

Информация для размещения на официальном сайте ГБПОУ  
«Светлоградский региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	227
Дата	21.02.2025
Время	9-50-11-10
Наименование УД/МДК/УП/ПП	МДК 01.01 «Разработка программных модулей»
Ф.И.О. преподавателя	Сахарчук Т.В.
Электронная почта	<a href="mailto:saharchyk777@mail.ru">saharchyk777@mail.ru</a>
Основная литература	<p>1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ: УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ. Учебник и практикум для СПО Зараменских Е. П. Национальный исследовательский университет «Высшая школа экономики» (г. Москва) Финансовый университет при Правительстве РФ (г. Москва). Год: 2023 / Гриф УМО СПО</p> <p>2. <a href="https://biblio-online.ru/adv-search/get?scientific_school=68982D81-E102-419D-86AD-AD85638B9282">https://biblio-online.ru/adv-search/get?scientific_school=68982D81-E102-419D-86AD-AD85638B9282</a></p> <p>3. Рудаков А.В. Технология разработки программных продуктов (11-е изд., стер.) учебник «Академия» 2023г.</p> <p>4. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем (2-е изд., стер.) учебник «академия»2024г.</p> <p>5. Рудаков А.В. Технология разработки программных продуктов (12-е изд.) учебник «Академия»2023г.</p> <p>6. Федорова Г.Н. Сопровождение информационных систем (1-е изд.) учебник «Академия»2023г.</p>
Тема № 17-18	Конспект лекции на тему: Системы контроля версий: виды, принципы организации работы
Задание	<p><b>Часть 1: Введение в системы контроля версий</b></p> <p><b>1. Что такое система контроля версий (СКВ)?</b></p> <ul style="list-style-type: none"> <li>- <b>Определение и принцип работы СКВ.</b></li> <li>- <b>Преимущества использования СКВ.</b></li> </ul> <p>Система контроля версий (СКВ) является важным инструментом для эффективной работы с файлами и управления изменениями. Она позволяет разработчикам отслеживать все изменения файлов и возвращаться к предыдущим версиям при необходимости.</p> <p>Определение СКВ заключается в том, что это программа или сервис, предоставляющий возможность сохранения и управления версиями файлов. Она записывает все изменения, происходящие с файлами, и сохраняет их в специальном хранилище, известном как репозиторий. Разработчики могут выполнять операции добавления, удаления, изменения и перемещения файлов в репозитории, а также отслеживать и комментировать каждую измененную строку кода.</p> <p>Принцип работы СКВ основан на использовании репозитория, в котором хранятся все файлы и история их изменений. При каждом</p>

изменении файла в репозитории создается новая версия этого файла. Это позволяет в любой момент вернуться к предыдущей версии файла или сравнить изменения между разными версиями. СКВ также предоставляет функционал для совместной работы разработчиков над проектом и решения конфликтов при одновременном редактировании одного файла.

Преимущества использования СКВ в разработке программного обеспечения и управлении проектами являются значительными:

1. История изменений: СКВ позволяет отследить все изменения, сделанные в файловой системе. Это полезно для контроля качества кода, анализа проблем, поиска ошибок и восстановления данных. Например, если в новой версии проекта появилась ошибка, разработчик может легко вернуться к предыдущей версии кода и исправить проблему.

2. Обратимость изменений: Возможность вернуться к предыдущим версиям файлов является важной особенностью СКВ. Например, если разработчик внес изменения, которые повлияли на работоспособность программы, он может легко откатиться к стабильной версии и продолжить работу.

3. Коллаборация: СКВ содействует совместной работе разработчиков над проектом. Каждый может работать над своей задачей или веткой проекта, а СКВ автоматически объединяет изменения и решает конфликты. Например, пусть есть команда разработчиков, работающая над усовершенствованием программы. Каждый разработчик может создать свою собственную ветку в репозитории СКВ, вносить изменения и затем объединять их с остальными членами команды.

4. Ветвление и слияние: СКВ позволяет создавать ветки разработки для индивидуальной работы над определенной функциональностью или исправлением ошибок. После того, как задачи решены, ветки могут быть объединены с главной веткой разработки. Например, разработчики могут создать ветку для исправления ошибки и работать над ней независимо. По завершении работы, изменения могут быть объединены с основной веткой, чтобы внести исправления в основной код программы.

5. Безопасность данных: СКВ обеспечивает сохранность данных и предоставляет возможность их восстановления в случае потери или повреждения. Репозиторий в СКВ является надежным хранилищем, где файлы могут быть защищены от непредвиденных событий. Например, если компьютер разработчика выходит из строя, данные можно легко восстановить из репозитория на новом компьютере.

СКВ является мощным инструментом для работы с файлами, отслеживания изменений и совместной работы разработчиков. Его использование обеспечивает контроль над проектом, улучшает процесс разработки и повышает эффективность работы команды разработчиков.

## **2. Типы систем контроля версий**

- **Локальные СКВ (преимущества и недостатки).**

- **Централизованные СКВ (преимущества и недостатки).**

- **Распределенные СКВ (преимущества и недостатки).**

Существует несколько типов систем контроля версий (СКВ), каждый из которых имеет свои преимущества и недостатки. В этой лекции мы рассмотрим три основных типа СКВ: локальные, централизованные и распределенные.

#### 1. Локальные СКВ:

Локальные СКВ подразумевают, что все файлы и история их изменений хранятся на локальном компьютере разработчика. Примером локальной СКВ является система RCS (Revision Control System).

##### Преимущества:

- Простота в настройке и использовании. Такие системы подходят для небольших проектов или индивидуальной разработки.
- Высокая скорость работы, поскольку все операции выполняются локально.

##### Недостатки:

- Отсутствие возможности совместной работы над проектом с другими разработчиками.
- Отсутствие централизованного хранилища, что делает восстановление данных после сбоев или потери локального компьютера сложным.

#### 2. Централизованные СКВ:

Централизованные СКВ предполагают наличие центрального сервера, на котором хранится вся история изменений файлов. Примером централизованной СКВ является система SVN (Subversion).

##### Преимущества:

- Возможность совместной работы нескольких разработчиков над одним проектом.
- Централизованное хранение данных, что позволяет легко восстановить прошлые версии файлов.

##### Недостатки:

- Отсутствие возможности работать в офлайн-режиме или без доступа к серверу.
- Зависимость от центрального сервера, что может спровоцировать проблемы при его недоступности или сбоях.

#### 3. Распределенные СКВ:

Распределенные СКВ предполагают наличие нескольких копий репозитория, где каждый разработчик имеет полную копию истории изменений. Примером распределенной СКВ является система Git.

##### Преимущества:

- Возможность совместной работы над проектом и внесения изменений без доступа к центральному серверу.
- Высокая скорость работы, так как все операции выполняются локально.

##### Недостатки:

- Сложность в настройке и освоении системы для начинающих разработчиков.
- Возможность возникновения конфликтов при слиянии изменений из разных копий репозитория.

Каждый тип СКВ имеет свои особенности, и выбор определенного типа зависит от требований и потребностей конкретного проекта или команды разработчиков.

### 3. Популярные системы контроля версий

- *Git (основные понятия и команды).*
- *Subversion (основные понятия и команды).*
- *Mercurial (основные понятия и команды).*

Существует множество систем контроля версий (СКВ), но в этой лекции мы рассмотрим три из наиболее популярных систем: Git, Subversion (SVN) и Mercurial (Hg). У каждой из этих систем есть свои особенности, понятия и команды.

#### 1. Git:

Git — распределенная система контроля версий, разработанная Линусом Торвальдсом. Она широко используется в индустрии и в open source-проектах.

Основные понятия:

- Репозиторий Git: хранилище всех версий файлов и истории их изменений.
- Коммит: сохранение текущего состояния файлов в репозиторий.
- Ветка (branch): параллельная линия разработки с отдельной историей коммитов.
- Слияние (merge): объединение изменений из одной ветки в другую.
- Вилка (fork): создание независимой копии репозитория с возможностью внесения собственных изменений.

Основные команды:

- git init: создание нового репозитория Git.
- git clone: копирование существующего репозитория на локальный компьютер.
- git add: добавление файлов в индекс для последующего коммита.
- git commit: сохранение изменений в репозитории.
- git push: отправка изменений на удаленный сервер.
- git pull: получение последних изменений с удаленного сервера.

#### 2. Subversion (SVN):

Subversion (SVN) — централизованная система контроля версий, позволяющая разработчикам работать с общим репозиторием.

Основные понятия:

- Репозиторий SVN: централизованное хранилище файлов и истории изменений.
- Ревизия (revision): определенная версия репозитория на определенный момент времени.
- Ветвь (branch): создание параллельной линии разработки с отдельной историей коммитов.
- Слияние (merge): объединение изменений из одной ветки в другую.

Основные команды:

	<ul style="list-style-type: none"> <li>- svn checkout: создание рабочей копии репозитория на локальном компьютере.</li> <li>- svn update: получение последних изменений из репозитория.</li> <li>- svn add: добавление файлов в репозиторий.</li> <li>- svn commit: сохранение изменений в репозитории.</li> <li>- svn merge: объединение изменений из разных веток.</li> </ul> <p><b>3. Mercurial (Hg):</b> Mercurial (Hg) — распределенная система контроля версий, поддерживающая как малые, так и большие проекты.</p> <p>Основные понятия:</p> <ul style="list-style-type: none"> <li>- Репозиторий Hg: распределенное хранилище файлов и истории изменений.</li> <li>- Коммит: сохранение текущего состояния файлов в репозитории.</li> <li>- Ветка (branch): создание отдельной линии разработки с собственной историей коммитов.</li> <li>- Обновление (update): переход на другую ветку или ревизию.</li> <li>- Слияние (merge): объединение изменений из разных веток.</li> </ul> <p>Основные команды:</p> <ul style="list-style-type: none"> <li>- hg init: создание нового репозитория Hg.</li> <li>- hg clone: клонирование существующего репозитория на локальный компьютер.</li> <li>- hg add: добавление файлов в репозиторий.</li> <li>- hg commit: сохранение изменений в репозитории.</li> <li>- hg push: отправка изменений на удаленный сервер.</li> <li>- hg pull: получение последних изменений с удаленного сервера.</li> </ul>
Контрольный тест	<p>В ходе лекции мы ознакомились с понятием систем контроля версий, изучили основные типы СКВ, а также познакомились с некоторыми популярными системами контроля версий.</p> <p>Git, Subversion и Mercurial — три популярные системы контроля версий с разными особенностями и командами. Охарактеризуйте их и сделайте конспект.</p>

Дата 21.02.2025

Подпись

Ф.И.О. преподавателя

Информация для размещения на официальном сайте ГБПОУ «Светлоградский  
региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	227
Дата	28.02.2025
Время	9-50-11-10
Наименование УД/МДК/УП/ПП	МДК 01.01 «Разработка программных модулей»
Ф.И.О. преподавателя	Сахарчук Т.В.
Электронная почта	<a href="mailto:saharchyk777@mail.ru">saharchyk777@mail.ru</a>
Основная литература	<p>1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ: УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ. Учебник и практикум для СПО Зараменских Е. П. Национальный исследовательский университет «Высшая школа экономики» (г. Москва) Финансовый университет при Правительстве РФ (г. Москва). Год: 2023 / Гриф УМО СПО</p> <p>2. <a href="https://biblio-online.ru/adv-search/get?scientific_school=68982D81-E102-419D-86AD-AD85638B9282">https://biblio-online.ru/adv-search/get?scientific_school=68982D81-E102-419D-86AD-AD85638B9282</a></p> <p>3. Рудаков А.В. Технология разработки программных продуктов (11-е изд., стер.) учебник «Академия» 2023г.</p> <p>4. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем (2-е изд., стер.) учебник «академия»2024г.</p> <p>5. Рудаков А.В. Технология разработки программных продуктов (12-е изд.) учебник «Академия»2023г.</p> <p>6. Федорова Г.Н. Сопровождение информационных систем (1-е изд.) учебник «Академия»2023г.</p>
Тема № 19-20	Конспект лекции на тему: Системы контроля версий: виды, принципы организации работы
Задание	<p>Принцип работы системы контроля версий</p> <p>Пожалуй, многие представляют себе, каким образом вносятся изменения в тот или иной проект в процессе работы. То есть, тут стоит задача не потерять существующую работоспособную версию проекта. Для этого обычно создается новая папка (чаще всего ее так и называют «Новая папка», дополняя имя, к примеру, датой или парой символов), в которую вы копируете имеющуюся версию и все дальнейшие изменения производите уже именно с ней.</p> <p>Подобных папок может набраться довольно много. Из-за этого в какой-то момент усложняется откат к предыдущим версиям, труднее становится контролировать и упорядочивать вносимые изменения и т.д. А если проект ведут несколько специалистов, картина усугубляется.</p> <p>Системы контроля версий (СКВ или VCS) разработаны специально для того, чтобы максимально упростить и упорядочить работу над проектом (вне зависимости от того, сколько человек в этом участвуют). СКВ дает возможность видеть, кто, когда и какие изменения вносил; позволяет формировать новые ветви проекта, объединять уже имеющиеся; настраивать контроль доступа к проекту; осуществлять откат до</p>

предыдущих версий.

#### Задачи для системы контроля версий

В целом задачи для системы контроля версий уже описаны выше, но можно пояснить еще, для тех, кто сталкивается с термином впервые. Для наглядности представьте, что работа над проектом – это игра, которую нужно пройти, двигаясь от одной контрольной точки – к другой (это и будут промежуточные версии проекта).

Работа в системе контроля версий дает возможность сохранять промежуточные варианты проекта, плюс это еще и помогает справляться с проблемами в случае их возникновения. Когда, например, несколько программистов, работающих над одной опцией, одновременно сбросят в репозиторий созданные версии, то система в автоматическом режиме займется устранением конфликта.

Вот перечень задач, выполняемых системой контроля версий файлов:

~ Сохранение исходного кода. Информация сваливается на удаленный сервер и в репозитории остаются даже файлы, удаленные с компьютера разработчика.

~ Возможность привлекать группу программистов, не покупая отдельно специальные инструменты для командной работы. Каждый свою задачу на персональном компьютере, обновляя файлы, когда это нужно.

~ Отмена внесенных изменений. Всегда есть возможность вернуться к контрольной точке, провести ревью исходного кода и текущего, а затем обновить основную ветку.

~ Распределенная работа над проектом. То есть, программисты могут создавать видоизмененный плагин, пока основная его версия спокойно функционирует на сайте.

#### Распространённые системы управления версиями

- ~ Subversion
- ~ Darcs
- ~ Microsoft Visual SourceSafe
- ~ Bazaar
- ~ Rational ClearCase
- ~ Perforce
- ~ BitKeeper
- ~ Mercurial
- ~ Git
- ~ GNU Arch
- ~ CVS — устаревшая. Потомок: Subversion
- ~ RCS — устаревшая. Потомок: CVS

#### DVCS

##### Распределённые системы управления версиями

Такие системы используют распределённую модель вместо традиционной клиент-серверной. Они, в общем случае, не нуждаются в централизованном хранилище: вся история изменения документов хранится на каждом компьютере, в локальном хранилище, и при необходимости отдельные фрагменты истории локального хранилища синхронизируются с аналогичным хранилищем на другом компьютере. В некоторых таких системах локальное хранилище располагается непосредственно в каталогах рабочей копии.

##### Принципы работы с VCS

###### Для нового проекта

- ~ Определить название репозитория и создать его
- ~ Создать структуру каталогов
- ~ Создать начальные файлы
- ~ Добавить все файлы и каталоги в репозиторий

###### Для существующего проекта

	~ ~ Выбрать место на диске для работы с проектом Подключиться к репозиторию и выгрузить проект на локальный компьютер Во время работы ~ ~ Добавлять в репозиторий только работающие артефакты При фиксации изменений кратко описывать их суть ~ ~ Если VCS позволяет то группы файлов и каталогов заносить совместно
Контрольный тест	Изучите лекцию и сделайте конспект.

Дата 28.02.2025

Подпись

Ф.И.О. преподавателя