**Лабораторная работа № 4. Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)**

**Цель занятия**: 1. Изучить на практике понятия и компоненты систем контроля версий (СКВ), приемы работы с ними.

**Задание:**

1. Установите TortoiseSVN на компьютере.

2. Создайте новый проект.

3. Создайте локальный репозиторий для своего проекта.

4. Удалите созданный проект на своем компьютере и обновите проект из репозитория.

5. Внесите изменения в файлах с исходными кодами и сохраните изменения в репозитории. Обновите файлы с исходными кодами из репозитория.

6. Внесите изменения в файлах с исходными кодами таким образом, чтобы у двух участников проекта изменения были в одном и том же файле. Попытайтесь сохранить изменения в репозитории. Устраните обнаруженные конфликты версий. Повторно сохраните изменения в репозитории.

7. Создайте отдельную ветку проекта. Внесите изменения в файлы с исходными кодами. Сохраните изменения в репозитории.

8. Объедините созданную на предыдущем шаге ветку с основной веткой проекта.

9. Выведите на экран лог изменений файла, в котором было наибольшее количество изменений.

10. Отобразите на экране сравнение фала до и после внесения одного из изменений.

11. Создайте репозитарий в сети Интернет. Повторите шаги 4 – 6.

**Теоретические сведения:**

Управление версиями - это искусство работы с изменениями информации. Долгое время оно было жизненно важным инструментом программистов, которым необходимо произвести небольшие изменения в программе, или же сделать ―откат‖ изменений, возвращаясь к предыдущей версии. Однако полезность систем управления версиями выходит далеко за пределы мира разработчиков программного обеспечения. Управление версиями требуется повсюду, где можно встретить людей, использующих компьютер для работы с постоянно изменяющейся информацией

**Software Configuration Management** или **Конфигурационное управление** подразумевает под собой комплекс методов, направленных на то, чтобы систематизировать изменения, вносимые разработчиками в [программный](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) продукт в процессе его разработки и [сопровождения](http://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%BE%D0%BF%D1%80%D0%BE%D0%B2%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5&amp;action=edit&amp;redlink=1), сохранить целостность системы после изменений, предотвратить нежелательные и непредсказуемые эффекты, а также сделать процесс внесения изменений более [формальным](http://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D0%B8%D0%B7%D0%BC). Изначально управление [конфигурацией](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D1%8F) применялось не в [программировании,](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) но в связи с высокой динамичностью сферы разработки [ПО,](http://ru.wikipedia.org/wiki/%D0%9F%D0%9E) в ней она особенно полезна. К процедурам можно отнести создание резервных копий, [контроль исходного кода](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8), [требований](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9_%28%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%9F%D0%9E%29) проекта, [документации](http://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F_%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и т. д. Степень формальности выполнения данных процедур зависит от размеров проекта, и при правильной ее оценке данная концепция может быть очень полезна. Конфигурационное управление требует выполнения множества трудоемких рутинных операций. На практике, в большинстве случаев, для конфигурационного управления применяются специальные системы контроля версий исходного кода программ. В качестве примера такой системы рассмотрим самую распространенную на сегодняшний день – Subversion.

Subversion — это бесплатная система управления версиями с открытым исходным кодом. Subversion позволяет управлять файлами и каталогами, а так же сделанными в них изменениями во времени. Это позволяет восстановить более ранние версии данных, даѐт возможность изучить историю всех изменений. Благодаря этому многие считают систему управления версиями своего рода «машиной времени».

Схема общей архитектуры Subversion. В нижней части схемы изображено хранилище Subversion, в котором хранится информация с версиями. В верхней части схемы показана программа-клиент Subversion, которая управляет локальными отражениями различных фрагментов этих данных (также называемыми «рабочими копиями»). Между этими сторонами проложены различные маршруты, проходящие через разные слои доступак хранилищу. Некоторые из этих маршрутов используют компьютерные сети и сетевые сервера, чтобы достичь хранилища, в то время как другие маршруты в сети не нуждаются и ведут к хранилищу напрямую.

Subversion может работать через сеть, что позволяет использовать еѐ на разных компьютерах. В какой то степени, возможность большого количества людей не зависимо от их местоположения совместно работать над единым комплектом данных поощряет сотрудничество. Когда нет того ответственного звена цепи, того контролирующего элемента, который утверждает все изменения, работа становится более эффективной. При этом не нужно опасаться, что отказ от контролирующего элемента повлияет на качество, ведь благодаря сохранению истории изменений, даже если при изменении данных будут допущены ошибки, всегда можно сделать откат изменений к прежнему состоянию.

Некоторые системы управления версиями выступают также в качестве систем управления конфигурацией программного обеспечения. Такие системы специально созданы для управления деревьями исходного кода и имеют множество особенностей, непосредственно относящихся к разработке программ: они понимают языки программирования и предоставляют инструменты для сборки программ. Subversion не является такой системой, она представляет собой систему общего назначения, которую можно использовать для управления *любым* набором файлов. Для Вас это будут исходники Ваших программ, а для кого-то другого это будет список продуктов или сведѐнное цифровое видео.

# TortoiseSVN

TortoiseSVN — это бесплатный Windows-клиент с открытыми исходным кодом для системы управления версиями *Apache™ Subversion®*. То есть TortoiseSVN управляет файлами и директориями во времени. Файлы хранятся в центральном *хранилище*. Хранилище больше похоже на обычный файловый сервер, кроме того он запоминает каждое изменение когда-либо сделанное в ваших файлах и директориях. Это позволяет вам восстановить старые версии ваших файлов и проверить историю изменений — как, когда и кто изменял ваши данные. Вот почему многие думают о Subversion, и вообще о системах управления версиями, как своего рода «машине времени».

Некоторые системы контроля версий являются также и системами управления конфигурацией программ (software configuration management - SCM). Такие системы специально созданы для управления деревьями исходного кода, и имеют множество возможностей, специфичных для разработки программ, таких как непосредственное понимание языков программирования, или предоставление инструментов для сборки программ. Однако Subversion не является такой системой, она является системой общего назначения, которая может быть использована для управления *любым* набором файлов, включая и исходные коды программ.

# Возможности TortoiseSVN

Что делает TortoiseSVN таким хорошим клиентом Subversion? Вот краткий список возможностей:

*Интеграция с оболочкой*

TortoiseSVN интегрируется непосредственно в оболочку Windows (т.е. в Проводник). Это значит, что вы можете работать с уже знакомыми инструментами, и вам не надо переключаться на другое приложение каждый раз, когда вам необходимы функции для управления версиями!

И вам даже не обязательно использовать именно Проводник. Контекстные меню TortoiseSVN работают во многих других файловых менеджерах, и в диалогах для открытия файлов, используемых в большинстве стандартных Windows-приложений. Однако вы должны учитывать, что TortoiseSVN изначально разработан как расширение для Проводника Windows, и, возможно, в других приложениях интеграция будет не полной, например, могут не отображаться пометки на значках.

*Пометки на значках*

Статус каждого версированного файла и папки отображается при помощи маленькой пометки поверх основного значка. Таким образом, вы сразу можете видеть состояние вашей рабочей копии.

*Графический интерфейс пользователя*

При просмотре списка изменений файла или папки вы можете кликнуть на ревизию, чтобы увидеть комментарии для этой фиксации. Также доступен список измененных файлов - всего лишь сделайте двойной клик на файле, чтобы увидеть, какие конкретно изменений были внесены.

Диалог фиксации это список, в котором перечислены все файлы и папки, которые будут включены в фиксацию. У каждого элемента списка имеется флажок, чтобы вы могли выбрать именно то, что вы хотите включить в фиксацию. Неверсированные файлы также могут быть представлены в этом списке, чтобы вы не забыли добавить в фиксацию новый файл или папку.

*Простой доступ к командам Subversion*

Все команды Subversion доступны из контекстного меню Проводника. TortoiseSVN добавляет туда собственное подменю.

Поскольку TortoiseSVN является клиентом Subversion, мы хотели бы показать и некоторые из возможностей самой Subversion:

*Версирование папок*

CVS отслеживает только историю отдельных файлов, тогда как Subversion реализует «виртуальную» версионную файловую систему, которая отслеживает изменения в целых деревьях папок во времени. Файлы *и* папки являются версированными. В результате, есть команды **переместить** и **копировать**, реально выполняемые на стороне клиента и работающие непосредственно с файлами и папками.

*Атомарные фиксации*

Фиксация сохраняется в хранилище либо полностью, либо не сохраняется вообще. Это позволяет разработчикам фиксировать изменения, собранные в логически связанные части.

*Версированные метаданные*

Каждый файл и папка имеет прикреплѐнный невидимый набор «свойств». Вы можете создавать и сохранять произвольные пары ключ/значение для собственных нужд. Свойства тоже версируются во времени, как и содержимое файла.

*Возможность выбора сетевого уровня*

В Subversion есть абстрагируемое понятие доступа к хранилищу, которое упрощает реализацию новых сетевых механизмов. «Усовершенствованный» сетевой сервер Subversion является модулем для веб-сервера Apache, который использует для взаимодействия диалект HTTP под названием WebDAV/DeltaV. Это даѐт Subversion большие преимущества в стабильности и совместимости, и предоставляет различные ключевые возможности без дополнительных затрат: проверка личности (аутентификация), проверка прав доступа (авторизация), сжатие потока данных при передаче, просмотр хранилища. Также доступна меньшая, автономная версия сервера Subversion, взаимодействующая по собственному протоколу, который легко может быть туннелирован через ssh.

*Единый способ обработки данных*

Subversion получает различия между файлами при помощи бинарного разностного алгоритма, который работает одинаково как с текстовыми (читаемыми человеком), так и с бинарными (не читаемыми человеком) файлами. Оба типа файлов содержатся в хранилище в сжатом виде, а различия передаются по сети в обоих направлениях.

*Эффективные ветки и метки*

Стоимость создания веток и меток не обязательно должна быть пропорциональна размеру проекта. Subversion создаѐт ветки и метки, просто копируя проект с использованием механизма, похожего на жѐсткие ссылки в файловых системах. Благодаря этому, операции по созданию веток и меток происходят за одинаковое, очень малое время и занимают очень мало места в хранилище.

**Выполнение работы:**

# Установка

TortoiseSVN поставляется в виде простого в использовании установочного файла (на странице загрузки <http://tortoisesvn.net/downloads.html> необходимо выбрать дистрибутив, соответствующий разрядности системы). Сделайте на нем двойной клик и следуйте инструкциям - остальное он сделает за вас. Также на странице загрузки необходимо скачать языковой пакет и установить после установки основной программы. Не забудьте перезагрузить компьютер после установки.

# Основная концепция

Перед тем, как мы погрузимся в работу с настоящими файлами, важно получить представление о том, как работает Subversion и какие термины используются.

Хранилище

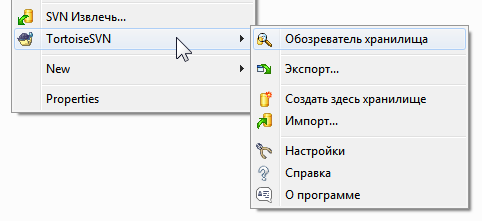
Subversion использует центральную базу данных, которая содержит все ваши версированные файлы с их полной историей. Эта база данных называется *хранилищем*. Хранилище обычно находится на файловом сервере, на котором установлен Subversion, по запросу поставляющий данные клиентам Subversion (например, TortoiseSVN). Если вы делаете резервное копирование, то копируйте ваше хранилище, так как это оригинал всех ваших данных.

Рабочая копия

Это именно то место, где вы работаете. Каждый разработчик имеет собственную рабочую копию, иногда называемую песочницей, на своем локальном компьютере. Вы можете получить из хранилища последнюю версию файлов, поработать над ней локально, никак не взаимодействуя с кем- либо еще, а когда вы будете уверены в изменениях вы можете зафиксировать эти файлы обратно в хранилище.

Рабочая копия не содержит историю проекта, но содержит копию всех файлов, которые были в хранилище до того, как вы начали делать изменения. Это обозначает, что можно легко узнать, какие конкретно изменения вы сделали.

Вам также нужно знать, где найти TortoiseSVN, потому что в меню "Пуск" его нет. Это так потому, что TortoiseSVN - расширение проводника Windows, поэтому для начала нужно запустить проводник. Сделайте правый клик на папке в проводнике и вы увидите новые пункты в контекстном меню, такие как эти:



# Рисунок 1. Меню TortoiseSVN для неверсированных папок

Создание хранилища

Для настоящего проекта вам понадобится хранилище, созданное в безопасном месте, и сервер Subversion, чтобы управлять им. Для обучения мы будем использовать функцию локального хранилища Subversion, которая разрешает прямой доступ к хранилищу, созданного на вашем жестком диске, и не требует наличия сервера.

Сначала создайте новую пустую директорию на вашем ПК.

Она может быть где угодно, но в этом руководстве мы собираемся назвать еѐ C:\svn\_repos. Теперь сделайте правый клик на новой папке и в контекстном меню выберите **TortoiseSVN** → **Создать здесь**  **хранилище...**. Хранилище, созданное внутри папки, готово к использованию. Также мы создадим внутренную структуру папок нажав кнопку.

**Создать структуру каталогов**

# Импорт проекта

Сейчас у нас есть хранилище, но оно совершенно пустое в данный момент. Давайте предположим, что у меня есть набор файлов вC:\Projects\Widget1, который я хотел бы добавить. Перейдите к папке Widget1 в Проводнике и сделайте правый клик на ней. Теперь выберите пункт **TortoiseSVN** → **Импорт...**, который вызовет диалог

# Рисунок 2. Диалог импорта

К хранилищу Subversion обращаются по URL-адресу, который позволяет нам указать хранилище где угодно в Интернете. В данном случае нам нужно указать на наше локальное хранилище, которое имеет URL-адрес file:///c:/svn\_repos/trunk, и к которому мы добавляем имя нашего проекта Widget1. Обратите внимание, что после file: есть 3 слэша и везде используются прямые слэши.

Другая важная функция данного диалога - это окно Сообщение импорта, в которое вы можете добавить сообщение о том, что вы делаете. Когда вам понадобится просмотреть историю проекта, эти сообщения будут ценным подспорьем для просмотра какие изменения и когда были сделаны. В нашем случае мы напишем что-нибудь простое

как «Импорт проекта Виджет1». Нажмите , чтобы добавить папку в ваше хранилище.

**OK**

Извлечение рабочей копии

Сейчас у нас есть проект в нашем хранилище, и нам надо создать рабочую копию для повседневной работы. Заметьте, что имортирование папки не превращает автоматически эту папку в рабочую копию. Для создания свежей рабочей копии в Subversion используется термин Извлечь. Мы собираемся извлечь папку Widget1 из нашего хранилища в папку для разработки называемуюC:\Projects\Widget1-Dev. Создайте эту папку, затем сделате правый клик на ней и выберите пункт **TortoiseSVN** → **Извлечь...**. Введите URL-адрес для извлечение, в данном случае file:///c:/svn\_repos/trunk/Widget1, и

кликните на . Наша папка для разработки заполнится файлами из хранилища.

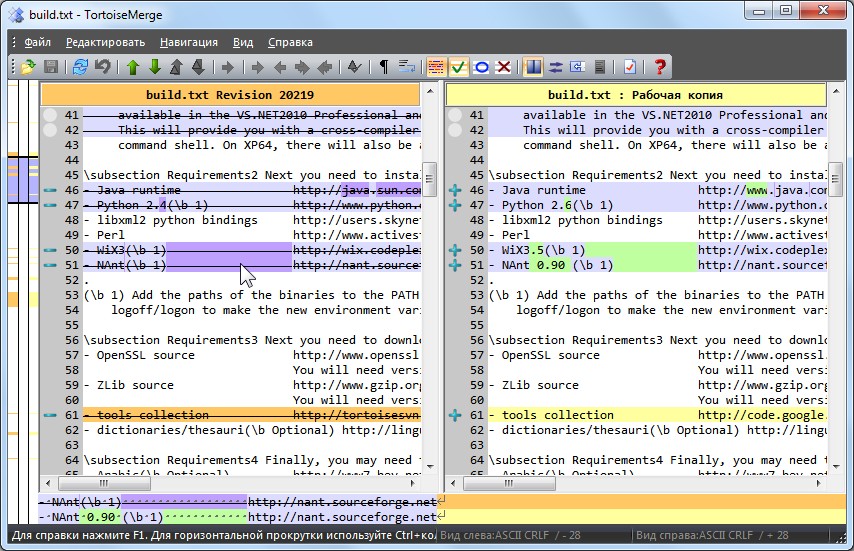
**OK**

Вы заметите что внешний вид этой папки отличается от обычной папки. У каждого файла появился зелѐный флажок в левом углу. Это значки статуса TortoiseSVN, которые присутствуют только в рабочей копии. Зелѐный статус означает, что файл не отличается от версии файла, находящегося в хранилище.

# Внесение изменений

Можно приступать к работе. В папке Виджет1-Дев мы начинаем изменять файлы - предположим, мы вносим изменения в файлыВиджет1.c и ПрочтиМеня.txt. Обратите внимание, что значки на этих файлах теперь стали красными и показывают, что изменения были сделаны локально.

Но какие были изменения? Нажмите правой кнопкой на одном из изменѐнных файлов и выберите команду **TortoiseSVN** → **Различия**. Запустится инструмент TortoiseSVN для сравнения файлов и покажет какие точно строки в файлах были изменены.



# Рисунок 3. Просмотрщик изменений в файлах

Ок, нас устраивают изменения, поэтому давайте обновим хранилище. Это действие называется Фиксировать изменения. Нажмите правой кнопкой на папке Виджет1-Дев и выберите команду **TortoiseSVN** → **Фиксировать**. Появится диалог фиксации со списком изменѐнных файлов и напротив каждого будет галочка. Вы можете выбрать лишь несколько файлов из списка для фиксации, но в нашем случае мы будем фиксировать изменения в обоих файлах. Введите сообщение с описанием сделанных изменений и нажмите ОК. Появится диалог с прогрессом процесса фиксации файлов в хранилище и мы закончили фиксацию.

**OK**

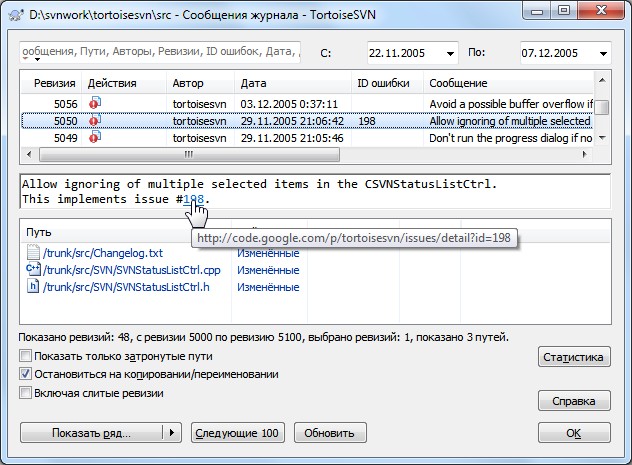
# Добавление новых файлов

Во время работы над проектом, вам понадобится добавлять новые файлы - предположим вы добавили новые функции в файлеЭкстра.c и добавили справку в существующем файле Создатьфайл. Нажмите правой кнопкой на папке и выберите команду**TortoiseSVN** → **Добавить**. Диалог добавления показывает все неверсированные файлы и вы можете выбрать те файлы, которые вы хотите добавить. Другой способ добавления файлов - это нажать правой кнопкой на самом файле и выбрать команду**TortoiseSVN** → **Добавить**.

Теперь, если вы откроете папку для фиксации, новый файл будет отображаться как *Добавлен* и существующий файл как *Изменѐн*. Обратите внимание, что вы можете дважды нажать на изменѐнный файл, чтобы просмотреть какие именно изменения были сделаны.

# Просмотр истории проекта

Одной из самых полезных функций TortoiseSVN является диалоговое окно журнала. Оно показывает список всех фиксаций изменений в файле или папке, а также все те детальные сообщения, которые вы вводили при фиксации изменений ;-)



# Рисунок 4. Диалоговое окно журнала

Верхняя панель показывает список всех фиксированных ревизий вместе с началом сообщения фиксации. Если вы выберите одну из этих ревизий, то средняя панель отобразит полное сообщение журнала для той ревизии и нижняя панель покажет список измененных файлов и папок.

У каждой из этих панелей есть контекстное меню, которое предоставляет много других способов использования информации. В нижней панели вы можете **дважды нажать** на файл, чтобы просмотреть какие именно изменения были внесены в той ревизии. Прочтите [«Диалоговое окно журнала ревизий»](http://tortoisesvn.net/docs/release/TortoiseSVN_ru/tsvn-dug-showlog.html), чтобы узнать больше.

# Отмена изменений

Одной общей функцией всех систем управления ревизиями является функция, которая позволяет вам отменить изменения, которые вы внесли ранее. Как вы и догадались, в TortoiseSVN это легко сделать.

Если вы хотите избавиться от изменений, которые вы еще не успели фиксировать и восстановить нужный файл в том виде, в котором он был перед началом изменений, то выберите команду **TortoiseSVN** → **Убрать изменения**. Это действие отменит ваши изменения (в Корзину) и вернет фиксированную версию файла, с которой вы начинали. Если же вы хотите убрать лишь некоторых изменения, то вы можете использовать инструмент TortoiseMerge для просмотра изменений и выборочного удаления измененных строк.

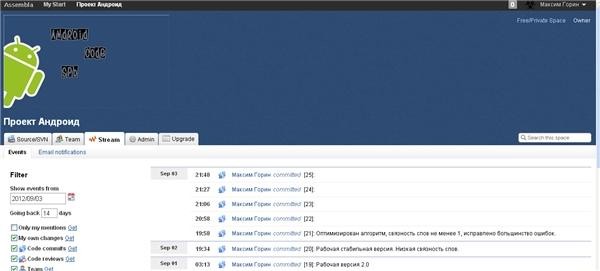
Если вы хотите отменить действия определѐнной ревизии, то начните с диалогового окна журнала и найдите проблемную ревизию. Выберите команду **Контекстное**  **меню** → **Отменить изменения из этой ревизии** и те изменения будут отменены.

# Работа с сетью.

С Subversion можно работать как посредством сети интернет, так и локально. Воспользуемся сервисом [Assembla](https://www.assembla.com/) (https://[www.assembla.com/).](http://www.assembla.com/)) Зарегистрировавшись там, вы получите 1 Gb места под репозиторий. Создав его и настроив вы получите ссылку вида https://subversion.assembla.com/svn/название\_репозитория, которую можно использовать в любом SVN клиенте. К примеру, чтобы в Visual SVN добавить свой проект в репозиторий, вам нужно нажать Add Solution to Subversion, после чего указать локальное хранилище вашего проекта, нажать Далее и ввести вашу ссылку. Все, теперь сверху появится панель с основными SVN-функциями (Show Log, Update, Commit, Switch Branch, Branch и Merge) и можно приступать к полноценной работе.



В общем и целом, сайт позволяет так же настроить репозиторий и получить на неделю или две пробный примиум (платные доп. функции, источник жизни сайта). Основные функции же полностью бесплатны. Чтобы товарищи по команде могли работать с общим для команды репозиторием, на assembla.com, необходима их регистрация на сайте. После регистрации, пользователи могут быть добавлены владельце репозитория в список команды. На самом сайте можно посмотреть всю информацию о проекте и изменениях, и даже поставить свой баннер с ссылкой.



Так, например, выглядит страница Stream, где отображаются последние изменения.

**Требования к отчету:** Текст должен быть написан шрифтом Times New Roman, 12. Интервал между строками и абзацами – 1,5. Отступ слева 1,5. Ориентация текста – по ширине страницы. Скриншоты необходимо подписать. Название практической работы, цель работы, ход работы, вывод, ответы на контрольные вопросы, должны быть выделены жирным шрифтом, так же как в методичке.

**Контрольные вопросы:**

1. Что такое системы контроля версий (СКВ) и для решения каких задач они

предназначаются?

2. Объясните следующие понятия СКВ и их отношения: хранилище, commit, история, рабочая копия.

3. Что представляют собой и чем отличаются централизованные и децентрализованные СКВ? Приведите примеры СКВ каждого вида.

4. Опишите действия с СКВ при единоличной работе с хранилищем.

5. Опишите порядок работы с общим хранилищем в централизованной СКВ.

6. Что такое и зачем может быть нужна разность (diff)?

7. Что такое и зачем может быть нужно слияние (merge)?

8. Что такое конфликты (conflict) и каков процесс их разрешения (resolve)?

9. Поясните процесс синхронизации с общим хранилищем («обновления»)

в децентрализованной СКВ.

10. Что такое и зачем могут быть нужны ветви (branches)?

11. Объясните смысл действия rebase в СКВ Git.

12. Как и зачем можно игнорировать некоторые файлы при commit?