

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г.И. Носова»
кафедра вычислительной техники и программирования

Лабораторная работа №4

по дисциплине «Проектирование программных средств»

название разработки: «Модуль системы CSIA для проверки целостности
описания сборки продукта»

Выполнил: Варламов М.Н. студент 4 курса, группа АВб-19-1

Проверила: Кочержинская Ю.В., доцент кафедры ВТ и П, к.т.н.

Магнитогорск, 2022

Цель проекта

В современной разработке ПО присутствует большое количество технологий и систем, которые являются вспомогательными для данного процесса. В зависимости от типа и сложности продукта используются различные разработанные продукты, но среди всего многообразия различных систем выделяются те, которые необходимы для написания практически любого программного обеспечения. К таким системам можно отнести:

- Система контроля версий (Git, SVN)
- Система управления проектом (Jira, YouTrack, Trello)

Системы контроля версий необходимы для ведения работ над одним проектом несколькими разработчиками одновременно. Они позволяют вносить изменения в проект с разных устройств, хранят различные версии продуктов с возможностью “отката”, фиксируют изменения. Также немаловажной функцией данных систем является ветвление и слияние различных модулей или частей программного кода, что очень удобно для тестирования различных новых, но не безопасных функций продукта. Такой подход упрощает ведение разработки с технической точки зрения, так как в случае ошибок можно легко откатить изменения, всегда можно посмотреть какой разработчик и когда изменил ту или иную часть программы, а также надежно хранить исходный код и не бояться возможного удаления, ведь всегда можно восстановить исходный код продукта на любую стадию разработки.

Системы управления проектом служат для ведения разработки ПО с точки зрения менеджера. Они позволяют отслеживать задачи, которые выполняют разработчики в данный момент, проставлять затраченное время и указывать пояснения к выполняемой задаче. Однако данными системами пользуются не только менеджеры продуктов, но и сами разработчики, а также клиенты, либо рядовые пользователи разрабатываемого продукта. Все эти люди могут создавать задачи различных типов, будь то ошибка в продукте, либо новый функционал. На основе данных, которыми заполняются системы управления проектом, менеджеры составляют аналитические планы, рассчитывают приблизительное время, затраченное на разработку продукта и получают

множество другой информации, полезной при составлении отчета по разрабатываемому проекту.

Из приведенной выше информации можно понять, что данные системы имеют очень близкое отношение друг к другу. Так например, при загрузке изменений в систему контроля версии можно указать номер задачи, в рамках которой велись изменения. Также можно указывать в самой задаче ее статус после внесения изменений в продукт.

На самом деле можно разработать множество различных правил похожих на те, что приведены выше, и которые будут упрощать ведение разработки программного продукта. Отсюда следует, что данные системы должны иметь интерфейсы для взаимодействия друг с другом, но так происходит не всегда. Это может происходить по многим причинам:

- Отсутствие интерфейса взаимодействия у одной из систем
- Системы могут быть устаревшими, что затрудняет их взаимодействие
- Системы могут быть слишком новыми, и данные интерфейсы могут быть еще не реализованы
- Системы могут быть закрытыми

Это лишь небольшая часть проблем, по которым взаимодействие данных систем затруднительно. По одной из этих причин внутри компании “CompassPlus”, существует продукт, разработанный на платформе “RadixWare”, под названием “CSIA”, который решает несколько задач, необходимых для упрощения администрирования разрабатываемого ПО. Одной из задач и является модуль, который позволяет “связать” системы контроля версий и управления проектом.

Реализуемый проект представляет собой модуль для “CSIA”, который служит для сбора статистических данных с системы управления проектом, их обработки и предоставлению в удобном для пользователя формате. Помимо этого данный проект может выполнять некоторые проверочные функции, связанные с системой контроля версий.

Пример пользовательской истории

Пользовательская история менеджера продукта.

Я, как менеджер проекта, хочу получать отчет по продукту, которым управляю, в котором будет содержаться статистика нарушений оформления задач и изменений, чтобы учитывать данную информацию при выпуске следующей версии продукта и повышать точности описания релиза, посредством исправления данных ошибок.

Пользовательская история разработчика продукта.

Я, как разработчик продукта, хочу получать отчет по продукту, который разрабатываю, в котором будет содержаться статистика моих нарушений оформления задач и изменений, чтобы максимально оперативно исправлять ошибки, для предотвращения возникновения дополнительных вопросов по разработке со стороны менеджера и сторонних отделов таких как: отдел тестирования, отдел документации и т.п., чья работа зависит от моего описания разработки.

На рисунках 1 и 2 представлены пользовательские истории настройки модуля и генерации отчета.

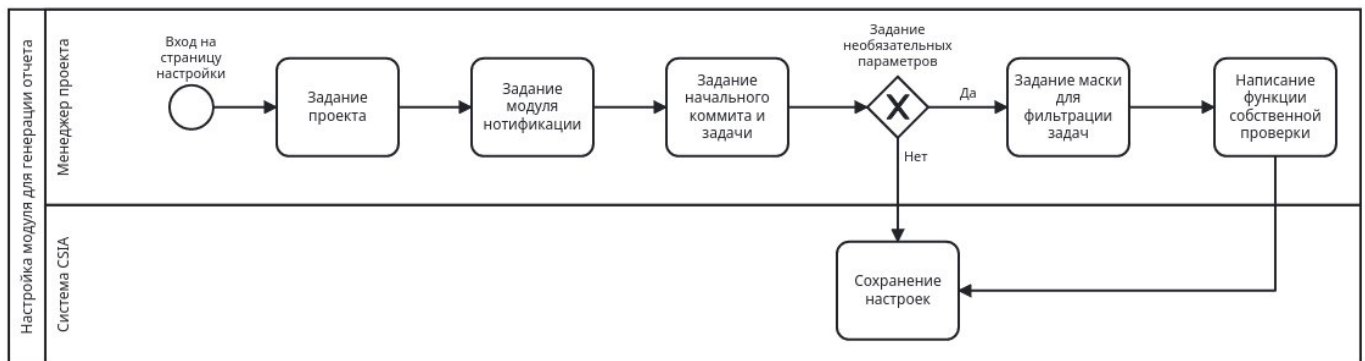


Рисунок 1 – Пользовательская история настройки модуля

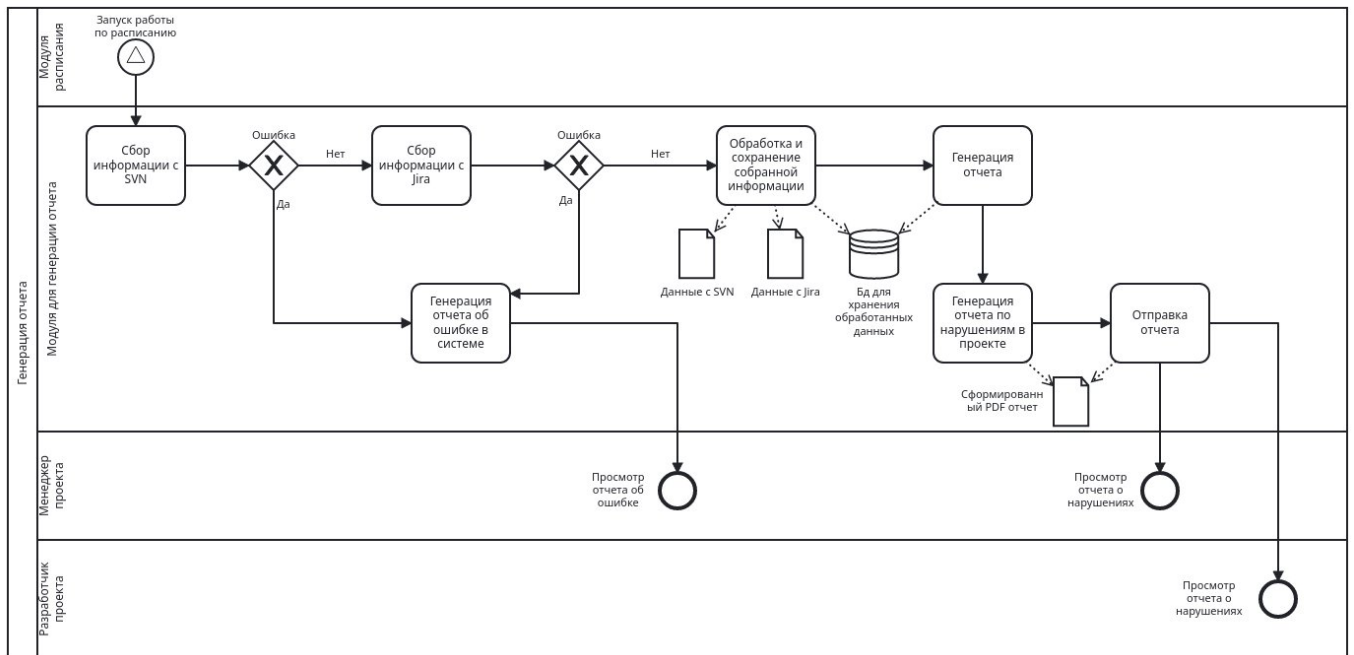


Рисунок 2 – Пользовательская история генерации отчета модулем