



# **RiNet Трекер**

## **Руководство пользователя**

v 0.1a

(с) Кроникс-плюс,

Москва 2010

---

## Оглавление

Оглавление.....	2
Введение.....	3
Основные сведения.....	4
Элементы интерфейса .....	6
Авторизация.....	6
Основной экран.....	6
Расширенный поиск.....	8
Окно редактирования заявки.....	9
Журнал изменений заявки.....	9
Пример использования.....	11
Описание работы системы на примере учета багов и модификаций программы.....	11

## Введение

В данном руководстве описываются элементы пользовательского интерфейса Трекера и возможности его использования конечными пользователями.

Создание и модификация деловых процедур (классов задач) описаны в отдельном документе «RiNet-Трекер. Конфигуратор». Там же описано управление пользователями и их правами.

Это руководство пользователя состоит из трех основных частей.

В первой приводятся основные сведения о Трекере и даются определения важнейшим терминам.

Во второй части описываются элементы интерфейса Трекера.

Третья часть представляет собой пример использования Трекера для поддержки решения одной из рутинных задач организации.

## Основные сведения

Трекер — это система учета заявок. Исторически Трекер разрабатывался для учета аварий в сети. Достаточно быстро выяснилось, что кроме аварий с помощью такой системы можно подвергнуть учету значительное число рутинных бизнес-процессов.

В отличие от стандартных систем учета аварий и баг-трекеров RiNet-Трекер позволяет создать нужное количество бизнес-процессов и настроить их точно под нужды организации. Таким образом Трекер представляет собой систему автоматизации потоков работ (work-flow).

Рутинные бизнес-процессы, автоматизируемые в Трекере мы дальше будем называть классами задач.

Настройке в Трекере поддается очень многое — начиная с состава и типов полей, в которых хранятся данные классов задач и этапов, которые должен проходить каждый бизнес-процесс и заканчивая отображением задач в сводных таблицах и параметрами фильтров, по которым можно искать отдельные заявки.

*Класс задач* — это, прежде всего, совокупность *этапов* и *полей*.

*Поле* может быть видимо и/или редактируемо на одном или нескольких *этапах*. Или может быть скрытым.

*Задача* может переходить с *этапа* на *этап* по командам пользователей, которые имеют право переводить *задачу* из данного *этапа* в любой другой.

Еще с самого начала важно сказать, что *класс задач* — это общая канва, по которой идет обработка заявок (задач) определенного типа. А каждую такую индивидуальную заявку (экземпляр класса задач) будем называть *задачей* или *заявкой*.

*Классов задач* может быть создано сколько угодно. В каждом классе количество индивидуальных задач-заявок тоже не ограничено.

Классы задач группируются в *рабочие места*.

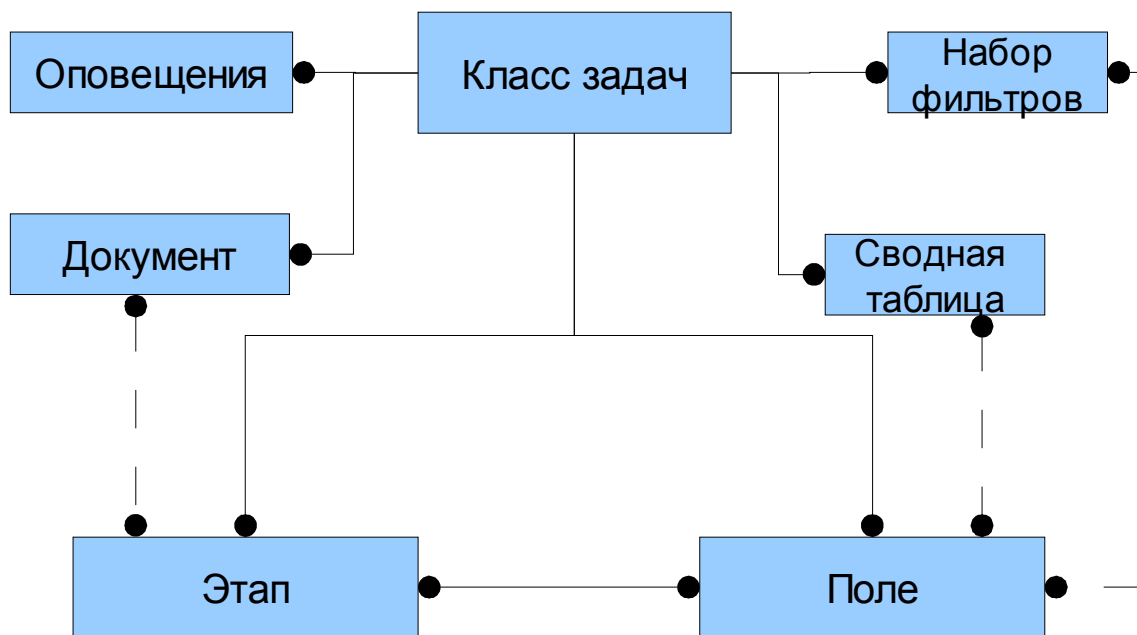
Все заявки определенного класса отображаются в *сводной таблице*. У каждого класса задач обязательно должна быть хоть одна *сводная таблица*, отмеченная как *сводная таблица по умолчанию*. В противном случае трекер ничего не покажет для данного класса задач. Сводная таблица состоит из *полей сводной таблицы*, каждое из которых может формироваться из одного или нескольких *полей класса задач*. Для класса задач может быть определено более одной сводной таблицы, например, для настройки отображения данных в *отчетах* и *сводках*.

Для каждого класса задач могут быть определены один или несколько *наборов фильтров*. Тот, что отмечен как *набор фильтров по умолчанию* — отображается над *сводной таблицей*. Если нет необходимости, *набор фильтров* можно не создавать.

Для каждого класса задач можно создать один или несколько *выходных документов* (например, заказ-наряд, накладная и пр.) и определить *этапы*, на которых данный *документ* может быть распечатан.

Для каждого класса задач и для каждого *события* из определенного перечня (см. в следующей части руководства) можно задать *шаблон оповещения (уведомления)*, которое будет отсылаться при наступлении этого события пользователям в соответствии с установленными для них *правами*.

Схематически все вышесказанное можно отобразить примерно так:





Кроме того, необходимо также сказать еще о двух понятиях: *рабочее место* и *оповещение по умолчанию*.

*Рабочее место* служит для группировки классов задач. Один класс задач может входить в одно или несколько рабочих мест.

*Оповещения по умолчанию* не привязаны к классам задач, и уведомления по этим шаблонам отсылаются в том случае, если для определенного класса задач для определенного события не задан шаблон оповещения.



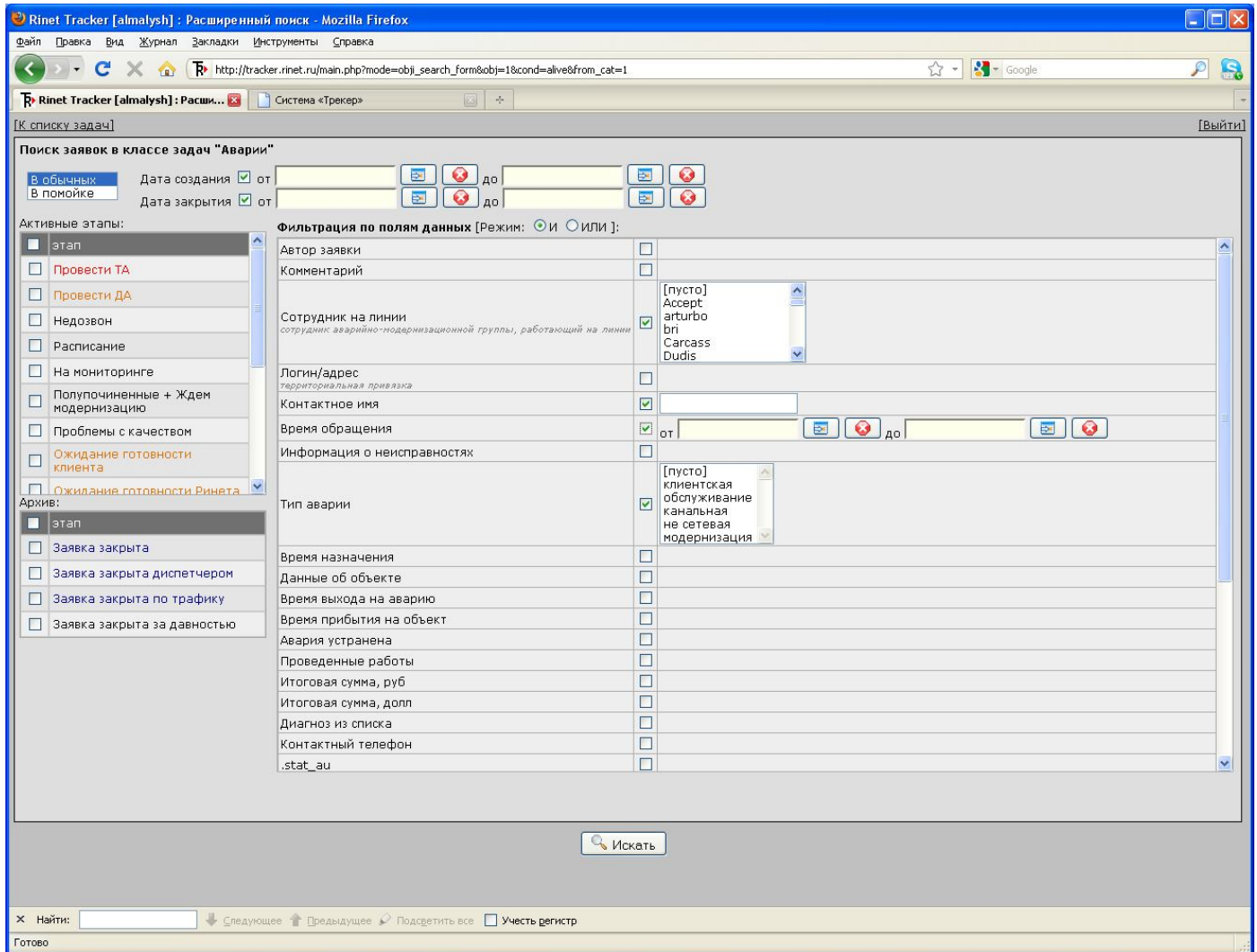
Перчислим основные области экрана:

1. Выбор рабочего места. В списке отображаются рабочие места, доступные данному пользователю. После выбора нужного рабочего места в списке классов задач (6) отобразятся классы задач, входящие в данное рабочее место.
2. Область поиска. Доступны 3 вида поиска: стандартный, мультипоиск и расширенный поиск.
  - стандартный поиск (Enter или кнопка ) ищет вхождения заданного в поле «Поиск» текста в текстовые поля только в текущем классе задач.
  - мультипоиск (кнопка ) ищет вхождения заданного в поле «Поиск» текста в текстовые поля всех классов задач, составляющих данное рабочее место; в результатах поиска найденные задачи, принадлежащие разным классам отображаются на разных закладках.
  - расширенный поиск позволяет задавать параметры поиска отдельно по всем полям задачи; после выбора ссылки «Расширенный» открывается отдельное окно описанное в разделе «Расширенный поиск».
3. Область сервисных функций.
  - Info – отображает информацию о составе и структуре данного класса задач (функция предназначена для аналитиков и разработчиков бизнес-процессов)
  - Кнопка перехода в «Корзину» - хранилище логически удаленных заявок.
  - «Сервис» - дополнительные служебные функции, среди которых есть стандартные, такие как «Печать», «Экспорт», «Выход», так и специфические для данного класса задач (в основном это различные сводки и отчеты).
4. Область фильтров. Наборы фильтров настраиваются администратором индивидуально для каждого класса задач. Для того, чтобы воспользоваться фильтром из набора нужно отметить галочками те поля, по которым необходимо провести поиск и ввести параметры поиска. По кнопке «Отфильтровать» введенные параметры поиска будут применены к списку задач.
5. Кнопка создания новой заявки данного класса.
6. Список классов задач, составляющих данное рабочее место. В первой колонке таблицы отобразится общее количество заявок в активных (т.е. не архивных) этапах данного класса задач и, в скобках, количество заявок в начальном этапе (новых заявок). По щелчку мышью на том или ином классе задач в списке на экран будут выведены этапы, фильтры и сводная таблица для выбранного класса задач.
7. Сводная таблица, в которой отображаются заявки принадлежащие данному классу задач находящиеся в выбранных этапах (8) и в соответствии с заданными условиями фильтрации. Отображаемые поля и вид сводной таблицы настраиваются для каждого класса задач индивидуально. Некоторые поля в сводной таблице могут быть отмечены небольшим треугольничком в левом верхнем углу: это признак возможности быстрого редактирования поля. Если два раза щелкнуть по такому полю мы сразу перейдем к его редактированию без необходимости входить в режим редактирования заявки.
8. Перечень этапов данного класса задач. Этапы делятся на две части — активные и архивные. Как правило, заявка попадает в архивный этап после того, как будет полностью обработана так или иначе. Можно выделить галочками набор нужных этапов, нажать на кнопку «Отфильтровать» и в сводной таблице отобразится список заявок, находящихся в выбранных этапах. Двойной щелчок мышью на названии этапа выводит в сводную таблицу заявки, находящиеся только в данном этапе (галочки на всех других этапах при этом сбрасываются). В последнем столбце таблицы отображается число заявок, находящихся на данном этапе. По столбцам таблицы можно производить сортировку. Если щелкать на названии столбца, то сортировка будет производиться только по нему, если на точке слева от названия, то эта

сортировка будет добавляться в качестве вторичной (или последующей) к уже выбранной. Изменение направления сортировки производится щелчком по стрелке в заголовке столбца сортировки.

### Расширенный поиск

Окно расширенного поиска для одного из классов задач выглядит следующим образом:



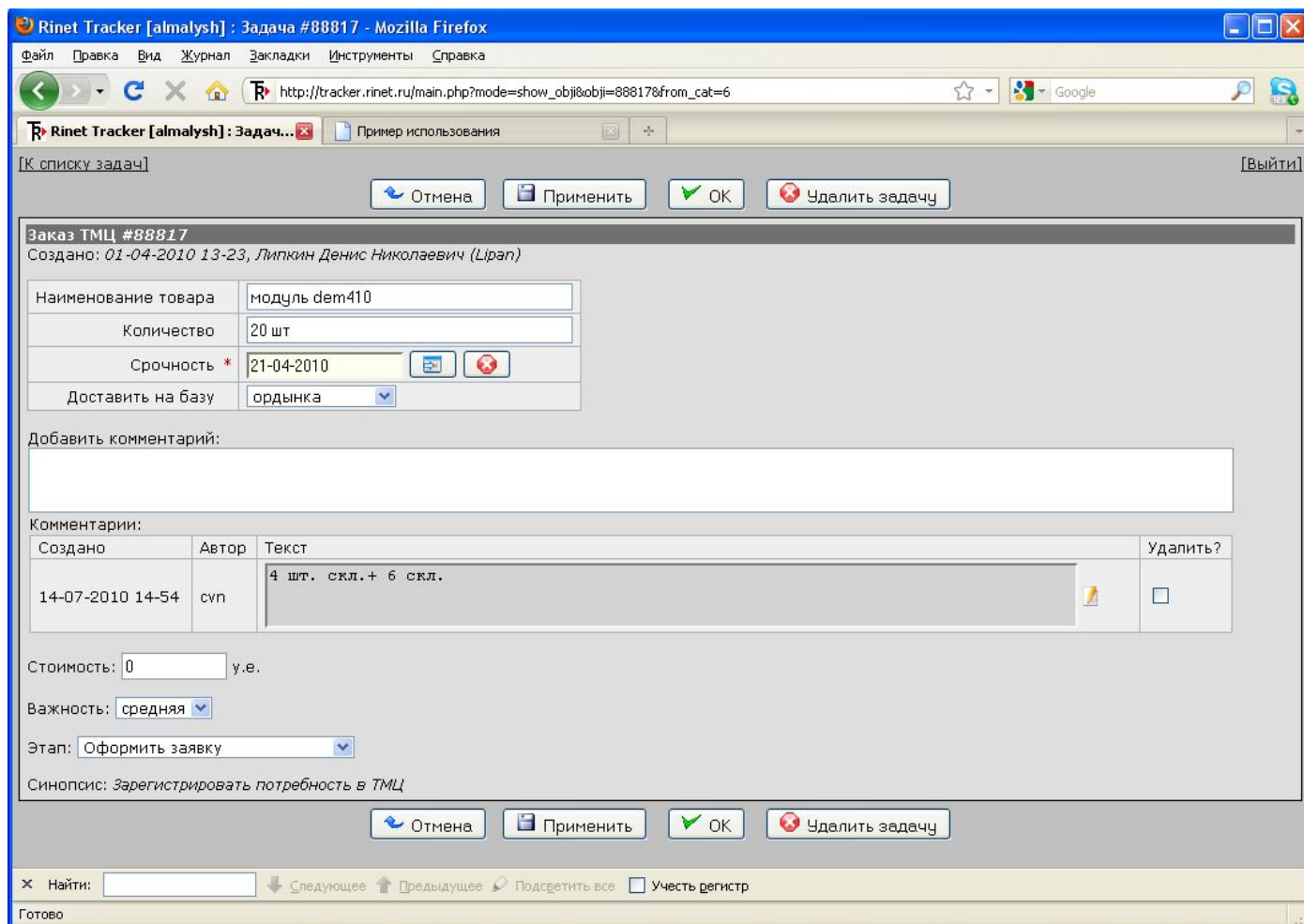
Стандартные поля поиска — это диапазон дат и времени создания и закрытия.

Слева находится область выбора этапов, в которых нужно производить поиск.

Справа перечислены все поля данного класса задач. Следует отметить галочками те поля, по которым нужно производить поиск и, в зависимости от типа поля ввести параметры поиска: выбрать нужные позиции в списке, ввести диапазон дат, образец текста или число.

## Окно редактирования заявки

Открывается, как правило, по нажатию кнопки «Edit» в первом столбце сводной таблицы.



На этом экране отображаются поля заявки, доступные для этапа, на котором находится заявка (в соответствии с настройками класса задач) и кнопки управления заявкой:

- «Отмена» - отменяет редактирование
- «Применить» - сохраняет изменения без выхода из формы редактирования
- «OK» - сохраняет изменения и возвращает к списку задач
- «Удалить задачу» - удаляет заявку

**Внимание!! При изменении этапа (поле «Этап») сохранение заявки происходит немедленно.**

Комментарии к заявке добавляются последовательно, при этом сохраняется, кто и когда добавил данный комментарий.

Можно отредактировать ранее введенный комментарий нажав в соответствующем поле комментария кнопку . Удалить ранее введенный комментарий можно отметив его галочкой в колонке «Удалить?» и сохранив заявку.

## Журнал изменений заявки

Для просмотра истории изменений заявки (кто, когда и какие поля правил) можно воспользоваться журналом изменений заявки. Как правило, этот отчет можно вызвать, щелкнув на поле с номером

заявки в первой колонке сводной таблицы.

Журнал изменений заявки выглядит следующим образом:

The screenshot shows a web browser window with the URL `http://tracker.rinet.ru/main.php?mode=obji_summary&obji=88817&from_cat=6&log`. The page title is "Заказ ТМЦ #88817".

**Создано:** 01-04-2010 13-23, Лирап (Липкин Денис Николаевич)

**Комментарий (посл. измен. 08-09-2010 15-47):**  
 14-07-2010 14-54, свп: 4 шт. скл.+ 6 скл.

**Стоимость:** 0 у.е.  
**Этап:** Оформить заявку

**Поля:**

Наименование товара	модуль dem410
Количество	20 шт
Срочность	21-04-2010
Доставить на базу	ордынка
Форма оплаты	[пусто]
Причина отклонения	[пусто]
Ответственный за прием	[пусто]
Через 1С Склад принял	[пусто]

**Лог редактирования:**

когда: 08-09-2010 15-47 кто: свп		
что изменено	прежнее значение	новое значение
Комментарий	14-07-2010 14-54, свп: 4 шт. скл.	14-07-2010 14-54, свп: 4 шт. скл.+ 6 скл.

когда: 14-07-2010 14-54 кто: свп		
что изменено	прежнее значение	новое значение
Комментарий	[пусто]	14-07-2010 14-54, свп: 4 шт. скл.

когда: 01-04-2010 13-39 кто: свп		
что изменено	прежнее значение	новое значение
Срочность	15-04-2010	21-04-2010

Готово

Отображаются как текущие значения полей заявки, так и информация о том, кто, когда и какое поле менял.

## Пример использования

### **Описание работы системы на примере учета багов и модификаций программы**

Проиллюстрируем принцип работы системы "Трекер" на примере того, как мы используем его для учёта процесса сопровождения и развития наших систем.

Напоминаем, что в абстрактном виде Трекер оперирует понятиями:

- заявка - учитываемая проблема
- поля заявки - данные пользовательского типа, описывающие проблему
- этап - особое поле заявки, характеризующее положение проблемы с точки зрения бизнес-процесса
- класс - настраиваемый шаблон, из которого создаются отдельные экземпляры - заявки; с точки зрения интерфейса класс - это метод группировки проблем одного типа
- права - пользователи системы наделены разными правами; в зависимости от класса и этапа пользователь может иметь право на просмотр, редактирование

Если вдуматься, то, оперируя этими понятиями, даже в чистом виде можно вести учёт любых процессов предприятия.

В простейшем случае Трекер мог бы состоять из одного класса – Предприятие.

В абстрактной форме процесс решения любой проблемы может быть разбит на следующие этапы:

- Регистрация проблемы
- Анализ проблемы
- Проблема находится в стадии решения
- Проблема решена
- Решение проблемы затормозилось

Трекер автоматически ведёт учёт времени, когда что-то изменилось в заявке, и автора изменений, что позволяет постфактум разобраться в произошедшем, анализируя лог.

В простейшем случае любой сотрудник имеет право на все действия в системе.

Сама же проблема в принципе может быть описана одним большим текстовым полем.

Настроив систему таким образом, мы получаем возможность вести учёт произвольного процесса, и очень во многих случаях такой примитивный подход позволяет принести пользу и решить многие проблемы. К такому подходу мы иногда прибегаем, когда проблемы подразделения носят очень разнообразный характер, и декомпозиция всех бизнес-процессов представляется слишком трудоёмкой. Или как вариант – если мы не знаем, какие именно задачи возникнут перед подразделением. Например, классический случай такой ситуации – работа менеджера по управлению проектами. Тем не менее, несмотря на соблазнительность такого подхода (несомненный плюс – простота развертывания), на практике, будучи реализованным, он приводит к мешанине задач, непониманию исполнителем, что именно он должен делать, и невозможности провести анализ работы подразделения.

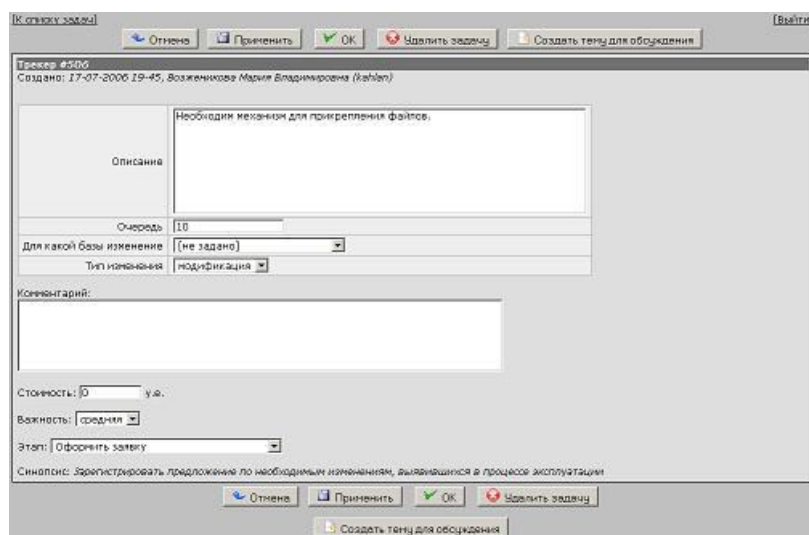
Чтобы Трекер принес вашему предприятию пользу, необходимо реализовать следующий подход:

1. Нужно уметь выделять типизированные задачи
2. Нужно в этих задачах выделить формальные поля, точно характеризующие проблему
3. Нужно уметь выделить в бизнес-процессе этапы, то есть ситуации, когда проблема меняет ответственного или состояние проблемы качественно меняется.

Практика показывает, что Трекер начинает приносить реальную пользу только тогда, когда

исполнитель видит в сводной таблице не больше тридцати задач. Если в сводной таблице задач больше, то большинство исполнителей теряется, или не делает вообще ничего, или начинает заниматься не совсем тем, чем нужно. Фактически, если вы настроили Трекер так, что в сводной таблице у исполнителя оказывается больше тридцати задач, значит вы его настроили неправильно, и необходимо проводить декомпозицию, вычленять этапы, делить ответственных, делить бизнес-процесс на разные потоки и фиксировать изменения в Трекере. Не исключено, что на предприятии неправильно поставлен сам бизнес-процесс, Трекер – неплохой инструмент, помогающий осознать точки перегрузки или недогрузки :)

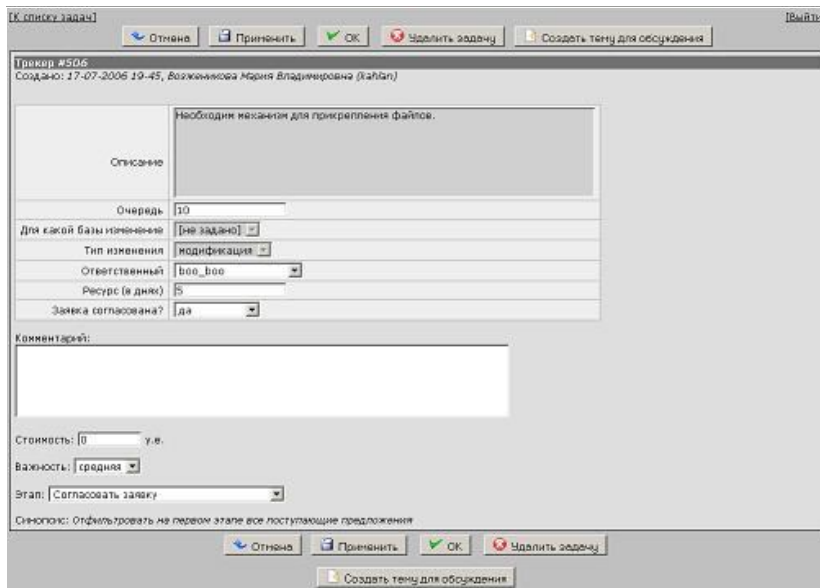
Девелоперский коллектив у нас разбит на несколько рабочих групп. Каждая группа занимается своим проектом. Работа над Трекером ведётся группой из трёх человек: два программиста и один аналитик. При этом у программистов есть свои профили – типы задач, которыми они занимаются. Сам рабочий процесс построен следующим образом. Пользователь системы, у которого есть какое-либо пожелание по функционалу, либо он хочет зафиксировать баг создает новую заявку. Права на создание новых заявок есть у всех сотрудников Ринета (реально этой возможностью пользуются руководители и ответственные сотрудники подразделений, которые естественным образом аккумулируют пользовательские проблемы). При создании заявки заполняются поля: описание, класс задач, тип изменения, комментарий, важность.



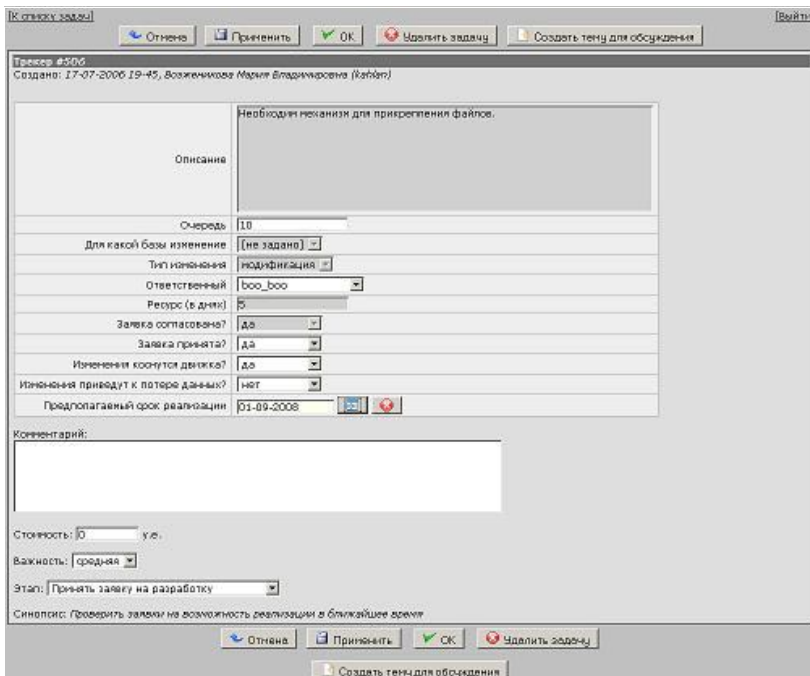
The screenshot shows a web-based form for creating a task in the 'Трекер' system. The form is titled 'Трекер #506' and includes the following fields and controls:

- Buttons:** Отмена, Применить, OK, Удалить задачу, Создать тему для обсуждения.
- Task Info:** Трекер #506, Создано: 17-07-2006 19:45, Возженкова Мария Владимировна (kashap).
- Description:** A text area containing the text 'Необходим механизм для прикрепления файлов.' (A mechanism is needed for attaching files).
- Priority:** Очередь: 10.
- Database:** Для какой базы изменение: (не задано).
- Type:** Тип изменения: модификация.
- Comments:** Комментарий: (empty text area).
- Cost:** Стоимость: 0 у.е.
- Importance:** Важность: средняя.
- Stage:** Этап: Оформить заявку.
- Footer:** Синоним: зарегистрировать предложение по необходимым изменениям, выявившимся в процессе эксплуатации.
- Bottom Buttons:** Отмена, Применить, OK, Удалить задачу, Создать тему для обсуждения.

Созданная заявка рассматривается аналитиком. При этом она уточняется и обычно переформулируется. Часть заявок на этом этапе отклоняется ввиду технической неисполнимости. Часть заявок закрывается ввиду того, что реализация либо запланирована, либо проходит в данный момент. Кроме этого аналитик согласовывает очередность и важность этой заявки у руководства компании. По результатам рассмотрения аналитик заполняет поля: ответственный (программист), очередь разработки, подтверждает согласованность заявки. По окончании работы аналитик переводит заявку в этап «Принять заявку на разработку».



Этап «Принять заявку на разработку» накопительный, так как скорость поступления заявок выше скорости разработки. Программист на основании очередности заявки знакомится с ней. Если у него возникают вопросы по формулировке, то он уточняет их у аналитика. На этом этапе определяется предположительная трудоёмкость разработки (в часах). Если в процессе реализации оказывается, что трудоёмкость превышает первоначальные оценки, то производится пересогласование у руководства компании. Кроме трудоёмкости, оцениваются другие параметры сложности (нужно ли менять движок, могут ли потеряться данные). Для того, чтобы учитывать эти особенности в форме на этом этапе заполняются соответствующие поля.



По мере того, как программист решает свои задачи, он переводит заявки из этапа «Принять заявку на разработку» в этап «Провести реализацию». Для того чтобы избежать мешанины и правильно оценивать загрузку программистов, этот этап у нас разбит на четыре этапа:

- «Провести реализацию: Слава короткие»
- «Провести реализацию: Слава длинные»
- «Провести реализацию: Костя короткие»

□ «Провести реализацию: Костя длинные»

Как видно, задачи у каждого программиста разбита на два потока: длинные задачи и короткие задачи. Рабочее время программистов разделено на время, когда они занимаются длинными задачами, и на время, когда они занимаются короткими задачами. Это сделано для того, чтобы очереди не блокировались. На данном этапе программист заполняет поля: План работ, предполагаемый срок реализации. По окончании реализации программист подтверждает её окончание.

The screenshot shows a web-based task management interface. At the top, there are navigation buttons: 'Отмена', 'Принять', 'OK', 'Удалить задачу', and 'Создать тему для обсуждения'. The main content area is titled 'Трекер #506' and 'Создано: 17-07-2006 19:45, Воржеников Андрей Владимирович (kafan)'. Below this is a large text area for 'Описание' with the text 'Несоблюдение механизма для прикрепления файлов.' To the right of the description is a 'План' section with a large empty text area. Below the plan are several dropdown menus for status tracking: 'Очередь' (value: IO), 'Для каждой базы изменения' (value: [не задано]), 'Тип изменения' (value: модификация), 'Ответственный' (value: bob\_bob), 'Ресурс (в днях)' (value: 5), 'Заявка согласована?' (value: да), 'Заявка принята?' (value: да), 'Изменения коснутся данных?' (value: да), and 'Изменения приведут к потере данных?' (value: нет). There is also a date field for 'Предполагаемый срок реализации' set to '01-09-2008'. At the bottom of the form are more status dropdowns: 'Разработка произведена?' (value: [не задано]), 'Проверено?' (value: [не задано]), and 'Работает корректно?' (value: [не задано]). A 'Комментарий:' text area is located below these. At the very bottom, there are input fields for 'Стоимость: 0 у.е.', 'Важность: средняя', and 'Этап: Провести реализацию: Слева короткая'. A status bar at the bottom indicates 'Синхронизация: Реализовать предлагаемые изменения' and includes the same navigation buttons as the top.

На следующем этапе программист заливает произведённые изменения на тестовую платформу для тестирования. Аналитик тестирует изменения, проверяет полноту реализованного функционала и пытается убедиться в том, что в систему не внесены новые баги. Если аналитик считает, что разработка по заявке произведена успешно, он заполняет поля «Проверено?» - «да», а также переводит в этап «Накатить задачу».

Копировать задачу

Отмена Применить ОК Удалить задачу Создать тему для обсуждения Выйти

Трекер #506  
Создано: 17-07-2006 19:45, Возжеников Марья Владимировна (kafim)

Описание: Необходим механизм для прирезгления файлов.

Очередь: 10

Для какой базы изменение: [не задано]

Тип изменения: модификация

Ответственный: boo\_boo

Ресурс (в днях): 5

Заявка рассмотрена?: да

Заявка принята?: да

Изменения коснутся дтшки?: да

Изменения приведут к потере данных?: нет

Предполагаемый срок реализации: 01-09-2006

План:

Разработка произведена?: да

Проверено?: [не задано]

Работает корректно?: [не задано]

Комментарий:

Стоимость: 0 у.е.

Важность: средняя

Этап: Проверить на тестовом сервере

Отмена Применить ОК Удалить задачу Создать тему для обсуждения

На данном этапе программист заливает произведённые изменения на основную платформу и переводит в этап «Проверить на основном сервере».

Копировать задачу

Отмена Применить ОК Удалить задачу Создать тему для обсуждения Выйти

Трекер #506  
Создано: 17-07-2006 19:45, Возжеников Марья Владимировна (kafim)

Описание: Необходим механизм для прирезгления файлов.

Очередь: 10

Для какой базы изменение: [не задано]

Тип изменения: модификация

Ответственный: boo\_boo

Ресурс (в днях): 5

Изменения коснутся дтшки?: да

Изменения приведут к потере данных?: нет

План:

Файлы обновления:

Проверено?: да

Задача накатана?: [не задано]

Работает корректно?: [не задано]

Комментарий:

Стоимость: 0 у.е.

Важность: средняя

Этап: Накатить задачу

Отмена Применить ОК Удалить задачу Создать тему для обсуждения

На этом этапе аналитик убеждается, что все нормально, и пытается показать результат пользователю, который создавал заявку.

К окошку задачи: [Выйти]

Отмена Принять ОК Удалить задачу Создать тему для обсуждения

Трекер #506  
Создано: 17-07-2006 10:45, Возжаникова Мария Владимировна (kafiro)

Описание: Необходим механизм для приеропления файлов.

Очередь: 10

Для какой базы изменения: [не задано]

Тип изменения: модификация

Ответственный: booo\_booo

Ресурс (в днях): 5

Заявка согласована?: да

Заявка принята?: да

Изменения коснутся дивизга?: да

Изменения приведут к потере данных?: нет

Предполагаемый срок реализации: 01-09-2008

План:

Разработка произведена?: да

Задача накатана?: да

Работает корректно?: [не задано]

Комментарий:

Стоимость: 0 у.е.

Важность: средняя

Этап: Проверить на основном сервере

Отмена Принять ОК Удалить задачу Создать тему для обсуждения

Если все нормально, то заявка переводится в архивное состояние (этап «Заявка закрыта»). Если есть проблемы, то заявка возвращается на доработку.

К окошку задачи: [Выйти]

Отмена Принять ОК Удалить задачу Создать тему для обсуждения

Трекер #506  
Создано: 17-07-2006 10:45, Возжаникова Мария Владимировна (kafiro)  
Закрыто: 2008-07-08 12:30:55

Описание: Необходим механизм для приеропления файлов.

Очередь: 10

Для какой базы изменения: [не задано]

Тип изменения: модификация

Ответственный: booo\_booo

Ресурс (в днях): 5

Заявка согласована?: да

Заявка принята?: да

Изменения коснутся дивизга?: да

Изменения приведут к потере данных?: нет

Предполагаемый срок реализации: 01-09-2008

План:

Разработка произведена?: да

Файлы обсуждения:

Проверено?: да

Задача накатана?: да

Работает корректно?: да

Комментарий:

Стоимость: 0 у.е.

Важность: средняя

Этап: Заявка закрыта

Синопис: Заведшить работу с заявкой (может быть проведена после любого этапа, описанного выше)

Отмена Принять ОК Удалить задачу Создать тему для обсуждения

С помощью механизма расширенного поиска руководство компании оценивает количество решённых

заявок за период. Кроме того, дополнительным механизмом контроля является просмотр количества заявок по этапам. Если в каком-то этапе заявок слишком много или слишком мало, то можно предположить наличие организационного сбоя.