

УДК 004.4'22, 004.4'24

ТЕХНОЛОГИЯ УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ОРБИТАЛЬНЫХ ИСПЫТАНИЙ СПУТНИКА СВЯЗИ

Елеферов Денис Сергеевич¹, Белов Михаил Александрович²

¹Инженер, магистр;

ФГУП «Космическая связь», Центр космической связи «Дубна»,

Цех контроля и управления;

141980, Московская обл., г. Дубна, ул. Александровка, 43;

ГБОУ ВПО Международный Университет природы, общества и человека «Дубна»,

Институт системного анализа и управления;

141980, Московская обл., г. Дубна, ул. Университетская, 19

e-mail: deleferov@mail.ru.

²Кандидат технических наук, доцент Института системного анализа и управления;

ГБОУ ВПО Международный Университет природы, общества и человека «Дубна»,

Институт системного анализа и управления;

141980, Московская обл., г. Дубна, ул. Университетская, 19

e-mail: belov@uni-dubna.ru.

С развитием информационных технологий необходимо все больше каналов связи, поэтому требуется запускать спутники, работающие в новых диапазонах частот. Перед эксплуатацией спутниковых транспондеров необходимо провести тестирование их характеристик. Для проведения орбитальных испытаний спутника связи требуется разработать программный комплекс. Внедрение современных технологий управления жизненным циклом приложений позволит решить проблемы, связанные с командной разработкой данного программного обеспечения.

Ключевые слова: спутник связи, транспондер, жизненный цикл, командная разработка, планирование, система орбитального тестирования.

TECHNOLOGY OF APPLICATION LIFECYCLE MANAGEMENT FOR SATELLITE IN-ORBIT TESTS

Eleferov Denis¹, Belov Michail²

¹Engineer, Master of engineering and technology;

Russian Satellite Communication company, Center of satellite communication «Dubna»,

Monitoring and Management Department;

141980, Dubna, Moscow reg., Aleksandrovka str., 43;

Dubna International University of Nature, Society and Man,

Institute of system analysis and management;

141980, Dubna, Moscow reg., Universitetskaya str., 19;

e-mail: deleferov@rscs.ru.

²Candidate of Science in Engineering, associate professor of Institute of system analysis and management;

Dubna International University of Nature, Society and Man,

Institute of system analysis and management;

141980, Dubna, Moscow reg., Universitetskaya str., 19;

e-mail: belov@uni-dubna.ru.

It is necessary to use more communication channels because of the progress of information technologies. That is why, it is required to run satellites working in new frequency bands. It is necessary to test features before exploitation of satellite channels. The special software package is required for in-orbit test system. Implementation of the modern application lifecycle management technologies will decide many problems of team work.

Keywords: satellite, transponder, lifecycle, team work, planning, in-orbit test system.

Введение

Для автоматизации проведения радиотехнических измерений в процессе орбитальных испытаний спутника связи необходима разработка специализированного программного обеспечения.

Измерения параметров транспондеров возможны исключительно после выведения космического аппарата в точку стояния на геостационарной орбите. В ходе летных испытаний необходимо убедиться, что космический аппарат прошел этапы выведения на орбиту и позиционирование в точке геостационарной орбиты (ГСО) без особых потерь в характеристиках модуля полезной нагрузки. Итоги летных испытаний служат также отправной точкой в ходе эксплуатации космического аппарата для проверки параметров транспондеров в период активного существования.

В ходе разработки программного обеспечения для измерительно-вычислительного комплекса для проведения орбитальных испытаний транспондеров спутника связи возникают проблемы организации взаимодействия разработчиков, доведения требований заказчика к конечному разработчику, контроль версий приложений. Все это оказывает достаточное влияние на скорость разработки и качество разрабатываемого программного обеспечения. Таким образом, появляется необходимость внедрения современных технологий управления жизненным циклом приложений.

Управление разработкой программного обеспечения – особый вид управления программными проектами, в рамках которого происходит планирование, отслеживание и контроль над проектами по разработке программного обеспечения [2].

Внедрение системы управления программными проектами:

- позволит упростить взаимодействие участников группы и их действий в процессе совместной работы;
- внедрение современных *ALM (application lifecycle management)* технологий позволит повысить качество программного обеспечения, скоординировать процесс разработки, управлять развитием программного проекта на всех его стадиях [2].

Фундаментальный принцип *ALM* заключается в содействии процессам разработки программного обеспечения, которые на протяжении процессов объединяют множество ролей, и в управлении всем контентом, который создается каждой ролью в ходе работы. Члены рабочей группы в обеспечении своей деятельности по разработке опираются на пять принципов [2]. Они должны:

- участвовать в совместной работе коллектива;
- обеспечить трассируемость своего задания до исходного запроса;
- автоматизировать нетворческие, повторяющиеся задачи;
- стараться найти стратегии для непрерывного совершенствования;
- если коллектив является рассредоточенным, обеспечить надежное подключение к цепочке поставки программного обеспечения.

Можно выделить основные этапы управления ЖЦ [3]:

- определение требований и управление изменениями;
- автоматизированное тестирование;
- управление версиями и конфигурациями;
- шаблонное проектирование;
- кодогенерация;
- генерация отчетов и проектной документации;
- планирование;
- измерение качества проекта.

Обзор платформ управления жизненным циклом приложений

Перед разработкой архитектуры системы управления жизненным циклом была проведена работа по исследованию и экспресс-анализу средств *ALM* крупных *IT*-компаний (см. табл. 1).

Таблица 1. Сравнительная характеристика *IT*-компаний

Название	<i>IBM</i>	<i>Borland</i>	<i>Telerik</i>	<i>Microsoft</i>
Определение требованиями и управление изменениями	<i>Rational</i>	<i>Caliber</i>	-	<i>Visual Studio Team Foundation Server</i>
Автоматизированное тестирование	<i>Rational</i>	<i>Silk, Optimizelt</i>	+	<i>Visual Studio Team Foundation Server</i>
Управление версиями и конфигурациями	<i>Rational</i>	<i>Star Team</i>	-	<i>Visual Studio Team Foundation Server</i>
Шаблонное проектирование	<i>Rational</i>	<i>Together</i>	-	-
Кодогенерация	<i>Rational</i>	<i>DevPartner</i>	+	<i>Visual Studio Team Foundation Server</i>
Генерация отчетов и проектной документации	<i>Rational</i>	-	-	<i>Visual Studio Team Foundation Server</i>
Планирование	<i>Rational</i>	-	-	<i>Visual Studio Team Foundation Server</i>
Измерение качества проекта	<i>Rational</i>	<i>Silk</i>	-	-

Borland

Продукты *Borland* почти полностью охватывают все этапы описанного выше жизненного цикла программного обеспечения. Обработку требований на базе централизованного хранилища, их фиксацию, сопровождение, отслеживание изменений и взаимосвязей (в частности, связей с пользовательским интерфейсом), регистрацию истории версий и ряд других возможностей обеспечивает линейка *Caliber*. К положительным особенностям *Caliber* следует отнести хранение в общем хранилище всех сопроводительных документов и их связь с базой требований, что упрощает взаимную синхронизацию [2].

За построение моделей и проектирование архитектуры системы отвечает продукт *Borland Together*, который, начиная с версии *Delphi 2006*, встроен в среду, что подтверждает линию корпорации на полную интеграцию своих решений. В качестве средств разработки программистам наряду с *Delphi* доступны *J Builder* (среда программирования на языке *Java*), а также встроенные в среду *Delphi* система программирования *C++ Builder* (она дополнилась системой анализа ошибок времени выполнения *CodeGuard*) и средства создания в рамках *Delphi* программ для платформы *.NET* на языках *C#* (*C# Builder*) и даже *VB.NET* [2].

Процесс управления и контроля за версиями, конфигурациями и изменениями обеспечивается системой коллективной работы *StarTeam*. Ее клиентская часть интегрирована в *Delphi*, и ее контролю поддаются не только исходные файлы с кодом, но и проекты, а также группы проектов. Для этого в контекстном меню соответствующих пунктов «менеджера проектов» имеется раздел *StarTeam* с соответствующими возможностями. Состыкован со *StarTeam* и «менеджер историй» *Delphi*, что позволяет использовать возможности среды контроля версий для учета изменений как на локальном компьютере, так и в рабочей группе [2].

Тестированием и оптимизацией созданных приложений занимается семейство серверных решений *Optimizelt*, хранением данных – СУБД *InterBase*, а за развертывание, масштабирование и надежную работу ответственны: сервер приложений *Borland Enterprise Server*; технология *Janeva* упрощающая стыковку приложений *.NET*, *J2EE* и *CORBA*; мобильная *Java* – СУБД *JDataStore* [2].

К преимуществам использования *Borland* для *ALM* можно отнести совместимость с продуктами сторонних разработчиков.

Есть и ярко выраженный недостаток: почти все этапы жизненного цикла решаются с помощью разных линеек продукции. Кроме того, несмотря на совместимость, основная среда разработки является *Delphi*, что не устраивает многих разработчиков.

Microsoft

Компания *Microsoft* предлагает готовое решение для *ALM*: *Microsoft Visual Studio Team Foundation Server. Team Foundation Server* (в дальнейшем *TFS*) работает в трехуровневой архитектуре [4]:

- Клиентский уровень – используется для создания и управления проектом, для доступа к хранимым элементам проекта. *TFS* не включает в себя пользовательский интерфейс на данном уровне, а вместо этого он предоставляет веб-сервисы, которые используют конечные клиентские приложения (например, *Visual Studio Team System*) для интеграции функциональности *TFS*.
- Уровень приложений – включает в себя веб-сервисы, используемые на клиентском уровне, веб-портал, который используется как центральная точка связи для проектов, и хранилище документов.
- Уровень данных – хранятся на *Microsoft SQL Server*.

TFS является законченным решением, которое охватывает почти полностью все этапы *ALM*, что является несомненным преимуществом данного продукта. Кроме того, базовой средой разработки для *TFS* является *Visual Studio*, а разработка программного обеспечения для орбитальных испытаний происходит именно в этой среде.

IBM

Линейка продуктов *IBM Rational* имеет огромное количество продуктов и услуг, которые в совокупности охватывают все этапы *ALM*. К преимуществам данной продукции можно отнести:

- Большой спектр конечных продуктов, а, следовательно, большой выбор технологий для решения конкретной задачи.
- Совместимость с большинством из известных интегрированных сред разработки.
- Наибольшая функциональность конечных решений.

В данной работе будет предложена технология управления жизненным циклом средствами *IBM Rational* во многом благодаря обширной функциональности данного продукта.

Функциональная модель системы управления

Для управления жизненным программным обеспечением орбитальных испытаний были выделены функции на всех этапах жизненного цикла.

Определение и управление требованиями:

- Создание и редактирование заданий (*Rational Team Concert*).
- Автоматическое создание ссылок на артефакты (*Rational Team Concert*).
- Быстрый обмен сообщениями (*Rational Team Concert*).
- Создание тегов для заданий (*Rational Team Concert*).
- Утверждение заданий (*Rational Team Concert*).
- Печать заданий (*Rational Team Concert*).
- Поиск дублирующих заданий (*Rational Team Concert*).
- Отслеживание изменений заданий (*Rational Team Concert*).
- Запрос заданий (*Rational Team Concert*).

Автоматизированное тестирование:

- Тестирование на основе рисков (*Rational Quality Manager*).
- Использование шаблонов тестирования (*Rational Quality Manager*).

- Тестирование на основе моделей (*Rational Rhapsody Developer*).
Управление версиями и конфигурациями:
- Просмотр результатов выполненных компоновок (*Rational Team Concert*).
- Запрос на выполнение компоновки (*Rational Team Concert*).
- Персональные компоновки своего кода (*Rational Team Concert*).
- Контроль версий (*Rational CrearQuest*).
- Развертывание на разных платформах исполнения (*Rational Team Concert*).

Шаблонное проектирование: использование шаблонов и моделей для формирования отчетов, тестирования, компоновки, генерации кода (*Rational Rhapsody Developer*, *Rational Quality Manager*, *Rational Team Concert*).

Кодогенерация:

- Управление исходным кодом *C#* (*Rational Team Concert*).
 - Изоляция задач для самостоятельной разработки (*Rational Team Concert*).
 - Визуализация кода *C#* (*Rational Rhapsody Developer*).
 - Генерация кода *C#* на основе моделей (*Rational Rhapsody Developer*).
Генерация отчетов и проектной документации:
 - Отчеты о состояниях компоновок (*Rational Team Concert*).
 - Отчеты об изменениях (*Rational Team Concert*).
 - Отчеты по исходному коду (*Rational Team Concert*).
 - Хранение проектной документации и отчетов (*Rational Team Concert*).
 - Конфигурирование отчетов на основе шаблонов (*Rational Team Concert*).
- Планирование:
- Планирование итераций (*Rational Team Concert*).
 - Планирование выпусков (*Rational Team Concert*).
 - Планирование ежедневной работы разработчиков (*Rational Team Concert*).
 - Проекция планов на шкалу времени (*Rational Team Concert*).
 - Планирование проведения орбитальных испытаний (*Rational Team Concert*).
 - Информация о прогрессе на основе отчетов (*Rational Team Concert*).

Измерение качество проекта:

- Контроль за показателями (*Rational CrearQuest*).
 - Выявление дефектов (*Rational Quality Manager*).
- Общие функции:
- Контроль доступа (*Rational Team Concert*, *Rational CrearQuest*).
 - Масштабирование (*Rational Team Concert*, *Rational CrearQuest*).
 - Совместимость с *Microsoft Visual Studio* (*Rational Team Concert*).
 - Интегрируемость с другими продуктами (*Rational Team Concert* – интегрируется с *Rational CrearQuest*, *Rational Rhapsody Developer*, *Rational Quality Manager*).
 - Поддержка *Windows* (*Rational Team Concert*, *Rational CrearQuest*, *Rational Rhapsody Developer*, *Rational Quality Manager*).

Архитектура системы управления

На основе приведенной функциональной модели был выбран конечный набор продуктов линейки *IBM Rational*.

Rational ClearQuest

Этот продукт обеспечивает отслеживание изменений, автоматизацию процессов, отчетность и контроль жизненного цикла для большей прозрачности и управляемости жизненного цикла разработки ПО. К достоинствам данного ПО можно отнести [1]:

- Гибкие, настраиваемые и автоматизированные рабочие потоки обеспечивают использование воспроизводимых, исполнимых и предсказуемых процессов.
- Готовый шаблон управления жизненным циклом приложений (*Application Lifecycle management, ALM*) создает прочный фундамент для применения лучших практических методов *ALM*.
- Отчетность и контроль за показателями в режиме реального времени повышают прозрачность проектов и упрощают принятие решений.
- Контроль доступа, электронные подписи, воспроизводимые процессы и журналы аудита упрощают управление соответствием требованиям.
- Масштабирование в соответствии с размерами коллектива – от небольших рабочих групп до распределенных групп на крупных предприятиях – позволяет удовлетворять растущие нужды компании.
- Интеграция с инструментами управления требованиями, портфелями, разработкой, сборкой, тестированием и внедрением способствует быстрому реагированию на изменения.
- Поддерживаемые операционные системы: *Linux, Windows*.

Rational Rhapsody Developer

Программное обеспечение *IBM Rational Rhapsody Developer* представляет собой среду визуальной разработки, предназначенную для разработки встроенного, доступного в режиме реального времени или технического прикладного программного обеспечения на основе языка *Unified Modeling Language (UML)*. Применение этого решения позволяет повысить производительность на всех этапах жизненного цикла разработки встроенного программного обеспечения – от описания требований до реализации, тестирования и внедрения. С ним вы сможете ускорить выход на рынок новых продуктов, имея в своем распоряжении автоматические процессы разработки программного обеспечения, возможности проверки на ранних этапах и расширенного тестирования моделей [3].

Rational Rhapsody Developer предоставляет следующие возможности:

- Полная генерация кода встроенных приложений – генерирует поведенческий код для диаграмм состояний и объекты построения для автоматизации разработки приложений [1].
- Имитационное моделирование и тестирование на основе моделей, а также визуальная разработка – обеспечивает возможности выполнения модели для проверки проектов на раннем этапе [1].
- Прослеживаемость требований – позволяет сохранять требования в элементах проекта и контрольных примерах, которые в дальнейшем будут служить информацией прослеживания в модели [1].
- Совместная работа в группах – облегчает сотрудничество внутри групп, позволяя управлять сложностью разработки согласующихся проектов в разных средах [1].
- Поддержка жизненного цикла и дополнительное программное обеспечение – интегрируется с другими продуктами *IBM Rational* для разработки полного жизненного цикла продукта. Кроме того, вы можете расширить функции *Rational Rhapsody Developer*, добавив предлагаемые отдельно программные продукты [1].

Rational Quality Manager

Rational Quality Manager предоставляет возможности для планирования, выполнения и отслеживания состояния ручного и автоматизированного тестирования ПО и систем [1]:

- *IBM Rational Quality Manager* помогает наладить взаимодействие между рабочими группами среднего или большого размера, позволяя им легко обмениваться информацией, использовать средства автоматизации для ускорения реализации проектов и составлять отчеты о

контрольных показателей проектов, обеспечивающие принятие обоснованных решений о выпуске программного обеспечения [1].

- Тестирование на основе рисков позволяет тестировать компоненты и функции с учетом приоритета и вероятности или влияния сбоя в соответствии с лучшими практическими методами работы в области управления рисками [1].
- Функция выявления дубликатов дефектов позволяет сканировать существующие дефекты, связанные с различными объектами, и составлять списки предполагаемых дубликатов с целью минимизации дублирующих действий персонала [1].
- Расширенные и настраиваемые отчеты отвечают потребностям и задачам бизнес-аналитиков и руководителей [1].
- Встроенные средства отчетности о показателях окупаемости инвестиций для лабораторий тестирования и активов тестирования соответствуют подходу *Measured Capability Improvement Framework (MCIF)* [1].
- План тестирования жизненного цикла, в котором определены роли, процессы и владельцы результатов работы, позволяет автоматизировать рабочие потоки и движение активов; возможно создание материалов тестирования вручную с использованием расширенного текстового формата, встроенных изображений и подсказок при вводе и проверке данных [1].
- Позволяет индивидуально подбирать содержимое и представлять данные с помощью настраиваемых информационных панелей [1].
- Обеспечивает взаимодействие между распределенными группами с помощью интерфейса на основе *Web 2.0*.
- Обеспечивает общий доступ к активам и возможность их отслеживания на протяжении всего жизненного цикла.
- Обеспечивает точное определение и правильное выполнение тестов.
- Помогает пользователям оптимизировать нагрузку на среду тестирования за счет возможности составления расписаний тестирования.
- Предоставляет основные средства управления лабораторным тестированием, которые помогают отслеживать и планировать физические и виртуальные активы лаборатории, а также выполнять тестирование с их использованием.
- В центральном хранилище сохраняется хронология использования активов и шаблонов тестирования с указанием версий, что обеспечивает их многократное применение.
- Обеспечивает поддержку рабочих групп среднего и большого размера (до 250 человек).
- Выполняет автоматизированный сбор данных и формирует всесторонние отчеты с возможностью фильтрации.

Rational Team Concert

IBM Rational Team Concert – это решение для управления жизненным циклом ПО, обеспечивающее контекстную коллективную работу для распределенных коллективов в реальном времени. Продукт *Rational Team Concert*, разработанный на платформе *IBM Rational Jazz*, предоставляет средства настройки, управления и применения процессов, которые поддерживает всю среду доставки ПО [3].

Данное решение предоставляет бесплатное серверное ПО и гибкие ценовые модели. Кроме того, оно обеспечивает поддержку различных платформ и вариантов лицензирования на основе ролей в едином выпуске, что позволяет развертывать отдельные компоненты по мере необходимости [3].

Кроме того, *Rational Team Concert* поддерживает функции коллективного управления изменениями. Эти функции доступны по отдельности и могут быть интегрированы с популярными системами контроля исходного кода [3].

Rational Team Concert обеспечивает совместную работу коллективов разработчиков для более быстрой доставки ПО [3]:

- Расширяет возможности взаимодействия коллективов благодаря встроенным функциям, включая управление рабочими заданиями, компоновкой и конфигурацией ПО.

- Обеспечивает высокую прозрачность операций и состояния проекта на основе многоуровневых сводных панелей и отчетов.
- Упрощает планирование и выполнение гибких или формальных проектов на основе инструментов планирования и шаблонов. Согласованные процессы помогают повысить качество ПО.
- Повышает производительность за счет улучшенных средств контроля исходного кода для распределенных коллективов разработчиков.

Rational Team Concert – решает почти все задачи управления ЖЦ, совместим с *Microsoft Visual Studio*, а также при необходимости интегрируется с другими продуктами линейки *IBM Rational*, а именно *Rational Quality Manager*, *Rational Rhapsody Developer*, *Rational ClearQuest*.

В качестве основного программного продукта для решения задач управления жизненным циклом был выбран *IBM Rational Team Concert*.

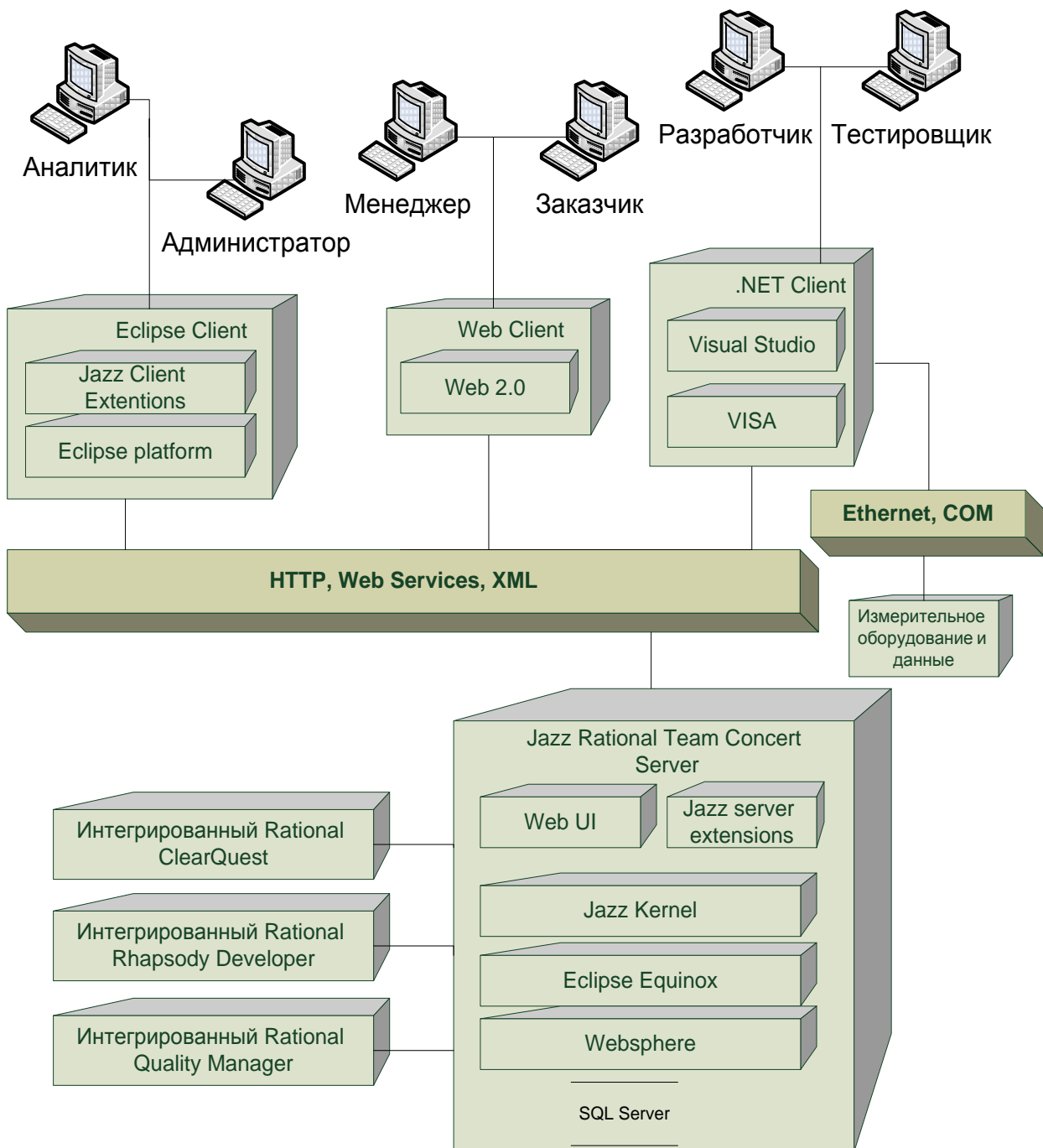


Рис. 1. Архитектура системы управления

Заключение

В результате была спроектирована система управления жизненным циклом программного обеспечения для проведения орбитальных испытаний спутника связи. Также было произведено внедрение продуктов *IBM Rational Team Concert, Rational ClearQuest, Rational Rhapsody Developer, Rational Quality Manager*.

Список литературы

1. IBM. [Электронный ресурс]. URL: <http://ibm.com> (дата обращения 10.03.2013).
2. ALM. [Электронный ресурс]. URL: <http://src-code.net> (дата обращения 12.02.2013).
3. ALM. [Электронный ресурс]. URL: <http://www.interface.ru> (дата обращения 17.05.2013).
4. Свободная энциклопедия. [Электронный ресурс]. URL: <http://ru.wikipedia.org> (дата обращения 10.04.2013).