

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Севастопольский государственный университет»**

**Институт информационных технологий**

**ИСПОЛЬЗОВАНИЕ МЕНЕДЖЕРА ЗАДАЧ В ПРОЦЕССЕ  
СОВМЕСТНОЙ РАБОТЫ НАД ПРОЕКТОМ**

**Методические указания  
к лабораторной работе №5  
по дисциплине  
“Управление IT-проектами”  
для студентов специальности 09.03.02 –  
"Информационные системы и технологии"  
всех форм обучения**

**Севастополь  
2024**

**1. Цель работы:** изучить особенности использования инструментов для планирования и управления задачами в процессе совместной работы над проектами.

## **2. ОСНОВНЫЕ ПОЛОЖЕНИЯ**

Системы планирования, постановки и контроля задач – основной инструмент работы над проектами в крупных компаниях и небольших стартапах.

В настоящее время существует множество веб-приложений, предназначенных для управления проектами: Pivotal tracker, Redmine, Jira, Asana и т.д. Ниже представлено краткое описание некоторых из них.

Redmine – открытое серверное веб-приложение для управления проектами и отслеживания ошибок. Redmine написан на Ruby и представляет собой приложение на основе широко известного веб-фреймворка Ruby on Rails.

Atlassian JIRA – программный продукт австралийской компании Atlassian для управления жизненным циклом любого проекта или рабочего процесса. Это система для коллективной работы с задачами в рамках бизнес-процесса или проекта. Система позволяет работать с несколькими проектами, разбивать их на этапы, настраивать любые типы задач, связывать задачи между собой, назначать ответственных по различным направлениям, настраивать роли участников проектов, легко формировать отчеты, и многое другое.

Asana – простой и удобный сервис для управления задачами и проектами. Интеграция с Email. Мобильные версии. Бесплатен для команд до 30 человек.

Worksection – русскоязычная онлайн-система для управления проектами. Содержит dashboard, задачи с комментариями, календарь, хранилище файлов, систему учета времени, тэги. Поддержка SSL, FTP пользователя, субдоменов пользователя. Отличается простотой и доступностью.

Easy Projects .NET – это веб-приложение для управления проектами по разработке программного обеспечения, написанное на .NET компанией Logic Software.

### **2.1. Краткое описание Pivotal tracker**

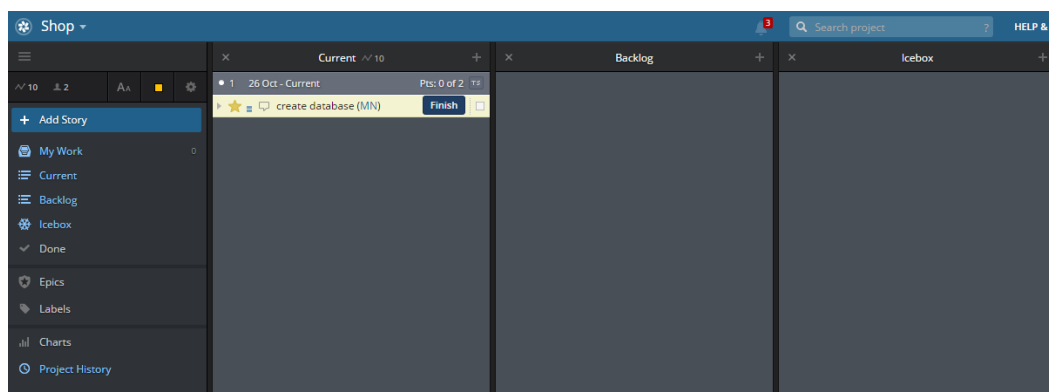
Pivotal предусмотрен для проектов, разрабатываемых по «гибким» методологиям. Написан на Ruby on Rails.

Практически вся работа над проектом сводится к работе с главной страницей проекта.

На главной странице содержатся stories, или по-другому тикеты.

*Пример story: 'A user should be able to add a product to their shopping cart'.*

Интерфейс в Pivotal представляет собой панели: backlog, icebox, current и done. На этой же странице можно показать колонки с Releases, My Work, Charts, Project History, Charts, Labels (tags or components).



Все истории сразу после создания находятся в «замороженном состоянии» (Icebox), и пребывают там до тех пор, пока их не начнут исполнять, или они будут перемещены в список активных задач (current, backlog).

**Icebox** – панель, на которую помещается тикет после создания, и хранится там, пока исполнитель не начинает над ним работать, перемещая его в Current.

Панель icebox предназначена для хранения идей и тех тикетов, которые не входят ни в одну версию.

**Current** – это список задач текущей итерации (длина итерации выбирается в неделях). Если тикеты не умещаются в текущей итерации, то они автоматически попадают в Backlog.

Панель backlog – лента утвержденных тикетов. Если количества Velocity (по умолчанию 10 на одну неделю) не хватает на все тикеты текущей итерации, то оставшиеся тикеты помещаются на эту панель.

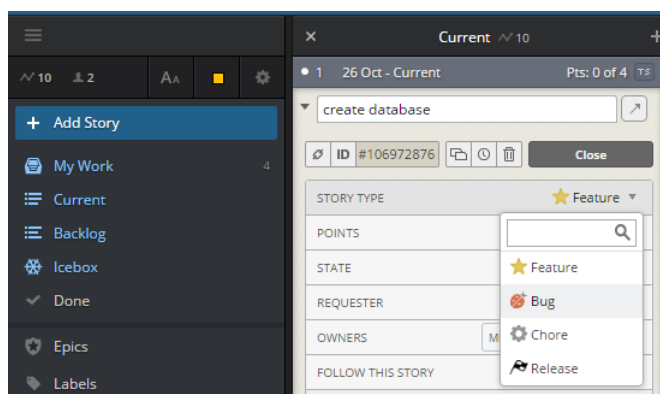
**Done** – это панель, на которую помещаются тикеты, закрытые после завершения текущей итерации.

Тикеты можно перемещать между панелями (кроме Done).

Нельзя выполняемый тикет положить в Icebox (или положить в Backlog, то, что выполнит команда в текущей итерации(Current)).

Продукт позволяет, в соответствии с методикой scrum, собирать, приоритезировать и отслеживать выполнение конкретных задач (story) – на разработку (feature), исправление

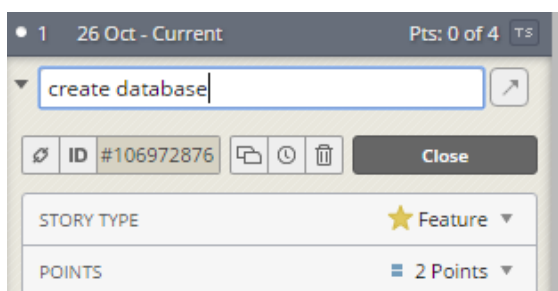
ошибок (bug), вспомогательные работы (chore). Дополнительно и явно отслеживаются планируемые релизы продукта (release).



Поле для интерпретации понятия "feature" довольно широкое, но согласно документации к продукту и рекомендациям методики – это "улучшения или новая функциональность, несущая непосредственную ценность для заказчика". feature" – единица самостоятельной функциональности проекта, которую пользователь может отдельно протестировать и использовать ("deliverable").

Планирование работы команды проводится на заранее определенном временном отрезке – этапе ("sprint"). Длительность этапа, равно как и выбирается таким образом, что любая отдельно взятая функциональность ("feature") может быть реализована (значит – продумана, запрограммирована, оттестирована и принята заказчиком) в течение этого этапа. Например, двухнедельный спринт при времени на программирование одной истории – в среднем 2 дня.

Относительная сложность реализации каждой "feature" заранее оценивается в условных целочисленных баллах ("points") по фиксированной, заранее определенной шкале.



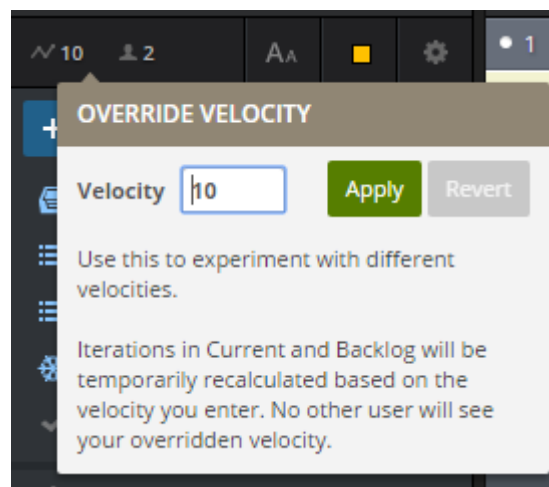
**Point** – это относительная мера «усилий» необходимых для закрытия тикета. Данное понятие можно воспринимать как сложность тикета или объема работ.

Оценке подлежат только истории, "**полезные**" для заказчика ("features") – ни исправления ошибок, ни вспомогательные работы оценке не подлежат.

Исходя из проведенной оценки историй, выбранной длительности этапа, и реальной статистики обработки списка историй, система вырабатывает значение важной проектной метрики – "скорость команды", количество реализованных баллов за этап [points/sprint].

**Velocity** – это среднее число point приходящееся на итерацию, оно рассчитывается по недавно завершенным итерациям. В настройках проекта можно выставить по скольким итерациям будет высчитываться Velocity.

Шкала выбирается один раз на проект, и не меняется. Таким образом, используя метрику скорости можно осуществлять оперативное планирование и корректировку усилий в проекте.



Допустим, в недельной итерации Velocity равно 10. В этом случае сумма поинтов у тикетов в текущей итерации не должна превышать 10, иначе они начнут падать в бэклог. Если же Velocity исчерпано и закрыты все тикеты, но при этом есть возможность на данной итерации сделать еще несколько тикетов, то их можно просто вернуть из бэклога, нажав на одном из тикетов старт, и он автоматически падает в Current, если даже на него не хватает Velocity.

Тикеты могут выступать в роли Feature, Bug, Chore, Release.

*Feature* – это разработка.

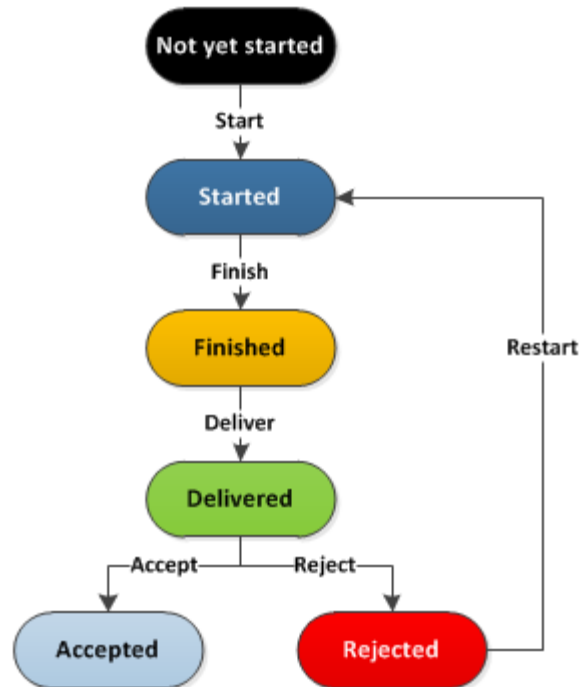
*Bug* – это ошибки.

*Chore* – это вспомогательные работы (необходимо реализовать функциональность, которая необходима для разработчика, но не просматривается заказчиком: рефакторинг, тестирование или что-либо еще).

*Release* – это просто тикет-флажок, который служит для отметки места, на котором необходимо сделать релиз. Помимо общих полей имеет поле Deadline, с датой сдачи релиза.

У Feature и Bug есть несколько состояний: Not yet started, Started, Finished, Delivered, Accepted, Rejected (доставлена, принята, отвергнута).

У Release есть только Not yet started, Accepted, а у Chore есть Not yet started, Started, Finished.



На строке каждой задачи появляется кнопка (или кнопки) соответствующая текущему состоянию задачи.

У всех этих типов могут быть следующие поля:

- Labels (tags)
- Owner
- Description
- Attach file
- Comments

Backlog

create database

ID

cancel Save

STORY TYPE ★ Feature

POINTS ⌚ Unestimated

REQUESTER TA Tanya

OWNERS <none> +

FOLLOW THIS STORY (1 follower) ☒

DESCRIPTION (edit)

LABELS

В каждом таске можно вести обсуждение с упоминанием других членов команды, можно добавлять любые файлы (чаще всего это скриншоты, например к UI-багу),

ACTIVITY

TA Tanya @tanya7tor

Add a comment or paste an image

cancel Add

можно вести сабтаски (удобно при доработках).

TASKS

☐ Add a task Add

Для предоставления доступа заказчику в Pivotal есть возможность пригласить “наблюдателя” с правами “read only”.

OVERVIEW SETTINGS MEMBERS REPORTS EXPORT IMPORT PROJECT PROFILE INTEGRATIONS

Access

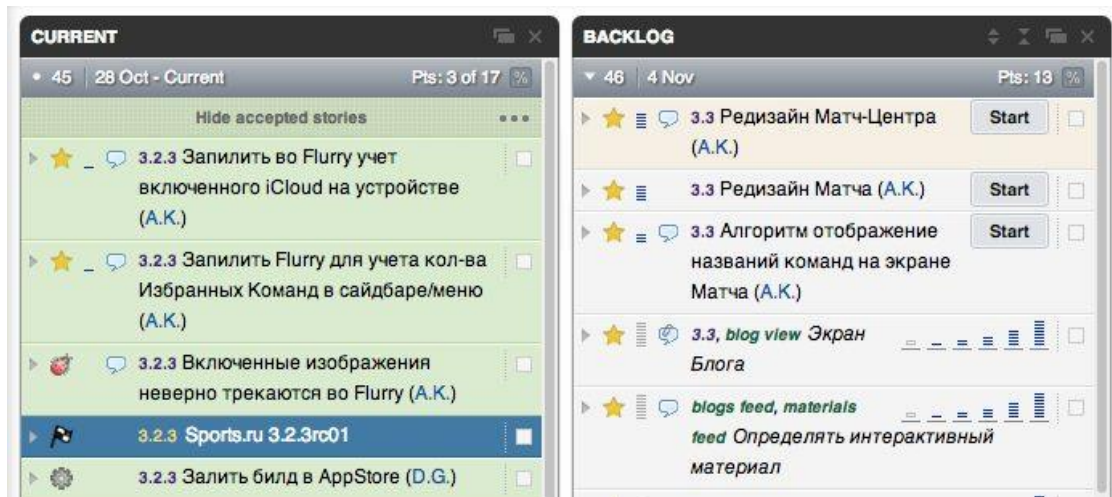
Project ID 1470296

Allow API Access ☒

Require HTTPS for API Access ☐ Prevent non-HTTPS access to this project via API versions older than V5

Enable RSS/Atom ☐ People can subscribe to the activity feed for this project

Public Access ☒ Allow anyone to view this project at <https://www.pivotaltracker.com/projects/1470296>. More about public projects here.

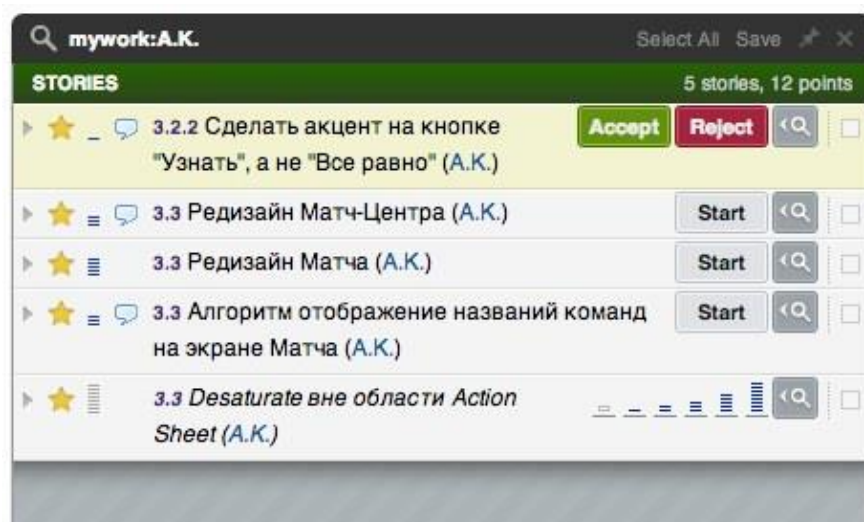


**Пример.** Менеджер имеет на руках ТЗ. Начинается процесс создания задач (тикетов). Все задачи попадают в Icebox.

Далее задачи перемещаются в Backlog. С этого момента они утверждены, у каждого задачи должен быть установлен исполнитель. Если на проекте один разработчик, то это не требуется. Если разработчиков несколько, то распределение задач осуществляет менеджер проекта.

У каждой задачи есть уровень сложности – это некое абстрактное значение в пунктах. После того, как разработчик приступает к задаче, он нажимает Start. По завершению задачи разработчик нажимает Finish. Следующий шаг – это Delivered (разработчик доставляет задачу в сборку).

Делая сборку, разработчик отмечает те задачи, которые попадут в сборку и можно принимать менеджеру проекта. Менеджер может принять задачу, если считает, что задача выполнена, а может сделать реджект Rejected (кнопка – Restart), указав причину.



После сборки менеджер может создать задачи, связанные с багами.

Разработчики любят указывать в коммите в GitHub ID задачи, тогда при пуше задача автоматически завершается.



Таким образом, в любой момент можно отследить чем занимается конкретный разработчик, какие задачи уже реализованы и попадут в сборку, какие еще в icebox и т.п.

### 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

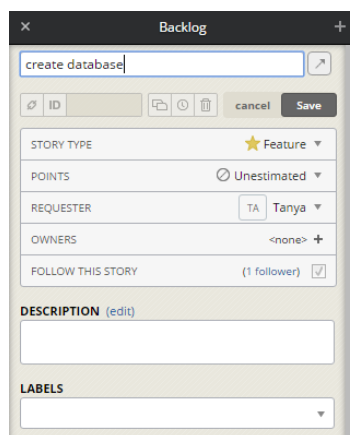
#### Ход работы:

1. Выберите проект согласно варианту (см. Приложение А).
2. Разбейте проект на задачи (при необходимости – на подзадачи).
3. Укажите исполнителей для каждой задачи (подзадачи)
4. Расставьте приоритеты для каждой задачи (подзадачи).
5. Составьте календарный план проекта.

Используя менеджер задач, создайте новый проект, над которым будет работать ваша команда.

1. Добавьте участников проекта, разослав им приглашения на email для участия в работе.
2. В основном поле программы **совместно** добавьте задачи (подзадачи), которые вам предстоит решить.
3. Распределите задачи между участниками проекта в соответствии с порядком выполнения.

Обратите внимание, что каждая задача имеет целый ряд настраиваемых свойств и опций, которые вы должны изменить в дополнительной правой панели: закрепить задачу за определенным членом команды, назначить срок выполнения, добавить описание, присвоить теги и т.д..



Каждую задачу можно разбить на отдельные подзадачи, что позволяет еще более гибко осуществлять процесс планирования.

Для коммуникации между сотрудниками можно оставлять комментарии к любой задаче и, разумеется, отвечать на них.

4. Сделайте проект публичным.
5. Ссылку на проект поместите в отчет по лабораторной работе.

### **3. СОДЕРЖАНИЕ ОТЧЁТА**

Отчёт должен содержать в себе:

1. Титульный лист.
2. Краткое описание работы с выбранным сервис-менеджером.
3. Практическая работа над выбранным ранее проектом с использованием менеджера проекта.
4. Календарный план.
3. Вывод о проделанной работе (полученные навыки, замеченные трудности или, наоборот, улучшения в работе).

### **4. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Обоснуйте необходимость использования веб-приложений для управления проектами.
2. Перечислите существующие веб-приложения для управления проектами.
3. Опишите процесс создания проекта с использованием Pivotal tracker.
4. Дайте определение понятия «Point» применительно к управлению IT-проектами.
5. Дайте определение понятия «Velocity» применительно к управлению IT-проектами.
6. В каких состояниях могут находиться задачи проекта?
7. В каких случаях в проекте используются подзадачи?
8. Какие панели включает в себя интерфейс Pivotal tracker?
9. Какое приложение было выбрано Вами для реализации проекта? Выбор поясните.
10. Опишите порядок работы в выбранном Вами приложении.

### **5. РЕКОМЕНДУЕМЫЕ ИНТЕРНЕТ-ИСТОЧНИКИ**

1. Pivotal Tracker help: <https://www.pivotaltracker.com/help>
2. Pivotal Tracker как инструмент в Waterfall-разработке: <http://habrahabr.ru/company/cleverpumpkin/blog/198930/>
3. Pivotal Tracker: <http://megamozg.ru/post/1166/>
4. Какая система управления задачами лучше – выбор IT-менеджеров и предпринимателей: <https://vc.ru/p/task-management>

5. Управление проектами: Жизненный цикл и фаза исследования  
<http://megamozg.ru/company/friifond/blog/19998/>
6. Asana в деталях, и как ее использовать: <http://megamozg.ru/post/1700/>

## ПРИЛОЖЕНИЕ А

### Перечень вариантов

Вариант	Описание
1	Интернет-магазин бытовой техники
2	Интернет-магазин автозапчастей
3	Интернет-магазин одежды
4	Интернет-магазин мобильных телефонов
5	Интернет-магазин мебели
6	Интернет-магазин детских товаров
7	Интернет-магазин зоотоваров