ТЕМА 16

1. Инструментальные средства для создания качественного программного обеспечения
2. Обзор продуктов корпорации Rational Software, позволяющих автоматизировать процесс разработки программных средств.

В объединении с другими программными пакетами Rational Rose приобретает дополнительные возможности. В сочетании со средствами документирования Rational SoDA он может давать полное представление о проекте. Полностью интегрируясь с Microsoft Visual Studio, этот пакет позволяет получать исходный код взаимодействующих классов и строить визуальные модели по уже написанному исходному коду. Возможность интеграции со средствами управления требованиями Requisite Pro, со средствами тестирования SQA Suite, Performance Studio, со средствами конфигурационного управления ClearCase, PVCS выводит процесс разработки программного проекта на качественно новый уровень. Рассмотрим более подробно другие продукты корпорации Rational Software, которые позволяют создавать сложные программные системы быстрее, качественнее и легче.

Результатом любой деятельности, связанной с разработкой ПО, является документ или отчет заранее установленного образца. На решение всех проблем с документооборотом направлен инструмент Rational ***SoDA***. Его основная обязанность – подготовить отчет по заранее установленному шаблону. Данные для отчета берутся из любой программы Rational. Например, необходимо получить готовый документ по имеющейся модели в Rational Rose. SoDA позволит сгенерировать подобный отчет в считанные минуты, не упустив при этом ни одной детали. Извлекая из моделей спецификации и комментарии, SoDA в автоматическом режиме строит результат в виде документа в формате MS Word. Фактически SoDA является макросом для MS Word. Система вызовов и меню интегрирована в Word и позволяет генерировать шаблоны на базе имеющихся файлов. SoDA допускает к использованию как стандартные шаблоны, так и созданные пользователем при помощи специального Wizard, также встроенного в систему меню Word. SoDA может строить отчеты по результатам работы ряда продуктов: Rational Rose, Rose RealTime, Requisite PRO, ClearCase, TeamTest.

Продукт ***RequisitePro*** – это удобный инструмент для ввода и управления требованиями, который может использоваться всеми участниками команды. Requisite PRO позволяет в наглядной форме получать, выводить, структурировать наборы вводимых требований. Для каждого требования поддерживается набор атрибутов, позволяющий эффективно управлять проектом на основе задания иерархий требований, установки их приоритетов, сортировки, назначения требований конкретным исполнителям.

Наборы атрибутов для требований могут быть расширены пользователем по его усмотрению, имеются возможности визуально определять схожие требования в рамках одного или нескольких проектов. Это позволяет применять готовые апробированные решения в новом проекте. Возможность задания связей между требованиями помогает проследить, какие требования следует подвергнуть анализу или пересмотру при модификации какого-либо другого требования или атрибута. Для каждого требования хранится его история, позволяющая отследить, какие изменения были внесены в требование, кем, когда и почему. Преимущества от управления требованиями с помощью RequisitePRO увеличиваются экспоненциально при использовании его всей командой разработчиков. RequisitePRO упрощает общение между разработчиками путем предоставления общего доступа ко всем требованиям проекта или к его части. Все документы и данные, относящиеся к требованиям, централизованно организуются при помощи RequisitePRO. Требования заказчика, дизайн подсистем, сценарии, функциональные и нефункциональные спецификации и планы тестирования распределяются и связываются таким образом, чтобы максимально облегчить управление проектом.

Следующее решение от Rational – это средство управления запросами на изменение ***ClearQuest***. Поскольку проект постоянно меняется, руководителям и менеджерам необходимо знать статистику того, что менялось, кем и в какое время. Желательно, чтобы эта информация была представлена в удобном для пользователя виде. ClearQuest является мощным средством управления запросами на изменение, специально разработанным с учетом динамической и сложной структуры процесса разработки ПО. ClearQuest отслеживает и управляет любым типом действий, приводящих к изменениям в течение всего жизненного цикла продукта, помогая тем самым создавать качественное ПО. Основные задачи, решаемые с помощью ClearQuest:

- управлять изменениями, возникающими в ходе процесса разработки ПО;
- оптимизировать путь прохождения запросов на изменения, а также связанные с ним формы и процедуры;
- поддерживать через Internet связь внутри команд, разделенных территориально;
- внедрить надежный и проверенный процесс CRM, либо изменить уже существующий процесс для удовлетворения специфическим требованиям;
- визуально анализировать полученный проект с помощью богатых возможностей графического представления информации и отчетов;
- нтегрироваться со средствами конфигурационного управления, такими как Rational’s ClearCase, позволяя создавать связи между запросами на изменение и развитие кода.

В качестве положительных черт данного продукта можно отметить его адаптивность, масштабируемость, простоту в администрировании и

использовании. В работе с ClearQuest каждый участник проекта может заходить в базу и определять собственные запросы. Запросы на изменения проходят цикл из нескольких состояний, начиная с подачи и заканчивая их разрешением. Например, только что поданный запрос находится в состоянии “Подан”. После передачи запроса сотруднику он переходит в состояние “Назначен”. Начало работы над запросом переводит его в “Открытое” состояние, и вся команда может видеть, что кто-то обрабатывает запрос. Наконец, когда запрос проверен и закрыт, он проходит соответственно стадии “Проверка” и “Закрыт”.

Разработчик постоянно сталкивается с проблемой необходимости увеличения производительности собственного приложения в сжатые сроки и с максимально возможной эффективностью. ***Rational Quantify*** – это простое, но в то же время мощное и гибкое средство учета производительности приложений, которое используется для сбора и анализа информации о производительности созданного программного продукта. Quantify генерирует в табличной форме список всех вызываемых в процессе работы приложения функций, указывая временные характеристики каждой из них. Quantify предоставляет полную статистическую выкладку по всем вызовам, невзирая на размеры тестируемого приложения и время его тестирования. Сбор данных осуществляется посредством технологии OCI (Object Code Insertion). Суть способа состоит в подсчете всех машинных циклов путем вставки счетчиков в код для каждого функционального блока тестируемой программы (все циклы приложения просчитываются реально, а не при помощи произвольных выборок, как в большинстве пакетов тестирования). Уникальность данного подхода заключается в том, что, во-первых, тестируется не только сам исходный код, но и все используемые компоненты, (например: библиотеки DLL, системные вызовы), а во-вторых, для подобного анализа необязательно иметь исходные тексты тестируемого приложения (правда, в этом случае нет возможности отслеживать внутренние вызовы). Статистическая информация по вызовам может быть перенесена в Microsoft Excel, где можно построить графики и сводные таблицы для разных запусков программы.

Тестируемое приложение можно перекомпилировать и запустить повторно, при этом Quantify способен запомнить все предыдущие вызовы и дать сравнительную оценку.

Продукт ***Purify*** направлен на разрешение всех проблем, связанных с утечками памяти и Run-time ошибками. Purify борется с тем, чтобы программные продукты не замыкали на себя во время работы все системные ресурсы без большой на то необходимости. Возникновение подобного рода ошибок достаточно трудно отследить стандартными средствами, имеющимися в арсенале разработчика.

В общих чертах работа Purify сводится к выводу детальной статистики об использовании памяти приложением. Программа собирает данные о любых потерях в памяти. К ним можно отнести и невозвращение блока, и неиспользование указателей, и остановку исполнения программы с выводом

состояния среды при возникновении ошибки run-time. Purify предоставляет возможность разработчику не только видеть состояние исполнения (предупреждения, ошибки), но и переходить к соответствующему исходному тексту. Такая возможность существует только для внутренних вызовов, поскольку исходные тексты динамических библиотек пока программистам недоступны.

Рассмотрим продукт ***Rational Pure Coverage***. Он позволяет разработчикам доводить программы до состояния абсолютной эффективности, освободив от ошибок. Назначение продукта – это выявление участков кода, пропущенных при тестировании приложения. Pure Coverage собирает статистику о тех участках программы, которые во время тестирования не были выполнены (пройдены). Дополнительно программа считает активно исполнявшиеся строки. В результате разработчик может оценить не просто, сколько раз вызывалась та или иная функция, а сколько раз исполнилась каждая строка, составляющая ее. Имея подобную статистику, можно выявить не исполнившиеся строки и проанализировать причину, по которой они не получили управления. Подобные строки Pure Coverage подсвечивает красным цветом, четко указывая на наличие черных дыр в программе в виде неоттестированного кода.

Следующие три программы направлены на высокоуровневое тестирование, такое как: тестирование интерфейса, нагрузочное тестирование приложений клиент-сервер. Кроме того, каждая из программ способна создавать специальные скрипты для последующего повторного использования, например, чтобы узнать, как сказались на функциональности изменения, внесенные разработчиком в программу.

Программа ***Visual Test*** является автоматизированным инструментом тестирования, воплощает в себе гибкость, мощь и имеет интерфейс, сходный с VisualStudio от компании Microsoft. Продукт позволяет проводить любое тестирование, начиная от 32-битного Windows-приложения, компонентов ActiveX, DLL, сервера автоматизации OLE (OLE Automation server) или приложения на основе Web. Visual Test дает возможность создавать поддерживаемые, расширяемые и пригодные для повторного применения компоненты тестирования, которые можно приспосабливать ко многим версиям и после некоторого планирования ко многим проектам. Основу гибкости и мощи Visual Test составляет производный от Visual Basic расширенный язык программирования TestBasic. Он включает в себя множество специфических для теста функций, специальных конструкций для облегчения тестирования и простого доступа к Windows API и открытой архитектуре.

Rational Robot – средство функционального тестирования, базирующееся на объектно-ориентированной технологии, что позволяет существенно превзойти традиционные средства тестирования графического интерфейса. Это связано с тем, что здесь тестируются сотни и тысячи свойств всех объектов приложения (даже скрытых), как вместе, так и каждого в отдельности. Программа способна работать в двух режимах:

автоматическом и ручном. В ручном режиме пользователь задает на специальном языке сценарий тестирования (скрипт), в автоматическом – пользователь тестирует приложение, а Robot автоматически генерирует необходимый скрипт для дальнейшего повторного тестирования.

Скрипты, создаваемые в Rational Robot, обеспечивают поиск ошибок в приложении, оставаясь виртуально независимыми от внесенных изменений и платформы. Объектное тестирование обеспечивает быстрое создание скриптов, которые в дальнейшем можно легко изменять, создавать заново и воспроизводить. Rational Robot поддерживает широкий спектр языков программирования и ERP-решений, позволяет редактировать, отлаживать и настраивать скрипты. Кроме того, допускает тестирование сложных систем клиент/сервер на платформе Windows.

LoadTest – средство автоматизированного тестирования характеристик распределенных сетевых приложений на платформах Windows и Unix. Тесты производительности выполняются с помощью программы LoadTest. При этом тестировании типично используется нагрузка сервера большим количеством виртуальных пользователей. Например, можно установить таймер для одного виртуального пользователя, чтобы определить, сколько времени займет выполнение запроса, когда тысячи других виртуальных пользователей посылают запросы на тот же самый сервер в то же самое время. Термин “тесты производительности” включает нагрузочные, стрессовые, конкурирующие и конфигурационные тесты. Совокупность этих тестов позволяет ускорить цикл тестирования производительности и достигнуть значимых и точных результатов.

Нагрузочное тестирование с использованием LoadTest выполняется тогда, когда нужно определить время отклика серверов или клиентских приложений при изменяющейся нагрузке. Также оно используется, когда нужно вычислить, какое максимальное количество транзакций может выполнить сервер за определенный временной отрезок. Если клиент/серверная система использует распределенную архитектуру или средства балансировки нагрузки, нагрузочное тестирование может быть использовано для того, чтобы проверить правильность выбранных методов для балансирования или конструирования системы. Нагрузочное тестирование выполняется как с использованием режима только виртуальных пользователей, когда измеряется время отклика серверной части или комбинации виртуальных пользователей, так и пользователей графического интерфейса для измерения времени отклика системы для конкретного клиентского приложения. Таким образом, при использовании данной программы становится возможным проверять на производительность любую систему клиент/сервер.

Теперь осталось описать программный продукт, который может использоваться на всех этапах проекта: от идеи и до внедрения. Таким продуктом является ***Rational ClearCase*** – средство конфигурационного и версионного контроля, которое сохраняет всю историю каждого файла проекта с целью возможного сравнения изменений, включая миграцию

файлов между независимыми проектами. ClearCase рекомендован как средство контроля для командной разработки. Являясь высоко масштабируемым приложением клиент-сервер, ClearCase объединяет всех участников проекта единой средой, хранящей всю возможную информацию, относящуюся к проекту, позволяя получать последние версии редактируемых файлов. Посредством ClearCase команда разработчиков может ускорить циклы разработки, убедиться в точности релизов, создавая новые, надежные в эксплуатации продукты, а также дорабатывать и поддерживать ранее реализованные продукты, организовывать эффективный процесс разработки. Все это осуществляется без изменения среды, инструментальных средств и подхода к работе. Продукт обеспечивает управление версиями исходных текстов, библиотек на протяжении всего жизненного цикла проекта, позволяя тем самым вернуться к любой версии редактируемого файла и откорректировать его, создав новую версию. Каждый участник проекта может иметь доступ к запрашиваемой части проекта. Для достижения подобного эффекта ClearCase использует мощную систему настраиваемых фильтров, скрывающих ненужную информацию. Система видов значительно отличает ClearCase от продуктов конкурирующих фирм, поскольку позволяет осуществить параллельную разработку, извлекая из проекта определенную его часть. После внесения изменений все части возвращаются в проект, а ClearCase осуществит при этом автоматическое слияние версий. В дополнение к описанным возможностям ClearCase позволяет объединять географически удаленные команды разработчиков посредством специального модуля MultiSite, осуществляющего передачу текущего состояния проекта на указанный сайт.

В условиях бурно развивающейся и подверженной изменениям индустрии информационных технологий становится все сложнее и сложнее давать оценку программному продукту как чему-то независимому, вырванному из общего контекста разработки. Поэтому принимается во внимание степень поддержки данного продукта компаниями, создающими средства разработки. В частности, продукт версионного контроля не может быть функционально полным без определенных механизмов интеграции со средствами разработки, генераторами отчетов и др.