**2 СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ**

Все системы контроля версий делятся на централизованные и децентрализованные. Соответственно, в централизованных системах контроля версий присутствует центральный сервер, на котором завязаны все остальные системы. В децентрализованных системах такой сервер тоже есть, но он играет только роль синхронизации работы, так как в принципе, копия версии с любого компьютера может стать основной.

Для проведения лабораторного практикума были выбраны обе системы контроля версий. На практике чаще всего используются децентрализованные, как более надёжные и простые в использовании. Тем ни менее, именно с использованием централизованных систем контроля версий на данный момент написано наибольшее количество программных комплексов, так как они исторически являлись первыми.

В БрГТУ.100529-07 91 00 приведены схемы взаимодействия программ.

**2.1 Обзор децентрализованных систем**

С приходом более совершенных IDE и языков программирования, вопрос взаимодействия с коллегами является ключевым для многих разработчиков. В основном чтобы решить эту проблему, начали появляться централизованные системы контроля версий (ЦСВ или CVS – Centralized Version Control systems). Примерами таких систем являются:

− CVS;

− Subversion;

− Perforce.

В них используется следующий метод сохранения информации: имеется один сервер, который хранит все версии файлов, а также некоторое количество клиентов, на которые поставляется информация с сервера. Диаграмма таких взаимоотношений представлена на схеме 2.1:



Схема 2.1 – Диаграмма взаимоотношений в VCS

Такой подход имеет много преимуществ, особенно относительно RCS. Основное из них в том, что разработчики могут работать с одним и тем же файлом одновременно. Кроме этого, есть ещё и другие плюсы, например, каждый разработчик в любой момент времени может знать, что делают все остальные; кроме этого, администраторы систем, имея точно такую же возможность, могут эффективнее управлять своими кадрами и всем продуктом в целом.

Но, несмотря на решение достаточно большого спектра проблем, у таких систем есть свои недостатки. Первый и наиболее существенный – единая точка отказа. Если сервер по каким-то причинам выходит из строя, то все работают только со своими локальными копиями, не имея возможности смотреть изменения других разработчиков.

**2.2 Обзор централизованных систем**

Чтобы решить эту проблему, и, в свою очередь забыть о проблемах RCC, существуют децентрализованные системы контроля версий (ДСКВ или DVCS – Distributed Version Control System). В качестве примеров можно привести следующие ДСКВ:

− Git;

− Mercurial;

− Darcs;

− Bazaar.

Как следует из названия, в таких системах клиенты не просто забирают последнюю (или выбранную, но всегда одну) версию проекта: на самом деле они полностью копируют его содержание и историю. Таким образом, при таком подходе у системы появляются два больших преимущества перед VCS:

В любой момент времени разработчик может посмотреть состояние проекта в какой-то момент времени даже без наличия доступа к центральному серверу;

Каждая копия проекта это, по сути дела, полная резервная копия проекта. Таким образом, если сервер выйдет из строя, то любой разработчик сможет восстановить его состояние на любой другой машине.

Диаграмма взаимоотношений между клиентом и сервером представлена на схеме 2.2:



Схема 2.2 – Децентрализованная система контроля версий

Также при выборе системы контроля версий учитывается популярность того или иного инструмента. С целью определить наиболее используемую систему, в 2014 году был проведён опрос среди разработчиков сайтов под систему WordPress. Результаты представлены на рисунке 2.3:

Рисунок 2.3 – Популярность систем контроля версий для WordPress проектов

Основная причина в использовании git заключается в его популярности. Git является одним из основных инструментов, которые требуют заказчики, а также в том, что именно для работы с git написан самая популярная работа для работы с файлами – GitHub.

Основная причина в использовании subversion заключается в необходимости поддержки уже существующего ПО.