

Использование определённой модели ветвления в системе Github для организации совместной работы над проектом

Круглик Роман Игоревич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье рассмотрена система Github и одна из моделей ветвления. В структуре участвуют 3 ветки: главная (Master), для разработки(develop), тематическая (feature).

Ключевые слова: Github, модель ветвления, репозиторий.

Using a specific branching model in the Github system to organize collaboration on a project

Kruglik Roman Igorevich

Sholom-Aleichem Priamursky State University

Student

Abstract

In article discusses the Github system and one of the branching models. 3 branches participate in the structure: main (Master), for development (develop), thematic (feature).

Keywords: Github, branching model, repository.

Тенденции развития современных технологий в области разработки программного обеспечения активно продвигаются в сторону свободно распространяемого и открытого кода.

При создании любого программного продукта неизбежным является разделение полномочий и обязанностей при реализации частей приложения или функциональных модулей. Для эффективной работы по созданию информационной системы необходимо правильно и рационально координировать работу всей группы специалистов, работающих над данным проектом. Именно поэтому появилась проблема эффективной организации совместной работы над проектом, которую решает GitHub.

GitHub - это система управления версиями. У Git две основных задачи: первая - хранить информацию о всех изменениях в коде, начиная с самой первой строчки, а вторая - обеспечение удобства командной работы над кодом.

Для начала необходимо выбрать правильную модель ветвления, так как это является основой для работы с системой. Именно поэтому в данной статье будет рассмотрена одна из самых популярных моделей.

Исследования в области применения систем контроля версий проводятся многими. В статье А.А. Акинина, А.С. Маликова [1] рассмотрены базовые принципы работы одной из основных систем контроля версий - GIT. В работе А.В. Гаспарян, Н.В. Тимошина [2] рассматривается решение проблем организации совместной разработки программных продуктов с помощью систем контроля версий, в частности - с помощью системы Git. В статье [3] А.Л. Мархакшинов рассмотрены вопросы использования системы контроля версий в учебном процессе. Я.Б. Ерошенко, К.К. Самхарадзе [4] разобран вопрос использования систем контроля версий, отмечена проблема автоматизированной сборки программных средств в системе Git.

Хранилище Git состоит из каталогов, которые в системы называются репозиториями, хранящие в себе файлы проекта. Система стала популярной из-за разделения репозитория по различным веткам. То есть несколько разработчиков могут выполнять работу над одним проектом одновременно. Для начала необходимо зарегистрироваться на официальном ресурсе (см. рис. 1).

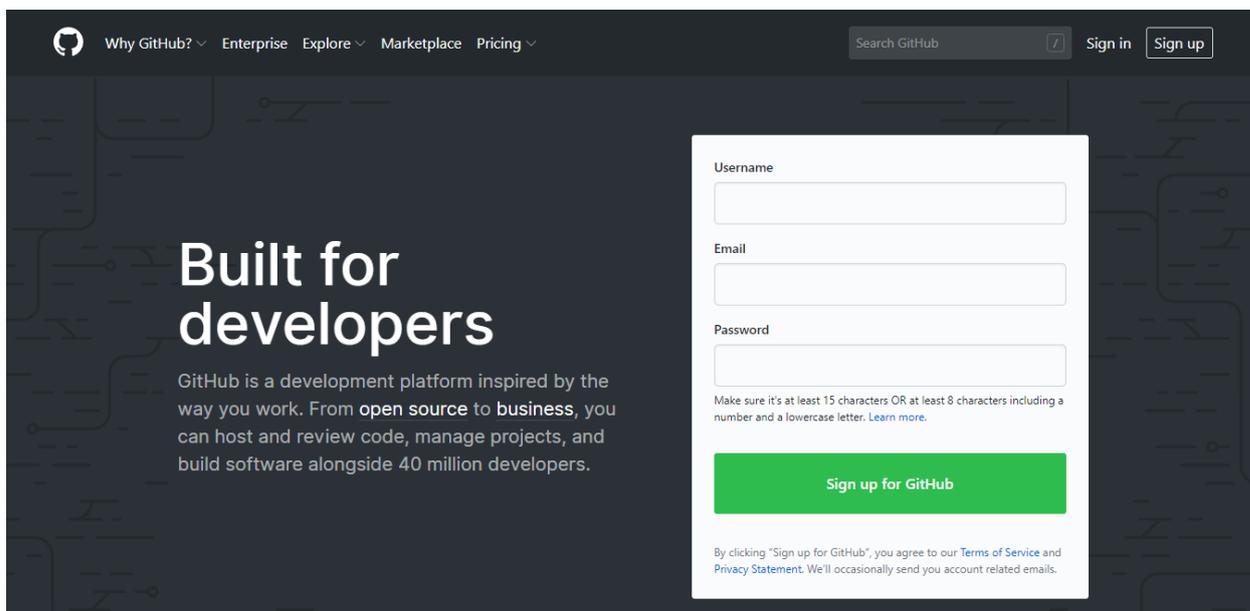


Рисунок 1. Официальный сайт GitHub

Работа над проектом с github заключается в создании нескольких веток одного проекта (см. рис. 2).

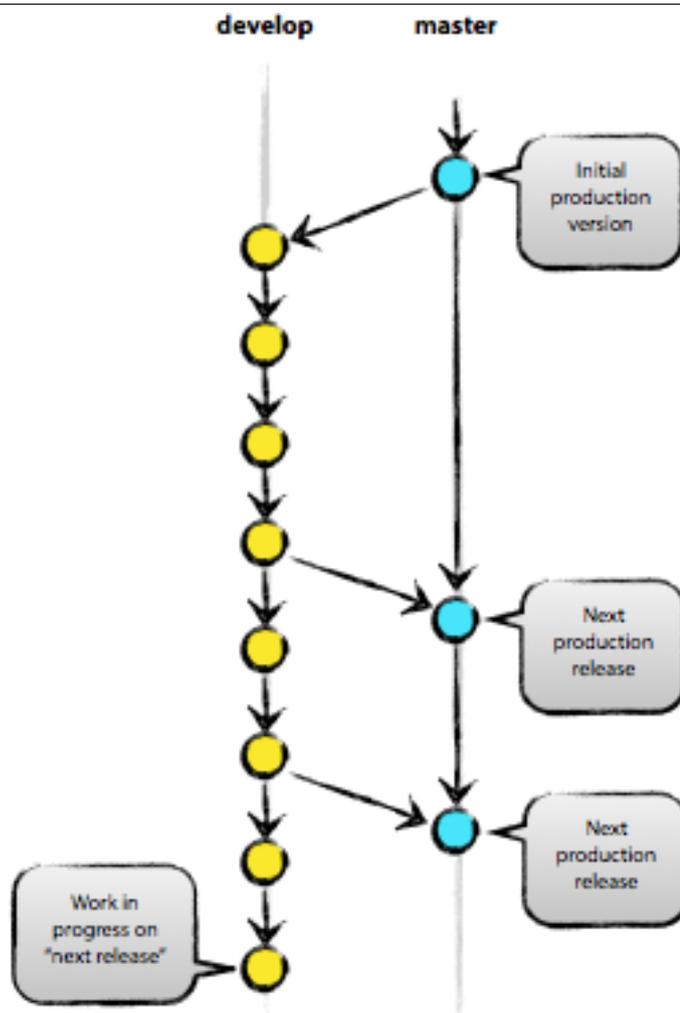


Рисунок 2. Основные ветки

Главной веткой является «Master», она изменяется только при глобальных обновлениях и создаётся сразу же после инициализации репозитория. Чаще всего это стабильная версия проекта, которая на любом этапе является полностью рабочей.

Ветка «develop» создаётся вручную и отвечает за разработку и в ней хранятся промежуточные результаты. Допустим необходимо добавить в проект один из модулей. Работа будет состоять из нескольких этапов:

1. Из главного репозитория переносятся файлы в ветку для разработки.
2. Добавляется модуль
3. Тестируется весь проект.
4. Исправляются ошибки.
5. Вносятся изменения в основную ветку.

Каждое действие сохраняется в ветке разработки. Когда проект достигает стабильного состояния после его модификаций, все изменения переносятся в ветку «Master»

Помимо 2 основных веток добавляют ещё одну «feature» (см. рис. 3).

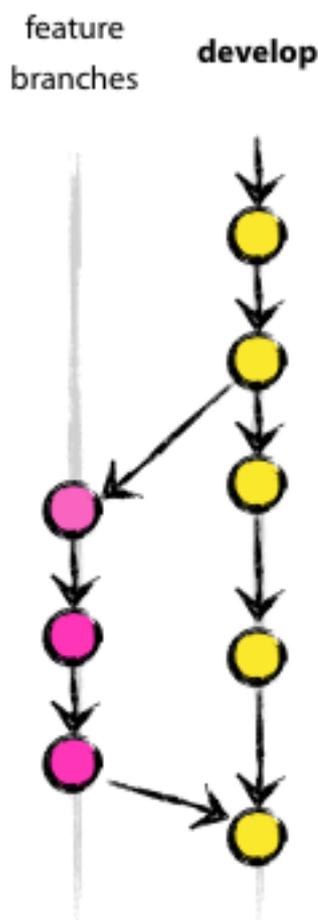


Рисунок 3. Дополнительная ветка «feature»

Данная ветка является побочной от «develop». Она создаётся для разработки отдельной функциональности проекта и живёт, пока цель не достигнута. Чаще всего эти функции способствуют улучшению интерфейса или созданию анимации в проект, то есть то, что может быть удалено при неудачном эксперименте.

Данный способ организации работы над проектом является достаточно простым и эффективным. Существует множество других способов взаимодействия репозитория и составление веток. Данную модель можно использовать как основу для организации работы над проектом в системе контроля версий.

Библиографический список

1. Акинина А.А., Маликова А.С. Рассмотрение принципов работы системы контроля версий на примере git // Синергия Наук. 2019. № 34. С. 372-392.
2. Гаспарян А.В., Тимошина Н.В. Совместная разработка по с использованием git // ИТпортал. 2017. № 1 (13). С. 3.
3. Мархакшинов А.Л. Применение системы контроля версий git в процессе обучения программированию на Java // Инновационные технологии в науке и образовании Материалы V Всероссийской научно-практической конференции с международным участием. Отв. ред. Е.Р. Урмакшинова,

С.Л. Буянтуев. 2017. С. 160-163.

4. Ерошенко Я.Б., Самхарадзе К.К. Оптимизация сборки программных средств в системе git // Аллея науки. 2017. Т. 2. № 11. С. 453-459.