Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Факультет непрерывного и дистанционного обучения

Кафедра информатики

Электронный учебно-методический комплекс

по дисциплине

**Современные средства проектирования информационных систем**

Для студентов специальности

**1-31 03 04 “Информатика“**

Минск 2011

# Общие сведения

## Сведения об ЭУМК

Данный комплекс специально разработан для студентов, обучающихся дистанционно. Целью его создания было сделать процесс изучения данной дисциплины максимально удобным и комфортным для студентов. Кроме того, обладание комплексом должно позволить студенту минимизировать свои непосредственные контакты с университетом и компьютером, что особенно удобно для иногородних студентов, а также тех, кто по различным причинам вынужден на долгое время уезжать из города. В принципе данный комплекс позволяет студенту изучить дисциплину и подготовиться к сдаче экзамена по ней “автономно”

Выше приведён перечень и описание составных частей данного комплекса. Они выполнены в виде гиперссылок и для перехода к нужной части требуется, удерживая клавишу “Ctrl”, щёлкнуть левой кнопкой мыши выбранную гиперссылку. Впрочем, поскольку весь комплекс представляет собой файл Microsoft Word, его можно просматривать в обычном режиме, а также полностью или частично печатать.

**Составитель: ПилецкийИ.И.**, доцент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»**,** кандидат ф.-м.н., **Шиманский В.В.** ассистент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

Рассмотрен и рекомендован к изданию на заседании кафедры информатики, протокол № \_\_ от \_\_.\_\_.2011.

## Методические рекомендации по изучению дисциплины

В соответствии с учебным планом студенты дистанционной формы обучения инженерных специальностей изучают курс «Современные средства проектирования информационных систем».

Учебным планом по данному курсу предусмотрено изучение теоретических вопросов, решение задач, выполнение 2 контрольных работ и 2 ИПР с ИКТ. Изучение курса заканчивается сдачей зачета. К сдаче зачета студенты допускаются только при условии выполненных и защищенных лабораторных работ.

Рекомендуется изучать курс «Современные средства проектирования информационных систем» в соответствии с рабочей программой. Сначала необходимо ознакомиться с содержанием курса, затем изучить рекомендуемую литературу, обращая внимание на вопросы, выделенные в рабочей программе, после чего изучить теоретическое изложение курса по приведенным разделам, темам и вопросам, ответить на контрольные вопросы, выполнить задачи для решения (выполнения лабораторных работ) в соответствии с заданием.

Так как теоретический материал излагается в строгой логической последовательности, рекомендуется изучать данную дисциплину, придерживаясь данной логики.

## Рабочая учебная программа

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

УТВЕРЖДАЮ

Декан факультета непрерывного и дистанционного обучения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. М. Бондарик

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2010 г.

Регистрационный № -\_\_\_/.

**Современные средства проектирования информационных систем**

Рабочая учебная программа

**для направления специальности 1-31 03 04**

**“Информатика“**

Факультет **непрерывного и дистанционного обучения**

Кафедра **информатики**

Курс **шестой (семестр 11)**

**Контрольные работы** **2 работы**

**ИПР с ИКТ 2 работы**

Всего часов **110 часов**

Зачет **6 курс**

Форма получения

высшего образования **дистанционная**

Минск 2011

Составиль Пилецкий И.И., Шиманский В.В.

Учебная программа составлена на основе типовой учебной программы  *«Современные средства проектирования информационных систем»* для специальности 1-31 03 04 Информатика, утвержденной Министерством образования Республики Беларусь 14. 04. 2010 регистрационный № 31-25/баз.

Рассмотрена и рекомендована к утверждению на заседании кафедры информатики

протокол № \_\_ от \_\_\_\_\_ \_\_\_\_

Заведующий кафедрой Минченко Л.И.

Одобрена и рекомендована к утверждению Научно-методической комиссией факультета компьютерных систем и сетей Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

протокол № \_\_ от \_\_\_\_\_ \_\_\_

Председатель Лукашевич М.М.

СОГЛАСОВАНО

Начальник отделаметодического обеспечения

учебного процесса \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ц. С. Шикова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цель преподавания дисциплины. Дисциплина «Современные средства проектирования информационных систем» ставит своей целью подготовку студентам по теоретическим и практическим аспектам дисциплины “Программная инженерия” как методологии интеграции ИТ знаний, управление программными проектами, определение технологии разработки ПО на основе различных моделей жизненного цикла ПО, современные средства разработки больших корпоративных приложений.

Задачи изучения дисциплины. В результате изучения дисциплины студенты должны интегрировать знания, полученные в процессе обучения, на основе полученных знаний об управлении программными проектами, на основе знаний о работах, выполняемых по разработке больших приложений на основе жизненного цикла разработки промышленного программного продукта и фундаментальных принципов технологий разработки программ и баз данных, научиться, на регулярной основе, применять CASE системы на практике.

В результате изучения дисциплины выпускник должен

знать:

- методы и средства управления программными проектами, методы и средства интеграции данных, приложений и действий пользователя уровня больших корпораций;

- технологию разработкипромышленного программного продукта сна основе применения CASEсистем уровня корпорации, методологию и технологию разработки и реализации бизнес - процессов, а так же международные стандарты производства промышленного программного продукта;

уметь:

- применять изученные методологии, технологии и инструменты, позволяющие получить качественный программный продукт, по проектированию, разработке и сопровождению промышленного программного обеспечения или промышленного программного продукта.

Перечень дисциплин, усвоение которых необходимо для изучения данной дисциплины

Дисциплина «Технология программирования» непосредственно связана и базируется на знании дисциплин «Технология программирования», «Дискретная математика и математическая логика», «Конструирование программ и языки программирования», «Объектно ориентированное программирование».

**СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

1. Название тем теоретического материала, их содержание, рекомендуемый объем в часах

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номертемы | Название тем | Контрольная работа (номер по п.2) | Индивидуальная практическая работас указанием вида (по п.1) | Оснащение контрольных и индиивидуальных практических работ(по п.6) | Литература (по п.5) | Рекомендуемый объем для изучения (в часах)2 | Форма контроля знаний (зачет по контрольной работе, тесты, защита лабораторной работы, защита курсового проекта, экзамен, зачет) |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **Шестой курс** | | | | | | | |
| 1 | **Управление программными проектами**  1.1. Введение в программную инженерию. Модели процесса разработки ПО.  1.1.1. Итеративная разработка.  1.1.2. Эффективное управление требованиями.  1.1.3. Компонентный подход  1.1.4. Визуальное проектирование системы.  1.2. Контроль качества продуктов. Команда проекта  1.2.1. Гарантия качества продуктов.  1.2.2. Команда проекта. |  |  |  | 5.1.1  5.1.2  5.1.3 | 15 |  |
| 2 | Проблемы разработки больших корпоративных проектов и методы их решения. 2.1. Организация совместной работы на основе IBM Team Unifying Platform 2.2. Методология Rational Unified Process. |  | 1 |  | 5.1.1  5.1.2  5.1.3 | 60 | зачет по индивидуальной практической работе |
| 3 | Корпоративная шина предприятия. Назначение, основные функции. 3.1. Моделирование и проектирование.  3.2. Средства разработки. | 1 | 2 |  | 5.1.1  5.1.2  5.1.3 | 25 | зачет по контрольной работезачет по индивидуальной практической работе |
| 4 | Тестирование и отладка программ4.1. Тестирование. 4.2. Управление выполнением проекта и документированием.IBM Rational ProjectConsole, IBM Rational SoDA.  4.3. Управление конфигурациями и изменениями. | 2 |  |  | 5.1.1  5.1.2  5.1.3  5.2.1  5.2.2 | 10 | зачет по контрольной работе |

1. ЛАБОРАТОРНЫЕ ЗАНЯТИЯ, ИХ ХАРАКТЕРИСТИКА

Лабораторные работы учебным планом не предусмотрены

**2.КОНТРОЛЬНЫЕ РАБОТЫ, ИХ ХАРАКТЕРИСТИКА**

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Тема | Содержание | Объем в часах |
| 1. | Назначение и основные функции корпоративной шины предприятия. | Трансформация и интеграция данных на примерах применения IBMWebShpereDataStage. Построение схем и реализация схем преобразования данных | 2 |
| 2. | Реализация бизнес – процессов на основе ESB шины предприятия. | Реализация бизнес процессов посредством ESB шины IBMWebShpereMessageBroker. | 4 |

3.ИНДИВИДУАЛЬНЫЕ ПРАКТИЧЕСКИЕ РАБОТЫ, ИХ ХАРАКТЕРИСТИКА

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Тема | Характеристика | Рекомендуемыйобъем в часах |
| 1 | 2 | 3 | 4 |
| 1. | Генерация скриптов баз данных. Reverseengineering. | Освоить основные функции пакета ErWinDataModeler, типы поддерживаемых диаграмм, основные функции по созданию, редактированию, обновлению. Изучить стандарт IntegrationDefinitionforInformationModeling(IDEF1x). Использую среду ErWin разработать информационную диаграмму базы данных предприятия. Изучить различия между логической и физической моделью данных. На основе диаграммы, разработанной в лабораторной работе №4, создать скрипт базы данных, сгенерировать базу данных для выбранной СУБД, произвести изменения в структуре сгенерированной базы данных и выполнить reverseengineering. | 2 |
| IDEF4. Составление диаграмм классов. | Изучить пакет RationalRose, основные функции и типы поддерживаемых диаграмм, освоить IDEF4 Object-OrientedDesignMethod. Создать диаграмму классов, заполнить классы основными полями и методами. Составить систему связей представленных на диаграмме классов. | 4 |

4.КУРСОВЫЕ РАБОТЫ (ПРОЕКТЫ), ИХ ХАРАКТЕРИСТИКА

Курсовые работыучебным планом не предусмотрены

5. ЛИТЕРАТУРА

5.1.ОСНОВНАЯ

* + 1. ИСО/МЭК 12207 – 95 «Информационная технология. Процессы жизненного цикла программных средств» или ISO/IEC 12207 (ISO – InternationalOrganizationofStandardization)

5.1.2. ГОСТ 34.601–90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы, стадии создания.

* + 1. Официальный сайт компании IBM<http://www.ibm.com/developerworks/ru/webservices/newto/>

Ibm.com/redbooks

5.2. Дополнительная

5.2.1. Бобровский С. Технологии Пентагона на службе российских программистов. Программная инженерия. – СПб.: Питер, 2003. – 222с.

5.2.2. MicrosoftSolutionFramework.Дисциплина управления рисками MSF. Вер. 1.1, 2002.

# Теоретический раздел

## Лекции

### ВВЕДЕНИЕ

При создании сложного программного обеспечения (ПО) для корпоративных информационных систем (ИС) требуется четко и грамотно организовать весь процесс разработки/заказа ПО — от написания технического задания до внедрения на предприятии и дальнейшего развития этого ПО.

Среди основных проблем, возникающих при разработке ПО без использования специальных технологий, можно выделить следующие:

• **Разночтения в требованиях.** Разработчики и пользователи разговаривают на "разных языках", что не позволяет точно перевести разрозненные неформальные требования в целостную формальную спецификацию системы. В результате трудно создать систему, отвечающую требованиям пользователей. Необходимы постоянные доработки и изменения.

• **Отсутствие “чертежей”.** Отсутствие проектных спецификаций ("чертежей") на систему приводит к отсутствию структуры и единой концепции системы. Развитие такой системы трудоемко и ведет к дальнейшему росту "хаотичности".

• Документирование постфактум. Трудоемкость документирования в ходе разработки выливается либо в неприемлемые сроки создания точной проектной документации в соответствии с требованиями стандартов, либо в неприемлемое качество документации, что влечет за собой проблематичность последующей модификации ПО ИС.

• Ошибки проектирования. Ошибки, возникающие на этапах анализа и проектирования, часто не удается обнаружить до самого начала внедрения, когда уже стоимость их исправления становится на порядок выше.

• Отсутствие общего контекста проекта. Подсистемы, создаваемые разными группами разработчиков, трудно интегрировать из-за отсутствия или недостаточной проработки общего контекста проекта.

• Обособленность проекта. Информационные системы не переносятся с одной платформы на другую, имеют сложное взаимодействие с внешними системами и являются тяжелыми для последующего развития. В результате разработка нового и изменение существующего программного обеспечения отнимают слишком много времени и средств.

Мировой опыт показывает, что для успешного создания подобного ПО необходимы апробированные современные методологии, опирающиеся на мощные и удобные инструментальные средства.

Осуществление таких проектов в заданные сроки с высоким качеством невозможно без применения инженерных методов автоматизации программного производства, т.е. без современных CASE-технологий.

### ТЕМА 1.Управление программными проектами

#### 1.1. Основные принципы организации работы над проектом

Ведущие идеологи инструментальной инфраструктуры IBM Rational (Г. Буч, Дж. Рамбо, А.Джекобсон), проанализировав опыт различных проектов в области разработки ИС, выделяют следующие обязательные факторы для успешного ведения любого проекта:

• постоянное взаимодействие с потенциальными пользователями с целью выяснения реальных требований к системе;

• тщательно проработанная архитектура системы, открытая для возможных усовершенствований;

• наличие высококвалифицированных специалистов;

• грамотно подобранный инструментарий;

• определение верного направления работ;

• продуманный процесс разработки, обеспечивающий адаптацию к изменяющимся потребностям бизнеса и требованиям новых технологий;

• высокая степень управляемости проектом и получение достоверной информации по его состоянию в любой момент времени.

При наблюдаемом в настоящее время взрывном росте количества приложений, как для исполнителя, так и для заказчика, необходимо выполнение высококачественных программных проектов быстрее, чем когда бы то ни было.

Программные проекты должны завершаться в ограниченные сроки и при этом оставаться в рабочем состоянии с гарантией качества. Возникает ключевая проблема — необходимо достичь баланса между качеством исполнения и скоростью разработки. Решения IBM Rational помогут вам преодолеть эту проблему, объединяя лучший опыт и методологии разработки, соответствующие требованиям качества SEI CMM/CMMI, с унифицированными инструментами и сервисом, ускоряющими промышленную разработку ПО.

Основная задача, которая должна быть решена для достижения успеха в проекте, — научиться разрабатывать и производить программное обеспечение наиболее предсказуемым и повторяемым образом. Участники проектов должны уметь повторять свой успех в будущих работах и своевременно устранять обнаруженные недостатки. Чтобы гарантировать успех в любом проекте, важно использовать стандартные практики, которые давно уже являются обязательными:

• разрабатывать ПОна основе итеративных принципов;

• управлять требованиями наиболее эффективным образом;

• использовать компонентный подход;

• проектировать систему с помощью визуальных средств;

• гарантировать качество создаваемых продуктов;

• контролировать любые изменения в ходе проекта.

* + 1. Итеративная разработка

Классический подход, широко применявшийся в прошлом и до сих пор часто встречающийся в настоящее время, — разработка программного обеспечения по методу “водопада”

При этом подходе разработка ПО движется линейно через стадии анализа требований, дизайна (проектирования), кодирования и тестирования отдельных модулей (компонентов), тестирование сборок и интегрированное тестирование всего конечного продукта. Основная проблема здесь заключается в том, что происходит нарастание риска преждевременного крушения проекта из-за накапливания различных ошибок, допущенных на ранних стадиях проекта. Если только к концу проекта, становится очевидно, что такие ошибки были допущены, то любой возврат к предыдущим стадиям с целью исправления ошибок становится крайне дорогостоящим. Метод “водопада” не позволяет эффективно выявлять и нивелировать последствия подобных рисков. “Если вы сами активно не атакуете риски, то потом они будут активно атаковать вас” (Том Глиб, IBM Rational).

Эффективной альтернативой методу “водопада” служит итеративный подход

Основой итеративного подхода является непрерывный анализ выполненных работ, последующее проектирование и физическое воплощение результатов проектирования. Итеративный подход акцентирует работу команды в более предсказуемом и повторяемом направлении. Основные преимущества итеративного подхода:

• нивелирование воздействия серьезных рисков на ранних стадиях проекта, пока это еще можно сделать с минимальными затратами;

• возможность организовать плодотворную обратную связь с будущими конечными пользователями с целью создания системы, реально отвечающей их потребностям;

• акцент усилий на наиболее важные и критичные направления проекта.

• непрерывное итеративное тестирование конечного продукта, позволяющее оценить успешность всего проекта в целом;

• раннее обнаружение несоответствий между требованиями, моделями и программным кодом;

•более равномерная загрузка участников проекта;

• эффективное использование накопленного опыта;

• реальная оценка текущего состояния проекта и, как следствие, большая уверенность заказчиков и непосредственных участников в его успешном завершении.

* + 1. Эффективное управление требованиями

Неизбежной реальностью проектов создания современных ИС, является то, что список требований к системе динамически изменяется в ходе всего жизненного цикла. В любой момент следует ожидать пересмотра и корректировки требований. Более того, непрерывное выявление и уточнение реальных требований к системе — один из путей успешного ведения проекта. На старте проекта практически невозможно наиболее точным образом и в полном объеме определить необходимые требования, за исключением наиболее простых систем.

Реализация каждого отдельного требования представляет собой добавление в систему некоторой способности, которой та должна соответствовать. Процесс управления требованиями охватывает несколько деятельностей: выявление, организацию и документирование. Существуют проверенные решения, которые позволяют гарантировать более эффективное ведение проекта, основанное на грамотном управлении требованиями. Они опираются на следующие аспекты:

• организованный подход к управлению требованиями;

• взаимодействие участников проекта на базе выявленных и утвержденных требований; ранжирование требований по приоритету, фильтрация их по необходимым параметрам и выявление зависимости между ними для контроля изменений;

• объективная оценка реализованного функционала и полученной производительности;

• раннее предсказание различных несоответствий и расхождений;

• использование инструментальных средств для организации более эффективного процесса управления требованиями.

##### 1.1.3.Компонентный подход

Компонентный подход при создании ПО крайне важен, т. к. он обеспечивает повторное использование уже существующих компонентов и позволяет эффективно распределить работу между участниками проекта.

Применение компонентного подхода дает возможность повысить эффективность процесса разработки следующим образом:

• использование компонентов повышает гибкость архитектуры создаваемой системы;

• модульность компонентов позволяет четко определить границы тех изменений, которые требуется внести в систему при ее доработке;

• наличие множества коммерческих компонентов, которые разработаны и протестированы третьими фирмами, а также построены на основе промышленных спецификаций COM+, CORBA, Enterprise Java

Beans (EJB) и др., облегчает реализацию и позволяет экономить проектные ресурсы;

• программные компоненты задают естественную основу для конфигурируемости продукта;

• средства визуального моделирования обеспечивают автоматизацию процесса разработки, опирающегося на компонентный подход.

* + 1. Визуальное проектирование системы

Модель — это абстрагирование, упрощение реальности, позволяющее описать систему в предопределенном разрезе. Модели формируются для выявления того, какая система в конечном итоге должна быть построена.

Используя универсальный язык моделирования UML (Unified Modeling Language), участники проекта могут эффективно взаимодействовать друг с другом. Продуманное моделирование в различных его проявлениях предоставляет следующие возможности:

• однозначное описание функционального поведения системы с помощью прецедентов и сценариев;

• спецификация и анализ технических особенностей системы с помощью моделей;

• ранний акцент на построение гибкой и надежной архитектуры.

• сокрытие излишней детализации;

• выявление и исключение на ранних стадиях проекта ошибок проектирования.

#### Контроль качества продуктов. Команда проекта

##### 1.2.1. Гарантия качества продуктов

Чем раньше участники проекта начинают заботиться о качестве разрабатываемой системы, тем дешевле им это обходится.

Непрерывный контроль качества реализуется с помощью тестирования. Данный процесс предполагает создание тестов для каждого ключевого сценария, реализуемого в системе. Качество системы проявляется, прежде всего, в количестве успешных и неуспешных сценариев, что как раз и выявляется в процессе тестирования. Тестирование и разработка новых тестовых сценариев проводятся на каждой итерации проекта. Наборы сценариев и программных скриптов дорабатываются итеративно вместе с создаваемым продуктом.

Непрерывный контроль качества приводит к следующим позитивным моментам:

• оценка состояния проекта приобретает большую объективность, т. к. оценивается реальное функционирование системы, а не качество проектной документации;

• оценка проекта позволяет раскрыть несоответствия в требованиях, моделях и реализации;

• тестирование акцентирует внимание на тех сторонах работы системы, которые имеют наибольшую важность и повышенный риск;

• дефекты выявляются на ранних стадиях, что снижает затраты на их устранение;

• автоматизированное тестирование обеспечивает высокий уровень функциональности системы, надежности и производительности.

##### 1.2.1. Команда проекта

Для любого проекта характерна задача по организации работы между различными людьми и их группами. Отсутствие продуманного управления процессом разработки неизбежно приводит к возникновению хаотичности проекта и снижению его эффективности. Основой для объективного мониторинга проекта служит налаженный контроль изменений, который позволяет своевременно реагировать на возникающие проблемы. Контроль изменений приводит к следующему:

• деятельность по изменению требований становится предсказуемой и повторяемой;

• запросы на изменения формируют основу четкого и налаженного взаимодействия между участниками проекта;

• статистика по запросам на изменения создает отличную базу для оценки состояния проекта в любой момент времени;

• все изменения проекта находятся под контролем;

• с помощью контролируемых изменений строится система, адекватно отвечающая потребностям заказчиков.

### ТЕМА 2. Проблемы разработки больших корпоративных проектов и методы их решения.

#### 2.1. Организация совместной работы на основе IBM Team Unifying Platform

Инструментарий IBM Rational Team Unifying Platform предназначен для организации совместной работы в составе многофункциональных коллективов с целью повышения производительности как команды в целом, так и отдельных специалистов. Использование средств коллективной разработки IBM Rational

помогает синхронизировать работу разнопрофильных специалистов, вовремя предупреждая всех участников проекта об изменениях.

Довольно часто члены команды проекта взаимодействуют недостаточно эффективно, что замедляет обмен информацией и служит причиной потери многих часов рабочего времени. Благодаря полному набору средств для ведения коллективной разработки, тесно интегрированных друг с другом, все члены проектной команды получают общее видение проекта. Специалисты более не изолированы и не ограничены рамками выполняемой функции. Инструментарий IBM Rational предупреждает специалистов по тестированию о том, что им необходимо изменить сценарий тестирования, разработчиков — об изменении приоритетов требований, а руководителя проекта — о превышении установленного предела серьезных ошибок. Такой обмен информацией крайне важен для повышения продуктивности в работе над проектом.

Набор средств IBM Rational под названием Team Unifying Platform (Объединяющая платформа)

включает в себя следующие средства совместного управления проектами и создания коллективной инфраструктуры:

• IBM Rational Unifying Process — методология создания информационных систем, обеспечивает всеобщее понимание и следование итеративному подходу;

• IBM Rational RequisitePro — средство управление требованиями, которое позволяет коллективу справляться с постоянно меняющимися требованиями;

• IBM Rational ClearQuest — средство управления запросами на изменения, предоставляет возможности для отслеживания дефектов и изменений, которое позволяет каждому понимать текущий статус качества ПО;

• IBM Rational ClearCase — средство конфигурационного управления, надежный инструмент для управления версиями, который служит в качестве общего репозитария для всех объектов разработки ИС;

• IBM Rational TestManager — средство организации процессов тестирования, которое объединяет в себе планирование тестов, их выполнение и анализ, включая частные и унаследованные тестовые объекты;

• IBM Rational Project Console - средство мониторинга ключевых показателей проекта, которое упрощает контроль за статусом проекта и генерирование объективных метрик проекта с целью улучшения его предсказуемости.

• IBM Rational SoDA — средство документирования, автоматизирует создание и сопровождение всеобъемлющей проектной документации и отчетов

В основе работы инструментария лежит подход Unifying Change Management (UCM), который позволяет организовать индивидуальные рабочие пространства для каждого участника проекта на базе единого проектного репозитария. В зависимости от специфики деятельности конкретного участника проекта он использует необходимые инструментальные средства. Интеграция указанных средств позволяет объединить деятельность всей команды в единый и четкий процесс построения эффективной ИС.

#### 2.2. Методология Rational Unified Process

В основе многих программных методологий, объединяющих инженерные методы создания ПО, лежит

"пошаговый подход". Он определяет этапы жизненного цикла, контрольные точки, правила работ для каждого этапа и, тем самым, упорядочивает проектирование и разработку ПО.

Для каждого этапа жизненного цикла методология задает:

• состав и последовательность работ, а также правила их выполнения;

• распределение полномочий среди участников проекта (роли);

• состав и шаблоны формируемых промежуточных и итоговых документов;

• порядок контроля и проверки качества.

Ведущей методологией, в которой инструментально поддерживаются все этапы жизненного цикла разработки ПО, является методология Rational Unified Process (RUP). Она опирается на проверенные практикой методы анализа, проектирования и разработки ПО, методы управления проектами. RUPобеспечивает прозрачность и управляемость процесса и позволяет создавать ПОв соответствии с требованиями заказчика на момент сдачи ПО, а также в соответствии с возможностями инструментальных средств поддержки разработки.

RUP как методология

Методология RUP позволяет объединить проектную команду, предоставляя в ее распоряжение проверенные мировой практикой лучшие подходы к разработке ИС. К ним относятся такие процессы жизненного цикла создания ПО, как управление проектами, бизнес-моделирование, управление требованиями, анализ и проектирование, тестирование и контроль изменений. Внедрение RUP в организации способствует выработке качественных внутрикорпоративных стандартов и повышению общей культуры разработки.

**Основа RUP — итеративный процесс разработки.** В условиях активно развивающегося мирового бизнеса практически невозможно создавать современные сложные программные системы последовательно, т. е. сначала выявлять все проблемы, затем принимать проектные решения, потом формировать программное обеспечение и, наконец, проверять полученное изделие. Итеративный подход позволяет улучшать понимание проблем на основе последовательных усовершенствований и конкретизировать их в эффективных решениях. Этот подход обеспечивает большую гибкость при изменяющихсятребованиях и тактических коррективах в бизнес-целях, что позволяет более эффективно и заблаговременно идентифицировать и снижать проектные риски.

**RUP — управляемый процесс.** Итеративный подход предполагает управление требованиями и изменениями, чтобы между всеми участниками проекта обеспечивать единое понимание ожидаемых функциональных возможностей, требуемый уровень качества, наилучшее управление затратами и графиками выполнения работ.

**RUP — процесс создания и физического воплощения визуальных моделей.** RUP фокусирует внимание не на создании большого количества бумажных документов, а на развитии и применении визуальных моделей — семантически богатых представлений разрабатываемой ИС. RUPсосредотачивает внимание на разработке и дальнейшем развитии надежной и гибкой архитектуры, которая облегчает параллельную разработку, минимизирует необходимость изменений, увеличивает возможность многократного использования и надежность эксплуатации системы. Подобная архитектура применяется для планирования использования программных компонентов и управления их развитием.

**RUP — процесс управления действиями с помощью прецедентов, определяющих функционал системы.** Понятия прецедентов и сценариев работы способствуют эффективному управлению технологическим маршрутом от бизнес-моделирования и требований вплоть до испытаний. Они обеспечивают связанные и доступные для анализа направления разработки и развертывания системы.

**RUP поддерживает объектно-ориентированную технологию.** Моделирование по методологии RUPявляется объектно-ориентированным и базируется на понятиях объектов, классов и зависимостей между ними. Эти модели, подобно многим другим техническим искусственным объектам

(артефактам), в качестве единого стандарта для организации взаимодействия участников проекта используют Unified Modelling Language™ (UML) — универсальный язык моделирования.

**RUP поддерживает компонентно-ориентированный подход.** Компоненты — это нетривиальные модули или подсистемы, которые выполняют конкретную функцию и могут быть использованы многократно. Как правило, компоненты соответствуют одной из промышленныхспецификаций, таких как CORBA, COM/DCOM, ActiveX, EnterpriseJavaBeans и др.

**RUP — адаптируемый и конфигурируемый процесс.** Опыт единичного проекта, даже успешно завершенного, вряд ли подойдет для создания ПО во всех случаях и условиях. Но способность RUPк адаптации подойдет как маленьким группам разработчиков, так и большим организациям. RUPсодержит рекомендации по конфигурированию процесса для удовлетворения потребностей практически любых компаний и их подразделений.

**RUP поддерживает объективно осуществляемое управление качеством.** Оценка качества всех работ, выполняемых любыми участниками проекта, использует объективные метрики и критерии.

Методология RUP создавалась с прицелом на поддержку управления качеством в рамках требований SEI CMM/CMMI.

Структура жизненного цикла проекта

Структуру жизненного цикла проекта, выполняемого по технологии RUP удобно рассматривать на координатной плоскости. При этом по горизонтальной оси отложено время, а по вертикальной —

основные деятельности, которые обычно выполняются в ходе любого проекта, претендующего на статус успешного.

Время отражает динамический аспект жизненного цикла проекта, выраженный в терминах циклов, фаз, итераций и контрольных точек, которые разделяют две отдельные фазы. Вертикальная ось отражает статический аспект проекта. Он описывается в терминах процессов, артефактов (единица информации, создаваемая или модифицируемая в ходе любого процесса) и ролей (ответственность за адекватное выполнение той или иной деятельности в процессе). RUP организует выполнение проекта по фазам, каждая из которых состоит из одной или нескольких итераций. При итеративном подходе объем работ по каждому процессу варьируется в течение всего жизненного цикла. Контрольные точки в конце фаз позволяют оценить, насколько успешной была предыдущая фаза и насколько успешен весь проект в целом. RUP определяет следующие основные процессы:

• моделирование бизнес-процессов;

• управление требованиями;

• анализ и проектирование;

• реализация;

• тестирование;

• развертывание;

• конфигурационное управление и управление изменениями;

• управление проектом;

• управлениесредой.

Моделирование бизнес-процессов применяется с тем, чтобы разобраться в структуре исследуемой предметной области, обеспечить единство понимания основных автоматизируемых процессов среди всех участников проекта и определить высокоуровневые требования, которые должны быть реализованы в ходе проекта.

Управление требованиями позволяет прийти к соглашению с заказчиками и конечными пользователями, определить, что должна уметь делать создаваемая система, предоставить более четкие инструкции участникам проекта о возможностях системы, создать базу для успешного планирования работ в проекте и оценки его статуса в любой момент жизненного цикла.

Анализ и проектирование служат для последовательного преобразования выявленных требований к системе в спецификации особого вида, которые описывают, как следует конкретно реализовать конечный продукт. Следует при этом делать различия между анализом и проектированием. Основное из них состоит в том, что спецификации анализа не зависят от конкретной платформы и технологии, для которой осуществляется создание ИС. А спецификации проектирования являются точным представлением проектируемой системы, часто позволяя автоматизировать процесс генерации на их основе программного кода.

Реализация необходима для выявления порядка организации программного кода в терминах отдельных подсистем, преобразования исходного кода в выполняемые компоненты, тестирования созданных компонентов и интеграции отдельных компонентов в подсистемы и системы.

Тестирование позволяет определять и контролировать качество создаваемых продуктов, следить за тем, насколько качественно осуществлена интеграция компонентов и подсистем, все ли требования к системе реализованы и все ли выявленные ошибки устранены до того, как система будет развернута на оборудовании конечного пользователя.

Развертывание является процессом, в ходе которого осуществляется доставка разрабатываемого продукта к конечному пользователю. В ходе данного процесса производится новый выпуск системы, распространение ПО, его установка на стороне конечного пользователя,обучение последнего навыкам эффективной работы с поставленным ПО, предоставление услуг по технической поддержке, бета-тестирование и т. п.

Конфигурационное управление и управление изменениями позволяет организовать эффективную работу с артефактами проекта, контролировать и управлять доступом к ним, вести историю изменений, обеспечить эффективное взаимодействие участников проекта, как в простых командах, так и в распределенных, находящихся на большом удалении друг от друга.

Управление проектом включает в себя непосредственное формирование условий для эффективного хода всего проекта, определение руководств и руководящих принципов для планирования, формирования команды и мониторинга проекта, выявление и управление рисками, организацию работы участников проекта, формирование бюджета, планирование фаз и итераций.

Управление средой позволяет осуществить поддержку всех участников проекта. В эту поддержку входят выбор инструментария и его приобретение, настройка и установка, конфигурирование процесса, доработка и адаптация методологии, используемой для ведения проекта, обучение.

Важнейшие акценты RUP

Главная цель любой организации, занимающейся созданием информационных систем — работать эффективнее, а значит, быстрее создавать более качественные продукты и получать бизнес-преимущества от успешного ведения проектов. Внедрение передовой методологии, подобной RUP, позволяет гарантировать выработку и дальнейшее развитие в организации необходимых для этого навыков.

Однако внедрение методологии – не столь уж простой процесс, как это может показаться на первый взгляд. Очень важно, стремясь к более эффективному ведению проектов, не разрушить то, что уже достигнуто. Особенность методологии RUP в том, что она может быть настроена и адаптирована в соответствии с особенностями и требованиями организации-разработчика, при этом варианты внедрения RUP могут варьироваться в зависимости от конкретных условий.

Для упрощения перехода к методологии RUP допускается постепенное его внедрение. Но при этом RUP акцентирует внимание на нескольких важнейших элементах, без которых сложно гарантировать успех в проекте.

• **Общее видение проекта.** Сюда относятся первоначальный анализ будущего проекта, организация единого словаря для общения и ведение спецификации требований. Это важно потому, что участники проекта должны четко понимать цели проекта.

• **Бизнес-перспективы проекта.** Важны потому, что в основном проект выполняется для реализации каких-либо бизнес-целей. И если такие цели существуют, то имеет смысл начинать процесс разработки. Это не относится напрямую к научным и исследовательским проектам, т. к. их финансирование имеет иные корни.

• **План работ.** Позволяет определить ресурсы проекта, привязать их к задачам и рассчитать бюджет проекта. Таким образом, имеется возможность заранее спрогнозировать, насколько выгодно вести конкретный проект и какие могут быть при этом затраты.

• **Анализ рисков.** Важен потому, что обычно намного легче и дешевле выявить и устранить возможные проблемы заранее, чем делать это уже на поздних стадиях проекта.

• **Гибкая и надежная архитектура системы.** Гарантирует, что проект не потерпит крушение задолго до его завершения, что разработчики смогут развивать данную систему при изменении условий и правил ведения бизнеса на стороне заказчика.

• **Управление запросами на изменения.** Позволяет организовать эффективную работу и взаимодействие участников проекта. Возрастает контроль за качеством выполнения задания любого уровня, отслеживанием устранения ошибок и обработки предложений по дальнейшему развитию ИС.

• **Тестирование.** Дает возможность гарантировать высокое качество продукта, а, следовательно, не даст заказчику повод усомниться в возможностях организации-разработчика.

• **Акцент на самом продукте.** Крайне важен, потому что продукт — конечная цель любого проекта.

Надо помнить в любой момент, что важны не модели или многочисленные документы проекта сами по себе, а именно конечный продукт. Все остальное создается только с тем, чтобы как можно скорее создать качественный продукт.

• **Документы для поддержки пользователя.** Необходимы, т. к. без них многие сильные стороны созданного продукта могут остаться неизвестными и недоступными.

• **Измерение проекта.** В любой момент времени необходимо, чтобы вовремя реагировать на возможные отклонения проекта от бюджета и на перерасход ресурсов.

Управление процессом разработки. IBM RationalUnified Process.

В каждом проекте разработчики стремятся к общей цели — в утвержденные сроки создавать качественное и надежное ПО, отвечающее текущим требованиям и масштабируемое для будущих задач. Достичь этой цели практически невозможно без хорошо организованного процесса разработки. IBM Rational Unified Process (RUP) предоставляет группам разработчикам возможность общего, интерактивного и интегрированного управления процессом.

**RUP — обширный справочник в виде портала, который организует работу проектной команды.**

RUP содержит инструкции, шаблоны и примеры для большинства критических задач, которые могут возникнуть в ходе проекта.

**RUP — практическое руководство, помогающее разработчикам выполнять их повседневные задачи.** Реализованный с использованием HTML и Java для простого и независимого от платформы доступа с локального компьютера и через корпоративную сеть, продукт RUP имеет мощные графические средства навигации, позволяющие быстро находить необходимую информацию по разработке ПО, а также шаблоны для документов, создаваемых в ходе проекта.

**RUP позволяет интегрировать функции инструментальных средств IBM Rational:**

• руководства по инструментальным средствам описывают, как эффективно использовать эти средства для реализации конкретных задач в различных фазах разработки;

• расширенная помощь облегчает поиск инструкций для реализации текущей задачи.

**RUP может быть адаптирован как для малых, так и для больших проектов** и использоваться при разработке приложений всех типов, включая электронную коммерцию (e-business) и Web-

приложения, приложения реального времени и встроенное ПО.

Существует два разных подхода к внедрению RUP: целиком или по шагам. Наиболее эффективно внедрять весь процесс целиком, когда начинается новый проект, участники проекта хорошо владеют методологией RUP или некоторый проект ведется настолько хаотично, что любая его реорганизация заведомо принесет пользу. Предпочесть внедрение RUP по шагам стоит, если речь идет о внесении изменений в уже существующий проект, в котором есть налаженные механизмы командного взаимодействия или участники проекта плохо знакомы с положениями RUP.

В каждом конкретном случае целесообразно подобрать свои эффективные пути внедрения новых методов работы. В этом могут помочь специализированные консалтинговые компании, имеющие обширный опыт внедрения средств и методологии RUP – это поможет избежать лишних затрат на определение собственного пути.

Для адаптации методологии RUP в организации IBM Rational предлагает средства, поддерживающие задачи моделирования и описания процессов разработки. Эти средства представляют собой набор инструментов для настройки и публикации Web-сайтов на основе RUP, которые позволяют вносить изменения в RUP и описывать собственные стандарты ведения проектов:

• IBM RUP Builder - средство автоматической генерации Web-сайта модифицированного RUP, которое входит в стандартную поставку IBM Rational Suite Enterprise. При разработке сайта можно использовать готовые модули расширения (plug-in), которые могут встраиваться в текущее описание процесса, добавляя в него некоторую специфику. Готовые модули расширения можно загрузить с сайта IBM Rational по ссылке: http://www-130.ibm.com/developerworks/rational/library/5823.html

• IBM Rational XDE Modeler - визуальное средство создания взаимосвязи процессов на основе метамодели RUP, повышающее уровень абстракции и облегчающее понимание и изменение процессов (возможно также использование IBM Rational XDE Developer for Java или IBM Rational

XDE DeveloperPlus for Java)

• IBM Rational Process Engineering - средство формирования контента сайта, может быть бесплатно получен с сайта IBM Rational.

Адаптация RUP проводится инженерами-технологами организации или внешними консультантами и включает этапы определения, описания и представления процессов. В качестве основы для определения процесса используется модель RUP, затем выполняется изменение и расширение базовой модели. Визуализация связей между элементами процесса показывает, например, какие артефакты задействованы в процессе и какие роли отвечают за их создание. Библиотека элементов процесса содержит текстовую информацию о каждом элементе в модели процесса. Библиотека содержит также все текстовые страницы RUP, а также необходимые шаблоны для создания новых страниц описания. На последнем этапе — этапе представления процесса, генерируется описание процессов, включающее текст и графику в виде Web-сайта и соединяющее модели процессов и библиотеку описаний в единое целое.

Внедрение RUP может быть проведено

• во всей организации с распространением на все проекты;

• в отдельном проекте;

• в отдельных подразделениях, участвующих в одном или нескольких проектах организации;

• в подразделениях, задействованных в какой-либо части крупных проектов, ведущихся в других компаниях (аутсорсинг).

Начинать проект по адаптации RUP рекомендуется с внедрения части RUP в одном или нескольких подразделениях, а в дальнейшем распространить успешный опыт на организацию в целом. Таким образом, можно организовать на основе RUP эффективные процессы визуального моделирования и управления требованиями, создания прототипов графического интерфейса, планирования работ и определения необходимых ресурсов, автоматизированного тестирования, управления изменениями

(дефектами, запросами на улучшение, задачами и т. д.), управления версиями и конфигурацией.

### ТЕМА 3.Корпоративная шина предприятия. Назначение, основные функции.

#### 3.1. Моделирование и проектирование

Моделирование представляет собой один из ключевых процессов создания программного обеспечения, направленный на решение следующих задач:

• снижение сложности понимания предметной области;

• понимание структуры и динамики предметной области, в которой должна быть развернута система (целевой организации);

• понимание текущих проблем целевой организации и определение потенциальных возможностей усовершенствования;

• обеспечение общего понимания целевой предметной области заказчиками, конечными пользователями и разработчиками;

• выявление системных требований, необходимых для автоматизации предметной области.

Область моделирования охватывает следующие дисциплины:

• бизнес-моделирование предметной области, которая рассматривается как потенциальная для внедрения процессов автоматизации.

• функциональное моделирование системы, которая позволяет автоматизировать некоторую часть исследуемой предметной области.

• анализ и проектирование системы, которые позволяют сформировать детальное представление системы на уровне конкретных средств реализации.

Многие компании добились большей эффективности в процессе создания сложных программных систем благодаря средствам моделирования начального уровня, которые на протяжении ряда лет предлагались в составе продуктов IBM Rational и хорошо знакомы сообществу разработчиков. Имея невысокую стоимость в расчете на одного пользователя, они позволили бизнес-аналитикам эффективно описывать процессы и данные. Однако, по мере изменения и усложенения бизнес-требований, появилась потребность в использовании более развитых и совершенных средств моделирования и проектирования, которые отражали бы бизнес-ориентированный подход к процессу создания программных систем.

Для выполнения этих требований, IBM существенно расширило возможности традиционных средств моделирования, таких как Rational Rose и XDE Modeler, новыми инструментами в составе пакета **IBM Rational Software Architect (RSA).** Одним из преимуществ новых средств моделирования является возможность автоматизированного преобразования моделей, позволяющая быстро переходить от высокоуровнего моделирования к разработке и тестированию приложений. Это позволяет более эффективно использовать шаблоны проектирования (patterns), стандарты и лучшие проектные решения для создания высококачественного кода и повышения общей эффективности проектов разработки программных систем. Базовые возможности новой линейки средств графического моделирования, дизайна и проектирования реализованы в продукте IBM Rational Software Modeler. IBM Rational Software Architect включает всю функциональность Rational Software Modeler, дополняя ее возможностями автоматизирвоанного перобразования моделей и поддержкой C++ в дополнение к Java.

Кроме того, новые средства моделирования в линейке IBM исключают риски использования нестандартной технологии моделирования, жестко привязанной к технологии конкретного поставщика. Решения IBM Rational Software Architect построены на базе Eclipse - платформы с открытым исходным кодом, написанной на Java, что дает возможность быстро наращивать объем функциональных возможностей предлагаемых решений в соответствии с конкретными требованиями проекта. Eclipse поддерживает использование модулей расширения независимых разработчиков (plug-in), что также способствует созданию оптимальной среды моделирования приложений как в среде Windows, так и в среде Linux.

Специализированные средства моделирования позволяют автоматизировать повторяющиеся действия, тем самым повышая не только продуктивность, но и зрелость процесса разработки программного обеспечения в целом. Во многом этому способствует использование стандартизованного языка моделирования UnifiedModelingLanguage (UML). Однако, как показывает опыт, текущей версии UML были присущи определенные ограничения, в частности, в сфере моделирования структуры и поведения сложных систем. Новые средства моделирования IBM поддерживают UML версии 2 (UML 2), в которой предлагаются структурированные классы, усовершенствованные блок-схемы, диаграммы действий и диаграммы конечных автоматов, что позволяет гораздо более точно отображать архитектуру будущих программных систем.

Кроме того, средства моделирования в составе IBM Rational Software Architect поддерживают Model DrivenArchitecture (MDA) – новую инициативу ObjectManagementGroup (OMG) в области описания управления процессами, благодаря которой становится возможным определять несколько уровней моделей, связанных с заданными пользователем преобразованиями между моделями, для более четкого разделения аспектов жизненного цикла. Новые средства проектирования IBM поддерживают какпоследнюю версию этого языка - UML 2.0, так и более ранние реализации.

Для поддержки инструментальных средств и платформ, не вошедших в пакет IBM Rational Software Architect, используются средства моделирования из пакета IBM Rational Suite. При совместном использовании пакетов IBM Rational Professional Bundle и IBM Rational Suite возможен импорт моделей из IBM Rational Rose в RSA для реализации моделей на платформе J2EE.

Возможности инструментального средства IBM Rational XDE Modeler включены в инструментальные средства нового пакета RSA. Однако, если в организации уже наработано большое количество скриптов для IBM Rational XDE Modeler, то, из-за возможной трудоемкости переноса имеющихся скриптов на новые инструментальные средства, может оказаться целесообразным продолжение использования IBM Rational XDE Modeler наряду с новой линейкой инструментальных средств RSA.

Кроме того, линейка средств моделирования IBM включает также IBM WebSphere® Business Integration Modeler (WBI Modeler) - средство моделирования бизнес-процессов, которое входит в пакет для бизнес- интеграции IBM WebSphere. WBI Modeler наряду с новыми средствами IBM Rational является частью интегрированной платформы разработки на основе Eclipse. По этой причине мы посчитали целесообразным также включить его описание в данный раздел.

IBMRationalRose

IBM Rational Rose — популярное средство визуального моделирования, которое считается стандартом де-факто среди средств визуального проектирования приложений. Этот продукт входит в состав пакета IBM Rational Suite и предназначен для моделирования программных систем с использованием широкого круга инструментальных средств и платформ. Инструментальное средство IBM Rational

Rose расширяет возможности моделирования программных систем, выходящих за рамки платформы J2EE и инструментальных средств моделирования в составе IBM Rational Professional Bundle.

Являясь простым и мощным решением для визуальной разработки информационных систем любого класса, Rational Rose позволяет создавать, изменять и проверять корректность модели.

Rational Rose объединяет команду разработчиков на базе универсального языка моделирования UML, который определяет стандартную графическую символику для описания архитектуры ПО. Любые участники проекта — аналитики, специалисты по моделированию, разработчики и другие — могут использовать модели, построенные в Rational Rose, для большей эффективности создания конечного продукта.

Для бизнес-аналитиков средство Rational Rose дает возможность детально описать и проанализировать бизнес-процессы данной предметной области. Системные аналитики, используя указанные описания, смогут разработать необходимый функционал ИС, который максимально удовлетворит запросы заказчика. Для архитекторов средство Rational Rose будет полезно при создании мощной и гибкой архитектуры системы. Для аналитиков, специализирующихся в области разработки баз данных, Rational Rose даст возможность визуально проектировать и генерировать базы данных любого размера. Таким образом, можно создавать базы данных Microsoft SQL Server, Oracle, Sybase

SQL Anywhere, IBM DB2 и любые другие, которые поддерживают возможность запуска скриптов стандарта ANSI SQL.

Существуют расширения Rational Rose, которые позволяют выполнять скелетную (round-trip) разработку ИС, создаваемых на базе языков C/C++, Java, Smalltalk, Ada, Object Pascal (Borland Delphi) и др. Таким образом, можно сгенерировать каркас программного кода на любом из указанных языков или выполнить процедуру обратного проектирования, что позволяет сформировать модель на базе существующего кода.

Возможность публикации модели в Интернете служит основой для объединения работы удаленных команд разработчиков.

Существует вариант Rational Rose RealTime, полезный при разработке систем реального времени. Данное средство включает расширенную нотацию визуального моделирования и отвечает всем требованиям разработчиков указанного направления. Rational Rose RealTime поддерживает ви-зуальную разработку конструкций реального времени, генерацию программного кода и исполнение моделей на протяжении всего жизненного цикла.

Rational Rose предоставляет огромные возможности для проектирования ИС с использованием лучших архитектурных принципов. С помощью инструмента можно визуализировать, анализировать и уточнять требования к создаваемому продукту. Способности описывать графический интерфейс отдельно от бизнес-логики приводят в конечном итоге к лучшим результатам на уровне всего проекта. Всегда лучше и эффективнее лишний раз проанализировать, что должно быть сделано, чем каждый раз заниматься трудоемкими переделками и подгонкой кода. Использование единого инструмента моделирования на протяжении всего жизненного цикла разработки помогает создавать "идеальную" систему.

Rational Rose предлагает плавный процесс разработки ИС. Любые модели, создаваемые с помощью данного средства, являются взаимосвязанными: бизнес-модель, функциональная модель, модель анализа, модель проектирования, модель базы данных, модель компонентов и модель физического развертывания системы.

Возможности по созданию и использованию шаблонов архитектурных решений позволяют эффективно использовать опыт, накопленный в предыдущих проектах.

Rational Rose является ведущим инструментом визуального моделирования в программной индустрии, благодаря полноценной поддержке UML и многоязыковой поддержке командной разработки. Инструмент полностью поддерживает компонентно-ориентированный процесс создания ИС.

Интеграция Rational Rose с Rational RequisitePro позволяет на базе визуальной модели разработать полный набор требований, которые необходимо реализовать при создании конечного продукта.

Интеграция Rational Rose с Rational TestManager позволяет создавать сценарии тестирования на базе визуальной модели.

Интеграция Rational Rose с Rational ClearCase позволяет поставить на версионный контроль модель целиком или по частям.

Интеграция Rational Rose с Rational SoDA позволяет автоматизировать процесс создания документов и отчетов по визуальной модели.

IBM Rational XDE Modeler

IBM Rational XDE Modeler - средство визуального моделирования информационных систем на основе UML v1.4, которое входит в пакет IBM

Rational Suite.

Возможности IBM Rational XDE Modeler

покрываются средствами нового пакета IBM

Rational Software Architect. Однако, если в организации уже наработано большое количество скриптов для IBM Rational XDE Modeler, то, из-за возможной трудоемкости переноса имеющихся скриптов на новые инструментальные средства, может оказаться целесообразным продолжать использования IBM Rational XDE Modeler наряду с новой линейкой инструментальных средств RSA.

IBM Rational XDE Modeler позволяет архитекторам и проектировщикам организовать управляемую моделями разработку с использованием UML.

Пользователи могут создавать независимые от платформ модели программной архитектуры, бизнес-требований, повторно используемых информационных ресурсов и коммуникационных средств управленческого уровня. Поддержка UML, ставшего промышленным стандартом, а также мощный механизм работы с шаблонами позволяет пользователям создавать семантически насыщенную прикладную архитектуру, соответст-вующую бизнес-требованиям и легко понимаемую коллективом разработчиков.

Архитекторы и проектировщики могут использо-вать поддержку множественных моделей в Rational XDE Modeler для реализации парадигмы управляемой моделями архитектуры (MDA).

Разработчики могут использовать архитектурные модели и шаблоны как основу для реализации, ускорив, таким образом, разработку приложений в строгом соответствии с предоставленной архитектурой. Кроме того, такие возможности, как моделирование в свободной форме, генерация отчетов и публикация в Web-среде позволяют организовать совместную работу пользователей и заинтересованных лиц над архитектурными и проектными решениями, независимо от того используют ли заинтересованные лица Rational XDE.

Rational XDE Modeler включает в себя расширенные возможности работы с шаблонами, что позволяет программным архитекторам с помощью автоматических трансформаций аналитических моделей контролировать проектные решения и их реализацию. Подобная автоматизация также позволяет повысить эффективность архитектуры за счет ее непротиворечивости и воспроизводимости.

Архитекторы начинают свою работу с использования шаблона модели, который предоставляет структуру для их проекта. Они могут воспользоваться либо стандартными шаблонами, либо быстро и просто создать свои собственные шаблоны. Встроенный в Rational XDE Modeler механизм работы с шаблонами позволяет создавать шаблоны на основе уже существующих моделей, с возможностью произвольных дальнейших модификаций шаблона под текущую задачу.

Автоматизированные шаблоны позволяют архитекторам быстро начать разработку, полностью использовать имеющиеся творческие способности, избежать выполнения повторяющихся рутинных задач, а также проектировать свои собственные шаблоны.

Используя Rational XDE Modeler, архитекторы получают исчерпывающее представление о влиянии изменений на проектные решения. При этом отслеживаются не только уже сделанные изменения и история развития проектных решений, но также и влияния предстоящих изменений.

Кроме того, архитекторы могут быстро оценить "глобальное" влияние предложенных архитектурных улучшений на будущие версии и редакции ПО. Без использования Rational XDE Modeler рассмотрение столь широкой области воздействия было бы затруднительно.

IBM Rational XDE Modeler позволяет пользователям создавать модели и диаграммы на любом уровне абстракции, начиная от диаграмм в свободной форме для управленческого звена и заканчивая подробными проектными моделями, предназначенными для преобразования в программный код.

Rational XDE Modeler интегрируется с Rational RequisitePro, что позволяет просматривать и управлять постоянно изменяющимися требованиями. Также предоставляются возможности публикации в среде

web, наряду с другими возможностями генерации отчетов, что позволяет обмениваться информацией с другими сотрудниками, независимо от их участия в проекте.

Rational XDE Modeler интегрируется и с IBM Rational ClearCase, что позволяет контролировать версии приложений непосредственно из IDE-среды.

RUP предоставляет конфигурируемую инфраструктуру процессов создания ПО, которая включает специальные конфигурации RUP для разработчиков на Java и .NET, а также специальный модуль для реализации техники экстремального программирования (XP).

IBM WebSphere Business Integration Modeler (WBI Modeler)

IBM WebSphere® Business Integration Modeler (WBI Modeler) представляет собой средство моделирования бизнес-процессов и включает средства для проектирования, тестирования и совершенствования комплексных бизнес-процессов, а также средства для организации коммуникации между ними. WBI Modeler используется для создания карты существующих бизнес-процессов и их дальнейшего моделирования и оптимизации в целях управления компанией как единым целым.

Решения WBI Modeler построены на базе платформы Eclipse и стали первыми реализациями нового поколения средств моделирования WebSphere, интегрированными с другими инструментами Eclipse, такими как, IBMRationalApplicationDeveloper, IBMRational, IBMWebSphereMQWorkflow (ранее известный как IBM MQSeries Workflow) и программным обеспечением IBM Tivoli.

WBI Modeler имеет в своем составе следующие средства моделирования бизнес-процессов, обеспечивающие высокий уровень их гибкости:

• WBI Modeler Entry Edition – средство моделирования начального уровня, настраиваемое в соответствии с потребностями индивидуальных пользователей и позволяющее растущим компаниям решить задачи формализации бизнес процессов при более низких затратах в расчете на одного пользователя

• WBI Modeler Advanced Edition включает расширенные средства моделирования для предприятий с протяженными бизнес-процессами, а также инструменты для более сложного моделирования и оптимизации, которые помогают анализировать эффективность выполнения бизнес-процессов и потенциальные бизнес-результаты еще до реального выделения ресурсов на разработку и развертывание

Оба варианта WBI Modeler - Advanced и Entry - поддерживают три разных профиля пользователя, а расширенная версия и три операционных режима.

**Выбор профиля пользователя.** В зависимости от потребностей, пользователи имеют возможность выбрать базовый, промежуточный или продвинутый профиль для моделирования бизнеса, каждый из которых обеспечивает свой уровень детализации и сложности информации, используемой при моделировании процессов. Пользователи могут легко переключаться между этими профилями, используя преимущества тех или иных средств.

• Базовый профиль моделирования бизнеса (Basic) является входной точкой для таких бизнес-пользователей, как бизнес-аналитики, проектировщики и эксперты и позволяет им легко и быстро фиксировать и моделировать последовательности операций, сложную логику и бизнес-правила.

• Промежуточный профиль моделирования бизнеса (Intermediate) позволяет пользователям с более глубокими техническими знаниями - разработчикам архитектуры бизнеса и бизнес-процессов - определять и просматривать дополнительные детали моделей процессов и данных, включая выражения и связи.

• Расширенный профиль моделирования бизнеса (Advanced) обеспечивает наиболее подробный уровень детализации для моделей процессов и данных и предназначен для разработчиков решений и ИТ-архитектуры, которые сконцентрированы на развертывании моделей в среде выполнения.

**Выбор технологического режима.** WBI Modeler позволяет пользователям выбрать один из трех разных технологических режимов:

• Операционный – доступен в начальной и расширенной версиях, является заданным по умолчанию режимом работы и обеспечивает наибольший уровень детализации среди трех технологических режимов.

• Business Process Execution Language (BPEL) – оптимизировандлямоделейэкспортавфайловыеформатыBPEL, WebServicesDescriptionLanguage (WSDL) иXMLSchemaDefinitionLanguage

(XSD), атакжедляимпортаизIBMWebSphereStudioApplicationDeveloperIntegrationEdition, обеспечиваяразработкуприложенийнаосновемоделей.

• MQ Workflow Flow Definition Language (FDL) – оптимизирован для моделей экспорта в формат FDL, позволяет компаниям быстро начать работу, используя IBM WebSphere MQ Workflow в качестве решения для создания автоматизированного потока операций.

Оба решения – WBI Modeler Advanced Edition и WBI Modeler Entry Edition – предоставляют репозитарий для централизованного хранения правил, процедур, графиков и другой важной корпоративной информации, что обеспечивает ее непротиворечивость и доступность в рамках всей компании.

Объединение этой информации с бизнес-процессами крайне важно для точного представления всех факторов, влияющих на результат каждого из процессов. В результате становится возможным идентифицировать важные корпоративные данные и правила, которые определяют эффективность процессов, что, в свою очередь, помогает оценить степень готовности к автоматизации.

WBIModelerAdvancedEdition и WBIModelerEntryEdition также имеют в своем составе инструменты для динамического анализа, которые помогают оптимизировать моделирование бизнес-процессов.

**Возможности компоновки для просмотра моделей.** Решения WBI Modeler Advanced Edition и WBI

Modeler Entry Edition позволяют просматривать модели либо в свободной компоновке, либо в виде секций (swimlane). Свободная компоновка предполагает использование всего диапазона средств редактирования, тогда как альтернативный вариант компоновки отображает присвоение операций процесса конкретным ролям, ресурсам или организационным единицам.

После ввода моделей к ним применяются правила проверки достоверности, которые позволяют провести простое тестирование документации и зафиксировать этапы процесса. С помощью WBI

Modeler Advanced Edition можно также провести более глубокое тестирование с целью прогнозирования общей продолжительности работ и предполагаемых затрат, ожидаемого отставания и т.д. Эти возможности появляются при моделировании сценария с определенной моделью.

Сравнивая ожидаемые расходы на существующие процессы с затратами для оптимизированных процессов, легко подготовить анализ окупаемости инвестиций (ROI).

**Экспорт и импорт данных.** WBI Modeler Advanced Edition позволяет быстро начать развертывание оптимизированных бизнес-процессов благодаря возможностям экспорта их определений в IBM

WebSphereMQWorkflow, V3.5, IBMWebSphereBusinessIntegrationServer (для обоих продуктов при помощи экспорта FDL) и WebSphereBusinessIntegrationServerFoundation (при помощи экспорта BPEL). Импорт FDL и существующих моделей WebSphereBusinessIntegrationWorkbench, V4.2.4 также позволяет многократно использовать существующие бизнес-данные, обеспечивая сокращение издержек. Для случаев, когда требуется усовершенствование бизнес-модели для создания приложений, WBI Modeler Advanced Edition позволяет экспортировать язык Unified Modeling Language

(UML) впрограммноеобеспечение IBM Rational® XDE™ и Rational Software Architect.Кроме того, WebSphereBusinessIntegrationModelerAdvancedEdition поддерживает моделирование в командной среде с использованием интегрированной среды разработки (IDE) Eclipse для моделей процессов с механизмом выписки и возврата элементов.

IBM WebSphere® Business Integration Modeler входит в состав программного пакета для бизнес-интеграции IBM WebSphere, который представляет собой современное связующее ПО, объединяющее приложения, пользователей и бизнес-процессы в единый механизм.

#### 3.2. Средства разработки

Основные цели разработки ПО:

• определение структуры кода на основе реализуемых подсистем, организованных по уровням;

• реализация классов и объектов в виде модулей (исходных, двоичных, исполняемых файлов и др.);

• тестирование разработанных модулей;

• интеграция результатов работы отдельных программистов (или групп) в рабочую систему.

Для ведения непосредственной разработки IBM Rational предлагает целый набор инструментария, описанного далее.

IBMRationalApplicationDeveloper

IBMRationalApplicationDeveloperforWebSphereSoftware – это средство быстрого проектирования, разработки, анализа, тестирования, профилирования и развертывания Web-приложений, Web-служб, Java, J2EE и порталов в среде IDE на базе Eclipse.

В условиях ведения бизнеса “по требованию” разработчики должны быстро реагировать на непрерывно изменяющиеся запросы. Для этой цели им необходима среда разработки, которая повышала бы производительность труда, сводила бы к минимуму длительность обучения и сокращала бы время разработки и тестирования. IBMRationalApplicationDeveloperforWebSphereSoftware – это всеобъемлющая интегрированная среда разработки, обеспечивающая полную поддержку модели программирования J2EE, включая Web, Java, Web-службы и EJB, которая позволяет значительно ускорить разработку приложений. Rational Application Developer включает в себя интегрированные средства разработки порталов, визуального редактирования UML, анализа кода, автоматизированного тестирования и развертывания – все, что необходимо разработчикам для производительной работы и для того, чтобы полученный код был хорошо спроектирован, масштабируем и готов к промышленной эксплуатации. Встроенные инструменты контроля версий и поддержки групповой работы позволяют разработчикам сложных проектов и большим командам координировать контроль версий и обеспечить защиту ресурсов группы.

IBM Rational Application Developer for WebSphere Software оптимизирован для программного обеспечения WebSphere и предоставляет возможности разработки для других технологических платформ.

**Полная поддержка модели программирования J2EE.** Параллельная поддержка проектов J2EEверсий 1.2 , 1.3 и 1.4, включая полную поддержку следующих средств: Enterprise JavaBeans™, EJB

Query Language (EJB QL), Container-Managed Persistence (CMP), Web-архивы (WAR) икорпоративныеархивы (EAR). Включает в себя среды тестирования модулей для нескольких конфигураций IBM

WebSphere Application Server, чтобы обеспечить поддержку проектов с различными тестовыми средами.

**Расширение возможностей программирования Java.** Application Developer включает в себя большой набор инструментов для создания приложений Java и поддерживает Java Development Kit

версии 1.4.2. Visual Editor for Java позволяет разрабатывать GUI для Java методом перетаскивания и вставки, с помощью компонентов AbstractWindow Toolkit (AWT), SWT или Swing, а также позволяет визуально привязывать данные к виджетам UI и немедленно просматривать изменения вида GUIблагодаря динамическому механизму обновления для исходного кода и визуального дизайна. Unified

Modeling Language (UML) Visual Editor содержит графические редакторы в качестве альтернативного способа визуализации и редактирования кода Java и таблиц данных при помощи стандартных диаграмм UML, а также помогает вам лучше понимать сложный код и управлять им.

**Интеграция приложений с помощью web-служб и архитектур, ориентированных на службы.**

Rational Application Developer содержит инструменты, необходимые для обнаружения, создания, сборки, тестирования, развертывания и публикации web-служб. Вы можете создавать новые web-

службы с нуля или добавлять в существующие приложения совместимые с WS-I web-службы, чтобы упростить интеграцию и повторное использование:

• автоматическая генерация файлов WSDL и WSIL, необходимых для описания web-служб и подготовки их к инспекции на основе имеющихся материалов, либо разработка файла WSDL и генерация кода Java;

• простота создания, проверки и обнаружения совместимых web-служб, соответствующих WS-I, с помощью редактора WSDL и браузера web-служб;

• создание Java-прокси для взаимодействия с клиентскими приложениями;

• генерация тестовых клиентов web-служб без кодирования;

• занесение в реестр UDDI версии 2

**Упрощение разработки XML.** Полный набор функций XML в составе Application Developer позволяет разработчикам создавать, редактировать и преобразовывать документы XML:

• визуальные инструменты для создания и просмотра файлов XML и для автоматической проверки документов на соответствие определениям типа документа (DTD);

• визуальные инструменты для преобразования элементов XML в таблицы реляционной базы данных;

• тесная интеграция с IBM DB2 для создания документов XML из существующих данных DB2 или разбора документов XML в данные DB2;

• инструменты для экономии времени при сборке и отладке сценариев преобразования Extensible

Stylesheet Language (XSL) и для преобразования документов XML в другие форматы разметки, в том числе HTML;

**Создание приложений на основе баз данных.** Rational Application Developer содержит интегрированные инструменты для создания приложений баз данных из запросов, элементов Beans

и EJB.

Мастер соединений с базой данных упрощает создание соединения Java Database Connectivity

(JDBC ) с СУБД, такими как IBM DB2 Universal Database, IBM Informix®, Oracle или Microsoft® SQL

Server®.

Мастер запросов SQL и построитель запросов SQL предоставляют визуальный интерфейс для создания и выполнения операторов SQL. Можно создать простой запрос с помощью мастера запросов SQL или использовать построитель запросов SQL, который поддерживает более широкий набор операторов. Встроенный мастер преобразования SQL-XML позволяет создавать документы XML и XSL, спецификации DTD, схемы XSD, файлы HTML и связанные артефакты.

Поддержка SQL for Java (SQLJ) позволяет быстро создавать и отлаживать приложения, использующие хранимые процедуры SQLJ и DB2 SQLJ, а также повышает производительность доступа к данным для статических соединений SQL.

**Упрощение объектно-реляционного соответствия.** Обеспечение объектно-реляционного соответствия дается легко и просто благодаря поддержке методов сверху вниз, снизу вверх и от середины. Мастера позволяют легко создавать и тестировать компоненты EJB. Вы можете строить приложения, которые используют, извлекают и отображают нужные данные, отформатированные в соответствии с вашими потребностями. Rational Application Developer поддерживает отношения

«многие ко многим», что помогает генерировать реализации со множественными cвязями.

**Упрощение web-разработки и управления web-сайтами.** Простые в работе мастера и инструменты объединяют практически все аспекты web-разработки (HTML, JSP, JSF и сервлеты) в единый интерфейс и позволяют разработчикам с различным характером технической подготовки и даже тем, кто не знаком с Java, строить многофункциональные приложения для работы с данными:

• визуальное проектирование и конструирование web-приложений с помощью Model-View-

Controller и Struts 1.1;

• быстрая разработка многофункциональных пользовательских web-интерфейсов и web-форм с помощью многократно используемых, перетаскиваемых компонентов JavaServer Faces, которые генерируют код обработки событий, проверки пользовательского ввода и привязки данных для

web-приложений;

• возможность подключения web-приложений к реляционным базам данных, компонентам EJB и

web-службам с помощью простых инструментов, поддерживающих Service Data Objects, новый промышленный стандарт для доступа к гетерогенным данным;

• создание интерактивных web-интерфейсов пользователя с производительностью и характеристиками техобслуживания на уровне тонких клиентов с помощью клиентских компонентов Faces, расширяющих спецификацию JSF;

• быстрая разработка визуальных отчетов по данным с помощью встроенных инструментов Crystal

Reports. Проектирование и разработка собственных отчетов или использование готовых шаблонов отчетов Компонент JSF Report Viewing позволяет внедрять отчеты в web-приложения и динамически исполнять их в Интернете;

• простая разработка многофункциональных web-страниц в режиме визуального редактирования или редактирования исходного кода с помощью богатого возможностями редактора HTML и JSP;

• возможность быстро импортировать web-сайт целиком, изменять стили и применять их в глобальном масштабе, визуально добавлять и удалять страницы из древовидного структурного представления с помощью Web Site Designer;

• простое создание и анимация оригинальной графики с помощью компонентов WebArt Designer и Animated GIF Designer.1.

**Кодирование на 4GL и генерация кода на Java.** Enterprise Generation Language (EGL) – это язык четвертого поколения, который позволяет разработчикам процедур быстро создавать и отлаживать управляемые данными web-приложения и бизнес-логику с помощью знакомых программных конструкций и без написания кода на Java. EGL содержит конструкции и функциональные возможности Informix 4GL, благодаря чему Application Developer является идеальной средой разработки для Informix. EGL тесно интегрирован с JavaServer Faces, так что вы можете создавать интерактивные, управляемые данными web-приложения с помощью компонентов JSF.

**Визуальная разработка портлетов и портальных приложений.** Application Developer содержит набор визуальных инструментов разработки порталов и среду тестирования WebSphere Portal, что позволяет разрабатывать и тестировать отдельные портлеты и портальныне приложения в целом. Новые портлеты создаются при помощи мастеров, которые генерируют структуру проекта портлета, совместимую с J2EE, и могут создать весь портлет. Мастера могут генерировать портлеты, соответствующие IBM

Portlet API или же JSR 168, спецификации промышленного стандарта для агрегации, персонализации, представления и обеспечения безопасности портлетов. Можно спроектировать интерфейс портлета с помощью Page Designer и JavaServer Faces, а затем визуализировать структуру и потоки событий с помощью Web Diagram Editor и схемы Struts, чтобы облегчить поддержку приложений. Кроме того, можно сочетать портлеты с Java Server Faces, чтобы разрабатывать портлеты визуально с помощью компонентов Faces. Portal Designer позволяет визуально создавать и редактирования приложения порталов, а также визуально редактировать их темы и внешний вид.

**Автоматизация сборки приложений.** При работе с инструментами предыдущих поколений разработчикам приходилось вручную компоновать архивы Java (JAR) и WAR в файл EAR для развертывания. Application Developer автоматизирует сборку EAR и генерирует дескрипторы развертывания для новых приложений, экономя ваше время.

**Повышение качества кода.** Чем раньше замечены ошибки, тем проще их исправить, поэтому в состав Application Developer входят инструменты, которые помогают разработчикам повысить качество кода.

Во-первых, функция автоматизированного анализа кода применяет более 200 стандартов и практических рекомендаций кодирования J2EE к коду приложения, помечая нарушения и подсказывая варианты устранения ошибок – включая вносимые одним щелчком быстрые исправления. В числе имеющихся категорий правил – практические рекомендации программирования на J2SE/J2EE, глобализации и доступности; интерактивный редактор правил позволяет определять новые правила.

Во-вторых, тестирование компонентов Java и web-служб автоматизирует создание тестовых заглушек, наборов и входных данных на основе анализа кода или файла WSDL. При создании и выполнении тестов используются приоритеты, основанные на различных указанных метриках; динамически генерируемые входные и выходные данные можно преобразовать в гибком редакторе данных, который поддерживает создание сложных объектов и вставки пользовательского кода. Поддерживается как локальное, так и удаленное тестирование.

**Ускорение отладки.** Для ускорения процесса отладки в Application Developer имеется множество инструментов анализа времени выполнения для приложений J2EE, выполняемых как на локальной, так и на удаленной машине. Визуализированный поток выполнения отображается на интерактивной блочной диаграмме на базе UML, которая коррелирует с информацией о потоках, с тем чтобы обнаруживать и диагностировать проблемные состояния. Графики вызовов для графического анализа производительности помогают выявлять узкие места на уровне методов, а отчеты с цветовым кодированием выявляют и отображают невыполненные фрагменты кода на уровне строк. Мощная утилита обнаружения утечки памяти использует технологию, разработанную IBM Research, чтобы сузить множество из десятков тысяч кандидатов на утечку до нескольких подозрительных участков.

Кроме того, в выполняемый код можно вставлять датчики для пользовательского анализа, которые позволяют выполнять определенный пользователем код при выполнении операций entry, exit и try/catch

над указанными классами и методами.

**Оптимизация тестирования модулей и систем.** Интегрированные среды тестирования модулей поддерживают тестирование и отладку локального и серверного кода на IBM WebSphere Application

Server, WebSphere Application Server Express, WebSphere Portal и Apache Tomcat.Вы можете создавать и настраивать экземпляры серверов с помощью мастеров, выполнять пошаговую трассировку приложений, задавать точки останова и даже модифицировать код во время отладки, не перезапуская сервер тестирования модуля. Можно создавать, тестировать и развертывать приложения J2EE на BEA

WebLogic Server, версии 6.1, 7.0 и 8.1, с помощью IBM Rational Deployment Toolkit for WebLogic Server.

**Повышение производительности труда группы благодаря эффективному управлению изменениями.**

Application Developer тесноинтегрировансрешениями IBM Rational ClearCase и IBM Rational ClearQuest

для управления конфигурацией. Такая глубокая интеграция позволяет выполнять множество операций по управлению изменениями прямо в Application Developer, а также дает интегрированное представление проектов, которое помогает повысить как качество сотрудничества, так и производительность труда группы. Application Developer включает в себя IBM Rational ClearCase LT, который обеспечивает надежный контроль версий на уровне записей и содержит готовые средства поддержки групповой работы. Также продукт поддерживает систему Concurrent Versions System и системы управления конфигурацией от независимых поставщиков через специализированные модули расширения (plug-in).

**Визуализация и графическое редактирование кода.** Редактор UML (Unified Modeling Language) Visual

Editor позволяет графически визуализировать и редактировать код и объекты данных J2EE с помощью стандартной нотации UML, благодаря чему разработчики, даже малознакомые с UML, могут лучше понимать сложный код и управлять им. UML Visual Editor также содержит мощные графические редакторы, с помощью которых вы можете добавлять диаграммы классов UML непосредственно в проект Java или EJB, который автоматически создаст соответствующий код. Затем этот код можно модифицировать – как напрямую, так и с помощью диаграмм классов UML. А поскольку визуализация выполняется динамически, она всегда синхронизирована с нижележащим кодом, и этот код можно переделывать, перемещая объекты в диаграмме класса.

Для создания нового кода Application Developer позволяет добавлять диаграммы классов UMLнепосредственно в проект Java или EJB, автоматически генерировать соответствующий код и редактировать его либо непосредственно в диаграммах классов UML, либо в проекте Java или EJB. Спомощью редактора диаграмм классов можно визуализировать объекты данных, создавать новые

«зерна», создавать и модифицировать связи между «зернами» или исследовать имеющиеся связи между

«зернами», классами и интерфейсами. Диаграммы помогают вам обнаружить и выделить связи между фрагментами кода и объектами данных, не столь очевидные в других редакторах, что поможет вам в беседах с архитекторами ПО и другими участниками команды разработчиков.

Возможности Rational Application Developer построены на платформе Eclipse с открытым исходным кодом, благодаря чему вы можете адаптировать и расширять свою среду разработки приложений с помощью модулей расширения Eclipse (plug-in), созданными IBM, бизнес-партнерами IBM и сообществом Eclipse.

Априиспользовании Rational Application Developer всочетаниисплатформой IBM Software Development

Platform вы можете получить доступ к широкому спектру требований и функций управления изменениями непосредственно из Rational Application Developer и оценить высокий уровень интеграции на протяжении всего цикла разработки.

IBMRationalWebDeveloper

IBMRationalWebDeveloper – это инструмент для быстрой разработки, тестирования и развертывания приложений Web, web-служб и приложений Java в среде IDE, простой в обучении и работе. IBM Rational Web

Developer полностью соответсвует потребностям разра-ботчиков, которым необходимы инструменты, максимально облегчающие работу со сложными задачами и интуитивно понятные в работе.

Web Developer представляет собой интегриро-ванную среду разработки, в которой можно быстро и легко разрабатывать web-приложения, web-

службы и приложения Java. Благодаря широкому спектру простых в изучении и работе визуальных инструментов для разработки, тестирования и разверты-вания Web Developer является превосходным выбором для разработчиков ПО, не имеющих большого опыта работы с Java или не нуждающихся в создании собственных компонентов EJB.

Web Developer основан на платформе Eclipse с открытым исходным кодом, благодаря чему вы можете адаптировать и расширять свою среду разработки приложений в соответствии с вашими конкретными потребностями. При использовании продукта в сочетании с платформой IBM Software

Development Platform вы можете получить доступ к широкому спектру требований и функций управления изменениями непосредственно из Web Developer.

IBM Rational Web Developer for WebSphere Software оптимизирован для программного обеспечения WebSphere и содержит возможности разработки для других технологических платформ.

Основныепреимущества IBM Rational Web Developer

**Упрощение web-разработки и управления web-сайтами.** Мастера и инструменты быстрой разработки приложений (RAD) объединяют практически все аспекты web-разработки (HTML, JSP, JSFи сервлеты) в единый интерфейс и позволяют разработчикам с различным уровнем подготовки и даже тем, кто не знаком с Java, быстро строить многофункциональные приложения для работы с данными.

• Визуальное проектирование и конструирование web-приложений с помощью Model-View-

Controller и схем Struts 1.1 и/или JavaServer Faces

• Быстрая разработка многофункциональных пользовательских web-интерфейсов и web-форм с помощью многократно используемых, перетаскиваемых компонентов JavaServer Faces, которые генерируют код обработки событий, проверки пользовательского ввода и привязки данных для

web-приложений

• Возможность подключения web-приложений к реляционным базам данных, компонентам EJB и

web-службам с помощью простых инструментов, поддерживающих Service Data Objects, новый промышленный стандарт для доступа к гетерогенным данным

• Создание интерактивных web-интерфейсов пользователя с производительностью и характеристиками техобслуживания на уровне тонких клиентов с помощью клиентских компонентов Faces, расширяющих спецификацию JSF

• Простая разработка многофункциональных web-страниц в режиме визуального редактирования или редактирования исходного кода с помощью богатого возможностями редактора HTML и JSP

• Возможность быстро импортировать web-сайт целиком, изменять стили и применять их в глобальном масштабе, визуально добавлять и удалять страницы из древовидного структурного представления с помощью Web Site Designer

• Простое создание и анимация оригинальной графики с помощью компонентов WebArt Designer и Animated GIF Designer1

**Кодирование на 4GL и генерация кода на Java.** Enterprise Generation Language (EGL) – это язык четвертого поколения, который позволяет разработчикам процедур быстро создавать и отлаживать управляемые данными web-приложения и бизнес-логику с помощью знакомых программных конструкций и без написания кода на Java. EGL содержит конструкции и функциональные возможности Informix 4GL, благодаря чему Web Developer является идеальной средой разработки для Informix. EGL тесно интегрирован с JavaServer Faces, что дает возможность создавать высокоинтерактивные, управляемые данными web-приложения с помощью компонентов JSF.

**Интеграция приложений с помощью web-служб и архитектур, ориентированных на службы.** Web

Developer содержит инструменты, необходимые для обнаружения, создания, сборки, тестирования, развертывания и публикации web-служб, совместимых с WS-I. Вы можете создавать новые web-

службы с нуля или добавлять web-службы в существующие приложения, чтобы упростить интеграцию и повторное использование.

• Автоматическая генерация файлов WSDL и WSIL, необходимых для описания web-служб и подготовки их к инспекции на основе имеющихся материалов, либо разработка файла WSDL и генерация кода Java

• Простота создания, проверки и обнаружения совместимых web-служб, соответствующих WS-I, с помощью редактора WSDL и браузера web-служб

• Кодирование входных и выходных сообщений с помощью SOAP

• Создание Java-прокси для взаимодействия с клиентскими приложениями

• Генерация тестовых клиентов web-служб без кодирования

• Занесение в реестр UDDI версии 2

**Упрощение разработки XML.** Полный набор функций XML в составе Web Developer позволяет разработчикам создавать, редактировать и преобразовывать документы XML.

• Визуальные инструменты для создания и просмотра файлов XML и для автоматической проверки документов на соответствие определениям типа документа (DTD)

• Визуальные инструменты для преобразования элементов XML в таблицы реляционной базы данных

• Тесная интеграция с IBM DB2 для создания документов XML из существующих данных DB2 или разбора документов XML в данные DB2

• Инструменты для экономии времени при сборке и отладке сценариев преобразования Extensible

Stylesheet Language (XSL) и для преобразования документов XML в другие форматы разметки, в том числе HTML

**Создание приложений на основе баз данных.** Web Developer содержит интегрированные инструменты для создания приожений баз данных из запросов и элементов Beans («зерен»).

Мастер соединений с базой данных упрощает создание соединения Java

DatabaseConnectivity (JDBC )сСУБД, такимикакIBMDB2 UniversalDatabase, IBMInformix, OracleилиMicrosoftSQL

Server.

Мастер запросов SQL и построитель запросов SQL предоставляют визуальный интерфейс для создания и выполнения операторов SQL. Можно создать простой запрос с помощью мастера запросов SQL или использо-вать построитель запросов SQL, который поддерживает более широкий набор операторов. Встроенный мастер преобразования SQL-XML позволяет создавать документы XML и XSL, спецификации DTD, схемы XSD, файлы HTML и связанные артефакты. Поддержка SQL for Java (SQLJ) позволяет быстро создавать и отлаживать приложения, использующие хранимые процедуры SQLJ и DB2 SQLJ.

**Быстрое создание приложений Java с помощью визуальных инструментов.** Web Developer

включает в себя большой набор инструментов для создания приложений Java и поддерживает Java

Development Kit версии 1.4.2. Visual Editor for Java позволяет разрабатывать GUI для Java методом перетаскивания и вставки, с помощью компонентов Abstract Window Toolkit (AWT), SWT или Swing, а также позволяет визуально привязывать данные к виджетам (widget)

UI и немедленно просматривать изменения вида GUI благодаря динамическому механизму обновления для исходного кода и визуального дизайна.

**Оптимизация тестирования приложений.** Интегрированные среды тестирования модулей и визуальный отладчик значительно повышают производительность труда разработчиков и снижают общие затраты на разработку, так как можно реализовать полный тестовый сервер на локальной машине. Web Developer

предоставляет среду для быстрого тестирования и отладки локального и серверного кода на WebSphere

Application Server, WebSphere Portal и Apache Tomcat.

• Встроенные мастера помогают создавать и настраивать экземпляры серверов; кроме того, Web

Developer может создавать их автоматически.

• Возможно выполнять пошаговую трассировку приложений, задавать точки останова и даже модифицировать код на лету и возобновлять отладку, не перезапуская приложение.

• Тестирование JavaScript и Active Script на клиентской стороне с помощью Active Script Debugger

• Можно создавать, тестировать и развертывать приложения на BEA WebLogic Server, версии 6.1,

7.0 и 8.1, спомощью IBM Rational Deployment Toolkit for WebLogic Server.

**Повышение производительности разработки благодаря эффективному управлению изменениями.** Для быстрого и экономичного создания конкурентоспособных web-приложений необходимы скоординированные усилия всей группы разработчиков, включая дизайнеров графики, программистов и бизнес-аналитиков. Web Developer тесно интегрирован с решениями IBM Rational

ClearCase и IBM Rational ClearQuest для управления конфигурацией. Такая глубокая интеграция позволяет выполнять множество операций по управлению изменениями прямо в Web Developer, а также дает интегрированное представление проектов, которое помогает повысить как качество сотрудничества, так и производительность труда группы. Web Developer также поддерживает систему Concurrent Versions System и системы управления конфигурацией от независимых поставщиков через специализированные модули расширения (plug-in).

IBM Rational Web Developer входит в состав пакета IBM Rational Software Architect и существует как отдельный продукт.

IBMRationalXDEDeveloper

Продукты RationalXDEDeveloper и RationalXDEDeveloperPlus являются дальнейшим развитием возможностей, заложенных в Rational XDE Modeler. Эти инструменты позволяют выполнять разработку информационных систем с использованием Round-Trip подхода, предполагающего автоматизированную генерацию каркаса кода на базе визуальной модели и обратное проектирование

– построение модели на основе существующего кода.

Rational XDE DeveloperPlus дополнительновключаетпродукты Rational Purify, Quantify и PureCoverage.

Также в данный инструмент входит Rational Visual Trace, который позволяет создать диаграмму последовательности на основе выполнения приложения. Этот продукт позволяет отображать созданные объекты приложения и последовательность вызовов их методов.

Средства Rational XDE Developer и Rational XDE DeveloperPlus поставляются для конкретных платформ разработки приложений. Таким образом, существуют Rational XDE Developer для платформы Java

(RationalXDEDeveloperPlusforJava) и RationalXDEDeveloper для платформы .NET (RationalXDE

DeveloperPlus for .NET).

TestFactory

Rational TestFactory — инструмент для автоматической генерации полного набора тестов для разрабатываемой информационной системы.

Оценить качество ПО можно по трем различным критериям:

• надежность — работает ли приложение без сбоев, зависания или вызова исключений;

• функциональность — делает ли приложение то, что от него требуется;

• производительность — работает ли приложение с приемлемой скоростью при доступе к нему многих пользователей.

Первым и основным критерием в этом списке является надежность. Нельзя успешно протестировать функциональность или производительность приложения, которое регулярно "зависает", и разумеется, оно не может быть поставлено конечным пользователям.

Rational TestFactory шаг за шагом проводит через процесс устранения дефектов и повышения надежности. Тестирование надежности включает в себя проверку всех вариантов исполнения приложения. Поскольку в приложении может потенциально существовать огромное число путей выполнения, то проблема состоит в том, чтобы создать тесты, которые проверяют все части приложения за минимальное количество шагов. Новаторство TestFactory заключается в автоматическом создании таких "наилучших сценариев". TestFactory строит карту приложения, а затем Rational® Robot

автоматически генерирует тестовые скрипты. После этого TestFactory использует метрики покрытия кода для определения того, какая часть системы была протестирована.

Поскольку с помощью TestFactory тестовые скрипты генерируются автоматически, то тестировщики могут протестировать каждую сборку, проверяя ее на надежность, независимо от стадии разработки.

Rational TestFactory является наиболее эффективным инструментом для устранения дефектов и повышения надежности, но в полной мере его достоинство становится очевидным при использовании как части всеобъемлющего решения по контролю за качеством ПО.

#### 3.3. Управление требованиями. Rational RequisitePro

Успешный опыт разработки показывает, что эффективное управление требованиями является ключевым фактором всего процесса разработки ПО. Требования определяют то, что должна делать система. Поэтому в течение всего жизненного цикла проекта нужно организовать эффективную работу с ними. Первым шагом в этом направлении служит организация хранения всех выявленных требований.

Основные цели процесса управления требованиями:

• понять структуру и динамику предметной области, в которой должна быть развернута создаваемая информационная система;

• понять текущие проблемы предметной области и определить потенциальные возможности ее усовершенствования;

• обеспечить общее понимание предметной области заказчиками, конечными пользователями и разработчиками;

• выявить системные требования, необходимые для поддержки автоматизации предметной области;

• установить и поддержать соглашение с клиентами и другими заинтересованными лицами на том, что система должна делать;

• обеспечить разработчиков системы лучшим пониманием требований к ее созданию;

• определить функциональные границы создаваемой информационной системы;

• обеспечить базис для планирования технического содержания фаз разработки;

• обеспечить базис для оценки стоимости и времени на разработку информационной системы;

• определить графические интерфейсы пользователей с учетом их потребностей и целей.

Основным инструментом для организации работы с требованиями в проекте является IBM Rational

RequisitePro. Он позволяет команде работать с требованиями, отслеживать возможные изменения в них и организовывать обсуждения.

Rational RequisitePro дает возможность организовать контролируемый доступ к требованиям, позволяет всем членам команды проекта быть в курсе текущих требований, их контекста и связей и совместно трудиться над созданием "правильного" продукта, создавать его быстро и качественно.

Благодаря удаленному доступу и привычным способам навигации Интернет стал идеальной средой доступа к данным для совместно работающих команд. Особенно это касается команд, работающих в распределенной или многоплатформенной среде разработки. Rational RequisitePro предоставляет возможность работы как в режиме стандартного интерфейса Windows, так и Web-интерфейса (Rational

RequisiteWeb).

Rational RequisitePro через Web-интерфейс открывает доступ к требованиям всем участникам команды, работающим удаленно или на разных платформах. Rational RequisiteWeb позволяет организовать доступ к требованиям, запросы необходимых требований, их добавление и коррекцию через обычный Web-браузер.

Кроме того, члены группы могут участвовать в онлайновых дискуссиях. Любая дискуссия может быть ассоциирована с определенным требованием, набором требований или проектом в целом. Участие в дискуссии может осуществляться через Web- или Windows-клиента, а также по электронной почте, благодаря чему состав участников может быть расширен за счет представителей заказчика, не использующих RequisitePro. Дискуссии как способ выработки предложений и изменений представляют собой открытый и контролируемый механизм выяснения вопросов, обсуждения предложений, совместной работы над идеями и уведомления об изменениях.

Документы являются естественной средой, в которой можно создавать требования, указывая для них контекстную и дополнительную информацию в привычном редакторе Microsoft Word. При этом все требования хранятся в отдельной базе требований. И когда требования создаются в Microsoft Word, в действительности просто образуется их связь с требованиями в базе требований.

При этом с требованиями в документах можно работать таким же образом, как и с обычными требованиями в базе, устанавливая для них дополнительную информацию, определяя их приоритеты реализации и любые иные атрибуты. Такая уникальная и надежная архитектура объединяет возможности базы данных и простоту использования Microsoft Word для оптимизации процесса управления требованиями.

Архитектура Rational RequisitePro является крайне гибкой, что позволяет отделять документы от базы требований без ущерба для дальнейшей работы или доступа других членов команды. Авторы требований, подрядчики или иные лица, внешние по отношению к организации, могут составлять, просматривать, редактировать и работать иным способом с этими документами без необходимости доступа к базе требований. Эти документы автоматически синхронизируются с проектом при их возвращении в проект.

Rational RequisitePro позволяет создавать базы требований с использованием СУБД Oracle, Microsoft

SQL Server, SQL Sybase Anywhere Server и Microsoft Access.

RequisitePro дает возможность определить необходимые типы требований и стандартные атрибуты для них в случае нового проекта. Дополнительные атрибуты (приоритет, сложность, состояние, ответственный за реализацию и др.) помогают управлять требованиями намного эффективнее. Это было бы невозможно с помощью ведения одних только документов требований.

RequisitePro предусматривает механизмы для визуального контроля за изменениями в требованиях.

Это позволяет точно определить влияние любых изменений в требованиях или в документах на остальные артефакты проекта. В Rational RequisitePro автоматически ведется история изменений требований, что помогает отслеживать их эволюцию в ходе проекта. В результате всегда можно получить информацию о том, кто, когда, зачем и какие конкретно изменения вносил в проектные требования.

Значительно облегчается процесс анализа требований для выявления тенденций их изменения, получения отчетов о степени готовности и оценок состояния проекта. Последнее возможно через определение фильтров, создание запросов и отчетов по требованиям и последующим экспортом полученных данных в Microsoft Excel, где строятся необходимые аналитические диаграммы и таблицы.

Объединяя мощь базы данных с удобством Word, инструментальный продукт Rational RequisitePro

предлагает надежное и легкое в использовании решение в области управления требованиями.

Интеграция с другими инструментальными средствами IBM Rational предоставляет мощные средства по управлению ходом проекта.

Web-интерфейс RequisitePro (RequisiteWeb), позволяет организовать эффективную удаленную работу с требованиями.

Интеграция Rational RequisitePro с Rational Software связывает требования с визуальными моделями.

Участники проекта могут легко формировать требования верхнего уровня на основании существующих прецедентов и детализировать эти требования с помощью анализа сценариев указанных прецедентов.

Интеграция Rational RequisitePro с Rational Rose связывает требования с визуальными моделями.

Участники проекта могут легко формировать требования верхнего уровня на основании существующих прецедентов и детализировать эти требования с помощью анализа сценариев указанных прецедентов.

Интеграция Rational RequisitePro и Rational ClearQuest облегчает внесение изменений в проект, связывая запросы на изменения с требованиями, определенными на базе первых. При этом легко получить набор требований, связанных с конкретным запросом на изменение, и наоборот, набор запросов на изменения, связанных с конкретным требованием.

Интеграция Rational RequisitePro с Rational ClearCase позволяет организовать эффективный версионный контроль проектных требований.

Интеграция Rational RequisitePro с Rational TestManager гарантирует, что на основе существующих требований тестировщиками или специалистами по контролю качества будут созданы необходимые сценарии тестирования.

Интеграция Rational RequisitePro с Rational SoDA позволяет автоматизировать создание проектных отчетов по требованиям, используя шаблоны оформления документов, принятые в организации.

Интеграция Rational RequisitePro с Microsoft Project позволяет наладить планирование в проекте в кратчайшие сроки. Требования — это то, что должно быть реализовано, а значит, на их основе можно определить задачи для участников проекта и спланировать их выполнение.

### ТЕМА 4.Тестирование и отладка программ

#### 4.1. Тестирование

Тестирование программного обеспечения занимает от 30 до 50 процентов от всей стоимости разработки. Однако многие полагают, что приложение не может быть хорошо протестировано до момента внедрения. Это заблуждение основано на двух фактах. Во-первых, тестирование ПО является чрезвычайно сложным процессом. Выполнение любой программы может иметь неисчислимое количество различных путей. Во-вторых, тестирование часто проводится без четкой методологии и без требуемой автоматизации с помощью соответствующих инструментальных средств. Сложность создаваемого ПО делает невозможным проведение 100-% тестирования, но хорошо продуманная методология и использование современных инструментальных средств, могут значительно улучшить производительность и эффективность тестирования ПО.

Основные цели тестирования:

• проверить взаимодействие между объектами;

• проверить корректную интеграцию всех модулей системы;

• проверить, что все требования были корректно реализованы;

• идентифицировать дефекты и убедиться, что они максимально выявлены еще до развертывания системы.

Хорошо выполненные тесты, запуск которых осуществляется еще на ранней стадии жизненного цикла, могут значительно снизить стоимость завершения проекта и поддержки ПО. Это может также значительно снизить риски или штрафы, связанные с поставкой ПО плохого качества, исключить низкую производительность работы приложений конечных пользователей, неудобство ввода данных, наличие вычислительных ошибок и ошибочное функциональное поведение системы.Для систем "с особыми требованиями к безопасности", когда отказ в работе может причинить вред людям или принести значительные убытки компании - таких, как системы управления воздушными полетами, управления ракетами или медицинскими поставками, финансовые приложения - высокие требования к качеству ПО являются необходимыми для успеха разрабатываемой системы. Для обычной информационной административной системы такие требования не являются настолько критичными, но эффект от всего лишь одного дефекта может быть, тем не менее, достаточно дорогостоящим.

В соответствии с RUP принята следующая базовая классификация видов тестирования:

**1. Функциональное тестирование**

• непосредственно функциональное тестирование (Function testing);

• тестирование целостности данных (Data integrity testing);

• тестирование на разных платформах (Configuration testing);

• тестирование отказоустойчивости (Failover & recovery testing);

• тестирование доступа (Security testing);

• инсталляционное тестирование (Installation testing);

• тестирование пользовательского интерфейса (User interface testing).

**2. Нагрузочное тестирование**

• непосредственно нагрузочное тестирование (Load testing);

• профилирование производительности (Performance profiling);

• тестирование цикла работы (Business cycle testing);

• тестирование при большой пользовательской нагрузке (Stress testing);

• тестирование на больших объемах данных (Volume testing).

Для автоматизации процессов тестирования IBM Rational предлагает следующие инструментальные средства.

Средства тестирования в линейке продуктов Atlantic, входящие в пакет IBM Rational Professional

Bundle:

• IBM Rational Performance Tester - средствонагрузочноготестирования

• IBM Rational Functional Tester - средство функционального тестирования, включает также

• IBM Rational Manual Tester - средство для организации и проведения ручного тестирования

Эти средства предназначены для функционального и нагрузочного тестирования программного обеспечения, создаваемого на платформе J2EE. Они интегрированы с инструментом IBM Rational Test

Manager, входящего в пакет IBM Rational Team Unified Platform и служащего средством планирования и мониторинга процесса тестирования как для инструментов тестирования новой линейки продуктов Atlantic, так и многоплатформенных средств тестирования линейки продуктов IBM Rational Suite.

Средства тестирования, в составе IBM Rational Suite:

• IBM Rational Robot – cредство разработки, записи и выполнения скриптов автоматизированного функционального и регрессионного тестирования приложений, предоставляющее полную поддержку тестирования всех средств управления Visual Studio.NET.

• IBM Rational XDE Tester - расширенные средства автоматизированного функционального и регрессионного тестирования Java- и Web-приложений из сред разработки Eclipse IDE, IBM WSADи Rational XDE.

• IBM Rational Purify - средство выявления ошибок, связанных с обращением к динамической памяти (версии для Windows и UNIX)

• IBM Rational Quantify - средство выявления узких мест в коде, оказывающих влияние на производительность разрабатываемой информационной системы.

• IBM Rational PureCoverage - средство определения полноты тестирования кода.

• IBM Rational TestFactory - средство для полуавтоматического формирования набора тестовых скриптов, предназначенных для проведения функционального тестирования и обеспечивающих его полноту для конкретной информационной системы; способен выполнить анализ графического интерфейса разрабатываемой ИС и сгенерировать для нее комплексный набор тестов, позволяющий провести максимально полное функциональное тестирование.

Средства управления тестированием в составе Team Unifying Platform:

• IBM Test Manager – средство планирования и мониторинга процесса тестирования, входит в объединяющую платформу Team Unifying Platform, а также в пакет IBM Rational Suite.

Основным инструментом для планирования процесса тестирования, описания его сценариев и управления всем ходом данного процесса является IBM Rational TestManager, который входит в состав пакета Team Unifying Platform и подробно описан в соответствующем разделе. На основе требований на тестирование, которые аккумулируются с помощью IBM Rational RequisitePro, специальный механизм интеграции RequisitePro и TestManager позволяет сформировать план тестирования.

Элементами плана служат сценарии, каждый из которых позволяет протестировать какой-либо аспект работы создаваемой ИС. Сформированный план тестирования дополняется тестовыми скриптами, которые формируются с помощью IBM Rational Robot и позволяют автоматизировать процесс тестирования.

При проведении распределенного тестирования, в частности, когда требуется одновременно проверить систему на разных программно-аппаратных платформах, используются специальные агенты (Test Agents). Таким образом, можно одновременно протестировать систему в операционных системах Windows, Linux, HP-UX, Solaris и AIX.

Для проведения расширенного функционального и регрессионного тестирования Java- и Web-

приложений рекомендуется использовать новый продукт - IBM Rational XDE Tester. Для автоматизации процесса создания необходимых отчетов — плана тестирования, отчетов по результатам тестирования и т. п. — могут быть использованы Seagate Crystal Report и IBM Rational SoDA.

Для анализа работы систем, построенных с помощью языков C/C++, Microsoft Visual Basic, Java, C#

.NET, VB. NET и Java .NET, можно использовать средства IBM Rational Purify, Quantify и PureCoverage.

Основное назначение Purify при тестировании native-приложений (приложения, работа которых не управляется с помощью дополнительного окружения, например, так называемых сборщиков мусора) — выявить всевозможные утечки памяти и любые иные ошибки работы с ней. Для приложений класса managed applications

(такие, как Java- и NET-приложения)

Purify позволяет провести эффективное профилирование памяти с целью оптими-зации ее использования. С помощью Quantify

становится проще определить узкие места производительности системы и провести ее настройку вплоть до отдельных строк программного кода. PureCoverage окажется незаменимым, если требуется оценить полноту тестирования системы.

IBM Rational TestFactory - еще одно специализированное средство автоматизи-рованного тестирования, предназначенное для анализа графического интерфейса пользователя (GUI) разрабатываемой информационной системы и генерации полного набора тестов, позволяющего провести максимально полное функциональ-ное тестирование.

Управление тестированием. IBM Rational TestManager

Специализированный инструмент для организации тестирования в составе Team Unifying Platform - Rational

TestManager - значительно повышает эффективность процесса тестирования, предоставляя в распоряжение команды общие средства планирования, проектирования, исполнения и анализа результатов тестов. Объединение операций тестирования в единый процесс дает возможность проектной команде наладить эффективный процесс контроля качества создаваемого продукта.

В Rational TestManager ведется план тестирования. Источником для конкретных сценариев тестирования могут служить элементы визуальных моделей и требования. С помощью Rational TestManager организуется единое рабочее пространство, объединяющее элементы плана тестирования, сценарии и тестовые скрипты, отчеты по результатам тестирования и другие необходимые данные.

Rational TestManager связывает наборы тестов с конкретной итерацией процесса разработки и конфигурацией разрабатываемой информационной системы.

Интеграция TestManager с Robot позволяет эффективно автоматизировать процесс тестирования благодаря объединению возможностей планирования и управления распределенным тестированиям с мощными возможностями записи и воспроизведения автоматизированных тестов для ИС, построенных практически по любой технологии.

Интеграция TestManger с Eclipse IDE предоставляет богатые возможности для автоматизации процесса тестирования с продуктами, являющимися производными по отношению к Eclipse (XDE Tester, средства тестирования в составе инструментов Atlantic и т.д.).

Интеграция TestManager с Rational Rose позволяет использовать функциональные элементы визуальной модели в качестве источника сценариев тестирования.

Интеграция TestManager с Rational RequisitePro позволяет использовать требования репозитория требований для определения других сценариев тестирования.

Интеграция TestManager с Rational ClearQuest дает возможность прямо заносить описания ошибок в репозиторий запросов на изменения проекта.

Интеграция TestManager с Rational SoDA позволяет наладить процесс автоматической генерации документов по репозиторию тестирования.

IBM Rational Robot

IBM Rational Robot — средство, которое позволяет создавать, изменять и выполнять автоматизированные тесты для интернет-приложений, ERP- и клиент-серверных приложений. Rational Robot обеспечивает объектную поддержку для тестирования приложений, созданных с использованием технологий, спецификаций и инструментария Java, HTML, DHTML, Visual Basic, PowerBuilder, Oracle Developer/2000, Microsoft Visual Studio и т. д. Rational Robot является решением для автоматизированного тестирования, которое позволяет, единожды написав тест, вторично использовать его на любых платформах.

Технология Object Testing, используемая в Rational Robot, позволяет всесторонне тестировать приложения, что особенно важно в современных средах разработки, где производительность разработки ПО с использованием встроенных компонентов резко возросла. Эти различные библиотеки классов Java, средств управления ActiveXControl, OLEControl (OCX), VisualBasicControl (VBX), объекты Visual Basic, Win32 Control и т. д. Тестирования всего лишь графических интерфейсов этих объектов недостаточно. Также важно протестировать невидимые свойства, например вложенные SQL-запросы и свойства, контролирующие их поведение.

Rational Robot генерирует сценарии функциональных тестов на языке SQABasic, который синтаксически подобен обычному Visual Basic. С помощью SQABasic можно просматривать и редактировать сценарии тестов прямо во время записи. Rational Robot включает в себя встроенный редактор и отладчик с режимом анимационного воспроизведения и онлайновой проверкой синтаксиса скрипта. В любое время можно дополнить сценарии тестов какими-либо процедурами и логическими условиями. При этом доступен вызов любой функции из DLL или из API-интерфейса Windows.

Для всеохватывающего тестирования Web-, ERP- и клиент-серверных приложений необходимо протестировать все компоненты приложения при различных условиях. Rational Robot предоставляет тестовые сценарии для меню, списков, буквенно-числовых символов и многих других объектов. Но можно и самостоятельно определить тестовые сценарии, вызывающие внешние DLL или файлы исполнения.

Rational Robot также предоставляет специализированные тестовые сценарии для специфических объектов, например Java Control, ActiveX Control, OCX, VBX, PowerBuilder DataWindow, специальных объектов Oracle

Form, объектов Visual Basic и т. д. Можно даже "научить" Rational Robot понимать неизвестные объекты и настраивать его таким образом, что он будет уверенно распознавать их в процессе тестирования.

Rational Robot позволяет определять многочисленные таймеры для дополнительной оценки производительности приложения.

Rational Robot автоматически вносит информацию с результатами всех тестов в масштабируемый встроенный репозитарий Rational Repository. Инструмент Test Log Viewer служит для облегчения визуального анализа результатов и отображает их результаты тестирования различными цветами.

Простой выбор данных позволит перейти непосредственно на соответствующую исполняемую строку в тестовом скрипте, ускоряя при этом анализ результатов.

Rational Robot обеспечивает возможностью тестирования приложений, созданных в различных интегрированных средах разработки:

• HTML и DHTML;

• Java-среды разработки (Sun JDK, Symantec Visual Cafe, Microsoft J++);

• OracleDeveloper/2000;

• SAPR/3;

• средыразработкиVisualBasic-приложений;

• PeopleTools;

• PowerBuilder;

• MicrosoftVisualStudio .NET.

IBM Rational Quantify

Rational Quantify позволяет точно определить узкие места в производительности приложения. Rational

Quantify предлагает графическое отображение данных производительности, с помощью которого легко оценить, какие участки кода являются наиболее критичными с точки зрения производительности.

Проблемы с производительностью приложения возникают по множеству различных причин. Среди них — проблемы в проектировании, проблемы в кодировании, чрезмерные или ненужные вызовы процедур. Обнаружение таких медленных функций и их коррекция — от самых медленных к самым быстрым — обычно не является эффективным способом улучшения производительности приложения.

Rational Quantify помогает быстро и эффективно добраться до реального источника проблем, связанных с производительностью. С помощью утилиты Call Graph можно увидеть целостную картину

c графическим интуитивно понятным представлением последовательности вызовов функций.

Жирные линии на графическом представлении покажут, какая последовательность вызовов функций потребовала больше всего времени при выполнении программы, а также важность конкретной функции с точки зрения производительности системы.

При разработке ПОс использованием компонентов обнаружение "узких" мест производительности особенно затруднено. При этом важно выявить, где именно находятся эти "узкие" места. Без Rational

Quantify можно потратить недели в попытках решить проблему производительности, которая находится вовсе не в написанном коде, а скрыта где-то в стороннем компоненте.

Rational Quantify позволяет непосредственно перейти к тому коду приложения, где обнаружена проблема. С помощью инструмента легко выполнить анализ либо всего приложения, либо его отдельных участков за счет настройки дополнительных фильтров. Фильтры Rational Quantify дают полный контроль над представлением и организацией данных о производительности, что помогает выборочно отображать те части приложения, которые особенно выиграют от настройки производительности.

Rational Quantify дает возможность установить "узкие" места производительности вплоть до отдельной строки кода.

Thread Analyzer проверяет каждый исполняемый поток и показывает их состояния. Иначе достаточно трудно определить, что каждый поток делает в конкретный момент времени. Rational Quantify дает ответ на этот вопрос в наглядном виде.

Rational Quantify предлагает мощные функции анализа полученной информации. Анализ различий между различными прогонами дает возможность визуально сравнить время выполнения между двумя прогонами, оценить насколько эффективной оказались сделанные в коде изменения по результатам предыдущего тестирования.

Интеграция Quantify с Rational ClearQuest позволяет сохранить информацию об обнаруженной ошибке в общем репозитории запросов на изменения.

Интеграция Quantify с Rational TestManager позволяет выполнить анализ в реальном времени в процессе общего тестирования системы. Таким образом, можно выявить узкие места производительности системы.

**IBM Rational Purify**

Rational Purify — средство автоматического выявления ошибок, связанных с памятью. Rational Purify

помогает командам разработчиков выявлять источники ошибок с самого начала процесса разработки и в конечном счете создавать качественный продукт.

Rational Purify предлагает быстрый и эффективный способ отладки приложений в реальном времени. Сиспользованием данного средства можно основное внимание сосредоточить на непосредственном написании кода и значительно сократить время на его отладку. При использовании Purify с начальной стадии проекта и регулярно в течение всего процесса разработки заметно повышается гарантия необходимого уровня качества к тому моменту, как программный код будет передан тестировщикам. Purify помогает обнаружить фатальные ошибки доступа к памяти и избежать "аврала" в последние часы сдачи проекта.

Компонентные технологии являются составной частью современных программных приложений, и ошибки в таких компонентах могут оказывать на приложения катастрофическое воздействие.

Обнаружение ошибок програм-мирования, таких как использо-вание неинициализированной памяти, утечек памяти и повторной инициализации уже используемой памяти, состав-ляют особую трудность при разработке ПО с применением компонентов. Необходимо про-верять доступ к памяти вне и внутри компонентов; в против-ном случае можно потратить огромное количество времени на исправление программной ошибки, которая находится не в написанном вами коде, а скрыта в самом компоненте.

Rational Purify способен автома-тически выявить источник ошибки в исходном тексте. Если доступен исходный текст, то непосредственно из Rational Purify может быть запущен редактор, с помощью которого можно быстро исправить обнаруженные ошибки.

Запатентованная технология Object Code Insertion позволяет Purify выявлять фатальные и разрушающие ошибки доступа к памяти не только в исходном тексте, но и в библиотеках. При этом необязательно наличие их исходного текста. Purify проверяет компоненты, включая элементы управления ActiveX, COM/DCOM-компоненты, ODBC-компоненты, DLL-библиотеки, компоненты сторонних разработчиков, а также любые компоненты C++ или Java в приложениях, поддерживающих OLE. С помощью Rational Purify можно добиться высокого качества, тщательно тестируя приложение и выявляя дефекты в работе компонентов.

Java-программисты и тестировщики могут использовать Rational Purify совместно с виртуальными машинами Java, повышая эффективность использования памяти в Java-аплетах и Java-приложениях.

Rational Purify может выполнять аплеты и целые Java-приложения. Аналогичным образом Rational

Purify оказывается полезным при анализе использования памяти в приложения, создаваемых по технологии Microsoft .NET.

Использовать Rational Purify исключительно легко: он не отвлекает разработчика от непосредственной работы по созданию кода. Поскольку он легко интегрируется в популярные среды разработки приложений (Microsoft Visual Studio, IBM WSAD, Eclipse IDE), то можно в автоматическом режиме использовать Purify и редактировать исходный код при возникновении ошибок прямо в среде. Таким образом, нарушение привычного процесса разработки сводится к минимуму, и практически нет никакой потери в скорости программирования. При этом достигается значительный выигрыш в общей эффективности работы. Активное тестирование системы и подсистем в реальном времени — мощное дополнение к функциональному и нагрузочному тестированию.

Вообще говоря, тестирование качества приложений имеет несколько измерений: функциональность, надежность, производительность и др. Достаточно часто организации, проверяющие качество, обнаруживают проблемы с надежностью только тогда, когда они случайно наталкиваются на них в ходе функционального тестирования. Программные ошибки, связанные с доступом к памяти и ответственные за сбой приложения, могут проявляться не на каждом компьютере, на котором исполняется такое приложение. В ходе разработки и тестирования эти ошибки могут оставаться скрытыми, а симптомы проявятся только после того, как пройдет некоторое время (иногда через несколько месяцев). Когда указанные ошибки проявятся, для решения подобных проблем придется создавать заплатку за заплаткой. Разумеется, лучше предотвратить появление таких ошибок.

Помимо нахождения критических ошибок Rational Purify также может сообщить, насколько глубоко проведено тестирование программного кода в режиме реального времени. Если установлен Rational

PureCoverage, то Purify может автоматически собирать данные о степени тестирования кода при каждом запуске. Это повысит уверенность в том, что осуществлен полный поиск ошибок.

Интеграция Purify с Rational ClearQuest позволяет сохранить информацию об обнаруженной ошибке в общем репозитории запросов на изменения.

Интеграция Purify с Rational TestManager позволяет выполнить анализ в реальном времени в процессе общего тестирования системы. Таким образом, можно провести анализ утечек памяти (C/C++ и др.

приложения, при работе которых не задействованы “сборщики мусора” – Garbage Collectors) или профайлинг памяти (Java и .NET приложения).

**Rational PureCoverage**

Главная причина раннего завершения многих проектов — обнаруживаемые слишком поздно проблемы с надежностью. В результате стандарты качества часто начинают приноситься в жертву по мере того, как релиз проекта выходит за рамки сроков и бюджета. Можно уменьшить этот риск, пытаясь обнаружить проблемы с надежностью на более ранних этапах цикла разработки.

Rational PureCoverage — средство автоматического определения полноты тестирования кода в приложениях и компонентах, разработанных на Visual C/C++, Java и Visual Basiс, C# .NET, Java .NET, Visual

Basic .NET. Rational PureCoverage позволяет разработчикам обеспечивать качество и определять эффективность своих усилий в области тестирования. Таким образом, команды разработчиков могут быстрее выпускать качественное ПО, т.е. процесс разработки становится эффективнее.

Независимо от того, производится ли неформальное специальное тестирование или формальное тестирование модуля как части цикла разработки, всеобъемлющее тестирование и быстрая идентификация проблемных областей в процессе разработки являются необходимыми для выпуска качественного ПО. Но без дополнительного инструмента, помогающего определить, какая часть кода была проверена, невозможно эффективно протестировать приложение. Есть вероятность пропуска кода в процессе тестирования, а значит, вероятность пропуска критических ошибок.

Rational PureCoverage устраняет случайный характер тестирования и помогает более полно тестировать код. Являясь важным инструментом разработки приложений и компонентов, Rational PureCoverage

позволяет производить тестирование до или во время важных этапов разработки, например проверки кода, тестирования модуля, в процессе интеграции и определять неработающий код и упущения в тестировании.

Используя PureCoverage с начальных этапов процесса разработки для нахождения и устранения упущений в формальном и неформальном тестировании, можно проверить выполнение всего кода и получить возможность обнаружить ошибки.

Rational PureCoverage позволяет объединять и сравнивать данные о покрытии после многочисленных запусков одного и того же исполняемого кода для получения агрегированной информации о полноте тестирования. Это дает возможность оценить параметры тестирования и обеспечить полноту тестирования всего кода после многочисленных его запусков.

Интеграция PureCoverage с Rational ClearQuest позволяет сохранить информацию об обнаруженной ошибке в общем репозитории запросов на изменения.

Интеграция PureCoverage с Rational TestManager позволяет выполнить анализ в реальном времени в процессе общего тестирования системы. Таким образом, можно определить степень покрытия тестами разрабатываемого ПО.

**IBMRationalPerformanceTester**

IBMRationalPerformanceTester – это инструмент для создания, выполнения и анализа тестов производительности, предназ-наченный для проверки надежности и масштабируемости Web-приложений до их развертывания. Этот инструмент объединяет простое в использовании средство для записи тестов с масштабируемым механизмом выполнения, поддерживающим планирование, создание отчетов в реальном времени и автоматическое варьирование данных. Все это позволяет гарантировать, что приложения смогут справиться с обработкой больших объемов пользовательской нагрузки.

Успешность применения Web-приложений определяется такими факторами как функциональность и производительность. «Функциональность» обозначает доступные пользователю функции, включая выполнение транзакций и доступ к информации. «Производительность»

характеризует способность системы быстро и точно выполнять транзакции и предоставлять информацию несмотря на работу в многопользо-вательском режиме или ограни-ченные аппаратные ресурсы.

Отказ приложения, связанный с производи-тельностью, можно предотвратить с помощью предва-рительного тестирования до этапа развертывания. Однако большинство команд сталкиваются с одной или несколькими из следующих проблем, которые могут повлечь снижение доступности, надежности и масшта-бируемости при развертывании приложения в реальных условиях:

• Ошибки, связанные с пониманием реального поведения пользователей

• Неполное тестирование производительности до развертывания, либо отсутствие такого этапа

• Недостаток опыта cоздания тестов производительности, которые точно отражают действия пользователей.

Решение IBM® Rational® Performance Tester было создано для решения этих проблем. Performance

Tester уменьшает уровень сложности тестирования рабочих нагрузок и позволяют командам любой квалификации эффективно оценивать производительность приложения до его развертывания.

**Простота освоения для новичков.** Rational Performance Tester содержит функции, специально созданные для быстрого освоения. Для записи теста необходимо простое взаимодействие с целевым Web-приложением через браузер специалиста по тестированию – Internet Explorer, Netscape или Mozilla, – работающего под управлением операционной системы Windows или Linux. Результаты теста представляются в визуальном редакторе высокого уровня, который поддерживает любой требуемый эксперту уровень детализации. Динамические отклики сервера автоматически идентифицируются и обрабатываются, что помогает новичкам проводить тестирование на основе данных с варьированием входных данных для разных эмулируемых пользователей – не прибегая при этом к ручной модификации теста. С помощью встроенного средства для просмотра Web-страниц каждую загруженную страницу можно просматривать в оффлайновом режиме, анализируя взаимодействие пользователей и транзакции.

Тестовые скрипты можно объединять в различных комбинациях, моделируя поведение различных групп пользователей, входящих в состав предполагаемой аудитории. После этого специалисты по тестированию могут задать ряд моделируемых транзакций пользователей для выполнения эмуляции.

Выполнение теста сопровождается созданием удобных для чтения отчетов, которые обновляются в реальном времени в ходе всего тестового прогона. Узкие места идентифицируются по таким показателям, как производительность при работе «по замкнутому циклу», скорость выполнения транзакций и диагностика системы.

**Расширенные функции для экспертов.** Помимо средств, которые помогают ускорить освоение инструмента новичками, Rational Performance Tester включает также расширенные функции для экспертов, включая членов команды, опытных специалистов по тестированию и ИТ-администраторов.

Время отклика для отдельной страницы может быть «разложено» на аналогичные показатели для отдельных элементов страницы (например, JPG, JSP, ASP), что помогает идентифицировать элементы, ответственные за медленный отклик. В дополнении к средствам автоматической генерации и корреляции данных, разработчики могут вставлять собственный программный код Java, который может быть исполнен в любой момент в ходе тестирования. Это позволяет реализовать усовершенствованные методы диагностики и манипулирования данными.

В ходе проведения теста, информация о системных ресурсах, например, коэффициент использования центрального процессора (ЦП) и памяти, может собираться с удаленного сервера и корректироваться с учетом времени отклика и данных о пропускной способности. Собранные данные об использовании ресурсов имеют критически важное значение для определения удаленной системы

– маршрутизатора, Web-сервера, сервера приложений, сервера базы данных и т.д. – которая отвечает за обнаруженные простои, а также для идентификации компонента системы (ЦП, оперативная память, жесткий диск и т.д.), который стал причиной узкого места.

**Уменьшение расходов на тестирование производительности.** Rational Performance Tester

характеризуется низкими требованиями к ресурсам процессора и памяти при эмулировании многопользовательского режима. В результате, высокий уровень масштабируемости достигается даже в том случае, если команда не имеет доступа к мощным компьютерам. Кроме того, тесты могут выполняться под управлением операционных систем Windows и Linux, что оптимизирует использование существующих аппаратных ресурсов.

**Поддержка открытых стандартов.** Rational Performance Tester создан на базе архитектурной оболочки Eclipse и ее расширения Hyades, которое предлагает расширенные функции тестирования и мониторинга. Eclipse и Hyades являются проектами на основе открытого исходного кода, которые поддерживают совместно используемые службы (на основе открытого исходного кода) для разных инструментальных сред, предназначенных для мониторинга, разработки, тестирования и развертывания приложений. Использование этой инфраструктуры приносит ряд дополнительных преимуществ – от «ухода» от патентованных форматов хранения данных до поддержки настройки для внутренних и сторонних систем. Инвестирование средств в инструменты на базе Eclipse и Hyades

помогает избежать «зацикливания» на одном поставщике и поощряет внедрение инноваций.

IBM Rational Performance Tester являетсясоставнойчастьюплатформы IBM Software Development

Platform. Интеграция с платформой Software Development Platform обеспечивает совместный доступ пользователей к различным средствам поддержки команд разработчиков. С точки зрения тестирования производительности, это позволяет разработчикам и специалистам по тестированию создавать, совместно использовать и анализировать тесты производительности с использованием того же пользовательского интерфейса, с помощью которого они осуществляют мониторинг, строят, тестируют и развертывают свои приложения в рамках Software Development Platform с применением и других инструментов.

IBMRationalPerformanceTester входит в состав пакета IBMRationalProfessionalBundle, который включает все инструменты, необходимые для проектирования, создания и тестирования приложений J2EE/порталов/служб.

**IBMRationalFunctionalTester**

IBM Rational Functional Tester представляет собой усовершенствованный инструмент для автоматического функционального и регрессионного тестирования, предназначенный для разработчиков интерфейсов GUI и специалистов по тестированию, которым необходим полный контроль за тестированием программных приложений. Это инструмент упрощает сложные задачи для новичков и, в то же время, предоставляет в распоряжение экспертов средства для работы со скриптовыми языками тестирования – Java и VB.NET, обеспечивая неограниченные возможности для адаптации.

Обычно функциональное тестирование входит в сферу ответственности отделов контроля качества, которые контролируют успешную реализацию требований с точки зрения функциональности, удобства и простоты использования и т.д. Функциональное тестирование позволяет оценить какая доля требований к приложению уже выполнена, какой процент из них успешно реализован и насколько команда отклонилась от плановых показателей проекта.

Хотя функциональное тестирование может быть выполнено и с использованием ручного подхода, автоматизация позволяет получить значительные выгоды:

• Позволяет проводить регрессионное тестирование – по отличительным признакам различных версий – без «связывания» тестовых ресурсов

• Освобождает персонал контроля качества от выполнения базового тестирования, помогая им создавать дополнительные, более полные тесты

• Автоматизирует вспомогательные операции, такие, как подготовка стендового оборудования и конфигурирование баз данных

• Уменьшает количество ошибок, связанных с человеческим фактором, в ходе выполнения таких операций, как прогон шагов теста и запись результатов тестирования.

Решение IBM® Rational® Functional Tester позволяет автоматизировать функциональное и регрессионное тестирование. Созданный на основе технологий Java, Web и VS.NETWinForm, FunctionalTester объединяет надежное средство для записи действий пользователя с многочисленными опциями настрой-ки и интеллектуальными функциями для работы со скриптами, что позволяет создавать и выполнять тесты с необходимым уровнем гибкости, соответст-вующим постоянному изменению приложений.

**Снижение затрат на функциональное тестирование.** Преимущества автоматического тестирования быстро сводятся на нет, если команде приходится тратить много времени на ручное обновление скриптов тестирования на протяжении всего проекта. Functional Tester использует усовершенствованную технологию ScriptAssure для того, чтобы «изучить» контрольные характеристики пользовательского интерфейса, что позволяет идентифицировать те же самые средства управления в новой версии, несмотря на внесенные изменения.

Эти характеристики сохраняются в объектной карте, совместный доступ к которой могут получить различные скрипты и участники проекта. Благодаря этой карте изменения, внесенные в характеристики распознавания объекта, будут отражены во всех скриптах тестирования, что существенно упрощает обслуживание. А новые, расширенные средства фильтрации и поиска по объектным картам сделали обслуживание еще более простой задачей.

В текущую версию Functional Tester включена обновленная версия функции по автоматическому соотнесению данных, которая позволяет командам реализовывать скрипты для тестирования на основе данных – меняя входные данные в ходе теста – без ручного изменения программного кода. Новый редактор пулов данных использует интерфейс на основе электронных таблиц для моделирования и сохранения ценных данных тестирования, которые будут использоваться в ходе выполнения теста.

Контрольные точки в составе программного кода теста, которые используются для сравнения фактических данных с ожидаемыми, поддерживают применение обычных выражений для согласования колебаний данных, например, при сравнении фрагментов сложных последовательностей. Все это позволяет дополнительно уменьшить объем операций по обслуживанию тестов при работе с приложениями, в которые вносились изменения.

**Неограниченные возможности адаптации и настройки.** Модификация и совершенствование скриптов тестирования неизбежны. Но абсолютно все – от простых циклических структур до расширенного управления файлами и взаимодействия с операционными системами – может быть в полной мере реализовано с помощью инструмента для автоматического тестирования. Functional Tester поддерживает популярные языки программирования, для работы с которыми можно использовать надежные и эффективные редакторы и отладчики.

Functional Tester поддерживает создание скриптов тестирования на языке Java – с использованием редактора Eclipse Java Developer Toolkit, или VB.NET – с использованием редактора VS.NET. Любой специалист, знакомый с этими языками программирования, сможет быстро модифицировать существующий код, а также создать пользовательские библиотеки, которые могут быть вызваны другими скриптами тестирования. Написание программного кода значительно упрощается благодаря использованию редакторов Eclipse и VS.NET, которые предоставляют в распоряжение пользователя расширенные средства редактирования, например, автоматическое завершение кода и улучшенные опции отладки. Возможности адаптации и настройки практически ничем не ограничены, а передача скриптов в рамках команды в значительной степени упрощена. Фактически, разработчики интерфейсов GUI и другие члены команды могут получить доступ к Functional Tester для создания собственных тестов из своей среды IDE (интегрированная среда разработки) на базе Eclipse или VS.NET. В качестве примеров среды IDE на базе Eclipse можно привести IBM Rational Software

Architect и IBM Rational Application Developer for WebSphere Software.

**Поддержка распределенных команд разработчиков.** В ряде случаев, важное значение приобретает возможность сохранения нескольких версий одного тестового скрипта, что особенно актуально для поддержки работы распределенных команд. Для поддержки контроля версий команды разработчиков могут использовать продукт IBM Rational ClearCaseLT, который поставляется в одном пакете с Functional Tester, а географически распределенные команды могут расширить его до полной версии IBM Rational ClearCase.

**Поддержка открытых стандартов.** Средства Rational Functional Tester, предназначенные для создания скриптов на языке Java, созданы на базе архитектурной оболочки Eclipse и ее расширения Hyades, которое предлагает расширенные функции тестирования и мониторинга. Eclipse и Hyades

являются проектами на основе открытого исходного кода, которые поддерживают совместно используемые службы (на основе открытого исходного кода) для разных инструментальных сред, предназначенных для мониторинга, разработки, тестирования и развертывания приложений.

Использование этой инфраструктуры приносит дополнительные преимущества –от «ухода» от патентованных форматов хранения данных до поддержки настройки для внутренних и сторонних систем. Инвестирование средств в инструменты на базе Eclipse и Hyades помогает избежать

«зацикливания» на одном поставщике и поощряет внедрение инноваций.

IBM Rational Functional Tester являетсясоставнойчастьюплатформы IBM Software Development

Platform. Интеграция с платформой Software Development Platform обеспечивает доступ к различным средствам по управлению жизненным циклом разработки ПО, предназначенным, например, для определения требований к приложениям, и поддерживает совместную работу членов команды разработчиков. С точки зрения функционального тестирования, это позволяет разработчикам и специалистам по тестированию создавать, совместно использовать и анализировать функциональные тесты с использованием того же самого пользовательского интерфейса, с помощью которого они осуществляют мониторинг, строят, тестируют и развертывают свои приложения в рамках платформы Software Development Platform (с применением и других инструментов).

IBMRationalFunctionalTester входит в состав пакета IBMRationalProfessionalBundle, который включает все инструменты, необходимые предприятию для проектирования, создания и тестирования приложений J2EE/порталов/служб. Для всего пакета предлагается единый контракт на обслуживание всего входящего в него ПО.

**IBM Rational Manual Tester**

IBM Rational Manual Tester – это инструмент для ручного создания и выполнения тестов, который дает возможность повторно использовать шаги теста и тем самым сократить трудоемкость работы тестировщиков и бизнес-аналитиков при внесении изменений в программное обеспечение. Этот продукт упорядочивает процесс ручного тестирования, повышая эффективность и скорость работы по оценке качества приложения.

Лишь немногие организации полностью перешли на автоматизацированные тесты. Поэтому ручное тестирование остается важной частью процесса тестирования – опыт показывает, что более 75% всех работ по тестированию выполняется вручную. Популярность ручного тестирования объясняется следующими причинами:

• возможен прием на работу в качестве тестировщиков экспертов по соответствующему направлению, не имеющих опыта тестирования;

• недостаточный опыт работы с автоматизацией тестирования;

• недостаток инструментов автоматизации тестирования;

• плохая проработка или полное отсутствие в тестируемых приложениях интерфейсов, которые поддерживали бы автоматизацию IBM Rational Manual Tester предназначен для команд, в которых хотя бы часть тестирования выполняется вручную. Он позволяет преодолеть привязанный к конкретике характер ручных методов проверки, с тем чтобы гарантировать оптимальное тестирование и обнаружение дефектов в сфере деятельности, которая иначе была бы связана с высокими затратами и давала малую отдачу.

**Ручное тестирование – сложнее, чем вы думаете.** Ручное тестирование на первый взгляд выглядит обманчиво простым, но при неграмотной реализации превращается в настоящее испытание. Проектные команды быстро осознают всю сложность успешного управления несколькими электронными таблицами и другими средствами для хранения тестов и их результатов. Скоро становится ясно, какой объем повторной работы сопряжен с отсутствием возможности повторного использования и модульности среди множества задействованных тестовых сценариев. И все это время они борются с человеческими ошибками и непоследовательностью исполнителей. Эти недостатки зачастую сводят на нет сам смысл ручного тестирования – оценку качества приложения и обнаружение его дефектов.

Rational Manual Tester помогает группам преодолеть эти сложности на пути к успеху проекта, предоставляя средства для повышения производительности ручного тестирования. Результатом становится то, что большая часть времени и усилий направляется на выполнение сложных, информативных ручных тестов, составление лучшего представления о качестве приложения и возможность проверить больше возможностей приложения до его выпуска.

**Расширенные возможности создания и поддержки тестов.** Rational Manual Tester способствует использованию ссылочного контента – возможности определить те или иные шаги теста как многократно используемые компоненты, которые можно применять в нескольких тестах. Преимущество здесь не только в возможности повторного использования, которая в любом случае экономит время и деньги.

Преимущество еще и в том, что оптимизируется контроль качества и единообразия тестовых сценариев.

При внесении изменений в ссылочный контент они автоматически применяются ко всем тестовым сценариям, использующим этот контент, что гарантирует точное применение изменений во всей группе тестирования.

Редактор тестов поддерживает настраиваемые шрифты, создание скриншотов и вложение файлов, что позволяет тестерам четко и ясно вести обмен информацией по поводу шагов тестов. Manual Tester

позволяет значительно снизить высокий уровень неопределенности, который часто является бичом других методов определения и описания шагов теста.

**Автоматизация – это простота.** Rational Manual Tester снижает вероятность человеческой ошибки в областях, наиболее чреватых такими ошибками – вводе и верификации данных. Во-первых, Manual Tester

хранит данные в специальном поле тестового сценария, что позволяет тестировщику вставлять в него данные, не набирая их напрямую с клавиатуры. Во-вторых, данные, скопированные из пользовательского интерфейса приложения, автоматически сравниваются с ожидаемыми значениями, что автоматизирует сравнение и сохранение результатов в консолидированном тестовом отчете.

В ручные тесты также можно добавлять пользовательские точки принятия решений. Для записи успеха или неудачи конкретных событий используются точки верификации, которые позволяют запросить у тестировщика ввод дополнительных данных о выполняемой задаче. Вся эта информация содержится в отчетах, которые могут экспортироваться в документы формата CSV для хранения и анализа с помощью любого инструмента, включая популярные электронные таблицы.

**Применение для различных целей.** Rational Manual Tester позволяет настраивать тесты так, чтобы они отражали принятые в команде тестировщиков метрики процесса и значения. Можно определять пользовательские поля и сопоставлять их отдельным шагам тестов, чтобы охватить больший спектр тестирования, а также хранить дополнительную информацию, собранную в ходе выполнения тестов.

Тестировщики также могут прилагать файлы к любым шагам в ходе выполнения теста – простой способ приложить скриншоты или другую нетекстовую информацию, которая может оказаться полезной при анализе результатов тестирования. И папку Reuse, в которой хранятся многократно используемые блоки шагов тестов, и собственные хранилища данных можно поместить в сетевую папку общего доступа или подчинить механизму контроля изменений. Manual Tester может использоваться на любых РС под управлением Windows®, поддерживая работу распределенных групп и мобильных сотрудников. Кроме того, этот программный продукт хранит тесты и результаты тестов в централизованном расположении, независимо от географического распределения и состава группы, что гарантирует простой доступ к тестовым ресурсам и составление отчетов по ним.

**На базе открытых стандартов.** Rational Manual Tester основан на базе архитектурной схемы Eclipse

и ее расширения Hyades для тестирования, трассировки и мониторинга. Eclipse и Hyades – это два проекта с открытым исходным кодом, которые предоставляют совместно используемые службы с открытым кодом в инструментальных средах, предназначенных для разработки, тестирования, развертывания и мониторинга приложений. Преимущества этой инфраструктуры очень многочисленны и разнообразны – от устранения нестандартных форматов хранилищ данных до поддержки как внутренней, так и независимой доработки и расширения возможностей приложений.

Капиталовложения в инструменты на базе Eclipse и Hyades позволяют избежать жесткой привязки к поставщику и способствуют развитию новаторства в будущем.

IBM Rational Manual Tester является неотъемлемым компонентом платформы IBM Software

DevelopmentPlatform.

IBMRationalManualTester входит в состав пакета IBMRationalProfessionalBundle, который состоит из всех настольных инструментов, необходимых предприятию для разработки, конструирования и тестирования приложений, ориентированных на J2EE/порталы/службы, в едином наборе и по единому контракту на техобслуживание

**IBM Rational XDE Tester**

IBM Rational XDE Tester является эффективным инструментом функционального тестирования для приложений на Java и для Web-приложений и предоставляет следующие возможности:

• Java в качестве языка тестовых сценариев, тестирование приложений на Java (J2EE, J2SE, SWT, средства управления AWT/JFC);

• технологию ScriptAssure, которая практически полностью устраняет трудности с сопровождением сценариев, наряду с предоставлением бесшовной проверки достоверности динамических данных;

• мощную среду тестирования в IDE-оболочке Eclipse

IBM Rational XDE Tester интегрируется с инструментами IBM Rational, предназначенными для управления тестированием, управления требованиями, обнаружения дефектов и управления версиями сценариев тестирования. Эти инструменты и возможности интеграции упорядочивают процесс тестирования, предоставляя удобные возможности взаимодействия между бизнес-аналитиками, тестировщиками и разработчиками.

IBM Rational XDE Tester обладает четко выверенной комбинацией возможностей, предназначенной для экономии времени тестировщиков Java- и Web-приложений, вместе с тем позволяя проводить более тщательное тестирование приложений.

IBM Rational XDE Tester предоставляет в ваше распоряжение всю мощь Java для создания тестовых сценариев. Язык сценариев Java облегчает тестирование заказных пользовательских интерфейсов и средств управления Java.

Вследствие популярности Java становятся легко доступными образцы кода, эффективные приемы и техники. Кроме того, использование общего языка может привести к росту взаимопонимания между тестировщиками и разработчиками, повысив в итоге эффективность реализации всего проекта.

Технология ScriptAssure имеет набор уникальных возможностей, позволяющих:

• устанавливать чувствительность тестовых сценариев в соответствии с изменениями пользовательского интерфейса, при этом устраняя необходимость обновления тестовых сценариев при создании новых версий приложения;

• запускать тестирование функциональности пользовательского интерфейса, прежде чем интерфейс будет окончательно готов;

• проводить проверку достоверности информационного содержимого динамических приложений путем анализа диапазона допустимых откликов.

Технология ScriptAssure освобождает тестировщиков от рутинной работы, позволяя уделить больше времени самому тестированию.

IBM Rational XDE Tester использует мощные возможности Eclipse и IBM WebSphere Studio в качестве среды тестовой разработки. Это предоставляет тестировщикам следующие возможности:

• профессиональную открытую среду разработки;

• современный редактор с автоматическим дополнением кода, предназначенный для помощи в создании и изменении сценариев тестирования;

• полнофункциональный отладчик для простой идентификации и разрешения проблем со сценариями тестирования;

• расширенные возможности прямого обмена информацией с разработчиками.

При функциональном и регрессионном тестировании "толстых клиентов" Java- и Web-приложений выбор IBM Rational XDE Tester наиболее предпочтителен.

IBM Rational XDE Tester интегрирован с IBM Rational Team Unifying Platform — мощным набором средств управления тестированием, обнаружения дефектов, управления версиями сценариев тестирования и управления требованиями. Инструментальные средства, входящие в состав этой платформы, ускоряют разработку приложений, значительно облегчая координацию и коммуникацию внутри коллектива разработчиков.

С помощью инструментов, имеющихся в IBM Rational Team Unifying Platform, весь коллектив разработчиков будет обладать четкой информацией о результатах тестирования. В частности, разработчики смогут легко получить доступ к информации об обнаруженных тестировщиками дефектах, что поможет легко воспроизвести найденные ошибки. Кроме того, управленческому звену будет предоставлена информация, необходимая для оценки степени рисков по каждому этапу проекта.

IBM Rational Team Unifying Platform освобождает тестировщиков от обязанности вручную отслеживать изменения в требованиях, что достигается путем автоматической пометки тестовых сценариев, ссылающихся на измененные требования.

Использование IBM Rational XDE Tester вместе с IBM Rational Team Unifying Platform упорядочивает и рационализирует весь процесс тестирования.

IBM Rational XDE Tester поддерживается со стороны всемирной сервисной организации, объединяющей онлайновые ресурсы с персонализованным обучением, консультированием и технической поддержкой. IBM

Rational Developer Network предоставляет доступ к статьям, технической документации, обучающим курсам, а также к онлайновому сообществу профессоналов-разработчиков, использующих наилучшие практики и инструменты IBM Rational.

Опытные консультанты, различные обучающие ресурсы и надежная техническая поддержка расширят возможности вашего коллектива разработчиков. Rational Services ускоряют достижение успеха, являясь основой для непрерывного совершенствования процессов разработки ПО.

#### 4.2. Управление выполнением проекта и документированием.IBM Rational ProjectConsole, IBM Rational SoDA

**IBMRationalProjectConsole**

IBMRationalProjectConsole автоматизирует процесс создания и мониторинга отчетов о состоянии проекта благодаря созданию динамического Web-сайта, который отображает панель основных показателей проекта. Это экономит время, необходимое для создания и поддержания Web-сайта команды проекта, а также позволяет избежать сбора подобных данных вручную.

Для руководителей проектов IBM Rational

ProjectConsole предоставляет объективную картину достигнутых результатов на протяжении всего жизненного цикла проекта. Rational

ProjectConsole собирает фактические данные о состоянии разработки из платформы Rational

Suite® и продуктов других поставщиков, представляя эти результаты в графическом виде, что позволяет легко и быстро оценить ход и качество проекта.

Rational ProjectConsole помогает ответить на следующие вопросы:

• Где следует сконцентрировать имеющиеся ограниченные ресурсы, чтобы обеспечить выполнение плана проекта

• Какие тенденции могут негативно повлиять на затраты и график работ

• Насколько стабильной является используемая архитектура.

• Сколько строк программного кода требует внесения изменений/добавления и как это соотносится с графиком выпуска релиза

Интеграция с существующими приложениями Rational позволяет получать и просматривать информацию из IBMRationalRose®, IBMRationalRequisitePro®, IBMRationalClearQuest®, IBMRational

ClearCase®, IBM Rational XDE® Developer и IBM Rational Test Manager®.

Точная и доступная информация позволяет снизить риски проекта и принимать своевременные решения с целью их недопущения. В результате IBM Rational ProjectConsole повышает предсказуемость проектов и позволяет соблюдать показатели, заложенные в графике и бюджете проекта.

**Упрощение доступа к информации по проекту.** Rational ProjectConsole предоставляет в распоряжение команды единую точку доступа к точной информации по проекту. Вам больше не придется тратить время на ожидание отчетов о состоянии, отправку многочисленных запросов об обновлении, «копание» в различных приложениях, базах данных, каталогах и документах с целью получения требуемой информации. Вы больше не будете принимать решения на базе устаревшей информации. Благодаря ProjectConsole вы экономите время, уменьшаете число ошибок, связанных с ручным вводом информации, и получаете более полный контроль за проектами по разработке ПО.

**Настраиваемые шаблоны.** Rational ProjectConsole предлагает полный набор типовых шаблонов для создания отчетов, а также возможность создавать собственные шаблоны, что позволяет представлять информацию, нужную членам команды, в удобном формате. Каждый отчет автоматически извлекает информацию из платформы разработки Rational Suite и инструментов других компаний согласно заданному расписанию или по запросу. После этого данные публикуются на Web-сайте, которые сразу доступны всем членам команды. Упрощенный доступ к информации по проекту позволяет уменьшить время на поиск информации и уделять больше внимания самому проекту.

Измерение степени завершенности и качества проекта. Продукт Rational ProjectConsole позволяет автоматически получать количественную оценку текущего состояния проекта и анализировать тенденции.

Измеряемые показатели автоматически формируются средствами Rational Suite и инструментами других поставщиков с последующим их сохранением в хранилище данных.

Результаты измерений, полученные из RationalRequisitePro, RationalClearQuest и RationalClearCase, могут быть графически отображены в Rational ProjectConsole.

Rational ProjectConsole позволяет с помощью гиперссылок связать все объекты и отображать результаты анализа в виде диаграмм, индикаторов и таблиц, публикуя их на автоматически создаваемом и обновляемом Web-сайте.

Объективное состояние проекта. Графическая панель позволяет всем членам команды быстро определять реальное состояние и качество проекта. Rational ProjectConsole позволяет анализировать подробную информацию по отдельным операциям, такие как моделирование, создание программного кода, тестирование. Кроме того, можно визуально сравнить плановые показатели с фактическими, проследить тенденции в хронологических данных и просмотреть междисциплинарные измерения, получив полное представление о состоянии проекта в целом. Все это позволяет команде разработчиков ПО:

• принимать своевременные корректирующие меры;

• анализировать причины задержек;

• иметь реалистичные ожидания в отношении проекта;

• лучше прогнозировать этапные события проекта;

• обеспечивать объективное и точное измерение состояния и качества проекта RationalProjectConsole входит в состав платформы RationalSuiteTeamUnifyingPlatform.

**IBM Rational SoDA**

Формирование проектной документации обычно требует много времени и усилий. Нередко составление документов и отчетов в течение жизненного цикла проектов осуществляется случайным образом, а то и вовсе игнорируется из-за нехватки времени и ресурсов, которые требуются на их разработку и сопровождение.

Специализированное средство IBM Rational SoDA помогает успешно решить эти проблемы, позволяя автоматизировать создание и сопровождение необходимой проектной документации. SoDA значительно облегчает процесс документирования в проекте. Инструмент позволяет более согласованно и оперативно генерировать необходимые документы и отчеты, автоматически извлекая исходную информацию из репозитариев и артефактов проекта, созданных в других инструментальных средствах. Таким образом, SoDA — мощный инструмент для организации обмена жизненно важной информацией проекта между его непосредственными и внешними участниками (заказчиками, конечными пользователями и т. д.).

Использовать Rational SoDA крайне просто. Сначала создаются шаблоны документов, а затем на их основе генерируются сами документы. При этом участники проекта акцентируют внимание только на создании продукта и сопутствующих артефактов. Можно быть абсолютно уверенным, что сгенерированный документ будет всегда включать актуальную информацию, полученную из проектных репозитариев и артефактов. Rational SoDA изначально включает несколько десятков шаблонов.

Основу Rational SoDA составляет Microsoft Word. Любой внешний вид документа, который можно создать в Word, может быть представлен в виде шаблона SoDA. Таким образом, Rational SoDAподдерживает возможность стандартизации типов документов в рамках отдельного проекта или всей организации в целом. Эта стандартизация может обеспечить соответствие документов таким стандартам, как ISO, SEI CMM/CMMI и IEEE, повышая качество проектной документации и облегчая взаимодействие занятых в проекте сотрудников.

Шаблоны SoDA содержат информацию о форматировании документа, его структуре и стилях. В них фиксируется расположение источников информации, из которых извлекаются необходимые данные. По желанию пользователя Rational SoDA может автоматически генерировать документы и отчеты в формате HTML. Это значительно упрощает публикацию документов в Интернет. Данный функционал особенно полезен для распределенных проектных команд.

Rational SoDA генерирует документы, извлекая информацию из следующих проектных репозитариев:

• репозитарий требований Rational RequisitePro;

• репозитарий тестирования Rational TestManager;

• базы данных запросов на изменения Rational ClearQuest;

• версионные объектные базы (VOB) Rational ClearCase;

• общий проектный репозитарий Rational Administrator.

Кроме того, Rational SoDA может извлекать данные из следующих отдельных артефактов проекта:

• модели Rational Rose и Rational Rose RealTime;

• планы Microsoft Project;

• документы Microsoft Word

Документация программного обеспечения для крупных проектов может составлять сотни и тысячи страниц. При этом документы должны быть постоянно актуальными, чтобы отражать текущее состояние дел в проекте. Rational SoDA позволяет обновлять лишь отдельные части документов, что значительно упрощает ведение процесса документирования.

#### 4.3. Управление конфигурациями и изменениями

**IBMRationalClearCase, IBMRationalClearQuest**

Перефразируя модель зрелости процессов Института программной инженерии (SEI CMM), можно сказать, что конфигурационное управление и управление изменениями обеспе-чивает контроль за изменениями и обеспечивает взаимосвязь артефактов проекта. Методы, процессы и инструментальные средства, используемые для обеспечения конфигура-ционного управления и управления изменениями в организации, могут рассматриваться как единая система конфигурационного управления.

Конфигурационное управление и управление изменениями включают:

• идентификацию объектов конфигурационного управления;

• ограничение возможности изменения этих объектов;

• аудит изменений, произведенных с объектами конфигурационного управления;

• определение конфигураций ОКУ и управление этими конфигурациями.

Система конфигурационного управления является необходимой и неотъемлемой частью всего процесса разработки и содержит ключевую информацию о процессах разработки продуктов, их развитии, развертывании и внедрении. Она сохраняет для повторного использования артефакты, получаемые в ходе выполнения проекта.

Инструментом, позволяющим организовать эффективное управление версиями и конфигурациями является ClearCase, а управления изменениями – ClearQuest.

**Rational ClearCase**

Rational ClearCase является инструментом, который значительно упрощает ведение процесса управления версиями и конфигурациями. Он помогает наладить эффективный контроль за любыми артефактами проекта: документами, исходниками, моделями, целыми репозиториями проекта, дополнительными файлами и т.д. Все, что создается и меняется в процессе разработки, может быть поставлено на версионный и конфигурационный контроль. Таким образом, в значительной мере повышается надежность и качество работы проектной команды.

Задача управления различными конфигурациями, релизами и версиями возникает в проектах любого уровня и масштаба. Все усложняется, когда двое или более разработчиков в одно и то же время работают над одним и тем же артефактом. Такая проблема особенно актуальна, если эти разработчики территориально удалены друг от друга и не могут оперативно согласовывать свою деятельность.

Очень важно при этом наладить четкий контроль за работой участников проекта, которые работают с его репозиториями. Это могут быть, например, менеджеры проекта, аналитики, разработчики, тестировщики и многие другие. Очень важно четко представлять, кто и что делает в проекте, кто и что изменил в тот или иной момент времени. Не менее важно, чтобы в проекте всегда была возможность откатиться до некоторого предыдущего состояния, если разработка пошла по ошибочному пути. Это может относиться как к проекту в целом, так и отдельным его частям.

С помощью ClearCase можно организовать версионный контроль не только отдельных файлов, но и целых каталогов. Таким образом, можно создавать новые версии целых рабочих пространств и окружений.

Для каждого артефакта, поставленного на контроль, ведется история изменений. К любой версии можно вернуться в любой момент. Для любых двух разных версий одного и того же артефакта можно посмотреть отличия, если, конечно, эти артефакты принадлежат стандартным типам документов, таким как текстовые файлы, документы Microsoft Word и модели Rational Rose.

Мощные возможности ClearCase в области конфигурационного управления позволят организовать параллельную работу над отдельными конфигурациями и версиями одного и того же продукта или его части.

ClearCase значительно ускоряет командную разработку, обеспечивая поддержку параллельной работы. В этом значительно помогут мощные возможности в организации ветвления версий. Всегда можно провести интеграцию нескольких ветвей в основную. Для этого существуют удобный графический инструментарий, автоматически запускающийся, если требуется осуществить подобный процесс интеграции. В ходе интеграции участник проекта может в автоматическом или ручном режиме указать, что следует включить в конечную версию того или иного артефакта.

Данные, поставленные на версионный и конфигурационный контроль, под управлением ClearCaseхранятся в базах данных специального формата, т.н. Version Object Bases (VOB). Участники проекта имеют доступ к любой из этих VOB через «представления» (Views), каждое из которых отображает некоторый срез проектных данных из VOB. Таким срезом может быть, например, набор артефактов, относящихся к некоторой версии или конфигурации разрабатываемого продукта.

Представления бывают двух типов: динамические и статические. Динамическое представление

(dynamic view) отображает артефакты, актуальность которых гарантируется в любой момент времени.

Здесь имеется в виду, что если один из участников проекта изменил один из артефактов, то любой другой участник проекта будет работать всегда с последним вариантом этого артефакта. Но для работы с динамическими представлениями требуется постоянное подключение к серверной части ClearCase. Статическое представление (snapshot view) представляет собой своеобразный снимок набора артефактов и позволяет работать с ними без постоянного подключения к серверу.

Существует вариант Rational ClearCaseLT, предназначенный для небольших рабочих групп. Его основные отличия от полнофункционального ClearCase:

• Отсутствие динамических видов.

• Работа в однодоменном окружении.

• Отсутствие поддержки ClearCase Multisite.

• Отсутствие возможности создания мультипроектной среды.

Когда у организации появляется необходимость в конфигурационном управлении на уровне предприятия, например, в расширенном управлении процессами сборки, использовании распределенных версионных хранилищ, использовании удаленной репликации данных, то легко провести модернизацию и перейти на полную версию ClearCase. При этом не потребуется изменять определенные процессы работы, какие-либо данные или подходы.

ClearCase включает мощный Web-интерфейс, с помощью которого легко организовать удаленную работу с артефактами проекта.

Для географически распределенных рабочих групп, работающих совместно в одних и тех же проектах, будет крайне полезен Rational ClearCase MultiSite, являющийся стандартом для организации удаленной совместной работы и обеспечивающий непрерывный параллельный процесс разработки. Технология расширенной репликации позволяет распределенным командам работать одновременно в проектах с локальным доступом к актуальным рабочим материалам. ClearCase

плавно интегрируется в следующие рабочие среды: Microsoft Visual Studio, IBM VisualAge for Java, IBM

WebSphere Studio, Sybase PowerBuilder, Microsoft Word идр. Специальные настройки графического интерфейса позволяют разработчикам в большей степени концентрироваться на решении конкретных задач, не отвлекаясь на примитивные рутинные процедуры.

Объединение процессов конфигурационного управления и управления изменениями с помощью механизмов Unifying Change Management (UCM) значительно повышает возможности проекта в области контроля изменений в отдельных артефактах. Являясь функциональной надстройкой над ClearCase, UCM автоматизирует многие операции, обеспечивая параллельную работу с артефактами и их наборами.

Интеграция ClearCase и Rational ClearQuest гарантирует, что никто не внесет изменения в артефакты проекта, пока для него не определена соответствующая задача.

Интеграция ClearCase с Rational Rose позволяет ставить на версионный контроль целые модели или их части прямо из среды визуального моделирования.

Интеграция ClearCase с Rational RequisitePro позволяет организовать процесс управления версиями и конфигурациями для всего репозитория требований. В любой момент времени могут быть восстановлены наборы документов и требований для любой версии продукта в необходимой конфигурации.

Интеграция ClearCase с Rational TestManager позволяет аналогичным образом наладить версионный и конфигурационный контроль над всем репозиторием тестирования. В любой момент времени можно восстановить сценарии и тестовые скрипты для проведения процесса тестирования конкретной версии в необходимой конфигурации.

Интеграция ClearCase с Rational SODA позволяет автоматизировать генерацию документов по артефактам проекта и изменениям, связанными с ними.

Интеграция ClearCase с Eclipse IDE и продуктами, базирующимися на Eclipse предоставляет доступ к возможностям использования версионного и конфигурационного контроля для широкого круга дочерних продуктов, таких как XDE Modeler, XDE Developer for Java, IBM Websphere Studio ApplicationDeveloper, средства Atlantic и т.д.

**RationalClearQuest**

Rational ClearQuest представляет собой мощный инструмент для отслеживания всех типов запросов на изменения для любого проекта. Возможность создавать новые типы запросов на изменения, обладающих собственными атрибутами и имеющих нестандартный жизненный цикл, позволяет адаптировать ClearQuest в любой организации.

Как показывает практика, чтобы обеспечить высокое качество создаваемого продукта, недостаточно просто наладить поиск ошибок. Для этого необходимо обеспечить эффективный процесс управления всеми типами запросов на изменения, возникающих в процессе разработки. К таким запросам относятся, в первую очередь, запросы на улучшение самого продукта или процесса по его созданию, описания ошибок, выявленных в продукте, и любые поручения к участникам проекта. Поскольку все запросы на изменение фиксируется в едином репозитории, то появляется возможность наладить четкий контроль за работой коллектива и осуществлять непрерывный мониторинг проекта.

Размер проекта по разработке информационных систем при этом неограничен, т.к. репозиторий запросов на изменения может быть создан на базе известных СУБД, таких как Microsoft SQL Server, Oracle Server, IBM DB2 Server, Sybase SQL Anywhere Server. Для небольших команд вполне подойдет возможность создания репозитория на базе Microsoft Access.

Web-интерфейс ClearQuest, известный как ClearQuestWeb, позволяет взаимодействовать участникам проекта удаленно практически с любой программно-аппаратной платформы. Кроме того, наличие клиентов для Linux, Sun Solaris, AIX, HP/UX и др. предоставляет мощные возможности для организации процесса запроса на изменения в проекте, в котором происходит создание распределенной многоплатформенной ИС.

Для распределенных команд предлагается расширение в виде Rational ClearQuest Multisite, который на базе реплицирования репозитория запросов на изменения предоставит этим командам полный набор возможностей, предлагаемых инструментарием, что недоступно при использовании Web-интерфейса.

ClearQuest предоставляет полностью документированный интерфейс API на базе COM. Это дает возможность интегрировать ClearQuest, например, с центром приема и обработки телефонных звонков или иными системами, участвующими в бизнес-процессе организации.

Интеграция ClearQuest с Rational RequisitePro позволяет связать требования к разрабатываемой ИС с конкретными запросами на изменение. Таким образом, если появились новые требования или изменились существующие, то легко определить, какие запросы на улучшение или обнаруженные ошибки в системе послужили источником для этого.

Интеграция ClearQuest с Rational ClearCase предоставляет возможность наладить контроль не только за запросами на изменения, но и непосредственно за любыми изменениями в артефакты проекта. Вэтом случае никто не сможет изменить ни один артефакт до тех пор, пока для него не определена соответствующая задача. Таким образом, значительно повышается контроль за ходом процесса разработки.

Интеграция ClearQuest со средствами тестирования позволяет вносить описания обнаруженных ошибок прямо по его результатам. Таким образом, можно создать описание ошибки прямо из лога тестирования, сформированного в Rational TestManager, или из отчета по результатам анализа, выполненного в реальном времени с помощью RationalPurify, RationalQuantify, RationalPureCoverage.

Интеграция ClearQuest с Rational SODA позволяет автоматически формировать отчеты в необходимом виде по любым запросам на изменения.

Интеграция ClearQuest с Microsoft Project позволит быстро сформировать список задач для участников проекта, детализировать этот список до конкретных поручений и реально отслеживать, не выходит ли проект за определенные в плане проекта сроки, ресурсы и бюджет.

### ЛИТЕРАТУРА

#### 5.1.ОСНОВНАЯ

* + 1. ИСО/МЭК 12207 – 95 «Информационная технология. Процессы жизненного цикла программных средств» или ISO/IEC 12207 (ISO – InternationalOrganizationofStandardization)

5.1.2. ГОСТ 34.601–90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы, стадии создания.

* + 1. Официальный сайт компании IBM<http://www.ibm.com/developerworks/ru/webservices/newto/>

Ibm.com/redbooks

5.2. Дополнительная

5.2.1. Бобровский С. Технологии Пентагона на службе российских программистов. Программная инженерия. – СПб.: Питер, 2003. – 222с.

5.2.2. MicrosoftSolutionFramework.Дисциплина управления рисками MSF. Вер. 1.1, 2002.

# Практический раздел

ИНДИВИДУАЛЬНЫЕ ПРАКТИЧЕСКИЕ РАБОТЫ, ИХ ХАРАКТЕРИСТИКА

### Указания по выбору варианта

Рабочей программой дисциплины «Современные средства проектирования информационных систем» предусмотрено выполнение двух индивидуальных практических работ.

Выбор вариантов задания осуществляется студентом самостоятельно на основании двух последних цифр номера зачетной книжки. Соответствующий вариант работы выбирается по правилу: предпоследняя цифра в номере зачетки + 1 определяет номер задания; последняя цифра в номере зачетки 0-1 – подвариант a) выбранного первой цифрой задания, последняя цифра 2-4 – подвариант b) выбранного первой цифрой задания, последняя цифра 5-6 – подвариант c) выбранного первой цифрой задания, последняя цифра 8-9 – подвариант d) выбранного первой цифрой задания.

### ИПР №1

#### Введение

По результатам работы студентом должен быть представлен и защищен отчет. Содержание отчета включает:

1. Введение, содержащее постановку задачи.
2. Распечатку построенных информационных диаграмм базы данных.
3. Скрипт базы данных. Reverseengineering с произведенными изменениями.
4. Вывод.

**Цель:** Ознакомление студентов с генерацией скриптов баз данных. Reverseengineering.

**Общая постановка задачи:**  Освоить основные функции пакета ErWinDataModeler, типы поддерживаемых диаграмм.

**Методические указания**: Процесс генерации физической схемы базы данных из логической модели данных называется прямым проектированием. Процесс генерации логической модели из физической базы данных называется обратным проектированием (Reverseеngineering).

Изучить стандарт IntegrationDefinitionforInformationModeling (IDEF1x). Используя методологию построения реляционных структур для среды ErWin разработать информационную диаграмму базы данных.

На основе различий между логической и физической моделью данных:

создать скрипт базы данных;

сгенерировать базу данных для выбранной СУБД;

произвести изменения в структуре сгенерированной базы данных и выполнить Reverseengineering. (Создание модели ERwin позволяет произвести обратное проектирование структуры базы данных, а затем легко перенести его в другой формат базы данных)

Сохранить отчет о генерации схемы в текстовом файле

Варианты заданий

1) Система управления погрузочно-разгрузочными работами.

2) Кадровая система офиса

3) Ежедневник с возможностью напоминания о предстоящих событиях и

телефонным справочником.

4) Каталог продукции.

5) Управление поставщиками и клиентами.

6) Комплекс по управлению документооборотом офиса.

7) Электронная картинная галерея.

8) Система приема заказов и управления складом.

9) Интернет-магазин.

10) Система предоставления услуг «в облаке»

Подварианты реализации

a) ORACLE

**b) SQL Server 2008**

**c) MS SQL 2010**

c) DB2

### ИПР №2

#### Введение

По результатам работы студентом должен быть представлен и защищен отчет. Содержание отчета включает:

1. Введение, содержащее постановку задачи.

2. Распечатку построенных информационных диаграмм классов, с заполненными основными полями и методами классами.

4. Распечатку системы связей классов.

3. Вывод.

**Цель:** Работа студентов с IDEF4. Составление диаграмм классов.

**Общая постановка задачи:** Изучить пакет RationalRose, основные функции и типы поддерживаемых диаграмм, освоить IDEF4 Object-OrientedDesignMethod.

**Методические указания**:

Программные средства, основанные на реализации структурного подхода к анализу и проектированию систем и методологий IDEF наиболее используемы при разработке логической модели системы в виде диаграммы классов. Диаграмма классов является типом диаграммы статической структуры. Она описывает структуру системы, показывая её классы, их атрибуты и операторы, а также взаимосвязи этих классов.

Создать диаграмму классов, заполнить классы основными полями и методами. Составить систему связей представленных на диаграмме классов. При создании диаграммы нужно помнить, что каждый объект представляет собой экземпляр соответствующего класса, а отношения между объектами описываются с помощью связей (links), которые являются экземплярами соответствующих отношений.

Варианты заданий

1) Система управления погрузочно-разгрузочными работами.

2) Кадровая система офиса

3) Ежедневник с возможностью напоминания о предстоящих событиях и

телефонным справочником.

4) Каталог продукции.

5) Управление поставщиками и клиентами.

6) Комплекс по управлению документооборотом офиса.

7) Электронная картинная галерея.

8) Система приема заказов и управления складом.

9) Интернет-магазин.

10) Система предоставления услуг «в облаке»

Подварианты реализации

a) ORACLE

**b) SQL Server 2008**

**c) MS SQL 2010**

c) DB2

## КОНТРОЛЬНЫЕ РАБОТЫ, ИХ ХАРАКТЕРИСТИКА

### Указания по выбору варианта

Рабочей программой дисциплины «Современные средства проектирования информационных систем» предусмотрено выполнение двух контрольных работ.

Выбор вариантов задания осуществляется студентом самостоятельно на основании двух последних цифр номера зачетной книжки. Соответствующий вариант работы выбирается по правилу: предпоследняя цифра в номере зачетки + 1 определяет номер задания; последняя цифра в номере зачетки 0-1 – подвариант a) выбранного первой цифрой задания, последняя цифра 2-4 – подвариант b) выбранного первой цифрой задания, последняя цифра 5-6 – подвариант c) выбранного первой цифрой задания, последняя цифра 8-9 – подвариант d) выбранного первой цифрой задания.

### *Контрольная работа №1*

Ознакомление с назначением и основными функциями корпоративной шины предприятия.

#### Введение

По результатам выполнения контрольной работы студентом должен быть представлен и защищен отчет по созданному средству визуального проектирования ETL-процедур и их выполнения.

**Общая постановка задачи:**

IBMWebShpereDataStage - средство визуального проектирования ETL-процедур и их выполнения

**Методические указания**:

Продукты WebSphere построены на основе связующего программного обеспечения для динамического проектированию

Средства интеграции информации позволяют эффективно объединить данные из нескольких разнотипных источников без перемещения, предотвращая образование избыточных данных. Показать возможности интеграции информации в комплексе с процессами из класса управления информацией (Information management) и решает следующие задачи:

управление преобразованием и потоками данных (ETL - Extraction, Transformation, Loading);

управление качеством данных;

управление жизненным циклом корпоративных данных.

Реализовать трансформацию и интеграцию данных на примерах применения IBMWebShpereDataStage, в т.ч. автоматизации процессов по извлечению, преобразованию и загрузке данных. Выполнить построение схем и реализацию схем преобразования данных различных источников в задании.

Охарактеризовать программную конфигурацию, которая использовалась для тестирования сценариев использования.

Варианты заданий

1) Система управления погрузочно-разгрузочными работами.

2) Кадровая система офиса

3) Система управления автотранспортом.

4) Каталог продукции.

5) Управление поставщиками и клиентами.

6) Комплекс по управлению документооборотом офиса.

7) Электронная картинная галерея.

8) Система приема заказов и управления складом.

9) Интернет-магазин.

10) Система предоставления и учета услуг «в облаке»

Подварианты реализации

a) ORACLE

**b) SQL Server 2008**

**c) MS SQL 2010**

c) DB2

### *Контрольная работа №2*

Реализация бизнес – процессов на основе ESB шины предприятия.

#### Введение

По результатам выполнения контрольной работы студентом должен быть представлен и защищен отчет.

**Общая постановка задачи:**

Реализация бизнес процессов посредством ESB шины IBMWebShpereMessageBroker.

**Методические указания**:

IBM WebSphere MessageBroker:

* использует простую модель программирования для связи и предобработки, включая надежный набор предварительно настроенных функций предобработки и способов индивидуальной настройки предобработки;
* использует лучшую инфраструктуру обмена сообщениями WebSphere MQ и поддерживает возможности преобразования с графическим отображением;
* широкие возможности администрирования и управления системами для разработанных решений

Освоить среду выполнения для поддержки потоков сообщений в Message Broker.

Выполнить разработку архитектуры и ПО на основе продукта IBM WebSphereBusiness Integration Message Broker

В процессе выполнения задания продемонстрировать возможности взаимодействия, преобразования, маршрутизации и обогащение информации в процессе её обработки в гетерогенных ИТ-средах и отразить это в отчете.

Возможность применения IBM WebSphere MessageBroker для построения платформо-независимых решений, основанных на передаче сообщений

Варианты заданий

1) Система управления погрузочно-разгрузочными работами.

2) Кадровая система офиса

3) Система управления автотранспортом.

4) Каталог продукции.

5) Управление поставщиками и клиентами.

6) Комплекс по управлению документооборотом офиса.

7) Электронная картинная галерея.

8) Система приема заказов и управления складом.

9) Интернет-магазин.

10) Система предоставления и учета услуг «в облаке»

Подварианты реализации

a) ORACLE

**b) SQL Server 2008**

**c) MS SQL 2010**

c) DB2

## Задание на курсовую работу, ее характеристика.

Курсовая работа учебным планом не предусмотрена.