Слайд 1.

На сегодняшний день система отслеживания ошибок и управления проектами Jira остается наиболее популярной. Однако она не располагает средством для контроля большого потока создаваемых задач при выполнении работы, запланированной на некоторый длительный период, например, на полгода. Поэтому возникла идея создания для этой цели расширения.

Слайд 2.

Была реализована часть плагина, позволяющая создавать планы. Целью моей работы является реализация функциональности, представленной на слайде.

Слайд 3.

Кроме того, приложение должно поддерживать локализацию интерфейса на русском и английском языке.

Слайд 4.

На основе наличия необходимой функциональности был проанализирован рынок существующих решений. Подробный анализ нескольких наиболее подходящих плагинов отражен в разделе 2.1 моей работы. На слайде представлены сделанные в результате выводы.

Слайд 5.

Был проведен анализ реализованной части приложения. На слайде отражено взаимодействие его модулей.

Слайд 6.

Приложение построено на базе шаблона MVC, структура которого представлена на слайде.

Слайд 7.

Клиентская часть реализована в виде html-шаблонов и файлов, написанных на JavaScript, которые схематически изображены на слайде. Здесь есть файл для работы с каждой из трех страниц плагина, использующий свой файл с Backbone-моделями и представлениями.

Слайд 8.

На данном слайде представлена диаграмма пакетов серверной части. Пакеты “restApi” и «servlet» содержат классы контроллеров и сервлетов, «ao» - классы моделей и сервисов, работающих с БД. Был сделан вывод, что необходимо провести рефакторинг и разделить бизнес-логику и логику по работе с БД.

Слайд 9.

Задача реализации работы со снимками деревьев задач делится на создание снимков вручную и автоматически. При этом, для экономии дискового пространства должно выполняться автоматическое удаление снимков. На слайде схематически изображено, какие снимки должны оставаться, а именно ежедневные за последнюю неделю, еженедельные за последний месяц и ежемесячные за остальной период плана.

Слайд 10.

Для создания, установки и сборки плагина использовался Atlassian Plugin SDK и Maven. Серверная часть реализовывалась на Java с использованием для построения SQL-запросов библиотеки Active Objects, основанной на ORM-технологии. Для клиентской части использовался JS с jQuery и фреймворк Backbone.js. (представлены на слайде и все)

Слайд 11.

Требования к аппаратному и программному обеспечению представлены на слайде.

Слайд 12.

Рассмотрим пользовательский интерфейс.

Для создания снимков вручную была добавлена кнопка в панель управления, фрагмент которой показан на слайде.

Слайд 13.

При нажатии на нее открывается окно для ввода некоторого названия или комментария.

Слайд 14.

На данном слайде показано окно настроек плана, в нижней части которого расположены чекбоксы для включения и отключения функций автоматического создания снимков и рассылки почтовых уведомлений об ошибках плана.

Слайд 15.

Для сравнения снимков требуется на странице задач плана выбрать режим «Сравнение снимков».

Слайд 16.

В режиме сравнения на странице расположены выпадающие списки для выбора снимков и одного или нескольких критериев сравнения. В первом списке по умолчанию выбран пункт «Текущее состояние», т.к. часто требуется сравнить текущее состояние дерева со снимком, не создавая для этого отдельный снимок. Во втором списке по умолчанию, устанавливается последний снимок. Сравнение запускается по нажатию на кнопку «Сравнить».

Слайд 17.

Результат выводится следующим образом: зеленым цветом отмечаются новые задачи, красным – удаленные. Оранжевым цветом отмечаются все изменения, при наведении выводится старое значение. Для каждой задачи выводятся иконки для обозначения наличия добавлений, удалений или изменений в ее поддереве.

Слайд 18.

Для функции поиска последних измененных задач доступна настройка их количества в виде чекбокса в окне настроек.

Слайд 19.

Поиск и вывод последних измененных задач осуществляется при наведении на иконку часов.

Слайд 20.

На данном слайде показан пример использования JQL-функции поиска задач плана или его части, т.е. поддерева. В данном случае для функции были заданы название плана и ключ задачи. Предупреждающее сообщение информирует пользователя о том, что он не задал номер версии плана и поиск по умолчанию был выполнен для актуальной версии.

Слайд 21.

Для функции авто-корректировки оценки была создана панель, встроенная в страницу задачи. Она выделена на скриншоте зеленой рамкой и содержит информацию о вхождении задачи в действующий план и кнопку для корректировки.

Слайд 22.

По нажатию на которую осуществляется поиск всех родительских задач вплоть до корневой и открывается окно.

Следует кликнуть на задачу, оценка которой будет скорректирована.

Слайд 23.

Для вывода ошибок плана сделана отдельная вкладка. Ошибки группируются по типам. На вкладке выводится бейдж с их общим количеством.

Слайд 24.

На следующем слайде показан образец электронного письма с ошибками. Ежедневно администратор и наблюдатели плана получают подобное письмо, а остальные сотрудники уведомляются только об их личных ошибках, таких как добавление отчета в задачу вне плана.

Слайд 25.

Здесь изображен процесс перетаскивания задачи «PLM-7». Как только кнопку мыши отпустят, она будет привязана как дочерняя задача для «PLM-4».

Слайд 26.

На слайде представлена новая диаграмма пакетов.

В сервисах только бизнес-логика, работа с базой данных реализована в DAO-классах, находящихся в пакете «dao». Сервисы и dao-классы теперь представлены в виде интерфейсов, которые реализуются классами в соответствующих подпакетах «impl». Пакет «entity» содержит интерфейсы моделей.

Созданы новые пакеты:

* «upgrade» - содержит классы для обновления структуры БД с сохранением обратной совместимости;
* «job» - содержит классы, выполняющие проверку плана на наличие ошибок, рассылку уведомлений об ошибках и автоматическое создание и удаление снимков по расписанию;
* «treebuilder» - содержит классы для построения дерева задач и вспомогательные методы для работы с деревом;
* «planerrors» - содержит классы, представляющие собой ошибки плана;
* «webpanel» - содержит класс, обеспечивающий контекст для отображения панели авто-корректировки оценки;
* «condition» - включает специальные классы различных условий, в т.ч. условия отображения панели авто-корректировки;

Слайд 27.

Для хранения снимков были созданы две таблицы, представленные на физической модели части БД на слайде. Таблица Snapshot хранит данные снимков. Таблица SnapshotIssue хранит данные задач снимков. Таблица Version приведена без атрибутов для понимания связи таблиц снимков с остальной БД. Так, снимки связаны с версиями планов.

Слайд 28.

Рассмотрим диаграмму классов ошибок плана. Все они наследуются от абстрактного класса. Т.о. можно однотипно работать с коллекцией различных ошибок.

Слайд 29.

В результате проделанной работы были успешно разработаны соответствующие требованиям программные средства для перечисленных на слайде функций.

Слайд 30.

Приложение внедрено в работу предприятия и будет опубликовано на Atlassian Marketplace.