

Управление требованиями как дисциплина. Применение Borland CaliberRM для управления требованиями.

Автор: Михаил Голованов

Требование. Что это?	1
Роль и место требований в методологии разработки ПО. Виды требований.	1
Каким образом формируются требования?	3
Первичный сбор требований	3
Разработка и анализ требований.....	4
Понятие Base line.....	5
Управление изменениями требований	5
Пример разработки требований.....	6
Общая постановка задачи. Обзор и настройка CaliberRM для работы.	6
Первое знакомство с CaliberRM Client.....	7
Формирование бизнес требований	8
Формирование пользовательских требований.....	11
Функциональные требования	12
Нефункциональные требования	13
Формирование и отслеживание связей (Traceability)	13
Формирование спецификации требований	15
Запросы на изменение требований	15
Заключение	16

Требование. Что это?

Значительная часть моих коллег и знакомых еще год назад при фразе «управление требованиями» честно спрашивали меня: а что это? Признаться, полтора года назад я и сам не мог им ответить что-то вразумительное, однако прогресс не стоит на месте. В данной статье я постараюсь рассказать, об управлении требованиями, как одной из дисциплин разработки ПО, и убедить читателей в том, что управление требованиями вещь хорошая и приносящая большую практическую пользу.

Итак, для начала неплохо было бы определиться с термином «требование». Требование можно определить как:

- Условие или возможность, необходимая пользователю для решения проблемы или достижения цели
- Условие или возможность, необходимая для обеспечения системой или компонентом системы соответствия контракту, стандарту, спецификации или другому формализованному документу
- Документ, описывающий возможности системы и/или спецификация функций системы.

Ключевыми моментами в данных определениях является, то, что требования содержат спецификацию необходимых возможностей, но не привязаны к реализации и основным источником требований являются пользователи системы.

Однако практическая польза от данного определения невелика, так как из него недостаточно ясно как же получать, обрабатывать и применять требования в процессе создания ПО. Попробуем очертить круг наиболее часто задаваемых вопросов относительно управления требованиями и затем ответить на них.

Роль и место требований в методологии разработки ПО. Виды требований.

Независимо от выбранной методологии разработки, можно выделить следующие этапы жизненного цикла ПО:

1. Анализ рынка, технико-экономическое обоснование проекта
2. Сбор и анализ бизнес требований к ПО
3. Проектирование архитектуры системы
4. Кодирование
5. Тестирование
6. Развертывание
7. Эксплуатация и обслуживание

Наиболее активно интенсивно управление требованиями осуществляется на этапах 2 и 3, менее активно на этапах 4-5.

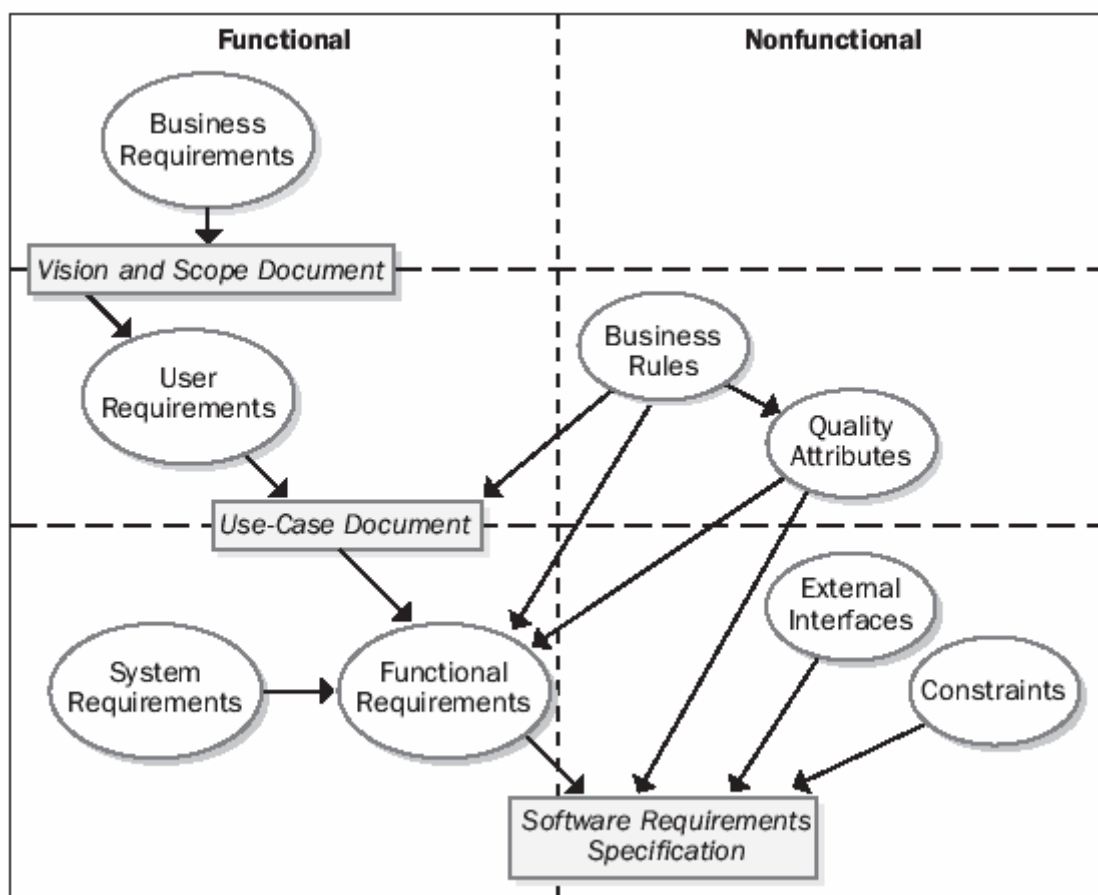
Общая классификация требований приведена на рис. 1.

Требования можно разделить на две большие категории: функциональные и нефункциональные.

Функциональные требования описывают, что именно должно делать разрабатываемое ПО. Нефункциональные требования содержат описание таких характеристик, как качество, соответствие стандартам и корпоративным правилам.

В свою очередь функциональные требования делятся на три основных группы:

- Бизнес требования
- Пользовательские требования
- Функциональные требования



Бизнес требования описывают пожелания пользователей с точки зрения задач бизнеса. Примеры бизнес требований:

- Ускорение обработки заказов на 30 %.
- Снижение издержек ведения бухгалтерии на 200 тыс. рублей в год.

Источником бизнес требований, как правило, является отдел маркетинга и высшее руководство. Бизнес требования описывают критерии успешности проекта с точки зрения

спонсоров проекта. Бизнес требования являются требованиями высшего уровня, остальные требования подчинены им.

Пользовательские требования отражают пожелания непосредственных пользователей системы. Хорошей практикой является оформление пользовательских требований в виде вариантов использования (use cases). Работе с пользовательскими требованиями следует уделять пристальное внимание, так как мнение о системе, сложившееся у непосредственных пользователей является ключевым фактором успешности внедрения и эксплуатации системы.

Самый нижний уровень требований – это функциональные требования. Данный вид требований детально описывает функции, которые должно выполнять ПО для удовлетворения пользовательских требований. Функциональные требования фактически являются заданиями для архитекторов и программистов.

К нефункциональным требованиям, относят:

1. Бизнес правила. Набор принятых приемов организации бизнес процессов, которые должны быть учтены при разработке ПО
2. Quality Attributes – атрибуты качества. Чаще всего это характеристики надежности отказоустойчивости системы.
3. Внешние интерфейсы. Набор программных интерфейсов, которые необходимо реализовать для интеграции в существующую в организации среду программных средств.
4. Ограничения. Законодательные и другие ограничения накладываемые на создаваемое ПО.

Каким образом формируются требования?

В данном разделе мы подробно рассмотрим основные этапы создания требований. Основные этапы создания требований включают:

- Первичный сбор требований
- Разработка и анализ требований
- Управление изменениями требований

Первичный сбор требований

Процесс первичного сбора требований (gathering) является первым шагом в процессе создания требований. Сбор требований следует начинать с бизнес требований, так как все остальные виды требований подчинены им.

Основной задачей этапа сбора бизнес требований является выработка образа продукта (product vision). Образ продукта описывает, что сейчас представляет собой продукт и каким он станет впоследствии. Границы проекта (project scope) показывают, к какой области долгосрочного образа продукта направлен текущий проект. Образ продукта подразумевает определение долгосрочных перспектив развития проекта, в то время как границы проекта могут быть определены для конкретной итерации. Следует очень внимательно отнестись к формированию образа и границ проекта. Это позволит избежать или, по крайней мере, запланировать на более позднее время реализацию менее критичных для успеха проекта возможностей ПО (наверно вам знакомы ситуации когда войдя во вкус клиент просит сделать еще и вот это и это, при этом не выделяя дополнительных ресурсов). Формирование образа и границ проекта позволяет руководству понять, за что именно они будут платить деньги, а разработчикам понять чего хочет от них руководство. В прилагаемых к статье материалах мы привели шаблон документа об образе и границах проекта, взятый с сайта Карла Вигерса (<http://processimpact.com/>).

Следующим этапом является сбор пользовательских требований, другими словами вам необходимо выяснить чего ожидают от вас непосредственные пользователи. Естественно, что пользовательские требования должны быть жестко подчинены бизнес требованиям.

Практически любая информационная система имеет несколько категорий пользователей. Очень важно проанализировать потребности каждой из таких групп, достигнув компромисса между противоречащими требованиями различных групп пользователей. Необходимо четко определить приоритетность каждой группы и при разрешении противоречий между требованиями в первую очередь заботится об удовлетворении наиболее приоритетных групп.

Наиболее удачным подходом является оформление пользовательских требований в виде вариантов использования.

Собранные требования оформляются в виде спецификации требований (System Requirement Specification - SRS).

Основными ошибками, допускаемыми этапе первичного сбора требований, являются:

- Нечеткое определение границ проекта, что приводит к затягиванию сроков проекта, перерасходу бюджета за счет включения в продукт второстепенных возможностей.
- Недостаточно четко определенные группы пользователей продукта
- В каждой группе не выделены представители, наиболее заинтересованные в продукте, готовые продуктивно сотрудничать с командой разработчиков
- Попытка с первого раза максимально детализировать и проанализировать все требования

Чтобы не допустить этих ошибок можно дать несколько рекомендаций:

- Разделите пользователей продукта на категории в соответствии с их ролями
- В каждой группе выберите одного - двух человек в качестве представителей интересов группы. Выбранные люди должны быть лидерами (пусть даже и неформальными) в своей группе, четко представлять бизнес процессы, в которых участвует группа и лояльно настроенных по отношению к проекту создания и внедрения разрабатываемой системы.
- Не пытайтесь сразу и досконально описать все требования. Это приведет лишь к затягиванию работ. Требования должны прорабатываться и детализироваться в ходе проекта.
- Каждому требованию присваивайте приоритетность. Это позволит реализовывать в первую очередь самые необходимые требования, откладывая реализацию второстепенных возможностей на более поздний срок. Как правило, достаточно ввести три – четыре градации приоритетности требований.

Разработка и анализ требований.

Данный этап является самым продолжительным и наиболее важным. Он может состоять из нескольких итераций. Основные задачи этапа: детализация первичных требований, обеспечение непротиворечивости и полноты требований.

В результате анализа мы должны получить набор требований, охватывающий все области разработки приложения. Непротиворечивость требований означает однозначное понимание изложенных требований всеми участниками проекта, а так же отсутствие противоречий между различными сформулированными требованиями.

Кроме того, каждому требованию необходимо назначить приоритетность и построить матрицу взаимных зависимостей требований. Назначение приоритетности позволяет выделить наиболее критичные для успеха проекта требования и запланировать их реализацию на более ранние сроки по сравнению с менее значимыми требованиями. Обычно, шкала приоритетности требований включает три-пять ступеней. Матрица зависимостей между требованиями позволяет построить наиболее оптимальный календарный график работ по реализации требований (критический путь реализации проекта). Построение матрицы связей требований преследует еще одну цель – обеспечение возможности анализа влияния вносимых изменений на проект.

При анализе и детализации требований основными приемами работы с участниками проекта являются проведение круглых столов, обсуждений, «мозговых штурмов», демонстраций предварительных версий разрабатываемой системы.

Можно дать несколько практических рекомендаций, которые позволят вам сэкономить время и силы в ходе проведения таких мероприятий:

- Четко очерчивайте круг обсуждаемых на встрече.
- Заранее оцените время, которое потребуется.

- Заранее ознакомьте участников обсуждения с повесткой и расписанием предстоящего мероприятия.
- Не позволяйте одному из участников подавлять остальных, вовремя останавливайте, чтобы остальные могли высказать свою точку зрения на обсуждаемую проблему.

На круглые столы и обсуждения достаточно пригласить лишь представителей групп пользователей и разработчиков, которые непосредственно участвуют в решении обсуждаемых вопросов. На показы промежуточных версий полезно пригласить всех, с одной стороны это позволяет пользователям увидеть и «пощупать» систему, а разработчикам получить живые отклики. Как показывает практика, оптимальное количество участников круглого стола 4-8.

Важным моментом проведения обсуждений является их документирование и протоколирование. Это позволяет не держать постоянно в голове все важные детали проекта, а так же отследить источники возникновения тех или иных требований. Довольно часто представители пользователей боятся подписывать протоколы обсуждений. Страх связан с ожиданием каких-либо карательных мер. Следует объяснить пользователям, что цель получения их подписи под протоколом не в том, чтобы потом наказать их, а всего лишь в том, чтобы убедиться, что между участниками обсуждения достигнуто взаимопонимание и выработано совместное решение. Еще раз хочу обратить ваше внимание на чрезвычайную важность протоколирования обсуждений.

Понятие Base line.

В ходе выполнения проекта требования постоянно развиваются, детализируются и изменяются. В то же время требования постоянно используются командой проекта. Таким образом, с одной стороны аналитику необходимо постоянно вносить изменения, с другой стороны команда разработки должна пользоваться в любой момент времени согласованным и непротиворечивым набором требований. Очевидно что, для разрешения описанного противоречия необходимо ввести версию требований, при этом недостаточно хранить версии каждого отдельного требования, необходимо вести версии среза требований на заданный момент времени. Срез согласованных требований на определенный момент времени называется базовой линией (base line). По мере окончания итераций детализации и анализа требований аналитик создает базовые линии. Как правило, одна из последних созданных аналитиком базовых линий предназначена для команды разработки, а последняя по дате линия является рабочей для аналитика требований. Аналитик вносит изменения в рабочую базовую линию, при этом данные изменения не отображаются в других базовых линиях, используемых другими участниками проекта. По окончании итерации анализа базовая линия публикуется, другими словами становится видна участникам команды разработки. Другими словами разработчики не видят «художеств» аналитика, до тех пор, пока аналитик явным образом не сделает доступной разработчикам очередную базовую линию.

Управление изменениями требований

В ходе развития проекта требования неоднократно корректируются и изменяются. Так как требования связаны друг с другом, изменения, вносимые в одно требование, могут повлиять на связанные с ним другие требования. Требования так же являются основаниями для включения в продукт тех или иных функциональных возможностей, поэтому изменения в требованиях могут потребовать доработки или переработки продукта. Таким образом, для того чтобы проект не потерял управляемость и был создан в срок в рамках бюджета, необходимо выработать и утвердить процедуру внесения изменений в требования.

Для внесения и обсуждения изменений в требования, как правило, выделяется отдельный вид требований, называемый запросами на изменение. Создавать запросы на изменение могут все заинтересованные участники проекта, однако, до того как запрос не будет рассмотрен, проанализирован и утвержден аналитиком требований. При анализе аналитик выявляет степень влияния предлагаемых доработок на требования, сроки реализации и бюджет. Затем внесенные изменения либо утверждаются, либо отвергаются.

Пример разработки требований

Как известно, теория без практики - абсолютно бесполезна. Оставшаяся часть статьи будет посвящена рассмотрению примера управления требованиями. Цель данной части статьи показать практические приемы управления требованиями с помощью существующих средств автоматизации управления требованиями.

На сегодняшний день на рынке наиболее известными средствами управления требованиями являются:

- Rational Requisite Pro
- Borland Caliber RM
- Telelogic DOORS
- Popkin Software Architector

В качестве инструмента управления требованиями в нашем примере я использовал CaliberRM по нескольким причинам:

1. Реализация объектно-ориентированного подхода управления требованиями
2. Простота
3. Наличие подробной документации
4. Возможность организации групповой работы
5. Широкие возможности
6. Доступность evaluation лицензии

Общая постановка задачи. Обзор и настройка CaliberRM для работы.

Итак, как это всегда случается начальство в кратких деталях дает вводную. Необходимо разработать приложение для автоматизации деятельности средней по размерам библиотеки, приступить надо было еще вчера. Справившись с легкой паникой, возникающей в таких случаях, приступаем к фазе сбора первичных требований.

Прежде всего, обдумывание задачи начнем с установки и краткого обзора CaliberRM ©. Итак CaliberRM состоит из нескольких приложений:

- CaliberRM Server – серверная часть продукта. Достаточно установить один сервер в локальной сети.
- CaliberRM Client – клиентская часть, основное приложение для пользователя. Устанавливается на каждой машине, где планируется работать с CaliberRM.
- Framework Administrator – утилита настройки и администрирования. По умолчанию пароль и имя администратора системы admin.
- Document Factory – генератор отчетов.

Первым шагом является создание на сервере необходимых групп пользователей и учетных записей пользователей, отражающих организационную структуру команды проекта. Сделать это можно из меню File | New Group. В стандартной поставке уже введены наиболее часто встречающиеся группы участников проекта, однако естественно, что введены они на английском языке. Для рассматриваемого проекта мы ввели новые группы участников проекта на русском языке:

- Администраторы
- Аналитики
- Маркетологи
- Разработчики
- Тестировщики
- Технические писатели

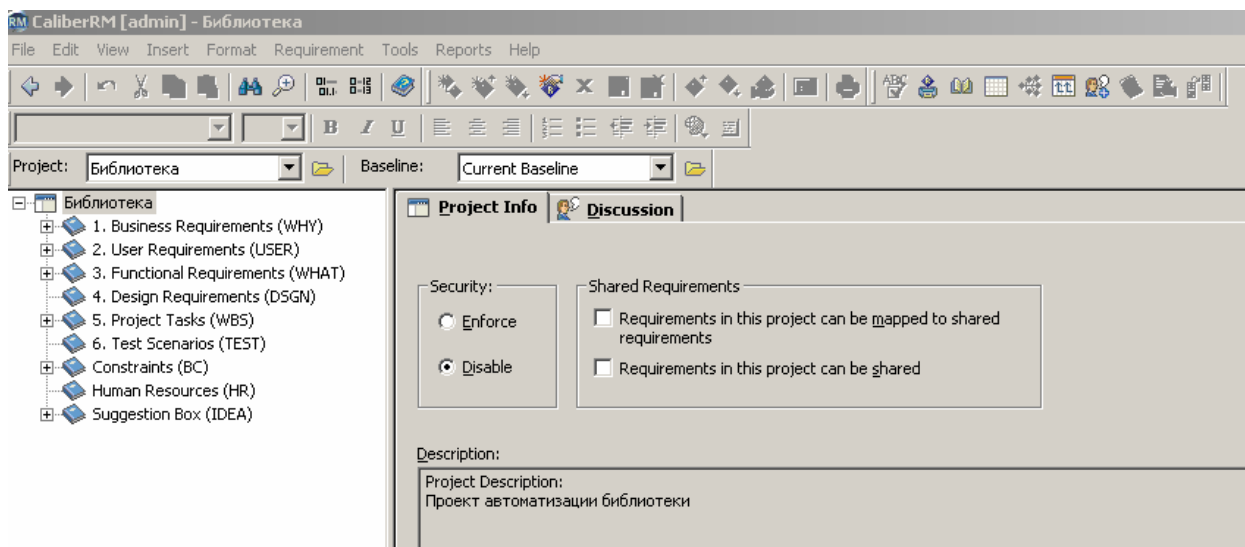
В каждую группу были добавлены один два пользователя.

Следующим шагом является создание нового проекта. Для создания нового проекта необходимо запустить Framework Administrator и войти в него под учетной записью администратора (admin). Далее либо в меню, либо с помощью правой кнопки мыши необходимо выбрать пункт File| New Project. В появившемся мастере создания нового проекта необходимо указать имя создаваемого проекта (в нашем случае «Библиотека»). Далее можно ввести описание проекта и группы участников проекта. Последним шагом работы мастера

является указание словарей, которые мы будем формировать в ходе работы над требованиями. На этом подготовительный этап можно считать завершенным.

Первое знакомство с CaliberRM Client.

После авторизации на экране появится основное окно CaliberRM Client, вид которого приведен на рисунке ниже.



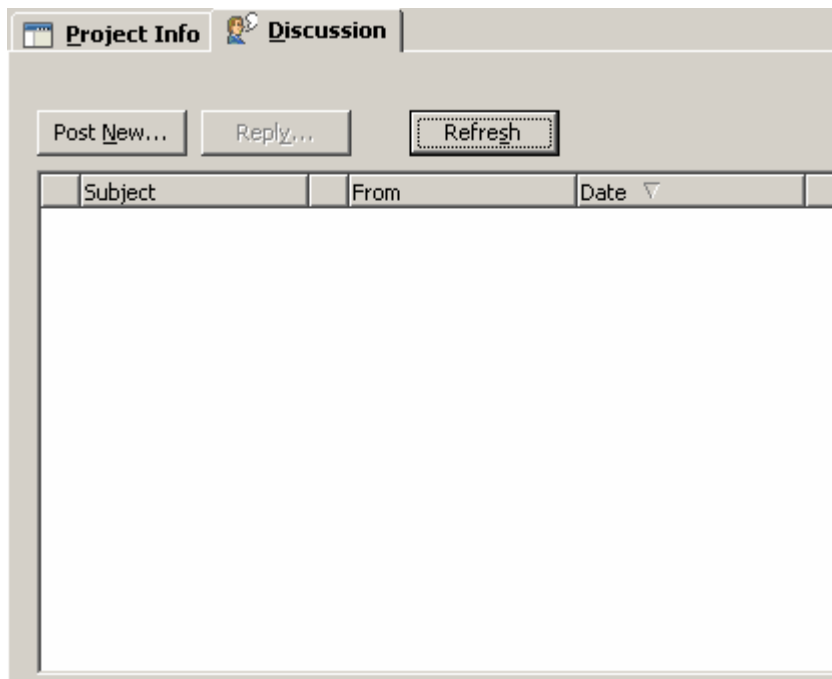
Окно можно условно разделить на три части: Панель инструментов и меню (сверху), дерево требований (слева), окно с детальной информацией, зависящей от выбранного узла в дереве требований.

Рассмотрим более подробно дерево требований. Корневой элемент дерева соответствует активному проекту. Проект можно выбрать из выпадающего списка Project. Вложенные узлы первого уровня соответствуют типам требований. Внутри каждого вида требований располагаются ветки требований. В соответствии с выбранным узлом дерева меняется содержимое окна детальной информации. Для узла проекта окно детальной информации содержит две закладки: Project Info и Discussion. Закладка Project Info позволяет настроить проверку безопасности и возможность использования требований в других проектах.

Настройки безопасности представлены группой переключателей Security. Можно либо включить проверку безопасности (флажок Enforce), либо отключить (флажок Disabled). Настройка политик безопасности осуществляется в Framework administrator. На первом этапе, когда идет активная работа над требованиями и роли в проекте еще жестко не закреплены, рекомендуется отключить проверку безопасности, чтобы не создавать себе лишних проблем с настройкой. Включать проверку безопасности имеет смысл, после того как требования будут стабилизированы и роли в команде детально распределены.

Вторая группа флажков Share Requirements предназначена для настройки доступности требований вне рамок проекта. Рассмотрение совместного использования требований выходит за рамки статьи, поэтому мы не будем заострять наше внимание на данной группе настроек.

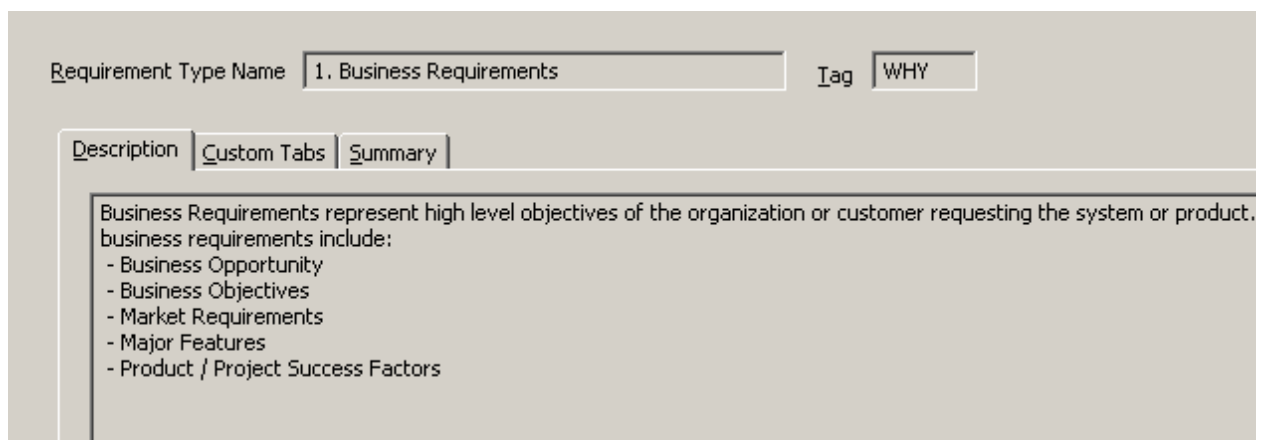
Вторая закладка Discussion позволяет организовать обсуждение проекта участниками проекта. Закладку Discussion имеют практически все узлы дерева. Внешний вид интерфейса организации дискуссий напоминает обычную доску сообщений и приведен на рисунке ниже.



Кнопка Post New позволяет создать новую тему обсуждения, кнопка Reply написать ответ на выбранное сообщение, а кнопка Refresh обновить список сообщений.

Формирование бизнес требований

Узел дерева требований Business Requirements(WHY) предназначен для хранения бизнес требований. Если выбрать этот узел (как впрочем, и любой другой узел типов требований) окно детальной информации будет иметь вид представленный на рисунке ниже.



Поле ввода Requirement Type Name содержит имя типа требований. Поле Tag – краткое описание назначения типа. На закладке Description хранится краткое описание типа требований, закладка Custom Tabs содержит перечисление специфичных для данного вида требований закладок атрибутов. На закладке Summary можно делать пометки о ходе работы над данным видом требования. Отметим, что можно создавать собственные типы требований с помощью Framework Administrator.

Создать новое бизнес требование можно либо с помощью пункта меню Requirement | Create Requirement | Child, либо щелчком правой кнопкой мыши на узле Business Requirements из контекстного меню New| Child. При этом в узле бизнес требований создается дочерний узел требования. Ниже приведен внешний вид окна детальной информации для бизнес требования.

The screenshot shows a software interface with a top navigation bar containing tabs: Validation, Discussion, History, Details, User Attributes, Responsibilities, References, and Traceability. The 'Details' tab is selected. Below the tabs, there are several input fields and dropdown menus:

- Requirement Name:** Снижение времени выдачи к
- Requirement Tag/Id:** WHY292
- Requirement Version:** V 3.0 (dropdown menu)
- Status:** Submitted (dropdown menu)
- Priority:** Essential (dropdown menu)
- Owner:** akhtyrchenko (dropdown menu with a user icon)

Большинство закладок являются общими для всех видов требований. К специфическим для бизнес требований является лишь закладка User Attributes.

Закладка Details содержит общую информацию о требовании:

- Requirement Name. Имя требования. Имя отображается в узле дерева.
- Requirement TagId. Идентификатор требования. Уникальный идентификатор, состоящий из имени тэга требования и его порядкового номера. Позволяет однозначно идентифицировать требование.
- Requirement Version. Версия требования. История изменений требования может быть отслежена с помощью версий. Формирование новой версии происходит автоматически при внесении изменений в требование.
- Status. Статус требования. В стандартной конфигурации требование может иметь следующие статусы:
 - Submitted –предложено к рассмотрению (статус по умолчанию)
 - Pending – ожидает рассмотрения
 - Accepted – принято
 - Draft – версия требования является черновой
 - Deferred – отложено
 - Review Complete – просмотр данного требования завершен
- Priority. Приоритетность реализации требования. В стандартной конфигурации приоритетность может иметь следующие значения:
 - Unassigned – не присвоена (уровень по умолчанию)
 - Essential – обязательно реализовать
 - Useful – полезно реализовать
 - Desirable – реализовать по возможности
- Owner. Владелец/создатель требования. Пользователь, владеющий требованиями.

Закладка Responsibilities содержит список пользователей и групп, отвечающих за разработку и реализацию данного требования.

Закладка References позволяет привязать к требованию внешние файлы, текстовые документы, Web документы.

Discussion – закладка дискуссий по требованию. Была описана в предыдущем разделе.

History – история создания версий требования.

Validation. Процедура проверки требования. На данной закладке можно описать общую процедуру тестирования и проверки правильности требования. В дальнейшем команда тестирования на основе данной информации создаст более детализированные процедуры тестирования.

Traceability – отслеживаемость. Данная закладка позволяет установить и настроить связи требования с другими требованиями. Информация о связях требований используется при анализе изменений.

Специфической закладкой для данного вида требований является закладка Custom Attributes, которая содержит следующие элементы управления:

Owner Priority – приоритет, установленный владельцем требования

Sponsor Name – имя организации или лица, финансирующего работу по данному требованию.

Source – источник требования. Описания источника возникновения требования.

Funded – выделены ли финансовые ресурсы для реализации требования в отдельную строку бюджета.

Notes – заметки и дополнительные сведения.

Due Date – дата сдачи заказчику реализации требования.

После того как мы разобрались с представлением бизнес требований в CaliberRM Client самое время организовать первое интервью с заказчиком. В ходе этого интервью необходимо уяснить детали наиболее важных бизнес процессов организации и ожидания заказчика от внедрения разрабатываемой системы в плане бизнеса. В случае когда планируется создание коробочного продукта формулированием бизнес требований занимается отдел маркетинга. Эффект от внедрения необходимо представить в виде измеряемых показателей, которые легко рассчитать и проконтролировать. На основе системы этих показателей вырабатываются критерии успешности проекта. Вехой данного этапа является создания документа о видении и рамках проекта и формирование базовой линии «Инициация проекта».

В ходе общения с заказчиком были выявлены следующие бизнес требования:

- Снижение времени выдачи книг
- Ведение электронной картотеки посетителей
- Создание электронного каталога книг для посетителей
- Создание отчета о неудовлетворенном спросе на книги.

Данные требования были задокументированы. В ходе работы над бизнес требованиями было внесено предложение об использовании штрих кодов для идентификации экземпляров книги.

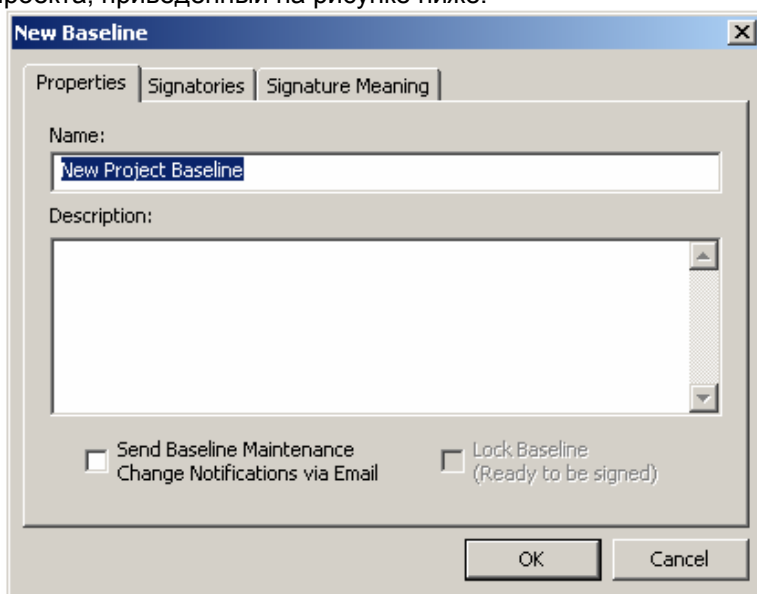
В результате анализа бизнес требований был создан документ видения и рамок проекта. Шаблон данного документа и сам документ прилагаются к статье.

Обратите внимание, что при работе с требованиями в выпадающем списке Baseline по умолчанию выбран элемент Current baseline. После сбора и анализа бизнес требований нам необходимо создать базовую линию «Инициация проекта», чтобы зафиксировать срез бизнес требований для заинтересованных лиц.

Замечание 1. Для создания и модификации статуса базовой линии Вы должны обладать правами администратора.

Замечание 2. Фиксация базовых линий обычно совпадает с окончанием очередной итерации проекта или прохождением вехи.

Создание базовой линии производится с помощью Framework Administrator. На закладке Project Baselines проекта нажмите кнопку New. Вы увидите диалог создания новой базовой линии проекта, приведенный на рисунке ниже.



В поле ввода Name необходимо ввести имя базовой линии. В поле ввода Description – опционально дополнительные сведения и описание содержания базовой линии. Флажок Send Baseline Maintenance Change Notification via Email указывает на необходимость рассылки уведомлений о создании базовой линии по электронной почте. Флажок Lock Baseline позволяет заблокировать любые изменения базовой линии. На первом этапе работы с

базовой линией флажок сброшен, в линию нельзя добавлять новые требования, но можно модифицировать существующие требования. После того, как все вопросы к требованиям в линии разрешены, линия блокируется, после чего никакие изменения уже внести нельзя.

Закладка Signatories позволяет указать участников проекта, которые утверждают готовность базовой линии. Закладка Signatore Meaning содержит список возможных резолюций ответственных лиц.

После создания базовой линии нужно ее наполнить требованиями, другими словами инициализировать. В подавляющем большинстве в линию включают все существующие требования. Инициализация линии происходит при первом ее выборе в CaliberRM Client.

Формирование пользовательских требований.

Пользовательские требования описывают требования непосредственных пользователей программы.

В CaliberRM пользовательские требования расположены в категории User Requirements (USER).

Чаще всего пользовательские требования описываются с помощью вариантов использования. Варианты использования могут быть описаны как в текстовом виде, так и с помощью UML. Построение вариантов использования является отдельной темой, достойной отдельной статьи. Здесь я буду использовать готовый шаблон взятый с сайта www.processimpact.com и переведенный на русский язык. Варианты использования описаны в виде текстового документа и прикреплены к пользовательским требованиям в CaliberRM.

Пользовательские требования могут быть собраны в ходе:

- Анализа бизнес процессов организации
- Интервью с пользователями
- Круглых столов с участием пользователей
- Демонстраций предварительных версий ПО.

При организации общения с пользователями полезно придерживаться нескольких простых правил:

1. Четко формулируйте цель встречи, сформулируйте обсуждаемые вопросы
2. Заранее определите состав участников, оповестите их о месте, времени и продолжительности встречи.
3. Оптимальная продолжительность встречи 40 минут – час. Не стоит затягивать встречу более полутора часов.
4. Давайте слово всем участникам встречи. При обсуждении проблем не давайте доминировать кому-то одному, обязательно должны высказаться все участники дискуссии
5. Обязательно документируйте принятые в ходе встречи решения.

Пользовательские требования подчинены бизнес требованиям, а значит, каждое из них должно иметь как минимум одно связанное с ним бизнес требование.

В ходе анализа и сбора были выявлены следующие пользовательские требования к системе:

- Поиск книг в каталоге.
 - Новый поиск
 - Поиск в найденном
- Поиск книги библиотекарем по ISBN
- Выдача книг посетителю
 - Формирование квитанции
- Прием книги у посетителя
 - Закрытие квитанции
- Формирование отчета о наиболее востребованных книгах
- Идентификация посетителя
- Формирование каталога книг

Требования разрабатывались последовательно в рамках двух базовых линий: Первичные пользовательские требования и Пользовательские требования. Ниже приведен пример описания варианта использования для требования.

Use Case ID:	3		
Имя варианта использования:	Выдача книги посетителю		
Создал:	Голованов Михаил	Последние изменения внес:	
Дата создания:	28 июля 2004	Дата последнего изменения:	
Действующие лица:	Библиотекарь, Посетитель		
Описание:	Цель данного варианта использования – выдача книг на руки посетителю		
Предусловия:	1. Сформирован список заказанных книг, все книги имеются в наличии		
Постусловия:	Квитанция о выдаче сформирована и подписана посетителем, книги выданы на руки		
Нормальный поток выполнения:	<ol style="list-style-type: none"> 1. Библиотекарь находит на полках заказанные книги и приносит их 2. Библиотекарь распечатывает квитанцию о выдаче книг 3. Посетитель подписывает квитанцию и возвращает библиотекарю. 4. Библиотекарь выдает книги 		
Альтернативные потоки			
Исключения			
Включения			
Приоритетность			
Частота использования			
Бизнес правила			
Специальные требования	Минимизировать расход бумаги при печати квитанций		
Предположения			
Заметки			

Функциональные требования

Функциональные требования описывают функции, которые должно выполнять разрабатываемое ПО. Функциональные требования вытекают из пользовательских требований. В стандартном шаблоне функциональные требования представлены группой требований 3. Functional Requirements. Функциональные требования можно рассматривать в качестве заданий команде разработки. Шаблон CaliberRM для функциональных требований имеет три дополнительные закладки:

- Ambiguity Review
- Estimate
- User Attributes

Закладка Ambiguity Review предназначена для документирования анализа полноты и непротиворечивости функциональных требований. На данной закладке возможно указать лиц, осуществивших анализ полноты и непротиворечивости требования, количество выявленных неточностей и комментарии.

Закладка Estimate позволяет указать оценку трудозатрат на реализацию функционального требования (верхнюю, нижнюю границы и предполагаемое значение трудозатрат). Трудозатраты оцениваются в баллах по относительной шкале.

Закладка User Attributes содержит поля для ввода предполагаемой даты реализации требования, рейтинга данного требования у пользователей и версии ПО, в которой предполагается реализовать требование.

Таким образом, заполнив информацию о требовании на трех описанных закладках можно сформировать предварительный план реализации ПО.

В ходе работы над проектом функциональные требования были сгруппированы по категориям. В нижеприведенной таблице перечислены выявленные функциональные требования.

Категория / требование
Каталог
Создание, редактирование, удаление новой категории книг
Ввод новой книги
Удаление и редактирование книги
Поиск книг по каталогу
Пользователь
Создание нового пользователя, редактирование и удаление
Поиск и идентификация пользователя
Библиотекарь
Формирование заказа и выдача книг
Возврат книг посетителем
Менеджер отдела закупок
Формирование отчета о неудовлетворенном спросе на книги
Документация
Получение справки по работе с программой

Нефункциональные требования

К нефункциональным требованиям относят:

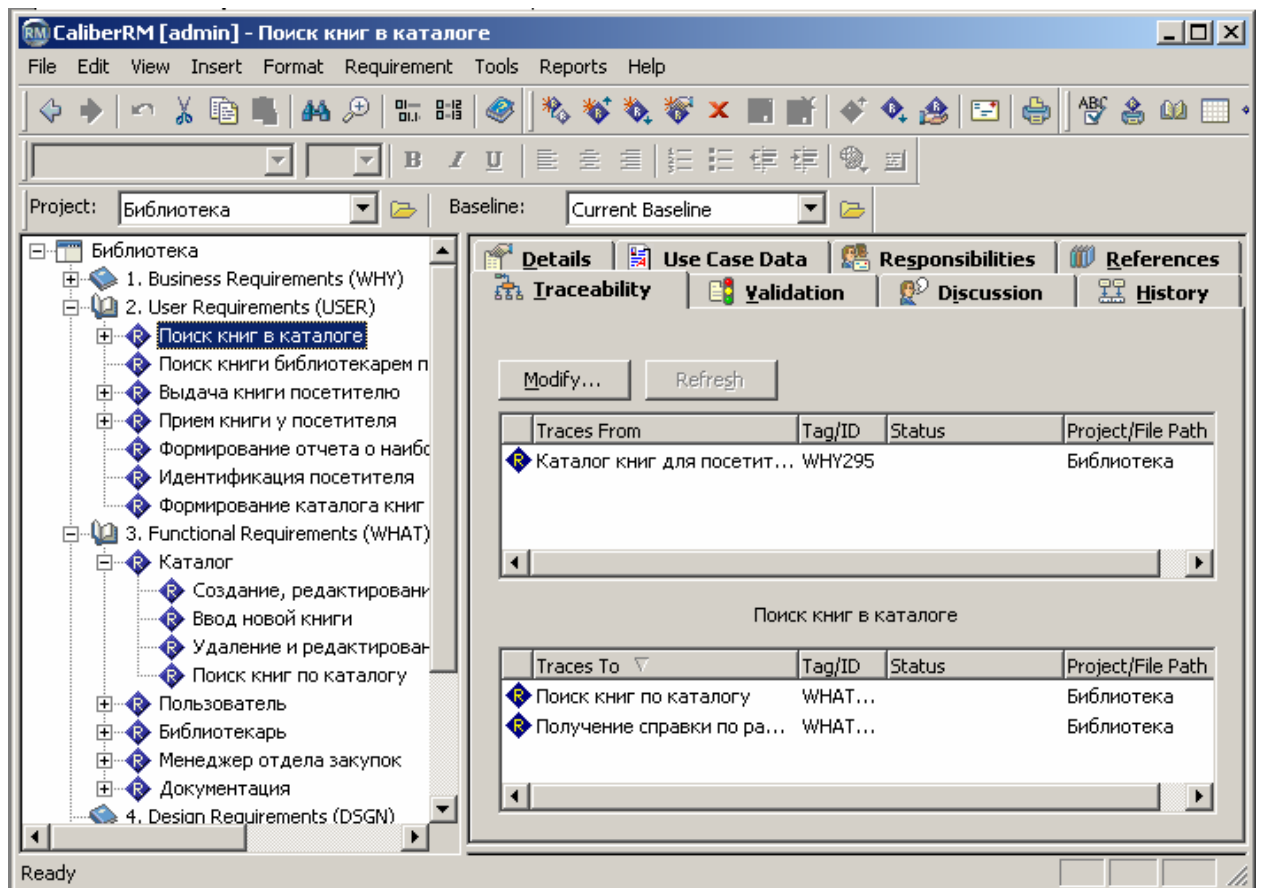
1. Бизнес правила. Под бизнес правилами понимаются процедуры и правила, регламентирующие особенности бизнес процессов в данной организации. Примерами бизнес правил являются следующие требования:
 - a. Расходы свыше 500 у.е должны утверждаться финансовым директором или его заместителем
 - b. Заявки на отпуск должны быть предоставлены не позднее 1 марта.
2. Атрибуты качества ПО. Примеры:
 - a. Допустимое количество сбоев в системе не более 1 сбоя в квартал
 - b. Режим работы сервера 24x7
 - c. Среднее время восстановления после сбоя не более 1,5 часа
3. Внешние интерфейсы. Набор программных интерфейсов, необходимых для взаимодействия с существующим ПО.
4. Ограничения. В рассматриваемом примере руководство выдвигает ограничение на затраты приобретения оборудования не более 10 тыс у.е. Ограничения могут носить не только финансовый характер.

При формировании нефункциональных требований следует формулировать их таким образом, чтобы они были контролируемы и измеряемы. Очень часто этого не делается, а требования формулируются в виде пожеланий, например, разрабатываемое ПО должно быть дружелюбным для пользователя. В результате такие требования приводят к тому, что разработчики вынуждены сами придумывать себе работу, которая как, потом оказывается была сделана ими впустую.

Формирование и отслеживание связей (Traceability)

Как мы уже упоминали, отслеживаемость связей является важной частью управления требованиями. Связи требований позволяют отследить источники возникновения требования. Так же отслеживаемость связей крайне важна при анализе влияния изменений требований (impact analyze).

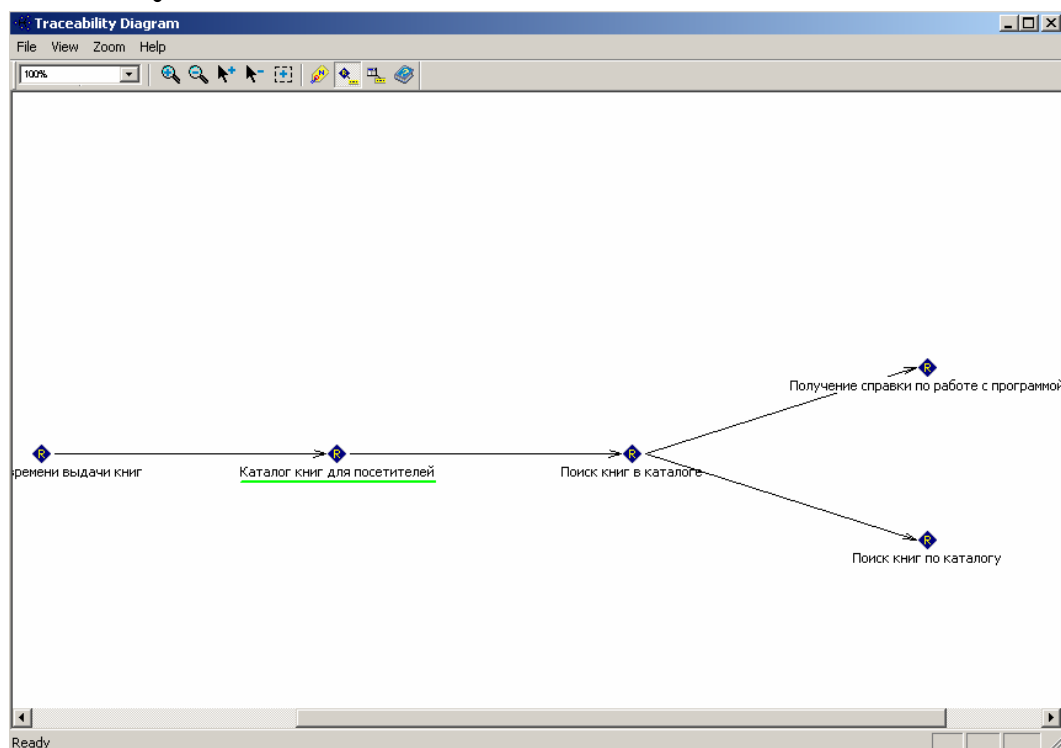
Управление отслеживаемостью связей осуществляется на закладке Traceability диалога просмотра требования.



Закладка отображает как ссылающиеся на данное требование требования (Trace To), так и те требования, которые являются корневыми для текущего требования (Trace From). Кнопка Modify позволяет добавлять новые связи обоих видов.

В состав Caliber RM так же позволяет быстро и качественно проводить анализ связей требований между собой. Для этого предназначены два инструмента:

- Traceability Diagram. Диаграмма связей требования. Позволяет отследить связи выбранного требования.



- Матрица связей. Инструмент отображающий взаимосвязи требований в виде таблицы

The screenshot shows the 'Traceability Matrix' application interface. It features a menu bar (File, Edit, View, Help) and a toolbar with various icons. The main area is a large table with requirements listed on the left and their relationships on the right. The requirements listed on the left include: Поиск в найденном, Поиск книги библиотекарем по ISBN, Выдача книги посетителю, Формирование квитанции, Прием книги у посетителя, Закрытие квитанции, Формирование отчета о наиболее востребованных книгах, Идентификация посетителя, Формирование каталога книг, Каталог, and Создание, редактирование, удаление новой категории книг. The table cells contain arrows indicating the direction of the relationships between these requirements.

Диаграмма связей требования позволяет быстро понять, откуда возникло анализируемое требование, и какие требования вытекают из данного требования. При анализе влияния изменений данная диаграмма является незаменимым инструментом аналитика. Матрица связей позволяет охватить всю картину взаимосвязей требований. Удобной и часто используемой возможностью матрицы является подсветка требований не имеющих связей с другими требованиями. Данная возможность позволяет быстро проверить качество и полноту заполнения связей между требованиями.

Формирование спецификации требований

Для формирования различных документов, являющихся результатом анализа требований, в состав CaliberRM входит средство Document Factory. Принцип работы Document Factory построен на использовании шаблонов Word и набора специальных тегов CaliberRM. Обсуждение набора тегов выходит за рамки данной статьи. Однако в ходе работы над проектом, был сформирован шаблон спецификации требований, который вы можете найти в прилагаемых к статье файлах.

Запросы на изменение требований

В ходе работы над проектом требования будут неоднократно дополняться, добавляться и изменяться. Изменения требований являются одной из главных причин затягивания сроков проекта.

Основными проблемами, возникающими в проекте при изменении требований, являются:

1. Отсутствие фиксации источника изменений. Часто при поступлении изменений не отслеживается источник возникновения изменений, что не позволяет в дальнейшем распределить ответственность за затраченные на реализацию изменения ресурсы.
2. Отсутствие процедуры утверждения изменений. Изменения могут вноситься различными участниками проекта, а ответственность за использование ресурсов проекта и его удачное завершение лежит на узком круге лиц. Поэтому необходимо организовать процедуру утверждения вносимых требований ответственными лицами.
3. Не оценивается влияние вносимых изменений на связанные требования.

Одним из решений описанных проблем является создание так называемой доски изменений (Change Control Board). Проект CaliberRM содержит группу требований Suggestion Box, в которую участники могут направлять запросы на изменение требований. При этом в запросе на изменение автоматически фиксируется его автор. Аналитик требований изучает и обсуждает поступившие изменения с заинтересованными лицами. С помощью матрицы связей аналитик может оценить степень влияния изменений на существующие требования и ресурсы, необходимые для реализации предложенных изменений. После рассмотрения изменения оно может быть либо принято к реализации в текущей версии, либо отвергнуто, либо отложено.

Таким образом, с помощью CaliberRM аналитик требований имеет возможность упорядочить процесс внесения требований и обеспечить реализацию требований в отведенные сроки.

Заключение

Итак, я надеюсь, что смог убедить Вас, что управление требованиями не просто очередная «теоретическая» выдумка, а набор методик, знаний и навыков, позволяющих сделать развитие проекта предсказуемым и управляемым, завершить его в срок и в рамках отведенных ресурсов.

Первой целью данной статьи было ознакомить читателя с базовыми понятиями дисциплины «Управление требованиями». Второй целью статьи было рассказать о замечательном инструменте управления требованиями CaliberRM, выпускаемом фирмой Borland. Объем статьи не позволяет рассказать о всех тонкостях работы с требованиями и использования CaliberRM.

Если у вас возникнут вопросы по темам, затронутым в данной статье, вы можете связаться с автором – Михаилом Головановым – по электронной почте golovanov@unislabs.com.