

Поток работ «Требования»

Введение

Цели потока работ

Rational Unified Process формулирует следующие цели для основного потока работ Требования:

- Добиться одинакового понимания с заказчиком и пользователями о том, что должна делать система.
- Дать разработчикам системы наилучшее понимание требований к системе.
- Определить границы системы.
- Обеспечить базу для планирования технического содержания итераций.
- Определить интерфейс пользователей с системой.

Для достижения этих целей необходимо собрать и проанализировать **Запросы совладельцев**, разработать артефакты **Видение**, **Модель прецедентов**, **Прецеденты** и **Дополнительные спецификации**, которые описывают, что будет делать система.

Документ Запросы совладельцев - это «список пожеланий» различных совладельцев проекта (заказчиков, пользователей, других заинтересованных лиц) относительно содержания системы, вместе с информацией о том, как каждое пожелание рассмотрено в проекте.

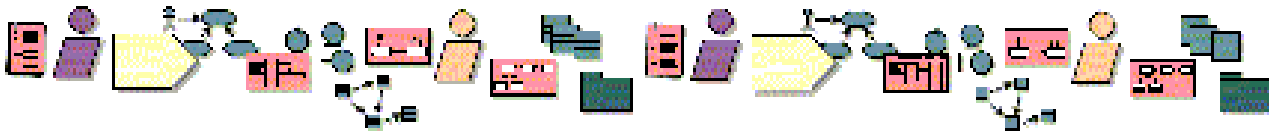
Документ Видение описывает все цели проекта и основные возможности системы.

Модель прецедентов демонстрирует одинаковое понимание системы заказчиком, пользователями и разработчиками. Она позволяет:

- Заказчику и пользователям проверить, будет ли система такой, как они ожидали.
- Разработчикам системы делать именно то, что ожидают заказчик и пользователи.

Модель прецедентов состоит из прецедентов и субъектов. Каждый прецедент в модели подробно, по шагам, описывает, как система взаимодействует с субъектами, и что система делает в прецеденте. Прецеденты функционируют как единая «красная нить» на всем протяжении жизненного цикла программы; одна и та же модель прецедентов используется при анализе системы, проектировании, выполнении и испытаниях (см. стр. 4-6).

Дополнительные спецификации - важное дополнение к модели прецедентов, так как вместе они



фиксируют все необходимые требования к программному обеспечению (функциональные и не функциональные), которые составляют полные технические требования к системе.

Законченные определения прецедентов и Дополнительные спецификации для конкретной «возможности» или некоторой части системы могут объединяться в **Спецификации требований к программному обеспечению** (Software Requirements Specification - SRS).

В дополнение к уже упомянутым, должны быть разработаны следующие артефакты:

- **Глоссарий**
- **Иллюстрированный сценарий прецедента**
- **Прототип интерфейса пользователя**

Глоссарий важен потому, что он определяет общую терминологию для всех моделей и текстовых описаний требований к системе.

Иллюстрированные сценарии прецедентов и прототип интерфейса пользователя - это результаты моделирования интерфейса пользователя и прототипирования, которые выполняются параллельно с другими действиями разработки требований.

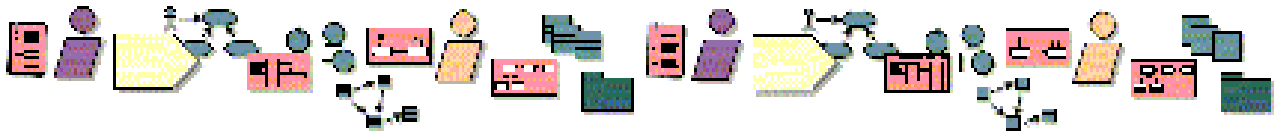
Поток работ Требования связан с другими потоками работ процесса.

- Поток работ **Деловое моделирование** представляет организационный контекст системы.
- Поток работ **Анализ и проектирование** получает из Требования свои первичные исходные данные (модель прецедентов и глоссарий). В ходе анализа и проектирования могут быть обнаружены недостатки модели прецедентов; в этих случаях генерируются и применяются к модели прецедентов **запросы изменения**.
- Поток работ **Испытание** тестирует систему, чтобы проверить код модели прецедентов. Прецеденты и Дополнительные спецификации обеспечивают исходную информацию для требований, используемых при планировании и проектировании испытаний.
- Поток работ **Среда** разрабатывает и обслуживает артефакты поддержки, которые используются при моделировании прецедентов, типа **Указаний по моделированию прецедентов** и **Указаний по разработке интерфейса пользователя**.
- Поток работ **Управление проектом** планирует проект, разрабатывая **План управления требованиями**, и каждую итерацию (разрабатывая **План итерации**). Модель прецедентов важна для планирования действий итерации.

Требования

Требование – это условие или возможность, которой должна соответствовать система.

Существует много различных видов классификаций требований. Rational Unified Process



использует для этой цели так называемую **FURPS+** модель, использующую акроним FURPS для обозначения главных категорий требований с их подкатегориями, как показано ниже:

- **Functionality** (Функциональность)
- **Usability** (Применимость)
- **Reliability** (Надежность)
- **Performance** (Производительность)
- **Supportability** (Пригодность к эксплуатации)

Значок «+» в названии FURPS+ помогает Вам не забыть включать также и такие требования, как

- проектные ограничения,
- требования выполнения,
- требования к интерфейсу и
- физические требования.

(См. стандарт IEEE Std 610.12.1990)

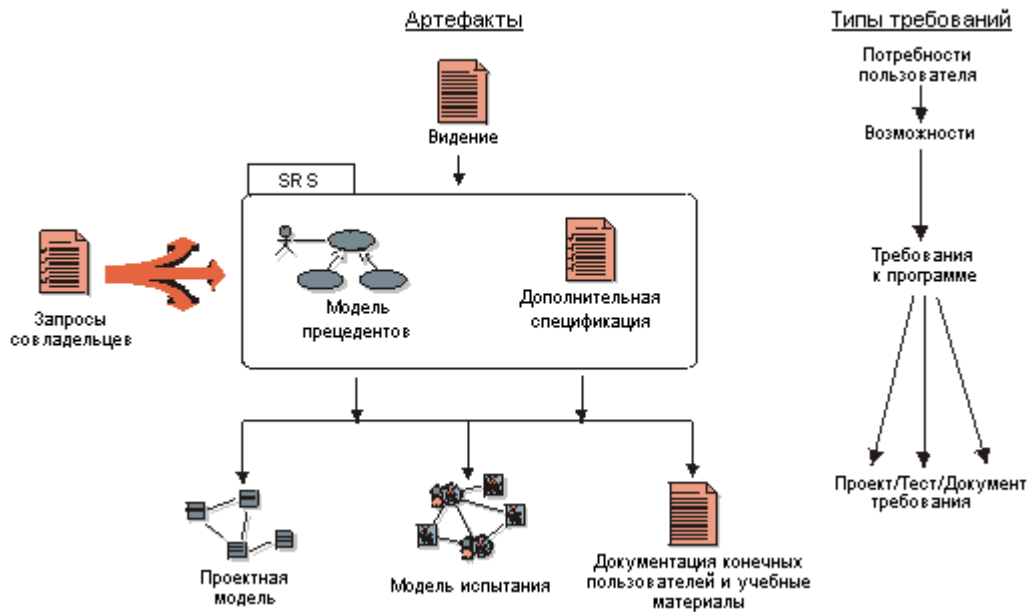
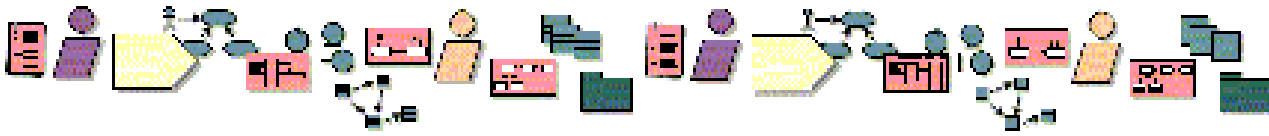
Функциональные требования определяют действия, которые должна быть способна выполнить система, без рассмотрения физических связей. Лучше всего они описываются в модели прецедентов и в прецедентах. Функциональные требования, таким образом, определяют внешнее поведение системы типа “ввода – вывода”.

Многие требования не являются функциональными. Они описывают только атрибуты системы или атрибуты среды. Такие требования называются **не функциональными требованиями**.

Я думаю, Вы уже успели убедиться, что одной из причин неудач традиционных процессов разработки программных систем (например, процесса «водопада»), является представление требований неизменными. Rational Unified Process с самого начала отвергает это предположение и вводит понятие «управление требованиями». Управление требованиями – это систематический подход к обнаружению, документированию, организации и сопровождению изменяющихся требований к системе (см. стр. 4-12).

Традиционно требования рассматривались как текстовые предписания, касающиеся одной из категорий, упомянутых выше на этой странице. Каждое требование – это условие или возможность, которой должна соответствовать система.

Для эффективного управления требованиями мы должны больше знать о том, что мы поддерживаем как требования. Rational Unified Process вводит понятие **типов требований**, чтобы разделить различные уровни абстракции и целей представления требований. Каждый тип требования должен иметь уникальный набор атрибутов, связанных с каждым требованием, определенным на этом уровне.



Уровни абстракции и типы требований

Мы должны следить за неоднозначными пожеланиями и за формальными запросами наших совладельцев, чтобы удостовериться, что мы знаем, как о них позаботиться. Документ Видение помогает нам отслеживать ключевые потребности пользователя и возможности системы.

Указания Видения проекта определяют необходимые пользователям возможности программы и требования к ней. Прецеденты или другие формы модели и Дополнительные спецификации управляют требованиями к проекту.

Требования разделены на функциональные и не функциональные требования, и представлены в моделях и диаграммах анализа и проектирования.

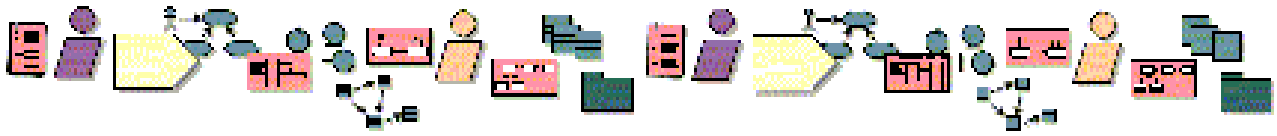
Требования к испытаниям, производные от вышеупомянутых, - это процедуры тестирования и другие артефакты испытаний.

Точно так же появляются требования, касающиеся содержания документации поддержки конечных пользователей и учебных материалов.

Трассируемость требований

Цель установления **трассируемости** требований состоит в том, чтобы помочь:

- Оценить последствия для проекта, вызванные изменением требования



-
- Оценить последствия неудачного испытания требования (то есть, если испытание терпит неудачу, требование не может быть удовлетворено)
 - Управлять контекстом проекта
 - Проверять, что все требования к системе выполнены
 - Проверять, что приложение делает только то, что оно должно было делать
 - Управлять изменениями

Рассмотрите еще раз рисунок на стр. 11-4.

Трассируемость помогает Вам проследить, как запросы совладельцев оттранслированы в набор свойств в документе Видение и в Модель прецедентов, которые в свою очередь детализированы в прецедентах и в Дополнительных спецификациях.

Далее Вы должны проследить, как эти детальные технические требования оттранслированы в проект, как они проверены и как они документированы для пользователя.

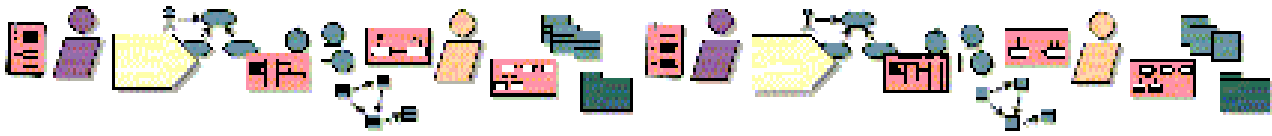
Трассируемости должны быть установлены так, чтобы они помогли Вам ответить на следующий типовой набор вопросов:

- Покажите мне потребности пользователя, которые не связаны с возможностями программы.
- Покажите мне состояние испытаний по всем прецедентам в итерации #n.
- Покажите мне все дополнительные требования, связанные с испытаниями, состояние которых не проверено.
- Покажите мне результаты всех испытаний, которые потерпели неудачу, в порядке их критичности.
- Покажите мне свойства, намеченные для этого выпуска, в удовлетворении которых нуждается пользователь, и их состояние.

Инструментальная поддержка

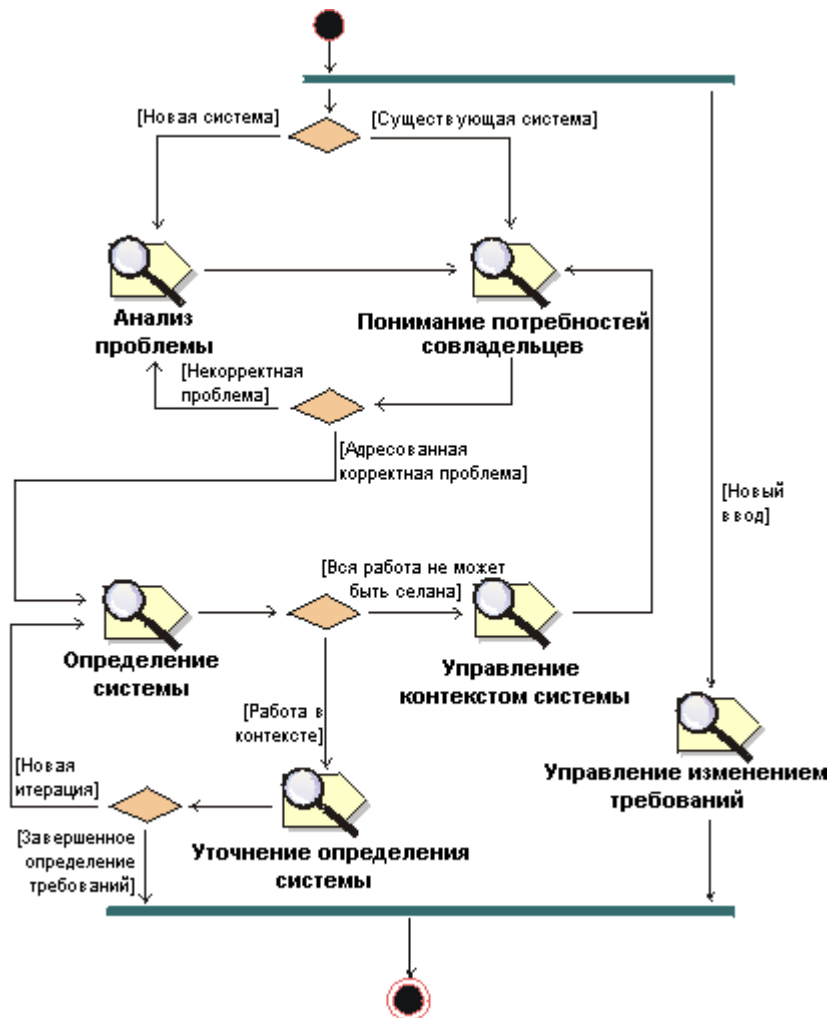
Для поддержки трассируемости требований Rational Unified Process предлагает использовать продукт Rational Requisite Pro (см. стр. 5-7).

Ключевым понятием Rational Requisite Pro в помощи для управления изменениями требований является понятие «сомнительной» связи трассируемости. Когда требование (или другой элемент трассируемости) изменяется с одного из концов связи трассируемости, все ссылки, связанные с этим требованием помечаются как «сомнительные». Это пометка возлагает на ответственного работника обзор изменений и определение того, должны ли измениться также и связанные объекты. Это понятие помогает также при анализе конфликтов потенциальных изменений.



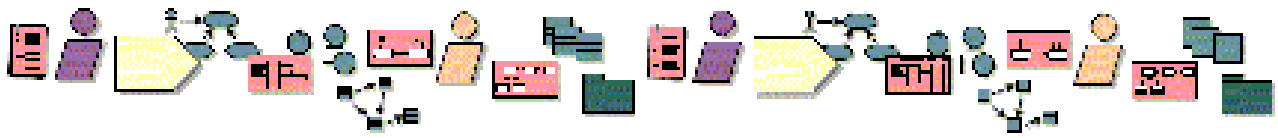
Краткий обзор потока работ

Для объяснения основного потока работ Требования интерактивная версия Rational Unified Process представляет следующую диаграмму действий:



Представление деталей потока работ Требования на диаграмме действий

Состояния действия (см. стр. 8-16) на этой диаграмме представляют детали потока работ (см. стр. 3-5).



Детали потока работ представляют ключевые навыки, которые должны применяться для эффективного управления требованиями. Детали Анализ проблемы и Понимание потребности совладельцев фокусируются на стадии Начало, а детали Определение системы и Уточнение определения системы – на стадии Уточнение. Детали Управление контекстом системы и Управление изменением требований выполняются непрерывно в течение всего проекта.

Детали потока работ представлены в порядке логической последовательности. Но они, как указано выше, применяются непрерывно в различном порядке, необходимом в проекте. На диаграмме действий они указаны в той последовательности, которая наиболее вероятна в первой итерации нового проекта.

Анализ проблемы

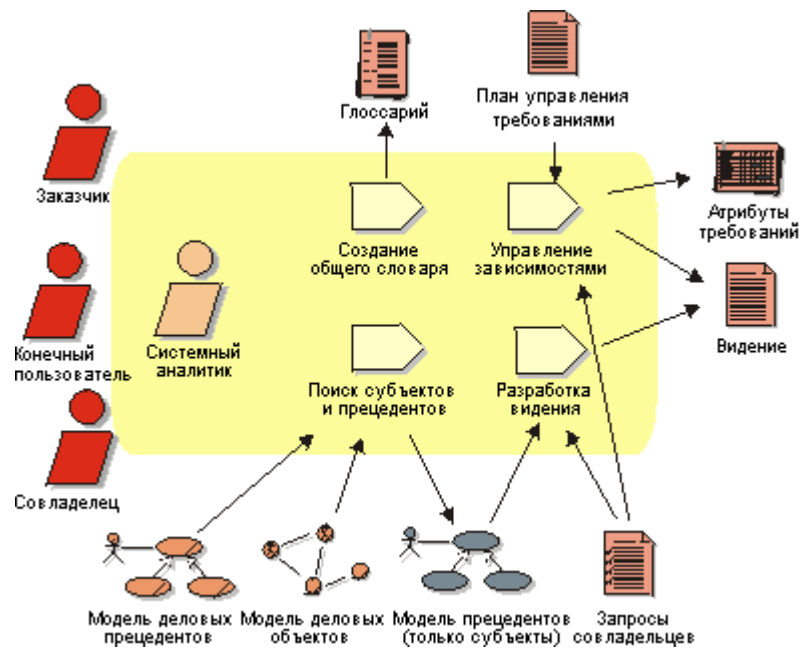
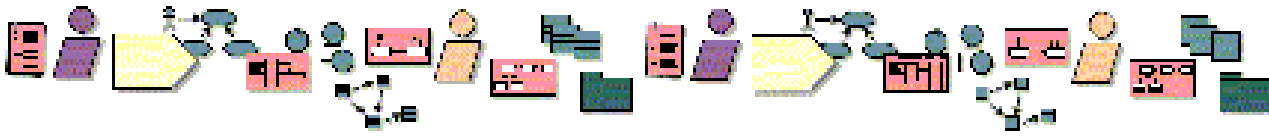


Диаграмма деталей потока работ Анализ проблемы

Целями этой части потока работ являются:

- Произвести для проекта документ Видение
- Договориться о возможностях и целях системы



Анализ проблемы выполняется для того, чтобы понять проблему, начальные потребности совладельцев и предложить решения высокого уровня. Это акт рассуждения и анализа с целью найти «проблему позади проблемы». В ходе анализа проблемы приобретает взаимное согласие о реальной проблеме и о том, кто является ее совладельцами. Кроме того, Вы определяете, какие из деловых перспектив ограничивают решение, а также деловые ограничения решений. Вы должны также проанализировать деловые прецеденты проекта так, чтобы получить хорошее понимание того, какие преимущества ожидаются от инвестиции в формируемую систему.

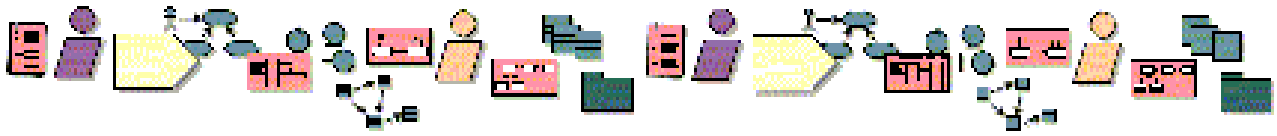
Эта часть потока работ может несколько раз повторяться в течение начала и раннего уточнения. Когда запросы от совладельцев будут поняты, тогда будут развиваться решения деловых процессов и технические решения.

Первичное действие в анализе проблемы – это разработка документа Видение, который идентифицирует высокоуровневое представление о создаваемой системе ее будущего пользователя или заказчика. В Видении начальные требования выражены как главные возможности, которыми должна обладать система, чтобы решить наиболее важные проблемы. Возможностям должны быть назначены атрибуты типа обоснования, относительного значения или приоритета, источника запроса и так далее, так, чтобы можно было начинать управлять зависимостями. По мере разработки Видения, системный аналитик идентифицирует пользователей и взаимодействующие системы - субъекты системы.

Если Вы разработали модель деловых прецедентов и модель деловых объектов, посмотрите еще раз раздел «Переход от деловых моделей к системным» на стр. 10-17.

Члены проектной группы, участвующие в анализе проблемы, должны иметь опыт общения с заказчиками и обнаружения проблемы позади проблемы. Конечно, знакомство с целевой технологией желательно, но не существенно. Требуется активное участие различных совладельцев проекта.

Для поиска проблемы позади проблемы Rational Unified Process рекомендует использовать метод мозговой атаки и диаграммы «рыбной кости» или диаграммы Парето. Все эти методы подробно изложены в разделе «Рабочие рекомендации» интерактивной версии Rational Unified Process.



Понимание потребностей совладельцев

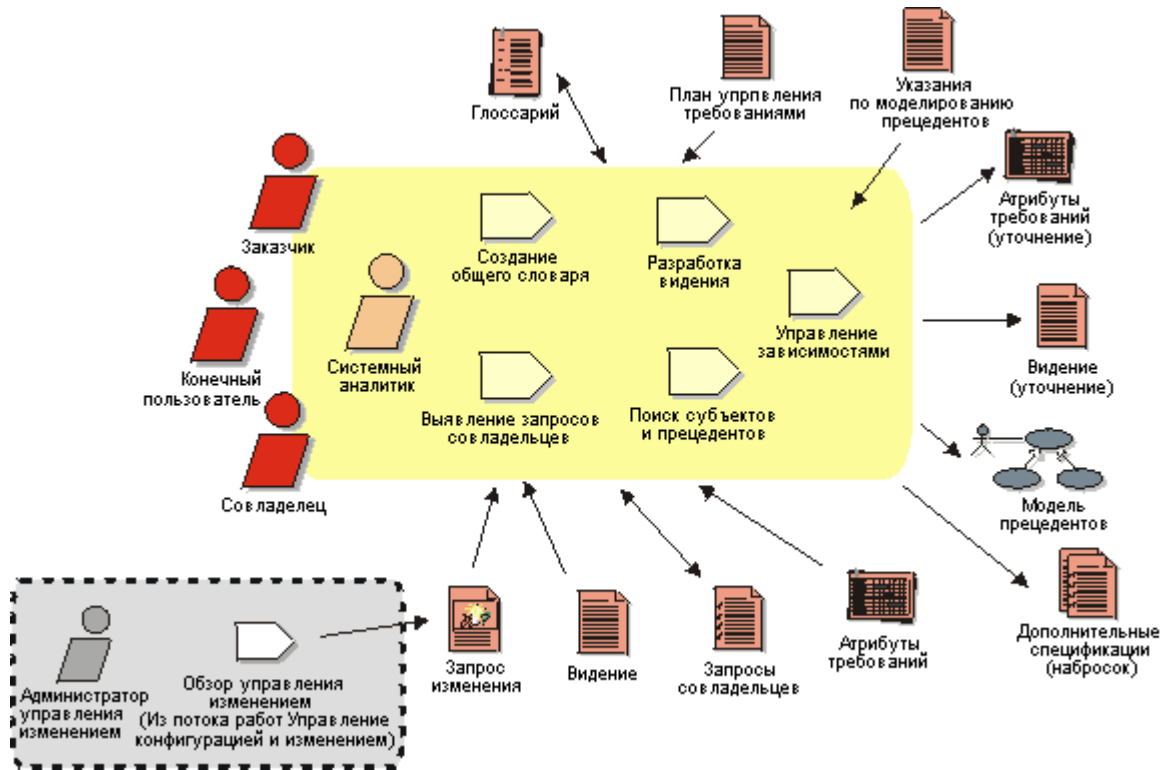
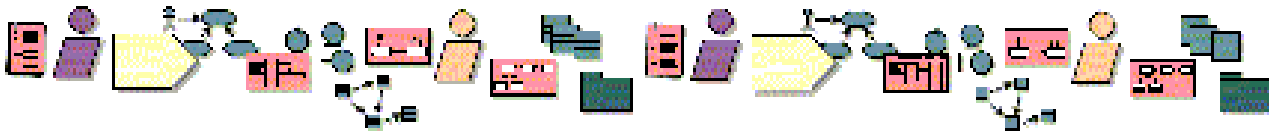


Диаграмма деталей потока работ Понимание потребностей совладельцев

Цель состоит в том, чтобы собрать и выявить информацию от совладельцев проекта. Собранные запросы совладельцев могут быть расценены как «список пожеланий», которые будут использоваться в качестве первичной информации для определения модели прецедентов, прецедентов и дополнительных спецификаций.

Как правило, это действие выполняется главным образом в итерациях стадий Начало и Уточнение, однако запросы совладельцев должны собираться в течение всего проекта, используя Запросы изменения в процессе Управления запросом изменения.

Требования происходят из многих источников, примерами которых могут быть заказчики, партнеры, конечные пользователи и специалисты предметной области. Вы должны знать, как лучше установить, какими должны быть эти источники и как получить к ним доступ, а также как лучше получить от них информацию. Если вы разрабатываете информационную систему, которую нужно использовать внутри вашей компании, Вы можете включить в вашу группу разработки людей с опытом конечного пользователя и знанием предметной области. Достаточно часто проводят обсуждения на уровне деловой модели. Если вы разрабатываете изделие на



продажу, Вы можете привлекать к работе ваших специалистов по маркетингу, чтобы лучше понять потребности заказчиков на рынке.

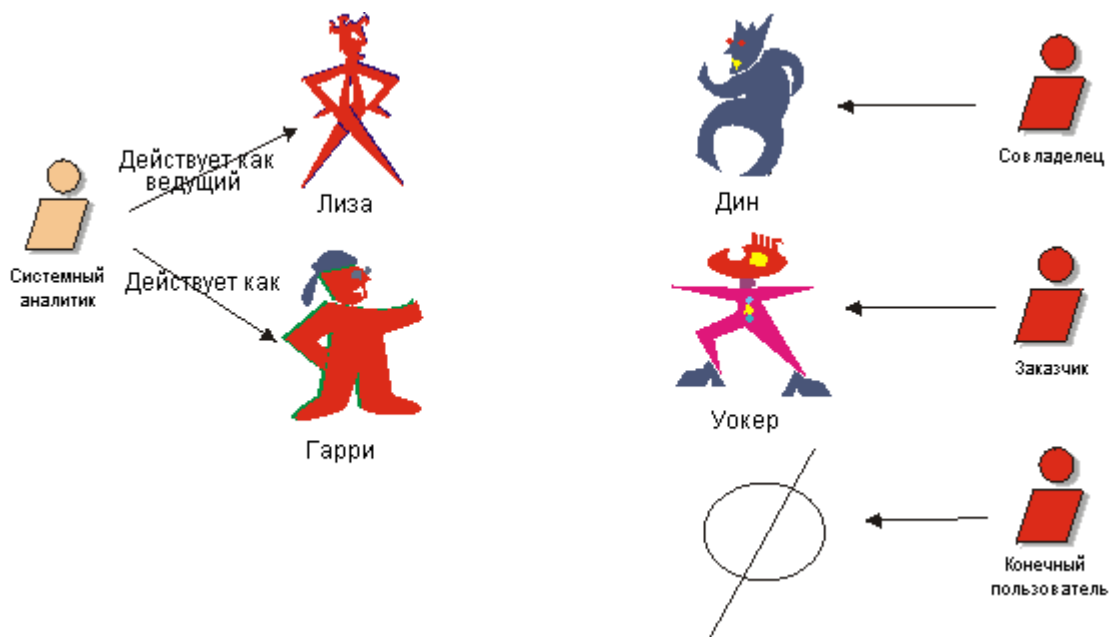
Первичные результаты - коллекции расположенных по приоритетам Запросов совладельцев, которые позволяют усовершенствовать документ Видение и лучше понять атрибуты требований. Кроме того, в течение этого потока работ Вы можете начать обсуждение системы в терминах ее прецедентов и субъектов. Другой важный выход - модифицированный Глоссарий терминов, который устанавливает общий словарь для членов группы.

Члены проектной группы, участвующие в этой части потока работ, должны иметь опыт общения с заказчиками и выявления информации. Конечно, знакомство с целевой технологией желательно, но не существенно.

Действия по выявлению информации могут происходить с использованием методов интервью, мозговой атаки, концептуального прототипирования, анкетных опросов и анализа конкурентоспособности.

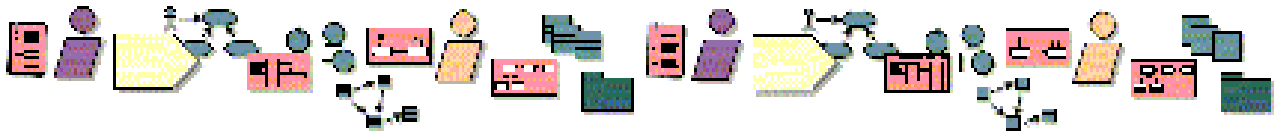
Пример

В стадии Начало проекта АТМ (автоматической кассовой машины) Лиза и Гарри действуют как системные аналитики.

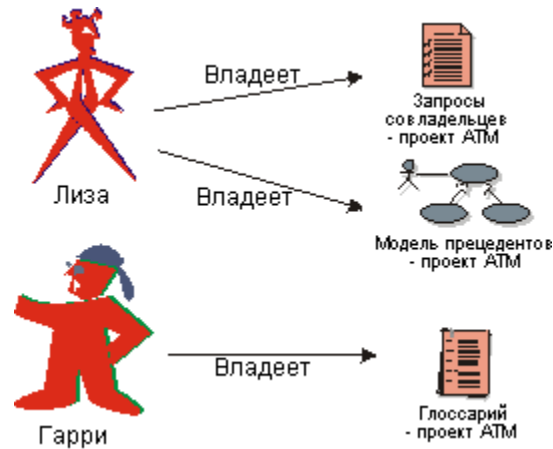


Участники деталей потока работ Понимание потребности совладельцев

Лиза назначена ведущим системным аналитиком, так как она имеет предшествующий опыт проведения различных семинаров по требованиям. Дин - один из поборников проекта АТМ,



и поэтому был приглашен на семинар как совладелец. Уокер представляет операторов АТМ, которые будут использовать новую систему. Они не приглашали специально конечных пользователей, но это не вызывало никакого беспокойства, так как все члены этой группы имели опыт использования банкоматов.



С этих пор Лиза и Гарри действуют как системные аналитики. Оба они отвечают за завершение артефактов этого шага. Лиза назначена владельцем документа Запросы совладельцев и модели прецедентов, а Гарри отвечает за Глоссарий.

Определение системы

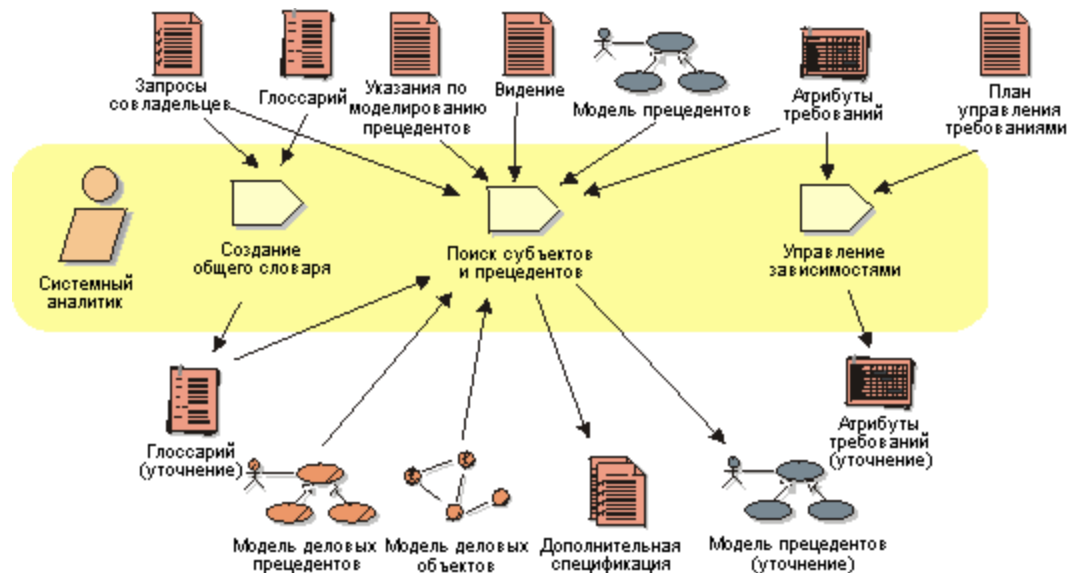
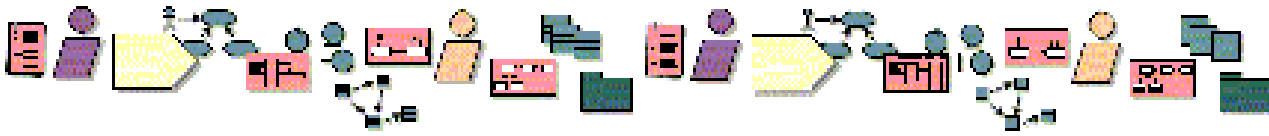


Диаграмма деталей потока работ Определение системы



Целями этой части потока работ являются:

- Добиться одинакового понимания системы всеми участниками проектной группы.
- Выполнить высокоуровневый анализ результатов сбора данных о потребностях совладельцев.
- Более формально документировать результаты анализа требований в моделях и документах.

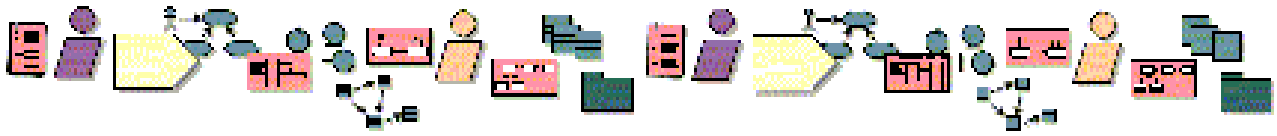
Как правило, эта часть потока работ выполняется только в итерациях стадий Начало и Уточнение.

Для определения системы способами трансляции и реорганизации определения запросов совладельцев должно быть сформировано выразительное описание системы. В начале определения системы решения строятся на том, что составляют требования, формат документирования, формальности языка, степень специфики требований (сколько и в какой степени детализации), запросы приоритетов и оценки усилий (две очень разные оценки, обычно назначаемые различными людьми в отдельных упражнениях), технические и управленческие риски и начальный контекст. Часть этих действий может включать начальные прототипы и модели проекта, непосредственно связанные с наиболее важными запросами совладельцев. Результатом определения системы является описание системы с использованием естественного языка и графического представления.

В этой работе участвуют все члены проектной группы.

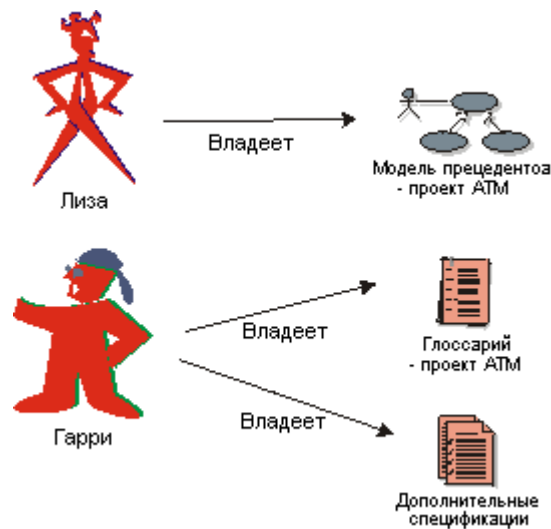
При определении системы могут применяться следующие типовые методы:

- Семинар по требованиям
- Семинар по прецедентам
- Иллюстрированный сценарий

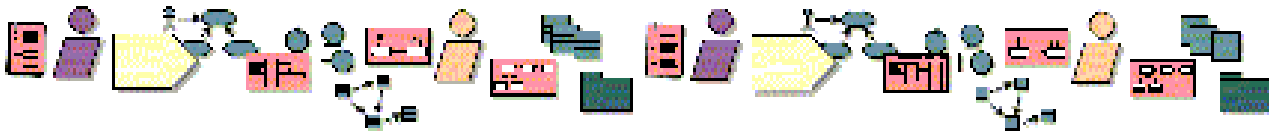


Пример

В стадии Начало Лиза и Гарри действуют как системные аналитики. На этом шаге они встречаются, чтобы проанализировать результаты своих встреч с совладельцами и заказчиками, и договориться о первых набросках модели прецедентов и Дополнительных спецификаций.



И далее Лиза и Гарри действуют как системные аналитики, они оба отвечают за завершение артефактов. Лиза - владелец модели прецедентов, а Гарри - владелец Глоссария и Дополнительных спецификаций.



Управление контекстом системы

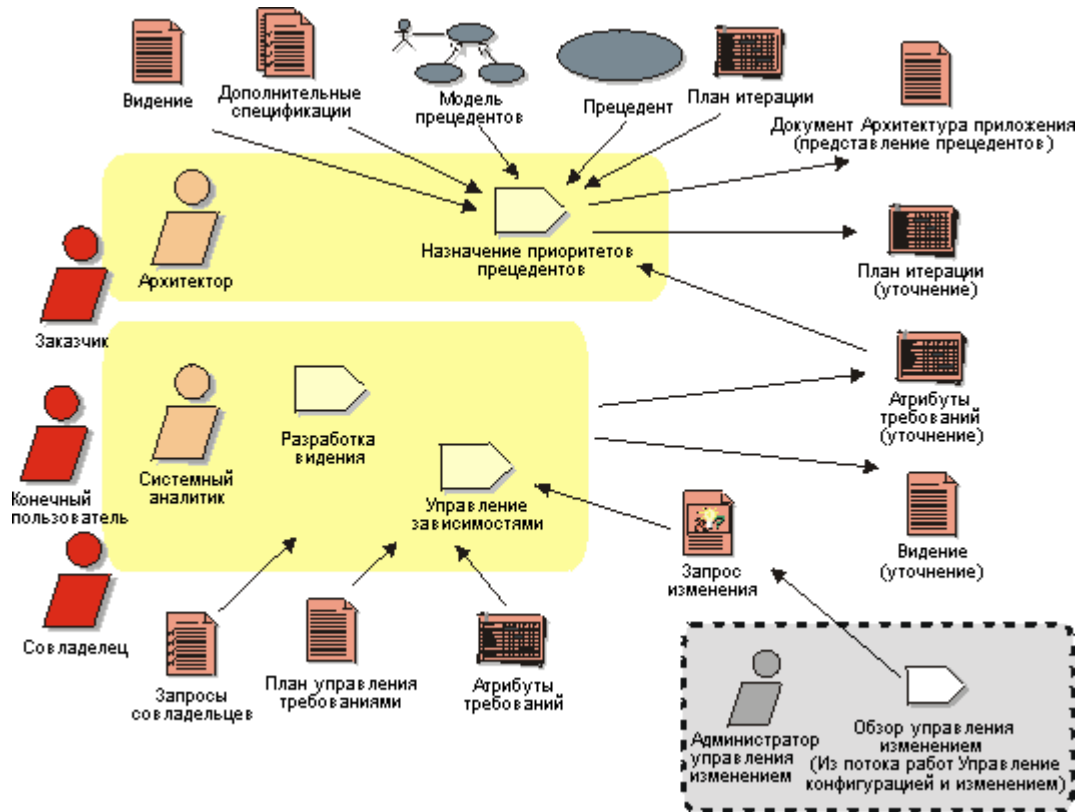
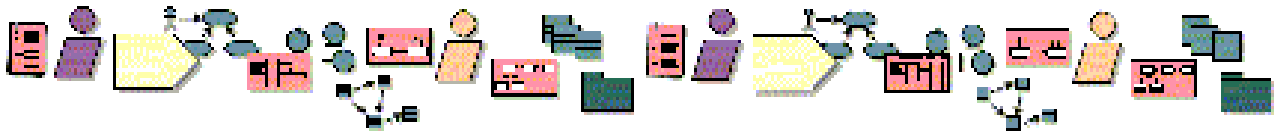


Диаграмма деталей потока работ Управление контекстом системы

Целями этой части потока работ являются:

- Определить исходную информацию для выбора требований, которые должны быть включены в текущую итерацию.
- Определить набор возможностей и прецедентов (или сценариев), которые представляют некоторые существенные, центральные функциональные возможности.
- Определить, какие аргументы требований и трассируемости необходимо поддерживать.



Управление контекстом – непрерывная деятельность, которая требует итерационной разработки в приращениях, и которая разбивает контекст проекта на меньшие, более управляемые части.

Для управления контекстом при принятии решений о включении или не включении требования особенно полезно использование атрибутов требования, типа приоритета, усилия и риска. Уделение внимания атрибутам больше, чем самим требованиям, помогает снизить накал страстей на переговорах с совладельцами, которые иначе могут перерасти в спор.

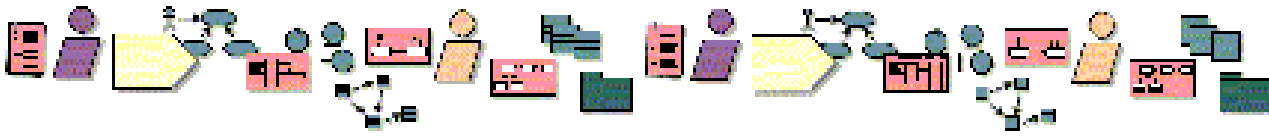
Хотя контекст проекта должен управляться непрерывно, лучшее понимание функциональных возможностей системы, полученных при идентификации большинства субъектов, прецедентов и дополнительных спецификаций, позволяет системному аналитику более точно устанавливать атрибуты требований, и дает возможность архитектору идентифицировать архитектурно существенные прецеденты.

Исходным для управления контекстом, не участвующим в других частях потока работ Требования, является План итерации, разрабатываемый параллельно в потоке работ Управление проектом. Контекст проекта, определенный в этой части потока работ, будет иметь значительное воздействие на План итерации, поскольку самые высокие элементы риска в пределах контекста будут запланированы в ранних итерациях. Другими важными результатами управления контекстом являются начальный вариант документа Архитектура приложения и усовершенствованное Видение, которое отражает лучшее понимание функциональных возможностей системы и проектных ресурсов со стороны системных аналитиков и ключевых совладельцев.

В этой части потока работ должны участвовать все члены группы архитектуры.

Для руководителей группы очень полезно иметь опыт в ведении переговоров, а также иметь лоббиста проекта в организации и на стороне заказчика. Лоббисты изделия/проекта должны иметь организационную власть отказаться от расширений контекста вне доступных ресурсов или выделить дополнительные ресурсы для обеспечения расширения контекста.

Группа архитектуры должна провести коллективное обсуждение того, как лучше расположить требования по приоритетам.



Уточнение определения системы

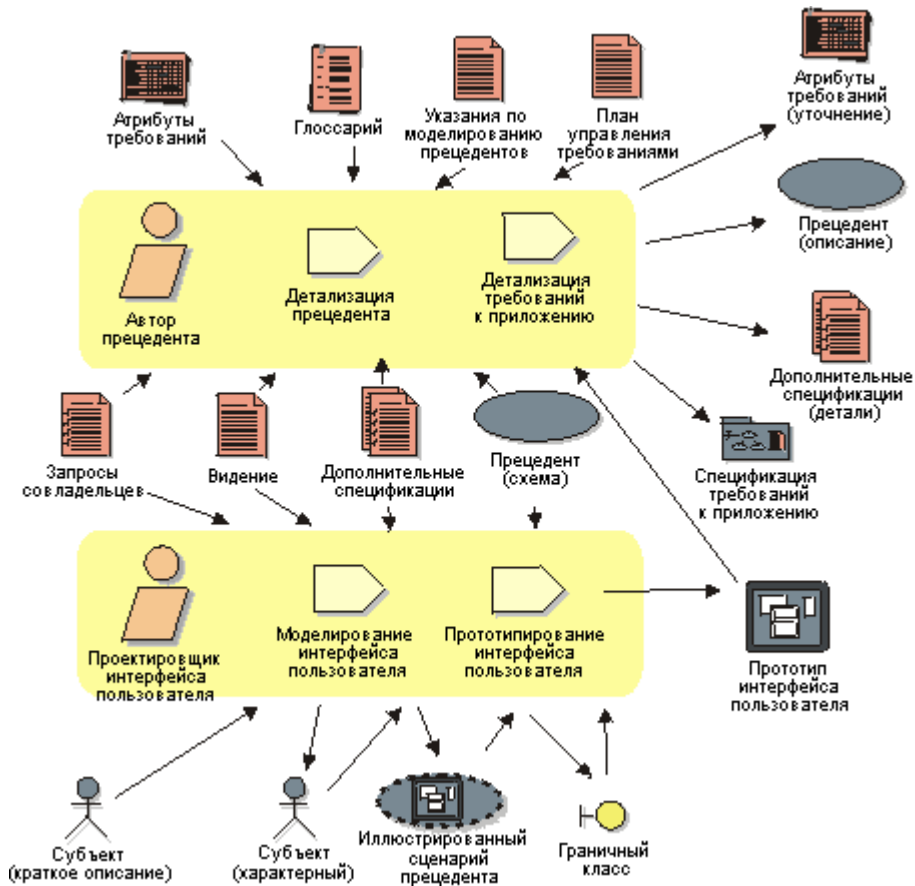
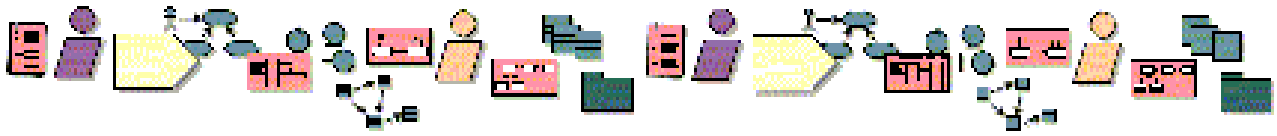


Диаграмма деталей потока работ Уточнение определения системы

Цель этой части потока работ состоит в дальнейшем совершенствовании требований, чтобы:

- Подробно описать поток событий прецедента.
- Детализировать Дополнительные спецификации.
- Моделировать и прототипировать интерфейс пользователя.

Детальное определение системы должно быть представлено таким образом, чтобы ваши совладельцы могли понимать его и соглашаться или не соглашаться с ним. Оно требует не только освещения функциональных возможностей, но также и соответствия всем юридическим



или нормативным требованиям, применимости, достоверности, эффективности, пригодности и надежности в эксплуатации. Часто совершаемая ошибка состоит в том, что Ваше ощущение сложности разработки толкает Вас на формирование комплексного определения. Это ведет к трудностям в объяснении цели проекта и системы. Люди могут быть увлечены, но они не дадут хорошей отдачи, так как у них нет полного понимания. Вы должны приложить максимум усилий для понимания аудиторией тех документов, которые Вы производите для описания системы. Часто может возникать потребность производить различные виды описаний применительно к аудитории.

Мы видели, что методология прецедента, часто в комбинации с простыми визуальными прототипами, является очень эффективным способом указания цели системы и определения ее подробностей. Прецеденты помогают помещать требования в контекст, они сообщают историю того, как будет использоваться система.

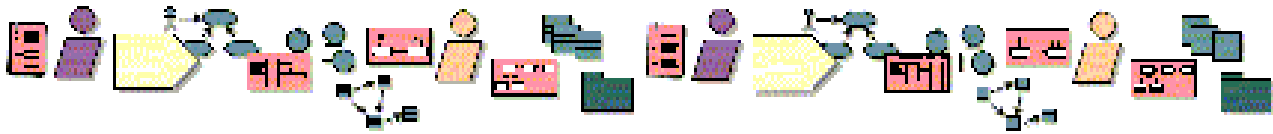
Другой компонент детального определения системы - это констатация того, как система должна быть испытана. Планы проведения и определения предметов испытания говорят о том, какие возможности системы будут проверены.

Усовершенствование системы начинается со схем прецедентов, кратких описаний субъектов и уточненного понимания контекста проекта, отраженного в Видении в форме списка расположенных по приоритетам возможностей, которые, как предполагается, являются достижимыми в рамках имеющихся ресурсов и сроков. Выход этого потока работ - более глубокое понимание функциональных возможностей системы, выраженных в детальных описаниях прецедентов, исправленных и детализированных Дополнительных спецификациях и в элементах интерфейса пользователя.

Авторы прецедентов должны быть квалифицированы в описании прецедентов и иметь хорошее представление о прикладной области. Часто полезно включать в вашу группу авторов прецедентов людей, которые позже будут действовать как проектировщики и тестировщики.

В больших проектах проектировщики интерфейса пользователя – это часто отдельная группа людей, сосредоточенная только на разработке интерфейса пользователя.

Авторы прецедентов и проектировщики интерфейса пользователя должны работать вместе, чтобы совместно определять детальные требования к системе. Хотя многие действия могут выполняться по отдельности, необходимо использовать сквозной контроль, чтобы убедиться, что элементы не противоречат друг другу.



Управление изменением требований

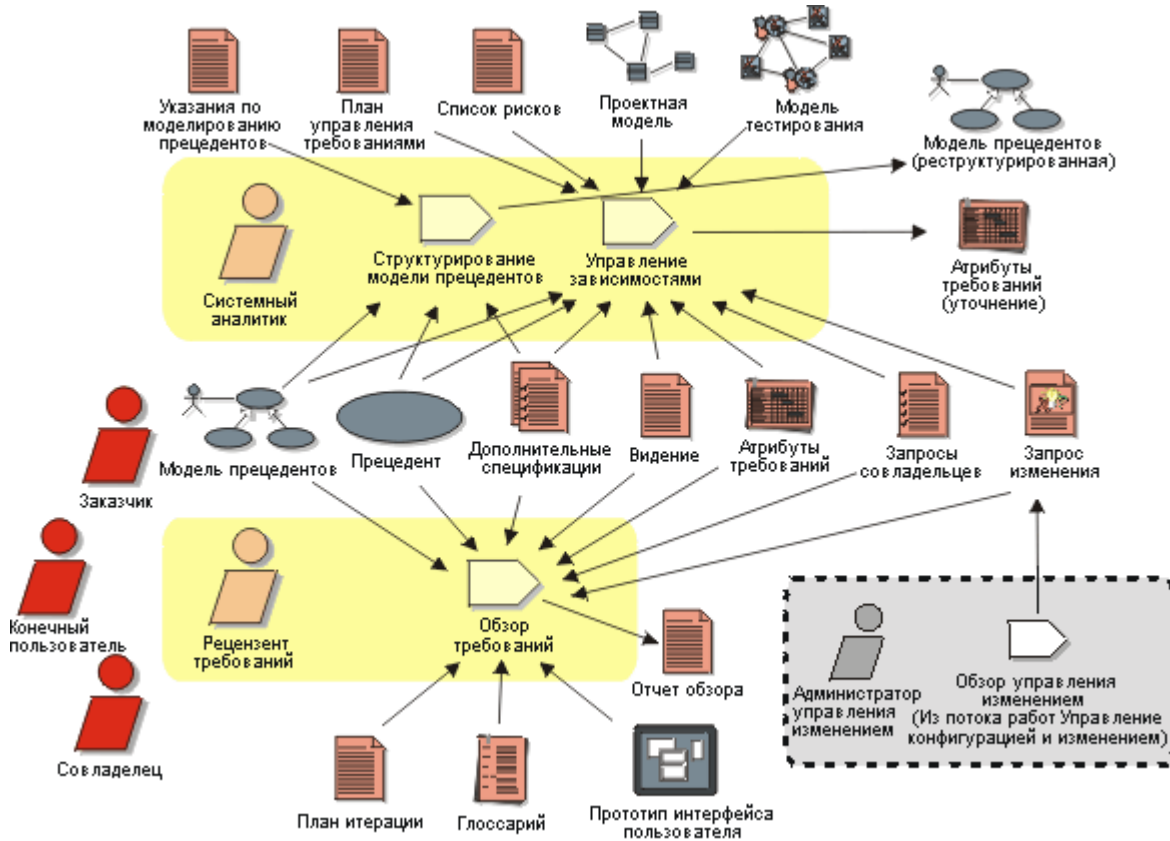
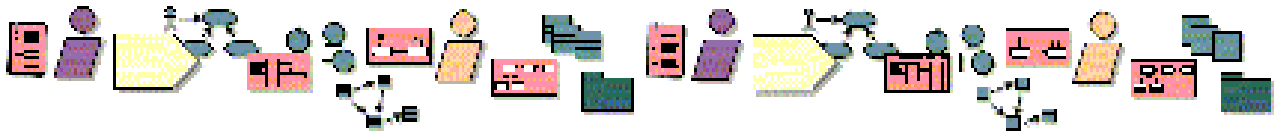


Диаграмма деталей потока работ Управление изменением требований

Целями этой части потока работ являются:

- Оценить формально представленные запросы изменения и определить их противоречия с существующим набором требований.
- Структурировать модель прецедентов.
- Установить соответствующие атрибуты требований и трассируемости.
- Формально убедиться, что результаты потока работ Требования соответствуют взглядам заказчика системы.

Независимо от того, насколько Вы осторожны при определении ваших требований, всегда найдутся вещи, требующие изменения. Что делает изменяющиеся требования сложными для управления? Изменение требования означает не только то, что должно быть потрачено большее или меньшее количество времени на реализацию новой специфической возможности, но также и то, что изменение в одном требовании может вступить в конфликт с другими требованиями.



Вы должны убедиться, что Вы придаете вашим требованиям структуру, которая является гибкой к изменениям, и что Вы используете ссылки трассируемости для представления зависимостей между требованиями и другими артефактами жизненного цикла разработки. Управление изменением включает действия подобные установлению опорной линии, определению зависимостей, которые важно проследить, устанавливая трассируемость между связанными предметами и управлением изменением.

Изменение требований естественно приводит к противоречиям с моделью, произведенной в потоке работ Анализ и проектирование, а также с моделью тестирования, созданной как часть потока работ Испытание. Связи Трассируемости между требованиями, идентифицированные в действии Управление зависимостями этого и других потоков работ - ключ к пониманию этих противоречий.

Другая важная концепция - отслеживание хронологии требований. Фиксируя характер и обоснование изменений требований, рецензенты (в данном случае эту роль играет любой член группы разработки) воспринимают информацию, необходимую для того, чтобы ответить на изменение должным образом.

Привлеките расширенную группу (совладельцев, представителей заказчика, специалистов по проблемной области и других). Старайтесь эффективно управлять ресурсами рецензентов – не привлекайте всю расширенную группу, если Вы не можете гарантировать, что это даст положительный эффект для проекта.

Общая квалификация членов расширенной группы должна охватывать хорошее знание прикладной области, технические трудности проекта, а также навыки в управлении требованиями и моделировании прецедентов.

Основная группа разработки должна провести несколько раундов внутреннего сквозного контроля, чтобы прояснить явные несообразности прежде, чем работа будет формально проинспектирована и подвергнута экспертизе расширенной группой.

Вы должны разделить материал так, чтобы группа не делала обзор всего сразу. Совещание обзора не должно занимать больше одного дня (например, проведите отдельно обзор интерфейса пользователя).

