

# Четыре ключевых подхода к повышению качества в рамках V-модели системного инжиниринга

*Сбалансированный подход к обеспечению качества сложных систем, разрабатываемых для аэрокосмической и оборонной промышленности*





## Введение

Проекты по разработке систем для аэрокосмической и оборонной промышленности характеризуются все большей сложностью и необходимостью обеспечивать высокое качество. В то же время меняющиеся рыночные условия и требования заказчиков вынуждают разработчиков ускорять выпуск систем и сокращать затраты. Таким образом, для сохранения конкурентоспособности на современном рынке необходимо находить возможности для более оперативной разработки новых продуктов, соответствующих требованиям к качеству.

Создавая более сложные и более интеллектуальные системы, основывающиеся на взаимодействиях программного обеспечения с механизмами, датчиками, микроэлектронным оборудованием и различными компонентами систем, организации неизбежно сталкиваются с непредвиденными сложностями, которые могут привести к значительным задержкам в реализации программ и проектов. В качестве примера можно вспомнить задержки в реализации крупных коммерческих проектов по разработке летательных аппаратов, широко обсуждавшиеся в последние годы в средствах массовой информации, или существенный перерасход денежных средств в проектах по созданию систем для оборонной отрасли США, таких как разработка конвертоплана V-22 Osprey.

Множество проблем неизбежно обнаруживается и при интеграции систем, и эти проблемы становятся особенно серьезными, если это случается на поздних этапах цикла разработки. Причем интеграционные проблемы возникают не только вследствие сложности, присущей современным системам. Дело в том, что жизненный цикл разработки охватывает деятельность множества специализированных рабочих групп, зачастую территориально распределенных и не связанных между собой, относящихся к разным культурам, использующих разные языки общения, работающих в разных временных зонах, и эта разобщенность также отрицательно сказывается на результатах интеграции.

К сожалению, для исключения перерасходов и отставаний от графиков не существует быстрых решений, которые можно внести в цикл разработки и тут же достичь желаемых результатов. Фактически усовершенствования в одной области жизненного цикла могут привести к непредвиденным проблемам в других областях. Например, сосредоточившись на этапе определения требований и игнорируя этапы разработки и тестирования, вы рискуете создать систему с тщательно проработанными, но плохо реализованными инженерными спецификациями. С другой стороны, учитывая сложность современных систем, невозможно протестировать качество только в самом конце проекта. Механизмы контроля качества должны быть встроены в весь жизненный цикл разработки. Более сбалансированный и проактивный подход к управлению качеством позволяет сокращать затраты и ускорять выпуск продукта благодаря совершенствованию коллективной работы, повышению эффективности взаимодействия, более раннему обнаружению и исправлению дефектов, а также строгому выполнению требований к изделию.

В данном документе рассматривается жизненный цикл проектирования систем и обсуждаются четыре ключевых подхода к повышению качества на всех этапах процесса разработки, представленного V-моделью, призванных помочь компаниям аэрокосмической и оборонной промышленности своевременно выпускать качественные продукты при сокращении затрат:

- предпринимать все возможное для правильного понимания требований (осознавая, что они могут меняться);
- моделировать систему и как можно раньше приступать к исполнению моделей;
- интегрировать, затем тестировать;
- использовать интегрированную среду коллективной разработки для унификации всей V-модели.

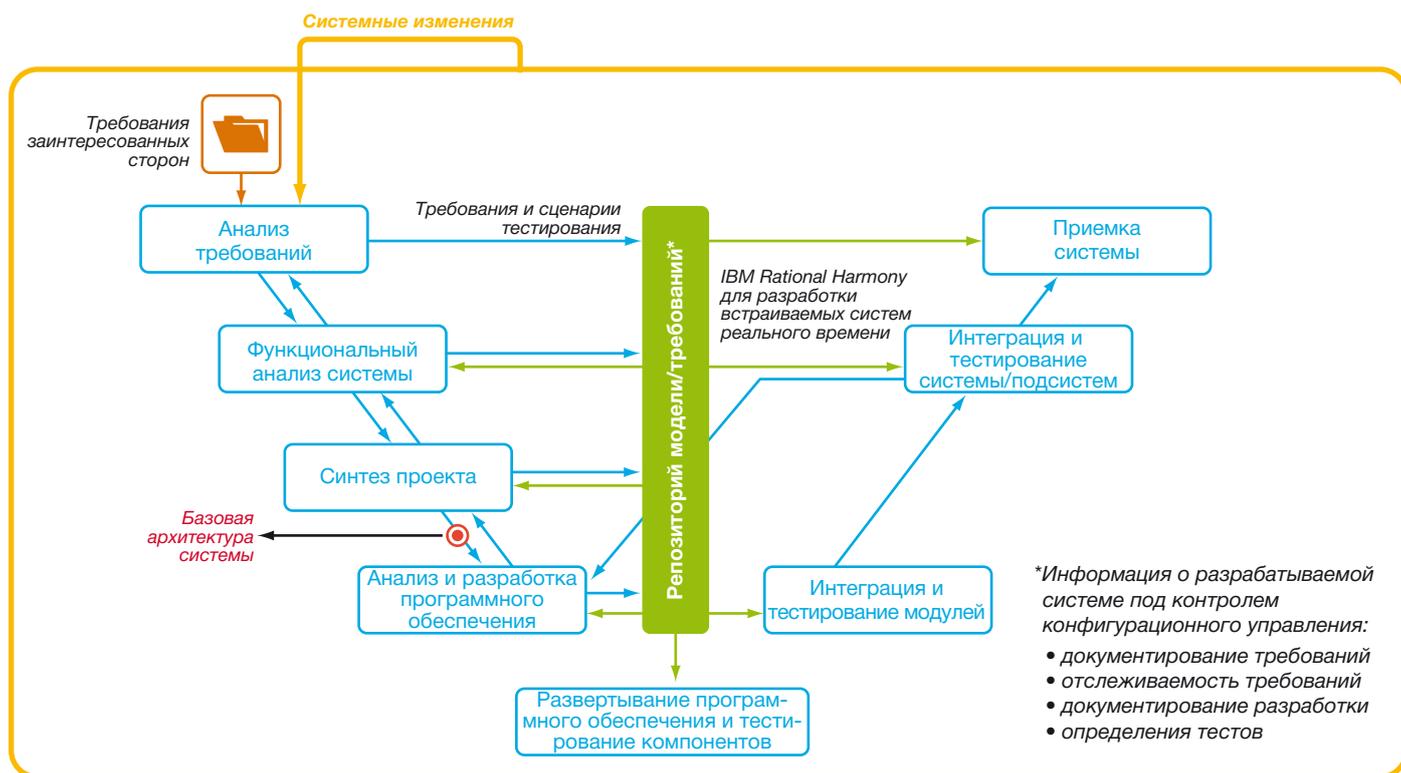


Рисунок 1. V-модель системного инжиниринга

### Предпринимать все возможное для правильного понимания требований (осознавая, что они могут меняться)

Проблемы с реализацией требований в аэрокосмической и оборонной промышленности известны уже давно: именно с ними связан перерасход денежных средств и неудачи в реализации программ<sup>1</sup>. Широко обсуждавшееся в прессе прекращение разработки президентского вертолета VH-71 в США является хорошим подтверждением вышесказанного. Очевидно, что управление требованиями в сложной цепочке поставок, охватывающей множество географических регионов и организаций, может быть весьма непростой задачей. Поэтому тщательное определение требований и управление

их изменениями в течение всего периода реализации программы видится чрезвычайно важным критерием успеха. При управлении требованиями необходимо уделять особое внимание двум ключевым аспектам:

- Важным первым шагом к снижению рисков является анализ требований, который должен фокусироваться на определении основных сценариев использования и дальнейшей приоритизации требований.
- По мере декомпозиции требований (например, от уровня требований пользователей до уровня системных требований), а также в ходе функционального анализа и построения архитектуры необходимо обеспечить отслеживаемость (трассировку) требований. Для этого лучше с самого начала побеспокоиться о

необходимых инструментах и процессах, поскольку обеспечение отслеживаемости на более поздних этапах потребует значительно большего времени и будет сопровождаться большим количеством ошибок.

Возможность и умение управлять изменениями требований также является критическим фактором успеха, поскольку требования практически всегда меняются. Понимание этого и выработка эффективных стратегий управления изменениями имеют огромное значение для успеха программы. Автоматизация механизма отслеживания (трассировки) позволит эффективно оценивать влияние изменений и управлять ими.

Кроме того, важно осознавать, что старт любого проекта по разработке инновационного продукта, связан с определенным дефицитом знаний и со значительными рисками, поэтому требует соответствующего подхода и определенной структуризации программы. Не стоит даже и надеяться, что требования будут устойчивыми и не изменятся после первого этапа реализации проекта – знания о выполняемой задаче (читай требования к продукту) должны непрерывно собираться, обновляться и совершенствоваться. Необходимо выработать итеративную стратегию, которая позволит постепенно совершенствовать знания и снижать риски. Отсутствие такой стратегии может привести к дорогостоящим ошибкам.

### **Моделировать систему и как можно раньше приступать к исполнению моделей**

Необходимость непрерывного совершенствования и анализа знания подводит нас ко второму ключевому подходу к повышению качества.

Помощью здесь окажется моделирование, позволяющее повысить уровень абстрагирования от множества создаваемых документов или программных кодов до семантически насыщенных графических представле-

ний. Моделирование обеспечит повышение качества за счет совершенствования коммуникаций, улучшения взаимодействия и коллективной работы, а также за счет понимания и заблаговременного выявления потенциальных интеграционных проблем.

А применение таких языков, как Systems Modeling Language (SysML) или Unified Modeling Language (UML) значительно упрощает выявление компонентов многократного использования. Строгая природа этих языков способствует повышению качества благодаря уменьшению разного рода неопределенностей и устранению двусмысленностей. Кроме того, наличие языковой формализации обеспечивает автоматизацию семантических проверок моделей (то есть проверки согласованности и полноты), что также способствует улучшению качества. В общем и целом, модели предлагают проектировщикам эффективную помощь в понимании и анализе требований и разработке надежных спецификаций.

Кроме того, использование моделей уже давно вышло за пределы только лишь графического представления разработок. Применение формализованных языков моделирования с отслеживаемостью до требований позволяет инструменту автоматически исполнять модели, чтобы рабочие группы могли проверять полноту и правильность реализации требований, и помогает повышать качество системы, обеспечивая большую уверенность при реализации. При этом утвержденным требованиям, исходя из их значимости или возможной сложности реализации, могут присваиваться разные уровни риска, чтобы рабочие группы могли уделять особое внимание контролю областей с высоким риском. Кроме того, детализация требований, производимая в левой части V-модели, может использоваться для создания и приоритизации сценариев тестирования, под управлением которых будут запускаться тестовые примеры в правой части V-модели.

Еще один нюанс. Переходя к интеграционной фазе, предприятия обычно руководствуются документами, описывающими интерфейсы системы (Interface Control Document, ICD). Однако у такого подхода есть и свои недостатки, поскольку документы ICD усложняют, например, контроль за последовательностью обмена сообщениями. Ниже как раз и описывается такая ситуация.

*«Примером может служить сбой при взаимодействии наземной станции слежения и спутника при использовании системы передачи файлов CFDP. Предполагалось, что спутник будет отправлять серии файлов, а наземная станция подтверждать их получение. В систему управления наземной станцией был встроены программный клапан, позволяющий операторам – в связи с ограничениями полосы пропускания – блокировать исходящие подтверждающие сообщения CFDP в случае отправки более важных команд. В какой-то момент, когда клапан был закрыт, наземная станция определила, что файловые данные потеряны, и отреагировала на эту ситуацию передачей на спутник сообщения NACK (нет подтверждения). Но поскольку клапан был закрыт эти сигналы не смогли быть переданы и встали в очередь, ожидая своей отправки после открытия клапана. Однако после открытия клапана, получив сообщение NACK, спутниковая система снова отправила все потерянные сообщения. Фактически получилось так, что в очереди находилось множество сообщений NACK для каждого файла, поскольку NACK отправлялся после каждого превышения времени ожидания обмена данными. Поэтому потерянные файловые данные отправлялись многократно, забивая канал связи и приводя к неразумному использованию ценного ресурса полосы пропускания. Следует отметить, что в данном примере обе системы вели себя как раз в соответствии с установленным протоколом»<sup>2</sup>.*

В связи с подобными проблемами использование при интеграции диаграмм последовательностей языков UML и SysML стало одной из самых успешных методик. Такой анализ системы крайне необходим группе проверки и приемки для создания тестов, оценивающих качества интеграции, и проведения приемочных испытаний. Если эта группа получает текстовые требования, то ее участники могут только гадать о том, какие тесты смогут точно отражать реальную последовательность обмена сообщениями при взаимодействии подсистем в ходе интеграционного тестирования или системы и человека в ходе приемочных испытаний.

Хотя на протяжении всего процесса разработки систем обычно применяется подход, основанный на использовании документов (и наличие соответствующей документации обуславливается условиями контрактов), использование моделей для подготовки такого количества документов может значительно ускорить завершение проекта и сократить затраты. Разумеется, тщательно структурированные требования и документы ICD могут обеспечить отслеживаемость и формализацию спецификаций интерфейса, однако они могут и затруднить понимание архитектуры системы и сложных функциональных концепций заинтересованными сторонами и разработчиками. При этом подход, использующий моделирование, предлагает более эффективный способ управления сложными разработками, свойственными системам аэрокосмической и оборонной промышленности. Кроме того, он предоставляет более эффективный метод обработки изменений: вначале нужные изменения вносятся в модели, а затем на их базе формируется необходимая документация. Такой подход значительно сокращает усилия и объемы работ, особенно в тех случаях, когда требуется обеспечить синхронизацию множества блоков информации.

Разработка систем для аэрокосмической и оборонной промышленности неизменно сопровождается рисками. Продолжительный период разработки и поздняя интеграция аппаратного и программного обеспечения зачастую приводят к отставанию от графика, особенно в тех случаях, когда возникают непредвиденные обстоятельства или обнаруживается неправильно работающий компонент. При этом даже если «виновником» проблемы является аппаратная составляющая, ради удовлетворения общих требований заказчика изменения обычно вносятся в программное обеспечение, что как раз и ведет к увеличению продолжительности разработки и дополнительным затратам. И вне зависимости от того, кто и в чем виноват, степени рисков, в конечном итоге, определяются тем, что происходит в левой части V-модели. Поэтому раннее моделирование и последующее исполнение этих моделей позволяют значительно снизить такие риски.

### **Интегрировать, затем тестировать**

Интеграция системы – кульминация V-модели системного инжиниринга – является определяющим моментом в проектах аэрокосмической и оборонной промышленности, поскольку именно здесь разрозненные рабочие группы должны интегрировать аппаратные и программные компоненты. Исторически сложилось так, что временной разброс в готовности компонентов системы и подсистемных компонентов приводили к тому, что проектные группы воспринимали позднюю интеграцию как нечто само собой разумеющееся и неизбежное. Однако жизнь показала: чем дольше ожидание, тем более серьезными будут последствия, если что-то пойдет не так. Это особенно характерно для крупных проектов, составляющей частью которых является разработка разного рода аппаратных средств, на что требуется больше времени

для итераций, чем при разработке программного обеспечения. Вот почему именно в этой переломной точке обнаруживаются самые серьезные проблемы проектирования, ведущие к непоправимым задержкам. Так, например, имели место конкретные случаи, когда крупные производители авиационной техники при попытках интеграции аппаратного и программного обеспечения обнаруживали проблемы лишь на самых последних этапах жизненного цикла, что приводило к задержкам с выпуском систем на месяцы или даже годы, не говоря уже о потерях доходов в миллиарды долларов.

Уокер Ройс (Walker Royce), ведущий эксперт IBM по экономическим аспектам разработки программного обеспечения, и Майкл Мотт (Michael Mott), почетный инженер IBM, обращают особое внимание на всю важность интеграции на ранних этапах.

*Уокер Ройс отмечает: «Важной составляющей итеративного подхода является то, что интеграция по большей части предшествует тестированию модулей, поэтому многие проблемы архитектуры и разработки систем с самыми высокими уровнями риска исключаются еще до инвестиций в тестирование модулей и полное развертывание системы. Это самый важный показатель «здоровья» проекта по разработке программного обеспечения»<sup>3</sup>.*

Но следует отметить, что применение такого подхода к крупным проектам в аэрокосмической и оборонной промышленности, с расширенными цепочками поставок и зависимостями от аппаратной составляющей, может быть непростой задачей. Однако это вполне выполнимая задача, что было подтверждено Майклом Моттом на примерах некоторых проектов по разработке космических систем с продолжительными периодами подготовки оборудования. Он настаивает на том, чтобы

отдельно разрабатываемые подсистемы интегрировались постепенно и как можно раньше для снижения общего уровня интеграционных рисков, которые неизбежно возникают, если интеграция проводится по методу «большого взрыва». При этом параллельная разработка и постепенная интеграция значительно упрощаются при использовании соответствующих методик моделирования.

*В процессе формирования архитектуры модель постоянно обновляется, но меняющаяся информация о состоянии системы все время остается видимой для других пользователей модели. И дополнительным преимуществом здесь является тот факт, что разработка не сдерживается более длительным периодом подготовки аппаратных компонентов, поздним определением требований или неполнотой спецификаций. Таким образом, все сервисы разрабатываются в едином стиле, а функциональные возможности всей системы интегрируются и тестируются на самых ранних этапах выполнения проекта. При этом использование моделирования для определения архитектуры системы позволяет группам разработчиков создавать сервисы параллельно».*

*«Наращивая функциональные возможности системы постепенно, параллельная разработка значительно ослабляет риски неудач на этапе финальной интеграции и тестирования. Отход от интеграции по принципу «большого взрыва в конце проекта» позволяет распределять внесение исправлений по всему циклу разработки, не дожидаясь перехода от разработки к интеграции и тестированию»<sup>4</sup>.*

Кроме того, эффективное использование системного моделирования совместно с макетированием обеспечивает раннюю интеграцию аппаратных и программных средств<sup>5</sup>. Ценность моделирования с учетом этого контекста заключается в том, что рабочие группы могут начинать интеграционные активности намного раньше, еще даже до получения в свое распоряжение

дорогостоящего и сложного в модификации аппаратного обеспечения или до начала разработки программного обеспечения, внесение изменений в которое на более поздних этапах может обходиться весьма дорого. Т.е. группы проектировщиков могут совместно работать над интеграцией, руководствуясь информацией о модели системы, чтобы эффективно устранять дефекты по мере их обнаружения.

В конечном итоге моделирование и тщательное управление проектом позволяют существенно ослабить риски неудач. Продолжительность интеграции может быть сокращена на 70–90%, что может способствовать значительно более раннему и более частому тестированию. В результате общее качество, как правило, повышается, а сложности управления проектом, характерные для подходов к интеграции по методу «большого взрыва» и приводящие к длительным задержкам, сводятся к минимуму. Т.е. рабочие группы могут своевременно выявлять и решать найденные проблемы на соответствующем уровне разработки еще до перехода к следующему этапу. Не следует оставлять без внимания и такой вопрос, как эффективная подготовка разного рода отчетов, которые необходимы для того, чтобы каждый участник группы понимал приоритеты разработки и обладал достаточной информацией, необходимой для работы в соответствии с этими приоритетами.

#### **Тестирование на основе моделей и автоматизация тестирования**

В конечном итоге качество работ на этапах, отображенных в правой части V-модели, зависит от качества требований и планирования работ в левой части. Другими словами, для повышения эффективности тестирования важно использовать утвержденные требования, когда они становятся доступными для разработки планов и стратегий тестирования. Уровни

рисков, первоначально присвоенные требованиям, вносят важный вклад в эту стратегию, позволяя должным образом определить и расставить приоритеты тестирования. Кроме того, трассируемость требований, проектных артефактов и тестов способствует организации эффективного регрессивного тестирования, что помогает избежать исправлений, которые могут нарушить функционирование других компонентов и составляющих системы.

В то же время существует и обратная зависимость: качество любых артефактов левой части V-модели зависит от активностей ее правой части, поскольку последние итеративно проверяют левую сторону. (Можно даже утверждать, что левая сторона ориентирована на теорию, а правая – на практику.) Как можно более раннее погружение в процессы правой части поможет обнаружить ошибки и те области, в которых могут потребоваться последующие усовершенствования. Приступая к интеграции на ранних этапах, можно ускорить получение такой информации. Ранняя интеграция вынуждает переходить от теории (например, моделей и документов) к практике (исполняемым компонентам системы), туда, где можно проще понять отношения между требованиями, увидеть компромиссы дизайна и связанные с этим последствия для качества.

Распространение моделирования на процесс тестирования также является важным шагом к повышению эффективности и улучшению конечных результатов. Если преимущества разработки на основе моделей многократно и документально подтверждены, то тестирование программного обеспечения все еще зачастую отстает от передовых методов и по-прежнему основывается на проверке кода. Применение такого

тестирования в правой части V-модели, но при использовании моделирования в ее левой части, может привести к несбалансированности рабочих нагрузок, поскольку может создаться ситуация, при которой артефакты разработки будут создаваться значительно быстрее, чем они смогут тестироваться.

Тестирование на базе моделей обеспечивает раннее и эффективное тестирование модельных разработок, помогая сократить циклы исправления дефектов. Из года в год результаты исследований организации Embedded Market Forecasters ([www.embeddedforecast.com](http://www.embeddedforecast.com)) свидетельствуют о том, что тестирование на базе моделей помогает компаниям своевременно завершать сложные разработки, при этом полностью соответствующие ожиданиям заказчиков в отношении эффективности и системной функциональности. Сотрудники компании сравнивают реальные данные, собранные в ходе опросов практикующих специалистов, предварительно разбив эти данные на три группы по типу проектов по разработке и тестированию: устаревший, переходный и передовой. В группе с устаревшим подходом вообще не используются никакие методы на базе моделей – ни при разработке, ни в процессе тестирования. В проектах второй группы – переходный подход – при разработке используется моделирование, но тестирование базируется на проверке кода. А в передовых проектах модели используются и при разработке, и в ходе тестирования. Так вот выясняется, что разработки в передовых проектах (а) являются значительно менее дорогостоящими, чем в устаревших или переходных, (б) могут выполняться быстрее и (в) с меньшим количеством дефектов (см. таблицу 1).

	Тестирование, основанное на коде; разработка без моделирования (устаревший подход)	Тестирование, основанное на коде, но с разработкой на основе моделей (переходный вариант)	Разработка и тестирование на основе моделей (передовой подход)
<b>Эффективность</b>	76,4%	80%	88,8%
<b>Функциональность систем</b>	76,7%	80%	81,5%

Таблица 1. Проекты, в которых моделирование используется и при разработке, и при тестировании, выполняются значительно быстрее и с результатами, соответствующими первоначальным ожиданиям<sup>6</sup>.

Учитывая сложность разрабатываемых систем, которые необходимо тестировать, и количество доступных и используемых технологий, представляется важным суметь развернуть такую единую платформу, которая обеспечивала бы тесную интеграцию разрозненной экосистемы инструментов и технологий для всех областей тестирования (не исключая и ручное тестирование, если это требуется) – функционального, нагрузочного, на безопасность и т.д. В идеале такая унифицированная платформа должна быть способна не только управлять экосистемой инструментов тестирования, но и фиксировать данные о состоянии и результатах тестирования. Кроме того, при обнаружении дефектов или возникновении запросов на изменения платформа должна автоматически вводить проблемы или запросы в поток разработки для их последующего разрешения. Подобная синхронизация между рабочими группами обеспечивает более эффективное управление дефектами.

Для проектов в аэрокосмической и оборонной промышленности важными аспектами являются отчетность и документация, поскольку это диктуется строгими нормативными требованиями, такими как стандарт DO-178B, регулирующий использование программного обеспечения в воздушных системах и сертификацию оборудования. Описываемый в данном документе подход к разработке и тестированию на основе

моделей позволяет значительно уменьшить нагрузки, связанные с выполнением нормативных требований. Это достигается благодаря средствам автоматизированного формирования отчетов, которые в помощь руководителям, ответственным за принятие решений, предлагают специальные информационные панели, обновляемые практически в режиме реального времени, а также актуальные отчеты для проверяющих инстанций и аудиторов.

### **Использовать интегрированную среду коллективной разработки для унификации всей V-модели системного инжиниринга с помощью ПО IBM Rational**

Поддержка коллективной работы чрезвычайно важна для объединения усилий множества специалистов и заинтересованных сторон. Проекты становятся все более масштабными, и коммуникационные проблемы нарастают экспоненциально.

Реализация преимуществ управления качеством в течение всего жизненного цикла требует стратегического сочетания интегрированных возможностей, позволяющих управлять совместной разработкой и тестированием. Программная платформа IBM Rational® для разработки систем позволяет удовлетворить потребности разработчиков и программного



Рисунок 2. Платформа IBM Rational для разработки систем

обеспечения, и инженерных систем. Эта платформа призвана оказывать помощь в выпуске высококачественных систем при снижении уровней затрат и рисков. Базовая платформа предлагает пользователям интегрированное программное обеспечение IBM Rational DOORS®, IBM Rational Rhapsody®, IBM Rational Quality Manager и IBM Rational Team Concert™ вместе с лучшими методиками и вариантами развертывания для предоставления полного комплекса средств разработки, охватывающего управление требованиями, разработку на основе моделей, управление качеством, управление изменениями и конфигурациями. Возможности платформы могут быть расширены путем интеграции с другими решениями IBM и решениями третьих сторон.

### Эффективное управление требованиями

Инструментальное средство IBM Rational DOORS предназначено для управления требованиями, что является необходимым условием успешной разработки систем. Оно обеспечивает функции контроля за изменениями, ведения истории версий требований и документов, управления правами доступа, трассировку требований, а также предлагает разного рода аналитику, связанную с требованиями. С помощью IBM Rational DOORS организации могут управлять изменениями в требованиях и обеспечивать трассируемость, которая является основой управления качеством на протяжении всего жизненного цикла. При этом требования связываются с архитектурными компонентами, планами и сценариями тестирования, требованиями других уровней.

### Разработка на основе моделей

Инструментальное средство IBM Rational Rhapsody Architect for Systems Engineers позволяет рабочим группам взаимодействовать более успешно для эффективного проектирования систем. Оно может интегрироваться с IBM Rational DOORS для развертывания интегрированной среды управления требованиями и моделирования, позволяющей рабочим группам визуально абстрагироваться от сложных построений с использованием возможностей языка SysML или UML. При помощи IBM Rational Rhapsody рабочие группы могут проверять функциональность и архитектуру на ранних этапах разработки, эффективно использовать полную трассируемость на всем протяжении жизненного цикла, совершенствовать возможности анализа влияния изменений, повышать согласованность разработок с проверкой моделей, а также обнаруживать проблемы дизайна и ошибки на ранних стадиях разработки посредством исполнения моделей.

### Централизованное управление качеством

Инструментальное средство IBM Rational Quality Manager представляет из себя подстраиваемый под запросы пользователей Web-центр коллективного управления тестированием и качеством в рамках жизненного цикла управления качеством. Где бы ни находились эксперты по обеспечению качества и другие специалисты, отвечающие за принятие решений, – на другом этаже или в другой стране, – они могут использовать IBM Rational Quality Manager для коллективной работы над практически любыми аспектами повышения качества, такими как планирование тестов, управление тестами, сбор статистики и результатов, управление дефектами, включая и обнаружение повторяю-

щихся дефектов. Это решение предлагает интерфейс и гибкость технологий Web 2.0, средства автоматизированной подготовки отчетов для предоставления участникам проекта актуальных проектных показателей и метрик, а также настраиваемые в соответствии с ролями средства анализа, помогающие заблаговременно выявлять и исключать дефекты и обеспечивать успешное выполнение проекта. Специалисты, несущие ответственность за принятие решений, могут оперативно получать доступ к точной информации для выявления возникающих тенденций и внесения текущих усовершенствований.

Обеспечение качества сложных систем – это многосторонний процесс, требующий взаимодействия с различными технологиями и инструментами, поэтому IBM Rational Quality Manager может интегрироваться с другими решениями Rational и предоставляет открытые интерфейсы для подключения различных инструментальных средств и решений IBM для тестирования, а также средств сторонних производителей.

### Управление дефектами и внесение изменений

Для проектов по разработке сложных систем в аэрокосмической и оборонной промышленности характерны непрерывные изменения, поскольку меняются требования, обнаруживаются и устраняются ошибки и дефекты. В связи с этим эффективность управления изменениями может оказывать значительное влияние на успешность проектов. Инструментальное средство IBM Rational Quality Manager интегрировано с IBM Rational Team Concert, чтобы объединить процессы управления дефектами и изменениями и поддерживать деятельность распределенных групп разработчиков, обеспечивая таким

образом повышение продуктивности как индивидуальных специалистов, так и групп программистов. Кроме того, интеграция позволяет автоматизировать доставку уведомлений о состоянии сборки из IBM Rational Team Concert в Rational Quality Manager, что дает возможность выполнять тесты для новых разработок и управлять устранением обнаруженных в процессе тестирования дефектов в единой среде управления изменениями.

### **Использование процессов, ориентированных на конкретные области**

Помимо современных инструментов, IBM предлагает пользователям методики и технологические процессы на базе лучших проверенных и подтвержденных практик, которые позволяют радикально повысить качество программного обеспечения и систем при повышении производительности, продуктивности и взаимодействия коллективной работы. IBM может предоставить ориентированные на конкретные индустрии лучшие методики для проектирования систем – через структуру процессов IBM Rational Harmony for Systems Engineering для системного инжиниринга, и для разработки программного обеспечения – через структуру процессов IBM Rational Harmony for Embedded RealTime Development для разработки встраиваемого программного обеспечения реального времени.

Предлагаемые IBM сервисные услуги позволяют реализовать последовательный подход к внедрению технологий и решений IBM для подтверждения окупаемости инвестиций, ослабления рисков и более уверенного развертывания решений.

### **Заключение**

Проекты в аэрокосмической и оборонной промышленности становятся все более сложными, в то время как бюджеты и сроки не растут так же быстро, как ожидания в отношении функциональных возможностей. Растут требования клиентов и регулирующих органов, а значит больший объем работ необходимо выполнять в более сжатые сроки. Поэтому компании, разрабатывающие системы для аэрокосмической и оборонной промышленности, должны находить новые пути для снижения рисков и сокращения затрат, чтобы сохранить свою конкурентоспособность. Описанный в данном документе многосторонний подход, сбалансированный в масштабе всего жизненного цикла разработки систем, поможет вам справиться с этой сложной проблемой.

Успех всего проекта в значительной степени определяется на ранних этапах разработки и фазах интеграции. Практически все сводится к управлению рисками. А для их эффективного управления вам необходима продуманная стратегия, процесс для реализации этой стратегии и инструменты для эффективного внедрения этого процесса. IBM поможет вам контролировать риски и выпускать сложные системы с использованием лучших методик, инструментов и услуг, охватывающих весь жизненный цикл разработки.

### **Дополнительная информация**

Чтобы узнать больше о платформе IBM Rational Software Platform for Systems, свяжитесь с региональным представителем или бизнес-партнером IBM или посетите Web-сайт: [ibm.com/software/rational/solutions/aerospace](http://ibm.com/software/rational/solutions/aerospace).

Кроме того, предложения по финансированию службы IBM Global Financing позволят вам эффективно управлять наличными средствами, отказаться от использования устаревших технологий, сократить совокупную стоимость владения и ускорить окупаемость инвестиций. Наша служба Global Asset Recovery Services поможет вам заботиться об окружающей среде, развертывая новые, более энергоэффективные решения. Дополнительную информацию о предложениях IBM Global Financing можно получить на Web-странице [ibm.com/financing](http://ibm.com/financing).





---

Copyright © 2010, Guardium, an IBM Company.  
Все права защищены.

123370, Москва  
Пресненская наб., 10  
Тел.: +7 (495) 775-8800  
Факс: +7 (495) 258-6468, 258-6404

IBM, логотип IBM, [ibm.com](http://ibm.com) и Rational являются товарными знаками International Business Machines Corporation, зарегистрированными во многих юрисдикциях по всему миру. Другие названия продукции и услуг могут являться товарными знаками IBM или других компаний. Актуальный список товарных знаков IBM представлен на странице «Информация об авторских правах и товарных знаках» по адресу [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml).

Упомянутые в данной публикации продукты или услуги IBM могут быть недоступны в ряде стран, где IBM ведет свою деятельность.

Содержащиеся в данном документе сведения предоставлены исключительно в информационных целях. Несмотря на предпринятые усилия по проверке полноты и точности содержащейся в данном документе информации, она предоставляется по принципу «как есть», без каких-либо гарантий, явных или подразумеваемых. Кроме того, данная информация основывается на текущих планах и стратегиях IBM по разработке продукции, которые IBM может изменить без уведомления. IBM не несет ответственности за какой бы то ни было ущерб, возникший в результате использования этой и любой другой документации. Содержание данного документа не подразумевает никаких заявлений или гарантий со стороны IBM (а также компаний, поставляющих ее продукцию или продающих лицензии на ее использование), или изменения условий применимых лицензионных соглашений, регулирующих использование программного обеспечения IBM.

- <sup>1</sup> Air Command and Staff College – Air University, Past UAV Program Failures and Implications for Current UAV Programs, Barak J. Carlson, майор ВВС США, апрель 2001 г.; <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA407103&Location=U2&doc=GetTRDoc.pdf>
- <sup>2</sup> IEEE, Using Sequence Diagrams to Detect Communication Problems between Systems, Mikael Lindvall и др., 2008 г.
- <sup>3</sup> IBM, Improving Software Economics, Walker Royce, май 2009 г.; <ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14148usen/RAW14148USEN.PDF>
- <sup>4,5</sup> Bruce Chesley, Ph.D.; Erik Daehler; Michael Mott; L. Dale Thomas, Ph.D. Model driven systems development for space systems. Доклад на 58-м Международном конгрессе по аэронавтике (IAC-07-D1.3.04), сентябрь 2007 г.
- <sup>6</sup> Embedded Market Forecasters, The Economics of Defective Software, Jerry Krasner, июль 2008 г.



Подлежит повторной переработке

---