



Bruce Powell Douglass  
Mats Gothe

# Инструментальные средства IBM Rational для системного и программного инжиниринга

Умная планета начинается с внедрения более умных продуктов. Инженерные команды, которые проводят инновации, как никогда нуждаются в новых возможностях для организации совместной работы с тем, чтобы разрабатывать и поставлять продукты в точно отведенные сроки, с должным качеством и в рамках выделенных бюджетов. Набор инструментальных средств IBM Rational для системного и программного инжиниринга как раз и предоставляет командам, разрабатывающим такие системы, возможность работать интегрировано, как единое целое.

Процесс разработки сложных систем обычно подразделяется на множество категорий, например, системное проектирование, управление проектом, разработка программного обеспечения, управление качеством. Эти категории взаимосвязаны между собой и на протяжении всего жизненного цикла разработки систем решают собственные задачи и создают собственные рабочие артефакты. Предлагаемое решение для системного и программного инжиниринга является тем решением для управления жизненным циклом разработки, которое поддерживает решение взаимосвязанных задач такого рода и обеспечивает сопряжение между различными рабочими артефактами жизненного цикла. Более того, предлагаемое решение устанавливает последовательность действий при разработке систем (workflow) и обеспечивает возможность управления задачами с целью эффективного выполнения проекта.

Данная публикация предназначена для внимания команд разработчиков, которым необходимо иметь интегрированное решение для управления жизненным циклом разработки инженерных систем и программного обеспечения.

# Комплексное решение для управления жизненным циклом разработки систем

При отсутствии интегрированного подхода, охватывающего все этапы жизненного цикла разработки инженерных систем и программного обеспечения, команды разработчиков вынуждены работать изолированно и нескоординировано, что, безусловно, отрицательно сказывается на эффективности выполнения проекта. Более того, чтобы создавать более «умные» продукты, которые отвечают постоянно меняющимся запросам рынка, командам системных и программных разработчиков просто необходимо создавать условия для эффективной коллективной работы и совместного управления созданием рабочих артефактов жизненного цикла.

Набор инструментальных средств IBM Rational, изображенный на рис. 1, как раз обеспечивает интегрированное решение для управления жизненным циклом разработки.



Рис. 1. Набор инструментальных средств IBM Rational для системного и программного инжиниринга

Рассматриваемый набор средств – комплексное решение для следующих специалистов:

## ► Системные инженеры.

Предлагаемое решение обеспечивает команды проектировщиков и разработчиков систем интегрированной средой для совместной работы, позволяющей анализировать требования и контролировать разработку архитектуры, а также управлять работами, изменениями и конфигурациями. Основные используемые инструменты, предназначенные для решения задач системного инжиниринга, – IBM Rational DOORS и IBM Rational Rhapsody, интегрированные с IBM Rational

Team Concert, который, в свою очередь, предназначен для создания и управления рабочими артефактами на всем протяжении их жизненного цикла. Интеграция с IBM Rational Quality Manager с самого начала проекта обеспечивает тесное взаимодействие разработчиков с командой, отвечающей за тестирование и сертификацию системы.

► **Менеджеры проектов, руководители команд разработчиков и тестировщиков.**

IBM Rational Team Concert и IBM Rational Quality Manager обеспечивают возможность управления работами и их планированием на всех этапах жизненного цикла проекта, а также обеспечивают полную прозрачность процессов благодаря автоматизации, взаимодействию, координированной работе и отчетности, как о состоянии рабочих артефактов, так и общего статуса проекта.

► **Разработчики программного обеспечения.**

Возможности продукта IBM Rational Rhapsody, интегрированного с IBM Rational Team Concert в среде разработки Eclipse, предлагают оптимальное решение для разработки программного обеспечения. Такая интеграция обеспечивает большие преимущества, объединяя основанный на моделировании подход к разработке с возможностями IBM Rational Team Concert, в части взаимодействия и координации совместных действий членов команды: контроля за структурой модели, управления заданиями, отслеживания изменениями и интеграционными сборками разрабатываемого программного обеспечения. Помимо этого, использование IBM Rational DOORS и IBM Rational Rhapsody обеспечивает глубокую трассировку между рабочими артефактами технологической цепочки как в сторону требований, так и в сторону интеграционных сборок и тестирования.

► **Тестировщики систем и программного обеспечения.**

IBM Rational Quality Manager обеспечивает среду для совместного планирования, создания и проведения тестов, причем поддерживает не только тестирование программного обеспечения, но и управление испытаниями в ходе проверки работоспособности системы и ее приемочной аттестации. IBM Rational Test Lab Manager повышает эффективность работы тестовых подразделений, а также управляет запросами и выделением необходимых ресурсов для тестирования.

## Интеграция улучшает взаимодействие

Все вышеупомянутые инструменты интегрированы с платформой IBM Rational Jazz. Настроенная на поддержку распределенных команд специалистов платформа Jazz выводит разработку инженерных систем и программного обеспечения на качественно новый уровень, делая ее более координированной, продуктивной и прозрачной. Платформу Jazz можно представить себе в виде «эластичной» структуры, которая динамически интегрирует, контролирует и синхронизирует людские ресурсы, процессы и другие активы, связанные с работой над проектом. В отличие от монолитных и закрытых решений прошлого Jazz является открытой платформой, поддерживающей подход OSLC (Open Services for Lifecycle Collaboration), направленный на улучшение взаимодействия различных инструментов. Совокупность инструментов, объединенных платформой Jazz, может предложить командам разработчиков систем и программного обеспечения значительно больший набор возможностей.

Поскольку средства интегрированы в платформу Jazz, то это обеспечивает совместную разработку, автоматизацию и отчетность на всех этапах жизненного цикла разработки систем (см. рис. 2).



**Рис. 2.** Интеграционные возможности платформы Jazz поддерживают совместную работу, автоматизацию и отчетность

В отличие от ситуаций, когда предлагается одно решение на все случаи жизни, что в нынешних условиях было бы не очень гибко, интеграционная архитектура Jazz следует подходам и стандартам «всемирной паутины» (World Wide Web), при которых любые рабочие артефакты жизненного цикла свободно связаны с ресурсами посредством ссылок, стирая тем самым привычные границы репозитория.

По этой причине принципы построения архитектуры Jazz-платформы олицетворяют собой ключевой отправной момент по уходу от традиционных методов хранения данных, принятых в прошлом. Интеграция данных между различными репозиториями в этом случае достигается за счет сопряжения артефактов при использовании интерфейсов, поддерживающих REST (Representational State Transfer). Навигация и доступ к информации может осуществляться как через web-интерфейсы, так и посредством «толстого» клиента.

Такой подход позволяет компаниям более эффективно и без «провалов» интегрировать рабочие артефакты, технологические процессы и команды разработчиков.

Подход OSLC, как ключевая составляющая Jazz-платформы, является отраслевой инициативой, направленной на обеспечение совместимости инструментов и ресурсов в разных дисциплинах, а также решений, предлагаемых различными поставщиками. Способность принимать и передавать спецификации OSLC позволяет рассматриваемому решению открыто интегрироваться в решение по управлению жизненным циклом разработки систем. Для пользователей это несет дополнительную гибкость в плане использования и интеграции инструментов от многих поставщиков с целью сбора и управления информацией. Эта возможность имеет чрезвычайную важность при создании открытости и организации взаимодействия в рамках совместной работы над проектами по разработке систем, где умные продукты должны быть интегрированы с приложениями или с операциями, имеющими отношение к информационным технологиям.

## Организация рабочего процесса при проектировании систем

Разработка и создание инженерных систем состоит из множества взаимосвязанных активностей, среди которых можно выделить системное проектирование, планирование проекта, разработку программного обеспечения, управление качеством. В рамках системного проектирования выявляются требования к системе, ее функциональности и уточняется ее архитектура, но на этом этапе разработчики еще не сосредотачиваются на сути конкретных компонентов системы – электронных, механических или программных.

Разработка программного обеспечения характеризуется тем, что на основе спецификаций, разработанных на этапе системного проектирования, реализуются программные продукты, которые будут либо загружаться в различные аппаратные составляющие системы, либо войдут в состав более сложных и умных приложений. На этом этапе также разрабатываются системы управления, цель которых – взаимодействие с умными механизмами и приложениями и управление ими.

Возможности планирования помогают составить планы, графики, регламенты проекта. Это позволяет гарантировать, что подразделения работают именно над теми задачами, которые необходимо выполнять. Стоит отметить, что помимо этого планирование помогает не только обеспечить прозрачность статуса проекта, но и отслеживать результаты разработки тех компонентов, реализация которых отдана внешним поставщикам.

Управление качеством обеспечивает непрерывную проверку работоспособности сборок программного обеспечения, планирование тестирования, контроль за реализацией требований, а также генерацию разного рода отчетов на основе метрических данных, собранных в процессе тестов, испытаний и сертификаций.

Давайте более подробно остановимся на тех задачах, которые решаются в ходе разработки инженерных систем, и рассмотрим, как набор инструментальных средств IBM Rational для системного и программного инжиниринга поддерживает этот рабочий процесс.

## Организация рабочего процесса при системном проектировании

В процессе создания сложных инженерных комплексов системный инжиниринг решает несколько основных задач:

- **Определение требований.**

Учет и формулирование потребностей всех заинтересованных лиц и объединение их в группы по сценариям использования.

- **Анализ требований.**

Уточнение и детальная разработка требований к функционалу системы.

► **Анализ функционала системы.**

Подробный анализ требований по методу «черного ящика», предусматривающий в том числе и создание исполняемых моделей, с высокой точностью воспроизводящих требования.

► **Синтез архитектуры системы.**

Прорисовка архитектуры системы с разбивкой по функциональности каждой подсистемы, определение требований к каждой из них, а также проработка интерфейсов взаимодействия между подсистемами.

Общая последовательность работ при проектировании систем показана на рис. 3.

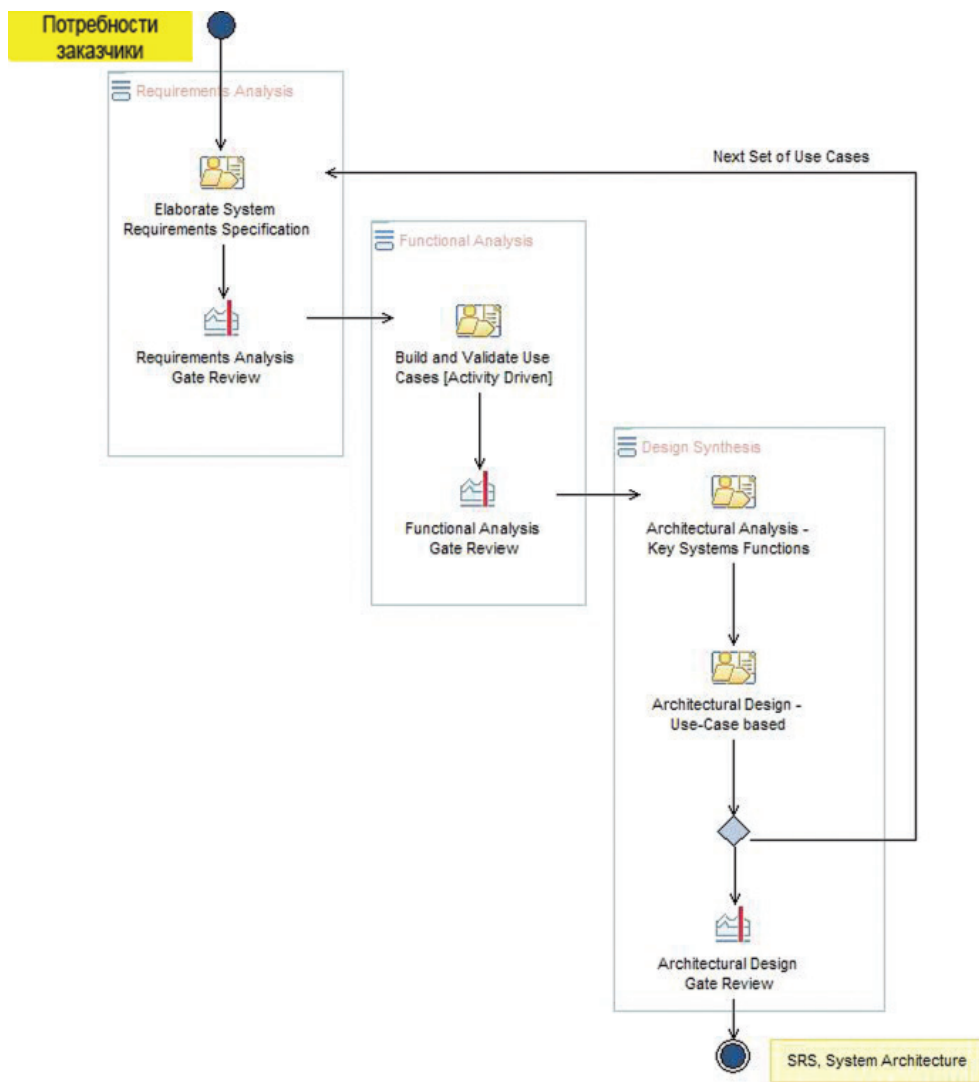


Рис. 3. Последовательность работ при системном проектировании

Идентификация требований и управление ими – есть две самые первые ключевые активности в рамках системного проектирования. С этой точки зрения под требованиями заказчика понимается совокупность потребностей всех заинтересованных в проекте лиц. В дальнейшем, в процессе анализа требований, рождаются системные требования, описывающие поведение системы и характеристики ее производительности, чтобы полностью удовлетворить запросы заказчика. Эти требования, в большинстве своем, имеют текстовый вид и принимаются всеми основными заинтересованными лицами, имеющими отношение к проекту. И хотя такие текстовые требования зачастую страдают неточностями и двусмысленностями, формирование таких пакетов требований, тем не менее, является жизненно важной необходимостью для любой организации, занимающейся разработкой систем.

Вот почему в состав рассматриваемого решения входит IBM Rational DOORS, который поддерживает формирование и управление требованиями заказчиков, и системными требованиями.

В рамках следующей активности, анализе функционала системы, осуществляется переход от описательных текстовых требований к использованию более строгого и однозначного языка, например, SysML (Systems Modelling Language), что позволяет более точно и аккуратно проводить анализ требований, имеющих отношение к функционалу и производительности. На этапе функционального анализа системы строятся точные модели всех вариантов ее использования (use-case), каждый из которых может быть проверен с помощью имитационного моделирования. Такой подход позволяет проектировщикам системы уже на самых ранних этапах проверять корректность ее поведения (например, контролировать, что получается на выходе при различных комбинациях входных сигналов), а также дает возможность всем заинтересованным лицам более осознанно участвовать в совместных обсуждениях тех требований, смысл которых, возможно, был не совсем ясен из текстовой формулировки.

В состав предлагаемого набора инструментов также входит IBM Rational Rhapsody (рис. 4), который поддерживает имитационное моделирование и обеспечивает связь (трассировку) функциональных требований с системными требованиями, которые хранятся в IBM Rational DOORS.

Широкие возможности совместной работы предлагаемого решения значительно улучшают эффективность и повышают качество разработки систем. Так например, в дополнение к возможностям моделирования вариантов использования, проектировщики могут также анализировать безопасность и надежность системы и при необходимости вносить корректировки как в текст требований, так и в планы испытания.

В процессе синтеза архитектуры группы проектировщиков работают над следующими основными задачами:

- ▶ Выполнение ряда исследований, оценок и анализов с целью выбора оптимальной архитектуры.
- ▶ Выявление в общей структуре системы архитектурных подсистем и разработка системных требований к ним.
- ▶ Корректировка результатов анализа безопасности и надежности с учетом выбранной архитектуры системы.



- ▶ Описание базовых режимов работы подсистем и определение интерфейсов их взаимодействия.
- ▶ Дальнейшая проработка и детализация требований на уровне подсистем.
- ▶ Создание правил управления системой и распределение их по подсистемам.
- ▶ Уточнение и обновление планов тестирования и испытаний системы.

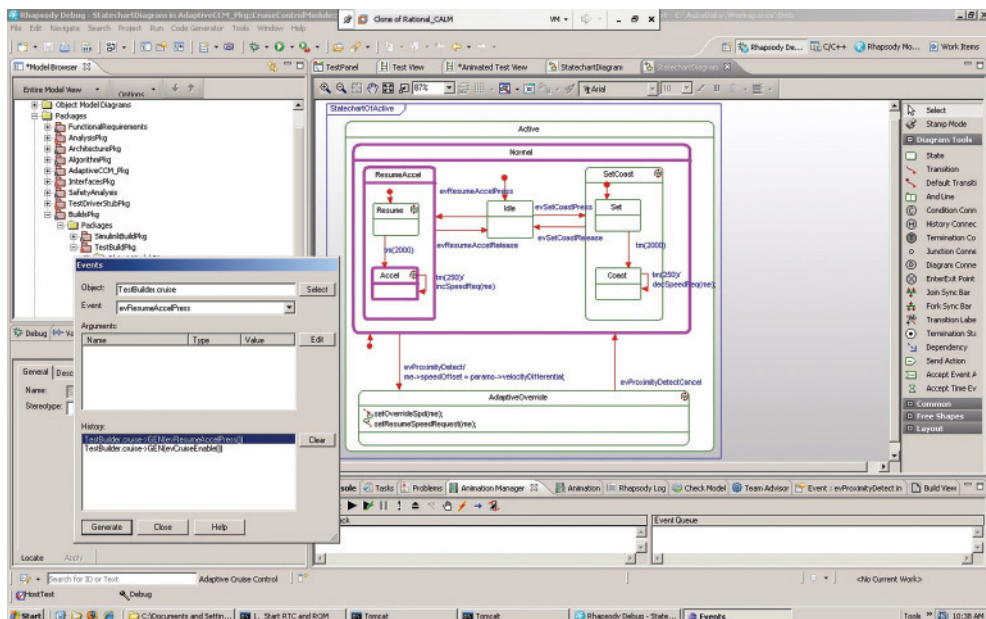


Рис. 4. Использование IBM Rational Rhapsody для анализа и валидации системных требований

## Управление системными требованиями и требованиями заказчика

Широко известный IBM Rational DOORS является приложением для управления требованиями, которое обеспечивает полный и комплексный подход к управлению требованиями в рассматриваемом наборе инструментальных средств IBM Rational для системного и программного инжиниринга. Именно этот инструмент осуществляет управление требованиями заказчиков, системными требованиями и, далее, декомпозированными требованиями к подсистемам, вовлекая в процесс совместной работы над требованиями всех заинтересованных в проекте лиц: заказчиков, системных инженеров, проектировщиков, разработчиков программного обеспечения и тестировщиков.

IBM Rational DOORS, взаимодействуя с IBM Rational Rhapsody и IBM Rational Team Concert, поддерживает выполнение стоящих перед инженерными командами задач по формированию системных требований и их «привязке» к требованиям заказчиков. При этом наличие интеграции с IBM Rational Rhapsody обеспечивает связь системных требований и с детализированными спецификациями требований для каждого варианта использования, и с исполняемыми моделями требований, как показано на рис. 5.



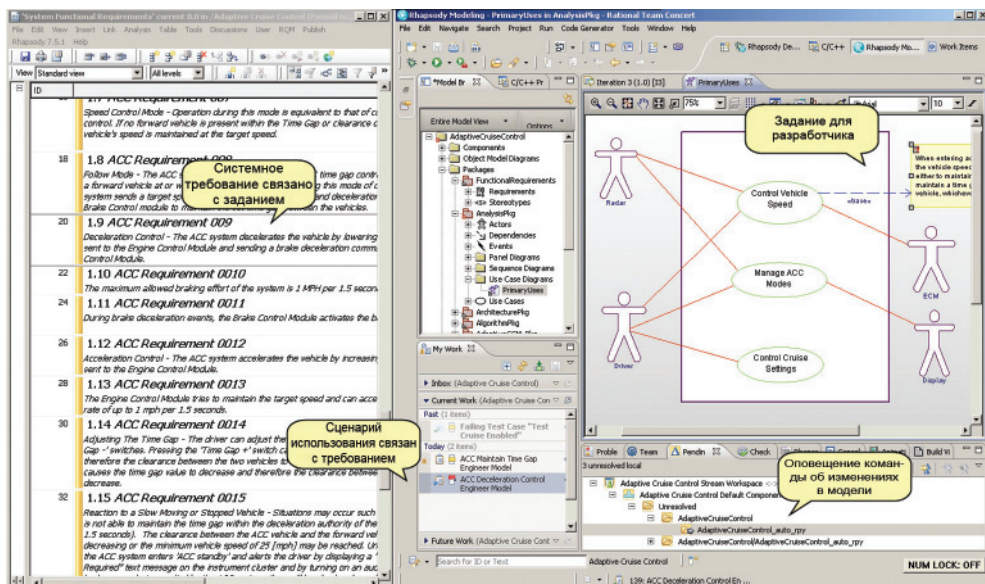


Рис. 5. Предлагаемое решение обеспечивает проектировщикам постоянный доступ к заданиям и рабочим артефактам

В свою очередь IBM Rational Team Concert (RTC) координирует задачи системного проектирования с производственными заданиями в соответствии с технологическими процессами, которые приняты в команде. Другими словами, требования, сформированные в IBM Rational DOORS, могут быть увязаны с рабочими заданиями при помощи RTC, и наличие такой связи позволяет разработчикам и заказчикам просматривать и анализировать требования в привязке к рабочим заданиям. В качестве примера такого задания можно упомянуть запрос на изменение требования, задание на разработку, устранение дефекта или любой другой принятый в компании тип единичного рабочего задания.

Поскольку системные требования являются основополагающими в проекте по разработке системы, то и управление изменениями этих требований является весьма важным моментом. IBM Rational DOORS в совокупности с RTC обеспечивает прекрасную возможность управления изменениями в требованиях. Каждый приходящий запрос на изменение требования может анализироваться в отношении текущей формулировки требования, рассматриваться через призму существующих планов разработки и проверяться на реализацию в RTC. Такой подход поможет оценить, насколько предлагаемый запрос на изменение сможет повлиять на развитие проекта в целом прежде, чем изменения будут приняты и зафиксированы в IBM Rational DOORS. Такая организация процесса внесения изменений объединяет всех людей, вовлеченных в проект: заказчиков, руководителей проекта и разработчиков.

## Планирование проекта и организация рабочего процесса

На завершающей стадии системного инжиниринга формируется набор детализированных спецификаций подсистем, который и составляет основу для планирования проекта и последующего «нисходящего» проектирования. (Следует заметить, что эти спецификации пока еще по-прежнему мультидисциплинальны.) Планирование проекта позволяет рассматривать каждое конкретное требование как отдельный пункт итерационного плана разработки и далее соотносить его с тестовым сценарием для проверки реализации (см. рис. 6). Пункты плана декомпозируются и служат основой для формирования собственных планов отдельных команд разработчиков и координации взаимозависимых задач.

Продукт IBM Rational Team Concert, входящий в рассматриваемое решение, как раз и является той средой для совместной разработки, которая сочетает в себе планирование проекта, утверждение последовательности выполняемых операций, управление рабочими заданиями, а также позволяет отслеживать текущее состояние проекта и обеспечивает отчетность.

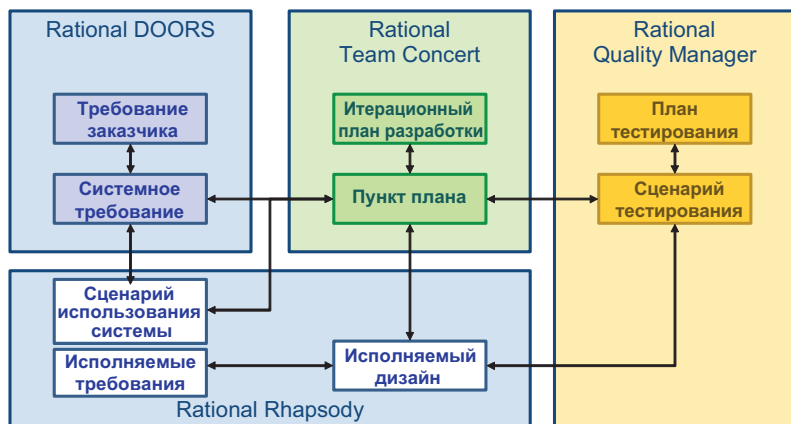


Рис. 6. Пункты плана и их связь с рабочими артефактами

При использовании Rational Team Concert команды системных инженеров и разработчиков программного обеспечения участвуют в следующих активностях:

- ▶ Управляют артефактами, необходимыми для выполнения назначенных им заданий, в рамках пунктов общего плана проекта.
- ▶ Используют панели управления заданиями для планирования индивидуальных рабочих активностей.
- ▶ Используют системные требования и сценарии применения системы для контроля за содержанием проекта, релизов и шагов итерационных планов.
- ▶ Контролируют соответствие планов команд проектировщиков и планов по тестированию системы с общим планом разработки системы.

- Участвуют в мониторинге за текущим состоянием проекта благодаря его прозрачности и совместной работе команд.

Возможности итерационного планирования в IBM Rational Team Concert показаны на рис. 7, где все единичные задания очередного шага итерации представлены в виде ранжированного иерархического списка.

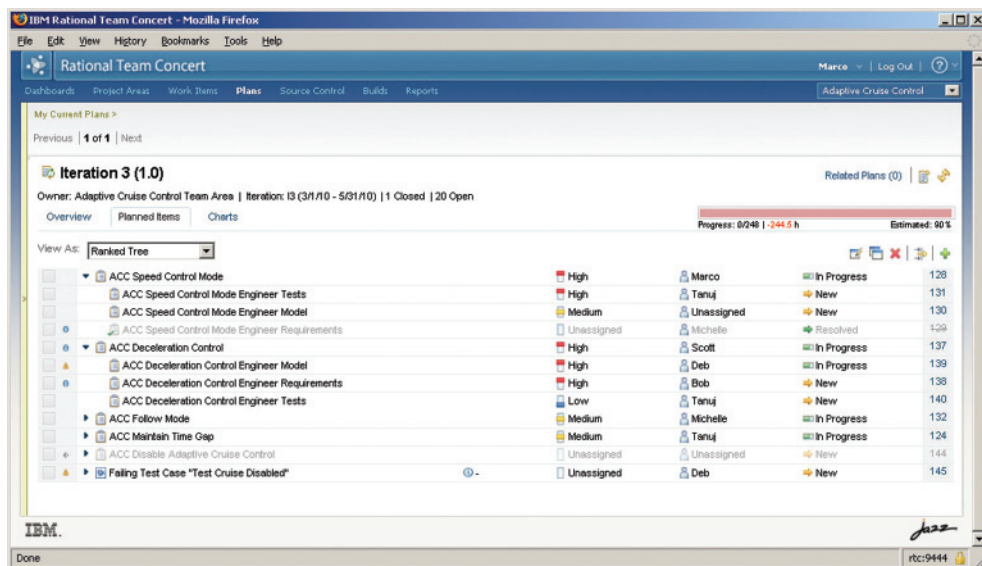
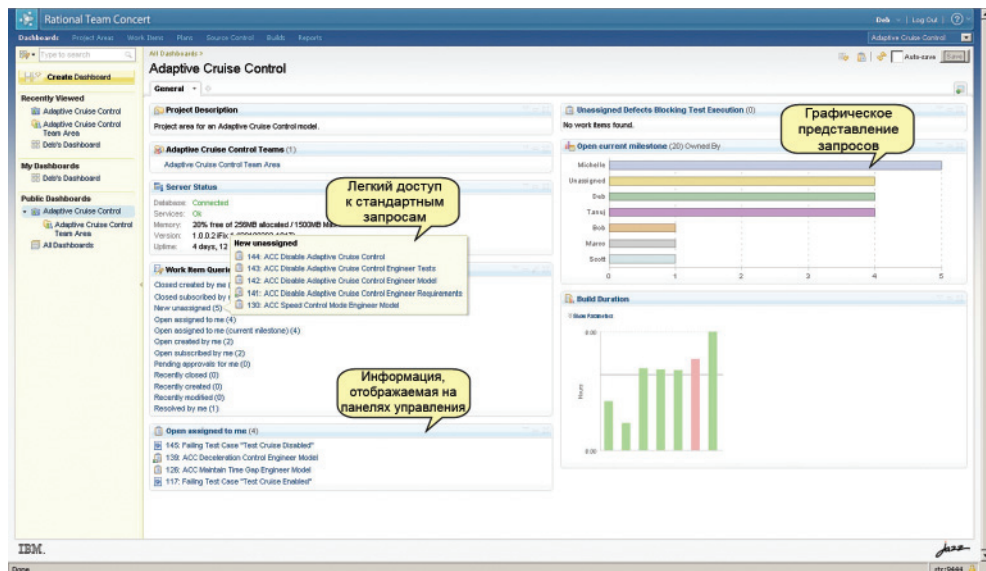


Рис. 7. Пункты итерационного плана в IBM Rational Team Concert

Общий план проекта и текущие планы команд динамически связаны с рабочими заданиями, назначенными командам. Это объясняется тем, что общие и текущие планы работ постоянно обновляются и корректируются по мере того, как команды разработчиков выполняют свои задания. Такой подход обеспечивает прозрачность не только в отношении показателей выполнения текущих заданий конкретной командой, но и общего состояния проекта, стимулируя совместную работу команд и способствуя их более эффективному взаимодействию. А наличие в предлагаемом решении интегрированного управления изменениями позволяет членам команды в рамках их текущего плана принимать участие в координировании и управлении всеми работами, например, в формировании рабочих заданий, обработке запросов на изменение, поиске и устранении ошибок в рамках одного и того же актуального плана.

Удобные панели управления заданиями в IBM Rational Team Concert и IBM Rational Quality Manager значительно повышают возможности контроля над общим проектом благодаря прозрачной и открытой отчетности команд по выполнению работ. Для удобства различных категорий пользователей панели управления заданиями могут быть сконфигурированы как общие – для всех участников проекта, или как индивидуальные – для каждой роли отдельно. На рисунке 8 показан пример такой панели. Актуальную плановую и проектную информацию панели управления заданиями могут представлять в виде таблиц или графиков, ответов на запросы, как реакцию на

произошедшие события, посредством viewlets. Отслеживая и контролируя информацию, хранящуюся в репозитории, используя богатую семантику ссылок, обеспечивающих трассировку в рамках всего жизненного цикла разработки, viewlets тем самым дополнительно расширяют возможности контроля за состоянием проекта.



**Рис. 8.** Панели управления улучшают контроль за состоянием проекта, увеличивают его прозрачность и повышают отчетность

Находящийся в «центре» предлагаемого решения IBM Rational Team Concert объединяет в единый коллектив разрозненные команды разработчиков, используя стандарты Интернета, и обеспечивает доступ к данным из любой точки мира, значительно уменьшая влияние глобальных проблем и препятствий.

## Организация рабочего процесса при разработке программного обеспечения

Помимо набора инструментальных средств рассматриваемое решение для системного и программного инжиниринга предлагает интегрированные рабочие процессы командам разработчиков программного обеспечения, практикующим поэтапный (инкрементальный) процесс разработки программного обеспечения. Речь идет об IBM Rational Harmony for Embedded RealTime, который покрывает аспекты разработки и внедрения высоконадежного и критичного к вопросам безопасности программного обеспечения. Обычно в такого рода процессах каждая итерация длится 4–6 недель, и за это время программное обеспечение разрабатывается только для одного или ограниченного (небольшого) числа конкретных сценариев использования.

На рисунке 9 показан рабочий процесс для отдельной итерации.

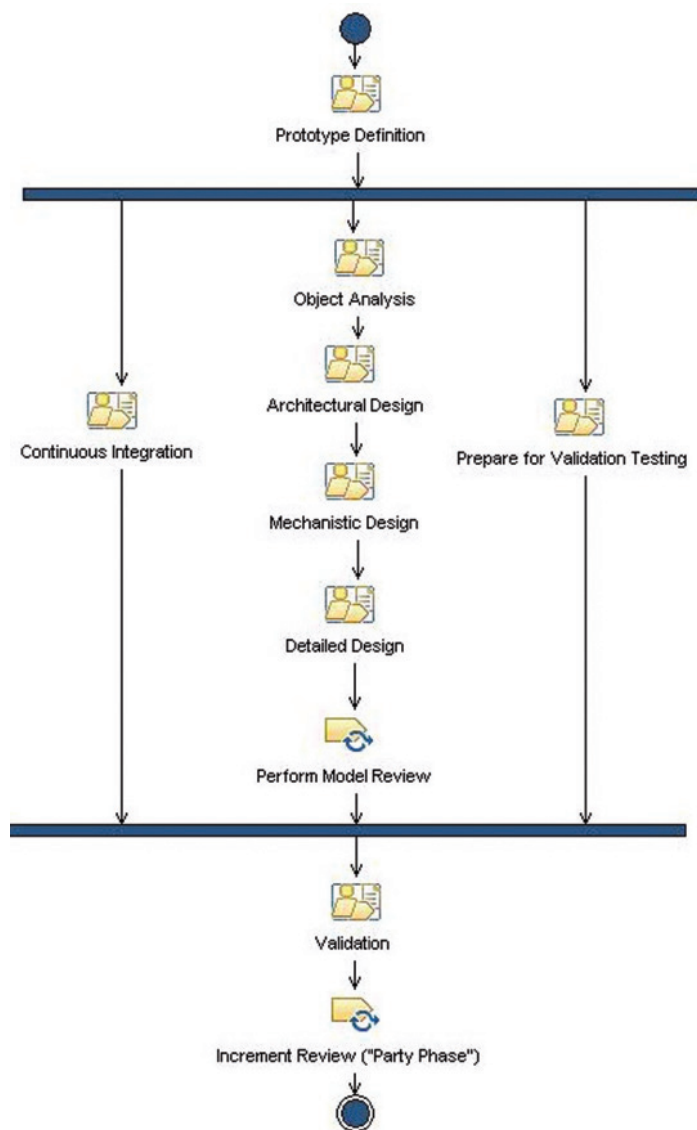


Рис. 9. Пример процесса итерации

Нетрудно заметить, что каждая отдельная итерация состоит из последовательности нескольких ключевых активностей, включая:

► **Определение прототипа (планирование итерации).**

В рамках этой активности определяются работы по данной итерации, которые в IBM Rational Team Concert расписываются в терминах итерационного плана рабочих заданий. При этом, например, одни рабочие задания, при анализе сценариев использования, определяются как пункты плана, другие же рабочие задания могут формироваться как перечень подлежащих исправлению дефектов и не-

обходимых усовершенствований или же формироваться как элементы риска. Таким образом, итерационный план конкретизирует те требования к программному обеспечению и к сценариям использования, которые должны быть реализованы в процессе итерации.

► **Объектный анализ.**

В рамках этой активности в IBM Rational Rhapsody строится функционально корректная модель и формируется базовая версия исходного кода, которые необходимы для реализации сценариев использования в данной итерации.

► **Архитектурный дизайн.**

В рамках этой активности несвязанные и мозаичные варианты (или их отдельные куски) предлагаемых архитектурных решений дополняются и уточняются в направлении пяти основных архитектурных представлений (разбивка на компоненты, совместимость, безопасность и надежность, распределение функциональности, внедрение), а также некоторыми другими дополнительными характеристиками. Для этих целей используются в основном образцы архитектурного дизайна, смоделированные в IBM Rational Rhapsody.

► **Механистический дизайн.**

Данная активность фокусируется на достижении оптимизации взаимодействий между каждым из сценариев использования системы. В основном для этих целей используются принципы и образцы механистического дизайна.

► **Детальное проектирование.**

В рамках этой активности детально прорабатываются мельчайшие элементы, такие как классы, объекты, функции, структуры данных. Команды, использующие модельно-ориентированную разработку на UML или SysML, или команды, разрабатывающие приложения, например, на C, C++ или Java™, будут использовать и IBM Rational Rhapsody, и IBM Rational Team Concert, и интегрированную в Eclipse среду разработки для проектирования, сборки, проверки и подтверждения правильности реализации задуманного.

► **Анализ модели.**

В рамках этой активности проводится анализ программного обеспечения и полученного дизайна, чтобы убедиться в том, что модель обеспечивает логичность и соответствие архитектурным замыслам, находится в строгом соответствии со стандартами качества компании и что структура программного обеспечения и поведение модели понятны всем членам команды. При этом команды, использующие IBM Rational Team Concert, могут выстроить свои рабочие процессы так, чтобы анализ и корректировку моделей сделать важной составной частью своего процесса разработки.

► **Непрерывные сборки и интеграции.**

Параллельно с большинством активностей итерации, связанных с разработкой дизайна, постоянно производятся сборки. В процессе сборки компонуется и интегрируется версия исходного кода с тем, чтобы гарантировать, что сборка логична, корректна и свободна от ошибок. Компонент командной сборки, имеющийся в IBM Rational Team Concert, поддерживает необходимый процесс и обеспечивает информационную осведомленность, возможность оперативного контроля и связь с командами тестировщиков и разработчиков программного обеспечения.



► **Подготовка к валидации.**

Объединяя разработку и тестирование, предлагаемое решение позволяет командам разработчиков заранее готовить создаваемый продукт к валидации. Планирование тестов ведется в IBM Rational Quality Manager параллельно с разработкой и позволяет создавать испытательные стенды, планы тестирования и тестовые сценарии.

► **Валидация.**

На стадии валидации программное обеспечение проверяется по принципу «черного ящика» для того, чтобы удостовериться, что оно в полной мере реализует все требования. Причем этот подход сохраняется и в случае добавления новой функциональности, и в случае создания новых тестов в рамках организации регрессивного тестирования. Это даст уверенность, что программное обеспечение, которое работало в предыдущих итерациях, не пострадало в рамках текущей итерации. Интеграция IBM Rational Quality Manager и IBM Rational Rhapsody позволяет тестировщикам многократно использовать и запускать контрольные примеры.

► **Инкрементальный анализ.**

При инкрементальном анализе (по-другому, ретроспективном) производится сравнение реальных и запланированных достижений и оценивается эффективность и результативность работ, а при необходимости текущие планы корректируются таким образом, чтобы отражать реальную действительность. IBM Rational Team Concert помогает формировать рабочие задания таким образом, чтобы облегчить проведение ретроспективных анализов.

Попытаемся более подробно рассмотреть три следующие активности:

- определение прототипа;
- объектный анализ;
- архитектурный дизайн.

## **Определение прототипа (планирование итераций)**

На стадии определения прототипа детали предстоящей итерации расписываются в терминах планов, графиков, требований (сгруппированных по сценариям использования) и контрольных точек тестирования (в виде наборов сценариев тестирования). Рисунок 10 показывает полную последовательность таких действий. Если разработке программного обеспечения предшествовали работы по системному инжинирингу, то даже несмотря на то, что требования к системе уже были сформированы, как показывает практика, требования, касающиеся программного обеспечения, могут несколько изменяться и по-новому распределяться между структурными компонентами программы.

В связи с этим необходимо дополнительно проанализировать влияние изменений и перераспределения требований на безопасность и надежность системы. Все это делается в рамках процесса определения прототипа.

IBM Rational Team Concert предлагает мощную среду для планирования, которая позволяет руководителям команд распределять работу в контексте требований и вариантов использования, корректировать план в зависимости от результатов оценки проделанной работы, а также контролировать процесс и управлять им с помощью текущих рабочих заданий.

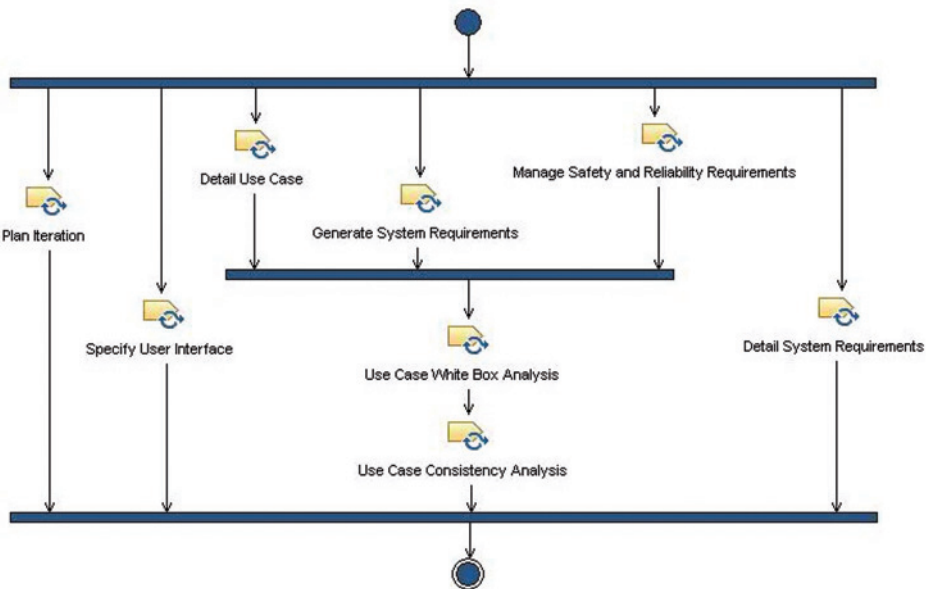


Рис. 10. Процесс определение прототипа

На рисунке 11 показано как выглядит информационная панель разработчика в IBM Rational Team Concert.

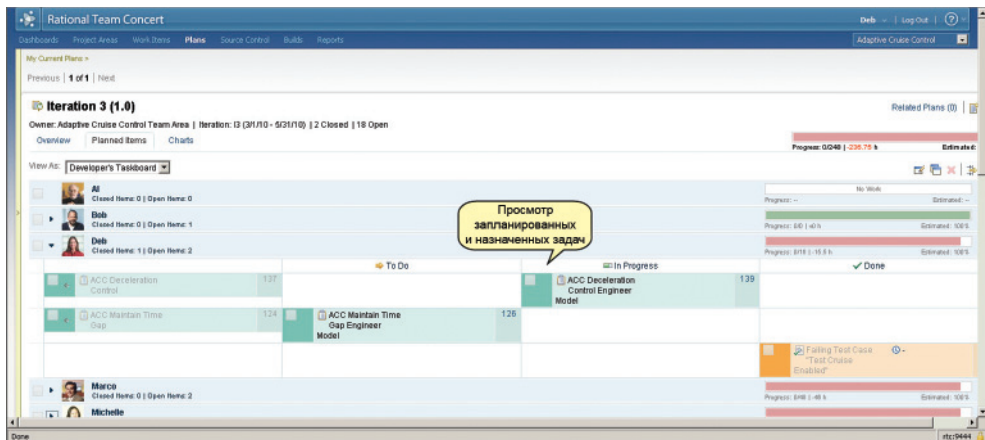


Рис. 11. Информационная панель разработчика IBM Rational Team Concert

## Объектный анализ

На рисунке 12 показана последовательность операций при объектном анализе. В ходе этого процесса, как и в процессе дизайна, инкрементально разрабатывается исходный код высокого качества. Рассматриваемый подход соответствует принципу разработки через тестирование – практики, при которой разработчик создает

модульные (unit) тесты и использует их непосредственно в процессе создания программного обеспечения (а не на этапе тестирования). Рисунок также дает представление о том, с какой периодичностью наработки поступают на конфигурационную сборку для ежедневной (а, возможно, и более частой) интеграции и тестирования. Такая практика носит название непрерывной интеграции.

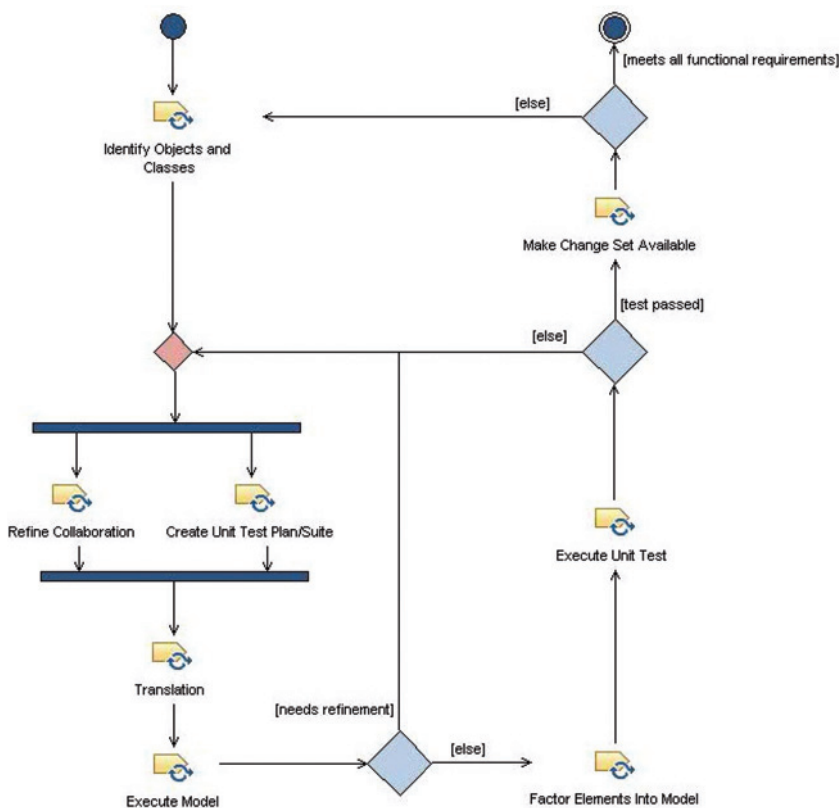


Рис. 12. Последовательность операций при объектном анализе

Процессы разработки системы следуют строгой архитектурной идее, которая базируется на пяти основных представлениях:

► **Архитектура подсистем и компонентов.**

Это представление архитектуры фокусируется на крупномасштабных элементах структуры системы, их предназначении и интерфейсах.

► **Архитектура параллелизма и ресурсов.**

Это представление архитектуры определяет модули, которые будут производить параллельные вычисления, и позиционирует функциональные элементы программного обеспечения в границах этих модулей. Это представление также определяет, как распланированы и синхронизированы операции и каким образом они будут использовать совместные ресурсы.

► **Архитектура распределения.**

Это представление архитектуры определяет, как разные составные части программного обеспечения рассредотачиваются между процессинговыми элементами (такими как процессоры и ядра процессоров), а также формирует правила и принципы взаимодействия этих частей и их совместной работы, необходимые для того, чтобы система соответствовала требуемым показателям.

► **Архитектура безопасности и надежности.**

Это представление характеризует, как система идентифицирует, выделяет и обрабатывает ошибку (сбой) в процессе своей работы, чтобы гарантировать безопасную и надежную деятельность.

► **Архитектура развертывания.**

Это представление сосредотачивается на распределении ответственности (функциональности) между различными инженерными дисциплинами, такими как программное обеспечение, электроника и механика.

## Архитектурный дизайн

Как уже упоминалось выше, в рамках активности, связанной с архитектурным дизайном системы, происходит добавление (наращивание) вариантов архитектурных решений, которые поддерживаются IBM Rational Rhapsody. Однако далеко не все архитектурные решения создаются в каждой итерации. Так, например, на первой итерации вы можете добавить архитектуру подсистем и компонентов, а также архитектуру развертывания, на третьей итерации – архитектуру безопасности и надежности, на шестой – архитектуру распределения, а архитектуру параллелизма и ресурсов – лишь на седьмой. Общая последовательность действий при создании архитектурного дизайна показана на рис. 13.

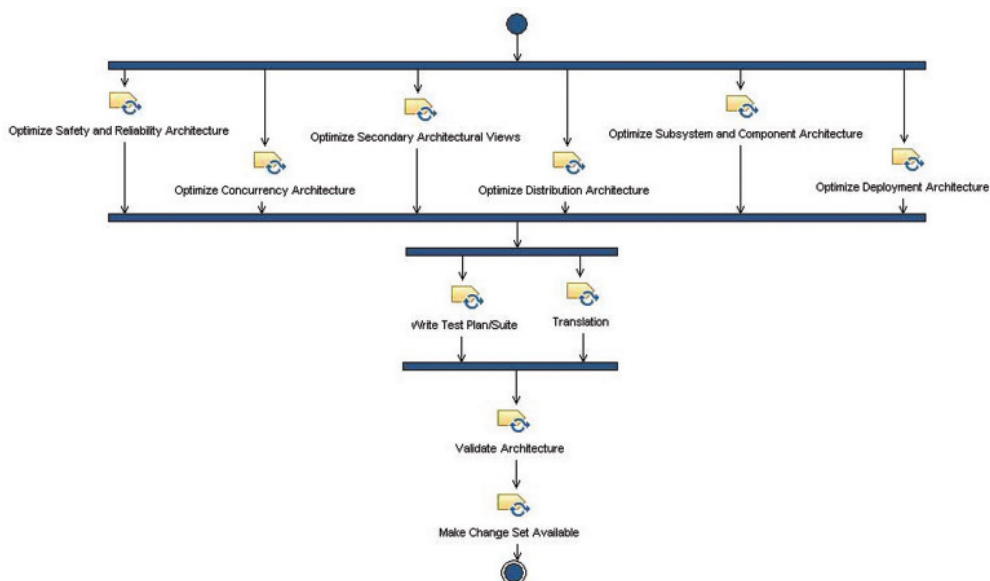


Рис. 13. Представление архитектурного дизайна программного обеспечения

IBM Rational Rhapsody обеспечивает не только возможности моделирования требований к системе и к программному обеспечению, но также поддерживает проектирование и разработку компонентов программного обеспечения. Данный продукт предоставляет собой достаточно простую в использовании, но в то же время полноценную среду для создания, управления и исполнения моделей, созданных как в SysML, так и в UML. Будучи интегрированными в IBM Rational Rhapsody системные требования связываются (соотносятся) с разрабатываемыми и детализируемыми требованиями к сценариям использования и, далее, с исполняемыми моделями требований, как это показано на рис. 14.

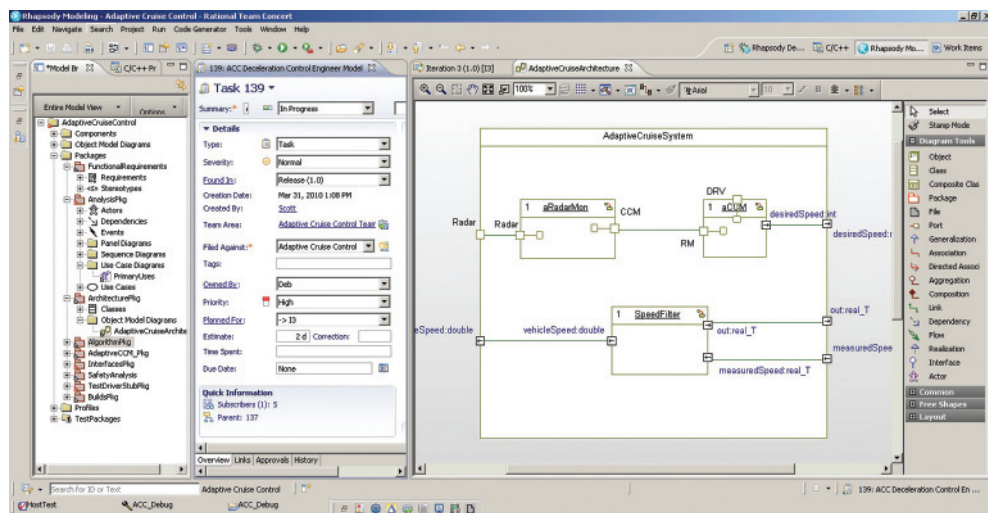


Рис. 14. Проектирование программного обеспечения с IBM Rational Rhapsody

IBM Rational Rhapsody может также взаимодействовать и с некоторыми другими инструментами, что позволяет раздвинуть границы предлагаемого решения и расширяет инженерные возможности. Этими дополнительными возможностями могут быть:

- ▶ Управление конфигурацией и изменениями (IBM Rational Team Concert).
- ▶ Управление требованиями (IBM Rational DOORS).
- ▶ Управление качеством (IBM Rational Quality Manager).
- ▶ Интеграция с продуктами партнеров, например, Wind River.
- ▶ Математические инструменты, такие как Simulink.

Помимо всего прочего IBM Rational Rhapsody имеет большое количество дополнительных опций специального назначения, например, таких как преднастроенный UML для DoDAF и MoDAF (UPDM) архитектур DoD и MOD, профиль для анализа критичных к безопасности и высоконадежных систем, профиль для разработки функционально-ориентированных встраиваемых программ на C. Также IBM Rational Rhapsody поддерживает UML-профиль для тестирования, включая инструменты IBM Rational Test Conductor и ATG (Automatic Test Generator). Эти инструменты, чер-

пая информацию из модели, позволяют автоматически создавать контрольные примеры, формировать тестовые сценарии, создавать тестовые стенды и даже автоматически запускать тестовые прогоны. Более того, инструмент ATG может анализировать модель и гарантировать, что все элементы и функционал модели протестированы, создавая при необходимости любые недостающие контрольные примеры.

## Управление качеством

Предлагаемое решение для системного и программного инжиниринга включает IBM Rational Quality Manager, является интегрированной средой для планирования, построения и исполнения тестов. Для команд тестировщиков и подразделений приемки (службы качества) IBM Rational Quality Manager служит эффективным средством для управления и контроля за процессом тестирования. Поскольку этот инструмент построен на web-решении, тестировщики могут легко получить доступ к тестовым активам и запустить или исполнить тест из любого web-браузера.

Используя инструментальную панель Rational Quality Manager, показанную на рис. 15, тестировщики могут просматривать и анализировать метрики качества системы и статус проекта, включая, например, задачи, назначенные исполнителям в IBM Rational Team Concert, или описание и статус интеграционных сборок программного обеспечения, готовых к тестированию.

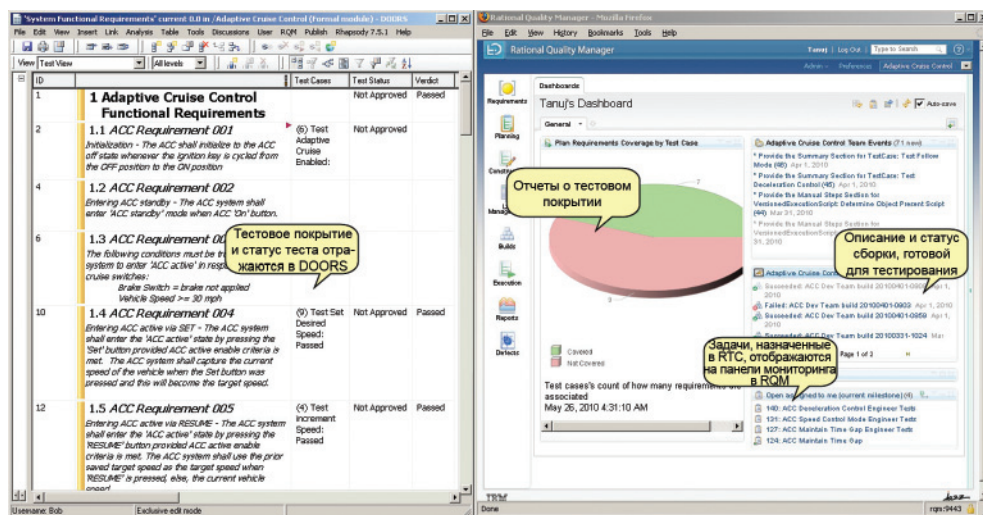


Рис. 15. Управление тестированием – планирование и исполнение в соответствии с требованиями

Подобно другим продуктам, IBM Rational Quality Manager интегрируется в предлагаемое решение с помощью платформы Jazz и принципов OSLC. Эта интеграция позволяет тестировщикам, используя web-технологии, просматривать рабочие артефакты и легко «путешествовать» между контрольными примерами, пунктами плана, требованиями и элементами модели. Использование OSLC также позволяет тестировщикам фиксировать ошибки и связывать их с артефактами репозитория, как это показано на рис. 16.



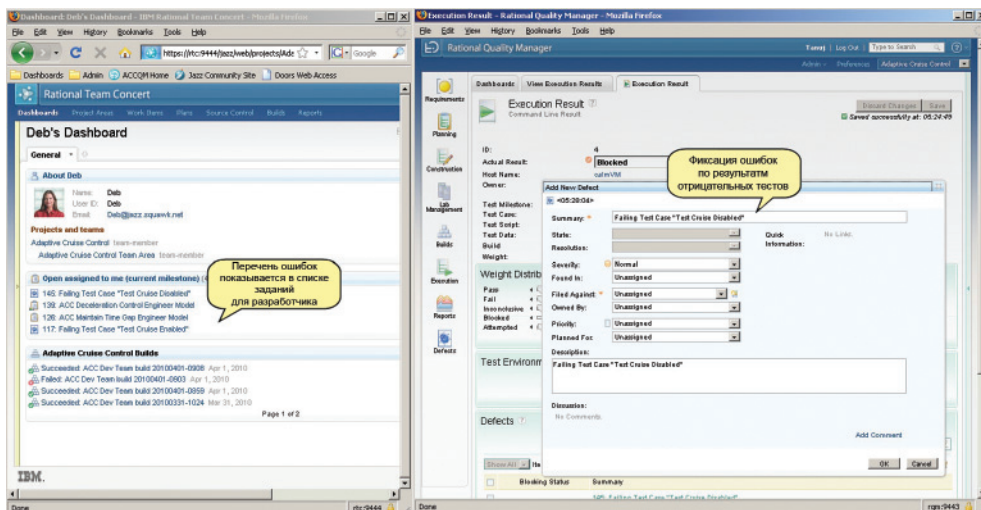


Рис. 16. Исполнение теста объединено с фиксированием ошибок

Интеграции платформы Jazz, основанные на OSLC, позволяют тестировщикам совместно с программистами работать над результатами тестов и зафиксированными ошибками. Возможность формировать запросы помогает отбирать либо завершенные инженерные задачи, готовые для первичного тестирования, либо исправленные ошибки, готовые для повторной проверки. Тестировщики имеют возможность фиксировать ошибки непосредственно в IBM Rational Team Concert, поэтому, если тестирование приостанавливается, разработчики незамедлительно получают уведомление. Привязка ошибки к результату исполнения теста и журналу тестирования обеспечивает разработчиков важнейшей информацией, позволяющей быстро обнаруживать и устранять источник проблем в программном обеспечении.

Продукт IBM Rational Test Lab Manager, будучи полностью интегрированным с IBM Rational Quality Manager, способствует значительному повышению эффективности работы тестовых лабораторий. IBM Rational Test Lab Manager оптимизирует ситуацию с запросами и предоставлением имеющихся ресурсов. Он также интегрируется с решениями для тестирования инженерных систем и программного обеспечения.

## Наращивание решения для системного и программного инжиниринга

Несмотря на то что предлагаемый набор инструментальных средств является законченным решением для команд, занимающихся разработкой систем, его функциональность тем не менее может быть значительно расширена за счет дополнительных возможностей портфеля решений IBM Rational (см. рис. 17). Под этими возможностями подразумеваются:

► **Построение систем из систем.**

Для интеграции производимых программных продуктов с Интернет и IT-системами предприятия.

► **Планирование и выстраивание бизнеса.**

Возможности по построению архитектуры предприятия и управлению портфелем проектов в соответствии со стратегическими целями компании.

► **Управление жизненным циклом разработки программного продукта.**

Интеграция с механическими, электрическими и электронными инструментальными средствами и данными.

► **Отчетность и метрики.**

Панель мониторинга композитных данных из самых разных источников и автоматизированная генерация документов.

► **Управление службами.**

Связь с оперативной деятельностью и возможности управления активами предприятия.



Рис. 17. Расширение функциональности за счет дополнительных возможностей

## Продукты IBM Rational Publishing Engine и IBM Rational Insight

IBM Rational Publishing Engine является средством для автоматической генерации документов и помогает организации формировать официальные отчеты или выпускать сопроводительную документацию в соответствии с требованиями нормативных документов. Это помогает повысить производительность, а также уменьшить стоимость и риски. А поскольку IBM Rational Publishing Engine интегрирован в предлагаемое решение для системного и программного инжиниринга, то у организации появляется возможность генерации самых разнообразных сложных и комбинированных документов, которые основываются на данных, использующихся на протяжении жизненного цикла системы и хранящихся в репозиториях.

IBM Rational Insight, контролируя различные параметры проекта, предоставляет управленческому персоналу возможность отслеживать характеристики производительности и осуществлять анализ выполнения проекта. Это помогает повысить эффективность выполнения процессов за счет предоставления точной информации, которая попадает в отчеты и на панели мониторинга.

## **Продукт IBM Rational Method Composer**

Процессы, подобные тем, что рекомендуются IBM Rational Harmony, могут служить основой для дальнейшей их адаптации под нужды организации с целью повышения показателей процесса разработки. Продукт IBM Rational Method Composer предлагает среду моделирования для создания, управления и публикации информационного наполнения такого процесса. В IBM Rational Method Composer включен набор практик, которые используют передовой экспертный анализ и опыт в разработке систем, в соблюдении законов, норм и правил, в управлении требованиями, управлении качеством, управлении изменениями и версиями. Сконфигурированные процессы становятся доступными после их публикации во внутренней сети предприятия, и они уже учитываются при построении процессов, необходимых для разработки систем с помощью обсуждаемого здесь решения для системного и программного инжиниринга.

## **Заключение**

Умение и способность проектировать и создавать сложные инженерные системы является ключевой отправной точкой для разработки и производства умных продуктов. Такие продукты или устройства, входящие в состав еще более сложных и больших систем, содержат постоянно растущую долю программного обеспечения, которое интегрируется с аппаратными средствами и электроникой.

Эта статья проиллюстрировала технологический конвейер по созданию систем, который поддерживается набором инструментальных средств IBM Rational для системного и программного инжиниринга. Эти технологические потоки состоят из множества отдельных рабочих задач и заданий, работа над которыми приводит к созданию или модификации потенциально огромного количества рабочих артефактов. К рабочим артефактам можно отнести и требования заказчика, и системные требования, и требования к программному обеспечению, модели инженерных систем и программного обеспечения, релиз проекта и пункты плана, интеграцию, планы тестирования и приемки, контрольные примеры, компоненты программного обеспечения и исходный код, сборки, отчеты. Разработка высоконадежных и критичных к безопасности систем является сложным и трудоемким мероприятием. Предлагаемый вашему вниманию набор инструментальных средств IBM Rational для системного и программного инжиниринга как раз и нацелен на преодоление таких сложностей и разрешение возникающих проблем, предлагая инженерным командам интегрированное решение для поддержания всего жизненного цикла разработки систем.

Системный инжиниринг фокусируется на формировании требований, которые должны быть корректными, полными, непротиворечивыми и достаточно детализированными на соответствующем уровне; концентрируется на создании оптимальной

архитектуры, которая поддерживает необходимую функциональность при оптимизации различных критериев разработки; а также сосредотачивается на распределении таких требований между обозначенными архитектурными элементами. Разработка встроенного и специализированного программного продукта – есть процесс создания высококачественного программного обеспечения, которое обеспечивает максимальную функциональность и индивидуальность системы, создавая высокую значимость и ценность для потребителей.

Набор инструментальных средств IBM Rational для системного и программно-инжиниринга – есть решение для разработки систем, которое поддерживает совместную работу, обеспечивает автоматизацию и повышает отчетность. На протяжении всего жизненного цикла разработки это решение помогает создавать и управлять рабочими артефактами, необходимыми для создания современных сложных систем. IBM Rational Team Concert предоставляет базовую платформу для команд-разработчиков инженерных систем и программного обеспечения, поддерживая не только совместную работу специалистов, но и обеспечивая руководству беспрецедентный уровень прозрачности при управлении качеством и контроле за развитием проекта. IBM Rational DOORS – это уникальный и выдающийся инструмент для управления требованиями, обеспечивающий такие ключевые возможности, как трасируемость, анализ покрытия, анализ результата от внесения изменения. IBM Rational Rhapsody является наилучшим в своем классе инструментом моделирования, как для инженерных систем, так и для создаваемого встроенного и специализированного программного обеспечения. IBM Rational Quality Manager обеспечивает планирование и управление тестированием, контролирует исполнение тестов и создание отчетов. Все инструментальные средства в предлагаемом решении интегрированы, чтобы обеспечить ряд наилучших и оптимальных возможностей для проектирования и разработки систем на общей и единой платформе.

## В дополнение

Если вы хотите получить больше информации о том, как IBM может помочь вам в разработке и поставке лучшего программного обеспечения, свяжитесь с представителем IBM в вашем регионе, или бизнес-партнером IBM, или посетите следующий ресурс:

<http://ibm.com/rational/solutions/systems>

Кроме того, использование возможностей подразделения IBM Global Financing может содействовать эффективному урегулированию финансовых отношений, защите от технологического устаревания, оптимизации общей стоимости владения и эксплуатации, а также получению дохода от инвестиций. Также помощь нашего подразделения Global Asset Recovery Services способствует разрешению экологических проблем, благодаря большому количеству новых энергосберегающих решений. Для получения дополнительной информации об IBM Global Financing посетите следующий ресурс:

<http://ibm.com/financing>

## Публикации по теме

В перечисленных ниже публикациях более детально обсуждаются темы, затронутые в данной статье:

- ▶ Douglass, *Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit*, Elsevier Science & Technology Books, 2010, ISBN 1856177076.
- ▶ Douglass, *Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns*, Addison-Wesley, 1999, ISBN 0201498375.
- ▶ Douglass, *Real Time UML Workshop for Embedded Systems*, Elsevier Science, 2006, ISBN 0750679069.
- ▶ Douglass and Booch, *Real-Time Agility: The Harmony/ESW Method for Real-Time and Embedded Systems Development*, Addison-Wesley, 2009, ISBN 0321545494.
- ▶ Douglass and Harel, *Real Time UML: Advances in the UML for Real-Time Systems*, Addison-Wesley, 2004 ISBN 0321160762.
- ▶ Douglass and Jensen, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*, Addison-Wesley, 2002, ISBN 0201699567.

## Об авторах

Данная статья была написана специалистами, работающими при International Technical Support Organization, Raleigh Center:

**Bruce Powel Douglass** имеет степень доктора нейрокибернетики. Более 30 лет он разрабатывал приложения, работающие в режиме реального времени в различных средах, критических по отношению к безопасности, и создавал приложения реального времени во множестве сред жесткого реального времени. Он разработал и вел курсы по быстрым и гибким методам разработки, объектно-ориентированному программированию, MDA, системам реального времени и по разработке систем, критичных к безопасности. Он является автором многих технических книг, при этом он возглавляет группу технических писателей по направлению IBM Rational, воспитывает штат профессионалов для IBM, консультируется со многими клиентами по всему миру; представляет IBM на многих различных конференциях, а также является автором инструментария и процессов для производства встроенного программного обеспечения, работающего в реальном времени.

**Mats Gothe** – архитектор решения Collaborative Life Cycle Management, предназначенного для ИТ и систем. Более 19 лет он работал с брендом IBM Rational и занимал различные позиции в IBM, включая позицию менеджера по развитию IBM Rational Rose в Швеции, занимался продажами и сервисом IBM Rational в Скандинавии, в том числе и для Ericsson Corporate Account. Он является автором нескольких публикаций, включая IBM Redbooks publication *Collaborative Application Lifecycle Management with IBM Rational Products*, SG24-7622. Он имеет степень доктора физических наук университета Uppsala. Живет и работает в Швеции.

## Вы также можете стать автором

У вас есть возможность показать ваши навыки, получить карьерный рост и стать опубликованным автором – все одновременно! Присоединитесь к проекту корпорации ITSO, помогите создать книгу в области ваших знаний и опыта использования передовых технологий. Ваша работа поможет продвижению продукта и повышению компетентности потребителя, а вы получите сеть контактов и сможете наладить отношения по работе. Команды работают от двух до шести недель, и вы можете участвовать как лично, так и дистанционно, со своего основного места пребывания.

## Примечания

Информация в этом руководстве охватывает продукцию и услуги, предлагаемые в США.

Предложения IBM по услугам, товарам и их возможностям, описанным в руководстве, могут не действовать в других странах. За информацией о текущем ассортименте доступных продуктов и услуг обращайтесь в местные представительства IBM. Явные и неявные упоминания услуг, продуктов и их возможностей не означают необходимость их применения. Допускается их замена любыми функционально эквивалентными продуктами и службами сторонних производителей, не нарушающими прав на интеллектуальную собственность IBM, при этом ответственность за проверку совместимости и продуктивности решений сторонних производителей принимает на себя пользователь.

IBM может обладать патентами или патентными заявками на технологии, описанные в настоящем руководстве, предоставление которых не означает наличия лицензии на технологии. Письменные запросы лицензий следует направлять по адресу *IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 USA*.

Приведенный ниже абзац не относится к Соединенному Королевству и иным странам, законодательству которых противоречит. КОРПОРАЦИЯ ИБМ ПРЕДОСТАВЛЯЕТ ДАННОЕ РУКОВОДСТВО «КАК ЕСТЬ» И НЕ ДАЕТ НИКАКИХ ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ (НО НЕ ОГРАНИЧИВАЯСЬ) ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ПАТЕНТНОЙ ЧИСТОТЫ, КОММЕРЧЕСКОЙ ПРИГОДНОСТИ И СООТВЕТСТВИЯ КОНКРЕТНОМУ НАЗНАЧЕНИЮ.

В отдельных государствах отказ от явных и подразумеваемых гарантий по ряду сделок запрещен, так что указанное ограничение может вас не коснуться.

В руководстве возможны опечатки и технические неточности. Приводимые в нем сведения регулярно обновляются, соответствующие изменения будут внесены в новую редакцию руководства. Корпорация IBM оставляет за собой право в любое время без уведомления модифицировать описанные в этом руководстве продукты и программные средства.

Любые ссылки на сайты сторонних компаний в этом руководстве носят исключительно информационный характер и не свидетельствуют об их поддержке корпорацией IBM; риск, связанный с применением ресурсов этих сайтов, принимает на себя пользователь. По своему усмотрению и без каких-либо обязательств IBM может использовать и распространять любые предоставленные сведения.




За информацией о сторонних продуктах обращайтесь к производителям, их публикациям и другим открытым источникам. IBM не тестировала эти продукты и не может подтвердить точность оценок производительности, совместимости и прочих параметров. Соответствующие вопросы следует направлять поставщикам этих продуктов.

Настоящее руководство содержит в качестве примеров данные и отчеты, используемые в повседневной практике предприятий. Для наиболее полной иллюстрации примеров в руководстве встречаются имена лиц, названия компаний, торговых марок, товаров. Все они вымышлены, и любые совпадения с именами и данными реально существующих компаний случайны.

## Товарные знаки

IBM, логотип IBM и *ibm.com* являются товарными знаками или зарегистрированными товарными знаками of International Business Machines Corporation в США и/или других странах. Эти и иные товарные знаки корпорации IBM отмечены в данной статье соответствующим символом (® или ™), указывая на торговые марки зарегистрированные в US или товарные знаки, охраняемые нормами общего права, принадлежавшие IBM, на момент публикации статьи. Такие торговые марки могут быть также зарегистрированы или охраняться нормами общего права в других странах. Перечень торговых знаков, актуальный на сегодняшний день, приведен на странице <http://www.ibm.com/legal/copytrade.shtml>.

Следующие названия являются торговыми знаками International Business Machines Corporation в США и/или других странах.

DOORS®	Rational Team Concert™	Redpapers™
IBM®	Rational®	Redbooks (logo)  ®
Jazz™	Redbooks®	Rhapsody®
Rational Rose®	Redpaper™	

Другим компаниям принадлежат следующие товарные знаки:

Java и все названия, образованные со словом Java, являются торговыми знаками Sun Microsystems, Inc. в США и/или других странах.

Названия других компаний, товаров и услуг могут быть товарными знаками и знаками обслуживания соответствующих владельцев.