

# **Требования и рабочие задания.**

## **Сравнение и противопоставление**

Paul Raymond, Telelogic, An IBM Company

Версия 1

Июнь 2008

# Содержание

Предисловие .....	3
Что такое требование и что такое рабочее задание .....	3
Где встречаются требования и рабочие задания .....	4
Отношения между требованиями и рабочими заданиями.....	5
Трассировка .....	7
Интеграция требований с рабочими заданиями .....	7
Требования и рабочие задания в краткосрочных и динамичных проектах .....	9
Отчет о дефектах как требования .....	10
Заключение .....	10
О компании Telelogic .....	11

## Предисловие

Поскольку самые разнообразные инструментальные средства, которые используются при разработке, становятся все более интегрированными, то тема схожести и различий в используемых терминах становится все более актуальной.

Это особенно заметно в области управления требованиями и характерно для инструментов, обеспечивающих трассировку в процессе разработки.

Поскольку самые современные инструменты для управления требованиями позволяют организовать трассировку на протяжении всего жизненного цикла, то им приходится не только взаимодействовать с другими инструментами разработки, но и оказывать влияние на терминологию и даже процессы, поддерживаемые этими инструментальными средствами.

Данный документ знакомит нас с теми областями, на которые управление требованиями оказывает особое влияние – мы постараемся показать, в чем схожи и чем различны между собой *требование* и артефакт, который обычно называют «*рабочим заданием*».

## Что такое *требование* и что такое *рабочее задание*

Из двух рассматриваемых понятий – требование и рабочее задание – именно термин «требование» употребляется наиболее часто. А в таких продуктах для управления требованиями как Rational DOORS, Rational RequisitePro, Borland Caliber RM этот термин является основным используемым термином.

Стоит заметить, что существует множество споров по поводу семантики термина «требование», но следует признать, что отправной точкой является тот факт, что определение (понимание) этого термина может изменяться в зависимости от того, в каком месте жизненного цикла производства продукта вы находитесь.

Так, например, «требование пользователя» может иметь следующее определение – «*изложение потребности одного или нескольких заинтересованных лиц в разрабатываемом продукте<sup>1</sup>*», а под «системным требованием» можно понимать такое определение – «*описание поведения системы или решения, удовлетворяющего изложенным потребностям заказчика*»

Совокупность всех требований описывает некоторое свойство продукта либо в терминах (на языке) проблемы, которую это свойство призвано решить, либо описывает способ, с

---

<sup>1</sup> Под словом «продукт» в данном случае понимается его наиболее широкое значение – программное обеспечение, аппаратные средства, услуги и т.д.

помощью которого данное свойство решает эту проблему. Отсюда следует вывод, что требования остаются имеющими силу и правомерными даже тогда, когда разработка продукта уже завершена и вы передали его заказчику. Т.е. можно вполне обоснованно утверждать, что требования являются (и остаются) «живым» представлением продукта и именно этим аспектом они ключевым образом отличаются от рабочих заданий.

Рабочие задания большей частью относятся к области решений по управлению проектами, где ими в основном оперируют как задачами. Их использование легко обнаружить в ряде инструментов, которые обеспечивают некоторый контроль за процессом или последовательностью операций, например, в Rational Change™ или Rational ClearQuest.

Этот список можно также продолжить упоминанием инструментов, которые составляют основу таких рабочих сред, как Eclipse или Microsoft's Team Foundation Server. И хотя в том же Eclipse рабочие задания также называются задачами, тем не менее, в данном документе - в целях избежания путаницы - все артефакты которые будут попадать в эту категорию далее будут называться рабочими заданиями.

Что можно сказать о рабочем задании?

Аналогично управлению требованиями, которое присутствует на всем протяжении жизненного цикла разработки, инструменты, предлагающие автоматизацию, обеспечивают поддержку процессов, используемых в этом жизненном цикле. И именно это их предназначение, оказывается, вносит дополнительную неразбериху в описание различий между требованием и рабочим заданием.

Обычного пользователя можно охарактеризовать списком работ ("to-do list"), который состоит из одного или нескольких рабочих заданий. Эти задания, по сути, есть задачи, которые либо напрямую поручаются пользователю другим человеком, либо этот список возникает потому, что технологический процесс достиг той своей точки, в которой этот пользователь является следующим участником процесса.

Рабочее задание имеет короткий жизненный путь.

Как только рабочее задание считается завершенным, его значимость (ценность) становится совсем уж небольшой или даже нулевой. Этим рабочее задание весьма сильно отличается от требования, которое остается существенно важным и значимым все время, т.е. если рабочее задание представляет собой нечто временное и мимолетное, то требование являет собой постоянство и неизменность (если не акцентировать внимание на том, что само требование может изменяться). Чтобы окончательно все прояснить, можно сказать так – рабочее задание есть составная часть *проекта* по разработке, в то время как требование есть часть создаваемого *продукта*.

## Где встречаются требования и рабочие задания

Вполне обоснованно, рабочие задания можно рассматривать как задачи, которые пользователю *требуется* выполнить. В этом смысле рабочие задания также могут считаться *требованиями*, но такой подход является достаточно натянутым. С точки

зрения управления требованиями мы весьма редко фиксируем формулировки в терминах того, что требуется сделать разработчику. Правильнее сказать, что мы скорее регистрируем проблему, которую пользователю следует решить, транслируя эти описания в формулировки, характеризующие, что именно должно достичь решение.

Мы не диктуем того, что именно аналитик или инженер должны сделать, чтобы сформировать и выстроить реальное решение. Эти активности лучше оставить инструментам по управлению проектами и технологическими процессами. Представляется более полезным рассматривать требование как формулировку потребности и описание решения, как это упомянуто выше, нежели рассматривать его как требование (задание) что-либо выполнить. Четкое понимание смысловой разницы между требованиями и рабочими заданиями является важным моментом, потому что в рамках жизненного цикла проекта эти понятия «живут» и существуют параллельно.

Смешение этих понятий увеличивает риск утраты множества важных смысловых нюансов и, как следствие, снижает гибкость и правильность использования терминологии.

## Отношения между требованиями и рабочими заданиями

Если, как утверждалось выше, требования и рабочие задания «живут» и существуют параллельно в рамках жизненного цикла проекта, то напрашивается вопрос – почему бы не установить некоторую паритетность и эквивалентность между этими понятиями и просто не соотнести одно с другим напрямую?

Ответ на это вопрос вытекает из сути подхода работы с требованиями.

Возникая на самом высоком уровне, зачастую в нечетком и бесформенном виде, требования выглядят как наброски того, что хотели бы видеть в продукте заинтересованные лица (правда, иногда требования возникают уже в виде описания ожидаемого решения, но даже в этом случае это требования высокого уровня).

В результате правильного процесса управления требованиями, требования высокого уровня декомпозируются затем на более детальные описания, которые проще понимать, потому что они сформулированы уже в терминах решения (не проблемы!), и их намного легче тестировать (проверять реализацию). Чтобы процесс декомпозиции был удачным, он требует определенных аналитических усилий, и **именно эти усилия** и распределяются с помощью и посредством рабочих заданий.

Ключом к пониманию отношений между требованиями и рабочими заданиями является осознание того факта, что требования не должны декомпонироваться индивидуально. Требования не должны рассматриваться как автономные элементы, поскольку заказчику требуется не набор обособленных характеристик и свойств, а потребность иметь единое гармоничное решение. Это аналогично случаю, когда клетки нашего организма, являясь индивидуальными элементами, будут бесполезными до тех пор, пока не начнут работать слаженно и все вместе.

Наилучшим образом требования декомпозируются тогда, когда они объединены в логические группы.

В качестве примера давайте рассмотрим разработку Web-сайта. Одно требование утверждает, что пользователь должен иметь возможность переходить с одной страницы на другую, а второе требование гласит, что пользователь должен иметь возможность видеть, какие другие страницы при этом доступны для перехода.

Если эти требования будут декомпозированы, а затем и реализованы по отдельности, то с очень большой вероятностью Web-сайт получится не очень дружелюбным по отношению к пользователю. Более предпочтительным способом работы с этими требованиями является передача их аналитику как части одной и той же общей задачи.

Результат отношения «много-к-одному» между требованиями и рабочим заданием показан на рис.1

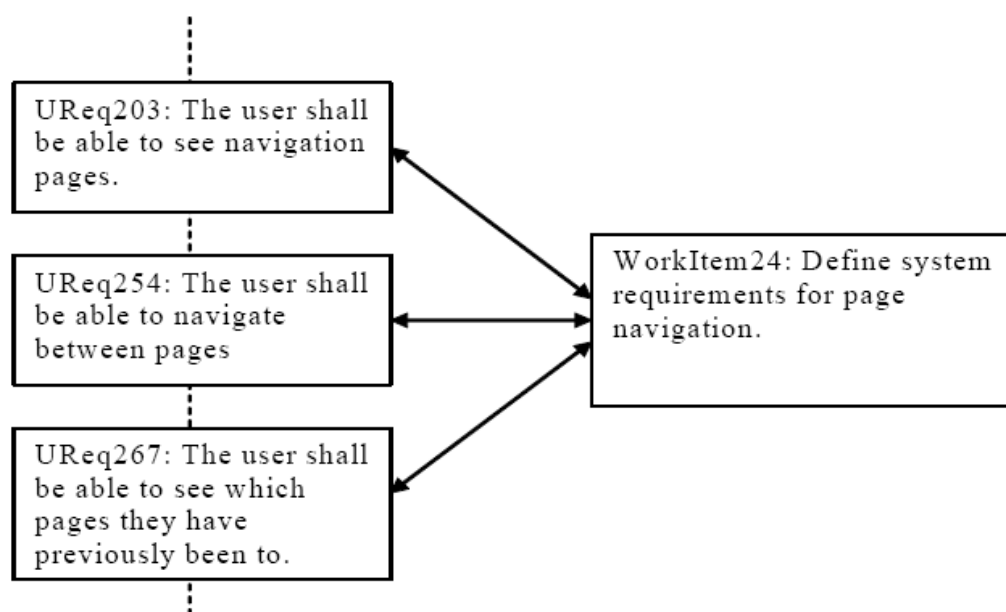


Рис.1. Требования, связанные с рабочим заданием

Вышесказанное справедливо и для случаев, когда наступает пора переходить от требований к дизайну или тестированию.

Если требования декомпозируются до соответствующего уровня детализации, то они обычно становятся входными данными для последующего процесса дизайна, а также отправной точкой для создания тестов. Но и в этом случае, нельзя сказать, что мы наблюдаем отношение «один-к-одному». Ведь реализация каждого конкретного требования не поручается отдельному аналитику или дизайнеру. Будучи сгруппированным логически и существуя такой группой как единичное рабочее задание, требования только тогда назначаются или персональному специалисту, или всей команде.

Аналогично выглядит ситуация и с тестированием.

Если даже для проверки каждого конкретного требования существует свой персональный тест, то стоит заметить, что и в этом случае каждый из них создавался не обособленно, а в логической связи и последовательности с другими тестами, т.е. и в этом случае специалисты по тестированию получают требования для проверки сразу группами.

Другими словами и здесь наблюдается отношение «много-к-одному».

Таким образом, становится вполне очевидным, что связи между требованиями и рабочими заданиями, которые формируются для реализации этих самых требований, на самом деле таковы, что они не должны непосредственно и напрямую соответствовать друг другу, потому что принцип «много-к-одному» является жизненно необходимой составляющей успешной разработки.

## Трассировка

Чтобы еще лучше понять разницу между требованиями и рабочими заданиями, обратимся к трассировке.

Одна из самых основных причин, поясняющих, почему не только существует, но и так востребовано решение по управлению требованиями, - это то, что оно имеет возможность обеспечить трассировку между артефактами. Причем трассировку не только между самими требованиями (любых уровней), но и трассировку от требований к любой информации, зарождающейся на более поздних этапах жизненного цикла (например, на этапе дизайна и тестирования).

Трассировка важна потому, что требования всегда остаются существенным и важными, и специалисты зачастую нуждаются в том, чтобы иметь возможность «пройтись» по требованиям к самым истокам, чтобы лучше представлять всю картину целиком.

Так, например, бывает весьма полезно, знакомясь с системными требованиями, которые сформированы аналитиком, обратиться к первоисточникам – к требованиям пользователя, - чтобы взглянуть на проблему глазами пользователя, т.е. с такого ракурса, как она описывается им самим.

А если вы работаете с рабочими заданиями, то необходимость использования трассировки отпадает. Скорей всего вы получите совсем небольшую пользу оттого, что сможете «пройтись» сквозь серию последовательных рабочих заданий, чтобы узнать какие активности в рамках проекта ранее поручались и выполнялись специалистом.

Из вышеописанного вытекает еще одна ключевая разница между требованиями и рабочими заданиями – при работе с требованиями трассировка имеет весьма важное значение, в то время как использование трассировки при работе с рабочими заданиями имеет мало пользы, а то и не имеет ее вовсе.

## Интеграция требований с рабочими заданиями

Интеграция различных инструментальных средств, которые используются для поддержания жизненного цикла разработки, зачастую бывает затруднительной и, как следствие, вытекает обычно в решения, которые нельзя назвать идеальными.

Ну, посудите сами, - когда вы интегрируете решение по управлению требованиями с приложением, обеспечивающим поддержку технологического процесса, то как вы предполагаете управлять отношением «много-к-одному» между данными?

Некоторые способы интеграции могут отображать данные из одного приложения в другом таким образом, что отношения, называемые линками (links), могут создаваться уже в среде только одного из приложений, что значительно упрощает дело.

В этом случае данные в основном просто копируются из одного приложения в другие.

Если же такие отношения должны существовать в обоих приложениях, то это ведет к удвоению (дубликации) данных (рис.2). На этой схеме как раз и демонстрируется пример, когда метаданные о требованиях и метаданные о рабочих заданиях хранятся по обе стороны интеграции.

В результате применения такого подхода, некоторые пользователи ошибочно пребывают в уверенности, что требования и рабочие задания это одно и то же. В то время как в действительности все то, что они наблюдают, есть просто следствие ограниченных возможностей самих приложений и их интеграции.

Такой подход не очень удобен еще и тем, что используемые данные не всегда бывают «свежими» и требуется дополнительная синхронизация для их обновления.

Отсюда напрашивается вывод, что те данные, которые видят пользователи, есть не совсем то, что они представляют собой на самом деле.

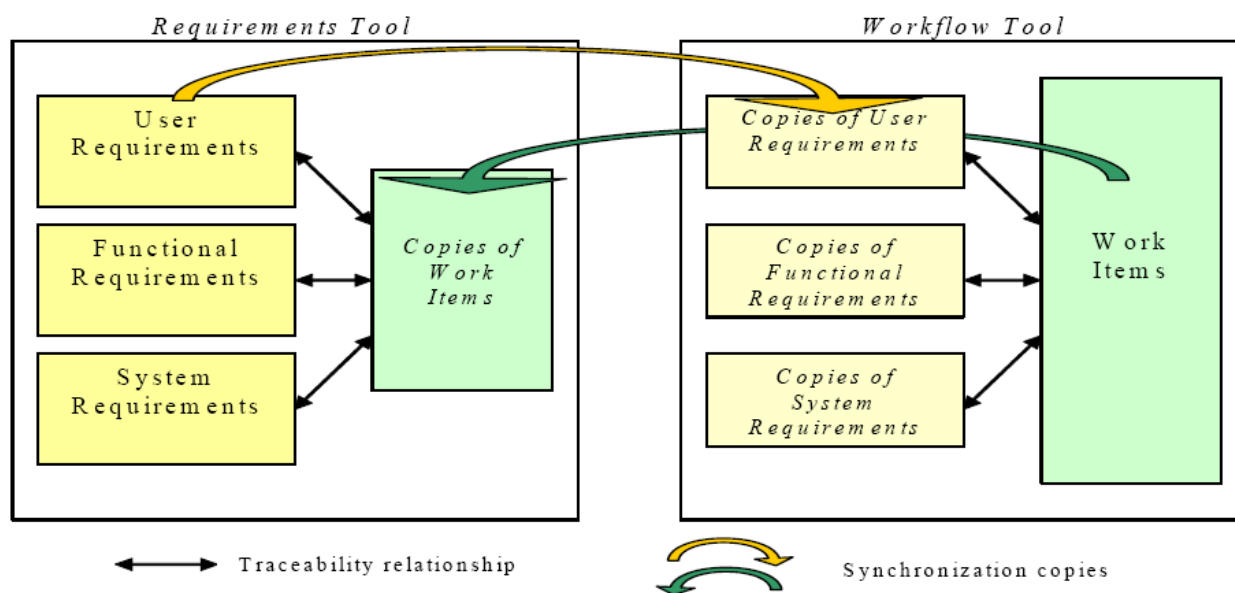


Рис.2. Дублирование данных при интеграции

Более приемлемым интеграционным решением можно назвать такое, при котором вовсе не требуется использование метаданных.

Если отношения между требованиями могут быть установлены непосредственно в одном инструменте, а между рабочими заданиями – в другом, то суть и содержимое данных по обе стороны интеграции не искажается, а сами данные всегда остаются «свежими» (рис.3).



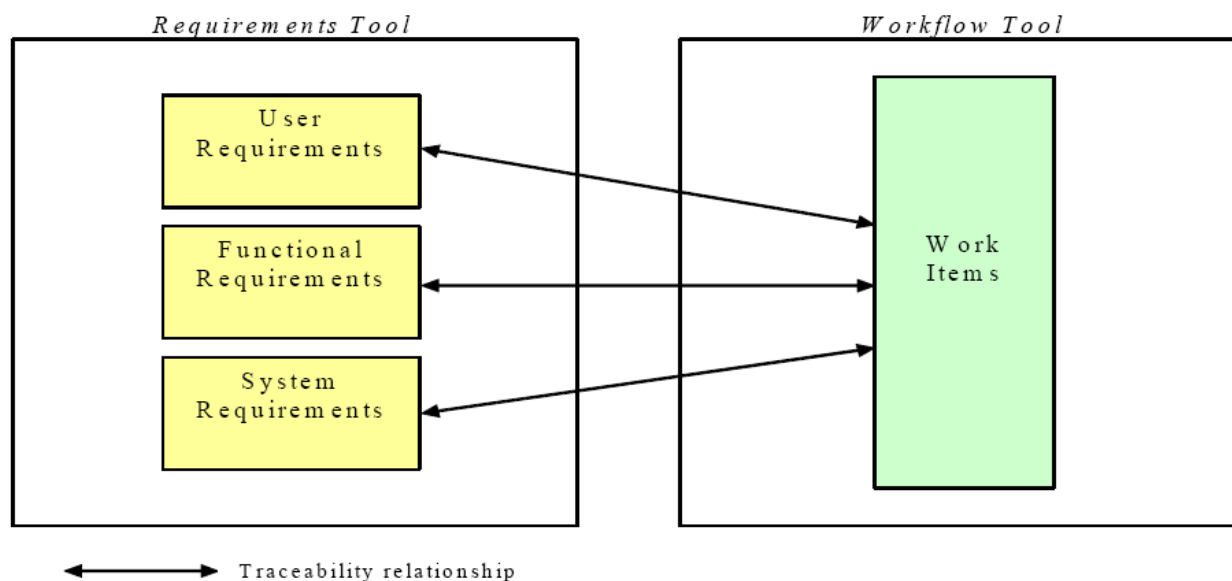


Рис.3. Линки непосредственно между инструментами

Более того, на этом примере еще легче увидеть различие между требованиями и рабочими заданиями, потому что естественная природа отношений между ними демонстрируется здесь во всей красе, включая и их характеристики «много-к-одному».

Таким образом, при использовании такой интеграции, каждому инструменту предоставляется возможность не только управлять именно теми данными, для которых он создавался и предназначен, но и правильно позиционироваться в рамках всего жизненного цикла разработки.

## Требования и рабочие задания в краткосрочных и динамичных проектах

Традиционно, большинство проектов по разработке использует модель водопада (waterfall model), - но что делать, если вы имеете дело с краткосрочными и динамичными проектами и процессами?

В такого рода проектах, где процесс управления требованиями не столь формализован, работа не с изолированными требованиями, а объединенными в логические группы, является еще более критичным моментом для достижения успеха.

Потому что команды специалистов, вовлеченные в проект, должны видеть и понимать весь масштаб предстоящих работ, а не только то, что им предстоит выполнить индивидуально.

Формирование и распределение рабочих заданий, покрывающих не одно требование,

является в таких проектах даже более важным фактором достижения успеха по сравнению с традиционными проектами, использующими модель водопада.

## Отчет о дефектах как требования

На этапе технической поддержки проекта, обнаруживаемые дефекты зачастую фиксируются и обрабатываются как изолированные проблемы.

В такой ситуации формируется стойкая тенденция связать дефект и рабочее задание в отношении «один-к-одному».

И хотя такой подход бывает вполне успешным, тем не менее, более эффективным и менее затратным может считаться такой подход, при котором проблемы и дефекты, относящиеся к одной и той же (общей) области продукта (системы), собираются в логические группы и только затем адресуются командам специалистов.

Понятно, что зачастую намного легче и удобней работать в одно и то же время с многочисленными дефектами, относящимися к одной и той же области, нежели индивидуально делегировать их исправление отдельным специалистам.

Другими словами, дефектные ведомости, контролируемые как требования, наилучшим образом соотносятся с рабочими заданиями именно при использовании отношения «много-к-одному».

## Заключение

Несмотря на то, что в некоторых компаниях существует устойчивая тенденция использовать термины *требование* и *рабочее задание* как эквивалентные и взаимозаменяемые, тем не менее, даже простой анализ показывает, что оба эти термина олицетворяют совершенно разные артефакты.

В то время, как требования и рабочие задания лишь дополняют друг друга, любые предположения, что суть этих терминов одинакова, ведут к риску причинить ущерб вашему процессу разработки.

Только тогда использование каждого из этих терминов достигает нужной цели и каждый из них при этом отлично выполняет свою собственную работу, когда они не смешиваются и не объединяются... и когда не используют одно понятие вместо другого для обеспечения трассировки – хотя эксперты и соглашались с тем, что трассировка между требованиями и рабочими заданиями является важнейшей составляющей процесса.

Понимание предназначения и уважение к функционалу, ради которого создавались и вводились термины *требование* и *рабочее задание*, является наилучшей практикой для достижения успеха в проектах по разработке.

## О компании Telelogic

Компания Telelogic® является ведущим в мире поставщиком решений для автоматизации и поддержания лучших промышленных практик — от мощнейших средств моделирования бизнес-процессов или архитектуры предприятия, до инструментов для разработки передовых инженерных систем и сложных программных приложений на основе метода связанных требований. Решения компании Telelogic позволяют организациям привести в соответствие жизненный цикл разработки продукции, систем или приложений с бизнес-целями компании и потребностями заказчика, что существенно повышает качество реализации и прогнозируемость проекта, а также сокращает срок выпуска изделия на рынок и снижает общие расходы по проекту.

С целью поддержки пользователей в процессе перехода на автоматизированный жизненный цикл разработки инструменты компании Telelogic поддерживают открытую архитектуру и используют стандартизированные языки. Будучи отраслевым лидером и ставя перед собой амбициозные цели в области технологий, компания Telelogic активно участвует в формировании будущего в области архитектуры предприятия, управления жизненным циклом приложений, управления запросами заказчиков, участвуя в таких отраслевых организациях как INCOSE, OMG, The Open Group, Eclipse, ETSI, ITU-T, TeleManagement Forum и AUTOSAR.

Штаб-квартиры компании находятся в Мальмо (Швеция) и Ирвине, Калифорния, (США). Компания Telelogic осуществляет деятельность в 20 странах мира. В число клиентов Telelogic входят такие компании как Airbus, Alcatel, BAE SYSTEMS, BMW, Boeing, DaimlerChrysler, Deutsche Bank, Ericsson, General Electric, General Motors, Lockheed Martin, Motorola, NEC, Philips, Samsung, Siemens, Sprint, Thales и Vodafone.

Более подробную информацию вы можете найти на сайтах [www.telelogic.ru](http://www.telelogic.ru) и [www.telelogic.com](http://www.telelogic.com)

### Главная штаб-квартира

P.O. Box 4128, SE-203 12 Malmö,  
Sweden  
T: + 46 40 650 00 00  
F: + 46 40 650 65 55

### Штаб-квартира в Америке

9401 Jeronimo Road Irvine, CA  
92618 USA  
T: + 1 949 830 8022  
F: + 1 949 830 8023

### Офис в России

Россия, 119021 Москва,  
ул. Фрунзе 16, строение 3  
T: +7 (495) 246 98 55,  
Ф: +7 (495) 245 50 78

© 2008 Telelogic AB. Telelogic DOORS, Telelogic Rhapsody, Telelogic System Architect и Telelogic Tau являются зарегистрированными торговыми марками компании Telelogic. Telelogic Synergy, Telelogic Change, Telelogic Dashboard, Telelogic Logiscope и Telelogic Focal Point являются торговыми марками компании Telelogic. Все остальные торговые марки являются собственностью их владельцев.

**Telelogic**  
Requirements-Driven Innovation