

Поток работ «Требования» (продолжение)

Артефакты управления требованиями

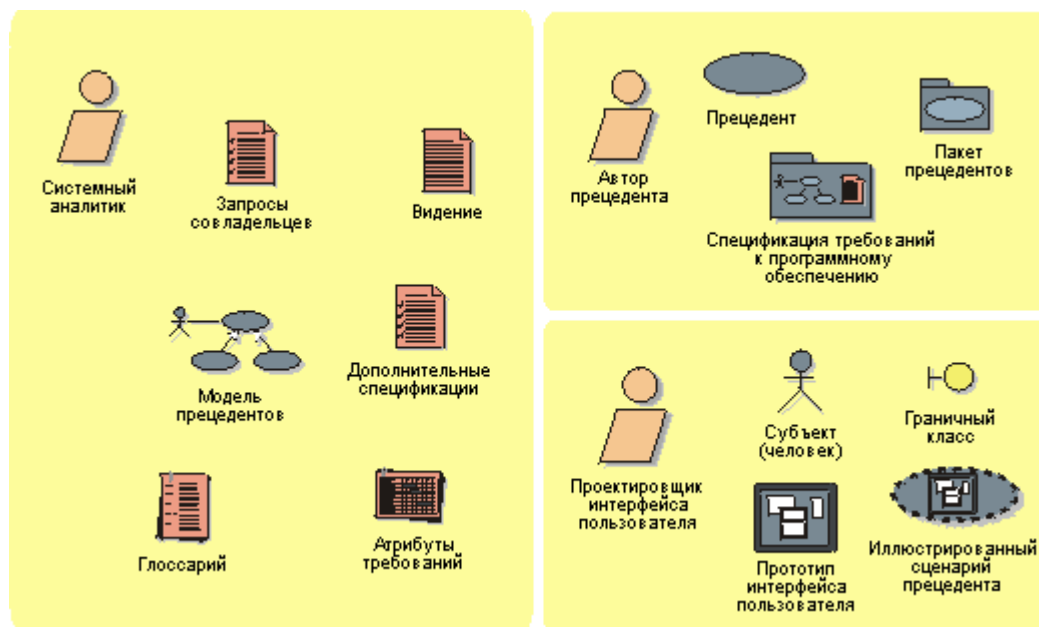
Все артефакты управления требованиями уже упоминались в предыдущем разделе “Краткий обзор потока работ”. Сейчас мы перейдем к более подробному описанию некоторых из них.

Хотелось бы организовать изложение в некотором логическом порядке, но ...

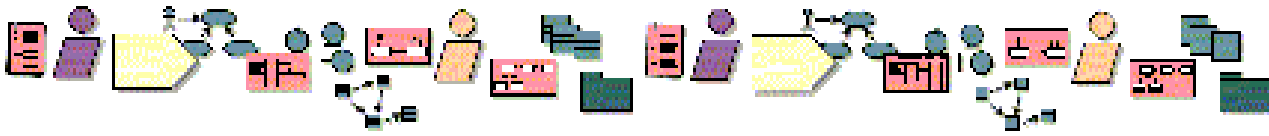
При внимательном просмотре диаграмм, иллюстрирующих поток работ, Вы могли обратить внимание, что одни и те же артефакты много раз встречаются на диаграммах и являются исходными и выходными для различных действий потока работ. Артефакты “живут активной жизнью” и изменяются (уточняются) в течение всего цикла разработки проекта. Артефакты могут разрабатываться (уточняться) параллельно, и могут быть зависимы друг от друга.

Давайте упорядочим рассмотрение артефактов следующим образом:

- Сгруппируем артефакты по работникам, ответственным за их создание и поддержание в актуальном состоянии:



Артефакты потока работ “Требования” и работники, ответственные за их создание и поддержание в актуальном состоянии



-
- В каждой группе будем рассматривать артефакты в том порядке, в котором они, скорее всего, будут создаваться (иницироваться) в совершенно новом проекте, начинаемом с чистого листа группой, впервые встретившейся с этой предметной областью.

Работник: Системный аналитик

Системный аналитик ведет и координирует выявление требований и моделирование прецедентов, выделяя функциональные возможности и границы системы. Например, устанавливая, какие субъекты и прецеденты существуют и как они взаимодействуют.

Артефакт: Глоссарий

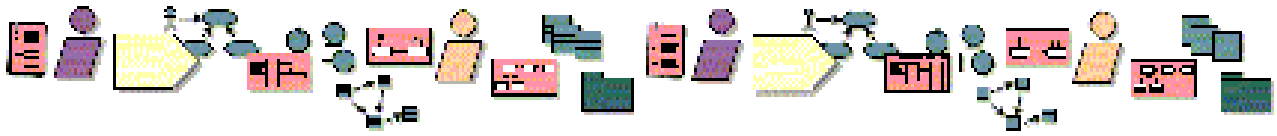
Краткий обзор

В потоке работ Требования Вы должны определить общий словарь, используя наиболее общие термины из прикладной области. Далее Вы должны постоянно использовать общий словарь во всех текстовых описаниях системы, особенно в описаниях прецедентов. Таким образом, Вы сохраняете текстовые описания непротиворечивыми, и избегаете недоразумений среди участников проекта относительно использования и значения терминов. Вы должны документировать словарь в Глоссарии.

Для поиска общих терминов в прикладной области рассмотрите термины, используемые в требованиях и употребляемые в группе разработки системы. Сосредоточьтесь на терминах, описывающих следующие концепции:

- Деловые объекты, представляющие концепции, используемые в повседневной работе организации или в оперативной среде будущей системы. Во многих случаях список концепций этого вида уже существует.
- Реальные объекты, которые должна знать система. Это объекты естественного происхождения, такие как автомобиль, зажим, баллон, самолет, пассажир, заповедник или счет.
- События, которые должна знать система. Это может быть момент времени или хронологический инцидент, о котором система должна знать, типа назначенной встречи или факта появления ошибки.

Каждый термин обычно описывается как существительное с определением. Термины должны быть в единственном числе: «заказ» и «задание», а не «заказы» и «задания». Все заинтересованные стороны должны договориться об определениях терминов.



Синхронизация

Общий словарь, определенный в Глоссарии, должен постоянно использоваться во всех текстовых описаниях системы и при переговорах с совладельцами.

В некоторых проектах Глоссарий является единственным артефактом, описывающим деловую сферу проекта. Это случается тогда, когда ни деловое моделирование, ни моделирование предметной области не выполняется.

Поэтому очень важно как можно раньше договориться об общей терминологии.

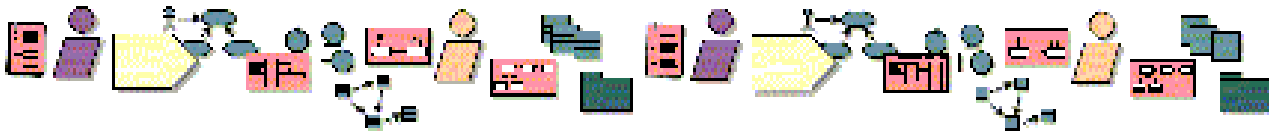
Глоссарий создается в стадии Начало, и к концу стадии Уточнение, в основном, стабилизируется.

Создание и уточнение Глоссария происходит в результате следующих действий:

- Фиксация общего словаря
- Назначение приоритетов прецедентов
- Рецензирование модели деловых прецедентов
- Обзор требований

Глоссарий предоставляет исходную информацию для следующих действий:

- Архитектурный анализ
- Детализация делового прецедента
- Детализация прецедента
- Детализация требований к программному обеспечению
- Поиск субъектов и прецедентов
- Поиск деловых субъектов и прецедентов
- Поиск деловых работников и сущностей
- Рецензирование модели деловых объектов
- Рецензирование модели деловых прецедентов
- Обзор требований
- Структурирование модели деловых прецедентов
- Структурирование модели прецедентов
- Анализ прецедента



Инструментальная поддержка

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

Шаблон Глоссарий.dot представлен в качестве отдельного приложения к этому выпуску.

Однако Вы можете не ограничиваться традиционным использованием этого шаблона для составления документа Word. Rational RequisitePro существенно расширяет возможности работы со словарем.

Rational RequisitePro позволяет выполнять следующие операции:

1. Создание документа Глоссарий.
2. Выполнение документа Глоссарий.
3. (Не обязательно) Маркирование требований в документе Глоссарий.

Рассмотрим несколько подробнее последнюю возможность.

Глоссарий не содержит требований по существу; однако, могут быть случаи, когда некоторые термины Глоссария можно использовать как названия для классов, особенно классов сущностей. Отмечая эти термины как требования, Вы можете автоматически создавать классы того же названия в Rational Rose с использованием Rational Synchronizer.

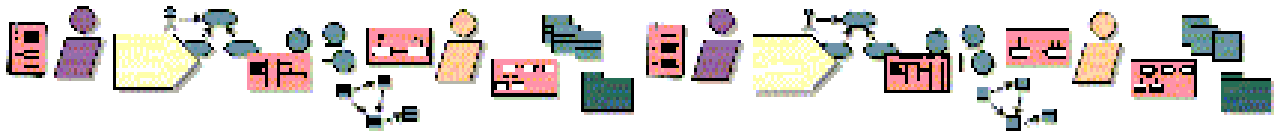
Вы можете также захотеть отслеживать факты переопределения терминов в Глоссарии и модифицировать некоторые документы, чтобы отразить эти изменения. Маркировка таких терминов как требований позволит Вам позже создать связи трассируемости между терминами Глоссария и возможностями изделия или требованиями к системе, которые должны быть перефразированы после того, как определение термина Глоссария было модифицировано.

Любой термин Вы можете пометить как требование типа Glossary Requirement Type (TERM). Это требование будет сохранено в базе данных Rational RequisitePro. Впоследствии Вы сможете установить нужные связи и трассировки с другими объектами требований.

Контрольные точки

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

- Все ли термины имеют ясное и краткое определение?
- Каждый ли термин глоссария включен где-нибудь в описания прецедентов? Если нет, это может означать, что отсутствует нужный прецедент или что существующие прецеденты не завершены. Более вероятно, что термин не включен потому, что в этом нет необходимости. В этом случае его нужно удалить.
- Согласованно ли используются термины в кратких описаниях субъектов и прецедентов?



- Означает ли термин одно и то же во всех прецедентах?

Пример артефакта: Введение в RUP. Глоссарий

Рабочая группа проекта OPFS Classics, Ink. (см. стр. 9-1) разработала Глоссарий, в котором определены используемые в проекте термины. Но, я думаю, знакомиться с ними нет практического смысла. Лучше взять пример поближе к интересующей нас предметной области.

В качестве примера предлагается рассмотреть фрагмент словаря терминов, представленного в глоссарии интерактивной версии Rational Unified Process. Я надеюсь, что этот фрагмент будет расширяться, и к моменту завершения серии статей охватит весь глоссарий.

Документ Глоссарий.doc представлен в качестве отдельного приложения к этому выпуску.

Артефакт: Видение

Краткий обзор

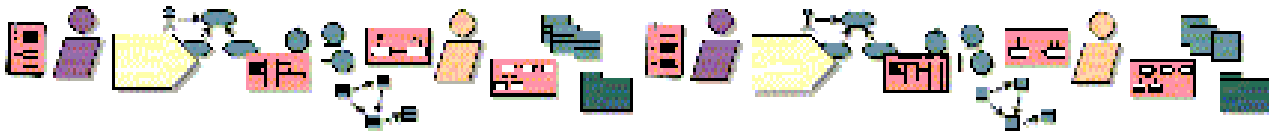
Документ Видение фиксирует требования и проектные ограничения очень высокого уровня, давая читателю понимание разрабатываемой системы. Он обеспечивает исходные данные для процесса рассмотрения проекта, и поэтому тесно связан с документом Деловые обстоятельства (см. стр. 6.3).

Видение сообщает основные «зачем и как», связанные с проектом, и критерии, на основании которых должны быть приняты все будущие решения.

Видение предоставляет основной исходный материал для последующих договоренностей о большей детализации технических требований. Может существовать также и формальная спецификация требований.

При разработке Видения выполняются следующие шаги:

- Достижение соглашения по решаемой проблеме
- Идентификация совладельцев
- Определение границ системы
- Идентификация ограничений, которые будут наложены на систему
- Формулирование постановки задачи
- Определение возможностей системы
- Оценка результатов



1. Достижение соглашения по решаемой проблеме

Один из самых простых способов достигнуть соглашения по определению проблемы – записать ее и посмотреть, все ли с этим согласны.

Спросите Ваших заказчиков или их представителей: В чем состоит проблема?

Затем повторите вопрос несколько иначе: Почему возникла эта проблема? В чем состоит проблема в действительности?

Не принимайте первую формулировку проблемы. Продолжайте спрашивать «почему?» Отыщите коренные причины или «проблему позади проблемы». Реальная проблема часто скрывается за тем, что воспринимается как проблема.

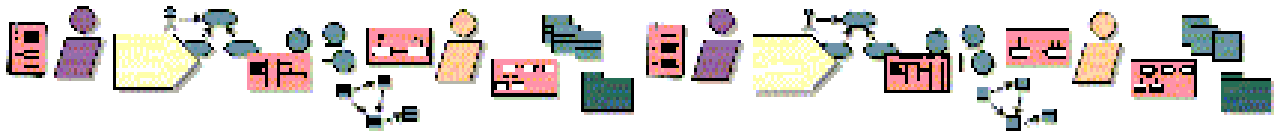
Иногда интервьюируемые могут так сосредоточиться на предполагаемом решении, что это заставит их сразу сформулировать основную проблему. В таком случае может быть полезно исследовать выгоды от этого решения, а затем попытаться найти проблемы, решаемые таким образом. Затем Вы можете исследовать, действительно ли эти проблемы являются «реальными» проблемами в организации.

Простой метод поиска проблемы позади проблемы – использование диаграмм «рыбьего скелета» или Парето. Оба метода описаны в разделе Рабочие рекомендации интерактивной версии Rational Unified Process.

2. Идентификация совладельцев

В зависимости от степени знакомства группы разработки с проблемной областью, идентификация совладельцев может быть шагом тривиальным или нетривиальным. Часто это просто интервью у людей, принимающих решения, потенциальных пользователей и других заинтересованных сторон. Полезно задать им следующие вопросы:

- Кто является пользователями системы?
- Кто является выгодным покупателем системы?
- На что еще будут воздействовать результаты, которые производит система?
- Кто оценит и благословит систему, когда она будет поставлена и развернута?
- Имеются ли любые другие внутренние или внешние пользователи системы, чьи потребности должны быть удовлетворены?
- Кто будет лоббировать новую систему?
- Имеется ли кто-либо еще из заинтересованных лиц?
- Хорошо, а имеется ли кто-либо еще?



3. Определение границ системы

Граница системы определяет границу между системой и реальным миром, который окружает систему. Информация, в форме вводов и выводов, циркулирует взад и вперед между системой и пользователями, которые находятся вне системы. Все взаимодействия с системой происходят через интерфейсы между системой и внешним миром.

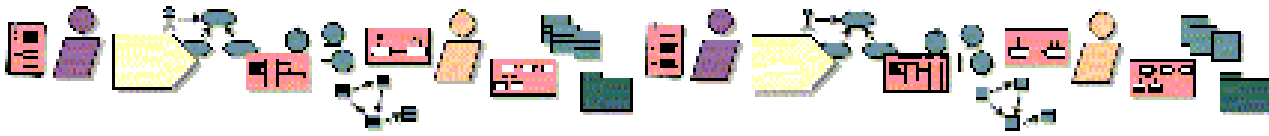
Во многих случаях границы системы очевидны. Например, границы пользователя, который персонально работает на Windows 2000®, относительно хорошо определены. Имеется только один пользователь и одна платформа. Интерфейсы между пользователем и прикладной программой состоят из диалогов интерфейса пользователя, к которым пользователь обращается, чтобы ввести информацию в систему, и каких-либо выводимых отчетов и путей связи, которые система использует для документирования или передачи итоговой информации.

Обычно для того, чтобы определить и описать границы системы, очень полезно использовать субъекты. О субъектах мы будем говорить в одном из следующих выпусков.

4. Идентификация ограничений, которые будут наложены на систему

Существует множество источников ограничений, которые должны быть рассмотрены. Ниже представлен список потенциальных источников ограничений и вопросов, которые должны быть выяснены:

- **Политический:** Имеются ли внутренние или внешние политические проблемы, которые затрагивают потенциальные решения? Межведомственные проблемы?
- **Экономический:** Какие финансовые или бюджетные ограничения применяются? Учитывается ли стоимость проданных товаров или изделий? Лицензирование выпусков?
- **Среды:** Имеются ли ограничения или регуляторы среды? Правовые? Другие стандарты, которыми мы ограничены?
- **Технический:** Ограничены ли мы в нашем выборе технологий? Вынуждены ли мы работать в рамках существующих платформ или технологий? Запрещено ли использование новых технологий?
- **Выполнения:** Определен ли график? Ограничены ли мы существующими ресурсами? Можем ли мы использовать внешнюю рабочую силу? Можем ли мы расширить ресурсы? Временными работниками? Постоянными?
- **Системный:** Решение должно базироваться на наших существующих системах? Мы должны поддержать совместимость с существующими решениями? Какие операционные системы и среды должны поддерживаться?



5. Формулирование постановки задачи

Полным составом группы поработайте над таблицами и заполните следующий шаблон для каждой проблемы, которую Вы идентифицировали:

Проблема: <описание проблемы>

затрагивает: <совладельцы, которых затрагивает проблема>.

Воздействие: <как воздействует проблема>.

Успешное решение: <список некоторых ключевых выгод от успешного решения>.

Цель этого шаблона состоит в том, чтобы помочь Вам отличить решения/ответы от проблем/вопросов.

Пример:

Проблема: несвоевременное и неверное решение вопросов послепродажного обслуживания

затрагивает: наших заказчиков, представителей поддержки заказчика и специалистов по обслуживанию оборудования.

Воздействие: неудовлетворенность заказчика, ощутимый недостаток качества, недовольство пользователей и потери дохода.

Успешное решение: обеспечит в реальном времени доступ к базе данных для устранения неисправности представителями поддержки и облегчит своевременную отправку специалистов по обслуживанию оборудования только туда, где действительно необходима их помощь.

6. Определение возможностей системы

Основываясь на выгодах, перечисленных в ваших постановках проблем, разработайте список возможностей, которые Вы хотите иметь в системе. Опишите их кратко, и дайте им атрибуты, чтобы помочь определить их общее состояние и приоритет в проекте. Подробнее об атрибутах см. Артефакт: Атрибуты требований в одном из следующих выпусков.

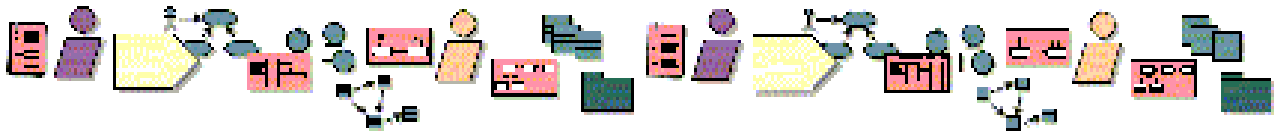
7. Оценка результатов

На этом шаге Вы должны проконтролировать документ Видение, чтобы проверить, как идет ваша работа, но не рассматривать его подробно. Ознакомьтесь с контрольными точками для документа Видение ниже в этом разделе.

Синхронизация

Документ Видение создается как можно раньше в стадии Начало и используется как основание для написания документа Деловые обстоятельства (см. стр. 6-3) и первого наброска списка рисков (см. стр. 6-4).

Документ Видение служит исходным для моделирования прецедентов, он модифицируется и поддерживается как отдельный артефакт на протяжении всего проекта.

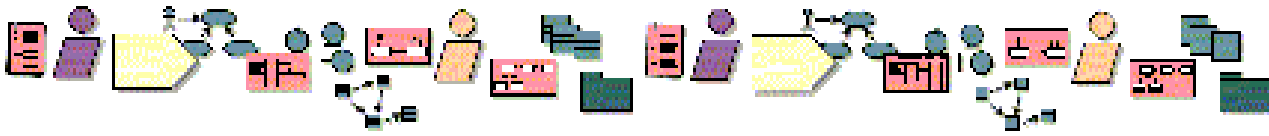


Создание и уточнение Видения происходит в результате следующих действий:

- Оценка итерации
- Разработка Видения
- Управление зависимостями
- Обзор требований

Видение предоставляет исходную информацию для следующих действий:

- Фиксация общего словаря
- Составление плана разработки программного обеспечения
- Определение организации проекта и укомплектование персоналом
- Детализация делового прецедента
- Детализация прецедента
- Детализация требований к программному обеспечению
- Изложение деловых обстоятельств
- Разработка плана итерации
- Разработка плана измерений
- Разработка плана приемки изделия
- Разработка плана обеспечения качества
- Разработка материалов для обучения
- Выявление запросов совладельцев
- Поиск субъектов и прецедентов
- Поиск деловых субъектов и прецедентов
- Поиск деловых работников и сущностей
- Управление зависимостями
- Моделирование интерфейса пользователя
- Назначение приоритетов прецедентов
- Рецензирование оценки проекта
- Рецензирование плана проекта
- Обзор требований
- Написание плана СМ



- Написание примечаний к выпуску

Инструментальная поддержка

Так же, как и в случае с артефактом Глоссарий, интерактивная версия Rational Unified Process содержит шаблон Microsoft Word для документа Видение.

Шаблон Видение.dot представлен в качестве отдельного приложения к этому выпуску.

Кроме того, если у Вас есть Rational RequisitePro, Вы можете выполнять следующие операции:

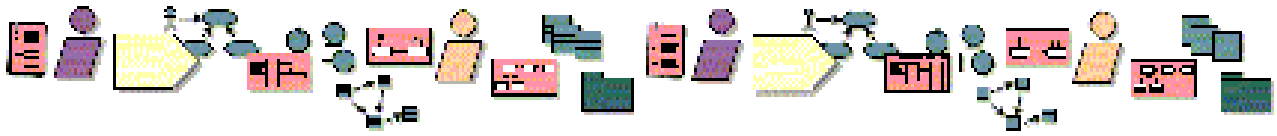
1. Создание документа Видение.
2. Выполнение документа Видение.
3. Маркирование требований в документе Видение.

Рассмотрим последнюю операцию.

RequisitePro позволяет создавать требования из возможностей, выделенных в разделе Возможности изделия. Это даст Вам возможность управлять атрибутами требований в матрице атрибутов, и поможет проследить связи трассируемости между возможностями изделия и требованиями к программе.

Для создания требования в документе Видение, если у Вас есть RequisitePro, сделайте следующее:

1. В разделе Возможности изделия документа Видение выберите текст, который определяет требование.
2. В рабочей области Word нажмите RequisitePro > Requirement > Create. Появляется диалоговое окно Requirement.
3. Выберите тип требования возможности (FEAT).
4. Нажмите на закладку Attributes и установите атрибут приоритета (Priority) в одно из предопределенных значений: высокий (High), средний (Medium) или низкий (Low). Очень полезно записывать приоритеты совладельца при выполнении Действия: Выявление запросов совладельцев.
5. Перейдите к атрибуту Origin и установите инициатора этой конкретной возможности. Это поможет Вам отслеживать совладельцев, предложивших реализовать определенные возможности изделия, способствует контактам с этими совладельцами для разрешения неясных вопросов или переговоров. Очень полезно записывать названия совладельца при выполнении Действия: Выявление запросов совладельцев.
6. Выберите атрибут Rationale и напечатайте причину для возможности. При работе над этой возможностью Вам будет легче сосредоточиться на выполнении назначенной потребности.



7. Нажмите ОК.
8. Повторите эти шаги для каждой возможности изделия, выделенной в разделе Возможности изделия документа Видение.

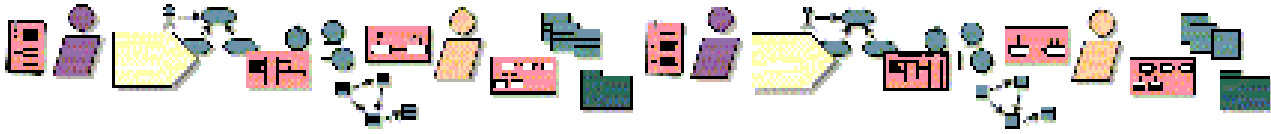
Для просмотра требований и атрибутов:

1. На панели инструментов нажмите Window > Show Views.
2. В рабочей области Views нажмите File > New View.
3. Выберите тип представления Attribute Matrix и тип требования FEAT, который Вы использовали в вашем документе Видение.
4. Нажмите ОК. Вы можете просматривать ваши требования возможностей и значения их атрибутов.

Контрольные точки

Если Вы сможете ответить «Да» на все перечисленные ниже вопросы, Вы разработали хороший документ Видение:

- Полностью ли исследованы «проблемы позади проблемы?»
- Правильно ли сформулирована постановка задачи?
- Является ли список совладельцев полным и правильным?
- Все ли согласились с определением границ системы?
- Если границы системы были выражены субъектами, все ли субъекты определены и правильно описаны?
- Достаточно ли исследованы ограничения, которым подчиняется система?
- Все ли виды ограничений учтены, например политические, экономические, окружающей среды и т.п.?
- Все ли главные возможности системы идентифицированы и определены?
- Все ли проблемы будут разрешены при реализации идентифицированных возможностей?
- Совместимы ли возможности с идентифицированными ограничениями?



Пример артефакта: Система обработки и выполнения заказов.

Видение

Рабочая группа проекта OPFS Classics, Ink. (см. стр. 9-1) разработала документ Видение. Данный документ представлен в качестве отдельного приложения к этому выпуску (см. файл OPFS_Vision.doc).

При рассмотрении документа нужно учитывать следующее:

Это всего лишь учебный пример. Не будьте к нему слишком строгим! Он предназначен для того, чтобы посмотреть, что и как принято писать в этом документе.

Этот документ исполнялся без использования Rational RequisitePro, и, следовательно, не имеет действующих меток требований. Эти метки имитируются двойным подчеркиванием текста требований (как принято в RequisitePro). В RequisitePro требования помечаются, кроме того, номерами (например, FEAT1.4).

В следующем выпуске мы продолжим обсуждение артефактов требований.