**ИТМО Кафедра Вычислительной техники**

Отчет по лабораторной работе №2

Основы программной инженерии

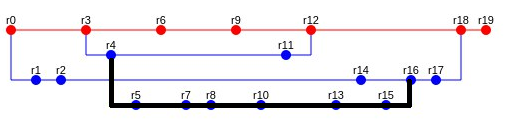
Вариант 190475

**Выполнил: студент группы P3217**

**Плюхин Д.А.**

**2016 год**

1. **Задание к лабораторной работе**

****

Сконфигурировать в своём домашнем каталоге репозитории svn и git и загрузить в них начальную ревизию файлов с исходными кодами (в соответствии с выданным вариантом). Воспроизвести последовательность команд для систем контроля версий svn и git, осуществляющих операции над исходным кодом, приведённые на блок-схеме.

1. **Cписок команд**
   1. **SVN**

cd opilab2

svn checkout https://svn.riouxsvn.com/opilab2svn

cd opilab2svn

svn copy https://svn.riouxsvn.com/opilab2svn/trunk https://svn.riouxsvn.com/opilab2svn/branches/zeio -m "Created new branch zeio"

svn update

cd trunk

svn switch https://svn.riouxsvn.com/opilab2svn/branches/zeio

svn add \*

svn commit -m "r5"

svn add \*

svn commit -m "r7"

svn add \*

svn commit -m "r8"

svn add \*

svn commit -m "r10"

svn add \*

svn commit -m "r13"

svn add \*

svn commit -m "r15**"**

* 1. **GIT**

git clone -b Anastasia https://github.com/Bizzonium/ifmo.opi.lab2/

cd ifmo.opi.lab2

git add -A

git checkout -b zeio

git commit -m "r5"

git push origin zeio

git add -A

git commit -m "r7"

git push origin zeio

git add -A

git commit -m "r8"

git push origin zeio

git add -A

git commit -m "r10"

git push origin zeio

git add -A

git commit -m "r13"

git push origin zeio

git add -A

git commit -m "r15"

git push origin zeio

1. **Вывод**

Так, в результате лабораторной работы были изучены две системы контроля версий – svn и git а также сделаны выводы о том, что git по сравнению с svn является более гибкой системой, хотя и более сложной при изучении. Были сделаны выводы о том, что при использовании систем контроля версий необходимо хотя бы несколько раз в день делать коммиты на удаленный сервер, при появлении задачи, решение которой может помешать работе других программистов над общим проектом и при этом требует большого количества времени, необходимо создавать новую ветку и вести параллельный процесс разработки. Был сделан вывод о том, что системы контроля версий представляют собой очень мощное и эффективное средство для управления процессом разработки программного продукта. Полученные знания являются необходимыми для любого программиста и пригодятся при работе в любой IT-компании.