клиентам, внутренним пользователям) совместно и открыто работать в режиме реального времени в контексте текущей задачи;
- обеспечения автоматизации и улучшенного управления процессом от начального определения требований до изменения программного обеспечения, управления выпусками, внедрения и поддержки.

### **2.2.3. Общие принципы «Внедрения ALM Системы»**

ALM Система позволяет всем участвующим в разработке программного продукта лицам работать в едином информационном пространстве, иметь единый интерфейс для доступа к любой информации, связанной с выпускаемым продуктом и при этом пользоваться привычными для них инструментами. Основой данной системы должно являться централизованное хранилище (одна или набор связанных баз данных).

## **2.3. Характеристика объектов автоматизации**

### **2.3.1. Объекты автоматизации**

В соответствии с настоящим ТЗ, объектами автоматизации являются:

- 1) процесс управления требованиями к разрабатываемой программной продукции
- 2) процесс координации действий групп разработчиков, отдела тестирования и определения уровня качества продукта, отдела планирования проектов, отдела документирования и сотрудников технической поддержки.

- 3) процесс управления интеллектуальной собственностью Компании в рамках департамента(разрабатываемый исходный код, проектная и сопроводительная документация)
- 4) процесс управления трудовыми ресурсами
- 5) процесс планирования выпусков
- 6) процесс тестирования
- 7) процесс сопровождения программного продукта

### **2.3.2. Общее описание автоматизируемых сервисов**

Управление проектами ведется группой менеджеров, между которыми распределены утвержденные проекты. Менеджер проводит разбиение проекта на задачи, расстановку приоритетов. На основе приоритетов, данных о наличии ресурсов и поставленных временных ограничениях создает Milestone'ы и помещает в них задачи по проекту.

Выпуск релиза продукта означает окончание Milestone'a.

Разработка продуктов ведется несколькими командами разработчиков, распределенными между несколькими офисами. Каждая команда имеет некую специализацию (по территориальной близости, по разрабатываемым компонентам, по языку реализации, по применяемой методике разработки).

Каждый менеджер определяет по имеющейся у него в различном виде информации, какой команде возможно поставить задачу в рамках проекта. Далее, менеджером совместно с командой определяется, в какой итерации задача может быть выполнена и задача попадает в список работ на эту итерацию. В рамках итерации задачи решаются в соответствии с приоритетом, выставленным выше. По окончании итерации команда представляет для владельцев проекта (менеджер и внутренние заказчики — поддержка, triage-группа) готовые решения поставленных задач. После чего цикл разработки повторяется снова.

Задачи, решаемые разработчиками можно разделить на несколько категорий:

- Разработка новой функциональности
- Поддержка старой функциональности (исправление дефектов ПО, перенос кода из одного продукта в другой)
- Оптимизация существующих продуктов
- Исследовательская деятельность

В рамках проекта, для сотрудников отдела тестирования и контроля качества, менеджерами проектов ставятся задачи на различные виды тестирования, в ходе которых отдел делает заключение о наличии в разработанном продукте дефектов или их отсутствии. Все найденные дефекты должны быть одобрены triage-группой, либо оценены как неподтвержденные. Если данный дефект был одобрен, менеджер должен поставить задачу команде разработки на его исправление.

Выпущенные продукты передаются сотрудникам поддержки для проведения внедрения у клиента.

## **2.4. Требования к системе**

### **2.4.1. Требования к сервисам**

#### **2.4.1.1. Общие требования к системе**

Система должна:

1. Позволять объединять все требования, предъявляемые заказчиками к продуктам, в едином хранилище, с одинаковым, для всех пользователей, способом доступа к ним. Требования внутри хранилища типизируются.
2. Позволять оперативно отслеживать процесс выполнения требований на всех этапах жизненного цикла продукта, как заказчикам, так и менеджерам, и разработчикам.
3. Доступ к информации о требованиях должен быть разграничен политиками предприятия.
4. Позволять проводить связи между требованиями, выполняемой работой и результатами на каждом этапе жизненного цикла продукта.
5. Позволять отслеживать историю разработки на всех этапах.
6. Доступ к информации о завершенной, текущей и запланированной работе по продукту должен быть разграничен политиками предприятия. Система должна позволять данные разграничения на уровне интеграции с соответствующими инструментами.
7. Позволять вести командную разработку, т.е. позволять координировать действия разработчиков.
8. Реализовывать обратную связь между пользователями создаваемого продукта и разработчиками.
9. Реализовывать сбор метрик работ по всем этапам жизненного цикла ПО и предоставлять их в графическом и символьном виде.
10. Иметь веб-интерфейс к описанным выше сервисам.

**11.** Реализовывать инструменты, необходимые для разработки по методологии Agile.

#### **2.4.1.2. Требования к интеграции**

Так как система затрагивает все этапы жизненного цикла ПО, то выше описанные сервисы должны интегрироваться с инструментами, применяемыми на этих этапах.

1. Интеграция с инструментами планирования
2. Интеграция с инструментами разработки
3. Интеграция с инструментами тестирования
4. Интеграция с инструментами развертывания

#### **2.4.2. Требования к этапам «Внедрения ALM Системы»**

«Внедрение ALM Системы» должно быть реализовано поэтапно и должно содержать следующие этапы:

1. Развертывание централизованного хранилища.
2. Развертывание базовых приложений системы.
3. Создание подсистемы управления требованиями и проектами.
4. Создание подсистемы командной разработки ПО.
5. Создание подсистемы тестирования.
6. Создание подсистемы поддержки.

## **2.5. Состав и содержание работ по созданию «ALM Системы»**

### **2.5.1. Перечень стадий и этапов работ по созданию системы**

Стадии и этапы работ для каждого из этапов внедрения системы определяются в соответствии с рекомендациями.

<b>Этапы</b>	<b>Работы</b>
<b>Техническое проектирование</b>	<ul style="list-style-type: none"><li>— Разработка проектных решений по системе и её частям</li><li>— Разработка документации на систему и её части</li><li>— Задание на рабочее проектирование</li></ul>
<b>Рабочее проектирование</b>	<ul style="list-style-type: none"><li>— Разработка рабочей документации на систему и её части</li><li>— Разработка и адаптация программ</li><li>— Тестирование программных модулей</li></ul>

В зависимости от специфики и условий выполнения работ по созданию системы допускается выполнять отдельные этапы работ до завершения предшествующих стадий, параллельное во времени выполнение этапов работ, включение новых этапов работ.

## **2.6. Порядок контроля и приемки подсистемы**

К сдаточным единицам системы относятся:

- Комплекс средств автоматизации, установленный на объекте внедрения.

Сдача системы должна осуществляться на объектах внедрения согласно перечню объектов и план-графику внедрения, утвержденным заказчиком.

Приемка должна осуществляться в виде проведения испытаний.

К видам испытаний относятся:

- Приемочные испытания.
- Опытная эксплуатация.

Состав, объем и методы испытаний должны быть определены в документах «Программа и методика испытаний», разрабатываемых для отдельных подсистем на этапах рабочей документации в соответствии с действующими нормами.

Перечень участвующих предприятий, организаций, место и сроки проведения испытаний, порядок согласования и утверждения приемочной документации должны определяться в организационно-распорядительной документации (приказы).

Для проведения испытаний должны быть созданы приемочные комиссии:

- для приемки системы в опытную эксплуатации составных частей системы;
- для проведения приемочных испытаний составных частей системы;
- для проведения испытаний системы.



## **2.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу подсистемы в действие**

Состав работ должен включать:

- работы по подготовке пилотных объектов (площадок) к вводу в опытную эксплуатацию опытных образцов ППО,
- работы по подготовке объектов внедрения к вводу в действие системы.

На объекте внедрения должна быть обеспечена необходимая комплектация пользователями и обслуживающим персоналом.

Завершение работ по подготовке объекта внедрения к вводу в действие системы оформляется актом, подписываемым заказчиком, исполнителем, исполнителями работ по внедрению составных частей системы и организацией - объектом внедрения.

## 2.8. Требования к документированию

Документация, разрабатываемая в ходе создания системы, должна соответствовать требованиям ГОСТ 34.201-89.

Перечень документации по созданию системы, уточняющей требования настоящего ТЗ:

Требования	Документ
Изменения к настоящему ТЗ	Дополнение или протокол
Требования к взаимодействию и характеристикам взаимосвязей со смежными системами	Регламенты, разрабатываемые на основании соответствующих договоров и соглашений
Вероятностно-временные характеристики выполнения отдельных функций и операции	Исследование времени
Требования к срокам и периодичности проведения работ по техническому обслуживанию для каждой подсистемы	ЧТЗ на создание составных частей системы
Требования к условиям окружающей среды	В составе рабочей документации
Перечень угроз, классификация передаваемой информации и требуемый уровень защиты информации для каждого из объектов	В составе рабочей документации

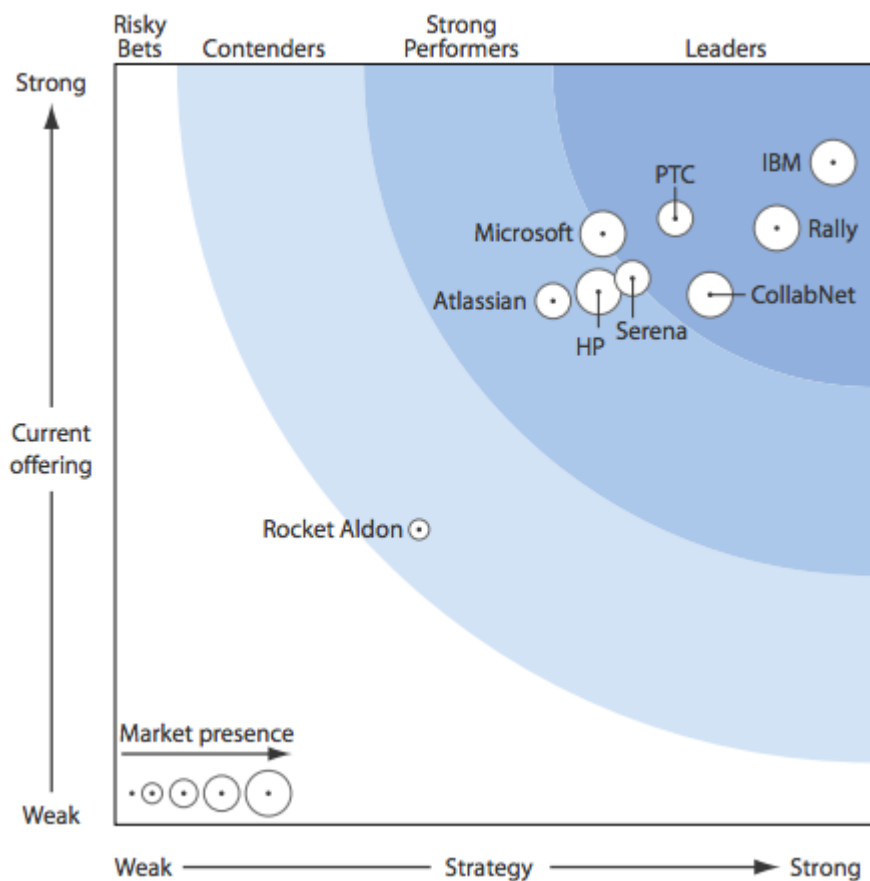
## **2.9. Другие требования к «Внедрению ALM Системы»**

Требования, необходимые для реализации отдельных этапов «Внедрения ALM Системы» описаны в регламентах и других документах, являющихся приложениями к данному ТЗ.

### 3. Концептуальный этап проектирования

#### 3.1. Анализ существующих программных продуктов и систем

По оценкам агентства Forrester (ежегодный аналитический отчет The Forrester Wave Application LyfeCycle Management) для сегмента рынка ALM, можно выделить группу систем, на основе которых можно создать ALM Систему.



Как видно, лидером этого сегмента является корпорация IBM. Исторически сложилось, что подразделение Rational Software, купленное в 2003 IBM, предоставляет наиболее полный комплекс средств для рынка разработки приложений. Его ближайшие конкуренты – PTC, Rally, CollabNet, Microsoft, - предоставляют достаточно нишевые продукты в сфере ALM. Наиболее комплексным из них является CollabNet TeamForge. Однако значительным минусом данного производителя является отсутствие поддержки в России. На ранних этапах внедрения данный факт окажет значительное влияние. Стоит обратить внимание на продукцию корпорации Microsoft, т.к. ее решение является комплексным, интегрированным с Visual Studio, а также другими программными продуктами Microsoft. Плюсом

является достаточно большое сообщество в России и наличие поддержки со стороны корпорации в нашей стране. Поэтому проведем сравнение решений двух корпораций – IBM и Microsoft.

Приведем несколько особенностей Rational и определим плюсы и минусы этих особенностей.

1) Rational делится на несколько независимых приложений.

Плюсы:

a. Гибкость компоновки системы, выбор только того, что нужно в данный момент.

b. Выход из строя 1 компонента не повлияет на другие.

c. Можно не переплачивать за другие компоненты.

Минусы:

d. Сложность развертывания, каждое приложение нужно устанавливать и настраивать отдельно.

e. Архитектурно нет единой системы хранения данных, необходимо объединять БД от разных компонентов самостоятельно.

2) Для интеграции Rational с Visual Studio необходим дополнительный компонент.

Минусы:

a. Нет возможности работать с системой сразу после установки средства разработки.

b. Лишнее звено между клиентским и серверным приложением.

3) Клиентское лицензирование ведется для каждого приложения.

Плюсы:

a. Гибкость компоновки системы, пользователь не должен платить за то, чем не пользуется.

b. Многие приложения (но не все) имеют минимальную бесплатную лицензию, так что элементарные операции сможет сделать любой пользователь.

Минусы:

- с. Сложность управления лицензиями.
- d. Переплата за многофункциональных сотрудников, по сравнению с комплексной лицензией.

4) Неполная интеграция с программными продуктами Microsoft.

Плюсы:

- a. Гибкость системы, многие компоненты не требуют наличия продуктов Microsoft (SQL Server, Windows Server, IIS).

Минусы:

- b. Сложность встраивания в существующую инфраструктуру, основанную на подавляющем количестве ПО от Microsoft.
- с. Потенциальная зависимость от конкретных версий продуктов в инфраструктуре предприятия. Неявные проблемы интеграции при обновлении данной инфраструктуры.

5) Языковая ориентированность Rational. В целом Rational ориентирован на разработку для C++\Java.

Плюсы:

- a. Отличная поддержка указанных языков.
- b. Кодогенерация и моделирование приложений на данных языках.

Минусы:

- с. Необходимость дополнительных решений для организации разработки .Net приложений.

Теперь проанализируем особенности Team Foundation Server.

1) Единый продукт с возможностью интеграции внешних приложений и отключения интегрированных компонент.

Плюсы:

- a. Простота развертывания комплексной системы.
- b. Простота обслуживания – единый интерфейс управления всеми компонентами.

- с. Простота обновления до новых версий.
- d. Неиспользуемые возможности просто отключить.
- e. Единая система хранения данных для всех компонент.

Минусы:

- f. В системе присутствуют неиспользуемые компоненты.
- g. Сложность тонкой настройки отдельных компонент.
- h. Зависимость компонент друг от друга.

2) Полная интеграция с Visual Studio. Visual Studio является одним из основных клиентских приложений для TFS.

Плюсы:

- a. Возможность быстро приступить к работе для большинства пользователей. Нужно лишь установить Visual Studio.
- b. Взаимодействие данного клиента с серверной частью системы максимально отлажено производителем.

3) Единое лицензирование для всех пользователей. Лицензия на Visual Studio автоматически дает клиентскую лицензию на пользование TFS.

Плюсы:

- a. Простота управления лицензиями.
- b. Нет необходимости покупать новую лицензию, если пользователь сменил компонент, с которым ему нужно работать.
- с. Для элементарных действий достаточно бесплатной лицензии.
- d. Экономия пользователям Visual Studio

Минусы:

- e. Переплата за лицензию при строгом разделений рабочей области.
- f. Дополнительные затраты при использовании сторонних клиентских приложений, в том числе и бесплатных.

4) Практически полная интеграция со всеми программными продуктами Microsoft. Интеграция со сторонним ПО - средняя.

Плюсы:

a. Простота встраивания в существующую инфраструктуру, основанную на подавляющем количестве ПО от Microsoft.

b. Наличие в программной линейке продуктов Microsoft ПО, расширяющего функционал системы, и простота использования такого ПО с TFS.

c. Поддержка интеграции с несколькими сторонними программными продуктами (Eclipse, Git, Code Collaborator, Enterprise Architect, Devart Code Compare & Review Assistant).

Минусы:

d. Компоненты требуют наличия продуктов Microsoft (SQL Server, Windows Server, IIS) для своего функционирования.

5) Отсутствие языковой ориентированности.

Плюсы:

a. Равная поддержка всех языков. Нет жесткой привязки к языкам, поддерживающим конкретные IDE или платформы.

Минусы:

b. Отсутствие кодогенерации и моделирования приложений в системе. Расширяется применением клиентских приложений.

6) TFS предоставляется бесплатно партнерам Microsoft.

Плюсы:

a. Значительная экономия на приобретении лицензии и поддержки.

Минусы:

a. Зачастую оказывается решающим фактором при принятии решения о внедрении, и не учитываются технические возможности системы.

Как видно, система от IBM рассчитана на широкий круг компаний, без учета специфики предприятий. Это удобно для системных интеграторов, которые могут гибко подбирать компоненты на основе собственных оценок работы компании, заказавшей автоматизацию. Зачастую такой подбор может существенно сэкономить ресурсы на внедрение. Однако дальнейшая



поддержка таких систем требует постоянного контакта с компанией-интегратором, что весьма невыгодно в долгосрочной перспективе. Также IBM в своей системе делает сильный упор на моделирование и разработку по методологии RUP.

Система Team Foundation Server, наоборот, рассчитана на предприятия, активно работающие с продуктами Microsoft, при этом не ограничивая возможность расширения за счет дополнительных средств поддержки разработки ПО. В TFS есть возможность поддержки собственных процессов разработки, отличных от какой-либо методологии, а также встроенная поддержка процессов MSF Agile/SCRUM, CMMI.

Аппаратные требования для компонент обеих систем приблизительно одинаковы. Программные требования зависят от конкретного развертывания и выбора используемых компонент. Обобщенное сравнение будет необъективным.

Также весомым преимуществом TFS является стоимость самой системы. Она существенно ниже ее конкурента от IBM, а за счет партнерских программ позволяет весьма сэкономить.

Для сравнения<sup>1</sup> – цена одного компонента из линейки продуктов Rational – от 13000 руб. до 140000 руб. Также нужно учитывать, что минимальная цена на лицензии с ограниченным количеством пользователей(1-5), без ограничения стоят больше. Стоимость комплексной системы, даже с учетом скидок при заказе сразу нескольких продуктов, будет превышать 200000 руб.

Стоимость TFS – 20000 - 30000 руб., в зависимости от конкретного продавца. Для пользователей MSDN, TFS предоставляется бесплатно. Клиентские лицензии на всю систему входят либо в подписку MSDN, либо в стоимость Visual Studio. Если же пользователь не использует для своей работы Visual Studio, а также не является подписчиком MSDN, то возможно ему подойдет бесплатная лицензия для минимального использования TFS Web Access. В остальных случаях стоимость лицензии 12000 – 18000 за

---

<sup>1</sup> Цены приведены на второй квартал 2013г по данным сайтов softline.ru, ibm.com, microsoftstore.com

пользователя. Таким образом стоимость минимального комплекта TFS для 5 разработчиков не превысит 120000 руб., что почти в 2 раза ниже цены на систему от IBM с аналогичной функциональностью.

На основании проведенного сравнения можно сделать вывод о том, что программный продукт Microsoft Team Foundation Server 2012 больше подходит в качестве основы ALM Системы для ДРПО ЗАО «Элвис-Неотек».

### **3.2. Выбор языка программирования**

Разрабатываемая система не требует прямого доступа к памяти и устройствам компьютера, соответственно она может быть реализована на платформе Java, или Microsoft.NET, что значительно менее трудоемко по сравнению с разработкой на традиционных компилируемых языках программирования высокого уровня (C++, ObjectiveC). Конечно программы, реализованные на этих платформах, работают несколько медленнее, чем заранее скомпилированные, но, при наличии JIT-компиляторов и современном уровне развития вычислительной техники, этот недостаток не является критичным. Ключевым фактором выбора языка является то, что TFS предоставляет широкий набор API для .NET и намного более скромный для Java.

К достоинствам платформы .NETFramework можно отнести:

- создание целостной объектно-ориентированной среды программирования допускающей различные варианты реализации: код может храниться и выполняться локально; выполняться локально, а распространяться через Интернет; или выполняться удаленно;
- предоставление среды выполнения кода, в которой число конфликтов при развертывании программного обеспечения и управлении версиями будет сведено к минимуму;
- обеспечение безопасности выполнения кода в среде - в том числе кода, созданного неизвестным разработчиком или разработчиком с частичным доверием;

- предоставление среды выполнения кода, позволяющей устранить проблемы, связанные с производительностью сред на основе сценариев или интерпретации;
- унификация работы разработчиков в совершенно разных приложениях: как в приложениях Windows, так и в веб - приложениях;
- использование промышленных стандартов во всех областях обмена данными и, как следствие, обеспечения совместимости кода, созданного в .NET Framework, с другими программами.
- наличие средств, позволяющих легко создавать удобный пользовательский интерфейс.

### **3.3. Основные принципы разработки**

ALM Система, в основу которой положен Microsoft Team Foundation Server должна разрабатываться по нескольким направлениям:

- Расширение поддержки процесса разработки в ДРПО, за счет доработки шаблона проекта на внутреннем языке описания;
- Создание новых алгоритмов автоматизации процесса разработки, выполнявшихся ранее вручную;
- Доработка пользовательских интерфейсов;
- Создание внешних, относительно TFS, подсистем выполняющих сервисное обслуживание.

#### **4. Технический этап проектирования**

Основное приложение, используемое в ALM системе – Team Foundation Server.

Для организации вывода информации, возможно использование следующих приложений – Microsoft SQL Server Reporting Services, Microsoft Office Excel, Team Foundation Server Web Access, Microsoft Visual Studio. Также можно организовать web-портал, позволяющий пользователям получать информацию из ALM системы в виде web-страниц. Основной задачей при создании такого сервиса будет написание клиентского приложения, использующего TFS API. Однако преимущество TFS в том, что в него заложена интеграция с MS SharePoint, позволяющего быстро развернуть web-портал для проекта в Team Foundation Server.

Excel и Visual Studio являются приложениями, установленными на клиентской машине. Для доступа к ним необходима установка ОС Windows (версии XP и выше) и наличие самого приложения. Поэтому они ограничивают доступ к данным ALM системы с мобильных устройств и устройств с ОС, отличной от MS Windows. Данные приложения не являются частью внедряемой ALM системы.

Team Foundation Server Web Access является предустановленным компонентом TFS, поэтому, при планировании ПО, его можно не учитывать, как отдельное приложение.

Microsoft SQL Server Reporting Services является удобным средством вывода информации из TFS, он не накладывает дополнительных требований на установку клиентских приложений.

Необходимым условием использования Microsoft SQL Server Reporting Services вместе с TFS является Microsoft SQL Server Analysis Services. Оба этих приложения являются компонентами Microsoft SQL Server в редакции Standard.

Для того, чтобы в полной мере использовать возможности платформы TFS, средством вывода информации в системе будет использован MS SharePoint Server.

Таким образом список ПО, необходимого для ALM Системы слежующий:

1. Team Foundation Server 2012
2. SQL Server Standard 2012
3. SharePoint Server Enterprise 2013
4. Microsoft Server 2012 Standard

#### **4.1. Выбор архитектуры системы**

Логическая структура TFS предполагает наличие 3 структурных едениниц – уровня приложений, уровня данных и клиентского уровня.

К уровню приложений относится сам Team Foundation Server, консоль администрирования и служебные утилиты.

К уровню данных относится один или несколько серверов с SQL Server, обслуживающих базы данных TFS и коллекций проектов.

К клиентскому уровню относятся все приложения и подсистемы, которые обращаются к Team Foundation Server через TFS Framework. К ним относятся:

1. Build Service
2. Test Management
3. Work Item Tracking
4. Version Control
5. TFS Web Access
6. SharePoint
7. Project
8. Visual Studio
9. Excel

Клиентские приложения 1-5 имеют внутреннюю клиент-серверную архитектуру. Их серверная часть встроена в TFS и поэтому, с точки зрения компонентов, их можно отнести также к уровню приложений.

Сам TFS не накладывает ограничений на возможность расположения компонентов системы в сетевой и физической инфраструктуре. Поэтому,

изоляции отдельных компонентов системы применим распределенную архитектуру из нескольких серверов.

Для оптимизации компоновки системы применим технологию виртуализации операционных систем.

#### 4.2. Выбор платформы виртуализации

Основными преимуществами технологии виртуализации информационной инфраструктуры являются:

- снижение капитальных и эксплуатационных затрат;
- реализация упрощенной, динамичной инфраструктуры системы;
- повышение готовности вычислительных ресурсов;
- уменьшение времени, необходимого для подготовки или предоставления новых сервисов;

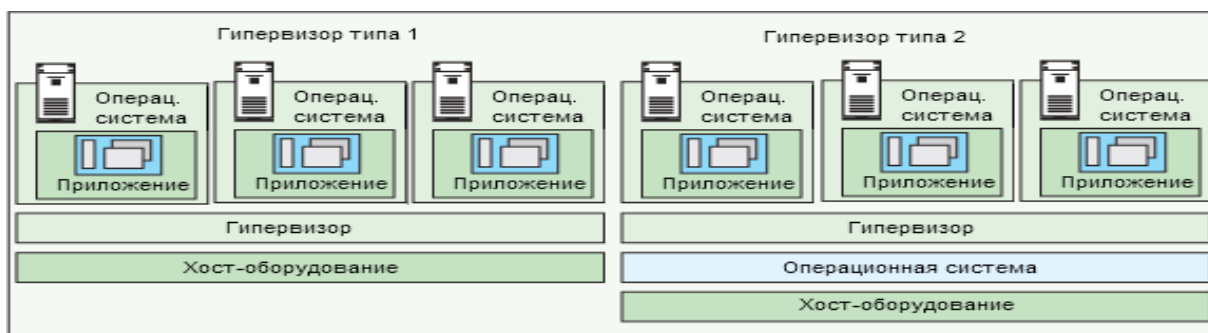


Рис. 4.2

- уменьшение сложности управления.

Основными платформами виртуализации являются:

- MS Hyper-V
- Oracle VM Server
- VMware vSphere
- Citrix XenServer
- RedHat RHEV

Стоимость VMware vSphere и не позволяет применять его для

виртуализации отдельной системы в инфраструктуре предприятия.

Citrix XenServer имеет высокую степень надежности, однако в ней присутствует ряд ограничений на виртуальные машины, такие как количество виртуальных процессоров, которые не позволяют выбрать ее в качестве основы инфраструктуры ALM Системы.

RedHat RHEV и Oracle VM Server предоставляют особые требования к аппаратной конфигурации, что также исключает возможность их выбора.

Таким образом, наиболее подходящей платформой виртуализации для ALM Системы является MS Hyper-V версии 3.0, входящей в состав Windows Server 2012 Standard.

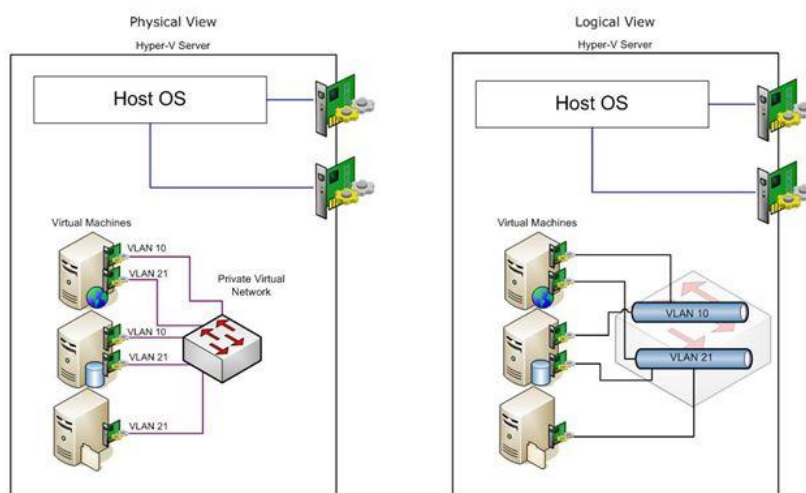


Рис. 4.3

### 4.3. Планирование уровня данных

Наиболее требовательным к вычислительным ресурсам уровнем является уровень данных, т.к. различные компоненты TFS взаимодействуют между собой в основном через базу данных.

Также, с целью консолидации ресурсов баз данных и упрощения их администрирования, объединим сервера данных для TFS и SharePoint Server в одном узле SQL Server. Дополнительным преимуществом такой агрегации является уменьшений накладных расходов при работе приложений, обращающихся и к базам TFS и к базам SharePoint, например Project Server или Reporting Services.

### 4.4. Планирование уровня приложений

Уровень приложений представляет из себя установленные службы TFS, консоль администрирования, набор сервисных утилит для тонкого

обслуживания сервера.

Рассматриваемый компонент размещает в себе следующие подсистемы ALM Системы:

- 1) служба версионного контроля;
- 2) служба отслеживания работы;
- 3) служба управления сборкой ПО;
- 4) TFS Web Access.

Уровень предоставляет возможность расширения системы, за счет размещения на нем дополнительных служб, написанных сторонними разработчиками. Реализация расширений проводится путем подключения пользовательских библиотек через компонент TFS Framework. Данный компонент входит в состав TFS SDK, предоставляющий API на платформе Microsoft.NET к серверным и клиентским приложениям системы. Таким образом уровень приложений является одной из точек интеграции ALM Системы в бизнес процесс департамента.

#### **4.5. Планирование клиентского уровня**

Уровень клиентских приложений включает в себя следующие подсистемы:

- Подсистему вывода информации;
- Подсистему выполнения сборки приложений.

##### **4.5.1. Подсистема вывода информации**

Представляет собой web-портал на сервере SharePoint, связанный через уровень данных с базами проектов и приложением TFS Web Access. Команды могут использовать данный портал для обмена инструкциями по проекту, проектной документацией, шаблонами и отчетами в соответствии с ролью каждого участника команды в рамках проекта.

Для обеспечения ведения проектов в ALM системе предлагается использовать MS Project Server 2013, основой для которого является SharePoint Server.

С целью упрощения управления клиентским уровнем, объединим в одном компоненте 2 клиентских приложения.



#### **4.5.2. Подсистема выполнения сборки приложений**

Состоит из 3 компонент:

- 1) Агент службы управления сборкой ПО;
- 2) Team Build – расширение утилиты MSBuild для работы с TFS.
- 3) Клиент службы версионного контроля.

Подсистема представляет собой сервер сборки, который может быть установлен практически на любой машине, поддерживающей Visual Studio. Она может использоваться разработчиками для выполнения полной сборки большинства последних версий программного обеспечения, хранимых в системе версионного контроля. Записи каждой сборки сохраняются вне зависимости от ее успешности или неуспешности, так что разработчики и администраторы сборок могут отслеживать прогресс проекта. Если сборка происходит последовательно, то анализируются изменения, сделанные в исходном коде после последней успешной сборки, а обновление рабочих элементов указывает на определенный прогресс. Например, если тестировщик заводит рабочий элемент, посвященный конкретной ошибке в сборке №15, а разработчик вносит изменения чуть ранее, чем была создана сборка №18, то элемент «Bug» обновится до статуса, указывающего, что ошибка исправлена. Тестировщик может как подтвердить, так и опровергнуть то, что ошибка была успешно исправлена.

#### **4.6. Расчет необходимого аппаратного обеспечения ALM Системы**

Для обеспечения работы системы необходимо определить оптимальные конфигурации аппаратных средств.

Расчет будем проводить исходя из рекомендуемых требований к ПО, используемому в системе. Основные рекомендации получены из официальной документации Microsoft, расположенной в сети MSDN.

В основе архитектуры расположен гипервизор MS Hyper-V, версии 3.0, входящий в состав Windows Server 2012 Standard. Требования к аппаратному обеспечению даны гипервизора:

- 1) x64-совместимый микропроцессор Intel или AMD;
- 2) поддержка виртуализации с использованием оборудования.

Технологии Intel Virtualization Technology (Intel VT) и AMD Virtualization (AMD-V);

3) 16Гб свободного дискового пространства.

Требования к остальным ресурсам отсутствуют.

Для размещения виртуальных машин с помощью Hyper-V, аппаратная конфигурация должна включать в себя реализацию требований к сумме ресурсов, требуемых каждой виртуальной машиной. Проведем анализ требований компонентов системы. Результат представим в сводной таблице(табл.4.1).

Компонент	Процессор	ОЗУ, ГБ	Дисковая подсистема	Дополнительные требования
Team Foundation Server 2012	x64, 2 ядра	4	10Гб свободного пространства	Накладные расходы на работу IIS
SQL Server Standard 2012	x64, 8 ядер	24	10Гб свободного пространства	Iops > 200, отдельный диск для хранения баз данных
SharePoint Server Enterprise 2013	x64, 8 ядер	10	30Гб свободного пространства	Накладные расходы на работу IIS
Project Server 2013	x64, 4 ядра	4	5Гб свободное пространства	Дополнительное пространство в размере двойного ОЗУ для кэширования данных

*Табл. 4.1*

Стоит учесть, что сервер баз данных будет использоваться для обслуживания двух компонентов системы – уровня данных и клиентского уровня. Соответственно, при консолидации 2 серверов на одной виртуальной

машине можно применить коэффициент ( $k = 1.75$ ) к ресурсам данной виртуальной машины.

Каждая виртуальная машина будет работать под управлением MS Windows Server 2012 – наиболее оптимизированная серверная система для Hyper-V 3.0.

Накладные расходы на каждый экземпляр ОС:

- 1) 768 Мб ОЗУ;
- 2) 1 вычислительное ядро;
- 3) 32 Гб свободного дискового пространства.

Проссумируем требования к аппаратной спецификации сервера ALM Системы:

- 1) x64-совместимый микропроцессор Intel или AMD, с 32 вычислительными ядрами;
- 2) поддержка виртуализации с использованием оборудования. Технологии Intel Virtualization Technology (Intel VT) и AMD Virtualization (AMD-V);
- 3) 45Гб ОЗУ;
- 4) 150 Гб свободного дискового пространства;
- 5) Диск для хранения баз данных, обладающий  $iops > 200$ .

#### **4.7. Выбор аппаратного обеспечения ALM Системы**

На основе расчета выберем подходящее оборудование.

Компании HP, IBM, Dell - мировые лидеры среди производителей серверов. Среди их продукции наиболее полно соответствуют требованиям следующие модели:

- 1) HP DL 380p G8
- 2) IBM System x3650 M4
- 3) Dell PowerEdge R720

При аналогичных характеристиках у серверов HP есть 2 положительные особенности:

— система удаленного управления ILO, способная работать по 1 физическому порту с самис сервером;

— наиболее обширная поддержка производителя на территории России.

В дополнение ко всему, в серверах HP есть аппаратная поддержка гипервизора Hyper-V.

Приведем технические характеристики сервера HP DL 380p G8(табл.4.2).

Технические характеристики	
<b>Процессор</b>	Intel® Xeon® E5-2690 (8-ядерный, 2,9 ГГц, 20 МБ, 135 Вт)
<b>Количество процессоров</b>	2
<b>Количество ядер процессора</b>	16(Hyper Threading)
<b>Стандартный объем памяти</b>	32 ГБ (4x8 ГБ) RDIMM
<b>Слоты для памяти</b>	24 слота DIMM
<b>Тип памяти</b>	4R x4PC3L-10600L-9
<b>Сетевой контроллер</b>	2-портовый адаптер 10 Гбит 530FLR-SFP+ Ethernet на контроллер
<b>Блок питания</b>	2 блока питания Platinum Plus 750 Вт с возможностью горячей замены и общим слотом (комплект)
<b>Контроллер хранилища</b>	Smart Array P420i/2 ГБ FBWC
<b>Оптический привод</b>	Slim SATA DVD-RW
<b>Форм-фактор</b>	2U

*Табл. 4.2*

Для обеспечения отказоустойчивости и высокой скорости работы необходимо комплектовать данный сервер 2 жесткими дисками SAS, объемом 300Гб и объединенными в RAID0.

Обеспечить диск для хранения баз данных можно благодаря объединению 4-х дисков SATAIII в RAID 1+0. Объем выберем исходя из

средних размеров баз аналогичных систем - 500 Гб в год для среднего размера набора проектов(1-5 долгосрочных проекта). Таким образом объем диска должен быть не менее 1Тб на 2 года работы ALM Системы.

Реализовать требуемы объем ОЗУ можно добавив в стандартную комплектацию еще 4 модуля, т.к. память долэна работать в 2-х-канальном режиме. Таким образом мы получим 64Гб ОЗУ, что превышает необходимый объем на 25%.

#### **4.8. Проектирование развертывания ALM Системы**

В ходе проектирования компонентной структуры ALM Системы было принято решение объединить сервер уровня данных с сервером базы данных SharePoint Server.

Также, из-за невысоких требований к аппаратному обеспечению, возможно объединить уровень данных с уровнем приложений в одной операционной системе. Данное решение позволит убрать необходимость сетевого взаимодействия приложений и базы данных, что существенно упростит развертывание системы.

Для клиентского уровня необходимо развернуть SharePoint Server с размещенным на нем MS Project Server. Также для интеграции обеих служб с TFS необходимо установить TFS SharePoint Extensions и TFS Project Extensions. Общение с между серверами базы данных и совмещенным сервером уровня данных и приложений должно осуществляться по виртуальной внутренней сети, организованной средствами гипервизора.

Таким образом для серверной части ALM Системы необходимо создать два виртуальных сервера на 1 физическом, выбранном в предыдущем разделе.

Спецификации виртуальных машин приведены ниже в таблицах:

Имя	ALMDataAppTier	
-----	----------------	--

Процессор	16 cores	
Оперативная память	32 GB	
Жесткие диски	Том в VM	Комментарии
Системный	SystemDrive (C:) - 75 Гб	
Для данных	Storage(S:) - 3 Тб	Диск расположен на дисковом массиве RAID 1+0
OS	Windows Server 2012 Standard	

Табл. 4.3

Имя	ALMWebServices	
Процессор	14 cores	
Оперативная память	12 GB	
Системный диск	SystemDrive (C:) - 125 Гб	
OS	Windows Server 2012 Standard	

Табл. 4.4

При развертывании системы, старые сервера сборки должны быть заменены на один физический сервер, аналогичный выбранному, с добавленным жестким диском типа SSD для проверки компиляции проектов. На нем должны быть развернуты новые сервера сборки с установленными агентами подсистема выполнения сборки приложений.

#### 4.9. Обеспечение отказоустойчивости системы

Для реализации отказоустойчивой системы необходимо спроектировать несколько подсистем:

- Подсистему управления питанием аппаратного обеспечения
- Подсистему резервного копирования программного обеспечения
- Подсистему резервного копирования данных

Подсистему резервирования сетевого соединения проектировать не надо, т.к. ALM Система встраивается в инфраструктуру предприятия, где данная функциональность уже реализована.

Подсистему резервного копирования программного обеспечения и подсистему резервного копирования данных объединим в общую подсистему резервного копирования.

##### 4.9.1. Проектирование подсистемы управления питанием

Подсистема должна выполнять следующие функции:

- а) Передачу питания аппаратным средствам от основного источника питания предприятия;
- б) При краткосрочных сбоях основного питания (скачки напряжения, перебои в подаче питания до 10 минут) позволять системе работать в штатном режиме;
- с) При длительных перебоях питания корректно завершить работу системы и автоматически восстановить работу после восстановления питания;
- д) Уведомлять сотрудников по электронной почте о выключении системы.

Для реализации данных функции необходимо выбрать источник бесперебойного питания. В большинстве современных моделей есть необходимые функции для реализации требований к подсистеме.

Лидером рынка данных устройств является фирма APC.

Существует несколько классов источников бесперебойного питания, различаемых по технологии реализации переключения между основным и резервным источниками.

Модели класса систем с двойным преобразованием, обеспечивает наивысшую степень защиты электроприборов, к ней подключаемых.

Главным же недостатком резервных, как и линейно-интерактивных ИБП остается необходимость в переключении питаемой нагрузки с напряжения сети на генератор (инвертор), питаемый от батареи. Время такого переключения нагрузки, все же, нельзя считать «нулевым», само переключение вызовет переходный процесс, с наличием нежелательных «всплесков». Все это приводит как минимум к сбою в работе, кратковременному пропаданию питания на каждом устройстве, подключаемом к ИБП.

Питание с двойным преобразованием от указанной проблемы избавлено.

В любом ИБП APC серии Smart есть функция выключения или включения по расписанию, по сигналу и по событию, для чего используется сетевая плата Web/SNMP управления.

Адаптер APC SNMP, играет роль и SNMP-агента и платы Ethernet для всех устройств этой серии. На самой плате адаптера – только один RJ-разъем (LAN 10MBit), а компьютеры, получающие сигнал с ИБП, находятся том же сегменте локальной сети, что и сам ИБП.

Предупреждающий сигнал посылается на все рабочие станции, после чего, каждая, получившая этот сигнал, автоматически выполнит действия: остановку всех сервисов; выключение. Для реализации этих действий на устройстве должны быть установлены специальные программные службы, поставляемые вместе с источником бесперебойного питания.

На основе выше перечисленных особенностей и максимальной мощности 2 серверов выбираем устройство APC Smart-UPS X 3000VA Rack/Tower LCD 200-240V with Network Card, удовлетворяющее всем параметрам.

#### **4.9.2. Проектирование подсистемы резервного копирования**

В основе подсистемы резервного копирования положим функции SQL Server и MS Hyper-V. Подсистема будет располагаться в виде запускаемого по расписанию модуля на физическом сервере.

Приведем основные этапы работы подсистемы:

- 1) Резервное копирование баз данных;
- 2) Резервное копирование системных разделов и конфигураций виртуальных машин;
- 3) Создание архивов соответствующих копий;
- 4) Передача архивов на файловый сервер ДРПО.

Листинг программы подсистемы резервного копирования представлен в Приложении 11.3.

#### **4.10. Планирование работ по адаптации TFS к процессу ДРПО**

Проведя анализ шаблона процесса работы с проектом в TFS, были выявлены следующие особенности, не позволяющие использовать данный шаблон:



- 1) Набор состояний сущности «Bug» не соответствует процессу работы с дефектами;
- 2) Положение сущности «Bug» в иерархии сущностей не позволяет вести работу над ней, как над обычным требованием;
- 3) Сущность «Issue» не может использоваться, т.к. находится вне иерархии сущностей, с которыми могут работать разработчики;
- 4) У сущности «Task» отсутствует состояние Review;
- 5) Пользователям не понятно, как заполнять базовые поля при создании сущности «Bug», нужен шаблон в текстовом поле;
- 6) Работать с TFS Web Access Board весьма неудобно, т.к. на ней не видны ID элементов;
- 7) Не хватает автоматизации постановки задач на приемочное тестирование;
- 8) Не хватает автоматизации при закрытии и открытии вложенных задач;
- 9) Не хватает возможности создавать гибкие пользовательские оповещения.

Таким образом, для интеграции спроектированной ALM Системы необходимо переработать шаблоны проекта для TFS, добавить модули интеграции и модифицировать пользовательский интерфейс.

## **5. Исследовательская часть**

### **5.1. Постановка задачи**

Одним из требований внедрения ALM Системы была возможность ее дальнейшей поддержки сотрудниками ДРПО.

Необходимо провести исследование времени, затрачиваемого сотрудником на проведение основных мероприятий по администрированию, поддержке и доработке старой и новой систем.

### **5.2. Методика исследования**

Сотруднику ДРПО, проводящему обслуживание старой системы, была поставлена задача выполнять аналогичную работу в течение итерации на тестовом стенде с ALM Системой.

### **5.3. Результат**

Временные показатели двух систем приведены на сравнительной диаграмме.

Как можно заметить, почти по всем показателям новая система сокращает время на поддержку и администрирование. Доработка занимает чуть больше времени, т.к. приходится работать с комплексом, а не отдельными компонентами.

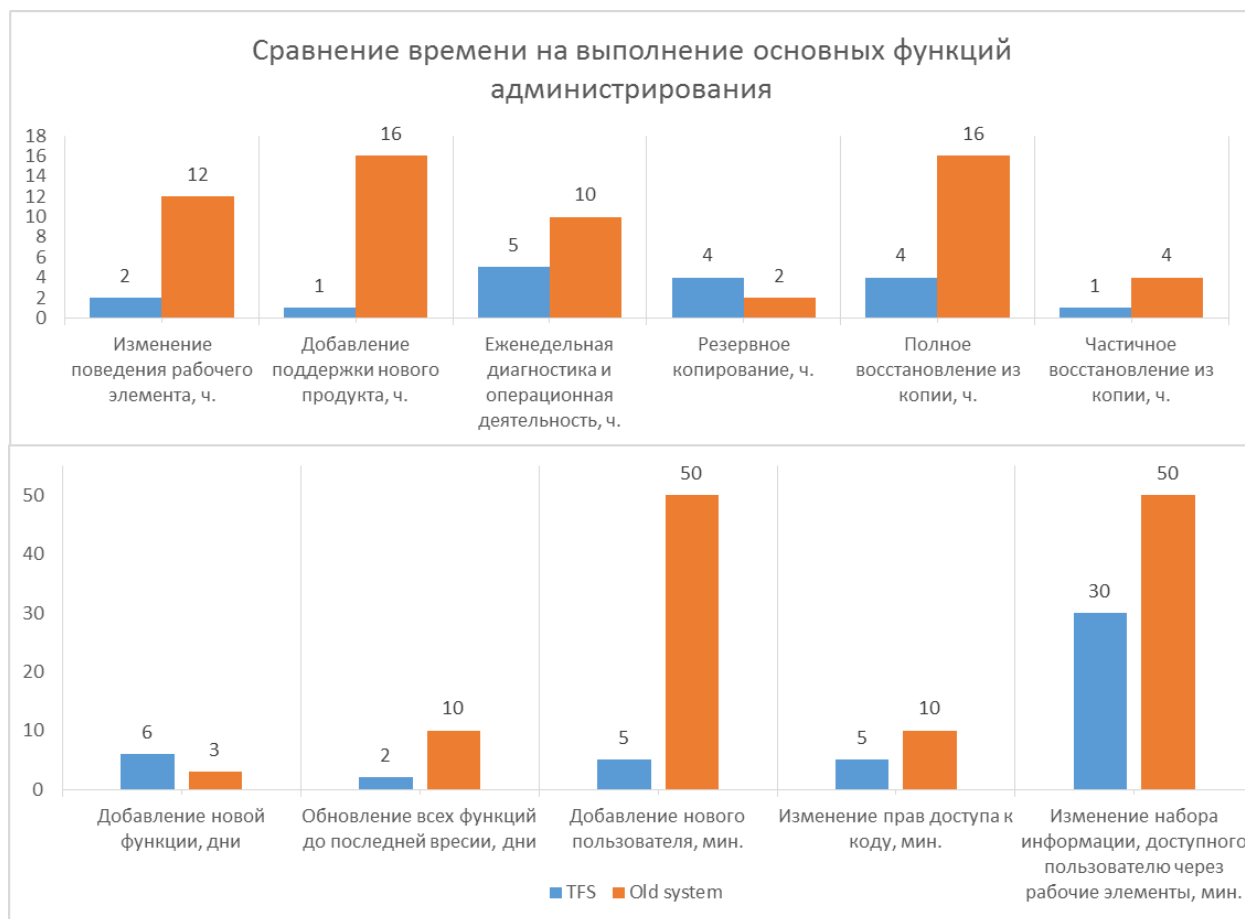


Рис. 5.4

## **6. Рабочее проектирование**

### **6.1. Переработка шаблона проекта**

Вся работа в проекте TFS строиться на основе шаблонов процесса. Шаблоны содержат в себе следующие основные составляющие:

- **Руководство по процессу.**

Для любого нового проекта создается свой web-портал, который содержит полное описание процесса на основе, которого ведется разработка. Руководство содержит в себе детальное описание ролей, работ, которые эти роли выполняют, а также полное описание рабочих элементов, их полей и состояний.

- **Рабочие элементы.** Для каждого рабочего элемента существует отдельное в специальном виде описание полей и их размещение на форме, состояний и возможные переходы между ними. Все новые рабочие элементы, которые формируются в проекте, создаются на основе этих описаний. Также предусмотрен механизм изменения полей, формы и жизненного цикла рабочих элементов.

- **Отчеты.** Каждый проект должен подвергаться отчетности. Отчетность дает понимание, в каком состоянии находится проект, входит ли он рамки бюджета, сколько работы выполнено и сколько еще предстоит выполнить, сколько дефектов было обнаружено и сколько уже исправлено и т.д. Шаблон процесса содержит в себе иницирующие отчеты, которые будут создаваться для каждого нового проекта.

- **Шаблоны документов.** Шаблоны процессов в TFS позволяют размещать и использовать в проектах не только рабочие элементы для выполнения проектного управления, но использовать проекты как общий источник проектной документации, планов работ, планов тестирования и т.д.

Проведя анализ на этапе технического проектирования были составлены требования, которые необходимо реализовать в вде измененного шаблона. Изменения затронут рабочие элементы в шаблоне.

### 6.1.1. Элемент Bug

Рабочий элемент – сущность, описывающая дефект в проекте. Для интеграции с процессом ДРПО были внесены следующие изменения:

- 1) Заменены все состояния;
- 2) Добавлены новые поля, в частности поле Triaged;
- 3) Добавлен шаблон заполнения полей, отображающийся в интерфейсе пользователя;
- 4) Изменены иерархические связи с другими элементами.

Полученное описание элемента приведено в приложении 11.4.

### 6.1.2. Элемент Task

Рабочий элемент – сущность, описывающая техническую задачу для разработчика. Для интеграции с процессом ДРПО были внесены следующие изменения:

- 1) Добавлено состояние Review, описывающее нахождение задачи на этапе инспекции кода.

```
<STATE value="Review">
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Scheduling.OriginalEstimate">
      <FROZEN />
      <READONLY />
    </FIELD>
  </FIELDS>
</STATE>
```

Полученное описание элемента приведено в приложении 11.5.

### 6.1.3. Иерархия элементов

Для изменения иерархии рабочих элементов нужно:

- 1) Заменить описания в компонента Categories в шаблоне проекта;

```
<CATEGORY refname="Microsoft.BugCategory" name="Bug Category">
  <DEFAULTWORKITEMTYPE name="Bug" />
</CATEGORY>
<CATEGORY refname="Microsoft.RequirementCategory" name="Requirement
Category">
  <DEFAULTWORKITEMTYPE name="User Story" />
  <WORKITEMTYPE name="Bug" />
</CATEGORY>
```

- 2) Заменить описания компонента CommonProcessConfig в шаблоне проекта;

Полученное описание компонента приведено в приложении 11.6.

- 3) Поменять иерархические связи в описаниях самих элементов.

3й пункт был выполнен на этапе модификации самих элементов.

## **6.2. Модификация пользовательского интерфейса**

Для добавления ID элементов был написан небольшой модуль на языке JavaScript. Он анализирует элементы на отображаемой странице с интервалом в 10 секунд и дописывает в них полученные ID рабочих элементов.

Листинг модуля приведен в Приложении 11.7

## **6.3. Создание пользовательского обработчика событий**

Целью создания данного модуля является решение 3 проблем:

- 1) Не хватает автоматизации постановки задач на приемочное тестирование;
- 2) Не хватает автоматизации при закрытии и открытии вложенных задач;
- 3) Не хватает возможности создавать гибкие пользовательские оповещения.

В качестве решения был создана библиотека реализующая следующие классы:

- `WorkItemChangedEventHandler` – обработчик события в системе
- `WorkItemChangeProcessor` – исполнитель изменений над рабочим элементом. Базовый класс для `UserStoryChageProcessor`, `TaskChangeProcessor`, `BugStoryChageProcessor`.
- `UserStoryChageProcessor` – исполнитель изменений над рабочим элементом типа «User Story»
- `TaskChangeProcessor` – исполнитель изменений над рабочим элементом типа «Task»
- `BugStoryChageProcessor` – исполнитель изменений над рабочим элементом типа «Bug»

Алгоритм работы данной библиотеки, подключенной к системе, показан на диаграмме активностей(рис.6.3)

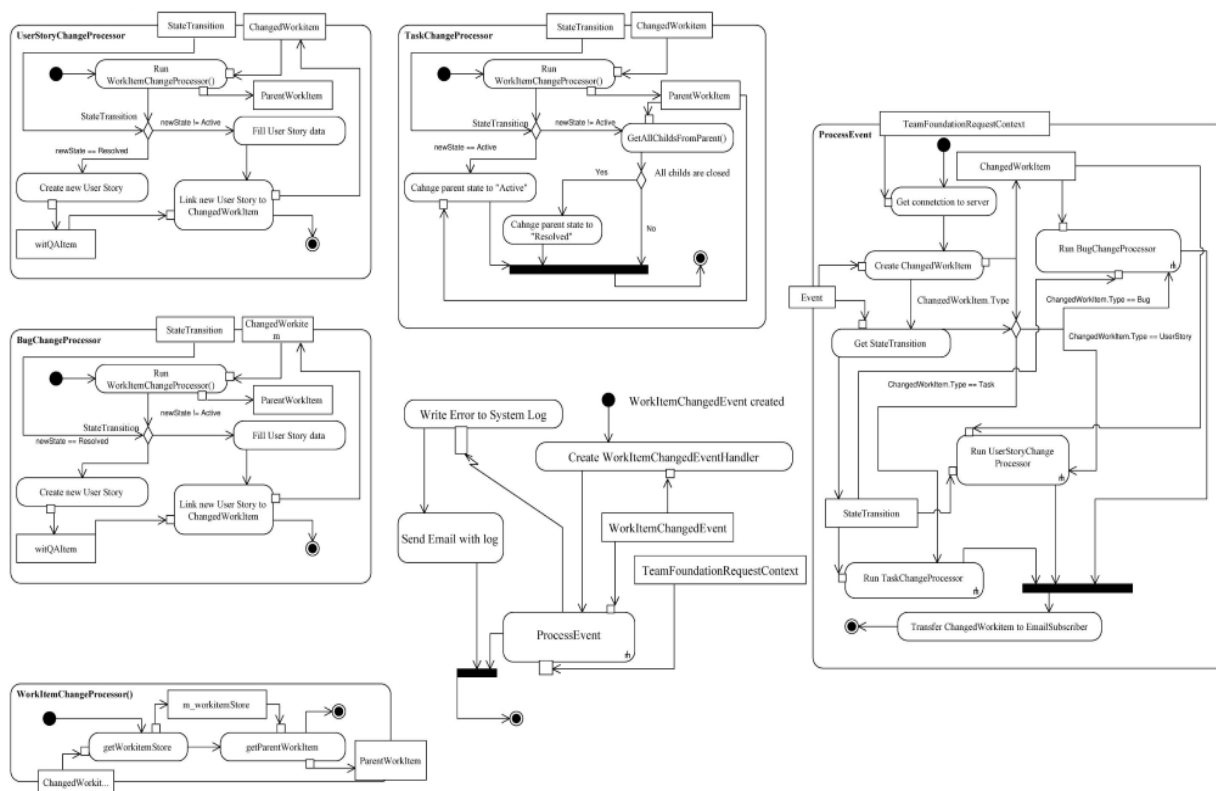


Рис. 6.5

Листинг модуля приведен в Приложении 11.8

## **7. Организационно-экономическая часть**

### **7.1. Введение**

В организационно-экономической части дипломного проекта производится расчет затрат внедрения ALM Системы. Для определения данного экономического показателя необходимо провести расчет трудоемкости внедрения ALM систем, разработки ПО, стоимости закупки оборудования, миграции процессов с используемых средств автоматизации на единую систему.

Чтобы оценить целесообразность разработки и внедрения ALM системы с экономической точки зрения, необходимо определить ориентировочную стоимость дополнительно разрабатываемых подсистем, стоимость приобретаемых систем, а так же сроки разработки и внедрения.

При определении плановой себестоимости рассчитываются все затраты, связанные с разработкой программного комплекса. Определение затрат на проведение работ производится путем составления калькуляции плановой себестоимости. Она является основным документом, на основании которого осуществляется планирование и учет затрат на разработку программного комплекса.

Затраты на разработку могут быть представлены в виде сметы затрат, включающей в себя затраты по следующим статьям[!]:

- затраты на оплату труда;
- отчисления на единое социальное страхование;
- амортизационные отчисления;
- накладные расходы.

При определении затрат на внедрение ALM системы в данной части работы будут представлены данные на второй квартал 2013 года (заработная плата рабочих, стоимость оборудования и др.) и произведены такие расчеты, как:

- оценка трудоёмкости разработки и внедрения ПО;



- оценка состава и численности разработчиков ПО;
- оценка трудоёмкости работ каждого участника проектной группы;
- оценка стоимости приобретаемых компонентов системы;
- оценка стоимости разработки внедрения системы.

## **7.2. Оценка трудоёмкости и сроков разработки модулей интеграции**

Трудоёмкость разработки ALM системы зависит от ряда факторов:

- степень новизны разрабатываемого ПО;
- сложность алгоритма его функционирования;
- объем используемой информации;
- вид представления и способ обработки информации;
- уровень используемого алгоритмического языка программирования.

По степени новизны разрабатываемый проект относится к группе новизны В, т.е. к системам, для которых имеются аналоги. Примером аналога разрабатываемой подсистемы являются существующие подсистемы различных ALM систем. По степени сложности алгоритма функционирования проект относится к 3 группе сложности - программная продукция, реализующая алгоритмы стандартных методов решения задач.

По виду представления исходной информации, на данный момент она представлена в электронном – база данных существующих процессов, регламентов и внутренних требований ДРПО. Таким образом, необходимо осуществлять контроль входной информации, представленной в электронной форме, но различных форматах, учитывая её взаимовлияние. Соответственно, по виду представленной исходной информации, способа ее контроля и структуры выходных документов, разрабатываемая подсистема относится к группе 12 - исходная информация представлена в форме документов, имеющих одинаковый формат и структуру, требуется форматный контроль информации. По виду выходной информации разрабатываемая подсистема относится к группе 22 - требуется печать документов одинаковой формы и содержания, вывод массивов данных на машинные носители, так как

требуется вывод на печать не большого количество одинаковых документов (требования на программные модули, список запланированных работ на ближайшую итерацию).

### 7.2.1. Стадии разработки и состав работ

Для определения трудоемкости внедрения системы необходимо составить перечень основных этапов работ, которые необходимо выполнить. Трудоемкость оценивается по сумме трудоемкостей этапов работ, определяемых в человеко-днях. Перечень стадий и состав работ при разработке системы определяется с помощью ГОСТ 34.601-90. Укрупненный состав работ по стадиям разработки модулей интеграции представлен в таблице (табл.6.1).

Стадия внедрения	Состав выполняемых работ
<b>Техническое задание (ТЗ)</b>	<p>Разработчиком постановки задач выполняются следующие работы:</p> <ul style="list-style-type: none"> <li>- постановка задач;</li> <li>- выбор критериев эффективности;</li> <li>- расчет технико-экономического обоснования.</li> </ul> <p>Разработчиком ПО выполняются следующие работы:</p> <ul style="list-style-type: none"> <li>- определение состава и архитектуры пакета прикладных программ;</li> </ul> <p>Оба исполнителя совместно выполняют работы:</p> <ul style="list-style-type: none"> <li>- выбор языков программирования;</li> <li>- предварительный набор методов выполнения работы;</li> <li>- разработка календарного плана выполнения работ.</li> </ul>
<b>Эскизный проект (ЭП)</b>	<p>Состав работ, выполняемых разработчиком постановки задач:</p> <ul style="list-style-type: none"> <li>- предварительная разработка структуры входных и выходных данных;</li> <li>- разработка общего описания алгоритмов реализации решения задач;</li> <li>- разработка пояснительной записки.</li> </ul> <p>Состав работ, выполняемых разработчиком ПО:</p> <ul style="list-style-type: none"> <li>- консультации с разработчиком постановки задач.</li> </ul> <p>Состав работ, выполняемых разработчиком</p>

	<p>постановки задач и разработчиком ПП совместно:</p> <ul style="list-style-type: none"> <li>- согласование и утверждение эскизного проекта.</li> </ul>
<b>Технический проект (ТП)</b>	<p>Состав работ, выполняемых разработчиком постановки задач:</p> <ul style="list-style-type: none"> <li>- разработка алгоритмов решения задач;</li> <li>- разработка пояснительной записки;</li> <li>- согласование и утверждение технического проекта.</li> </ul> <p>Состав работ, выполняемых разработчиком ПП:</p> <ul style="list-style-type: none"> <li>- разработка структуры программы;</li> <li>- разработка программной документации и передача ее для включения в технический проект.</li> </ul> <p>Состав работ, выполняемых разработчиком постановки задач и разработчиком ПП совместно:</p> <ul style="list-style-type: none"> <li>- уточнение структуры;</li> <li>- анализ и определение формы представления входных и выходных данных;</li> <li>- выбор конфигурации технических средств.</li> </ul>
<b>Рабочий проект (РП)</b>	<p>Состав работ, выполняемых разработчиком постановки задач:</p> <ul style="list-style-type: none"> <li>- комплексная отладка задач;</li> <li>- сдача в опытную эксплуатацию;</li> <li>- разработка проектной документации.</li> </ul> <p>Состав работ, выполняемых разработчиком ПП:</p> <ul style="list-style-type: none"> <li>- программирование;</li> <li>- отладка программ;</li> <li>- описание контрольного примера;</li> <li>- разработка программной документации.</li> </ul> <p>Состав работ, выполняемых разработчиком постановки задач и разработчиком ПП совместно:</p> <ul style="list-style-type: none"> <li>- разработка, согласование программы и методики испытаний;</li> <li>- предварительное проведение всех видов испытаний.</li> </ul>
<b>Внедрение</b>	<p>Состав работ, выполняемых разработчиком постановки задач:</p> <ul style="list-style-type: none"> <li>- подготовка и передача программной документации для сопровождения.</li> </ul> <p>Состав работ, выполняемых разработчиком постановки задач и разработчиком ПП совместно:</p> <ul style="list-style-type: none"> <li>- проверка алгоритмов и программ решения задач;</li> <li>- корректировка документации после опытной эксплуатации ПП.</li> </ul>

Далее расчет трудоемкости внедрения ALM системы будет производиться в соответствии с данным перечнем стадий разработки.

### 7.2.2. Расчет трудоемкости разработки программного продукта

Трудоемкость разработки программного продукта,  $\tau_{\text{пр}}$  может быть определена как сумма величин трудоемкости выполнения отдельных стадий разработки ПП из выражения:

$$\tau_{\text{пр}} = \tau_{\text{мз}} + \tau_{\text{эп}} + \tau_{\text{тп}} + \tau_{\text{рп}} + \tau_{\text{в}}, \text{ где}$$

$\tau_{\text{мз}}$  – трудоемкость разработки технического задания на создание ПП;

$\tau_{\text{эп}}$  – трудоемкость разработки эскизного проекта ПП;

$\tau_{\text{тп}}$  – трудоемкость разработки технического проекта ПП;

$\tau_{\text{рп}}$  – трудоемкость разработки рабочего проекта ПП;

$\tau_{\text{в}}$  – трудоемкость внедрения разработанного ПП.

Трудоемкость разработки технического задания на создание программного продукта,  $\tau_{\text{мз}}$  определяется суммой:

$$\tau_{\text{мз}} = T_{\text{рз}}^3 + T_{\text{рп}}^3, \text{ где}$$

$T_{\text{рз}}^3$  – затраты времени разработчика постановки задач на разработку ТЗ, чел.-дни;

$T_{\text{рп}}^3$  – затраты времени разработчика ПП на разработку ТЗ, чел.-дни.

Значение величин  $T_{\text{рз}}^3$  и  $T_{\text{рп}}^3$  рассчитывают по формулам:

$$T_{\text{рз}}^3 = t_3 \cdot K_{\text{рз}}^3, \quad T_{\text{рп}}^3 = t_3 \cdot K_{\text{рп}}^3, \text{ где}$$

$t_3$  – норма времени на разработку ТЗ на ПП в зависимости от функционального назначения и степени новизны разрабатываемого ПП, чел.-дни;

$K_{\text{рз}}^3$  – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задач на стадии ТЗ (в случае совместной разработки с разработчиком ПП  $K_{\text{рз}}^3 = 0,65$ , в случае

самостоятельной разработки ТЗ  $K_{pz}^3 = 1,0$ );

$K_{pn}^3$  – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком ПО на стадии ТЗ (в случае совместной разработки с разработчиком ПП  $K_{pn}^3 = 0,35$ , в случае самостоятельной разработки ТЗ  $K_{pn}^3 = 1,0$ ).

Работы по разработке постановки задачи и ПП на стадии ТЗ ведутся разработчиком постановки задач совместно с разработчиком ПП, поэтому  $K_{pz}^3 = 0,65$  и  $K_{pn}^3 = 0,35$ . Поскольку система относится к группе новизны В, норма времени на разработку ТЗ принимается равной  $t_3 = 37$ , чел.-дни.

$$T_{pz}^3 = 37 \cdot 0,65 = 24,05, \text{ чел.-дней}; T_{pn}^3 = 37 \cdot 0,35 = 12,95, \text{ чел.-дней.}$$

$$\tau_{m3} = 24,05 + 12,95 = 37, \text{ чел.-дней.}$$

Трудоёмкость разработки эскизного проекта,  $\tau_{эн}$  определяется следующей суммой:

$$\tau_{эн} = T_{pz}^э + T_{pn}^э, \text{ где}$$

$T_{pz}^э$  – затраты времени разработчика постановки задач на разработку ЭП, чел.-дни;

$T_{pn}^э$  – затраты времени разработчика ПП на разработку ЭП, чел.-дни.

Значение величин  $T_{pz}^э$  и  $T_{pn}^э$  рассчитывают по формулам:

$$T_{pz}^э = t_э \cdot K_{pz}^э; T_{pn}^э = t_э \cdot K_{pn}^э,$$

где  $t_э$  – норма времени на разработку ЭП ПП в зависимости от функционального назначения и степени новизны, чел.-дни;

$K_{pz}^э$  – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задач на стадии ЭП (в случае совместной с разработчиком ПП разработки ЭП  $K_{pz}^э = 0,7$ , в случае самостоятельной разработки ЭП  $K_{pz}^э = 1,0$ );

$K_{pn}^э$  – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком ПП на стадии ЭП (в случае совместной разработки с разработчиком постановки задач ЭП  $K_{pn}^э = 0,3$ , в случае самостоятельной разработки ЭП  $K_{pn}^э = 1,0$ ).

Работы по разработке постановки задачи и ПП на стадии ЭП ведутся разработчиком постановки задач совместно с разработчиком ПП, поэтому  $K_{pz}^э = 0,7$  и  $K_{pn}^э = 0,3$ . Поскольку система относится к группе новизны В, норма времени на выполнение работ эскизного этапа создания системой принимается равной  $t_э = 57$  чел.-дней.

$$T_{pz}^э = 57 \cdot 0,7 = 39,9, \text{ чел.-дней}; T_{pn}^э = 57 \cdot 0,3 = 17,1, \text{ чел.-дней.}$$

$$\tau_{эп} = 39,9 + 17,1 = 57, \text{ чел.-дней.}$$

Трудоёмкость разработки технического проекта,  $\tau_{mn}$  зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации и определяется как сумма времени, затраченного разработчиком постановки задач и разработчиком ПП, т.е.:

$$\tau_{mn} = (t_{pz}^m + t_{pn}^m) K_э \cdot K_p, \text{ где}$$

$t_{pz}^m, t_{pn}^m$  – норма времени, затрачиваемого на разработку ТП разработчиком постановки задач и разработчиком ПП соответственно, чел.-дни;

$K_э$  – коэффициент учета вида используемой информации;

$K_p$  – коэффициент учета режима обработки информации.

Значение коэффициента  $K_э$  определяется из выражения:

$$K_э = \frac{K_{II} \cdot n_{II} + K_{НС} \cdot n_{НС} + K_B \cdot n_B}{n_{II} + n_{НС} + n_B}, \text{ где}$$

$K_{II}, K_{НС}, K_B$  – значения коэффициентов учета вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

$n_{II}$ ,  $n_{HC}$ ,  $n_B$  – количество наборов данных переменной, нормативно-справочной информации и баз данных соответственно.

Т.к. группа сложности алгоритма 3, и при этом степень новизны программы В, принимаем норму времени, затрачиваемого на разработку ТП разработчиком постановки задач  $t_{pz}^m = 38$ , чел.-дней, а норму времени, затрачиваемого на разработку ТП разработчиком ПП  $t_{pn}^m = 12$ , чел.-дней,

Учитывая то, что рассчитывается трудоемкость работ на стадии технического проекта, режим обработки информации РВ (реального времени), степень новизны программы В, выбираем  $K_p = 1,36$ .

Количество наборов данных переменной информации  $n_{II} = 0$ .

Количество наборов данных нормативно-справочной информации  $n_{HC} = 0$ .

Количество наборов данных баз данных  $n_B = 1$ .

Учитывая также, что степень новизны программы В, получается:

$$K_s = \frac{1 \cdot 0 + 0,72 \cdot 0 + 2,08 \cdot 1}{0 + 0 + 1} = 2,08$$

Трудоемкость для разработчика постановки задач:

$$\tau_{mn}^{pz} = 38 \cdot 2,08 \cdot 1,36 = 107,5, \text{ чел.-дней.}$$

Трудоемкость для разработчика программного кода:

$$\tau_{mn}^{pn} = 12 \cdot 2,08 \cdot 1,36 = 34, \text{ чел.-дней.}$$

Таким образом, полная трудоемкость стадии технического проекта:

$$\tau_{mn} = (38 + 12) \cdot 2,08 \cdot 1,36 = 141,5, \text{ чел.-дней.}$$

Трудоемкость разработки рабочего проекта,  $\tau_{\partial r}$  зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации, сложности алгоритма функционирования, сложности контроля информации, степени использования готовых

программных модулей, уровня алгоритмического языка программирования и определяется по формуле:

$$\tau_{pn} = K_K \cdot K_P \cdot K_J \cdot K_3 \cdot K_{IA} \cdot (t_{p3}^p + t_{pn}^p), \text{ где}$$

$K_K$  – коэффициент учета сложности контроля информации;

$K_J$  – коэффициент учета уровня используемого алгоритмического языка программирования;

$K_3$  – коэффициент учета степени использования готовых модулей;

$K_{IA}$  – коэффициент учета вида используемой информации и сложности алгоритма ПП.

Значение алгоритма  $K_{IA}$  определяется из выражения:

$$K_{IA} = \frac{K'_П n_П + K'_{HC} n_{HC} + K'_Б n_Б}{n_П + n_{HC} + n_Б}, \text{ где}$$

$K'_П$ ,  $K'_{HC}$ ,  $K'_Б$  – значения коэффициентов учета сложности алгоритма ПП и вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

$t_{p3}^p$ ,  $t_{pn}^p$  – норма времени, затраченного на разработку РП на алгоритмическом языке высокого уровня разработчиком постановки задач и разработчиком ПП соответственно, чел.-дни.

Т.к. по степень сложности контроля входной информации подсистема относится к группе 12, а по степени сложности контроля выходной информации – к группе 22, то коэффициент учета сложности контроля информации можно принять равным  $K_K = 1$ .

Учитывая то, что рассчитывается трудоемкость работ на стадии рабочего проекта, режим обработки информации РВ, степень новизны программы В, выбираем  $K_P = 1,44$ .

Т.к. используемый язык С# относится к группе алгоритмических языков высокого уровня, коэффициент учета уровня используемого



алгоритмического языка программирования  $K_я = 1,0$ .

Степень использования готовых программных модулей около 30%, следовательно, коэффициент учета степени использования готовых модулей  $K_3 = 0,5$ .

Алгоритмы относятся к третьей группе сложности, а степень новизны программной продукции В, следовательно: коэффициент учета сложности алгоритма ПП и вида используемой информации для переменной информации  $K'_л = 1$ ;

Коэффициент учета сложности алгоритма ПП и вида используемой информации для нормативно-справочной информации  $K'_{нс} = 0,48$ ;

коэффициент учета сложности алгоритма ПП и вида используемой информации для баз данных  $K'_б = 0,4$ .

Количество наборов данных переменной информации  $n_л = 0$ ,

Количество наборов данных нормативно-справочной информации  $n_{нс} = 0$ ,

Количество наборов данных баз данных  $n_б = 1$ .

Следовательно, коэффициент учета вида используемой информации и сложности алгоритма ПП:

$$K_{иа} = \frac{1,0 \cdot 0 + 0,48 \cdot 0 + 0,4 \cdot 1}{0 + 0 + 1} = 0,4$$

Количество разновидностей форм входной информации – 1, а видов форм выходной информации – 2, соответственно принимаем следующие нормы времени, затраченного на разработку программного кода на алгоритмическом языке высокого уровня, для разработчика постановки задач  $t_{пз}^p = 10$ , чел.-дней, а для разработчика программного кода  $t_{пн}^p = 54$ , чел.-дней.

Трудоемкость для разработчика постановки задач:

$$\tau_{пн}^{пз} = 1,0 \cdot 1,44 \cdot 1,0 \cdot 0,5 \cdot 0,4 \cdot 10 = 2,9, \text{ чел.-дней.}$$

Трудоемкость для разработчика программного кода:

$$\tau_{pn}^{pn} = 1,0 \cdot 1,44 \cdot 1,0 \cdot 0,5 \cdot 0,4 \cdot 54 = 15,6, \text{ чел.-дней.}$$

Полная трудоемкость разработки рабочего проекта:

$$\tau_{pn} = 2,9 + 15,6 = 18,5, \text{ чел.-дней}$$

Трудоемкость выполнения стадии внедрения,

$$\tau_B = (t_{pz}^B + t_{pn}^B) \cdot K_K \cdot K_P \cdot K_3, \text{ где}$$

$t_{pz}^B, t_{pn}^B$  – норма времени, затрачиваемого разработчиком постановки задач и разработчиком ПП соответственно на выполнения процедуры внедрения ПП, чел.-дни.

Работа разработчика ПП на стадии внедрения разработанной подсистемы заключается в установке разработанного ПО, а так же ПО, требуемого для его работы, и контроле правильности его функционирования.

Количество разновидностей форм входной информации – 1, а видов форм выходной информации – 2, соответственно принимаем следующие нормы времени, затраченного на разработку программного кода на алгоритмическом языке высокого уровня, для разработчика постановки задач  $t_{pz}^B = 11$ , чел.-дней, а для разработчика программного кода  $t_{pn}^B = 12$ , чел.-дней.

Учитывая то, что рассчитывается трудоемкость работ на стадии внедрения, режим обработки информации РВ, степень новизны программы В, выбираем  $K_P = 1,26$ .

Степень использования готовых программных модулей более 60%, следовательно, коэффициент учета степени использования готовых модулей  $K_3 = 0,5$ .

Т.к. по степени сложности контроля входной информации подсистема относится к группе 12, а по степени сложности контроля выходной

информации – к группе 22, то коэффициент учета сложности контроля информации можно принять:  $K_K = 1,0$ .

Трудоемкость для разработчика постановки задач:

$$\tau_{pn}^{pz} = 11 \cdot 1,0 \cdot 1,26 \cdot 0,5 = 6,93, \text{ чел.-дней.}$$

Трудоемкость для разработчика программного кода:

$$\tau_{pn}^{pn} = 12 \cdot 1,0 \cdot 1,26 \cdot 0,5 = 7,56, \text{ чел.-дней.}$$

Таким образом, полная трудоемкость внедрения:

$$\tau_B = 6,93 + 7,56 = 14,5, \text{ чел.-дней.}$$

Полученные данные по трудоемкостям выполнения работ приведены в виде таблицы:

Название работ	Трудоемкость проведения работ, чел.-дни
Разработка технического задания, $\tau_{mz}$	37
Разработка эскизного проекта, $\tau_{en}$	57
Разработка технического проекта, $\tau_{mn}$	142
Разработка рабочего проекта, $\tau_{pn}$	19
Внедрение, $\tau_{\epsilon}$	15
Общее время выполнения проекта, $\tau_{nn}$	270

Планирование и контроль над ходом выполнения проекта по разработке ПП проводят по календарному графику выполнения работ. Проект осуществляет небольшим коллективом, состоящим из двух человек.

С целью осуществления планирования выполнения работ построим ленточный график (диаграмму Ганта). Для построения ленточного графика необходимо знать срок начала работ, срок окончания работ и количество

работников, участвующих на каждом этапе разработки.

Для того чтобы определить продолжительность всех работ по созданию ПП рассчитаем продолжительность каждого этапа, исходя из соответствующих трудоемкостей и количества занятых участников на каждом этапе.

Расчёт производится по формуле:

$$T_i = \frac{\tau_i + Q}{n_i},$$

где  $\tau_i$  – трудоемкость  $i$ -й работы, чел.-дни;

$Q$  – трудоемкость дополнительных работ, выполняемых исполнителем, чел.-дни;

$n_i$  – количество исполнителей, выполняющих  $i$ -ю работу, чел.

Трудоемкость дополнительных работ, выполняемых исполнителем на каждом этапе принимается равной 0.

Для построения календарного плана необходимо перевести рабочие дни в календарные. Для этого длительность каждого этапа нужно разделить на поправочный коэффициент  $K = 0,7$ .

Результаты расчёта приведены в таблице 6.4. Количество исполнителей и требования к их квалификации на каждой стадии разработки подсистемы планирования представлены ниже:

- Разработка ТЗ.

На этой стадии работают два исполнителя. Это люди с наивысшей квалификацией, имеющие опыт работы на всех стадиях разработки подсистемы.

- Эскизное проектирование.

На этой стадии работают два исполнителя. Уровень их квалификации должен позволять оценивать возникающие альтернативные варианты проектов и выбрать оптимальный.

- Техническое проектирование.

На этой стадии работают два исполнителя. Они должны быть специалистами по математическим и техническим методам решения задач.

- Рабочее проектирование.

На этой стадии работают два исполнителя – специалисты по кодированию алгоритмов, полученных на стадии технического проектирования.

- Внедрение.

На этой стадии работают два исполнителя. Уровень их квалификации должен позволять проверять и корректировать алгоритмы и программы решения задач.

Название работ	Персонал	Трудоемк., дней	Трудоемк., календарн . дни	Время выполнения , календарн. дни
Разработка технического задания.	Проектировщик	24	34	34
	Программист	13	19	
Разработка эскизного проекта.	Проектировщик	40	58	58
	Программист	17	24	
Разработка технического проекта.	Проектировщик	108	154	154
	Программист	34	49	
Разработка рабочего проекта.	Проектировщик	3	4	23
	Программист	16	23	
Внедрение.	Проектировщик	7	10	11
	Программист	8	11	
<b>Итого</b>		<b>270</b>	<b>384</b>	<b>280</b>

### 7.3. Расчет себестоимости модулей интеграции

Стоимость продукции представляет собой стоимостную оценку используемых в процессе производства продукции природных ресурсов, сырья, материалов, топлива, основных фондов, трудовых ресурсов, а так же других затрат на ее производство и реализацию. Произведем расчет сметной стоимости по каждой статье затрат.

Основная заработная плата. В статью включается основная заработная плата всех исполнителей, непосредственно занятых разработкой данного ПП,

с учетом их должностного оклада и времени участия в разработке. Расчет ведется по формуле:

$$C_{30} = \sum_i \frac{z_i}{d} \tau_i, \text{ где}$$

$z_i$  – среднемесячный оклад  $i$ -го исполнителя, руб.;

$d$  – среднее количество рабочих дней в месяце;

$\tau_i$  – трудоемкость работ, выполняемых  $i$ -м исполнителем, чел.-дни.

Среднемесячный оклад разработчика постановки задачи<sup>2</sup> (системный архитектор)

$$z_1 = 125000 \text{ руб.}$$

Среднемесячный оклад разработчика ПП на языке программирования С#<sup>3</sup> (программиста)

$$z_3 = 77000 \text{ руб.}$$

Среднее количество рабочих дней  $d = 20$  дней.

Название работ	Персонал	Трудоемкость, дней	Оплата труда, рубл. дни
Разработка технического задания на создание ПП	Проектировщик	24	150000
	Программист	13	50000
Разработка эскизного проекта.	Проектировщик	40	250000
	Программист	17	65500
Разработка технического проекта.	Проектировщик	108	675000
	Программист	34	131000
Разработка рабочего проекта.	Проектировщик	3	19000
	Программист	16	61500
Внедрение ПП	Проектировщик	7	44000
	Программист	8	31000
<b>Итого</b>			<b>1477000</b>

<sup>2</sup> Данные взяты из интернет-ресурса [rabota.yandex.ru](http://rabota.yandex.ru) для города Москвы на 2ой квартал 2013 г.

<sup>3</sup> Данные взяты из интернет-ресурса [rabota.yandex.ru](http://rabota.yandex.ru) для города Москвы на 2ой квартал 2013 г.

Дополнительная заработная плата. В этой статье учитываются все выплаты непосредственным исполнителям за время (установленное законодательством), не проработанное на производстве, в том числе: оплата очередных отпусков, компенсации за недоиспользованный отпуск, оплата льготных часов подросткам и др. Расчет ведется по формуле:

$$C_{зд} = C_{зо} \cdot \alpha_d,$$

где  $\alpha_d$  – коэффициент отчислений на дополнительную зарплату,  $\alpha_d = 0,2$ .

$$C_{зд} = 1477000 \cdot 0,2 = 295400, \text{ руб.};$$

$$C_3 = 1477000 + 295400 = 1772400, \text{ руб.}$$

Отчисления в фонды социального страхования. В статье учитываются отчисления в бюджет социального страхования по установленному законодательством тарифу от суммы основной и дополнительной заработной платы:

$$C_{cc} = \alpha_{cc} \cdot C_3,$$

где  $\alpha_{cc}$  – на.

Производство ПП относится к производству 05 класса опасности, соответственно ставка обязательного страхования от несчастных случаев на производстве составляет 0,6%. Отчисления на социальные нужды включают в себя отчисления на социальное страхование, а также – на медицинское страхование, в фонд занятости и в пенсионный фонд, итого 30% от основной заработной платы. Итого 30,6 %

Итого, затраты на уплату в фонды социального страхования:  
 $C_{cc} = 1772400 \cdot 0.306 = 542354, \text{ руб.}$

Амортизационные отчисления. В статье отражается сумма амортизационных отчислений на полное восстановление основных производственных фондов, исчисленная из балансовой стоимости и

утвержденных в установленном порядке норм, включая и ускоренную амортизацию их активной части, производимую в соответствии с законодательством.

$$C_A = \frac{A \cdot T}{F_D}, \text{ где}$$

$A$  – годовые амортизационные отчисления с одной ЭВМ;

$F_D$  – действительный годовой фонд рабочего времени на ЭВМ, час./год;

$T$  – время работы оборудования (в нашем случае – суммарная трудоемкость работ по созданию системы).

Средняя цена рабочего места программиста на C++/C#/Java (на 1 мая 2013 года) – 62000 руб. Амортизационные отчисления составляют – 20%.

Годовой фонд рабочего времени на ПЭВМ - 5-ти дневная неделя, 8-и часовой рабочий день, отсюда получаем  $F_D = 2080$  часов.

Требуемое ПО - ПО является бесплатным и распространяется по GNU GPL-лицензии, а также по партнерским соглашениям, имеющимся у заказчика внедрения. Стоимость ПО принимаем равной 0 руб.

Соответственно стоимость комплекта, необходимого для работы одного человека составляет 62000 руб.

Амортизационные отчисления на этапах разработки представлены в таблице:

Название работ	Кол-во требуемых комплектов ПК+ПО	Стоимость оборудования, руб	Продолжительность работ, календарн. дней	Амортизационные отчисления, руб.
Разработка технического задания.	2	124000	34	3243
Разработка эскизного проекта.	2	124000	58	5532
Разработка технического проекта.	2	124000	154	14689



Разработка рабочего проекта.	2	124000	23	2194
Внедрение.	2	124000	11	1049
<b>Итого</b>				<b>26707</b>

Накладные расходы. В статье учитываются затраты на общехозяйственные расходы, непроизводительные расходы и расходы на управление. Накладные расходы определяют в процентном отношении к основной заработной плате:

$$C_H = \alpha_H \cdot C_{30},$$

где  $\alpha_H$  – коэффициент накладных расходов,  $\alpha_H = 2,0$ .

$$C_H = 2,0 \cdot 1477000 = 2954000, \text{ руб.}$$

Сложив составляющие статей расходов, получаем общую стоимость ПП для подсистемы. Данные представлены в таблице(табл.7.6):

№	Наименование статьи	Сметная стоимость, руб.
1	Затраты на оплату труда, $C_3$	1477000
2	Отчисления в фонды социального страхования, $C_{CC}$	542354
3	Амортизационные отчисления, $C_A$	26707
4	Накладные расходы, $C_H$	2954000
<b>Итого, <math>C_{ПП}</math>:</b>		<b>5000061 ~ 5000000</b>

#### 7.4.Расчет себестоимости приобретаемой части ALM системы

Список необходимого ПО, полученный на этапе планирования архитектуры системы:

- Team Foundation Server 2012
- SQL Server Standard 2012
- SharePoint Server Enterprise 2013
- Microsoft Server 2012 Standard x 2

Список оборудования, которое необходимо закупить для внедрения ALM системы:

- HP Proliant DL380 G8(662257-421)
- APC Smart-UPS X 3000VA Rack/Tower LCD 200-240V with Network Card

Лицензии на ПО корпорации Microsoft получены по партнерской программе, стоимость ПО принимаем равной 0 руб.

Затраты на необходимое оборудование:

Наименование	Кол-во	Стоимость
HP Proliant DL380 G8(662257-421)	1	403500
APC Smart-UPS X 3000VA Rack/Tower LCD 200-240V with Network Card	1	80 766
<b>Итого:</b>		<b>484266</b>

#### 7.5. Заключение организационно-экономической части

В рамках организационно-экономической части были проведены расчеты себестоимости внедрения ALM Системы в ДРПО ЗАО «Элвис-Неотек». Были исследованы и рассчитаны следующие статьи затрат: материальные затраты, основная заработная плата исполнителей, дополнительная заработная плата исполнителей, отчисления на социальное страхование, амортизационные отчисления, накладные расходы.

При проведении расчетов была получена планируемая продолжительность разработки программного продукта, составляющая 384 календарных дня. Для снижения продолжительности, возможно, использовать наложение работ смежных этапов проектирования ПП.

Был проведен расчет стоимости программного продукта с учетом всех статей затрат, задействованные при разработке, включая затраты на основную и дополнительную заработанную плату исполнителей, отчисления на единый социальный налог, амортизационные отчисления и накладные расходы.

В итоге были получены суммарные затраты внедрения ALM Системы, составляющая приблизительно 5484327 рублей.

## **8. Требования безопасности при разработке модулей интеграции ALM Системы**

### **8.1. Введение**

В данном разделе дипломного проекта рассматриваются и анализируются опасные и вредные факторы при разработке модулей интеграции для программного продукта Team Foundation Server в рамках внедрения ALM Системы. Описываются мероприятия по обеспечению безопасности и безвредных условий труда, приводятся рекомендации по способам и методам ограничения действия вредных и исключению воздействия опасных факторов на разработчиков, проводящих деятельность в компьютерной сфере. При этом на человека, работающего в зоне действия ПЭВМ, влияет большое количество вредных факторов, состояние которых необходимо контролировать. Факторы, действующие на разработчиков [7]:

- уровень шума, источниками которого являются вентиляционные устройства ПЭВМ, устройства ввода-вывода, агрегаты кондиционирования и вентилирования воздуха, другие электрические приборы;
- уровни электростатического и электромагнитного излучения, источниками которого являются видеотерминалы;
- параметры микроклимата;
- освещенность в помещении и на рабочем месте.

<b>Тип</b>	<b>Опасные</b>	<b>Вредные</b>
Физические	1. Поражение электротоком 2. Пожарная опасность	1. Повышенный уровень шума агрегата кондиционирования и вентилирования воздуха 2. Повышенная или пониженная температура воздуха рабочей зоны 3. Повышенная или пониженная влажность воздуха 4. Повышенная или пониженная ионизация воздуха 5. Повышенный уровень электромагнитных излучений 6. Недостаточная освещенность рабочей зоны
Психофизиологические		1. Напряжение зрения 2. Напряжение внимания 3. Длительные статические нагрузки

Кроме того очевидна необходимость проведения мероприятий не только для улучшения состояния рабочего места по перечисленным вредным факторам, но и для удовлетворения требованиям эргономики, пожаро- и электробезопасности.

Ниже приведен анализ соответствия требованиям, определенным в СанПиН 2.2.2/2.4.1340-03 [9], к рабочим местам одной из лабораторных аудиторий ЗАО «Элвис-Неотек». Анализ проводился без использования специального измерительного оборудования, выполнен на основе наблюдений, качественного определения соответствия требованиям и соответствующих предположений. Выводы по результатам анализа носят рекомендательный характер.

## **8.2. Требования безопасности при отработке программного продукта**

### **8.2.1. Требования к персональным электронно-вычислительным машинам**

ПЭВМ должны соответствовать требованиям настоящих санитарных правил, и каждый их тип подлежит санитарно-эпидемиологической экспертизе с оценкой в испытательных лабораториях, аккредитованных в

установленном порядке. Рассматриваются следующие требования.

Перечень продукции и контролируемых гигиенических параметров вредных и опасных факторов представлены.

Допустимые уровни звукового давления и уровней звука, создаваемых ПЭВМ.

Временные допустимые уровни электромагнитных полей (ЭМП), создаваемых ПЭВМ.

Допустимые визуальные параметры устройств отображения информации.

Концентрации вредных веществ, выделяемых ПЭВМ в воздух помещений, не должны превышать предельно допустимых концентраций (ПДК), установленных для атмосферного воздуха.

Мощность экспозиционной дозы мягкого рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана и корпуса ВДТ (на электронно-лучевой трубке) при любых положениях регулировочных устройств не должна превышать 1 мкЗв/час (100 мкР/час).

Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана ВДТ. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4-0,6 и не иметь блестящих деталей, способных создавать блики.

Конструкция ВДТ должна предусматривать регулирование яркости и контрастности,

Документация на проектирование, изготовление и эксплуатацию ПЭВМ не должна противоречить требованиям настоящих Санитарных правил.

Результаты анализа:

- используется жидкокристаллическое средство отображения, которое при разрешении 1024x768 обеспечивает частоту обновления 75 Гц
- Конструкция ВДТ, дизайн соответствуют современным

требованиям.

- Поворот корпуса ВДТ в горизонтальной и вертикальной плоскости соответствует нормам. Корпуса ВДТ и ПЭВМ окрашены в мягкие тона и имеют матовую поверхность с коэффициентом отражения, не превышающим норму, монитор имеет антибликовое и антистатическое покрытие, а также снабжен ручками регулировки яркости и контраста.
- Монитор не оборудован защитным фильтром на экране, т.к. в этом нет острой необходимости. Монитор соответствует стандарту MPR II, имеет низкую радиационную активность, уменьшает влияние электрических и магнитных полей на человека. В монитор также встроена система управления энергопотребления, которая позволяет снижать потребляемую мощность компьютера во время простоя.
- Конструкция клавиатуры выполнена в виде отдельного устройства с возможностью свободного перемещения, имеет опорные устройства, позволяющие менять угол наклона клавиатуры. Основные параметры клавиатуры удовлетворяют нормативам.

#### **8.2.2. 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 (пункт в редакции, введенной в действие с 1 июля 2007 года Изменением N 1 от 25 апреля 2007 года, - см. предыдущую редакцию).

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 др.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) должна составлять не менее 4,5 м<sup>2</sup>.

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.

Результаты анализа:

- Помещение имеет искусственное комбинированное и естественное освещение, окна ориентированы на север. Искусственное освещение поддерживается 9х4 люминесцентными лампами ЛД-18. Площадь комнаты (8х6 метров) равна 48 м<sup>2</sup>. Высота потолка 3,6 м. Число рабочих мест - 10. Площадь на одно рабочее место равна 4,8 м<sup>2</sup>, что соответствует санитарным нормам: не менее 4,5 м<sup>2</sup> площади.

- Окна оснащены жалюзи.

- Комната имеет встроенные шкафы и полки для хранения одежды и вещей. Она так же оборудована системой центрального отопления и кондиционирования.

- Половое покрытие - линолеум.

- Помещение имеет один вход и не имеет смежных с ним помещений. Расположено на втором этаже здания.

- Помещение не граничит с помещениями с высоким уровнем шума и вибрации.

- Помещение оборудовано центральным защитным заземлением.

Стены помещения обклеены обоями бежевого тона.



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

В помещениях всех типов образовательных учреждений, где расположены ПЭВМ, должны обеспечиваться оптимальные параметры микроклимата.

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

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

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 (ГН 2.1.6.1338-03).

Результаты анализа:

- В теплое время года в помещении температура воздуха составляет от 20 до 23 С. Подвижность воздуха – от 0 до 0.1 м/с. Влажность воздуха – 55-60%.

- В холодное время года в помещении температура воздуха составляет 19-21 °С, подвижность воздуха менее 0.1 м/с, влажность воздуха – 55-60%, что соответствует нормам.

- Содержание вредных химических веществ в воздухе соответствует содержанию этих веществ на улице, т.к. проводится периодическое проветривание помещений, однако, фильтры кондиционера предотвращают попадание пыли с улицы в помещение. Поскольку в Москве (особенно в центральной части города) уличный воздух не является чистым, то воздух в нашем помещении не соответствует нормам.

- Содержание аэронов в воздухе помещения близко к оптимуму и равно соответственно: положительных – 1000-1200, а отрицательных – 1700-

2000.

В помещении присутствует система центрального кондиционирования. Работы за ПЭВМ, выполняемые в аудитории, не влекут за собой выделения пыли, влаги, загрязняющих веществ. Запыленность низкая. Ежедневно проводятся влажные уборки.

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

В помещениях всех учреждений, где расположены ПЭВМ, уровни шума не должны превышать допустимых значений, установленных для жилых и общественных зданий.

Согласно СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» для работы, связанной с обучением, уровень шума не должен превышать значения, указанного в приложении 3 (таблица 5):

Снизить уровень проникающего шума в помещениях с ВДТ и ПЭВМ при открытых окнах, где проходят занятия, возможно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63 - 8000 Гц (Допустимые уровни звука проникающего шума указаны в приложении 3 (таблица 5), согласно СН 2.2.4/2.1.8.562-96). Дополнительным звукопоглощением служат занавеси из плотной ткани. Ширина занавеси должна быть в 2 раза больше ширины окна.

В помещениях всех типов образовательных учреждений, в которых эксплуатируются ПЭВМ, уровень вибрации не должен превышать допустимых значений для жилых и общественных зданий в соответствии с действующими санитарно-эпидемиологическими нормативами ("Производственная вибрация, вибрация в помещениях жилых и общественных зданий" СН 2.2.4/2.1.8.566-96).

Для снижения вибрации в помещениях оборудование, аппараты и приборы необходимо устанавливать на амортизирующие прокладки. Также могут быть использованы средства индивидуальной защиты.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне

помещений с ПЭВМ.

#### **Результаты анализа:**

- Уровень шума в аудитории может превышать 50 дБА, т.к. ведутся лекционные занятия и проводится разъяснительная работа со студентами. На потолке установлено звукопоглощающее покрытие. Дополнительно, в качестве средства звукопоглощения в помещении, можно отметить, матерчатые жалюзи на окнах, однако их ширина не превосходит ширину окна в два раза;
- Вибрация на рабочих местах отсутствует. Выявлен источник вибрации в помещении - система кондиционирования. Типовой расчет виброизоляции для системы кондиционирования приведен в пункте 7.3.
- Уровень вибрации в помещении от оборудования не значителен (мало ощутим), поэтому амортизирующие прокладки или средства индивидуальной защиты не используются;
- Печатающие устройства – лазерные принтеры практически бесшумны и размещены в помещении.

#### **8.2.4. Требования к освещению на рабочих местах, оборудованных ПЭВМ**

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы 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 (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

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 блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более  $200 \text{ кд/м}^2$ .

Следует ограничивать отраженную блёскость на рабочих поверхностях (экран, стол, клавиатура и др.) 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  $40 \text{ кд/м}^2$  и яркость потолка не должна превышать  $200 \text{ кд/м}^2$ .

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 учебных помещениях не более 15.

Яркость светильников общего освещения в зоне углов излучения от  $50^\circ$  до  $90^\circ$  с вертикалью в продольной и поперечной плоскостях должна составлять не более  $200 \text{ кд/м}^2$ , защитный угол светильников должен быть не менее  $40^\circ$ .

Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее  $40^\circ$ .

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 поверхностями стен и оборудования 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных.

Для освещения помещений с ПЭВМ следует применять светильники с

зеркальными параболическими решетками, укомплектованными электронными пуско-регулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с ЭПРА, состоящими из равного числа опережающих и отстающих ветвей.

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 локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса ( $K_z$ ) для осветительных установок общего освещения должен приниматься равным 1,4.

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.

#### **Результаты анализа:**

- Требования к естественному освещению не нарушены. Расположение рабочего места по отношению к световым проемам является допустимым: мониторы ориентированны боковой стороной к световым проемам.

- Бликов от освещения на поверхности мониторов не наблюдается.
- Светильники местного освещения отсутствуют.
- Искусственное освещение поддерживается 9х4

люминесцентными лампами ЛД-18, которые расположены на потолке помещения. Светильники оснащены зеркальными параболическими решетками.

- Общее освещение выполнено в виде 3 прерывистых линий. Столы, по возможности, ориентированы так, чтобы свет подал на край, обращенный к оператору.
- Производится своевременная замена перегоревших ламп в помещении.
- В помещении не реже двух раз в год проводится чистка стекол оконных рам и светильников.

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

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей, а также в помещениях образовательных учреждений.

Методика проведения инструментального контроля уровней ЭМП на рабочих местах пользователей ПЭВМ.

##### **Результаты анализа:**

- Поскольку рабочие места внедряемой системы (места в лаборатории) оборудованы жидкокристаллическими мониторами, то установка поглощающих или отражающих экранов не требуется.

#### **8.2.6. Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах**

Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах.

##### **Результаты анализа:**

- Поскольку рабочие места внедряемой системы (места в лаборатории) оборудованы жидкокристаллическими мониторами с настраиваемыми визуальными параметрами, то все соответствуют указанным нормам.

#### **8.2.7. Общие требования к организации рабочих мест пользователей ПЭВМ**

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 2,0м,

а расстояние между боковыми поверхностями видеомониторов - не менее 1,2м.

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.

#### **Результаты анализа:**

- расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), составляет 1,6 м. Расстояние между боковыми поверхностями видеомониторов - равно 1,4 м, что в неполной мере соответствует нормам.
- Экран видеомонитора находится от глаз пользователя на расстоянии 600 - 700 мм
- При работе с ПЭВМ используется вариант офисного стула, конструкция которого не предполагает наличие поворотных и наклонных регулируемых механизмов фиксации положения тела оператора, что не соответствует указанным нормам.
- Поверхность сиденья, спинки и других элементов кресла является полумягкой.

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

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

Конструкция одноместного стола для работы с ПЭВМ должна предусматривать:

- две отдельные 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° с надежной фиксацией в оптимальном рабочем положении (12-15°);
- 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 находиться 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, не соответствующего 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  $\pm 5^\circ$ , допустимое  $\pm 10^\circ$ .

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

#### **Результаты анализа:**

• Рабочий стол представляет собой парту со следующими параметрами:

- высота рабочей поверхности стола составляет 800 мм, пространство для ног высотой 780 мм, шириной - 1400 мм, глубиной на уровне колен – 800 мм и на уровне вытянутых ног - 850 мм.
- Разделение поверхностей отсутствует.
- Ящики отсутствуют.
- Стояк для электропитания отсутствует.

Что не соответствует нормам.

• Конструкция рабочего кресла предусматривает:

- ширину 400 мм и глубину 400 мм поверхности сиденья
- высота сиденья составляет 450 мм.
- поверхность сиденья с закругленным передним краем
- высота нижнего края спинки над сиденьем – 170 мм.
- высоту опорной поверхности спинки 200 мм, ширину - 400 мм и радиус кривизны горизонтальной плоскости - 300 мм
- угол наклона сиденья – 1-5 градусов.

- угол наклона спинки в вертикальной плоскости составляет 5-10 градусов

Что, в целом, соответствует нормам.

- Рабочее место пользователя ПЭВМ не оборудовано подставкой для ног.
- Клавиатура расположена на поверхности стола на расстоянии 100-200 мм от края, обращенного к пользователю.

### **8.2.9. Требования к электроснабжению, электробезопасности на рабочих местах, оборудованных ПЭВМ**

Классификация помещений по опасности поражения электрическим током:

Помещения I класса. Без повышенной опасности.

- нет условий повышенной и особой опасности.

Помещения II класса. Помещения повышенной опасности поражения электрическим током.

- повышенная температура воздуха ( $t \geq + 35 \text{ C}$ );
- повышенная влажность ( $> 75 \%$ );
- наличие токопроводящей пыли;
- наличие токопроводящих полов (металлические, земляные, железобетонные, кирпичные);
- наличие электрических установок (заземленных) — возможности прикосновения одновременно и к электрической установке и к заземлению или к двум электрическим установкам одновременно.

Помещения III класса. Особо опасные помещения.

- 100 % влажность;
- наличие активной среды;
- наличие одновременно двух и более условий повышенной опасности.

Наше помещение относится к помещению I класса опасности и требует заземления или зануления.

Питание устройств ПЭВМ осуществляется от однофазной сети переменного тока с частотой 50 Гц, и напряжением 220 В.

В целях обеспечения необходимой электробезопасности при проведении

работ в помещениях с ПЭВМ, необходимо выполнять следующие требования:

- для обеспечения работы операторов ПЭВМ необходимо исключить возможность случайного соприкосновения людей с токонесущими частями оборудования. Это достигается путем изоляции токоведущих частей ЭВМ и приборов и размещения их в недоступных зонах;
- не оставлять ЭВМ и другое оборудование под напряжением без наблюдения.

#### **Результаты анализа:**

- Имеется автоматический выключатель;
- Все электрооборудование имеет рабочую изоляцию;
- К электрооборудованию допускаются лица не моложе 18 лет и прошедшие инструктаж по техники безопасности;
- Регулярно проводится технический осмотр оборудования.
- ПЭВМ подключены к розеткам, имеющим заземление.

#### **8.2.10. Пожаробезопасность**

Помещение (аудитория), где функционирует разработанная подсистема, относится к категории В — пожароопасные помещения, в котором имеются твердые сгораемые вещества, способные только гореть, но не взрываться при контакте с кислородом воздуха. Наиболее вероятной причиной пожара является неисправность электрооборудования и электросетей. При эксплуатации ЭВМ возможны возникновения следующих аварийных ситуаций: короткие замыкания, перегрузки, повышение переходных сопротивлений в электрических контактах, перенапряжение, возникновение токов утечки. При возникновении аварийных ситуаций происходит резкое выделение тепловой энергии, которая может явиться причиной возникновения пожара. Требования к пожаробезопасности зданий и сооружений определяются согласно СНиП 21.01-97.

Для снижения вероятности возникновения пожара необходимо проводить различные профилактические мероприятия:

- Организационные — правильная эксплуатация электрооборудования,

правильное содержание зданий и помещений;

- Технические — соблюдение противопожарных правил и норм, норм при проектировании зданий, при устройстве отопления, вентиляции освещения, правильное размещение оборудования;

Для снижения вероятности возникновения и распространения пожара на ранней стадии необходимо:

- установить пожарную сигнализацию с системой оповещения работников, дежурного по объекту и, желательно, автоматическое оповещение противопожарных служб;

- иметь в наличии несколько ручных углекислотных огнетушителей (например, огнетушитель марки ОУ-3).

#### **Результаты анализа:**

- В помещении предусмотрена противопожарная сигнализация и индивидуальные средства пожаротушения: ОУ-3.

### **8.3. Типовой расчет виброизоляции для системы кондиционирования**

В ходе обследования помещения, где происходит эксплуатация рассматриваемой системы, было обнаружено оборудование, производящее вибрации – кондиционер с частотой вращения вентилятора 900 оборотов в минуту. Проведем расчет жесткости виброизоляторов и их количества.

Оптимальное соотношение частоты вращения вентилятора в кондиционере и его собственной частоты определяется формулой (согласно

ГОСТ 12.4.093-80):  $\frac{f}{f_0} = 3$

Частота вращения вентилятора, рассчитывается по формуле:

$$f = \frac{n}{60} = \frac{900}{60} = 15 \text{ Гц}$$

Таким образом, собственная частота:  $f_0 = \frac{f}{3} = \frac{15}{3} = 5 \text{ Гц}$

С другой стороны, собственная частота рассчитывается по формуле:

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{q_1}{m}}, \text{ где}$$

$q_1$  - вертикальная жесткость виброизолятора, Н/см

$N$  - число виброизоляторов, штук

$m$  - масса виброизолятора, кг

Выразим и рассчитаем соотношение  $\frac{q_1 \cdot N}{m}$ :

$$\frac{q_1 \cdot N}{m} = (2 \cdot \pi \cdot f_0)^2 = (2 \cdot 3.1415 \cdot 5)^2 = 986.9$$

Выбираем виброизолятор ДО-38.

Обозначение	Нагрузка Р, Н		Вертикальная жесткость, Н/см	Высота в свободном состоянии	Осадка пружины под нагрузкой, мм		Число рабочих витков	Масса, кг
	Рабочая (Р <sub>раб.</sub> )	Предельная (Р <sub>пр.</sub> )			Р <sub>раб.</sub>	Р <sub>пр.</sub>		
ДО38	122	152	45	72	27	33,7	5,6	0,3

Для 6 опор-виброизоляторов, согласно таблице 7.2:

$$\frac{q_1 \cdot N}{m} = \frac{45 \cdot 6}{0.3} = 900$$

$$\text{собственная частота } f_0 = \frac{1}{2 \cdot \pi} \sqrt{\frac{q_1 \cdot N}{m}} = \frac{1}{2 \cdot \pi} \cdot \sqrt{900} = 4.78 \text{ Гц}$$

Таким образом, соотношение частоты вращения вентилятора в кондиционере и его собственной частоты будет равно:

$$\frac{f}{f_0} = \frac{15}{4.77} = 3.14$$

Это соотношение близко к оптимальному.

Таким образом, для устранения в помещении вибраций, создаваемых кондиционером, необходимо установить его на 6 опор-виброизоляторов марки ДО-38.

#### 8.4. Утилизация носителей информации

Утилизация носителей информации осуществляется в соответствии с ГОСТ Р ИСО/МЭК 17799-2005. Национальный стандарт Российской Федерации. Информационная технология. Практические правила управления информационной безопасностью (утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 29

декабря 2005 г. N 447-ст, настоящий стандарт идентичен международному стандарту ИСО/МЭК 17799:2000 “Информационная технология. Практические правила управления информационной безопасностью” (ISO/IEC 17799:2000 “Information technology. Code of practice for security management”).

Диски до дробления:



Рис.7.2. Диски до утилизации

и после:



Рис.7.3. Диски после дробления

Извлечение: пункт 8.6.2 ГОСТ: «Утилизация носителей информации»

«Носители информации по окончании использования следует надежно и безопасно утилизировать. Важная информация может попасть в руки посторонних лиц из-за небрежной утилизации носителей данных. Чтобы свести к минимуму такой риск, должны быть установлены формализованные процедуры безопасной утилизации носителей информации. Для этого необходимо предусматривать следующие мероприятия:

a) носители, содержащие важную информацию, следует хранить и утилизировать надежно и безопасно (например, посредством сжигания/измельчения). Если носители планируется использовать в пределах организации для других задач, то информация на них должна быть уничтожена;

b) ниже приведен перечень объектов, в отношении которых может потребоваться безопасная утилизация:

c) бумажные документы;

d) выводимые отчеты;

e) оптические носители данных (все разновидности, в том числе носители, содержащие программное обеспечение, поставляемое производителями);

f) тексты программ;

g) тестовые данные;

h) системная документация;

i) может оказаться проще принимать меры безопасной утилизации в отношении всех носителей информации, чем пытаться сортировать носители по степени важности;

j) многие организации предлагают услуги по сбору и утилизации бумаги, оборудования и носителей информации. Следует тщательно выбирать подходящего подрядчика с учетом имеющегося у него опыта и обеспечения необходимого уровня информационной безопасности;

k) по возможности следует регистрировать утилизацию важных объектов с целью последующего аудита.



## **9. Заключение**

В процессе выполнения дипломного проекта получены следующие результаты:

- проведен анализ бизнес-процесса разработки ПО департамента;
- проведено исследование существующих аналогичных систем;
- спроектирована и реализована компонентная структура системы;
- разработаны алгоритмов, реализующие поддержку процессов разработки ПО департамента;
- проведено исследование временных затрат на основные функции администрирования старой и новой систем;
- поставленные задачи выполнены а цели достигнуты - система прошла тестирование и введена в эксплуатацию.

## **10. Список используемой литературы**

- 10.1.** Официальная документация для разработчиков от Microsoft - Microsoft Development Network ([msdn.microsoft.com](http://msdn.microsoft.com))
- 10.2.** Фаулер М. Основы UML, 3-е издание. : СПб: Символ-Плюс, 2002 г. 185 с.
- 10.3.** Ларман, Крэг. Применение UML и шаблонов проектирования. 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 624 с.
- 10.4.** Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. ГОСТ 19.201-78.
- 10.5.** Единая система программной документации. Схемы алгоритмов, программ, данных и систем. ГОСТ 19.701-90. Условные обозначения и правила выполнения.
- 10.6.** Арсеньев В.В., Сажин Ю.Б. Методические указания к выполнению организационно-экономической части дипломных проектов по созданию программной продукции. М.: Изд-во МГТУ им. Баумана, 1994. - 52 с.
- 10.7.** Сажин Ю.Б., Самохин С.В. Выполнение организационно-экономической части дипломного проекта по разработке и использованию программного продукта: Методическое пособие. - М.: Изд-во МГТУ им. Баумана, 2006. - 60 с.: ил.
- 10.8.** Система стандартов безопасности труда. Термины и определения. ГОСТ 12.0.002-80
- 10.9.** Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. СанПиН 2.2.2/2.4.1340-03
- 10.10.** Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений. СанПиН 2.2.4.1294-03
- 10.11.** Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест. ГН 2.1.6.695-98
- 10.12.** Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. СН 2.2.4/2.1.8.562-96
- 10.13.** Производственная вибрация, вибрация в помещениях жилых и общественных зданий. СН 2.2.4/2.1.8.566-96

## 11. Приложения

### 11.1. Регламент работы с требованиями до внедрения ALM

#### Системы

Описание процессов постановки задач, уточнения задач, оценки задач, выполнения задач и предоставления результатов работы отделом документирования.

#### Решаемые проблемы:

В данный момент задачи отделу разработки часто ставятся устно, при этом, спустя некоторое время, постановка в трактовке автора требований и исполнителей может значительно различаться.

Задачи ставятся не всегда сразу после их возникновения, что приводит к необоснованным сдвигам сроков их завершения, либо неравномерной загрузке сотрудников исполнителей.

#### Цель:

Упорядочить процессы в отделе документации, определить ответственных за каждый процесс, требования к информации и механизмы передачи этой информации в рамках процессов.

#### Границы:

Данный документ не затрагивает процессы до выделения задачи на разработку или доработку документации (в дальнейшем Задачи) и после завершения Задачи и передачи ее в тестирование. Также, в данном документе не рассматриваются процессы внутри отдела документирования составляющие собственно сам процесс разработки документации.

#### Инструменты:

В процессах предполагается использовать Bugzilla (<http://bugzilla.elvees.com/bug/>)<sup>4</sup> и корпоративный портал, раздел Документация (<http://centaur.neotek/rnd/userdocs/default.aspx>)<sup>5</sup>. Выбор этих инструментов не окончательный, но в текущий момент представляется оптимальным из доступных.

Процессы.

Постановка.

**Кто?** Поставить Задачу может любой сотрудник, зарегистрированный в Bugzilla (в дальнейшем - Автор). В случае доработки документации в связи с изменением функционала продукта, Автором должен быть руководитель проекта (**PM**) в рамках проекта которого сделаны эти изменения, в случае изменения продукта, не затребованного в рамках какого-либо проекта – **разработчик**, внесший эти изменения, в случае обнаружения ошибки (несоответствия) в документации – **сотрудник, обнаруживший ошибку**.

**Когда?** Задача должна быть поставлена, как только она сформулирована. Ответственность за затягивание постановки Задачи, отсутствие постановки Задачи или невыполнение требований данного документа по оформлению Задачи лежит на Авторе.

**Как?** Задача ставится путем выписывания (**Bug / Ошибка**) в Bugzilla (в дальнейшем Тикет) на **имя руководителя отдела документирования** (в дальнейшем - Исполнитель). В

---

<sup>4</sup> В дальнейшем при переходе на другую систему контроля задач, этот инструмент может быть изменен, но функции, которые с его помощью выполняются останутся примерно теми же. Bugzilla можно ниже по тексту рассматривать как имя нарицательное.

<sup>5</sup> В дальнейшем этот инструмент будет заменен на другой, обеспечивающий контроль версий документов и надежное сохранение информации (backup).

Тикете должны быть заполнены следующие поля:

**Аннотация / Summary** – краткое описание Задачи. Основные требования – по возможности, уникальность, возможность максимально четко идентифицировать Задачу по заголовку, по возможности – суть Задачи.

**Продукт / Product** – платформа-продукт-решение для которого ставится Задача. Если необходимое значение отсутствует в Bugzilla, можно временно выписать Тикет на другой продукт. Ответственность за скорейшее расширение списка значений в Bugzilla и коррекции Тикета лежит на Авторе.

**Компонент / Component** – должно быть корректно заполнено. Если необходимое значение отсутствует в Bugzilla, можно временно выписать Тикет на другой компонент. Ответственность за скорейшее расширение списка значений в Bugzilla и коррекции Тикета лежит на Авторе.

**Важность / Importance** – может быть установлена Автором в соответствии с его мнением о важности Задачи.

**Приоритет / Priority** – должен быть установлен в значение «P5», если Автор не входит в список сотрудников, отвечающих за приоритезацию Задач (см. ниже). Если Автор не входит в упомянутый список, но считает, что приоритет задачи требуется повысить, он должен явно указать это в отдельном комментарии, включив в Подписку/СС List Тикета (в дальнейшем просто СС) [neo-pmo@elvees.com](mailto:neo-pmo@elvees.com).

**Подписчики / CC** – в это поле следует вносить всех, кто упоминается в Тикете, а также тех, кто может предоставить дополнительную информацию по теме Задачи и тех, кто может быть заинтересован ходом работ по Задаче. Кроме того, в отдельных случаях в это поле следует добавить сотрудников, занимающих определенные должности, о чем в документе есть отдельные упоминания.

**Описание / Description** – должно быть максимально подробно и включать в себя следующую информацию.

*Для Тикетов на ошибки*

Описание ошибки – должно отвечать на вопрос – «Что в документации не так?»

Описание – как воспроизвести ситуацию в продукте, если документация не соответствует его поведению

Ссылки на источники информации (внутренние описания, материалы в Bugzilla, письма), которые описывают поведение продукта в интересующей ситуации

Предложения по исправлению

*Для Тикетов на разработку-доработку*

Какой функционал (какие изменения функционала) должен быть описан

В какую часть документации требуется внести изменения

Ссылки на источники информации (внутренние описания, материалы в Bugzilla, письма), которые описывают поведение продукта в интересующей ситуации

Предложения по доработке

Срок выполнения Задачи (если этот срок определяется внешними обязательствами)

Определение приоритета.

**Кто?** Изменить приоритет Задачи могут следующие сотрудники.

руководитель департамента по разработке ПО

зам. руководителя департамента по разработке ПО

РМ

Другие сотрудники для изменения приоритета должны сделать запрос на изменение в отдельном комментарии, включив в СС [neo-pmo@elvees.com](mailto:neo-pmo@elvees.com).

**Когда?** Приоритет Задачи меняется при необходимости.

**Как?** Сотрудник, отвечающий за определение приоритетов Задач изменяет значения полей **Приоритет / Priority**, сопровождая свои действия комментарием, поясняющем мотивы этих изменений.

Оценка и планирование.

**Кто?** Оценка трудоемкости Задачи и определение срока завершения разработки выполняется Исполнителем.

**Когда?** Исполнитель должен в течение восьми часов рабочего времени оценить трудоемкость и срок исполнения Задачи. Если срок был задан при постановке Задачи, и он не был подтвержден Исполнителем, Автор должен согласовать коррекцию срока или коррекцию постановки Задачи с Исполнителем. При невозможности согласования проблема должна быть срочно эскалирована руководителю департамента по разработке ПО.

При невозможности в течение восьми часов рабочего времени дать оценку, Исполнитель в течение восьми часов рабочего времени в отдельном комментарии должен указать срок подготовки оценок.

**Как?** Исполнитель фиксирует оценки срока и трудоемкости задачи в отдельном комментарии и устанавливает **Состояние / Status** Тикета «Назначено / Assigned». Если в дальнейшем, оценки по Тикетам в состоянии «назначено» меняются, Исполнитель фиксирует эти изменения в отдельном комментарии. Если оценки выходят за рамки срока, согласованного ранее с Автором, изменение срока должно быть также с ним предварительно согласовано, при невозможности это сделать проблема должна быть срочно эскалирована руководителю департамента по разработке ПО.  
Уточнение.

**Кто?** Автор Задачи.

**Когда?** Как можно быстрее при получении запроса на уточнение от Исполнителя.

**Как?** Исполнитель, может запросить у Автора пояснения или дополнительные данные по Задаче. Для этого Исполнитель формулирует свой запрос в отдельном комментарии Тикета и устанавливает **Состояние / Status** Тикета «Решена / Resolved» и **Resolution / Решение** – «Workforme / Невоспроизводимая». Автор предоставляет запрошенную информацию и устанавливает **Состояние / Status** Тикета «Повторная / Reopened».  
Выполнение и Предоставление результата.

**Кто?** Исполнитель.

**Когда?** В соответствии с планами работ Исполнителя.

**Как?** После завершения работы Исполнитель устанавливает **Состояние / Status** Тикета «Решена / Resolved» и **Resolution / Решение** – «Fixed / Исправлена». Результат работы должен быть опубликован на корпоративном портале в разделе Документация, ссылки на результаты должны быть приведены в отдельном комментарии Тикета.  
Отмена или приостановка.

**Кто?** Автор или руководители Автора любого уровня, или руководитель департамента по разработке ПО.

**Когда?** Как можно раньше, в случае принятия решения о приостановке или отмене Задачи.

**Как?** Сотрудник, принявший решение о приостановке или отмене Задачи устанавливает **Состояние / Status** Тикета «Решена / Resolved» и **Resolution / Решение** – «Wontfix / Отказано» или «Invalid / Аннулирована».

## 11.2. Регламент работы с сущностью “Bug”

Life cycle бага в ТФС

Автор: Андрей Глазков

Модератор: Андрей Глазков

## Описание жизненного цикла бага

Для ясности разобьем всех пользователей на группы:

Тестировщики

Разработчики

ПМы

Суппорт

Прочие

Как трактовать и заполнять некоторые поля и связи бага

Поле **Assigned To** для багов в состояниях кроме Resolved и Closed указывает на сотрудника от которого ожидается очередное действие по багу. Пока баг не взят в итерацию, он должен быть назначен на ПМ-а. Когда команда берет баг в работу, он должен быть переназначен на одного из членов команды. В состоянии Resolved это поле указывает на сотрудника, который принял решение, что все необходимые для исправления ошибки изменения внесены. При переводе в состояние Closed это поле не меняется. Не следует переводить баги на тестеров для выполнения их проверки, но можно породить от бага task на проведение внутреннего тестирования.

Флаг **Triaged** указывает, был ли проведен разбор бага на триаж-митинге.

Связь типа **Related** должна быть установлена для всех майлстоунов, выполнение которых невозможно без исправления данной ошибки (это не единственное назначение связей типа Related).

Связь типа **Parent** должна указывать на майлстоун в рамках которого планируется исправить ошибку или на US более низкого уровня, для выполнения которой требуется исправить данный баг. Начальную установку этой связи должен выполнить автор, уточнение вносится по согласованию между ПМ-ами на триаж-митинге. Если планы изменяются, и ПМ перепланирует исправление бага в рамках иного майлстоуна, причем, у бага есть связи Related с майлстоунами, за которые отвечают другие ПМ-ы, то у бага должен быть сброшен флаг Triaged. Если нет реального майлстоуна, в рамках которого планируется исправление ошибки, то следует завести фиктивный, багов без связи Parent быть не должно.

Создание бага

Постить баги в состоянии New могут все, при этом поле Triaged по умолчанию имеет значение “No”.

При создании бага необходимо позаботиться, чтобы у него было корректно заполнено поле Assigned To и выставлена правильная связь типа Parent. Рекомендуется порождать баг от майлстоуна, в рамках которого выполняется работа по тестированию (чтобы облегчить тестерам выполнение этого требования ПМ-у следует задачи на тестирование порождать напрямую от майлстоуна, без промежуточных US). При этом баг автоматически назначится на ПМ-а, который отвечает за работы по выбранному майлстоуну.

Если автор тикета затрудняется с выбором майлстоуна, от которого следует породить баг (например, если баг найден сотрудником службы поддержки или разработчиком), следует выбрать майлстоун, максимально подходящий с точки зрения автора бага и назначенный на ПМ-а, который в данное время продолжает работать в компании (Дальнейшие переназначения будут проведены ПМ-ами на Тriage-митинге). В любом случае, новый баг должен быть назначен (Assigned) на одного из работающих ПМ-ов и порожден (иметь связь типа Parent) от одного из активных (не Closed) майлстоунов.

Triage-meeting

На триаже, если баг понятен, значение поля Triaged меняется на “Yes”. Если баг требует подтверждения тестером, то его стэйт переводится в состояние Unconfirmed, Iteration Path меняется на rnd/QA, в History фиксируется, что именно нужно подтвердить или прояснить, проставляется Stack Rank.

На триаж-митинге принимается решение, в рамках работ по какому майлстоуну будет

исправляться ошибка, при необходимости корректируется связь Parent и назначение бага (баг, который не взят в итерацию, должен быть назначен на ПМ-а, ответственного за майлстоун, в рамках которого планируется исправление бага). Кроме того определяется, какие еще майлстоуны затрагивает данный баг (с добавлением соответствующих связей Related). Связь Parent корректируется ведущим триаж-митинга, за добавление необходимых связей Related отвечают ПМ-ы, ответственные за работы по соответствующим майлстоунам.

Кроме того, ПМ-ы должны отслеживать, в какие ветки необходимо перенести исправления и добавлять в баг дочерние Task-и на перенос изменений (merge).

Работа над багом

Переводить из состояния Unconfirmed в состояние New могут только тестировщики. Из состояния Unconfirmed и New баг можно перевести в состояние Closed. У состояния Closed могут быть reason'ы: Rejected, Duplicate, Work as designed, Obsolete, Verified.

Когда баг берется в разработку, его состояние меняется на Active.

Из Active баг можно перевести в Closed(Rejected, Duplicate, Work as designed, Obsolete).

Из Active баг можно перевести в New.

Из состояния Resolved баг можно перевести либо в состояние New(Reopened), либо в состояние Closed(Verified).

Любая задача, которая переводится в состояние Closed(любой Reason), должна переводиться на автора бага(нужно заказать автоматизацию у Дениса).

Если при проверке бага тестер обнаруживает, что описание проблемы существенно изменилось по сравнению с первоначальным или изменилось значение Area (именно изменилось, а не уточнилось), то поле Triaged меняется на "No", чтобы баг был повторно рассмотрен ПМ-ами на Triage-митинге.

Все возможные пары State-Reason

State	Reason
New	New defect reported
	Reopened
Active	Work started
Resolved	Bug was fixed
Closed	Rejected
	Duplicate
	Obsolete
	Work as Designed
	Verified
Unconfirmed	Bug is unconfirmed

Общая таблица переходов и значения по умолчанию

From State:	To State:	Possible reasons:	Default reason:
---	New	New defect reported	New defect reported
New	Active	Work started	Work started
New	Closed	Rejected; Duplicate; Work as designed; Obsolete.	Rejected
New	Unconfirmed	Bug is unconfirmed	Bug is unconfirmed
Active	New	Reopened	Reopened
Active	Resolved	Bug was fixed	Bug was fixed
Active	Closed	Rejected; Duplicate; Work as designed; Obsolete.	Rejected
Resolved	New	Reopened	Reopened
Resolved	Closed	Verified; Obsolete.	Verified
Resolved	Active	Work started	Work started
Closed	New	Reopened	Reopened
Unconfirmed	New	New defect reported	New defect reported
Unconfirmed	Closed	Rejected; Duplicate; Work as designed; Obsolete.	Rejected



Таблица разрешенных действий по ролям состояний:

Curr. State	New State	Reason	Кто меняет <sup>6</sup>	Инфо
-	New	New defect reported	Все	Создание бага. Флаг Triaged сброшен
New	Closed	Rejected	Triage team, PM, QA	Добавляется комментарий о причине отклонения бага
	Closed	Duplicate	Triage team, PM, QA	В поле duplicate bug добавляется ID дубликата
	Closed	Work as designed	Triage team, PM, QA	Указывается пункт в требованиях, либо цитируется конкретное требование.
	Closed	Obsolete	Triage team, PM, QA	Указывается пояснение, почему данный баг считается устаревшим (ссылка на новые требования)
	New(Triaged)	New defect reported	Triage team	По багу все понятно – подтверждается на triage. Выставляется флаг Triaged
	Unconfirmed	Bug is unconfirmed	Triage team	Когда на triage решили, что баг требует подтверждения Поле Area переводится на rnd/QA (автоматически)
New(Triaged)	Closed	Duplicate	developer	Баг дублирующий. В поле duplicate bug добавляется ID дубликата
	Closed	Work as designed	PM	Указывается пункт в требованиях, либо цитируется конкретное требование.
	Closed	Obsolete	PM	Указывается пояснение, почему данный баг считается устаревшим (ссылка на новые требования)
	Active	Work started	developer	Баг берется в работу разработчиком
Unconfirmed	Closed	Rejected	QA	Добавляется комментарий о причине отклонения бага
	Closed	Duplicate	QA	В поле duplicate bug ID добавляется ID дубликата
	Closed	Work as designed	PM, QA	Указывается пункт в требованиях, либо цитируется конкретное требование.

<sup>6</sup> Пока на это поле не обращаем внимание, т.к. на первом этапе решили, что «все могут делать всё». Позже сформируем политики доступа и вернемся к этому вопросу.

	Closed	Obsolete	PM, QA	Указывается пояснение, почему данный баг считается устаревшим (ссылка на новые требования)
	New	New defect reported	QA	Баг подтвержден QA – возвращается на triage Поле Area меняется на оригинальное
Active	Closed	Rejected	PM	На этапе работы выяснилось, что баг не актуален
	Closed	Work as designed	PM	Указывается пункт в требованиях, либо цитируется конкретное требование.
	Closed	Obsolete	PM	Указывается пояснение, почему данный баг считается устаревшим (ссылка на новые требования)
	Closed	Duplicate	PM	На этапе работы выяснилось, что баг дублирующий - в поле duplicate bug добавляется ID дубликата
	New	Reopened	PM	Работа над багом отложена
	Resolved	Bug was fixed	developer	Работа над фиксом бага закончена. Поле Iteration Path переводится на rnd/QA (автоматически)
Resolved	Closed	Verified	QA	Баг исправлен
	New	Reopened	QA	Баг не исправлен
	Active	Work started	develop	Баг не исправлен
Closed	New	Reopened	Все	Баг переоткрыт. Флаг Triaged должен быть сброшен (автоматически)

### 11.3. Листинг программ подсистемы резервного копирования

#### 11.3.1. Модуль Backup.ps1

```
# @copyright
# 2012 ELVEES NeoTek CJSC. All rights reserved.

#Script calls other backup scripts and redirects its output to the log file.
Sending log by email.

# Parameters:
# - Collection of VMs. Example: -VMs "VM_win7x64,VM_win7x32,VM_win7x64_test"

param(
    [array] $VMs
)

# Number of strings in log file
[int] $StringsNumber = 400
# Email address for notification
[string]$Email = "neo-scm@elvees.com"
# Source folder of Databases
```

```

[string]$SourceDB = "\\TFS-apptier\BackupDB"
# Path to the working dir on localhost
[string]$workDir = "D:\TfsBackupSystem"
# Source folder for VMS
[string]$TragetVMSRootDir = "C:\Hyperv\Virtual Machines\TFS"
# Path to the remote storage
[string]$Destination = "\\SORC-PC\TFSBackup"

function getCreds
{
    param(
        [string]$keypath
    )
    $key = (Get-ItemProperty $keypath).key
    [byte[]]$bytekey = $key -split ","
    [string]$cod = Get-Content -Path "$workDir\cr"
    $secureStrPas = Convertto-Securestring $cod -Key $bytekey
    return $secureStrPas
}

function sendNotification
{
    param(
        [string]$To,
        [string]$From,
        [string]$Body,
        [string]$Subject,
        [System.IO.FileInfo[]]$Attachment
    )

    # Create SMTP Client
    $smtpClient = New-Object System.Net.Mail.SmtpClient
    $smtpClient.EnableSsl = $SSL
    $smtpClient.Host = "mail.elvees.com"
    $SMTPClient.Credentials = New-Object
System.Net.NetworkCredential("tfs_admin", (getCreds "hk1m:\software\TFSS"))

    # Create Message
    $Message = New-Object System.Net.Mail.MailMessage
    $Message.Body = $Body
    $Message.Subject = $Subject
    $Message.From = $From
    $Message.To.Add($To)

    foreach ($file in $Attachment)
    {
        $a = New-Object System.Net.Mail.Attachment($file)
        $Message.Attachments.Add($a)
    }
    $smtpclient.Send($Message)
    $Message.Dispose()
}

#Main caller of other scripts
function Bootstrap
{
    #Function Parameters
    # - path to the log
    # - path to the script
    param(
        [string]$LogName,
        [string]$Script
    )

    Try
    {
        # Check the existance of log file
        if ( Test-Path "$LogName.txt")
        {
            # Check the size of existing log file
            if ( ([IO.File]::ReadAllLines("$LogName.txt")).length -le
$StringsNumber)
            {
                # Call the script
                Invoke-Expression $Script | Out-File -FilePath "$LogName.txt" -
Append -noClobber -Encoding unicode
            }
            else
            {
                # Recieving all archived log files
                [array]$files = Get-ChildItem "$LogName(*)\.txt"|Sort -
Descending
            }
        }
    }
}

```

```

        [int]$count = $files.length
        foreach ($p in $files)
        {
            # Rename all archived log files
            $count_plus=$count+1
            Rename-Item -NewName "$LogName($count_plus).txt" -Path
$P.FullName
            $count--
        }
        # Remove the oldest file (fivth copy)
        if ( Test-Path "$LogName(5).txt")
        {
            Remove-Item "$LogName(5).txt"
        }
        # Archive log file
        Rename-Item -NewName "$LogName(1).txt" -Path "$LogName.txt"
        # Call the script to the new log
        Invoke-Expression $Script | Out-File -FilePath "$LogName.txt"
    }
    -Encoding unicode
    }
    else
    {
        # Call the script to the new log
        Invoke-Expression $Script | Out-File -FilePath "$LogName.txt" -
Encoding unicode
    }
    }
    Catch
    {
        # Error with logging or execution of script
        $Error[0] | Out-File -FilePath "ExecutionLog.txt" -Append -noClobber -
Encoding unicode
        sendNotification -To $Email -From "$env:COMPUTERNAME@elvees.com" -Subject
"TFS Backup common error" -Attachment (Get-ChildItem
"$WorkDir\ExecutionLog.txt")
    }
}

# Backup the VMs
If ($VMs)
{
    # Creating full expression( with arguments for script)
    $Expression = "$WorkDir\VMBackup.ps1 -Destination '$Destination' -VMStorage
'$WorkDir\System' -Source '$SourceVM' -VMS '$VMS'"

    # Execute script
    Bootstrap -LogName "$WorkDir\VM_backup_log" -Script $Expression

    if($LASTEXITCODE)
    {
        $EmailSubject = "TFS System(s) backup has failed"
    }
    else
    {
        $EmailSubject = "TFS System(s) backup has finished"
    }

    sendNotification -To $Email -From "$env:COMPUTERNAME@elvees.com" -Subject
$EmailSubject -Attachment (Get-ChildItem "$WorkDir\VM_backup_log*.txt")
}

# Backup the DB
If ($SourceDB)
{
    # Creating full expression( with arguments for script)
    $Expression = "$WorkDir\DBBackupTransfer.ps1 -Destination '$Destination' -
DBStorage '$WorkDir\DB' -Source '$SourceDB'"

    # Execute script
    Bootstrap -LogName "$WorkDir\DB_backup_log" -Script $Expression

    if($LASTEXITCODE)
    {
        $EmailSubject = "TFS DB Transfer has failed"
    }
    else
    {
        $EmailSubject = "TFS DB Transfer has finished"
    }
}

```

```

        sendNotification -To $Email -From "$env:COMPUTERNAME@elvees.com" -Subject
$EmailSubject -Attachment (Get-ChildItem "$workDir\DB_backup_log*.txt")
    }
}

```

### 11.3.2. Модуль VMBackup.ps1

```

# @copyright
# 2012 ELVEES NeoTek CJSC. All rights reserved.

# Script takes backups of VMs(xml and images) from local Hyper-v server,
# places it to the date-named archive and upload to the remote shared folder.

# Parameters:
#- path to the working directory
#- Source folder
#- Array of VMs
#- Destination

param(
    [string] $VMStorage,
    [string] $Source,
    [string] $VMs,
    [string] $Destination
)

function DateToStr($date)
{
    #Function Parameters
    # $date - date that needs to convert

    [string]$dd= $date.Day
    [string]$mm="_"+ $date.Month
    [string]$yyyy="_"+ $date.Year
    return "$dd$mm$yyyy"
}

function backupVM($VM)
{
    #Function Parameters
    # $VM - Name of VM for backup

    Write-Output "$(date) - Stopping Virtual Machine $VM"

    Stop-VM -Name $VM -AsJob -Force
    Start-Sleep -Seconds 15

    Write-Output "$(date) - Start backup $VM"
    $Result = Copy-Item "$Source\$VM" $VMStorage -Recurse -PassThru

    Write-Output "$(date) - The following files and folders have been copied from
$Source\$VM"
    Write-Output $Result

    Write-Output "$(date) - Launching $VM"
    Start-VM $VM
}

Try
{
    # Initialize
    # Create date string in format dd_mm_yyyy
    $Date = DateToStr(get-date)
    # Create path string to local archive with VMs
    $LocalArchivePath = "$(Split-Path $VMStorage)\System_$Date.7z"

    #Split to the VM list
    $VMList = $VMs.split(",")

    Write-output "$(date) - Prepare working directory"

    if ( Test-Path $VMStorage)
    {
        Remove-Item $VMStorage -Recurse
    }
    new-item $VMStorage -itemtype directory

    #Backup VMs files
    foreach ($V in $VMList)
    {
        backupVM $V
    }

    #Create date-named archive with minimal compression
}

```

```

$AllArgs = @('a', '-t7z', '-ssw', '-mx0', "$LocalArchivePath", $VMStorage)
& 'C:\Program Files\7-Zip\7z.exe' $AllArgs

#If not alright for archiving - exit
if($LASTEXITCODE)
{
    throw
}

write-output "$(date) - Start copying archive to $Destination"

$Result = Copy-Item "$LocalArchivePath" $Destination -PassThru
write-output $Result
}
Catch
{
    write-error "$(date) - VM Backup has failed"
    $Error[0]

    if ( Test-Path $LocalArchivePath)
    {
        Remove-Item $LocalArchivePath -ErrorAction Continue -ErrorVariable E
        $E
    }

    exit 1
}

# Clean local archive
Try
{
    remove-item "$LocalArchivePath"
}
Catch
{
    write-error "$(date) - VM archive cleaning failed. You can try to clean
archive manually."
    $Error[0]

    exit 2
}

write-output "$(date) - VM(s) backup has finished successfully"
exit 0

```

### 11.3.3. Модуль DBBackupTransfer.ps1

```

# @copyright
# 2012 ELVEES NeoTek CJSC. All rights reserved.

# Script takes backups from SQL instance, places it to the date-named arhive and
upload to the remote shared folder.
# Clean previous archive.

# Parameters:
#- path to the working directory
#- Source folder
#- Destination

param(
    [string]$DBStorage,
    [string]$Source,
    [string]$Destination
)

function DateToStr($date)
{
    #Function Parameters
    # $date - date that needs to convert

    [string]$dd= $date.Day
    [string]$mm="-"+ $date.Month
    [string]$yyyy="-"+ $date.Year
    return "$dd$mm$yyyy"
}

Try
{
    # Initialize
    # Create date string in format dd_mm_yyyy
    $Date = DateToStr(get-date)
    # Create path string to local archive with VMs
    $LocalArchivePath = "$(Split-Path $DBStorage)\$Date.7z"

```

```

write-output "$(date) - Prepare working directory"

if ( Test-Path $DBStorage)
{
    Remove-Item $DBStorage -Recurse
}
new-item $DBStorage -itemtype directory

write-output "$(date) - Start copying databases from $Source"

$Result = Copy-Item "$Source\*" $DBStorage -Recurse -PassThru

write-output "$(date) - The following files and folders have been copied from
$Source"
write-output $Result

#Create date-named archive with large compression
$AllArgs = @('a', '-t7z', '-ssw', '-mx7', "$LocalArchivePath", $DBStorage)
& 'C:\Program Files\7-Zip\7z.exe' $AllArgs

#If not alright for archiving - exit
if($LASTEXITCODE)
{
    throw
}

write-output "$(date) - Start copying archive to $Destination"

$Result = Copy-Item "$LocalArchivePath" $Destination -PassThru
write-output $Result
}
Catch
{
    write-error "$(date) - Database Transfer has failed"
    $Error[0]

    if ( Test-Path $LocalArchivePath)
    {
        Remove-Item $LocalArchivePath -ErrorAction Continue -ErrorVariable E
        $E
    }

    exit 1
}

# Clean old and local archives
Try
{
    if ( Test-Path "$Destination\$Prevdate.7z")
    {
        Remove-Item "$Destination\$Prevdate.7z"
    }

    remove-item "$LocalArchivePath"
}
Catch
{
    write-error "$(date) - Database archives cleaning failed. You can try to
clean archives manually."
    $Error[0]

    exit 2
}

write-output "$(date) - Database Transfer has finished successfully"
exit 0

```

#### 11.4. Описание рабочего элемента «Bug»

```

<?xml version="1.0" encoding="utf-8"?>
<Witd:WITD application="Work item type editor" version="1.0"
xmlns:Witd="http://schemas.microsoft.com/VisualStudio/2008/workitemtracking/typedef">
  <WORKITEMTYPE name="Bug">
    <DESCRIPTION>Describes a divergence between required and actual behavior, and tracks the work done to
correct the defect and verify the correction.</DESCRIPTION>
    <FIELDS>
      <FIELD name="Iteration Path" refname="System.IterationPath" type="TreePath" reportable="dimension">
        <HELPTEXT>The iteration within which this bug will be fixed</HELPTEXT>
      </FIELD>
    </FIELDS>
  </WORKITEMTYPE>
</WITD>

```

```

<FIELD name="Iteration ID" refname="System.IterationId" type="Integer" />
<FIELD name="External Link Count" refname="System.ExternalLinkCount" type="Integer" />
<FIELD name="Team Project" refname="System.TeamProject" type="String" reportable="dimension" />
<FIELD name="Hyperlink Count" refname="System.HyperLinkCount" type="Integer" />
<FIELD name="Attached File Count" refname="System.AttachedFileCount" type="Integer" />
<FIELD name="Node Name" refname="System.NodeName" type="String" />
<FIELD name="Area Path" refname="System.AreaPath" type="TreePath" reportable="dimension">
  <HELPTTEXT>The area of the product with which this bug is associated</HELPTTEXT>
</FIELD>
<FIELD name="Revised Date" refname="System.RevisedDate" type="DateTime" reportable="detail" />
<FIELD name="Changed Date" refname="System.ChangedDate" type="DateTime" reportable="dimension" />
<FIELD name="ID" refname="System.Id" type="Integer" reportable="dimension" />
<FIELD name="Area ID" refname="System.AreaId" type="Integer" />
<FIELD name="Authorized As" refname="System.AuthorizedAs" type="String" syncnamechanges="true" />
<FIELD name="Title" refname="System.Title" type="String" reportable="dimension">
  <REQUIRED />
  <HELPTTEXT>Stories affected and how</HELPTTEXT>
</FIELD>
<FIELD name="State" refname="System.State" type="String" reportable="dimension">
  <HELPTTEXT>New = Bug not approved; Active = not yet fixed; Resolved = fix not yet verified; Closed = fix
verified.</HELPTTEXT>
</FIELD>
<FIELD name="Authorized Date" refname="System.AuthorizedDate" type="DateTime" />
<FIELD name="Watermark" refname="System.Watermark" type="Integer" />
<FIELD name="Rev" refname="System.Rev" type="Integer" reportable="dimension" />
<FIELD name="Changed By" refname="System.ChangedBy" type="String" syncnamechanges="true"
reportable="dimension">
  <ALLOWEXISTINGVALUE />
  <VALIDUSER />
</FIELD>
<FIELD name="Reason" refname="System.Reason" type="String" reportable="dimension">
  <HELPTTEXT>The reason why the bug is in the current state</HELPTTEXT>
</FIELD>
<FIELD name="Assigned To" refname="System.AssignedTo" type="String" syncnamechanges="true"
reportable="dimension">
  <ALLOWEXISTINGVALUE />
  <VALIDUSER />
  <HELPTTEXT>The person currently working on this bug</HELPTTEXT>
</FIELD>
<FIELD name="Work Item Type" refname="System.WorkItemType" type="String" reportable="dimension" />
<FIELD name="Created Date" refname="System.CreatedDate" type="DateTime" reportable="dimension" />
<FIELD name="Created By" refname="System.CreatedBy" type="String" syncnamechanges="true"
reportable="dimension" />
<FIELD name="Description" refname="System.Description" type="HTML" />
<FIELD name="History" refname="System.History" type="History">
  <HELPTTEXT>Discussion thread plus automatic record of changes</HELPTTEXT>
</FIELD>
<FIELD name="Related Link Count" refname="System.RelatedLinkCount" type="Integer"
reportable="dimension" />
<FIELD name="Tags" refname="System.Tags" type="PlainText" />
<FIELD name="System Info" refname="Microsoft.VSTS.TCM.SystemInfo" type="HTML">
  <HELPTTEXT>Test context, provided automatically by test infrastructure</HELPTTEXT>
</FIELD>
<FIELD name="Repro Steps" refname="Microsoft.VSTS.TCM.ReproSteps" type="HTML">
  <REQUIRED />
  <DEFAULT from="value" value="&lt;b&gt;&lt;i&gt;Шаги к
воспроизведению:&lt;/i&gt;&lt;b&gt;&lt;/b&gt;&lt;br&gt;&lt;br&gt;&lt;b&gt;&lt;i&gt;Результат:&lt;/i&gt;&lt;b&gt;&lt;/b&gt;&lt;br&gt;&lt;br&gt;&lt;b&gt;&lt;i&gt;Ожидаемый
результат:&lt;/i&gt;&lt;b&gt;&lt;/b&gt;&lt;br&gt;&lt;br&gt;" />
  <HELPTTEXT>How to see the bug. End by contrasting expected with actual
behavior.</HELPTTEXT>
</FIELD>
<FIELD name="State Change Date" refname="Microsoft.VSTS.Common.StateChangeDate" type="DateTime">
  <WHENCHANGED field="System.State">
    <SERVERDEFAULT from="clock" />

```



```

    </WHENCHANGED>
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Activated Date" refname="Microsoft.VSTS.Common.ActivatedDate" type="DateTime"
reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Activated By" refname="Microsoft.VSTS.Common.ActivatedBy" type="String"
syncnamechanges="true" reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <ALLOWEXISTINGVALUE />
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Resolved Date" refname="Microsoft.VSTS.Common.ResolvedDate" type="DateTime"
reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Resolved By" refname="Microsoft.VSTS.Common.ResolvedBy" type="String"
syncnamechanges="true" reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <ALLOWEXISTINGVALUE />
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Resolved Reason" refname="Microsoft.VSTS.Common.ResolvedReason" type="String"
reportable="dimension">
    <ALLOWEDVALUES expanditems="true">
        <LISTITEM value="Fixed" />
    </ALLOWEDVALUES>
    <FROZEN />
    <HELPTTEXT>The reason why the bug was resolved</HELPTTEXT>
</FIELD>
<FIELD name="Closed Date" refname="Microsoft.VSTS.Common.ClosedDate" type="DateTime"
reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Closed By" refname="Microsoft.VSTS.Common.ClosedBy" type="String"
syncnamechanges="true" reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <ALLOWEXISTINGVALUE />
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Priority" refname="Microsoft.VSTS.Common.Priority" type="Integer"
reportable="dimension">
    <ALLOWEDVALUES expanditems="true">
        <LISTITEM value="1" />
        <LISTITEM value="2" />
        <LISTITEM value="3" />
        <LISTITEM value="4" />
    </ALLOWEDVALUES>
    <DEFAULT from="value" value="2" />
    <HELPTTEXT>Business importance. 1=must fix; 4=unimportant.</HELPTTEXT>
</FIELD>
<FIELD name="Severity" refname="Microsoft.VSTS.Common.Severity" type="String"
reportable="dimension">

```

```

    <ALLOWEDVALUES expanditems="true">
      <LISTITEM value="1 - Critical" />
      <LISTITEM value="2 - High" />
      <LISTITEM value="3 - Medium" />
      <LISTITEM value="4 - Low" />
    </ALLOWEDVALUES>
    <DEFAULT from="value" value="3 - Medium" />
    <HELPTTEXT>Assessment of the effect of the bug on the project</HELPTTEXT>
  </FIELD>
  <FIELD name="Stack Rank" refname="Microsoft.VSTS.Common.StackRank" type="Double"
reportable="dimension">
    <READONLY not="[project]\TFS RND Project Managers" />
    <HELPTTEXT>Work first on items with lower-valued stack rank. Set in triage.</HELPTTEXT>
  </FIELD>
  <FIELD name="Integration Build" refname="Microsoft.VSTS.Build.IntegrationBuild" type="String"
reportable="dimension">
    <SUGGESTEDVALUES expanditems="true">
      <LISTITEM value="&lt;None&gt;" />
    </SUGGESTEDVALUES>
    <SUGGESTEDVALUES expanditems="true" filteritems="excludegroups">
      <GLOBALLIST name="Builds - rnd" />
    </SUGGESTEDVALUES>
    <HELPTTEXT>The build in which the bug was fixed</HELPTTEXT>
  </FIELD>
  <FIELD name="Found In" refname="Microsoft.VSTS.Build.FoundIn" type="String" reportable="dimension">
    <SUGGESTEDVALUES expanditems="true">
      <LISTITEM value="&lt;None&gt;" />
    </SUGGESTEDVALUES>
    <SUGGESTEDVALUES expanditems="true" filteritems="excludegroups">
      <GLOBALLIST name="Builds - rnd" />
    </SUGGESTEDVALUES>
    <HELPTTEXT>The build in which the bug was found</HELPTTEXT>
  </FIELD>
  <FIELD name="Story Points" refname="Microsoft.VSTS.Scheduling.StoryPoints" type="Double"
reportable="measure">
    <HELPTTEXT>The size of work estimated for implementing this user story</HELPTTEXT>
  </FIELD>
  <FIELD name="Due Date" refname="Microsoft.VSTS.Scheduling.DueDate" type="DateTime"
reportable="dimension">
    <HELPTTEXT>The date by which this bug needs to be closed</HELPTTEXT>
  </FIELD>
  <FIELD name="Triaged" refname="Elvees.RND.Triaged" type="String">
    <DEFAULT from="value" value="No" />
    <REQUIRED for="[project]\Triage Committers" />
    <SUGGESTEDVALUES expanditems="true">
      <GLOBALLIST name="Yes-No" />
    </SUGGESTEDVALUES>
    <ALLOWEDVALUES expanditems="true">
      <GLOBALLIST name="Yes-No" />
    </ALLOWEDVALUES>
    <WHEN field="System.State" value="Active">
      <READONLY not="[project]\Triage Committers" />
    </WHEN>
    <WHEN field="System.State" value="Unconfirmed">
      <READONLY not="[project]\Triage Committers" />
    </WHEN>
    <WHEN field="System.State" value="Resolved">
      <READONLY not="[project]\Triage Committers" />
    </WHEN>
  </FIELD>
  <FIELD name="Duplicate Bug" refname="Elvees.RND.DuplicateBug" type="Integer">
    <HELPTTEXT>Define ID of the bug which is duplicated by this bug</HELPTTEXT>
  </FIELD>
  <FIELD name="Product Version" refname="Elvees.RND.ProductVersion" type="String">
    <HELPTTEXT>Define version of build where bug has been discovered</HELPTTEXT>

```

```

    <REQUIRED />
  </FIELD>
  <FIELD name="Duration" refname="Elvees.RND.Scheduling.Duration" type="Double" reportable="measure"
formula="sum">
    <HELPTEXT>The amount of work in human-days estimated by project manager</HELPTEXT>
  </FIELD>
</FIELDS>
<WORKFLOW>
  <STATES>
    <STATE value="New">
      <FIELDS>
        <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Elvees.RND.Triaged">
          <WHENNOTCHANGED field="System.State">
            <READONLY not="[project]\Triage Committers" />
          </WHENNOTCHANGED>
        </FIELD>
      </FIELDS>
    </STATE>
    <STATE value="Active">
      <FIELDS>
        <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
      </FIELDS>
    </STATE>
    <STATE value="Resolved">
      <FIELDS>
        <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />

```

```

    </FIELD>
  </FIELDS>
</STATE>
<STATE value="Closed" >
  <FIELDS>
    <FIELD refname="Elvees.RND.Triaged">
      <WHENNOT field="System.State" value="New">
        <READONLY not="[project]\Triage Committers" />
      </WHENNOT>
    </FIELD>
  </FIELDS>
  </STATE>
<STATE value="Unconfirmed" />
</STATES>
<TRANSITIONS>
  <TRANSITION from="" to="New">
    <REASONS>
      <DEFAULTREASON value="New defect reported" />
    </REASONS>
    <FIELDS>
      <FIELD refname="Elvees.RND.Triaged">
        <READONLY not="[project]\Triage Committers" />
      </FIELD>
    </FIELDS>
  </TRANSITION>
  <TRANSITION from="New" to="Active">
    <REASONS>
      <DEFAULTREASON value="Work started" />
    </REASONS>
    <FIELDS>
      <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
        <ALLOWEXISTINGVALUE />
        <COPY from="currentuser" />
        <VALIDUSER />
        <REQUIRED />
      </FIELD>
      <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
        <SERVERDEFAULT from="clock" />
      </FIELD>
      <FIELD refname="System.AssignedTo">
        <DEFAULT from="currentuser" />
      </FIELD>
    </FIELDS>
  </TRANSITION>
  <TRANSITION from="New" to="Closed">
    <REASONS>
      <REASON value="Duplicate">
        <FIELDS>
          <FIELD refname="Elvees.RND.DuplicateBug">
            <REQUIRED />
          </FIELD>
        </FIELDS>
      </REASON>
      <REASON value="Work as designed" />
      <REASON value="Obsolete" />
      <DEFAULTREASON value="Rejected" />
    </REASONS>
    <FIELDS>
      <FIELD refname="System.AssignedTo">
        <DEFAULT from="currentuser" />
      </FIELD>
      <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
        <SERVERDEFAULT from="clock" />
      </FIELD>
      <FIELD refname="Microsoft.VSTS.Common.ClosedBy">

```

```

    <ALLOWEXISTINGVALUE />
    <COPY from="currentuser" />
    <VALIDUSER />
    <REQUIRED />
  </FIELD>
</FIELDS>
</TRANSITION>
<TRANSITION from="New" to="Unconfirmed">
  <REASONS>
    <DEFAULTREASON value="Bug is unconfirmed" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Active" to="New">
  <REASONS>
    <DEFAULTREASON value="Reopened" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Elvees.RND.Triaged">
      <READONLY not="[project]\Triage Committers" />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Active" to="Resolved">
  <REASONS>
    <DEFAULTREASON value="Bug was fixed" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
      <SERVERDEFAULT from="clock" />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">
      <ALLOWEXISTINGVALUE />
      <COPY from="currentuser" />
      <VALIDUSER />
      <REQUIRED />
    </FIELD>
  </FIELDS>
  <ACTIONS>
    <ACTION value="Microsoft.VSTS.Actions.Checkin" />
  </ACTIONS>
</TRANSITION>
<TRANSITION from="Resolved" to="Closed">
  <REASONS>
    <REASON value="Obsolete" />
    <DEFAULTREASON value="Verified" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">

```

```

    <ALLOWEXISTINGVALUE />
    <READONLY />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
    <SERVERDEFAULT from="clock" />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
    <ALLOWEXISTINGVALUE />
    <COPY from="currentuser" />
    <VALIDUSER />
    <REQUIRED />
  </FIELD>
</FIELDS>
</TRANSITION>
<TRANSITION from="Resolved" to="Active">
  <REASONS>
    <DEFAULTREASON value="Work started" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <COPY from="currentuser" />
      <VALIDUSER />
      <REQUIRED />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <SERVERDEFAULT from="clock" />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Resolved" to="New">
  <REASONS>
    <DEFAULTREASON value="Reopened" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Elves.RND.Triaged">
      <READONLY not="[project]\Triage Committers" />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Active" to="Closed">
  <REASONS>
    <REASON value="Duplicate">
      <FIELDS>
        <FIELD refname="Elves.RND.DuplicateBug">
          <REQUIRED />
        </FIELD>
      </FIELDS>
    </REASON>
    <REASON value="Work as designed" />
    <REASON value="Obsolete" />
    <DEFAULTREASON value="Rejected" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">

```

```

    <ALLOWEXISTINGVALUE />
    <READONLY />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
    <SERVERDEFAULT from="clock" />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
    <ALLOWEXISTINGVALUE />
    <COPY from="currentuser" />
    <VALIDUSER />
    <REQUIRED />
  </FIELD>
</FIELDS>
</TRANSITION>
<TRANSITION from="Closed" to="New">
  <REASONS>
    <DEFAULTREASON value="Reopened" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Elves.RND.Triaged">
      <COPY from="value" value="No" />
      <PROHIBITEDVALUES expanditems="true">
        <LISTITEM value="Yes" />
      </PROHIBITEDVALUES>
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Unconfirmed" to="Closed">
  <REASONS>
    <REASON value="Duplicate">
      <FIELDS>
        <FIELD refname="Elves.RND.DuplicateBug">
          <REQUIRED />
        </FIELD>
      </FIELDS>
    </REASON>
    <REASON value="Work as designed" />
    <REASON value="Obsolete" />
    <DEFAULTREASON value="Rejected" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
      <SERVERDEFAULT from="clock" />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
      <ALLOWEXISTINGVALUE />
      <COPY from="currentuser" />
      <VALIDUSER />
      <REQUIRED />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Unconfirmed" to="New">
  <REASONS>
    <DEFAULTREASON value="New defect reported" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Elves.RND.Triaged">
      <READONLY not="[project]\Triage Committers" />
    </FIELD>
  </FIELDS>
</TRANSITION>
</TRANSITIONS>
</WORKFLOW>
</FORM>

```

```

<Layout HideControlBorders="true" HideReadOnlyEmptyFields="true">
  <Group Margin="(4,0,0,0)">
    <Column PercentWidth="100">
      <Control FieldName="System.Title" Type="FieldControl" LabelPosition="Top" ControlFontSize="large"
EmptyText="&lt;Enter title here&gt;" />
    </Column>
  </Group>
  <Group Margin="(10,0,0,0)">
    <Column PercentWidth="33">
      <Group Label="Status">
        <Column PercentWidth="100">
          <Control FieldName="System.AssignedTo" Type="FieldControl" Label="Assi&igned To"
LabelPosition="Left" EmptyText="&lt;No one&gt;" />
          <Control FieldName="System.State" Type="FieldControl" Label="Stat&e" LabelPosition="Left" />
          <Control FieldName="System.Reason" Type="FieldControl" Label="Reason" LabelPosition="Left" />
          <Control FieldName="Elvees.RND.Triaged" Type="FieldControl" Label="Triaged" LabelPosition="Left"
/>>
        </Column>
      </Group>
    </Column>
    <Column PercentWidth="33">
      <Group Label="Classification">
        <Column PercentWidth="100">
          <Control FieldName="System.AreaPath" Type="WorkItemClassificationControl" Label="&Area"
LabelPosition="Left" />
          <Control FieldName="System.IterationPath" Type="WorkItemClassificationControl"
Label="Ite&ration" LabelPosition="Left" />
          <Control FieldName="Elvees.RND.DuplicateBug" Type="FieldControl" Label="Duplicate"
LabelPosition="Left" />
        </Column>
      </Group>
    </Column>
    <Column PercentWidth="33">
      <Group Label="Planning">
        <Column PercentWidth="100">
          <Control FieldName="Microsoft.VSTS.Common.StackRank" Type="FieldControl" Label="Stack Rank"
LabelPosition="Left" EmptyText="&lt;None&gt;" NumberFormat="DecimalNumbers" MaxLength="10" />
          <Control FieldName="Microsoft.VSTS.Common.Priority" Type="FieldControl" Label="Priority"
LabelPosition="Left" />
          <Control FieldName="Microsoft.VSTS.Common.Severity" Type="FieldControl" Label="Severity"
LabelPosition="Left" />
          <Control FieldName="Microsoft.VSTS.Scheduling.DueDate" Type="DateTimeControl" Label="Due
Date" LabelPosition="Left" EmptyText="&lt;None&gt;" />
          <Control FieldName="Elvees.RND.Scheduling.Duration" Type="FieldControl"
Label="Duration" LabelPosition="Left" />
        </Column>
      </Group>
    </Column>
  </Group>
  <Group>
    <Column PercentWidth="50">
      <TabGroup>
        <Tab Label="Repro Steps">
          <Control FieldName="Microsoft.VSTS.TCM.ReproSteps" Type="HtmlFieldControl"
LabelPosition="Top" Dock="Fill" MinimumSize="(100,200)" />
          <Control FieldName="Elvees.RND.ProductVersion" Type="FieldControl"
Label="Build Version" EmptyText="For example: rel_O2KBase_11_1_ID00099, build 328" LabelPosition="Left"
Dock="Top" />
        </Tab>
        <Tab Label="System Info">
          <Group>
            <Column PercentWidth="50">
              <Control FieldName="Microsoft.VSTS.Build.FoundIn" Type="FieldControl" Label="Found In Build"
LabelPosition="Left" />
            </Column>
          </Group>
        </Tab>
      </TabGroup>
    </Column>
  </Group>

```



```

        <Column PercentWidth="50">
            <Control FieldName="Microsoft.VSTS.Build.IntegrationBuild" Type="FieldControl"
Label="Integrated in Build" LabelPosition="Left" />
        </Column>
    </Group>
    <Control FieldName="Microsoft.VSTS.TCM.SystemInfo" Type="HtmlFieldControl" Label="System
Info" LabelPosition="Top" Dock="Fill" />
</Tab>
<Tab Label="Test Cases">
    <Control Type="LinksControl" LabelPosition="Top" Name="TestedBy">
        <LinksControlOptions>
            <LinkColumns>
                <LinkColumn RefName="System.Id" />
                <LinkColumn RefName="System.WorkItemType" />
                <LinkColumn RefName="System.Title" />
                <LinkColumn RefName="System.AssignedTo" />
                <LinkColumn RefName="System.State" />
                <LinkColumn LinkAttribute="System.Links.Comment" />
            </LinkColumns>
            <WorkItemLinkFilters FilterType="include">
                <Filter LinkType="Microsoft.VSTS.Common.TestedBy" FilterOn="forwardname" />
            </WorkItemLinkFilters>
            <ExternalLinkFilters FilterType="excludeAll" />
            <WorkItemTypeFilters FilterType="include">
                <Filter WorkItemType="Test Case" />
            </WorkItemTypeFilters>
        </LinksControlOptions>
    </Control>
</Tab>
</TabGroup>
</Column>
<Column PercentWidth="50">
    <TabGroup Margin="(5,0,0,0)">
        <Tab Label="All Links">
            <Control Type="LinksControl" LabelPosition="Top" Name="GeneralLinks">
                <LinksControlOptions>
                    <LinkColumns>
                        <LinkColumn RefName="System.Id" />
                        <LinkColumn RefName="System.WorkItemType" />
                        <LinkColumn RefName="System.Title" />
                        <LinkColumn RefName="System.AssignedTo" />
                        <LinkColumn RefName="System.State" />
                        <LinkColumn RefName="Microsoft.VSTS.Common.Severity" />
                        <LinkColumn LinkAttribute="System.Links.Comment" />
                    </LinkColumns>
                </LinksControlOptions>
            </Control>
        </Tab>
        <Tab Label="Attachments">
            <Control Type="AttachmentsControl" LabelPosition="Top" />
        </Tab>
    </TabGroup>
</Column>
</Group>
<Group>
    <Column PercentWidth="100">
        <TabGroup>
            <Tab Label="History">
                <Control FieldName="System.History" Type="WorkItemLogControl" LabelPosition="Top" Dock="Fill"
/>
            </Tab>
        </TabGroup>
    </Column>
</Group>
</Layout>

```

```

</FORM>
</WORKITEMTYPE>
</Witd:WITD>

```

## 11.5. Описание рабочего элемента «Task»

```

<?xml version="1.0" encoding="utf-8"?>
<witd:WITD application="Work item type editor" version="1.0"
xmlns:witd="http://schemas.microsoft.com/VisualStudio/2008/workitemtracking/typedef">
  <WORKITEMTYPE name="Task">
    <DESCRIPTION>Tracks work that needs to be done.</DESCRIPTION>
    <FIELDS>
      <FIELD name="Iteration Path" refname="System.IterationPath" type="TreePath" reportable="dimension">
        <HELPTTEXT>The iteration within which this task will be completed</HELPTTEXT>
      </FIELD>
      <FIELD name="Iteration ID" refname="System.IterationId" type="Integer" />
      <FIELD name="External Link Count" refname="System.ExternalLinkCount" type="Integer" />
      <FIELD name="Team Project" refname="System.TeamProject" type="String" reportable="dimension" />
      <FIELD name="Hyperlink Count" refname="System.HyperLinkCount" type="Integer" />
      <FIELD name="Attached File Count" refname="System.AttachedFileCount" type="Integer" />
      <FIELD name="Node Name" refname="System.NodeName" type="String" />
      <FIELD name="Area Path" refname="System.AreaPath" type="TreePath" reportable="dimension">
        <HELPTTEXT>The area of the product to which this task contributes</HELPTTEXT>
      </FIELD>
      <FIELD name="Revised Date" refname="System.RevisedDate" type="DateTime" reportable="detail" />
      <FIELD name="Changed Date" refname="System.ChangedDate" type="DateTime" reportable="dimension" />
      <FIELD name="ID" refname="System.Id" type="Integer" reportable="dimension" />
      <FIELD name="Area ID" refname="System.AreaId" type="Integer" />
      <FIELD name="Authorized As" refname="System.AuthorizedAs" type="String" syncnamechanges="true" />
      <FIELD name="Title" refname="System.Title" type="String" reportable="dimension">
        <REQUIRED />
        <HELPTTEXT>Work required and how this will implement a User Story</HELPTTEXT>
      </FIELD>
      <FIELD name="State" refname="System.State" type="String" reportable="dimension">
        <HELPTTEXT>Active = work remains to be done. Closed = tested and checked in.</HELPTTEXT>
      </FIELD>
      <FIELD name="Authorized Date" refname="System.AuthorizedDate" type="DateTime" />
      <FIELD name="Watermark" refname="System.Watermark" type="Integer" />
      <FIELD name="Rev" refname="System.Rev" type="Integer" reportable="dimension" />
      <FIELD name="Changed By" refname="System.ChangedBy" type="String" syncnamechanges="true"
reportable="dimension">
        <ALLOWEXISTINGVALUE />
        <VALIDUSER />
      </FIELD>
      <FIELD name="Reason" refname="System.Reason" type="String" reportable="dimension">
        <HELPTTEXT>The reason why the task is in its current state</HELPTTEXT>
      </FIELD>
      <FIELD name="Assigned To" refname="System.AssignedTo" type="String" syncnamechanges="true"
reportable="dimension">
        <ALLOWEXISTINGVALUE />
        <VALIDUSER />
        <HELPTTEXT>The person currently working on this task</HELPTTEXT>
      </FIELD>
      <FIELD name="Work Item Type" refname="System.WorkItemType" type="String" reportable="dimension" />
      <FIELD name="Created Date" refname="System.CreatedDate" type="DateTime" reportable="dimension" />
      <FIELD name="Created By" refname="System.CreatedBy" type="String" syncnamechanges="true"
reportable="dimension" />
      <FIELD name="Description" refname="System.Description" type="HTML">
        <HELPTTEXT>What to do, pointers to resources and inputs, design notes, exit criteria</HELPTTEXT>
      </FIELD>
      <FIELD name="History" refname="System.History" type="History">
        <HELPTTEXT>Discussion thread plus automatic record of changes</HELPTTEXT>
      </FIELD>
      <FIELD name="Related Link Count" refname="System.RelatedLinkCount" type="Integer" />
      <FIELD name="State Change Date" refname="Microsoft.VSTS.Common.StateChangeDate" type="DateTime">
        <WHENCHANGED field="System.State">

```

```

        <SERVERDEFAULT from="clock" />
    </WHENCHANGED>
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Activated Date" refname="Microsoft.VSTS.Common.ActivatedDate" type="DateTime"
reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Activated By" refname="Microsoft.VSTS.Common.ActivatedBy" type="String"
synnamechanges="true" reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <ALLOWEXISTINGVALUE />
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Closed Date" refname="Microsoft.VSTS.Common.ClosedDate" type="DateTime"
reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Closed By" refname="Microsoft.VSTS.Common.ClosedBy" type="String"
synnamechanges="true" reportable="dimension">
    <WHENNOTCHANGED field="System.State">
        <ALLOWEXISTINGVALUE />
        <READONLY />
    </WHENNOTCHANGED>
</FIELD>
<FIELD name="Priority" refname="Microsoft.VSTS.Common.Priority" type="Integer"
reportable="dimension">
    <ALLOWEDVALUES expanditems="true">
        <LISTITEM value="1" />
        <LISTITEM value="2" />
        <LISTITEM value="3" />
        <LISTITEM value="4" />
        <LISTITEM value="5" />
        <LISTITEM value="6" />
        <LISTITEM value="7" />
        <LISTITEM value="8" />
        <LISTITEM value="9" />
    </ALLOWEDVALUES>
    <DEFAULT from="value" value="4" />
    <HELPTTEXT>Importance to business</HELPTTEXT>
</FIELD>
<FIELD name="Stack Rank" refname="Microsoft.VSTS.Common.StackRank" type="Double"
reportable="dimension">
    <HELPTTEXT>Work first on items with lower-valued stack rank. Set in triage.</HELPTTEXT>
</FIELD>
<FIELD name="Integration Build" refname="Microsoft.VSTS.Build.IntegrationBuild" type="String"
reportable="dimension">
    <SUGGESTEDVALUES expanditems="true">
        <LISTITEM value="&lt;None&gt;" />
    </SUGGESTEDVALUES>
    <SUGGESTEDVALUES expanditems="true" filteritems="excludegroups">
        <GLOBALLIST name="Builds - rnd" />
    </SUGGESTEDVALUES>
    <HELPTTEXT>The build in which the bug was fixed</HELPTTEXT>
</FIELD>
<FIELD name="Activity" refname="Microsoft.VSTS.Common.Activity" type="String"
reportable="dimension">
    <SUGGESTEDVALUES expanditems="true">

```

```

    <LISTITEM value="Development" />
    <LISTITEM value="Testing" />
    <LISTITEM value="Requirements" />
    <LISTITEM value="Design" />
    <LISTITEM value="Deployment" />
    <LISTITEM value="Documentation" />
    <LISTITEM value="UnitTesting" />
    <LISTITEM value="Investigation" />
    <LISTITEM value="Merge" />
    <LISTITEM value="Integration defect fix" />
  </SUGGESTEDVALUES>
  <REQUIRED />
  <HELPTTEXT>Type of work involved</HELPTTEXT>
</FIELD>
  <FIELD name="Remaining Work" refname="Microsoft.VSTS.Scheduling.RemainingWork" type="Double"
reportable="measure">
  <HELPTTEXT>An estimate of the number of units of work remaining to complete this task</HELPTTEXT>
  </FIELD>
  <FIELD name="Original Estimate" refname="Microsoft.VSTS.Scheduling.OriginalEstimate" type="Double"
reportable="measure">
  <HELPTTEXT>Initial value for Remaining Work - set once, when work begins</HELPTTEXT>
  </FIELD>
  <FIELD name="Completed Work" refname="Microsoft.VSTS.Scheduling.CompletedWork" type="Double"
reportable="measure">
  <HELPTTEXT>The number of units of work that have been spent on this task</HELPTTEXT>
  </FIELD>
  <FIELD name="Start Date" refname="Microsoft.VSTS.Scheduling.StartDate" type="DateTime"
reportable="dimension">
  <HELPTTEXT>The date to start the task</HELPTTEXT>
  </FIELD>
  <FIELD name="Finish Date" refname="Microsoft.VSTS.Scheduling.FinishDate" type="DateTime"
reportable="dimension">
  <HELPTTEXT>The date to finish the task</HELPTTEXT>
  </FIELD>
</FIELDS>
<WORKFLOW>
  <STATES>
    <STATE value="New">
      <FIELDS>
        <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
          <ALLOWEXISTINGVALUE />
          <EMPTY />
        </FIELD>
        <FIELD refname="Microsoft.VSTS.Scheduling.RemainingWork">
          <WHENNOTCHANGED field="Microsoft.VSTS.Scheduling.OriginalEstimate">
            <COPY from="field" field="Microsoft.VSTS.Scheduling.OriginalEstimate" />
          </WHENNOTCHANGED>
        </FIELD>
      </FIELDS>
    </STATE>
    <STATE value="Active">
      <FIELDS>
        <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
          <EMPTY />
        </FIELD>

```

```

<FIELD refname="Microsoft.VSTS.Common.ClosedBy">
  <ALLOWEXISTINGVALUE />
  <EMPTY />
</FIELD>
<FIELD refname="Microsoft.VSTS.Scheduling.OriginalEstimate">
  <FROZEN />
  <READONLY />
</FIELD>
</FIELDS>
</STATE>
<STATE value="Closed">
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
      <REQUIRED />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Scheduling.RemainingWork">
      <EMPTY />
      <FROZEN />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Scheduling.OriginalEstimate">
      <FROZEN />
      <READONLY />
    </FIELD>
  </FIELDS>
</STATE>
<STATE value="Removed" />
<STATE value="Review">
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Scheduling.OriginalEstimate">
      <FROZEN />
      <READONLY />
    </FIELD>
  </FIELDS>
</STATE>
</STATES>
<TRANSITIONS>
  <TRANSITION from="" to="New">
    <REASONS>
      <DEFAULTREASON value="New" />
    </REASONS>
  </TRANSITION>
  <TRANSITION from="New" to="Active">
    <ACTIONS>
      <ACTION value="Microsoft.VSTS.Actions.StartWork" />
    </ACTIONS>
    <REASONS>
      <DEFAULTREASON value="Work started" />
    </REASONS>
    <FIELDS>
      <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
        <ALLOWEXISTINGVALUE />
        <COPY from="currentuser" />
        <VALIDUSER />
        <REQUIRED />
      </FIELD>
      <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
        <SERVERDEFAULT from="clock" />
      </FIELD>
      <FIELD refname="System.AssignedTo">
        <DEFAULT from="currentuser" />
      </FIELD>
    </FIELDS>
  </TRANSITION>
  <TRANSITION from="Active" to="New">
    <ACTIONS>

```

```

    <ACTION value="Microsoft.VSTS.Actions.StopWork" />
  </ACTIONS>
  <REASONS>
    <DEFAULTREASON value="Work halted" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Active" to="Closed">
  <ACTIONS>
    <ACTION value="Microsoft.VSTS.Actions.Checkin" />
  </ACTIONS>
  <REASONS>
    <REASON value="Deferred" />
    <REASON value="Obsolete" />
    <REASON value="Cut" />
    <DEFAULTREASON value="Completed" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
      <SERVERDEFAULT from="clock" />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
      <ALLOWEXISTINGVALUE />
      <COPY from="currentuser" />
      <VALIDUSER />
      <REQUIRED />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="New" to="Closed">
  <ACTIONS>
    <ACTION value="Microsoft.VSTS.Actions.Checkin" />
  </ACTIONS>
  <REASONS>
    <REASON value="Deferred" />
    <REASON value="Obsolete" />
    <REASON value="Cut" />
    <DEFAULTREASON value="Completed" />
  </REASONS>
  <FIELDS>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
      <ALLOWEXISTINGVALUE />
      <READONLY />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
      <SERVERDEFAULT from="clock" />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
      <ALLOWEXISTINGVALUE />
      <COPY from="currentuser" />
      <VALIDUSER />
      <REQUIRED />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="Closed" to="Active">

```

```

<REASONS>
  <DEFAULTREASON value="Reactivated" />
</REASONS>
<FIELDS>
  <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
    <ALLOWEXISTINGVALUE />
    <COPY from="currentuser" />
    <VALIDUSER />
    <REQUIRED />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
    <SERVERDEFAULT from="clock" />
  </FIELD>
  <FIELD refname="System.AssignedTo">
    <COPY from="field" field="Microsoft.VSTS.Common.ClosedBy" />
  </FIELD>
</FIELDS>
</TRANSITION>
<TRANSITION from="Closed" to="New">
  <REASONS>
    <DEFAULTREASON value="Reactivated" />
  </REASONS>
  <FIELDS>
    <FIELD refname="System.AssignedTo">
      <COPY from="field" field="Microsoft.VSTS.Common.ClosedBy" />
    </FIELD>
  </FIELDS>
</TRANSITION>
<TRANSITION from="New" to="Removed">
  <REASONS>
    <DEFAULTREASON value="Removed from the backlog" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Active" to="Removed">
  <REASONS>
    <DEFAULTREASON value="Removed from the backlog" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Removed" to="New">
  <REASONS>
    <DEFAULTREASON value="Reconsidering the Task" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Active" to="Review">
  <REASONS>
    <DEFAULTREASON value="Review requested" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Review" to="Active">
  <REASONS>
    <DEFAULTREASON value="Task is not completed" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Review" to="Removed">
  <REASONS>
    <DEFAULTREASON value="Removed from backlog" />
  </REASONS>
</TRANSITION>
<TRANSITION from="Review" to="Closed">
  <ACTIONS>
    <ACTION value="Microsoft.VSTS.Actions.Checkin" />
  </ACTIONS>
  <REASONS>
    <DEFAULTREASON value="Review completed. All issues are resolved" />
  </REASONS>

```

```

<FIELDS>
  <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
    <ALLOWEXISTINGVALUE />
    <READONLY />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
    <READONLY />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
    <ALLOWEXISTINGVALUE />
    <REQUIRED />
    <COPY from="currentuser" />
    <VALIDUSER />
  </FIELD>
  <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
    <SERVERDEFAULT from="clock" />
  </FIELD>
</FIELDS>
</TRANSITION>
</TRANSITIONS>
</WORKFLOW>
<FORM>
  <Layout HideControlBorders="true" HideReadOnlyEmptyFields="true">
    <Group Margin="(4,0,0,0)">
      <Column PercentWidth="100">
        <Control FieldName="System.Title" Type="FieldControl" LabelPosition="Top" ControlFontSize="large"
EmptyText="&lt;Enter title here&gt;" />
      </Column>
    </Group>
    <Group Margin="(10,0,0,0)">
      <Column PercentWidth="30">
        <Group Label="Status">
          <Column PercentWidth="100">
            <Control FieldName="System.AssignedTo" Type="FieldControl" Label="Assi&gnd To"
LabelPosition="Left" EmptyText="&lt;No one&gt;" />
            <Control FieldName="System.State" Type="FieldControl" Label="Stat&e" LabelPosition="Left" />
            <Control FieldName="System.Reason" Type="FieldControl" Label="Reason" LabelPosition="Left" />
          </Column>
        </Group>
      </Column>
      <Column PercentWidth="20">
        <Group Label="Planning">
          <Column PercentWidth="100">
            <Control FieldName="Microsoft.VSTS.Common.StackRank" Type="FieldControl" Label="Stack Rank"
LabelPosition="Left" EmptyText="&lt;None&gt;" NumberFormat="DecimalNumbers" MaxLength="10" />
            <Control FieldName="Microsoft.VSTS.Common.Priority" Type="FieldControl" Label="Priority"
LabelPosition="Left" />
            <Control FieldName="Microsoft.VSTS.Common.Activity" Type="FieldControl" Label="Activity"
LabelPosition="Left" EmptyText="&lt;None&gt;" />
          </Column>
        </Group>
      </Column>
      <Column PercentWidth="30">
        <Group Label="Classification">
          <Column PercentWidth="100">
            <Control FieldName="System.AreaPath" Type="WorkItemClassificationControl" Label="&Area"
LabelPosition="Left" />
            <Control FieldName="System.IterationPath" Type="WorkItemClassificationControl"
Label="Ite&ratiion" LabelPosition="Left" />
          </Column>
        </Group>
      </Column>
      <Column PercentWidth="20">
        <Group Label="Effort (Hours)">
          <Column PercentWidth="100">

```



```

        <Control FieldName="Microsoft.VSTS.Scheduling.OriginalEstimate" Type="FieldControl"
Label="Original Estimate" LabelPosition="Left" />
        <Control FieldName="Microsoft.VSTS.Scheduling.RemainingWork" Type="FieldControl"
Label="Remaining" LabelPosition="Left" />
        <Control FieldName="Microsoft.VSTS.Scheduling.CompletedWork" Type="FieldControl"
Label="Completed" LabelPosition="Left" />
    </Column>
</Group>
</Column>
</Group>
<Group>
    <Column PercentWidth="50">
        <TabGroup>
            <Tab Label="Description">
                <Control FieldName="System.Description" Type="HtmlFieldControl" LabelPosition="Top" Dock="Fill"
/>
            </Tab>
            <Tab Label="Implementation">
                <Group>
                    <Column PercentWidth="50">
                        <Control FieldName="Microsoft.VSTS.Build.IntegrationBuild" Type="FieldControl"
Label="Integrated in Build" LabelPosition="Left" />
                    </Column>
                    <Column PercentWidth="50" />
                </Group>
                <Control Type="LinksControl" LabelPosition="Top" Name="Hierarchy">
                    <LinksControlOptions>
                        <LinkColumns>
                            <LinkColumn RefName="System.ID" />
                            <LinkColumn RefName="System.WorkItemType" />
                            <LinkColumn RefName="System.Title" />
                            <LinkColumn RefName="System.AssignedTo" />
                            <LinkColumn RefName="System.State" />
                            <LinkColumn LinkAttribute="System.Links.Comment" />
                        </LinkColumns>
                        <WorkItemLinkFilters FilterType="include">
                            <Filter LinkType="System.LinkTypes.Hierarchy" />
                        </WorkItemLinkFilters>
                        <ExternalLinkFilters FilterType="excludeAll" />
                        <WorkItemTypeFilters FilterType="include">
                            <Filter WorkItemType="Task" />
                            <Filter WorkItemType="User Story" />
                        </WorkItemTypeFilters>
                    </LinksControlOptions>
                </Control>
            </Tab>
        </TabGroup>
    </Column>
    <Column PercentWidth="50">
        <TabGroup>
            <Tab Label="All Links">
                <Control Type="LinksControl" LabelPosition="Top" Name="GeneralLinks">
                    <LinksControlOptions>
                        <LinkColumns>
                            <LinkColumn RefName="System.Id" />
                            <LinkColumn RefName="System.WorkItemType" />
                            <LinkColumn RefName="System.Title" />
                            <LinkColumn RefName="System.AssignedTo" />
                            <LinkColumn RefName="System.State" />
                            <LinkColumn LinkAttribute="System.Links.Comment" />
                        </LinkColumns>
                    </LinksControlOptions>
                </Control>
            </Tab>
            <Tab Label="Attachments">

```

```

        <Control Type="AttachmentsControl" LabelPosition="Top" />
    </Tab>
</TabGroup>
</Column>
</Group>
<Group>
    <Column PercentWidth="100">
        <TabGroup>
            <Tab Label="History">
                <Control FieldName="System.History" Type="WorkItemLogControl" LabelPosition="Top" Dock="Fill"
            />
            </Tab>
        </TabGroup>
    </Column>
</Group>
</Layout>
</FORM>
</WORKITEMTYPE>
</witd:WITD>

```

## 11.6. Описание компонента CommonProcessConfig

```

<?xml version="1.0" encoding="utf-8"?>
<CommonProjectConfiguration>
    <BugWorkItems category="Microsoft.BugCategory">
        <States>
            <State type="Proposed" value="New" />
            <State type="Proposed" value="Unconfirmed" />
            <State type="InProgress" value="Active" />
            <State type="InProgress" value="Resolved" />
            <State type="Complete" value="Closed" />
        </States>
    </BugWorkItems>
    <FeedbackRequestWorkItems category="Microsoft.FeedbackRequestCategory" plural="Feedback Requests">
        <States>
            <State type="InProgress" value="Active" />
            <State type="Complete" value="Closed" />
        </States>
    </FeedbackRequestWorkItems>
    <FeedbackResponseWorkItems category="Microsoft.FeedbackResponseCategory" plural="Feedback Responses">
        <States>
            <State type="InProgress" value="Active" />
            <State type="Complete" value="Closed" />
        </States>
    </FeedbackResponseWorkItems>
    <RequirementWorkItems category="Microsoft.RequirementCategory" plural="Stories">
        <States>
            <State type="Proposed" value="New" />
            <State type="Proposed" value="Unconfirmed" />
            <State type="InProgress" value="Active" />
            <State type="InProgress" value="Resolved" />
            <State type="Complete" value="Closed" />
        </States>
    </RequirementWorkItems>
    <TaskWorkItems category="Microsoft.TaskCategory">
        <States>
            <State type="Proposed" value="New" />
            <State type="InProgress" value="Active" />
            <State type="InProgress" value="Review" />
            <State type="Complete" value="Closed" />
        </States>
    </TaskWorkItems>
    <TypeFields>
        <TypeField refname="Microsoft.VSTS.Common.Activity" type="Activity" />
        <TypeField refname="Microsoft.VSTS.Common.StackRank" type="Order" />
        <TypeField refname="Microsoft.VSTS.Feedback.ApplicationLaunchInstructions"

```

```

type="ApplicationLaunchInstructions" />
  <TypeField refname="Microsoft.VSTS.Feedback.ApplicationStartInformation"
type="ApplicationStartInformation" />
  <TypeField refname="Microsoft.VSTS.Feedback.ApplicationType" type="ApplicationType">
    <TypeField Values>
      <TypeField Value type="ClientApp" value="Client application" />
      <TypeField Value type="RemoteMachine" value="Remote machine" />
      <TypeField Value type="WebApp" value="Web application" />
    </TypeField Values>
  </TypeField>
  <TypeField format="{0} h" refname="Microsoft.VSTS.Scheduling.RemainingWork" type="RemainingWork" />
  <TypeField refname="Microsoft.VSTS.Scheduling.StoryPoints" type="Effort" />
  <TypeField refname="System.AreaPath" type="Team" />
</TypeFields>
<Weekends>
  <DayOfWeek>Sunday</DayOfWeek>
  <DayOfWeek>Saturday</DayOfWeek>
</Weekends>
</CommonProjectConfiguration>

```

## 11.7. Модуль визуализации ID

```

/*! Copyright 2012 ELVEES NeoTek CJSC. All rights reserved. */
TFS.module("Elvees.RND.TFSTaskBoardIds", ["TFS.Host", "TFS.Admin"],
function () {
  $(document).ready(tileID); $(document).ready(rowID);
  function tileID() {
    $('[id^=tile-]').each(
    function () {
      var witTitle = $(this).find('.witTitle').first();
      if (witTitle.attr('_w') === undefined) {
        id = this.id.substring(5, 999);
        witTitle.attr('_w', id);
        witTitle.html("<strong>ID" + id + "</strong>|" + witTitle.text());
      }
    });
    setTimeout(tileID, 10);
  };
  function rowID() {
    $('[id^=taskboard-table_p]').each(
    function () {
      var witTitle = $(this).find('.witTitle').first();
      if (witTitle.attr('_w') === undefined) {
        id = this.id.substring(17, 999);
        witTitle.attr('_w', id);
        witTitle.html("<strong>ID" + id + "</strong>|" + witTitle.text());
      }
      else {
        if ($(location).attr('href').match(/.*team$/)) {
          witTitle.html(witTitle.html().replace(/<strong>.*\|/, ""))
        }
        else if (!(witTitle.html().match(/<strong>+.*))) {
          witTitle.html("<strong>ID" + witTitle.attr('_w') + "</strong>|" + witTitle.text())
        }
      }
    });
    setTimeout(rowID, 10);
  };
});
})

```

## 11.8. Модуль WorkItemChangedEventHandler

```

#region Using

```

```

using System;
using System.Reflection;

```

```

using Microsoft.TeamFoundation.Common;
using Microsoft.TeamFoundation.Client;
using Microsoft.TeamFoundation.Framework.Server;
using Microsoft.TeamFoundation.WorkItemTracking.Server;

using WorkitemStateTransition = System.Tuple <string, string>;

#endregion

namespace TFSServerEventHandle
{
    public class WorkItemChangedEventHandler : ISubscriber
    {
        public Type[] SubscribedTypes()
        {
            return new Type[1] { typeof(WorkItemChangedEvent) };
        }

        public EventNotificationStatus ProcessEvent( TeamFoundationRequestContext
requestContext      ,
                                                    NotificationType      notificationType ,
                                                    object                    notificationEventArgs,
                                                    out int                    statusCode      ,
                                                    out string                 statusMessage  ,
                                                    out ExceptionPropertyCollection properties )
        {
            statusCode = 0;
            properties = null;
            statusMessage = String.Empty;
            try
            {
                if (notificationType == NotificationType.Notification && notificationEventArgs is
WorkItemChangedEvent)
                {
                    var ev = notificationEventArgs as WorkItemChangedEvent;
                    Uri collectionUri = new Uri("http://localhost" +
requestContext.ServiceHost.VirtualDirectory.ToString());
                    ChangedWorkItem wit = new ChangedWorkItem(ref collectionUri,
GetCurrentWorkItemID(ref ev));
                    WorkitemStateTransition currentWorkItemStateTransition =
GetCurrentWorkItemStateTransition(ref ev);
                    #region TaskLogic

                    if (wit.GetChangedWorkItemTypeName() == "Task")
                    {
                        if (currentWorkItemStateTransition.Item2 !=
currentWorkItemStateTransition.Item1)
                        {
                            TaskChangeProcessor taskProcessor = new TaskChangeProcessor(ref wit, ref
currentWorkItemStateTransition);
                        }
                    }

                    #endregion
                }
            }
        }
    }
}

```

```

        #region USLogic

        if (wit.GetChangedWorkItemTypeName() == "User Story")
        {
            if (currentWorkItemStateTransition.Item2 !=
currentWorkItemStateTransition.Item1)
            {
                UserStoryChageProcessor usProcessor = new UserStoryChageProcessor(ref
wit, ref currentWorkItemStateTransition, ev.PortfolioProject.ToString());
            }
        }

        #endregion
    }

    }
    catch (Exception)
    {
    }
    return EventNotificationStatus.ActionPermitted;
}

public string Name
{
    get { return "WorkItemChangedEventHandler"; }
}

public SubscriberPriority Priority
{
    get { return SubscriberPriority.Normal; }
}

private int GetCurrentWorkItemID(ref WorkItemChangedEvent currentEvent)
{
    return Convert.ToInt32(currentEvent.CoreFields.IntegerFields[0].NewValue);
}

private WorkitemStateTransition GetCurrentWorkItemStateTransition(ref
WorkItemChangedEvent currentEvent)
{
    string OldValue = currentEvent.CoreFields.StringFields[3].OldValue.ToString();
    string NewValue = currentEvent.CoreFields.StringFields[3].NewValue.ToString();

    //Strange behavior of event - empty old value for non-new WI
    if (OldValue == "" && NewValue != "New")
    {
        return new WorkitemStateTransition(NewValue, NewValue);
    }
    else
    {
        return new WorkitemStateTransition(NewValue, OldValue);
    }
}
}

```

```

}

#region

using System;
using System.Collections.Generic;
using Microsoft.TeamFoundation.WorkItemTracking.Client;

#endregion

namespace TFSServerEventHandle
{
    class WorkItemChangeProcessor
    {
        public WorkItemChangeProcessor(ref ChangedWorkItem changedWit)
        {
            m_changedWit = changedWit;
            m_workItemStore = changedWit.GetChangedWorkItemStore();

            InitParentWorkitem();
        }

        private void InitParentWorkitem()
        {
            WorkItemLinkTypeEnd parentLinkTypeEnd =
m_workItemStore.WorkItemLinkTypes.LinkTypeEnds["Parent"];
            foreach (WorkItemLink link in m_changedWit.GetChangedWorkItemLinks())
            {
                if (link.LinkTypeEnd.Id == parentLinkTypeEnd.Id)
                {
                    m_parent = m_workItemStore.GetWorkItem(link.TargetId);
                    break;
                }
            }
        }

        protected List<WorkItem> GetAllChildsFromParent()
        {
            string wiqlQueryString = "Select * from WorkItemLinks where [Target].[System.Id] = '"
+ m_parent.Id + "'";
            wiqlQueryString += " and [Source].[System.State] <> 'Removed'";

            Query wiqlQuery = new Query(m_workItemStore, wiqlQueryString);
            WorkItemLinkInfo[] linkQueryResult = wiqlQuery.RunLinkQuery();

            List<WorkItem> witCollection = new List<WorkItem>();
            foreach (WorkItemLinkInfo link in linkQueryResult)
            {
                if (link.LinkTypeId == 0)//LinkTypeId of 'Child'
                {
                    witCollection.Add(m_workItemStore.GetWorkItem(link.TargetId));
                }
            }
        }
    }
}

```

```

        return witCollection;
    }

    protected ChangedWorkItem m_changedWit;
    protected WorkItem m_parent;
    protected WorkItemStore m_workItemStore;
}

}

#region

using System;
using System.Collections.Generic;
using Microsoft.TeamFoundation.WorkItemTracking.Client;

using WorkitemStateTransition = System.Tuple<string, string>;

#endregion

namespace TFSServerEventHandle
{
    class UserStoryChageProcessor : WorkItemChangeProcessor
    {
        public UserStoryChageProcessor(ref ChangedWorkItem changedUserStory, ref
WorkitemStateTransition stateTransition, string teamProjectName)
            : base(ref changedUserStory)
        {
            m_stateTransition = stateTransition;
            m_teamProject = m_workItemStore.Projects[teamProjectName];
            m_qaIterationId = 7; //Hardcode - Test ID of QA Iteration.
                //TODO: Implement automatic getter ID or Iteration object

            CreateQAItem();
        }

        private WorkItem CreateQAItem()
        {
            if (m_stateTransition.Item1.ToString() == "Resolved")
            {
                int changedWitId = m_changedWit.Id;
                WorkItemLinkType linkType =
m_workItemStore.WorkItemLinkTypes[CoreLinkTypeReferenceNames.Hierarchy];
                WorkItemLink parentLinkToChangedWit = new
WorkItemLink(linkType.ReverseEnd, changedWitId);

                WorkItemType workItemType = m_teamProject.WorkItemTypes["User Story"];
                WorkItem witQAItem = new WorkItem(workItemType);

                witQAItem.Title = "Test resolved User Story: ID - " + changedWitId;
                witQAItem.Description = m_changedWit.GetChangedWorkItemTitle();
                witQAItem.IterationId = m_qaIterationId;
                witQAItem.Save();
            }
        }
    }
}

```

```

        witQAItem.Links.Add(parentLinkToChangedWit);
        witQAItem.Save();

        return witQAItem;
    }

    return null;
}

private WorkitemStateTransition m_stateTransition;
private Project                m_teamProject;
private int                    m_qalterationId;
}
}

#region

using System;
using System.Collections.Generic;
using Microsoft.TeamFoundation.WorkItemTracking.Client;

using WorkitemStateTransition = System.Tuple<string, string>;

#endregion

namespace TFSServerEventHandle
{
    class TaskChangeProcessor: WorkItemChangeProcessor
    {
        public TaskChangeProcessor(ref ChangedWorkItem changedTask, ref
WorkitemStateTransition stateTransition)
            : base(ref changedTask)
        {
            m_stateTransition = stateTransition;
            ChangeParentState();
        }

        public void MainProcess()
        {
            //maybe unnecessary
        }

        private string GetParentWorkItemState()
        {
            return m_parent.State.ToString();
        }

        private void ChangeParentState()
        {
            string parentState = GetParentWorkItemState();
            if (m_stateTransition.Item1 == "Active")
            {
                m_parent.Open();
                m_parent.State = "Active";
            }
        }
    }
}

```



```

        m_parent.Save();
        m_parent.Close();
    }
    if (m_stateTransition.Item1 == "Closed")
    {
        List<WorkItem> childsWitCollection = GetAllChildsFromParent();
        childsWitCollection.RemoveAll(wit => wit.State.ToString() == "Closed");

        if (childsWitCollection.Count == 0)
        {
            m_parent.Open();
            m_parent.State = "Resolved";
            m_parent.Save();
            m_parent.Close();
        }
    }
}

private WorkitemStateTransition m_stateTransition;
}

#region

using System;
using Microsoft.TeamFoundation.Client;
using Microsoft.TeamFoundation.WorkItemTracking.Client;

#endregion

namespace TFSServerEventHandle
{
    public class ChangedWorkItem
    {
        public ChangedWorkItem(ref Uri collectionUri, int workitemID)
        {
            TfsTeamProjectCollection teamProjectCollection = new
TfsTeamProjectCollection(collectionUri);
            WorkItemStore workItemStore =
teamProjectCollection.GetService<WorkItemStore>();

            m_workItemId = workitemID;
            m_workItem = workItemStore.GetWorkItem(workitemID);
        }

        public Field GetChangedWorkitemField(string fieldName)
        {
            if (m_workItem.Fields.Contains(fieldName))
            {
                return m_workItem.Fields[fieldName];
            }
            else
            {
                return null;
            }
        }
    }
}

```

```

    }
}

public WorkItemType GetChangedWorkItemType()
{
    return m_workItem.Type;
}

public string GetChangedWorkItemTypeName()
{
    return m_workItem.Type.Name.ToString();
}

public string GetChangedWorkItemTitle()
{
    return m_workItem.Title.ToString();
}

public WorkItemStore GetChangedWorkItemStore()
{
    return m_workItem.Store;
}

public WorkItemLinkCollection GetChangedWorkItemLinks()
{
    return m_workItem.WorkItemLinks;
}

public int Id
{
    get { return m_workItemId; }
}

private WorkItem m_workItem;
private int    m_workItemId;
}
}

```