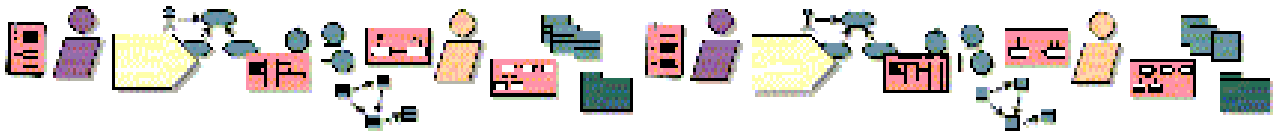


Поток работ «Требования» (продолжение 2)

Артефакт: Запросы совладельцев

Краткий обзор

Артефакт Запросы совладельцев содержит «список пожеланий», которые должны использоваться в качестве первичной информации для определения модели прецедентов, прецедентов и дополнительных спецификаций. Хотя подавляющее большинство «пожеланий» обычно выясняется в итерациях стадий Начало и Уточнение, запросы совладельцев должны собираться в течение всего проекта.

Цель документа Запросы совладельцев состоит в том, чтобы зафиксировать запросы к выполняемому проекту так, как они были первоначально сформулированы. Хотя за этот документ отвечает системный аналитик, ему будут помогать специалисты по маркетингу, конечные пользователи, заказчики – все, кто рассматривается как совладелец законченного проекта.

Артефакт может содержать любой тип запросов совладельцев к разрабатываемой системе. Кроме того, он должен содержать ссылки на все типы внешних источников, которым должна подчиниться система. Примерами таких внешних источников являются:

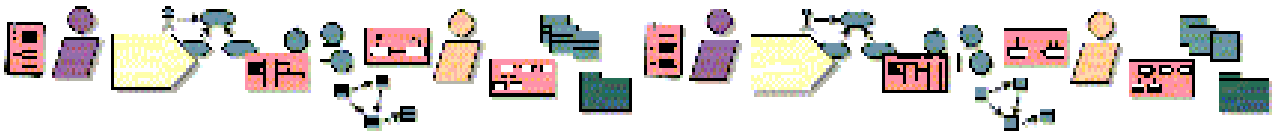
- Инструкции по работе
- Предложения и поручения
- Деловые правила
- Законы и другие правовые акты
- Наследуемые системы
- Деловые модели
- Любые результаты сеансов выявления требований и семинаров

Синхронизация

Запросы совладельцев собираются главным образом в течение стадий начала и уточнения, однако их сбор должен продолжаться в течение всего жизненного цикла проекта для планирования расширений и модификаций изделия.

Создание и уточнение артефакта Запросы совладельцев происходит в результате действия:

- Выявление запросов совладельцев



Запросы совладельцев предоставляют исходную информацию для следующих действий:

- Создание общего (делового) словаря
- Создание общего словаря
- Детализация делового прецедента
- Детализация прецедента
- Разработка документа Видение
- Поиск субъектов и прецедентов
- Поиск деловых субъектов и прецедентов
- Моделирование интерфейса пользователя

Инструментальная поддержка

Интерактивная версия Rational Unified Process содержит шаблон Microsoft Word для документа Запросы совладельцев.

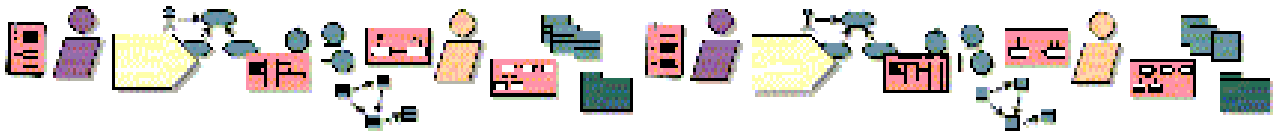
Шаблон Запросы совладельцев.dot представлен в качестве отдельного приложения к этому выпуску.

Rational RequisitePro существенно расширяет возможности работы с документом. Стиль работы аналогичен тому, который описан для артефакта Видение (см. стр. 12-10). Для запросов совладельцев существует предопределенный тип требований (SN – Stakeholder Needs).

Контрольные точки

Как и для других артефактов, для Запросов совладельцев Rational Unified Process предлагает ряд вопросов, положительный ответ на которые свидетельствует о высоком качестве артефакта. Это:

- Правильно ли выбраны совладельцы, привлеченные к созданию этого артефакта?
- Все ли прошлые запросы были пересмотрены для этого выпуска системы?
- Правильно ли был установлен набор источников информации?
- Подходящие ли методы применялись для извлечения информации?
- Достаточно ли полно отражает содержание этого артефакта все аспекты, представляющие интерес для проекта?



Артефакт: Модель прецедентов

Краткий обзор

Модель прецедентов – одно из ключевых понятий Rational Unified Process. Мы уже многократно ссылались на эту модель (см. например, стр. 4-4 – 4-7), и еще много раз будем ссылаться. Частным случаем модели прецедентов является модель деловых прецедентов (см. стр. 9-3 – 9-11).

Модель прецедентов используется следующим образом:

- Заказчик одобряет модель прецедентов. Получив это одобрение, Вы будете уверены, что система – это то, что хочет получить заказчик. Вы можете также использовать модель при обсуждении системы с заказчиком в ходе разработки.
- Потенциальные пользователи используют модель прецедентов для лучшего понимания системы.
- Архитектор использует модель прецедентов для идентификации архитектурно существенных функциональных возможностей.
- Проектировщики используют модель прецедентов для получения кратких сведений о системе. Например, когда Вы совершенствуете систему, Вам для облегчения этой работы понадобится документация по модели прецедентов.
- Руководитель проекта использует модель прецедентов для планирования работ по моделированию прецедентов и дальнейшему проектированию.
- Сотрудники организации, не работающие непосредственно над проектом, но заинтересованные в информации о его продвижении (администрация и т.д.), используют модель прецедентов для понимания того, что делается.
- Рецензенты просматривают модель прецедентов, чтобы обеспечить соответствующую регулярную обратную связь с разработчиками.
- Проектировщики используют модель прецедентов как основание для своей работы.
- Тестировщики используют модель прецедентов, чтобы как можно раньше иметь возможность планировать испытания (прецедента и его интеграции в систему).
- Те, кто будут разрабатывать следующую версию системы, будут использовать модель прецедентов, чтобы понять, как работает существующая версия.
- Авторы документации (технические писатели) используют прецеденты как основание при написании руководств для пользователей системы.

Толкование

Модель прецедентов – это модель назначенных системе функций и ее среды; она служит соглашением между заказчиком и разработчиками. Прецеденты – это тот ориентир, который направляет все действия по разработке системы. Одна и та же модель прецедентов – результат

потока работ Требования – используется как исходная информация в потоках работ Анализ и проектирование и Испытания.

Диаграмма ниже показывает часть модели прецедентов системы рециркуляторной машины.

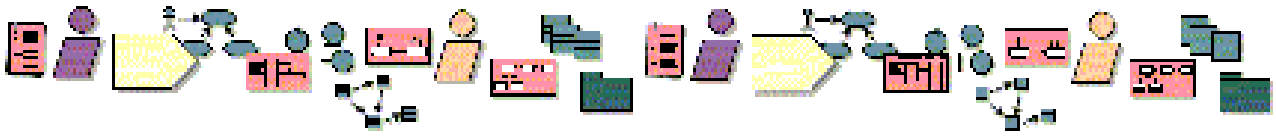
Диаграмма прецедентов показывает пример модели прецедентов с субъектами и прецедентами.

Существует множество способов моделирования системы, каждый из которых обслуживает различную цель. Однако, наиболее важная роль модели прецедентов состоит в том, чтобы объяснить поведение системы заказчику или конечному пользователю. Следовательно, модель должна быть проста для понимания.

Пользователи и любая другая система, которая может взаимодействовать с системой – это субъекты. Поскольку они представляют пользователей системы, субъекты помогают ограничить систему и дают более ясное представление того, что предполагается делать. Прецеденты разрабатываются исходя из потребностей субъектов. Это гарантирует, что система будет такой, как ожидали пользователи.

Достаточно часто модель прецедентов вырождается в функциональную декомпозицию системы. Чтобы избежать этого, обратите внимание на следующие признаки:

- «Маленькие» прецеденты означают, что описания потоков событий ограничиваются только одним или несколькими предложениями.
- «Много» прецедентов, означает, что число прецедентов кратно скорее сотне, чем десятку.
- Названия прецедентов, которые являются конструкциями подобными «эта операция над этими специфическими данными» или «выполнение этой функции с этими специфическими данными». Например, «Ввод в банкомат персонального идентификатора» не нужно моделировать как отдельный прецедент системы банкомата, так как никто не использовал бы систему, чтобы делать только это. Прецедент – это законченный поток событий, который приводит к чему-либо полезному для субъекта.



Чтобы уйти от функциональной декомпозиции, Вы должны убедиться, что модель прецедентов помогает ответить на вопросы, подобные следующим:

- Каков контекст системы?
- Зачем создается система?
- Чего хочет достичь пользователь при использовании системы?
- Какое значение имеет система для пользователей?

Синхронизация

Модель прецедентов, прежде всего, устанавливает функциональные требования к системе и предоставляет исходные данные для анализа и проектирования архитектуры. Ее желательно использовать как можно раньше в стадиях Начало и Уточнение для выделения контекста системы.

Субъекты и прецеденты отыскиваются с использованием запросов заказчиков и потенциальных пользователей как самой важной информации. Когда они обнаружены, прецеденты и субъекты должны быть кратко описаны. Прежде, чем описывать прецеденты подробно, модель прецедентов должна быть рассмотрена заказчиком, чтобы проверить, что все прецеденты и субъекты найдены, и что вместе они могут обеспечить то, что хочет заказчик.

Когда субъекты и прецеденты найдены, можно начинать подробно описывать потоки событий каждого прецедента. Эти описания показывают, как система взаимодействует с субъектами и что система делает в каждом конкретном случае. Описания выполняются, как правило, в стадии Конструирование. В итерационной среде разработки Вы будете выбирать подмножество прецедентов, которые будут детализированы в каждой итерации.

Наконец, законченная модель прецедентов (включая описания прецедентов) рассматривается, и разработчики и заказчики используют ее, чтобы договориться о том, что должна делать система.

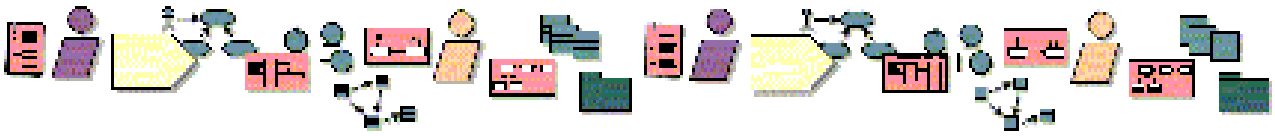
Поскольку модель прецедентов является очень мощным инструментом планирования, она вообще используется во всех стадиях цикла разработки.

Создание и уточнение Модели прецедентов происходит в результате следующих действий Rational RequisitePro:

- Выявление запросов совладельцев
- Поиск субъектов и прецедентов
- Обзор требований
- Структурирование модели прецедентов

Модель прецедентов предоставляет исходную информацию для следующих действий:

- Архитектурный анализ



- Фиксация общего словаря
- Проектирование испытаний
- Детализация прецедента
- Детализация требований к программному обеспечению
- Разработка материалов поддержки конечного пользователя
- Управление зависимостями
- Планирование испытания
- Назначение приоритетов прецедентов
- Обзор требований
- Структурирование модели прецедентов
- Анализ прецедента
- Написание примечаний к выпуску

Не функциональные требования

Нетрудно увидеть, что прецеденты – это очень хороший способ фиксации функциональных требований к системе. Но что же делать с не функциональными требованиями? Что они такое и где они фиксируются?

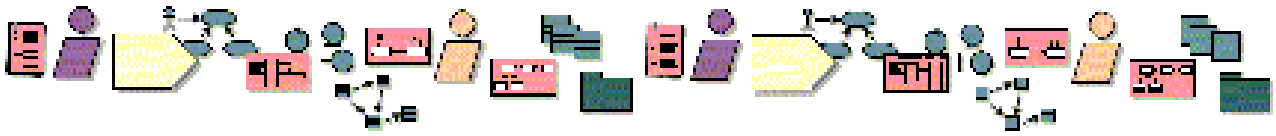
Не функциональные требования чаще всего выступают как требования применимости, надежности, эффективности и т.д. Часто это требования, которые указывают на необходимость согласованности с некоторыми юридическими и нормативными актами. Это могут быть проектные ограничения из-за используемой операционной системы, среды платформы, проблем совместимости или применяемых стандартов. Вообще, можно считать, что любое требование, которое учитывает не больше, чем один параметр проекта, должно быть расценено как проектное ограничение.

Многие не функциональные требования относятся к отдельному прецеденту и должны быть свойствами этого прецедента. В таком случае они фиксируются в пределах потока событий прецедента или как специальное требование прецедента.

Пример:

> В системе рециркуляторной машины не функциональным требованим, определенным для прецедента Возврат предметов залога, может быть:
>
> Машина должна быть способна распознать предметы залога с надежностью большей, чем 95 процентов.

Часто не функциональные требования относятся ко всей системе. Такие требования фиксируются в Дополнительной спецификации (см. стр. 13-15).



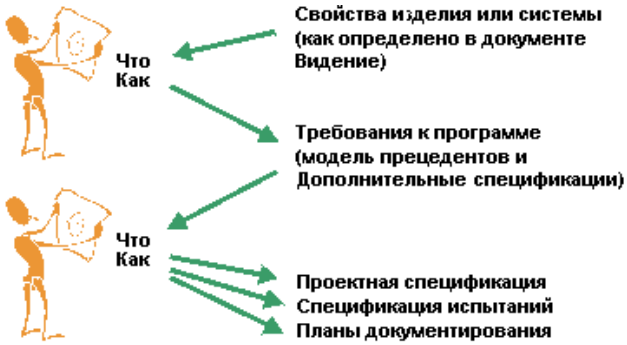
Пример:

В системе рециркуляторной машины не функциональным требованим, которое относится ко всей системе, может быть:

Машина обслуживает только одного пользователя одновременно.

Дилемма: «что» против «как»

Одна из наиболее трудных вещей – это научиться определять, на каком уровне детализации прецеденты должны «начинаться и заканчиваться». Где возникают возможности начинаться прецедентам, и где прецеденты заканчиваются, и начинается проектирование? Мы часто говорим, что прецеденты или требования к программному обеспечению должны заявить то, «что» делает система, а не то, «как» она это делает. Рассмотрите следующую иллюстрацию:



Потолок одного человека – это пол для другого.

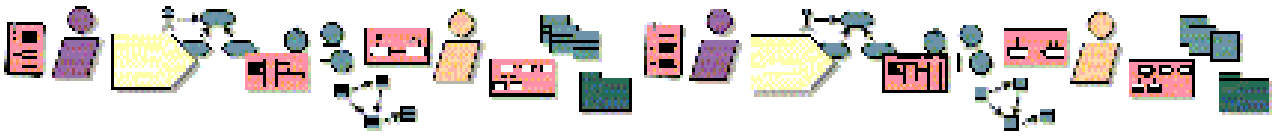
В зависимости от вашего фона, Вы будете использовать различный контекст для решения вопроса, что для Вас – «что» и что – «как». Это нужно учитывать при определении, действительно ли некоторые детали нужно включать в модель прецедентов.

Конкретные и абстрактные прецеденты

Существует различие между конкретными и абстрактными прецедентами. Конкретный прецедент инициализируется субъектом и представляет законченный поток событий. «Законченный» означает, что экземпляр прецедента исполняет полную операцию, которую запрашивает субъект.

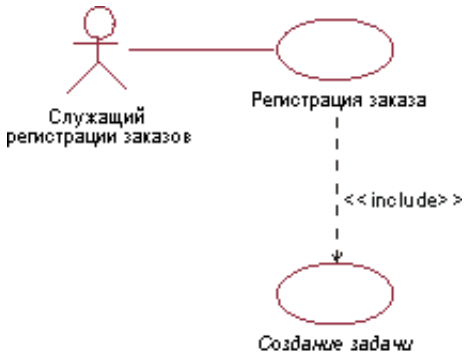
Абстрактный прецедент никогда не обрабатывается сам по себе. Абстрактные прецеденты включаются в другие прецеденты (связь включения), расширяют другие прецеденты (связь расширения) или обобщают другие прецеденты (обобщение прецедентов). Когда инициируется конкретный прецедент, создается экземпляр прецедента. Этот экземпляр показывает также и поведение, определенное связанными с ним абстрактными прецедентами. Таким образом, из абстрактных прецедентов никакие отдельные экземпляры не создаются.

Различие между этими двумя видами прецедентов важно, так как субъекты будут «видеть» в системе и инициализировать только конкретные прецеденты.



Вы указываете, что это абстрактный прецедент, записывая его название курсивом.

Пример:



Прецедент Создание задачи включен в прецедент Регистрация заказа. Создание задачи - абстрактный прецедент.

В системе складской обработки абстрактный прецедент Создание задачи включен в прецедент Регистрация заказа. Когда инициализируется Регистрация заказа, создается экземпляр Регистрация заказа, который, кроме потока событий прецедента Регистрация заказа выполняет также поток событий, описанный во включенном прецеденте Создание задачи. Создание задачи никогда не выполняется отдельно, а всегда как часть Регистрации заказа (или любых других прецедентов, в которые он включен). Следовательно, Создание задачи является абстрактным прецедентом.

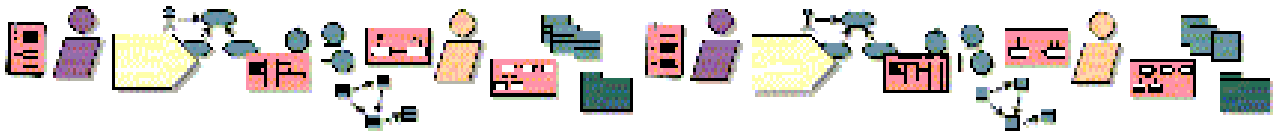
Структурирование модели прецедентов

Имеются три главных причины, по которым выполняется структурирование модели прецедентов:

- Упростить понимание прецедентов.
- Выделить разделы общего поведения, свойственного многим прецедентам
- Упростить обслуживание модели прецедентов.

Структурирование, однако, не первая вещь, которую Вы делаете. Нет никакого смысла в структурировании прецедентов, пока Вы не узнаете об их поведении немного больше, чем говорится в одном предложении краткого описания. Вы должны, по крайней мере, установить пошаговую схему потока событий прецедента, чтобы удостовериться, что Ваши решения основаны на достаточно точном понимании поведения.

Для структурирования прецедентов есть три вида связей. Вы будете использовать эти связи, чтобы выносить за скобки прецеденты, которые могут многократно использоваться в других прецедентах, как специализации или варианты прецедентов. Прецедент, который представляет



модификацию, называется добавляемым прецедентом. Прецедент, который при этом изменяется, называется основным прецедентом.

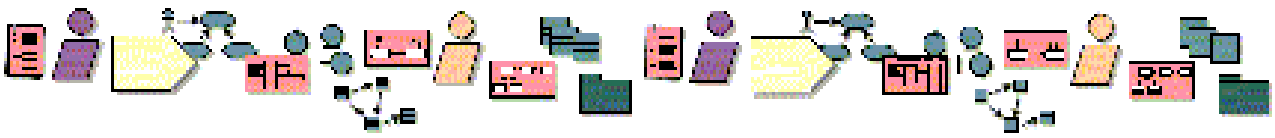
- Если имеется часть основного прецедента, представляющая функцию, от которой зависит только результат прецедента, а не способ получения результата, Вы можете вынести эту часть за скобки как добавляемый прецедент. Добавление, явно вставленное в основной прецедент, использует связь включения.
- Если имеется часть основного прецедента, которая является необязательной, или не необходимой для понимания основной цели прецедента, Вы можете вынести за скобки эту часть как добавляемый прецедент, чтобы упростить структуру основного прецедента. Добавление, неявно вставленное в основной прецедент, использует связь расширения.
- Если имеются прецеденты, которые имеют общее в поведении и структуру и подобие в цели, их общие части можно вынести за скобки как основной (исходный) прецедент, который наследуется добавляемыми (дочерними) прецедентами. Дочерние прецеденты могут вставлять новое поведение и изменять существующее поведение в структуре, которую они наследуют от исходного прецедента.

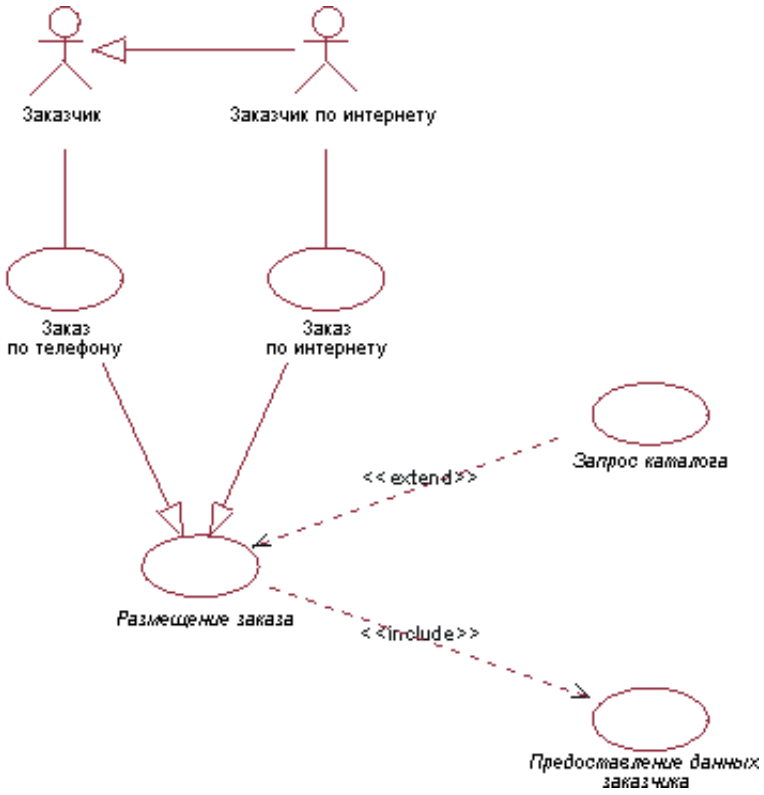
Пример:

Рассмотрите часть модели прецедентов для Системы управления заказом.

Полезно разделить обычного заказчика и заказчика по интернету, так как они имеют слегка различные свойства. Однако, так как заказчик по интернету обладает всеми свойствами обычного заказчика, Вы можете говорить, что заказчик по интернету – это специализация заказчика, обозначенного как субъект-обобщение.

Конкретные прецеденты на этой диаграмме – Заказ по телефону (инициализируемый субъектом Заказчик) и Заказ по интернету (инициализируемый Заказчиком по интернету). Эти прецеденты – два варианта общего прецедента Размещение заказа, который в этом примере является абстрактным. Прецедент Запрос каталога представляет необязательный сегмент поведения, которое не является частью основной цели Размещения заказа. Он выносится за скобки абстрактного прецедента, чтобы упростить прецедент Размещение заказа. Прецедент Предоставление данных заказчика представляет сегмент поведения, который выносится за скобки, так как его результат воздействует только на отдельную функцию прецедента Размещение заказа. Кроме того, прецедент Предоставление данных заказчика может многократно использоваться в других прецедентах. Прецеденты Запрос каталога и Предоставление данных заказчика в этом примере являются абстрактными прецедентами.





Эта диаграмма прецедентов отображает часть модели прецедентов для Системы управления заказами.

Ниже различия между тремя видами связей прецедента показаны более детально:

Вопрос: Как определяется направление связи?

Расширение – ссылка добавляемого прецедента на основной прецедент.

Включение – ссылка основного прецедента на добавляемый прецедент.

Обобщение – ссылка добавляемого прецедента (дочернего) на основной прецедент (исходный).

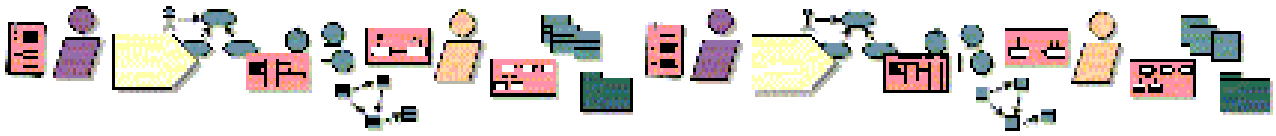
Вопрос: Может ли связь иметь множественность?

Расширение – да, с точки зрения добавления.

Включение – нет. Если Вы хотите включить один и тот же сегмент поведения больше, чем однажды, он должен быть объявлен в основном прецеденте.

Обобщение – нет.

Вопрос: Может ли связь иметь условие?



Расширение – да.

Включение – нет. Если Вы хотите выразить условие включения, Вы должен оговорить это явно в основном прецеденте.

Обобщение – нет.

Вопрос: Является ли добавляемый прецедент абстрактным?

Расширение – часто да, но не обязательно.

Включение – да.

Обобщение – часто нет, но это возможно.

Вопрос: Изменяет ли добавление поведение основного прецедента?

Расширение – неявно изменяет поведение основного прецедента.

Включение – явно изменяет эффект основного прецедента.

Обобщение – если обрабатывается основной прецедент (исходный), на него не влияет наличие дочернего. Чтобы получить результаты добавления, должен быть обработан добавляемый прецедент (дочерний).

Вопрос: Должен ли основной прецедент быть законченным и значимым?

Расширение – да.

Включение – вместе с добавлениями, да.

Обобщение – если он абстрактный, нет.

Вопрос: Должен ли добавляемый прецедент быть законченным и значимым?

Расширение – нет.

Включение – нет.

Обобщение – вместе с основным прецедентом (исходным), да.

Вопрос: Может ли добавляемый прецедент обращаться к атрибутам основного прецедента?

Расширение – да.

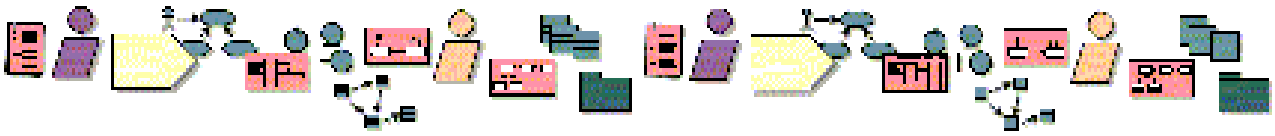
Включение – нет. Включение капсулировано, и «видит» только себя.

Обобщение – да, нормальными механизмами наследования.

Вопрос: Может ли основной прецедент обращаться к атрибутам добавляемого прецедента?

Расширение – нет. Основной прецедент должен быть правильным в отсутствие добавления.

Включение – нет. Основной прецедент знает только об эффекте добавления. Добавление капсулировано.



Обобщение – нет. Основной прецедент (исходный) должен в этом смысле быть правильным в отсутствие добавления.

Другой аспект организации модели прецедентов состоит в том, чтобы для упрощения понимания группировать прецеденты в пакеты. Модель прецедентов может быть организована как иерархия пакетов прецедентов, с «листьями», которые являются субъектами или прецедентами.

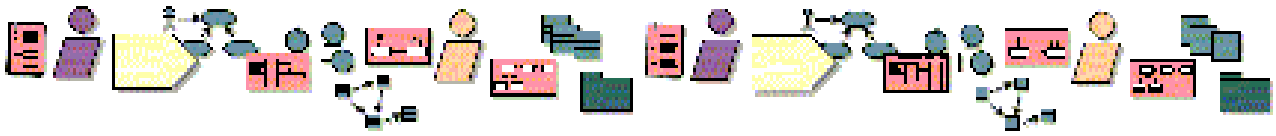
Эта диаграмма показывает иерархию модели прецедентов. Стрелки указывают возможное владение.

Всегда ли прецеденты связаны с субъектами?

Выполнение каждого прецедента связано с одним или несколькими субъектами. Экземпляр прецедента всегда начинает субъект, запрашивая какое-либо действие системы. Это подразумевает, что каждый прецедент должен иметь связи-ассоциации с субъектами. Такое правило обеспечивает создание системы для обеспечения только тех функциональных возможностей, которые необходимы пользователям, и ничего иного. Наличие прецедентов, которые никто не запрашивает – это признак того, что в модели прецедентов или в требованиях что-то неправильно.

Но для этого правила имеются некоторые исключения:

- Если прецедент абстрактный (не обрабатывается отдельно), его поведение не может включать взаимодействие с субъектом. В этом случае между этим абстрактным прецедентом и субъектами не будет связей-ассоциаций.
- Дочерний прецедент в отношениях обобщения не должен связываться с субъектом, если исходный прецедент полностью описывает связь субъекта.
- Основной прецедент в отношениях включения не должен связываться с субъектом, если прецедент включения полностью описывает связь субъекта.



- Прецедент может быть инициирован согласно расписанию (например, один раз в неделю или один раз в день). Это означает, что инициатором являются системные часы. Системные часы внутренние по отношению к системе, и прецедент инициализируется не субъектом, а внутренним событием в системе. Если никакое другое взаимодействие субъекта с прецедентом не происходит, он не будет иметь никаких связей с субъектами. Однако, для ясности, Вы можете использовать фиктивный субъект «Время», чтобы показать на диаграмме прецедентов, как инициируется прецедент.

Инструментальная поддержка

Как правило, модель прецедентов иллюстрируется одной, а чаще, многими, диаграммами прецедентов. Для работы с диаграммами используется Rational Rose.

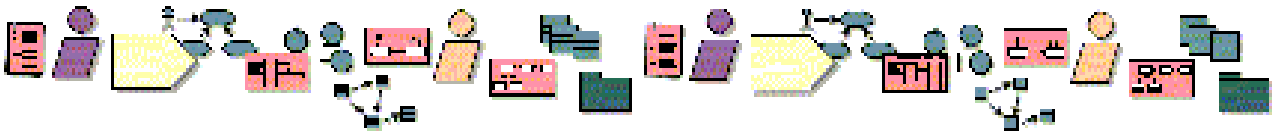
Диаграмма прецедентов отображает субъекты, прецеденты и их связи. Диаграммы прецедентов могут быть организованы в пакеты прецедентов, показывая только то, что является уместным в пределах отдельного пакета.

Никаких строгих правил относительно того, как иллюстрировать модель диаграммами прецедентов, не существует. Показывайте интересные, по Вашему мнению, связи в модели. Могут быть интересны следующие диаграммы:

- Субъекты, принадлежащие к одному и тому же пакету прецедентов.
- Субъект и все прецеденты, с которыми он взаимодействует. Диаграмма этого типа может функционировать как локальная диаграмма субъекта, и вероятно, быть связанной с ним.
- Прецеденты, которые обрабатывают одну и ту же информацию.
- Прецеденты, используемые одной и той же группой субъектов.
- Прецеденты, которые часто выполняются в одной последовательности.
- Прецеденты, которые принадлежат одному и тому же пакету прецедента.
- Наиболее важные прецеденты. Диаграмма этого типа может функционировать как резюме модели, и вероятно может быть включена в представление прецедентов верхнего уровня.
- Прецеденты, разработанные вместе (в пределах одного и того же приращения).
- Характерный прецедент и его связи с субъектами и с другими прецедентами. Диаграмма этого типа может функционировать как локальная диаграмма прецедента, и вероятно, быть связанной с ним.

Рекомендуется включить каждый субъект, прецедент и связь, по крайней мере, в одну из диаграмм. Если это сделает модель прецедентов более ясной, они могут быть частью нескольких диаграмм, и Вы можете показывать их несколько раз в одной и той же диаграмме.

Кроме диаграмм, для иллюстрации модели прецедентов выполняется ее **обзорное описание**.



Обзорное описание модели прецедентов должно:

- Перечислять основные прецеденты системы (причина создания системы).
- Суммировать важные технические факты относительно системы.
- Указывать границы системы – вещи, которые система, как предполагается, не делает.
- Указывать среду системы, например, целевые платформы и существующее программное обеспечение.
- Описывать все последовательности, в которых обычно выполняются прецеденты в системе.
- Определять функциональные возможности, не обработанные моделью прецедентов.

Пример:

Ниже следует пример обзорного описания модели прецедентов рециркуляторной машины:

Эта модель содержит три субъекта и три прецедента. Основной прецедент – Возврат предметов, который представляет главную цель работы рециркуляторной машины.

Прецеденты поддержки:

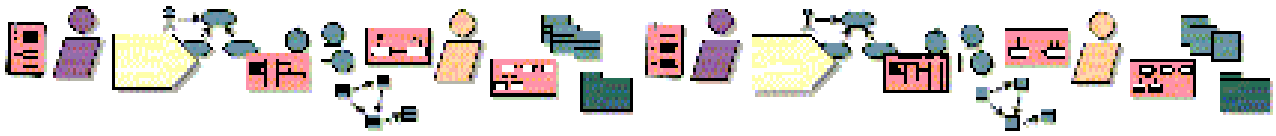
- Печать ежедневного отчета, который позволяет оператору получать отчет относительно того, сколько предметов возвращено.
- Администрирование предметами залога, который позволяет оператору изменять значение возмещения для типа предмета залога или добавлять новые типы предметов залога.

Rational RequisitePro содержит шаблон документа Обзор модели деловых прецедентов, который используется Rational SoDA для создания документа, базируясь на информации, имеющейся в модели Rational Rose

Контрольные точки

Вы имеете действительно хорошую модель прецедентов, если Вы сможете подтвердить все следующие утверждения:

- Раздел Введение описания модели прецедентов обеспечивает ясный, краткий обзор цели и функциональных возможностей системы.
- Модель прецедентов ясно представляет поведение системы; рассматривая модель, можно легко понять то, что делает система.
- Все прецеденты идентифицированы; прецеденты все вместе объясняют все требуемое поведение.
- Каждое функциональное требование отображается, по крайней мере, одним прецедентом.
- Все требования, которые не содержат функциональности, но которые должны быть удовлетворены определенными прецедентами, отображены в этих прецедентах.



- Модель прецедентов не содержит лишнего поведения; все прецеденты могут быть подтверждены прослеживанием их назад к функциональным требованиям.
- Все связи между прецедентами действительно нужны (то есть имеется подтверждение для всех связей включения, расширения и обобщения).
- Там, где модель большая, и/или ответственности частей модели распределены, используются соответствующие пакеты прецедентов.
  - Межпакетные зависимости сокращены или устранены для предотвращения конфликтов монопольного использования модельных элементов.
  - Пакетирование делает модель проще для понимания.

Артефакт: Дополнительная спецификация

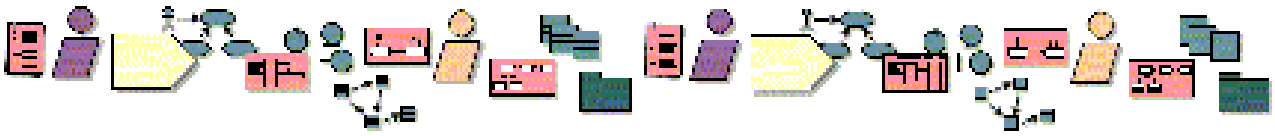
Краткий обзор

Дополнительные спецификации охватывают требования к системе, которые не охватываются в прецедентах модели прецедентов. Такие требования включают:

- Юридические и нормативные требования и применяемые стандарты.
- Атрибуты качества разрабатываемой системы, включая требования применимости, надежности, эффективности и пригодности к эксплуатации.
- Другие требования типа требований операционной системы и среды, совместимости и конструктивных ограничений.

Этот артефакт в разной мере используют следующие участники проекта:

- Системный аналитик создает и сопровождает Дополнительную спецификацию, которая служит средством общения между системным аналитиком, заказчиком и другими разработчиками.
- Проектировщики используют Дополнительную спецификацию как справочник при определении ответственности, действий и атрибутов классов и при приспособлении классов к среде выполнения.
- Кодировщики обращаются к Дополнительной спецификации за исходной информацией при выполнении классов.
- Руководитель проекта обращается к Дополнительной спецификации за исходной информацией при планировании итераций.
- Тестировщики используют Дополнительную спецификацию для проверки соответствия системы всем требованиям.



Синхронизация

Создание, сопровождение и использование Дополнительной спецификации идет «рука об руку» с созданием, сопровождением и использованием модели прецедентов, имея ввиду, что:

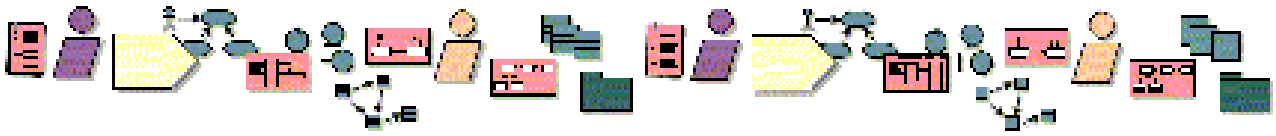
- Первоначально она рассматривается в стадии Начало, как дополнение к возможностям системы, определенным в модели прецедентов.
- Она постепенно усовершенствуется вместе с моделью прецедентов в течение стадий Уточнение и Конструирование.

Создание и сопровождение (изменение) артефакта Дополнительная спецификация в Rational Unified Process происходит в действиях:

- Разработка документа Видение
- Детализация прецедента
- Детализация требований к программному обеспечению
- Поиск субъектов и прецедентов
- Определение требования автоматизации
- Обзор требований

Артефакт Дополнительная спецификация предоставляет исходную информацию для следующих действий:

- Архитектурный анализ
- Проектирование класса
- Проектирование базы данных
- Описание архитектуры реального времени
- Проектирование испытания
- Детализация прецедента
- Детализация требований к программному обеспечению
- Идентификация проектных элементов
- Идентификация проектных механизмов
- Выполнение компонента
- Выполнение испытания
- Управление зависимостями
- Моделирование интерфейса пользователя



- Выполнение испытания модуля
- Планирование испытания
- Прототипирование интерфейса пользователя
- Обзор требований
- Обзор архитектуры
- Обзор проекта
- Структурирование модели выполнения
- Структурирование модели прецедентов
- Анализ прецедента
- Проектирование прецедента

Инструментальная поддержка

Интерактивная версия Rational Unified Process содержит шаблон Microsoft Word для документа Дополнительная спецификация.

Шаблон Дополнительная спецификация.dot представлен в качестве отдельного приложения к этому выпуску.

Для дополнительных требований в Rational RequisitePro существует предопределенный тип требований – SUPL.

Контрольные точки

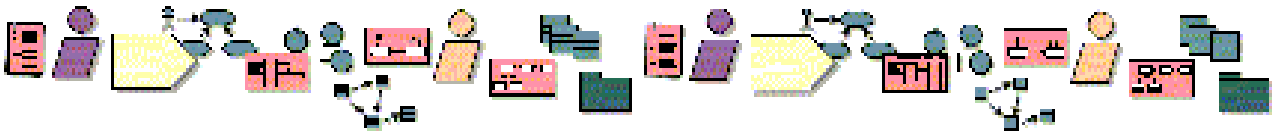
Чтобы детализировать все требования, которые не определены в пределах модели прецедентов, должны быть освещены следующие основные проблемы:

Функциональные возможности: Что делает, как предполагается, программное обеспечение? Эта информация должна включать:

- Контроль информации при вводе
- Общие ответы на аварийные ситуации, включая: переполнение, возможности коммуникаций, обработку ошибок и восстановление
- Влияние параметров
- Связи входной и выходной информации, включая последовательности ввода-вывода и формулы преобразования входной информации в выходную

Внешние интерфейсы: как программное обеспечение взаимодействует с людьми, аппаратными средствами системы, другими аппаратными средствами и другим программным обеспечением?

Эффективность: каково быстродействие, доступность, время срабатывания, время выполнения



различных программных функций и т.д.? Все ли статические и динамические требования включены?

Логические требования базы данных: все ли логические требования к информации, которая должна быть помещена в базу данных, определены? Они могут включать:

- Типы информации, используемой различными функциональностями
- Частоту использования
- Возможности доступа
- Сущности данных и их связи
- Ограничения целостности
- Требования к сохранности данных

Соответствие стандартам: все ли требования происходят из существующих стандартов и указанных правил? Как они будут отслеживаться?

Атрибуты: достаточно ли полно определены надежность, доступность, взаимозаменяемость, правильность, надежность в эксплуатации, защита и т.д.?

Проектные ограничения, наложенные на выполнение: действительно ли указаны все требуемые стандарты, язык физической реализации, стратегия поддержания целостности базы данных, ограничения ресурсов, среды и т.д.?

Артефакт: Атрибуты требований

Краткий обзор

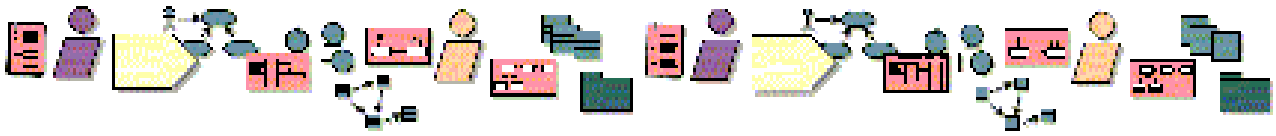
Артефакт Атрибуты требований предоставляет репозиторий текстов требований, их атрибутов и трассируемости для всех требований. Он должен обеспечить доступ для просмотра текущего состояния требований в следующих представлениях:

1. Матрицы атрибутов требований

Для каждого типа требований (см. стр. 11-3) представляется матрица, которая перечисляет требования по одной оси и все атрибуты этого типа требований по другой оси. Для каждого требования показываются значения его соответствующих атрибутов.

2. Матрицы трассируемости требований

Для каждого типа требований представляется матрица, которая перечисляет требования по одной оси и все трассируемые объекты по другой оси. Для каждой трассируемости показывается ее состояние (OK или сомнительная). О сомнительности трассируемости см. стр. 11-5)



3. Дерево трассируемости требований

Дерево трассируемости обеспечивает графическое представление связей трассируемости к или от требований определенного типа требований.

Синхронизация

Конфигурация этого репозитория определяется в Плане управления требованиями. Репозиторий нужно установить и начать заполнять в конце стадии Начало. Содержание будет пополняться и модифицироваться в течение всего цикла разработки.

Создание и модификация атрибутов требований происходит в результате следующих действий:

- Разработка документа Видение
- Детализация прецедента
- Детализация требований к программному обеспечению
- Разработка плана управления требованиями
- Назначение приоритетов прецедентов
- Управление зависимостями

Атрибуты требований предоставляют исходную информацию для следующих действий:

- Детализация требований к программному обеспечению
- Управление зависимостями
- Назначение приоритетов прецедентов

Инструментальная поддержка

Rational RequisitePro содержит все необходимые средства для поддержки артефакта Атрибуты требований.

В следующем выпуске мы закончим обсуждение артефактов и подведем итоги знакомства с потоком работ Требования.