**6. Системы контроля версий** – программное обеспечение для облегчения работы с изменяющейся информацией.

Такие системы позволяют:

- хранить несколько версий одного и тоже докумемнта;

-возвращаться к более ранним версиям

- определять, кто и когда сделал то или иное изменение

Некоторые особенности систем контроля версий в контексте разработки программного обеспечения:

- Отслеживание конфликтов. Если два человека изменяют один и тот же файл, один из них может случайно отменить изменения, сделанные другим. Системы контроля версий отслеживают такие конфликты и предлагают средства их решения.

-Автоматическое объединение изменений. Большинство систем может автоматически объединить изменения, сделанные разными разработчиками. Но обычно это возможно только для текстовых файлов и при условии, что изменялись разные части этого файла.

- Контроль прав доступа. Системы контролируют права доступа пользователей, разрешая или запрещая чтение или изменение данных, в зависимости от того, кто запрашивает это действие.

Системы контроля версий широко применяются при разработке программного обеспечения, для хранения исходных кодов разрабатываемой программы.

Типы СКВ

1. Распределенные СКВ – наличие нескольких копий репозитория на разных серверах, где каждый разработчик имеет копию истории изменений. (git)

2. Централизованные СКВ – все версии файлы хранятся на одном сервере (subversion)

3. Локальные СКВ – вся история хранится на компьютере в базе данных

Инструменты для работы с СКВ

1. IDE – интегрированные среды разработки с поддержкой IDE с поддержкой VSC (Visaul Studio, Intelij Idea)

2. Консольные клиенты – программы, позволяющие работать с VCS из cmd (GitBush, Mercurial Workbench)

3. Веб-интерфейсы – онлайн сервисы в браузере, подходят для небольших проектов (GitLab, GitHub)

**22. Разработка пользовательских интерфейсов**

**Разработка пользовательских интерфейсов (UI, User Interface)** — это проектирование и реализация визуальной части программного обеспечения или веб-продукта, с которой взаимодействует пользователь.  
Это кнопки, формы, меню, экраны, анимации — всё, что человек видит и нажимает.

Типы интерфейсов:

* Графический (GUI).
* Командной строки (CLI).
* Веб-интерфейсы.
* Голосовые.

Этапы разработки:

* Анализ пользователей.
* Прототипирование.
* дизайн
* Тестирование удобства (usability).
* Реализация.

**Связь с UX**

* UI — это **как выглядит**.
* UX (User Experience) — это **как работает** и что чувствует пользователь при взаимодействии.  
  Хороший UI невозможен без хорошего UX.