



**git**

# Система контроля версий

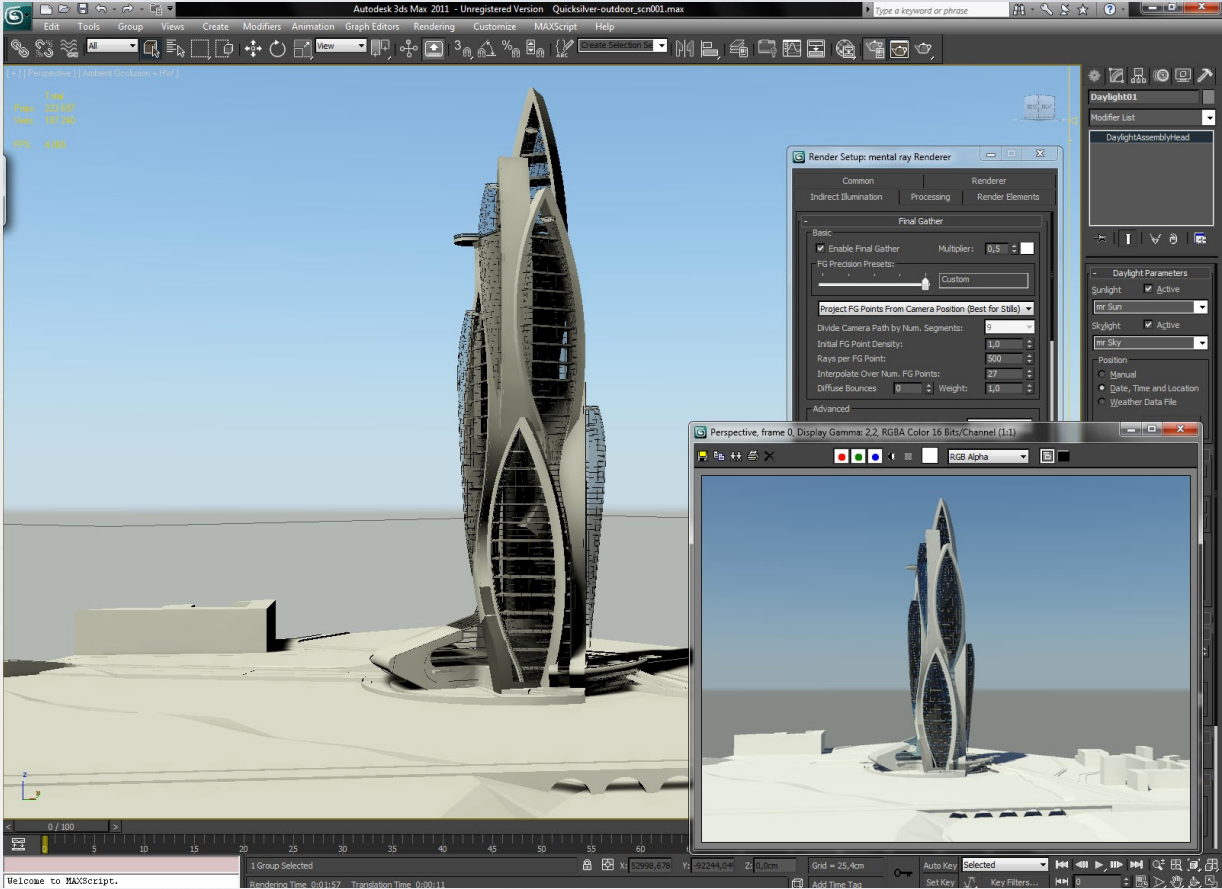
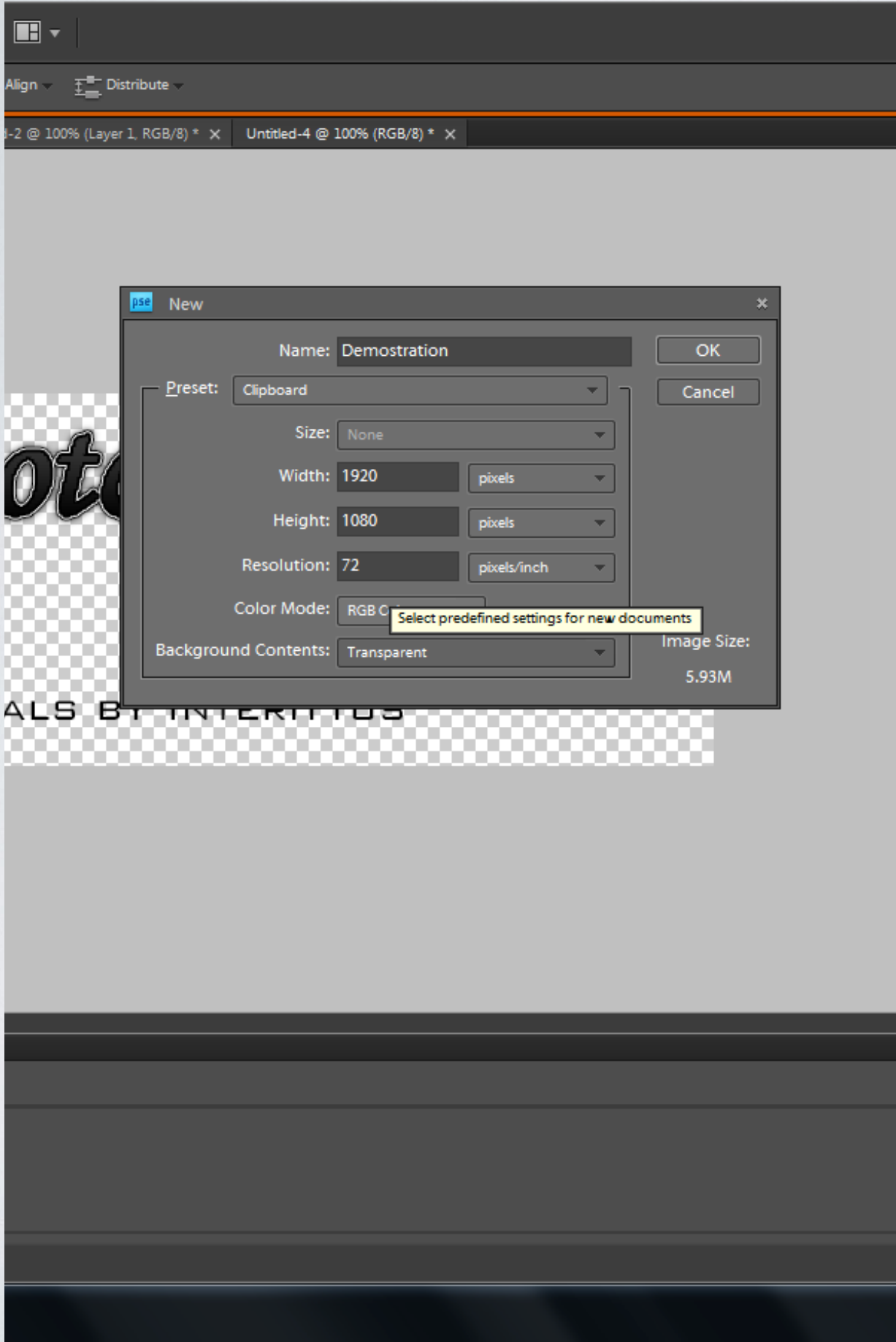
- это ПО для работы с изменяющейся информацией

## VCS позволяет:

- хранить несколько версий одного и того же документа
- возвращаться к ранней версии документа
- определять кто и когда сделал изменения







# Популярные решения

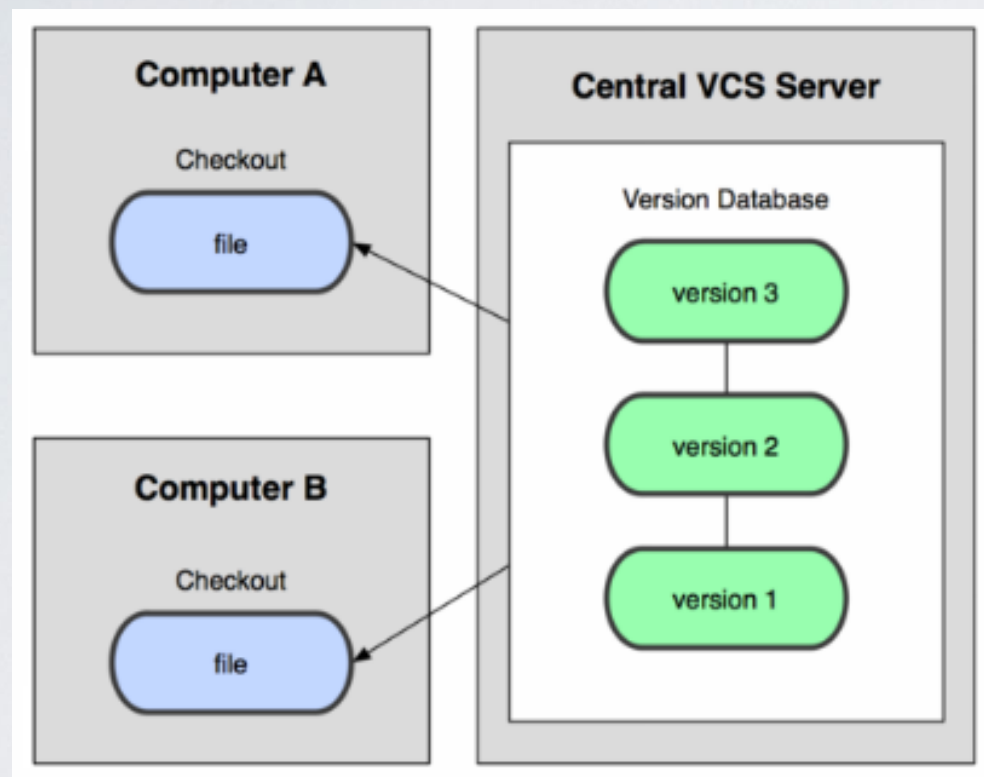




# Централизованные Системы Контроля Версий

Есть централизованный сервер.

Он отвечает за версионное хранение файлов  
за набор клиентов которые получают копии



## Недостатки

- обслуживание сервера
- зависимость от сервера
- повреждение сервера



Subversion (SVN)

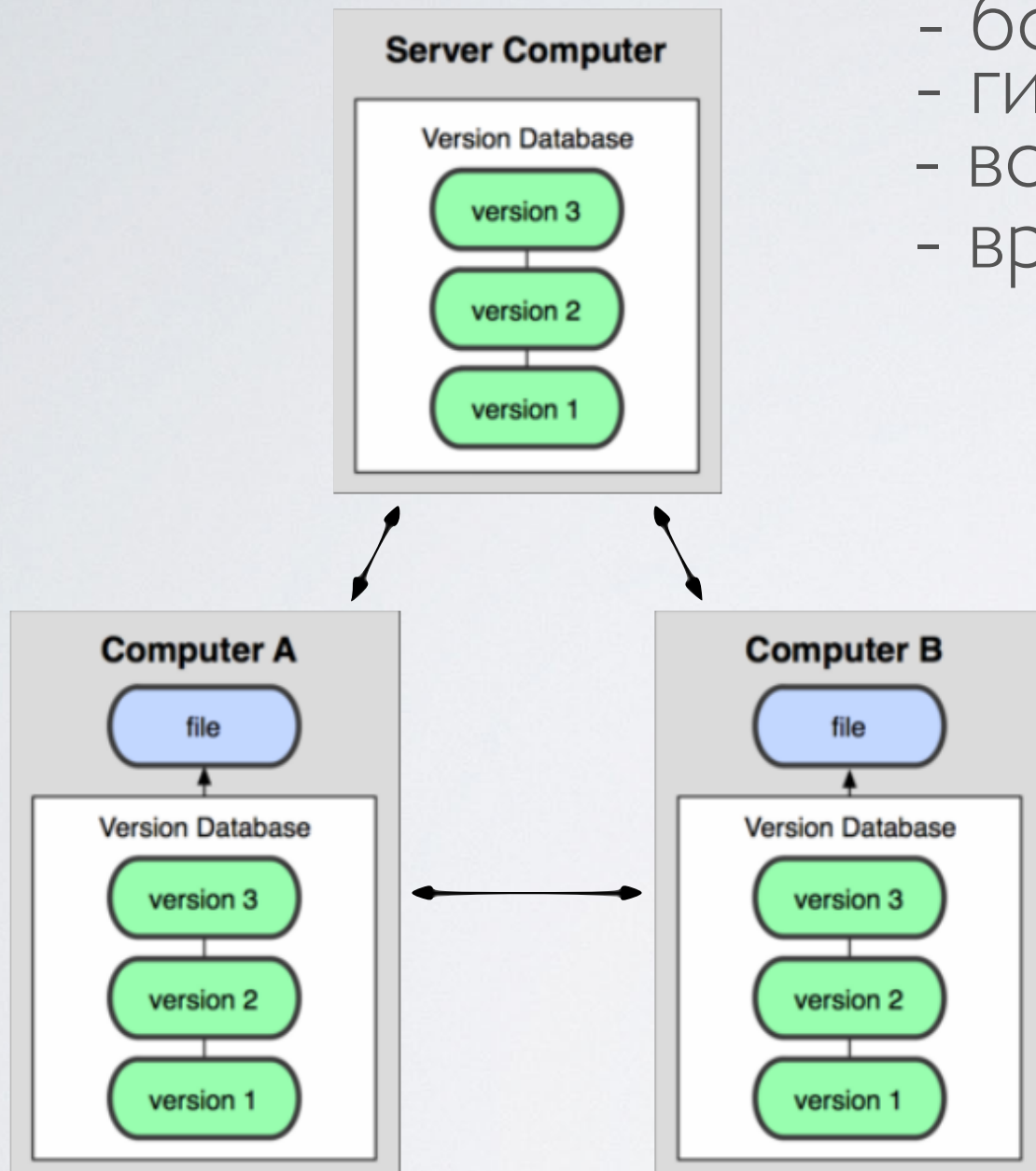
# Децентрализованные Системы Контроля Версий

## Достоинства

- более высокая надежность
- гибкость и автономия
- возможность работы с многими репо
- время выполнения

## Недостатки

- блокировка файлов
- единая нумерация версий
- много дискового пространства



# ИТОГ

## Централизованные

- группа разработчиков невелика
- сосредоточено на одной территории
- 24/7 доступные сервера

## Децентрализованные

- небольшая территориальная рассредоточенная группа dev
- крупный распределенный проект







Git - распределенная система управления версиями файлов

Создатель: Линус Торвальдс

Цель проекта: замена BitKeeper для управления  
разработкой ядра Linux

Написана на: C, Perl

Лицензия: GNU GPL v2

Первая версия: 7 апреля 2005 года

Последняя версия: 9 апреля 2014



# Why GIT?

- высокая производительность
- средства интеграции (SVN, Mercurial)
- качественный веб-интерфейс
- большое количество учебного материала
- интеграция со всеми IDE
- кроссплатформенность
- большое количество бесплатных хостингов

# Структура репозитория





# Типичная работа с веткой



# Branch.

Ветвь (branch) - направление разработки, независимое от других

Ветвь представляет собой копию части хранилища, в которую можно вносить изменения, не влияющие на другие ветви.

Ветвь в Git - это простой легковесный подвижный указатель на один из коммитов



# Единственная ветка

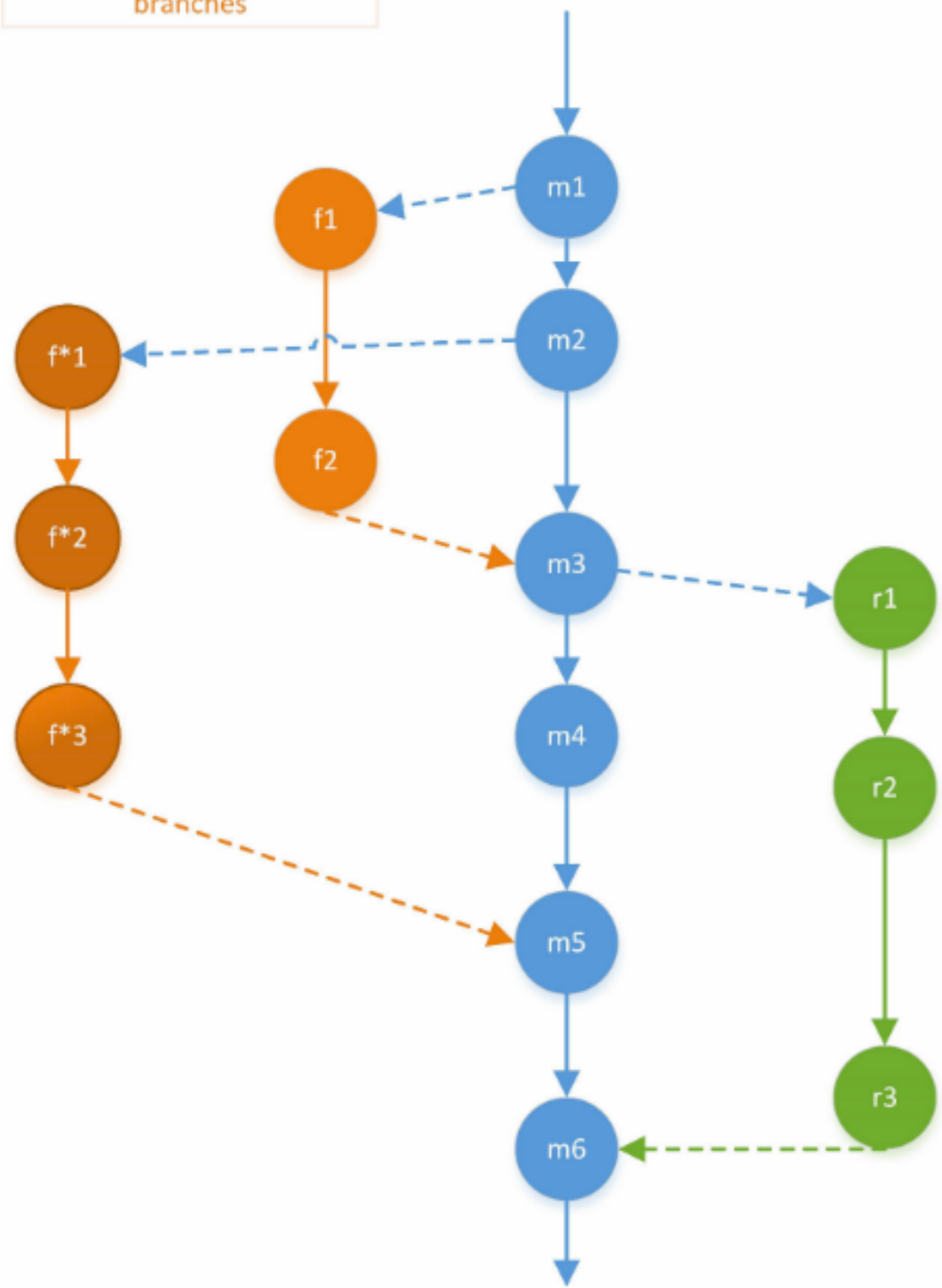
Master



Feature  
branches

Master

Release

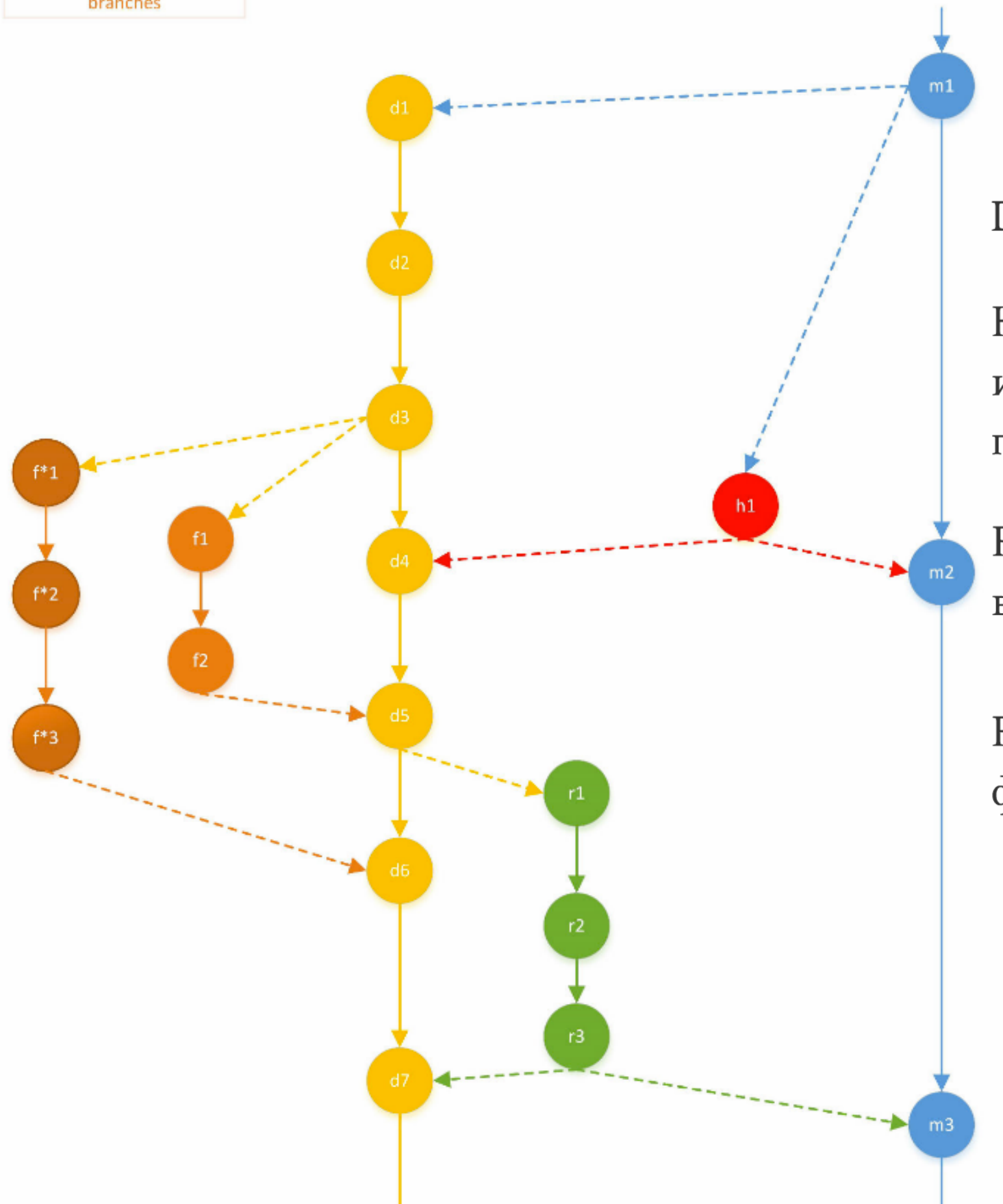
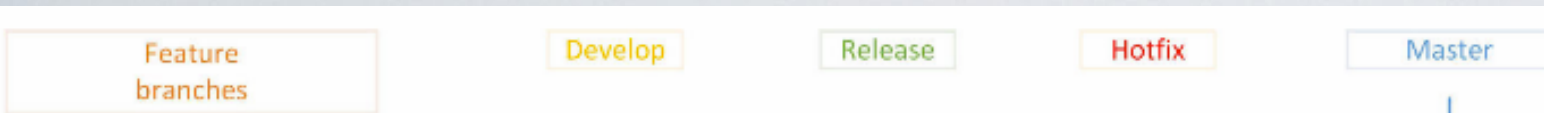


Master - основное направление в

Feature - разработка новых фич

Release - подготовка к релизу





Develop - главная ветвь разработки

Hotfix - используются для исправления критических ошибок в предыдущем релизе

Release - подготовка к выпуску новых версий продукта

Feature branches - разработка новых функций

# Branch Strategies

1. USER by Branch
2. TASK by Branch
3. STORY by Branch
4. SPRINT by Branch
5. RELEASE by Branch



# HOMEWORK