

# Структура научной презентации

Простейший шаблон

---

Прокопьева Марина Евгеньевна

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Прокопьева Марина Евгеньевна
- Студент
- 1132237370
- Российский университет дружбы народов

## Вводная часть

---

## Цель работы

---

Получение навыков правильной работы с репозиториями git.

## Задание

---

Выполнить работу для тестового репозитория.

Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional com



## Теоретическое введение

---

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow.

Gitflow Workflow опубликована и популяризована Винсентом Дриссенем.

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом  
Данная модель отлично подходит для организации рабочего процесса на основе ре

Работа по модели Gitflow включает создание отдельной ветки для исправлений ош

Последовательность действий при работе по модели Gitflow:

- Из ветки master создаётся ветка develop.

- Из ветки develop создаётся ветка release.

- Из ветки develop создаются ветки feature.

- Когда работа над веткой feature завершена, она сливается с веткой develop.

- Когда работа над веткой релиза release завершена, она сливается в ветки de

- Если в master обнаружена проблема, из master создаётся ветка hotfix.

- Когда работа над веткой исправления hotfix завершена, она сливается в ветк

1. Основные ветки (master) и ветки разработки (develop) Для фиксации истории проекта в рамках этого процесса вместо одной ветки master используются две ветки. В ветке master хранится официальная история релиза, а ветка develop предназначена для объединения всех функций. Кроме того, для удобства рекомендуется присваивать всем коммитам в ветке master номер версии.

При использовании библиотеки расширений `git-flow` нужно инициализировать с

```
git flow init
```

Для github параметр `Version tag prefix` следует установить в `v`.

После этого проверьте, на какой ветке Вы находитесь:

```
git branch
```

Семантический подход в версионированию программного обеспечения.

## Выполнение лабораторной работы

---

1. Установка git-flow
2. Установка Node.js
3. Настройка Node.js
4. Общепринятые коммиты

## Выводы

---



Получили навыки правильной работы с репозиториями git