Уважаемые члены комиссии, тема моего дипломного проекта «Система учета метрик и распределения ресурсов при Agile разработке».

1. На сегодняшний день программные продукты используются повсеместно в жизни человека: от мобильных телефонов до спутниковых систем.
2. Поэтому качество программного обеспечения, его готовность в срок, а так же стоимость стало играть немаловажную роль в построении процесса разработки программного продукта.
3. Существует набор подходов, рекомендаций, позволяющих менеджерам оптимизировать процесс разработки и «держать руку на пульсе» для предотвращения неожиданных выпадов из графика или бюджета – это гибкая методология разработки (некоторые из подходов, а так же цикл гибкой разработки на плакате 1) .
4. Целю дипломного проекта является упрощение отслеживания динамики разработки программного продукта, оптимизация распределения задач между разработчиками и облегчение составления отчетности и построения графиков по проекту.
5. Приложение должно предоставлять возможность настройки на любые типы рисков, работать с любыми методологиями управления проектами (scrum, kanban) и интегрироваться с любыми багтрекинговыми системами.
6. Идея проекта следующая: всякая задача в процессе решения проходит стандартный жизненный цикл: ее создание, формирование требований к ее решению, разработка, тестирование и закрытие.
7. Знание этого цикла помогает отслеживать риски касающиеся задач при переходе задачи из одного состояния в другое.
8. Существуют и риски, касающиеся непосредственно разработчиков: болезнь, текучесть кадров, производительность труда. Эти риски указаны на плакате 2 как «многокомпонентные» (причинные, слагающие).
9. Риски, касающиеся проекта в целом, называют совокупные, и они составляются из причинных рисков.
10. Неопределенность относительно совокупного результата является прямым результатом неопределенностей причинных факторов, ведущих к успеху или провалу.
11. Процесс преобразования набора причинных рисков в совокупный – модель риска.
12. Таким образом, программа должна собрать информацию о причинных рисках и составить диаграмму совокупного риска, прогнозирующую вероятности завершения проекта в диапазоне дат.
13. Для реализации поставленной задачи было разработано ядро системы. Полностью независимое от хранилища данных или пользовательского интерфейса.
14. В ядре реализованы методы, предоставляющие пользователю ядра возможность переводить задачи из состояния в состояние, импортировать и экспортировать дынные, получать данные для построения графиков, а так же распределять задачи между сотрудниками и выбирать задачи, которые могут быть решены в заданный временной интервал.
15. Для использования ядра была разработана система интерфейсов (чертеж 1), реализуя которые можно определить место и образ хранения данных в базах данных, способы импорта данных из багтрекинговой системы, а так же экспорта данных для составления отчетов.
16. Система выполнена с использование шаблона проектирования «Мост», что позволяет разделять ответственность между классами логики и классами подключения к базам данных.
17. Для создания сущности ядра необходимо реализовать интерфейс IDataStorage, содержащий методы для добавления, удаления и редактирования данных в хранилище данных.
18. Чтобы предоставить возможность приложению в перспективе подключаться к базам данных различных типов был создан еще 1 интерфейс IConnectToDB, реализация которого определяет тип хранилища, к которому выполняются запросы.
19. Для демонстрации работы ядра был создано клиент-серверное приложение с использованием MVC 3.
20. На чертеже 2 представлен пример запроса с клиента на список задач в системе.
21. Класс CoreWrapper – buisiness logic layer приложения, содержит методы необходимые для его работы. А так же предоставляет доступ к методам ядра, агрегируя его сущность.
22. Система ориентирована на 3 категории пользователей: менеджеры, разработчики и клиенты (заказчики).
23. На диаграмме использования (чертеж 3) представлены опции, доступные этим трем категориям: самый низкие права доступа у клиента: он может только просматривать динамику разработки проекта и добавлять комментарии к задачам.
24. На плакате 3 показаны результаты работы программы. Здесь статистика по выполненному объему работ за последние 5 спринтов, график сгорания для текущего активного спринта и пример экспорта данных по текущему спринту в формат XML.
25. Таким образом, была разработана программа, предоставляющая возможность отслеживать динамику разработки проекта, генерировать отчеты для заказчика, прогнозировать количество решенных задач в срок, а так же распределять задачи между разработчиками учитывая их показатели производительности.

На этом все, спасибо за внимание.