

ИСПОЛЬЗОВАНИЕ МЕТОДОЛОГИИ «ИНКРЕМЕНТНАЯ МОДЕЛЬ» ДЛЯ РЕАЛИЗАЦИИ ПЛАГИНА «ПРИНЯТИЕ РЕШЕНИЙ» ДЛЯ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ «JIRA»

Диденко А.А., студент гр. ПИм-171, I курс

Научный руководитель: Сарапулова Т.В., к.т.н., доцент

Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

В современном мире существует большое количество различных методологий для разработки информационных систем. Выбор конкретной методологии для проекта по большей мере зависит от его специфики, например, от таких показателей как сложность проекта, размер, количество сотрудников, наличие четкого техзадания, а также требований к системе и т.д.

Реализуемый в рамках данной статьи плагин, представляет из себя надстройку над базовой системой управления проектами «JIRA». Он собирает информацию из системы, обрабатывает её и выводит в виде дерева подзадач со значениями критерия «принятия решений», который высчитывается на основе таких параметров задач как продолжительность, сложность, дата завершения и т.д. Кроме того плагин отрисовывает график, посмотрев на который, легко можно выявить самую проблемную задачу текущего уровня подзадач (Рис. 1, 2, 3), а сопоставив графики всех уровней можно выделить самую проблемную цепочку задач проекта (Рис. 4), решение которой наиболее важно в данный момент времени для успешной реализации проекта. Сам плагин в техническом плане представляет из себя независимый от других модуль, который легко можно дорабатывать и приспособливать под изменяющиеся требования к системе.

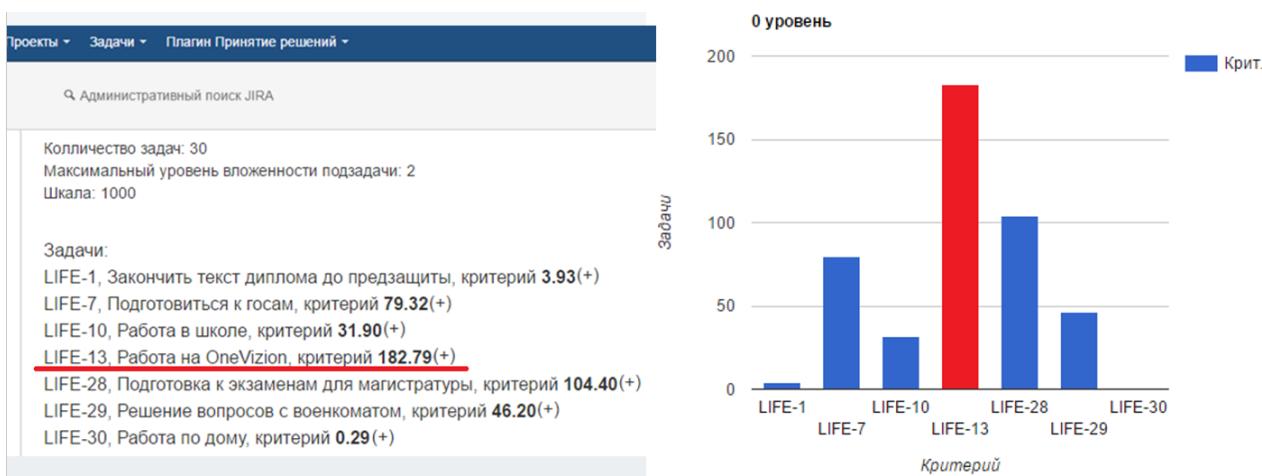


Рис. 1. Задачи проекта нулевого уровня.

LIFE-13, Работа на OneVizion, критерий 182.79(+)

- Описание: Все виды деятельности
- Процент завершения: 15.0%
- Уровень: 0
- Подзадачи: (+)
 - Подзадача LIFE-14, Работа над проектом Energy, критерий 24.01(+)
 - Подзадача LIFE-20, Проект обновления сайтов до 8 версии, критерий 38.39(+)
 - Описание: Все что связано с обновлением сайтов на 8 версию
 - Процент завершения: 40.0%
 - Уровень: 1
 - Подзадачи: (+)
 - Подзадача LIFE-21, TURF-GD, критерий 0.52(+)
 - Подзадача LIFE-22, GD, критерий 0.99(+)
 - Подзадача LIFE-23, AST, критерий 10.10(+)
 - Подзадача LIFE-24, SPRINT, критерий 15.42(+)
 - Подзадача LIFE-25, Тестирование, критерий 29.53(+)
 - Подзадача LIFE-26, Обучение, критерий 6.31(+)
 - Подзадача LIFE-27, Мелкие задачи, критерий 12.57(+)

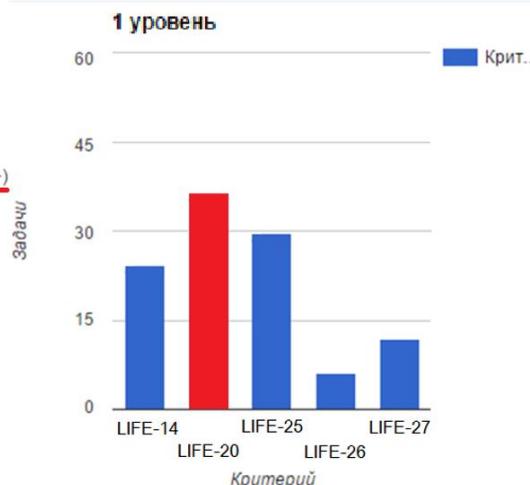


Рис. 2. Задачи проекта первого уровня.

- Подзадача LIFE-14, Работа над проектом Energy, критерий 24.01(+)
- Подзадача LIFE-20, Проект обновления сайтов до 8 версии, критерий 29.53(+)

- Описание: Все что связано с обновлением сайтов на 8 версию
- Процент завершения: 40.0%
- Уровень: 1
- Подзадачи: (+)
 - Подзадача LIFE-21, TURF-GD, критерий 0.52(+)
 - Описание: Подготовка и обновление скриптов для TURF-GD
 - Процент завершения: 90.0%
 - Уровень: 2
 - Подзадача LIFE-22, GD, критерий 0.99(+)
 - Подзадача LIFE-23, AST, критерий 10.10(+)
 - Подзадача LIFE-24, SPRINT, критерий 15.42(+)

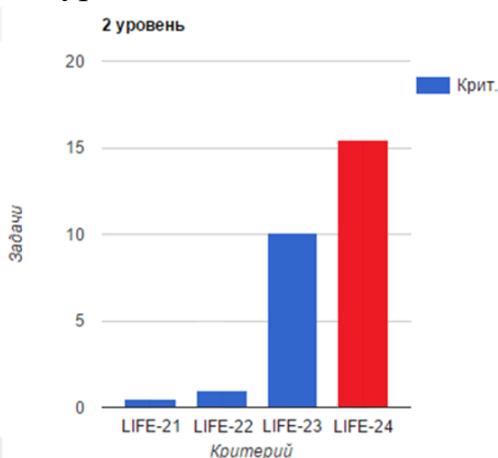
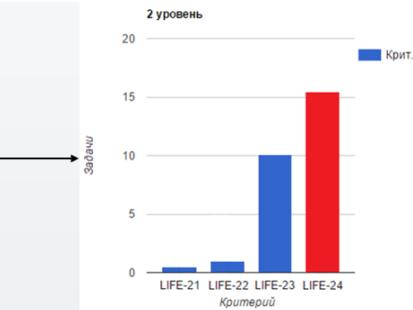
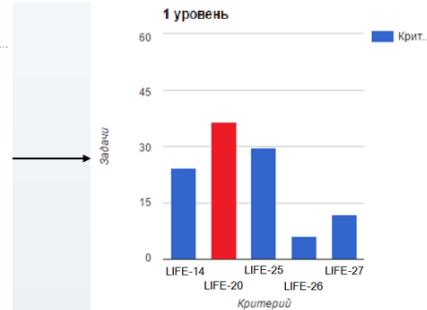
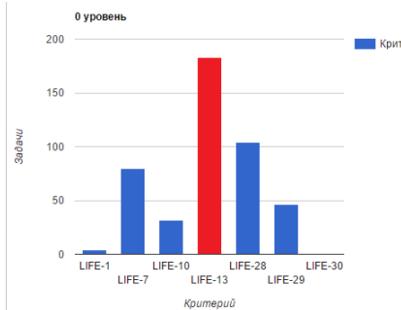


Рис. 3. Задачи проекта второго уровня.

**Проблемная цепочка:**

LIFE-13 → LIFE-20 → LIFE-24

Рис. 4. Проблемная цепочка проекта.

При разработке данного плагина было принято решение использовать методологию «Инкрементная модель» [1] (Рис. 5). Данному решению способствовало то, что хоть и базовое задание было четко, но некоторые моменты могли быть изменены в ходе разработки, например, такие как формат входных и выходных данных, формулы для расчета критерия и т.д. Кроме того данная задача требовала предоставления рабочих прототипов на ранних этапах разработки для корректировки формул расчета.



Рис. 5. Методология «Инкрементная модель».

Идея, которая лежит в основе инкрементной модели разработки, состоит в том, что программный продукт стоит разрабатывать по принципу приращений, то есть так, чтобы разработчик мог использовать данные, полученные во время разработки более ранних версий (релизов) программного обеспечения (ПО). Таким образом, новые данные получаются, не только во время разработки ПО, но также и в ходе его эксплуатации, где это возможно [2]. Ключевые этапы данного процесса – это простота в реализация требований к программе, а также улучшение ПО в серии последовательных релизов до тех пор, пока не будет реализован программный продукт, удовлетворяющий всем требованиям техзадания. В ходе каждой такой итерации организация модели изменяется, а также к ней добавляются все новые и новые функциональные модули.

Кроме того ПО можно вводить в эксплуатацию по частям, а значит, и поставлять заказчику и разрабатывать его также можно постепенно. Именно на этом основана инкрементная модель, которая предусматривает деление продукта на довольно независимые компоненты, которые можно разрабатывать и вводить в эксплуатацию раздельно друг от друга.

Данная модель выгодна для обеих сторон, как для заказчика, так и для разработчика, из-за того что позволяет продвигаться вперед, соблюдая интересы каждой из сторон. Но у нее также имеются и свои недостатки. Деление системы на функциональные блоки обычно замедляет процесс, из-за того что возникает необходимость в обеспечении их взаимодействия. Кроме того для многих решений данный метод неприменим, потому что из них нельзя выделить отдельные модули, которые могли бы функционировать независимо от других [3]. Также значительно возрастает нагрузка и на руководящий

персонал из-за усложнения задач по координированию работ над отдельными модулями системы, и растет стоимость внесения правок и изменений в готовые компоненты, которые уже были установлены и на данный момент работают у заказчика.

Инкрементная модель была применена в таких системах, как система защищенного постраничного просмотра электронных документов в PDF «DefView» и система электронных библиотек «Vivaldi».

Пример применения инкрементной модели при разработке плагина
«Принятие решений».

1 итерация.

1. Выбран простейший метод расчета критерия «Принятие решений», представляющий из себя сумму всех входящих параметров.
2. Разработан модуль получения и обработки данных из системы управления проектами «JIRA».
3. Разработан простейший модуль вывода, обработанной плагином, информации в виде таблицы.

2 итерация.

1. Метод расчета усовершенствован на основе полученных данных из прошлой итерации и литературы.
2. Добавлена возможность считывать данные из внешних баз данных.

3 итерация.

1. Метод расчета усовершенствован на основе данных из прошлых итераций и расширен дополнительными параметрами.
2. Добавлена возможность считывать сложные типы данных.
3. Добавлена возможность считывать данные с помощью REST.
4. Усовершенствован модуль вывода информации, теперь информация выводится в виде дерева, а так же отрисовываются графики.

Результаты применения инкрементной модели при разработке плагина
«Принятие решений».

Используя методологию «Инкрементная модель», был разработан, а затем в ходе итераций и значительно усовершенствован плагин «Принятие решений». В дальнейшем его формула расчета критерия будет еще сильнее оптимизирована, а также в него будут добавлены другие методы отрисовки данных. Выбранная методология идеально подходит для разработки данной подсистемы, так как существует необходимость в постоянном улучшении и оптимизации формулы расчета критерия под специфику проекта, а также в расширении базового плагин дополнительными подмодулями, которые в случае необходимости легко можно отключить, не нарушив работоспособность основного плагина.

Список литературы:

1. Ещё раз про семь основных методологий разработки, URL:
<https://habrahabr.ru/company/edison/blog/269789/> (дата обращения: 18.01.2018).
2. Жизненный цикл программных систем, Инкрементная модель, URL: <https://www.intuit.ru/studies/courses/3632/874/lecture/14297?page=5> (дата обращения: 18.01.2018).
3. Разработка ПО: модели жизненного цикла, URL:
<https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema3> (дата обращения: 18.01.2018).