

Инструментальные средства информационных систем

Группа: БИН-19-1

Лекция 8 Инструменты управления требованиями

Содержание

Введение	2
1 Понятие управления требованиями	2
2 Характеристики требований	3
3 Виды требований	5
4 Этапы разработки требований	9
5 Задачи систем управления требованиями	12
6 Обзор основных функций и возможностей систем управления требованиями	14
7 Примеры систем управления требованиями	15
8 Лучшие бесплатные инструменты управления требованиями	22

Введение

Мы рассмотрели задачи решаемые в процессе управления проектами и отслеживания выполнения задач. Теперь рассмотрим смежную область этих процессов, речь пойдет о требованиях.

Прежде чем в проекте, в backlog проекта, появятся задачи сначала будут созданы требования. Требования – это краткое описание чего хочет пользователь или заказчик, но уже не его история, а полноценное требование.

На первый взгляд кажется, что тут нечего особо изучать и автоматизировать, что управление требованиями можно свести к трем действиям: есть набор требований к программному обеспечению, т.е. кто-то их должен придумать, затем стоит их записать и затем реализовать ... и затем забыть. Но существует ряд задач в управлении требованиями, для которых создаются целые автоматизированные системы управления требованиями. Например, у вас IT компания, которая пишет программное обеспечение на заказ, функции в разных проектах повторяются, чтобы как-то учитывать, что и где было реализовано и отслеживать возможность повторного использования кода, и ведутся систем управления требованиями.

Существует книга Карла Вигерса – Разработка требований к программному обеспечению, почти 600 страниц, старые издания доступны в интернете.

Системы управления требованиями позволяют экономить время на разработку и другие ресурсы, в основном, время и деньги, при реализации проектов.

(Рассказать пример из жизни компании, кто является “системой управления требованиями” и о велосипедах в коде. Одинаковые виды просмотра и выбора объектов, одинаковый функционал у разных процессов имеют отдельную реализацию (Претензии 4 вида, Заказы 6 видов).)

В данной лекции рассмотрим какие задачи должны выполнять система управления требованиями и какими функциями должна обладать, чтобы в дальнейшем на практике можно было легко подобрать нужную систему или написать свою.

1 Понятие управления требованиями

Перед тем, как управлять требованиями разберемся, что такое требование и что такое управление требованиями и зачем это нужно.

Управление требованиями — процесс, включающий идентификацию, выявление, документацию, анализ, отслеживание, приоритезацию требований,

достижение соглашений по требованиям и затем управление изменениями и уведомление заинтересованных лиц.

Управление требованиями — непрерывный процесс на протяжении всего жизненного цикла продукта.

Системы управления требованиями (СУТ, англ. Requirements Management Systems, RMS) помогают аналитикам, проектировщикам и руководителям проводить сбор, фиксирование требований, их систематизацию, приоритизацию, построение взаимосвязей.

Требование — это любое условие, которому должна соответствовать разрабатываемая система или программное средство.

Требованием может быть **возможность**, которой система должна обладать и **ограничение**, которому система должна удовлетворять.

В соответствии с *Глоссарием терминов программной инженерии IEEE*, являющимся общепринятым международным стандартным глоссарием, **требование** это:

1. Условия или возможности, необходимые пользователю для решения проблем или достижения целей;
2. Условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
3. Документированное представление условий или возможностей для пунктов 1 и 2.

В соответствии со стандартом разработки требований ISO/IEC 29148, **требование** — это утверждение, которое идентифицирует эксплуатационные, функциональные параметры, характеристики или ограничения проектирования продукта или процесса, которое однозначно, проверяемо и измеримо. Необходимо для приемки продукта или процесса (потребителем или внутренним руководящим принципом обеспечения качества).

Также глоссарий ITIL v3 (*IT Infrastructure Library, сейчас есть v 4*) определяет такое понятие, как **набор требований** — документ, содержащий все требования к продукту, а также к новой или измененной ИТ-услуге.

2 Характеристики требований

Требования должны быть сформулированы определённым образом, например требование *“сделайте чтобы работало быстро и не было технических сбоев”* – это не требование. Правильное требование должно быть примерно таким: *“время отклика системы должно примерно столько секунд”*, ну а сбои вообще неизбежны (тоже самое отключение электричества), поэтому требование должно звучать так: *“время восстановления системы после сбоя должно быть 2 часа”*.

Требование высказанное устно и нигде не зафиксированное тоже не является требованием.

Требование должно обладать следующими характеристиками:

1. Единичность — требование описывает одну и только одну вещь.
2. Завершенность — требование полностью определено в одном месте и вся необходимая информация присутствует.
3. *Последовательность (сомнительно)* — требование не противоречит другим требованиям и полностью соответствует документации. Спорно, так как требования могут меняться со временем.
4. Атомарность — требование нельзя разделить на более мелкие.
5. Отслеживаемость — требование полностью или частично соответствует деловым нуждам как заявлено заинтересованными лицами и задокументировано.
6. Актуальность — требование не стало устаревшим с течением времени.
7. Выполнимость — требование может быть реализовано в рамках проекта.
8. Недвусмысленность — требование определено без обращения к техническому жаргону, акронимам и другим скрытым формулировкам. Оно выражает объекты и факты, а не субъективные мнения. Возможна одна и только одна его интерпретация. Определение не содержит не четких фраз, использование отрицательных и составных утверждений запрещено.
(*Про жаргоны и акронимы. ДД и КО не могут закрутить ТБД, так как ОСГ не зашит и SKU в ООС.*)
9. Обязательность — требование представляет собой определенную заинтересованным лицом характеристику, отсутствие которой ведет к неполноценности решения, которая не может быть проигнорирована. Необязательное требование — противоречие самому понятию требования.
10. Проверяемость — реализованность требования может быть проверена.
11. Полнота – самое сложное требование. Полнота системы требований – свойство, означающее, что совокупность артефактов, описывающих

требования, исчерпывающим образом описывает все то, что требуется от разрабатываемой системы. Иначе говоря, надо зафиксировать все, что система должна делать. Если на начальном этапе это не указать, то можно сильно ошибиться при проектировании и выборе архитектуры. При этом, надо понимать, что очень трудно сразу указать для системы, которую еще только надо спроектировать, весь функционал. Поэтому, часто на первых этапах указывают требования крупными мазками, а не в терминах функциональных требований.

Придумать примеры требований для игры “вот требование, что в нём не так, сформулируйте правильно”.

(Необходимо, чтобы на лист входило как можно больше информации.)

В соответствии с ITIL v3 все требования в проекте можно разделить на следующие группы:

1. Функциональные (Functional) — реализуют саму бизнес-функцию.
2. Управленческие (Manageability) — требования к доступным и безопасным сервисам; относятся к размещению системы, администрированию и безопасности.
3. Эргономические (Usability) — к удобству работы конечных пользователей.
4. Архитектурные (Architectural) — требования к архитектуре системы.
5. Взаимодействия (Interface) — к взаимосвязям между существующими приложениями и программными средствами и новым приложением.
6. Сервисного уровня (Service Level) — описывают поведение сервиса, качество его выходных данных и другие качественные аспекты, измеряемые заказчиком.

Какие требования следует выполнять в первую очередь? Удобство. При демонстрации созданных наработок, пусть сразу будет реализовано удобство. Чтобы расположить пользователей к продукту и склонить их к активному тестированию.

3 Виды требований

Требования можно разделить на две большие группы:

- 1) функциональные требования
- 2) нефункциональные требования

Функциональные требования отвечают на вопрос “что”, что система должна делать.

К функциональным требованиям относят следующие требования и они представляют собой иерархию из трех уровней:

1. Бизнес-требования - цели заказчика программного продукта, которые необходимо достигнуть путем реализации этого продукта.
2. Требования пользователей - цели и задачи, которые будут решать пользователи с помощью программного продукта.
3. Функциональные требования - подробное описание того, как должны быть реализованы требования пользователей.

1. **Бизнес-требования.** Что система должна делать с точки зрения бизнеса. Слово "бизнес" в данном контексте ближе к слову "заказчик". Пример бизнес-требования: промо-сайт, привлекающий внимание определенной аудитории к определенной продукции компании.

2. **Пользовательские требования** – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Эти требования часто представляют в виде вариантов использования (Use Cases). Иначе говоря, пользовательские требования - это что может сделать пользователь: зарегистрироваться, посмотреть определенную информацию, пересчитать данные по определенному алгоритму и прочее.

3. **Функциональные требования** – определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований и в контексте пользовательских требований. Другими словами, что будут делать разработчики, чтобы выполнить пользовательские требования.

В группу функциональных требований относят и системные требования (*Это странно, думаю, что не совсем так, надо прочитать в источнике.*). Эти характеристики могут описывать требования как к аппаратному обеспечению (тип и частота процессора, объем оперативной памяти, объем жесткого диска), так и к программному окружению (операционная система, наличие установленных системных компонентов и сервисов и т. п.). Обычно такие требования составляются производителем или автором ПО. Например, для игры это могут быть требования такого типа: видеокарта - объем памяти от 64 Мб, совместимость с DirectX 9.0b и новейшие драйвера. Для сайта: ОС - Windows не ниже XP, браузеры IE не ниже 7.0 и так далее.

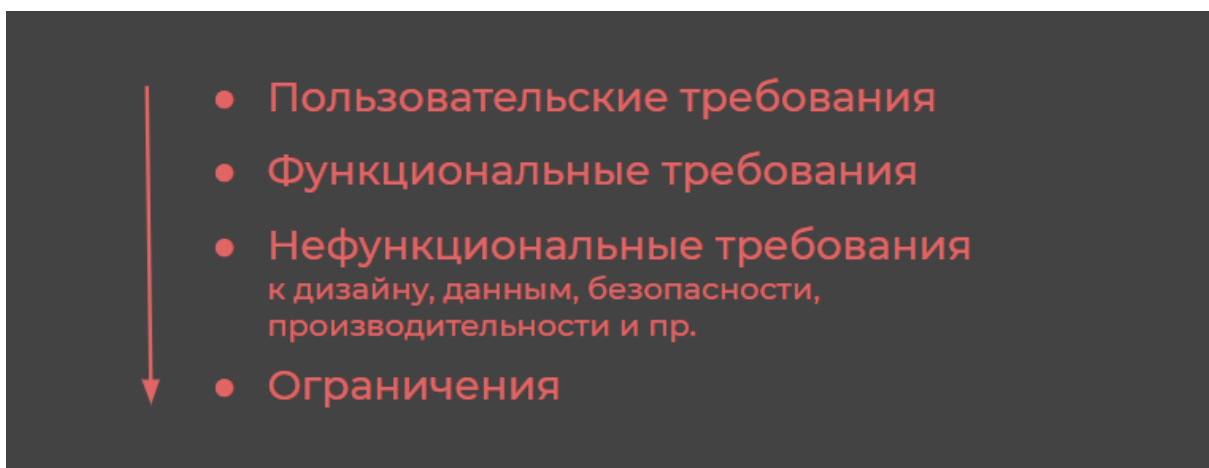
Почему важно указывать системные требования и утверждать их у заказчика? Если не указать, например, что важно обеспечить просмотр сайта в IE 6, то разработчики вполне могут выбрать такое архитектурное решение, которое не позволит корректно отображать сайт. Системные требования напрямую зависят от целевой аудитории проекта.

Вторая группа требований это **нефункциональные требования**. Иначе говоря, как будет работать система и почему именно так. Т.е. отвечают на вопрос как.

1. **Бизнес-правила.** Они определяют почему система работать должна именно так, как написано. Это могут быть ссылки на законодательство, внутренние правила заказчика и прочие причины. Часто упускают этот раздел и получается, что некоторые системные решения выглядят нетипичным и совсем неочевидными. Например, многие табачные компании и компании, производящие алкоголь требуют постоянного доказательства того, что промо-сайтами пользуются люди, достигшие определенного возраста. Это бизнес-правило (подтверждение возраста) возникает по требованию этических комитетов заказчика, хотя и несколько противоречит маркетинговым целям и требованиям по usability.
2. **Внешние интерфейсы.** Это не только интерфейсы пользователя, но и протоколы взаимодействия с другими системами. Например, часто сайты связаны с CRM системами. Особенности протокола взаимодействия "сайт-CRM" также относятся к нефункциональным требованиям.
3. **Атрибуты качества.** Атрибуты касаются вопросов прозрачности взаимодействия с другими системами, целостности, устойчивости и т.п. К таким характеристикам относятся:
 - легкость и простота использования (usability)
 - производительность (performance)
 - удобство эксплуатации и технического обслуживания (maintainability)
 - надежность и устойчивость к сбоям (reliability)
 - взаимодействия системы с внешним миром (interfaces)
 - расширяемость (scalability)
 - требования к пользовательским и программным интерфейсам (user and software interface).

Например:

- Анкета должна содержать файл с фото, так как фото необходимо при оформлении документов — это **бизнес-требование**. А возможно, и **бизнес-правило**.
- Из бизнес-требования следует, что у пользователя должна быть возможность прикрепить фото к анкете — это **пользовательское требование**. То есть требование, описывающее действия пользователя.
- Получается, что система должна иметь функционал прикрепления фото к анкете — это уже **функциональное требование**, описывающее поведение системы. Или как должна работать система, чтобы выполнять исходное пользовательское требование.
- Будем хранить все фото в формате base64 в отдельной таблице в БД. Это — **нефункциональные требования**.
- Фото в очень хорошем качестве нам не нужно, а также мы не хотим покупать много памяти для сервера. Поэтому сделаем **ограничение** на размер загружаемого фото: не более 10Мб.



На каждое бизнес-требование, как правило, приходится несколько пользовательских. Пользовательское требование декомпозируется на какое-то число функциональных. К каждому функциональному требованию нужно продумать нефункциональные требования и ограничения.

Также нефункциональные требования и ограничения могут напрямую вытекать как из пользовательских требований, так и из бизнес-требований и правил.

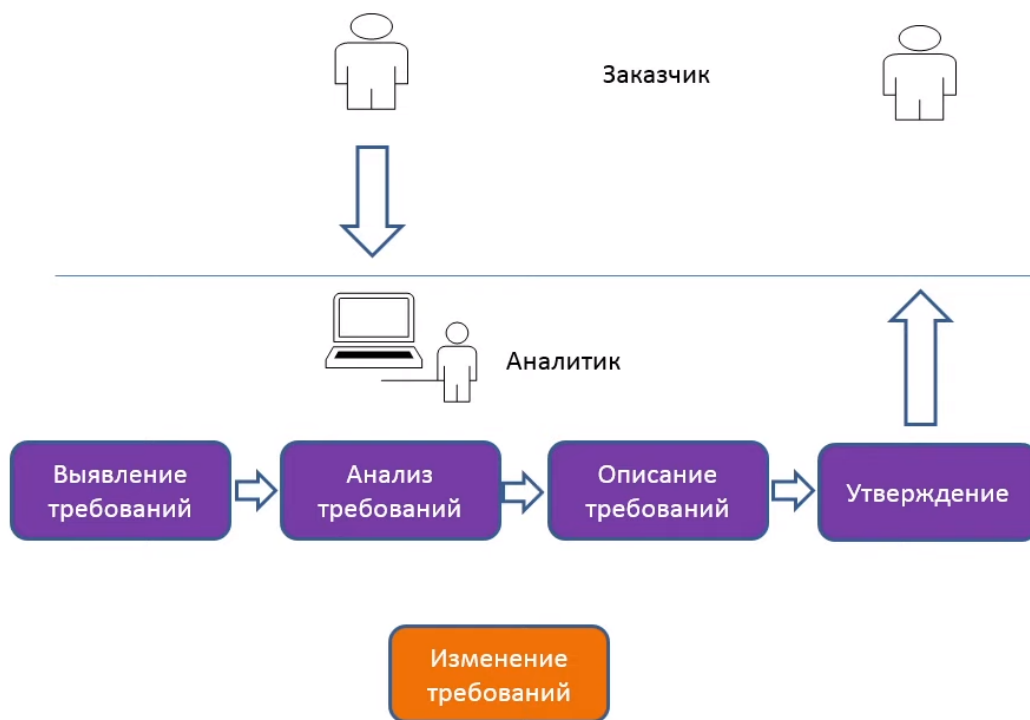
Таким образом получаются деревья из требований, в вершине каждого из которых — бизнес-требование.

4 Этапы разработки требований

Иногда заказчик думает, что он всё вам рассказал, что хочет, но вы его абсолютно понять не можете.

Поэтому стоит следовать определенному плану, чтобы требование получилось.

Этапы в управлении требованиями (картинка из лекции на youtube):



(Взято из статьи <https://habr.com/ru/post/501468/>)

Если коротко, то основные этапы разработки требований — это:

1. Зачем нам что-то делать? *(нужно больше золота)*
2. Что мы будем делать? *(все как у людей, но дешевле)*
3. Как мы это сделаем? *(с блокчейном и датасаентистами, естественно)*
4. Когда мы это сделаем? *(вчера, а отрефакторим «потом»)*

1 Зачем – Выявление требования

Даже если пользователь просит сделать что-то очень срочно и очень быстро, при этом вы сами видите, что доработка мелкая, все равно надо спросить “зачем?” это нужно.

Часто после выяснения зачем – т.е. установки цели, задача может поменяться или вовсе отпасть.

(Пример с ролью “просмотр всего”).

На первом этапе четко установите цель и критерии, при выполнении которых цель считается достигнутой. Все нужно согласовать с заказчиком.

Как извлечь требование из заказчика?

Бывают ситуации, когда выяснить что хочет заказчик или пользователь очень сложно, приходится буквально клещами тянуть. Но есть другой вид пыток заказчиков – это метод пять почему.

(Пример:

Заказчик: В системе числится отрицательный остаток пирогов. Исправьте остаток.

Аналитик/разработчик:

- 1. Почему пирогов числится отрицательное количество? – Потому что при продаже пирогов остаток был равен ноль*
- 2. Почему остаток был равен ноль? – Потому что не было прихода новых пирогов.*
- 3. Почему не было прихода новых пирогов? – Потому что повар не поставил пироги на приход.*
- 4. Почему повар не поставил пироги на приход? – Потому что повар ставит на приход всю приготовленную продукцию в конце дня.*
- 5. Почему повар сканирует все выпущенные блюда в конце дня, а не сразу при производстве? – Потому что сканер готовых блюд находится в другом конце магазина, повар не может ходить туда сюда каждый раз, потому что это долго (что-то пригорит) или повар не может покидать стерильную кухню.)*

2 Что – Анализ требования

Цель достигается разными путями. И второй важный шаг при разработке требований как раз про выбор пути — что конкретно мы будем делать, чтобы прийти к цели.

Чтобы продумать все варианты, надо разобраться — а что же происходит сейчас? Как устроен процесс без вашей системы, как работают пользователи и заказчики? Даже если процесса еще нет, подробная информация про текущее состояние очень важна. Так мы поймем, какое решение устранил проблему, а не создаст еще одну.

Т.е. мы увидим что надо поменять чтобы попасть в целевое состояние, какие шаги нужно сделать, чтобы от сейчас прийти к тому как надо.

У каждого варианта реализации свои плюсы и минусы, свои необходимые ресурсы и свой уровень результата. Смоделировав все опции, проработав или хотя бы просто проговорив с заинтересованными сторонами эту информацию — вы сможете сделать взвешенный и обоснованный выбор.

3 Как – Описание требования

Итак, мы поняли нашу цель и что мы будем делать, чтобы ее достичь. Осталось разобраться с тем — как мы это реализуем: какие страницы будем показывать пользователям, в каком виде отобразим отчет для заказчика, как получим данные из другой системы, как будем хранить их у себя и так далее.

Этот этап — дело техники. И если вы успешно прошли предыдущие два — будет гораздо проще.

Хоть техника и зависит от контекста, полезно двигаться по “чек-листу” Вигерса и других умных людей. Если для вас какой-то тип требований сейчас не актуален — окей, не описываем. Но важно не забыть и проверять себя.

По пункту лекции 2.

На каждое бизнес-требование, как правило, приходится несколько пользовательских. Пользовательское требование декомпозируется на какое-то число функциональных. К каждому функциональному требованию нужно продумать нефункциональные требования и ограничения.

Также нефункциональные требования и ограничения могут напрямую вытекать как из пользовательских требований, так и из бизнес-требований и правил.

Таким образом получаются деревья из требований, в вершине каждого из которых — бизнес-требование.

4 Когда – Описание требования

В “лесу” ваших требований скорее всего найдется сколько-нибудь взаимоисключающих и сколько-нибудь повторяющихся. Поэтому полезно всю эту красоту документировать и представлять в виде таблиц и диаграмм.

Тут есть много инструментов: например, BPMN для описания бизнес-процессов и UML для создания схем взаимодействий сервисов и компонентов.

Если у вас получается объяснять всем, что и как вы хотите сделать с системой, при помощи салфетки и 3х пятен от кофе — это потрясающе.

Однако, как правило, «пятенный» уровень детализации не позволяет увидеть подводные камни и продумать все возможные сценарии. Ведь вроде и так все понятно, а нарисовал схему — и вот бесконечный вызов, и забытая ветвь процесса.

Поэтому полезно знать, какие есть инструменты для обращения хаоса в порядок.

В схематическом и структурированном виде требования нужно приоритизировать — в зависимости от полезности (это вам скажет заказчик и пользователи) и трудоемкости (это вам скажет команда разработки).

А дальше можно уже раскидывать по спринтам/этапам разработки и внедрения. Ну и повторять эти упражнения в рамках каждой итерации.

Подробнее про требования автор статьи рекомендует почитать в книге Вигерса и Битти: “Разработка требований к программному обеспечению”.

5 Задачи систем управления требованиями

Задачи решаемые системами управления требованиями:

- 1) хранить требования – репозиторий,
- 2) отслеживать жизненный цикл требования – появилось, в работе, выполнено, протестировано, передано в продуктив,
- 3) поиск требования, например, при проверке на дублирование,
- 4) трассировка требований – при изменении одного из требований, отображаются связанные требования, на которые может влиять данное изменение,

5) повторное использование реализации требования

Как можно понять из приведенного выше обзора программного обеспечения для управления требованиями все оно базируется на одном принципе — человек, а в данном случае, аналитик, вводит требование в систему, смотрит, нет ли такого требования в системе уже. Если требование в той или иной формулировке уже присутствует в системе, то заново его не заносит, а отмечает, как дублирующее. В связи с тем, что поиск схожих требований вручную является сложной и трудозатратной задачей, которая требует постоянного участия аналитика, то в процессе управления требованиями именно ее стоит автоматизировать в первую очередь.

Чтобы реализовать возможность поиска и повторного использования требований необходима методика идентификации требований, представленных текстом на естественном языке. Тогда, представив требование не только как текст, но и совокупность некоторых концептов или лингвистических переменных станет возможным не только выполнять поиск сходных требований, но и использовать повторно созданные в процессе реализации требования артефакты. Под артефактами в данном случае не стоит понимать только исходный код, который реализует требование, но и жизненный цикл, которое оно прошло, т.е. код не всегда можно использовать повторно из-за специфики задач, но можно получить консультацию у его разработчиков. При разработке нескольких продуктов для одной предметной области это становится актуальной задачей. Например, при адаптации системы электронного документооборота на нескольких предприятиях часто возникает ситуация, что на разных предприятиях разные документы проходят одни и те же маршруты в процессе жизненного цикла и функциональность, которая реализует эти маршруты может быть использована повторно, пусть и без полного копирования.

Предположим, что следующие требования описывают маршруты документов соответственно на предприятиях А и Б:

А — {A, B, C, D, E}

Б — {F, B, C, D, E}

Под F и А имеются в виду документы, а B, C, D, E — их маршруты.

Рассчитав расстояние Хэмминга между этими двумя требованиями мы получим единицу, так как они различны только в одной позиции и, следовательно, при реализации требования Б стоит обратить внимание на требование А и его реализацию. Естественно, решение о повторном использовании принимает уже разработчик или другое лицо, принимающее решение, но уже хорошо, когда есть варианты, где можно подсмотреть реализацию.

(**Расстояние Хэмминга** (кодировое расстояние) — число позиций, в которых соответствующие символы двух слов одинаковой длины различны. В более общем случае расстояние Хэмминга применяется для строк одинаковой длины любых q -ичных алфавитов и служит метрикой различия (функцией, определяющей расстояние в метрическом пространстве) объектов одинаковой размерности.)

6 Обзор основных функций и возможностей систем управления требованиями

Администрирование

Возможность администрирования позволяет осуществлять настройку и управление функциональностью системы, а также управление учетными записями и правами доступа к системе.

Импорт/экспорт данных

Возможность импорта и/или экспорта данных в продукте позволяет загрузить данные из наиболее популярных файловых форматов или выгрузить рабочие данные в файл для дальнейшего использования в другом ПО.

Многопользовательский доступ

Возможность многопользовательской доступа в программную систему обеспечивает одновременную работу нескольких пользователей на одной базе данных под собственными учетными записями. Пользователи в этом случае могут иметь отличающиеся права доступа к данным и функциям программного обеспечения.

Наличие API

Часто при использовании современного делового программного обеспечения возникает потребность автоматической передачи данных из одного ПО в другое. Например, может быть полезно автоматически передавать данные из Системы управления взаимоотношениями с клиентами (CRM) в Систему бухгалтерского учёта (БУ). Для обеспечения такого и подобных сопряжений программные системы оснащаются специальными Прикладными программными интерфейсами (англ. API, Application Programming Interface). С помощью таких API любые компетентные программисты смогут связать два программных продукта между собой для автоматического обмена информацией.

(Например, у нас обращения покупателей через сайт в разделе “напишите нам”, автоматически создается заявка в системе отслеживания задач (Service Desk) для маркетологов.)

Отчётность и аналитика

Наличие у продукта функций подготовки отчётности и/или аналитики позволяют получать систематизированные и визуализированные данные из системы для последующего анализа и принятия решений на основе данных.

7 Примеры систем управления требованиями

В настоящее время широкое распространение получили такие системы управления требованиями как IBM Rational RequisitePro, Rational DOORS, Sybase PowerDesigner и Borland Caliber RM.

Краткие переводы основных функциональных возможностей приведенных систем, взятые с сайтов производителей.

IBM Rational Requisite Pro

Обладает всем необходимыми функциями и выполняет все задачи систем управления требованиями.

Имеет старомодный интерфейс, но продолжает развиваться в сторону облачных решений и web-приложений.

Инструмент IBM Rational RequisitePro предназначен для организации коллективной работы аналитиков и автоматизации их деятельности в области управления требованиями при разработке программного обеспечения.

Основные возможности RequisitePro:

- Проектирование и развертывание процесса управления требованиями "с нуля"
- Интеграция с Microsoft Word
- Трассировка требований
- Web-интерфейс
- Авторизация доступа
- Поддержка методологии IBM Rational Unified Process
- Автоматическая генерация и публикация отчетов
- Механизм управления базовыми линиями репозитория
- Механизм подписки на изменения в требованиях

- Управление дискуссиями
- Интеграция с другими инструментами

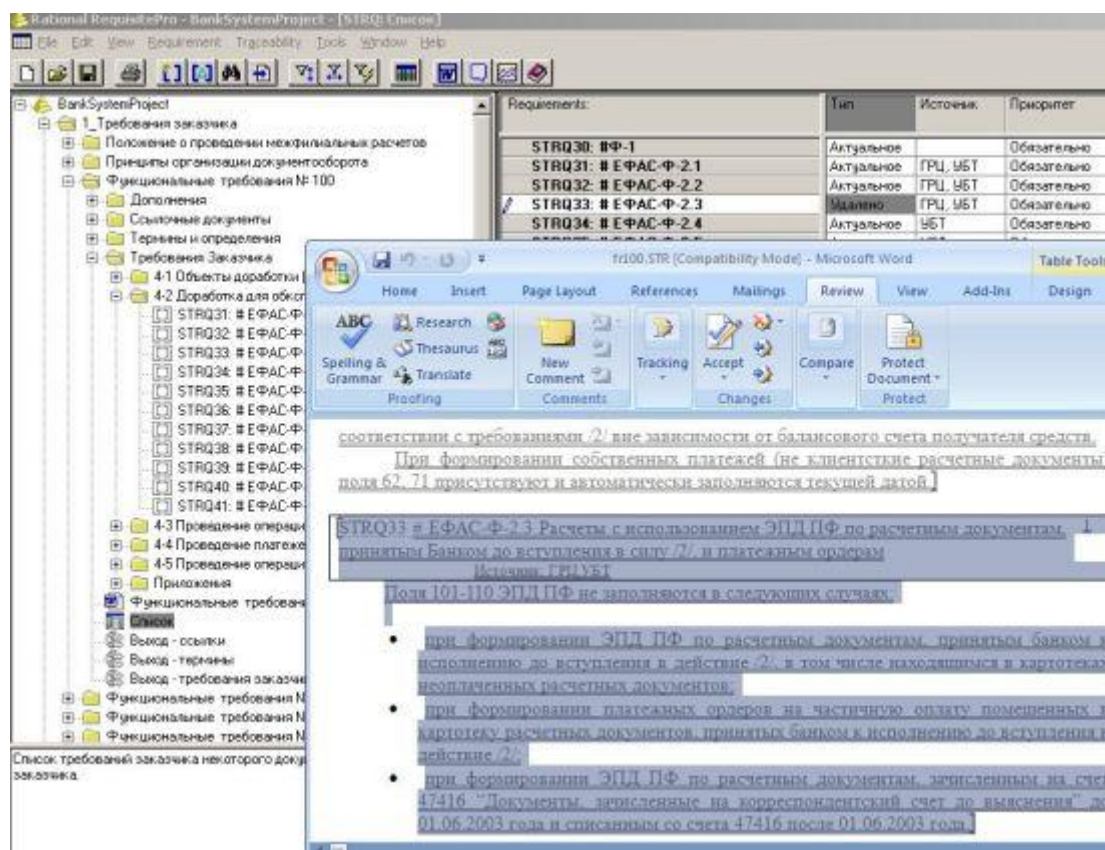


Рис. 1. IBM Rational RequisitePro - внешний вид

Проектирование и развертывание процесса управления требованиями "с нуля" предоставляет большие возможности для выстраивания процесса в организации, для которой управление требованиями является инновационной сферой деятельности или в которой очень специфические понятия о ней. При этом можно определить свои типы требований и необходимые атрибуты для них, свои типы документов и подготовить новые шаблоны документов Microsoft Word, выстроить иерархию требований (при этом требования могут находиться в разных документах), определить связи между требованиями для отслеживания изменений и т.д.

Интеграция с Microsoft Word значительно облегчает постановку процесса управления требованиями в организациях, в которых наработано огромное число документов в указанном формате. Но при этом, эти документы разобщены, в них сложно отслеживать изменения и тяжело находить необходимую информацию. Более того, непрерывным потоком идут различные дополнения и изменения к уже существующей информации. Тогда именно RequisitePro может оказать Вам неоценимую помощь.

В этом случае достаточно просто осуществить импорт существующих документов в репозиторий требований и выполнить в них полуавтоматическую "нарезку" на отдельные куски - требования необходимых типов. После этого

данные части документов фиксируются в репозитории в виде требований и могут быть связаны друг с другом. Такой подход в RequisitePro позволяет осуществлять удобную навигацию по информации. При этом не важно, находится эта информация в одном и том же или в разных документах.

Трассировка требований является одной из ключевых возможностей RequisitePro. Она означает, что при изменении любых требований наглядным образом активируются связи между ними. Таким образом, значительно облегчается работа по проверке актуальности информации в репозитории, т.к. требуется проверять не все данные, а только часть из них, логически связанную с выполненными изменениями.

В более широком смысле при применении RequisitePro в виде требований можно сохранять любые утверждения, ссылки на внешние источники или просто какую-либо значимую информацию. Мощные механизмы навигации и управления изменениями RequisitePro окажутся полезными и в этом случае.

Web-интерфейс RequisitePro предоставляет возможности полноценной удаленной работы с репозиторием требований без необходимости установки "толстого клиента". В последних версиях Web-интерфейса RequisitePro отчетливо видна тенденция компании IBM, связанная с переводом ключевых инструментов на технологии Интернет. Функционал Web-интерфейса RequisitePro, начиная с версии 7, практически полностью реализует возможности "толстого клиента", включая большой набор административных функций.

Авторизация доступа в RequisitePro основана на современных механизмах в области управления безопасностью, группами пользователей и отдельными пользователями и настройки прав их доступа к данным.

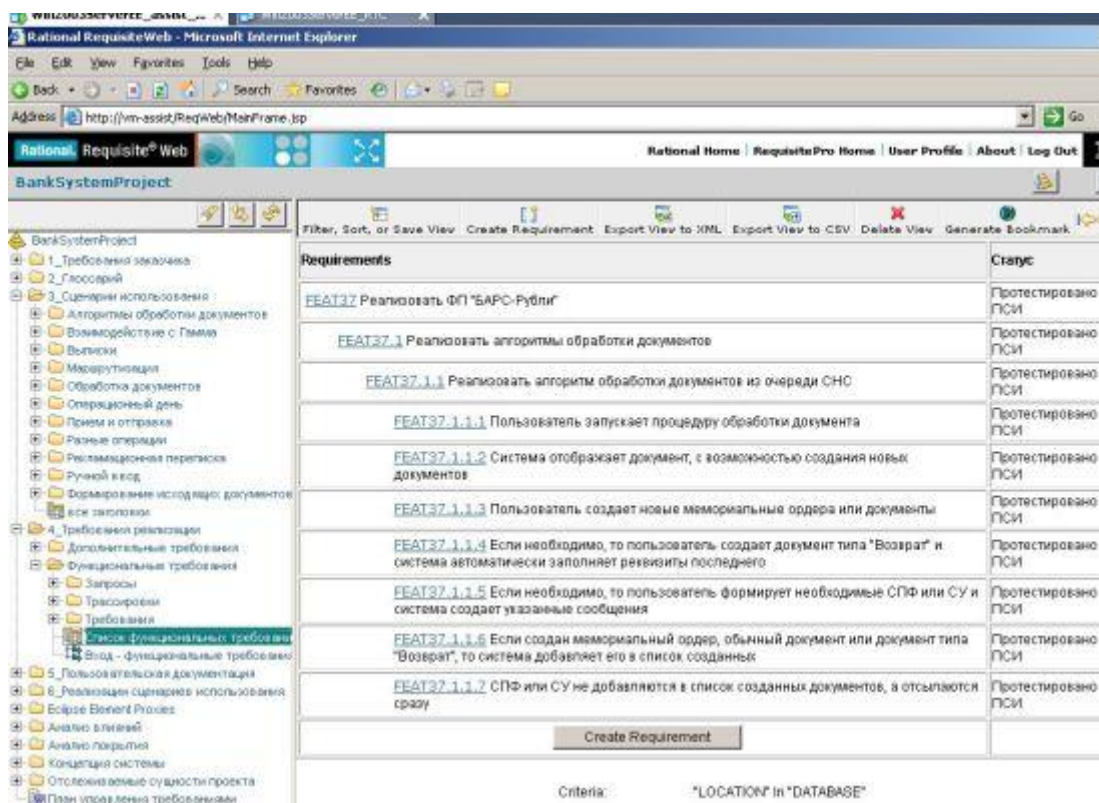


Рис.2. Web-интерфейс IBM Rational RequisitePro

Поддержка методологии IBM Rational Unified Process способна оказать помощь при необходимости ускоренного запуска процесса управления требованиями в организации. Если мировые "Best Practices" не являются пустым звуком для Вас и Ваша организация готова ориентироваться на них, то можно значительно снизить трудоемкость работ при внедрении управления требованиями.

Данная поддержка выражается в наличии готовых регламентов в составе IBM RUP, готовых шаблонов репозитория требований и шаблонов документов. При их использовании специалисты, знающие IBM RUP, легко включатся в работу с требованиями, и при этом организации не надо делать больших вложений на проектирование процесса управления требованиями и дорогостоящее приглашение консультантов для его внедрения в течение длительного времени.

Автоматическая генерация и публикация отчетов позволяет создавать и представлять данные репозитория требований в наглядной или унифицированной форме.

Для генерации отчетов используется инструмент BIRT (Business Intelligence & Reporting Tool), построенный на Eclipse в рамках направления Open Source. В составе пакета RequisitePro приходит специальный плагин, который и позволяет обучить инструмент BIRT внутреннему устройству первого и создавать необходимые отчеты.

На сервере публикаций RequisitePro можно опубликовать отчеты, созданные с помощью BIRT, в Web и, таким образом, сделать информацию по требованиям проекта доступной для всех заинтересованных лиц.

Механизм управления базовыми линиями репозитория создает основу для эффективного управления версиями и конфигурациями репозитория требований. Например, после выпуска очередного релиза системы можно создать снимок репозитория и в будущем в любой момент восстановить его структуру и данные.

Механизм подписки на изменения в требованиях позволяет любому специалисту, имеющему доступ в репозиторий требований, указать интересующие требования и затем периодически получать информацию по электронной почте о выполненных в них изменениях.

Управление дискуссиями окажет бесценную помощь на этапах выявления и согласования требований. Это удобный интерфейс, спроектированный по принципам классических форумов Интернета, и грамотно выстроенный функционал, с помощью которого в дискуссии могут участвовать конкретные пользователи, а предметом дискуссии могут быть одно или более требований.

Если же принципы форума Вам кажутся неудобными, то RequisitePro можно настроить так, что дискуссии будут вестись по электронной почте, но информация при этом будет складироваться в репозитории с возможностью обратиться к ней в любое время.

Интеграция с другими инструментами позволяет использовать RequisitePro для организации конвейерного процесса разработки ПО, основанного на управлении требованиями, начиная от их выявления и заканчивая проверкой в рамках тестирования.

Среди ключевых интеграций следует особенно отметить интеграцию с инструментами визуального моделирования IBM Rational Software Modeler и IBM Rational Software Architect, средством управления запросами на изменения IBM Rational ClearQuest, инструментом управления тестированием IBM Rational Quality Manager.

Благодаря интеграции со средствами визуального моделирования UML IBM Rational Software Modeler и IBM Rational Software Architect значительно упрощается работа по формированию спецификаций требований и уточнению существующих требований. Визуализация всегда способствовала преодолению сложностей в проектировании сложных систем и согласовании различных аспектов деятельности при командной работе. Аналогично и в данном случае, при грамотном использовании инструментов и правильном построении репозитория требований и моделей, можно значительно снизить затраты на управление требованиями и отслеживание изменений в них.

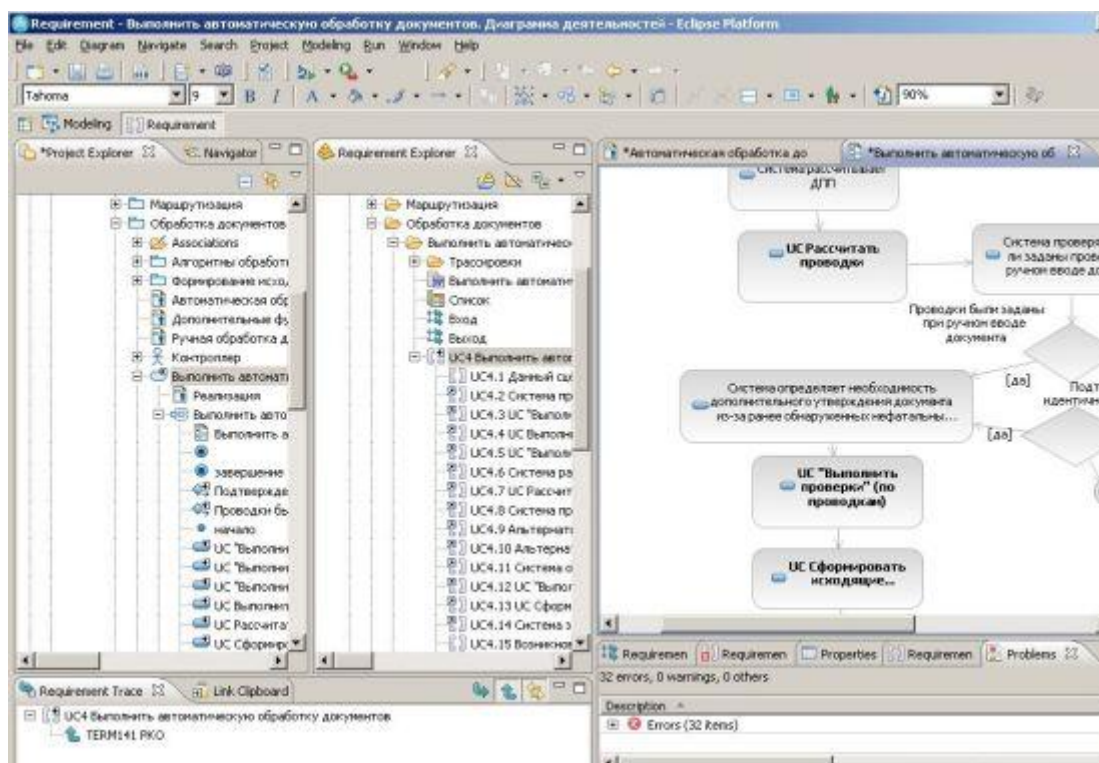


Рис.3. Интеграция IBM Rational RequisitePro и IBM Rational Software Architect (IBM RSA). Визуальное моделирование требований в IBM RSA

При этом можно связать практически любые элементы модели с любыми требованиями RequisitePro. Это означает, что с помощью простейших запросов будет легко отследить те элементы модели, которые были добавлены (для них еще не сформированы соответствующие требования RequisitePro), изменены (следует скорректировать требования) или удалены (необходимо объявить связанные требования RequisitePro устаревшими). И все это легко делается даже после значительных доработки и согласования моделей со многими участниками.

Интеграция с IBM Rational ClearQuest крайне важна, т.к. здесь фиксируется, как повлияло появление того или иного запроса (заявки, инцидента, запроса на улучшение или доработку, описания обнаруженной в системе ошибки и т.д.) на список требований к системе (какие требования были сформированы, скорректированы или признаны устаревшими). Верно и обратное - для любого требования всегда можно определить запросы, оказавшие влияние на него. Такая информация перед стартом итерации позволяет заинтересованным лицам согласовать, как должны быть реализованы запросы. По завершении итерации эта информация дает возможность продемонстрировать, были ли реализованы запросы и как это было сделано.

Интеграция с инструментом IBM Rational TestManager является необходимой при проведении качественного тестирования, которое должно подтвердить, реализованы ли сформулированные требования. Такая

интеграция позволяет получить следующие существенные преимущества: оценить степень покрытия требований сценариями тестирования и с большой точностью выявить сценарии тестирования, возможно которые следует скорректировать, т.к. изменились связанные с ними требования.

Открытость архитектуры IBM Rational RequisitePro позволяет интегрировать данное средство с огромным числом других инструментов, в том числе и Open Source. Особенно в этом ряду следует отметить популярное в последние годы средство Atlassian JIRA (коммерческая система отслеживания ошибок).

IBM Rational/Telelogic DOORS

IBM Rational/Telelogic DOORS — семейство решений для управления требованиями и создания сложных наукоемких изделий (авиа, судостроение, поезда, ракеты, автомобили т.п.).

Первоначально DOORS разрабатывался только как средство управления требованиями в процессе разработки программного обеспечения. Однако идеи, заложенные в DOORS, оказались успешными и в настоящий момент система используется даже в кампаниях, которые не имеют отношения к разработке программного обеспечения, но вынуждены контролировать большой объем взаимосвязанной информации, например, при разработке инженерных систем.

Из Telelogic DOORS можно получить следующую информацию:

- Статус выполнения работ по каждому требованию отдельно, а также по группе требований.
- Статус работы над всем проектом.
- Ответственное лицо для каждого требования или группы требований.
- Историю изменений требования.
- Ресурсы, которые потребуются для реализации требования еще до его внедрения в проект.
- Связь между требованиями заказчика, пунктами технического задания, программами верификации, тестирования и задачами управления проектом.
- Класс, модель или чертеж, в котором конкретное требование реализовано.

Borland Caliber RM

Borland Caliber RM – это корпоративная система управления требованиями, которая облегчает совместную работу, что позволяет группам разработчикам

подходить к вехам проекта вовремя и с запланированными затратами. Borland Caliber RM также помогает командам разработчиков удостовериться, что разрабатываемое приложение удовлетворяет пожеланиям конечных пользователей за счет непрерывного сбора пожеланий на всех этапах жизненного цикла от аналитиков, разработчиков, тестировщиков и других заинтересованных в проекте лиц.

Borland Caliber RM обладает следующими функциональными возможностями:

- Централизованное хранилище требований для всех проектов, разрабатываемых IT-компанией.
- Адаптируемость — Caliber RM можно сконфигурировать для использования в любом проекте, что повышает эффективность процесса управления требованиями.
- Трассируемость требований — открытая архитектура Caliber RM позволяет связать требования с другими артефактами на всех стадиях жизненного цикла программного продукта.
- Поддержка большого числа клиентов — Caliber RM прекрасно интегрируется с такими системами разработки, как Microsoft Visual Studio, Eclipse на платформе Windows.
- Интеграция с другими продуктами Borland для поддержки полного жизненного цикла программного продукта.

Другое ПО

Остались также достаточно известные системы управления требованиями, такие как:

- Sybase PowerDesigner
- OpenSource Requirements Management Tool
- RequirementsWin и другие

8 Лучшие бесплатные инструменты управления требованиями

На сегодняшний день существует множество отличных бесплатных инструментов управления требованиями. А что может быть лучше

того, что за них не придется платить ни копейки? Вот наш лучший выбор:

Платформа ALM для требований Visure —

Visure — одна из самых надежных платформ управления требованиями, специализирующаяся на **управление требованиями** для организаций всех размеров по всему миру. В число основных партнеров Visure входят критически важные для бизнеса и безопасности компании. Компания интегрирует все процессы управления жизненным циклом приложений, включая управление рисками, отслеживание проблем и дефектов, управление прослеживаемостью, управление изменениями и различные другие области, такие как анализ качества, управление версиями требований и мощные отчеты. Говоря о цене, Visure предлагает 30-дневную бесплатную пробную версию для начала.

[Confluence](#)

Confluence — это программное обеспечение для совместной работы, которое помогает командам разработчиков собирать идеи и обмениваться ими, совместно работать над проектами и организовывать работу. Он разработан Atlassian, той же компанией, что и Jira. Confluence предлагает бесплатную пробную версию на 14 дней без необходимости кредитной карты. Кроме того, у него есть бесплатная версия для команды до 10 человек.

[ClickUp](#)

Click.Up — это универсальная платформа управления работой, которая предлагает широкий спектр функций, включая управление задачами, управление проектами, CRM и многое другое. Он подходит для команд любого размера и поставляется с бесплатным планом навсегда, который включает неограниченное количество пользователей, задач, проектов, контрольных списков, комментариев и вложений. Говоря о стоимости, приложение предлагает бесплатную версию с хранилищем 100 ГБ, неограниченным количеством задач и пользователей.

[ReQtest --](#)

ReQtest — это онлайн-инструмент управления требованиями, который помогает вам управлять вашими требованиями на протяжении всего процесса разработки программного обеспечения. ReQtest предлагает модуль управления требованиями, который помогает вам управлять потребностями вашего бизнеса с помощью лучшего контроля, полной отслеживаемости и действенной информации. Он предлагает как платные планы, так и бесплатный план для небольших проектов с участием до трех пользователей.

[Jira](#)

Jira — популярный инструмент отслеживания проблем и управления проектами, разработанный Atlassian. Он подходит для команд любого размера и предлагает широкий спектр функций, включая управление требованиями, гибкое управление проектами, отслеживание времени и т. д. Jira предлагает бесплатную пробную версию на 14 дней без необходимости кредитной карты. Кроме того, у него есть бесплатная версия для десяти пользователей.

[Wrike](#)

Wrike — это инструмент управления требованиями, который предлагает такие функции, как управление задачами, совместная работа в команде, обмен файлами и т. д. Он подходит для команд любого размера и поставляется с бесплатным планом, который включает до неограниченного числа пользователей и неограниченное количество задач.

[Redmine](#)

Redmine — это гибкое веб-приложение для управления проектами, написанное с использованием фреймворка Ruby on Rails. Он подходит для команд любого размера и предлагает широкий спектр функций, включая управление требованиями, гибкое управление проектами, отслеживание проблем, отслеживание времени и т. д. У Redmine есть бесплатная версия с неограниченным количеством проектов и пользователей.

[RequirementsHub](#)

RequirementsHub — это инструмент управления требованиями, который помогает вам собирать, отслеживать и управлять вашими требованиями. Совместная работа и отслеживание среди заинтересованных сторон упрощаются благодаря единой панели инструментов. Участники из всех отделов могут легко представить информацию. Он предлагает как бесплатную, так и платную версии. Бесплатная версия подходит для небольших проектов с количеством пользователей до трех.

[TraceCloud](#)

TraceCloud — это инструмент управления требованиями, который помогает вам легко управлять требованиями на протяжении всего процесса разработки программного обеспечения. Кроме того, приложение предлагает такие функции, как управление рисками, сквозное соответствие требованиям и улучшение процессов. Он предлагает как платные планы, так и бесплатный план для небольших проектов с участием до трех пользователей.

[Xmind](#)

Xmind — это инструмент для картирования мыслей и мозгового штурма, который можно использовать для сбора требований и управления проектами. Он подходит для команд любого размера и поставляется с бесплатной версией без ограничений по проектам или пользователям.

[ZenTao](#)

ZenTao — это инструмент управления проектами с открытым исходным кодом, подходящий для Agile-команд любого размера. Он предлагает такие функции, как управление требованиями, отслеживание ошибок, управление тестированием и т. д. У ZenTao есть бесплатная версия с неограниченным количеством проектов и пользователей».

