и выносятся на зонированную общую доску. Но как быть, если члены команды работают удаленно, да еще и в разных часовых поясах? А как управлять проектом и задачами, если команда состоит из более чем ста человек? Здесь на помощь придет решение в виде веб-приложения по управлению задачами проекта.

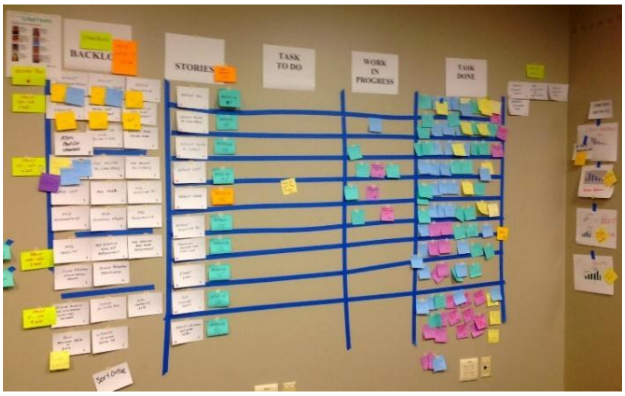

Рис. 1 – Пример Agile доски с задачами

Применение данного приложения поможет руководителям организовать работу в командах, создавать свой рабочий процесс для каждого вида задач, а интуитивно понятный интерфейс позволяет представить статистику и текущие статусы выполнения задач в удобном виде. Система обеспечивает доступ в любое время и в любом месте. Благодаря выбранному современному стеку технологий данные пользователей надежно защищены.

В основе принципа работы лежат задачи (Task, Ticket) – атомарные единицы работы над проектом, которые по мере своего выполнения изменяют свой текущий статус, двигаясь по рабочему процессу. Основной функционал включает в себя:

− создание задач и подзадач;
− организацию своего рабочего процесса для каждого вида задач;
− возможность учета запланированного и фактически затраченного времени;
− комментирование задач с целью лучшей коммуникации между исполнителями задач.

Достоинствами предоставляемого сервиса являются:

− отказ от представления задач и подзадач на старомодных «физических» досках, перенос данных на сервер, доступный круглосуточно;
− удобный и понятный интерфейс;
− возможность использования любыми рабочими группами в любых проектах;
− учет затраченного времени над задачами каждого сотрудника;
− наглядность текущего статуса исполнения всех задач на общей доске задач;
− безопасность данных пользователей;
− использование современных технологий в разработке.

Итогом работы является полноценное веб-приложение, которое поможет оптимизировать контроль выполнения задач, улучшить коммуникацию между сотрудниками, а значит, и увеличить скорость и качество создания программных продуктов.

Список использованных источников:
1. Гибкая методология разработки. [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Гибкая_методология_разработки. – Дата доступа : 08.04.2018.

БАЗИСЫ ГРЁБНЕРА. АЛГОРИТМ БУХБЕРГЕРА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

ак Д.С.

Калугина М.А.–канд. физ.-мат. наук, доцент

В течение всей истории математики стоял вопрос о решении систем алгебраических уравнений с большими целочисленными коэффициентами. Австрийский математик Вольфганг Грёбнер еще в 1930-х годах разрабатывал теорию стандартных базисов, которая позволяла разрешать различные САУ, однако его работа не нашла практического применения из-за больших практических подсчетов, которые занимали много времени, а иногда были вовсе неразрешимы.

В начале 60-х годов прошлого века был достигнут прогресс в области систем алгебраических уравнений. Австрийский математик Бруно Бухбергер, ученик Грёбнера ввел такие понятие как базис Грёбнера, а также разработал эффективный алгоритм, позволяющий находить базис Грёбнера для систем алгебраических уравнений с большими коэффициентами, и, как следствие, находить корни этих уравнений.

Пусть дана нелинейная САУ. Для решения системы нам необходимо перейти к эквивалентной системе. Введем понятие идеал системы:

Пусть $\mathbb{C}\{x_1 \dots, x_n\}$- множество всех многочленов от переменных $x_1 \dots, x_n$ с комплексными коэффициентами: $\begin{Bmatrix} f_1(x_1 \dots, x_n) = 0, \\ \dots \dots \dots \dots \dots \dots \dots \\ f_n(x_1 \dots., x_n) \end{Bmatrix}$

Определение: Идеал, порожденный данным набором многочленов $\{f_1 \dots f_n\}$, -это множество всех комбинаций вида $\{f_1 g_1 \dots f_n g_n\}$ где g_i- произвольные многочлены. Идеал $I = \langle f_1 \dots, f_n \rangle$.

Также важно понимать если $(I = \langle f_1 \dots, f_n \rangle) = (I = \langle g_1 \dots, g_n \rangle)$ идеалы двух систем совпадают, тогда системы многочленов от $(x_1 \dots, x_n)$:

$$\begin{Bmatrix} f_1(x_1 \dots, x_n) = 0, \\ \dots \dots \dots \dots \dots \dots \dots \\ f_n(x_1 \dots., x_n) = 0 \end{Bmatrix}$$

$$\begin{Bmatrix} g_1(x_1 \dots, x_n) = 0, \\ \dots \dots \dots \dots \dots \dots \dots \\ g_n(x_1 \dots., x_n) = 0 \end{Bmatrix}$$

эквивалентны. Т.е. имеют одно и то же множество решений (обратное утверждение неверно).

Вывод: Множество решений системы однозначно определяется идеалом системы. Различные базисы одного идеала отвечают эквивалентным системам, следовательно, каждая САУ эквивалентна конечной системе.

Для перехода к другой эквивалентной системе нам необходимы лишь идеалы САУ, для этого нам нужно выбрать другой базис, например, базис Гребнера.

Определение: пусть $\{f_1 \dots f_n\}$, – набор многочленов из $\mathbb{C}\{x_1 \dots, x_n\}$. Тогда они называются базисом Гребнера $I = \langle f_1 \dots, f_n \rangle$.если для любого многочлена $f \in I$ его старший член делится на старший член хотя бы одного из многочленов $f_1 \dots f_n$.

Определение: Многочлены f_i,f_jимеют зацепление, если их старшие члены делятся одновременно на некоторый одночлен w, отличный от констант.

Определение: Полином f редуцирован относительно g, если старший моном полинома f не делится на старшие члены полиномов из g.

Лексикографический метод упорядочивания многочленов от нескольких переменных: известно, что введение в исходном алфавите полного упорядочивания приводит к полному упорядочиванию составленных из него слов. В каждом выражении с коммутативной операцией производится упорядочивание составляющих его слов в соответствии с указанным отношением, и в качестве представителя класса эквивалентных выражений выбирается старшее из них.

В системе Maple существует пакет Groebner с командой Groebner[Basis] (), которая позволяет быстро находить базис Грёбнера для заданной системы.

```
> G1 := Groebner[Basis]( B1, plex(x,y,z) );
```

$$G1 := [z^4 - 3z^2 + 3, y^2 + z^2 - 3, -yz^3 + 3x]$$

```
> G2 := Groebner[Basis]( B2, plex(x,y,z) );
```

$$G2 := [1]$$

Также мы можем поэтапно найти базис Грёбнера в системе Maple с помощью реализации алгоритма Бухбергера.

Для этого нам понадобятся команды:

LeadingMonomial () –которая позволяет найти моном высшей степени из данного многочлена с помощью лексикографического упорядочивания.

SPolynomial () – позволяет находить зацепление (S (f_i, f_j)).

NormalForm () – редуцирование.

Рассмотрим пример: пусть дана САУ $\{f_1, f_2, f_3\}$

$f1 := a \cdot b - c^2 - c;$

$$f1 := a\,b - c^2 - c$$

$f2 := a^2 - a - b \cdot c$

$$f2 := a^2 - b\,c - a$$

$f3 := (a \cdot c) - b^2 - b;$

$$f3 := a\,c - b^2 - b$$

$$a\,b$$

Найдем старшие члены с помощью лексикографического упорядочивания командой LeadingMonomial()

> $LeadingMonomial(f1, plex(a, b, c));$

$$a\,b$$

> $LeadingMonomial(f2, plex(a, b, c));$

$$a^2$$

> $LeadingMonomial(f3, plex(a, b, c));$

$$a\,c$$

Найдем зацепление $f_4 = S(f_1, f_2)$ с помощью команды SPolynomial ().

> $f4 := SPolynomial(f1, f2, plex(a, b, c));$

$$f4 := -a\,c^2 + b^2\,c + a\,b - a\,c$$

Редуцируем f_1 относительно $G_0 = \{f_1, f_2, f_3\}$

```
> f4 := NormalForm(f4, G0, plex(a, b, c));
```

$$f4 := -b^2 - b\,c + c^2 - b + c$$

Если остаток не равен 0, добавляем f_4 вG_1= $\{f_1, f_2, f_3, f_4\}$

```
> G1 := [f1, f2, f3, f4];
```

Аналогично предыдущим действиям находим f_5 - зацепление $S(f_2 f_3)$, f_6 - зацепление $S(f_1\ f_3)$, f_7 - зацепление $S(f_1 f_4)$, редуцируем, добавляем к G_1, если остаток не равен нулю.

В итоге получаем базис Грёбнера

```
> G1;
```

$$[a\,b - c^2 - c,\ a^2 - b\,c - a,\ a\,c - b^2 - b,\ -b^2 - b\,c + c^2 - b + c,\ 2\,b\,c,\ -2\,c^3 - 2\,c^2]$$

Теорема (о количестве конечных решений):

Число решений системы $\langle f_1 \dots, f_n \rangle$конечно тогда и только тогда, когда базис Гребнера идеала $I = \langle f_1 \dots, f_n \rangle$ содержит $f_1 \dots f_n$, старшие члены которых являются степенями переменных $x_1 \dots, x_n$ соответственно.

За конечное число шагов мы получим набор $\{f_1, f_2, f_3, \dots f_n, f_{n+1}, \dots\}$. , f_n где каждое зацепление разрешимо. Это и есть базис Грёбнера идеала $I = \langle f_1 \dots, f_n \rangle$

Тогда базис Грёбнера для данной САУ:$\{f_1, f_2, f_3, f_4, f_5, f_6\}$. Найдем решения из $-2c^3 - 2c^2 = 0 \Rightarrow c = 0$ или $c = -1$. При $c = -1.\ 2bc = 0\ b = 0.$ Имеем $a^2 - a - bc = 0 \Rightarrow a^2 = a \ \Rightarrow a = 0$ или $a = 1$. Наборы (1,0,-1) – не подходят, (0,0,1) – подходят. При $c = 0,\ a^2 = a, a = 0, a = 1, b^2 + b = 0, b = 0, b = -1$. Наборы (0,0,0), (0,-1,0), (1,0,0)- подходят. (1,-1,0) - не подходят.

Ответ: {(0, 0, 0); (1, 0, 0); (0, −1, 0); (0, 0, −1)}.

Список использованных источников:
1. И.В.Аржанцев Базисы Грёбнера и системы алгебраических уравнений
2. https://habrahabr.ru/post/177237/
3. http://halgebra.math.msu.su/wiki/lib/exe/fetch.php/specialcourses:lnd_popovsky.pdf

ПРОГРАММА СБОРА ДАННЫХ О СТРУКТУРЕ ВЕБ-САЙТОВ

Белорусский государственный университет информатики и радиоэлектроники г. Минск, Республика Беларусь

Потехин А.С.

Стержанов М.В. – к.т.н. доцент

В настоящее время всеобщие глобальные тенденции приближаются к тому, что все операции и торговые сделки будут проходить с использованием веб-ресурсов. Для того, чтобы успешно вести бизнес очень важно получать актуальные данные о движения рынка (динамика цен и товаров) и локальные новости, которые порой всецело влияют на формирование спроса, своевременно. Но необходимые данные не всегда легко доступны пользователю и чаще всего они неструктурированны. Рассматривается приложение, которое будет обладать необходимым функционалом для сбора и структурирования данных с различных веб-ресурсов.

Целью исследования, для которого необходим сбор данных из Сети Интернет, является сентимент-анализ данных с различных новостных сайтов. Данные должны содержать полную информацию о новости, включая заголовок, текст, дату и автора новости. Для того, чтобы обеспечить сбор указанной информации, необходимо реализовать инструмент - web-scraper.

В широком понимании web scraping — это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: автоматизированный код выполняет запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат. Т.е. инструменты веб-скрапинга позволяют вручную или автоматически извлекать новые или обновленные данные и сохранять их для последующего использования.

Для того чтобы выполнять эту задачу, инструмент должен поддерживать работу со следующими данными:

HTML, JavaScript, так как большинство сайтов построены с использованием этих технологий;

Plaintext, PDF и другие форматы представления текстовых данных;

URLs, с возможностью построения ни их основе графа веб-ресурсов.

Также инструмент должен обладать требованиями [1],[2],[3]:

– Надежность – Веб содержит ресурсы, которые могут вводить скрапер в бесконечный цикл или недоступные сервисы, ожидать выполнения которых он не должен. Скрапер должен быть устойчивым к таким ловушкам;
– Вежливость – интернет-ресурсы имеют явные и неявные политики, регулирующие частоту, с которой скрапер может посетить их. Они описаны в файле robots.txt и эти политики должны соблюдаться;
– Распределенность – скрапер должен иметь возможность выполняться в распределенном режиме на нескольких машинах;