1. Единый репозиторий со структурированными папками

- Структура:

├── Проект\_Веб-студия/

│ ├── Подпроект\_Гостдок/

│ │ ├── docs/

│ │ │ └── отчет\_2023.md

│ │ └── README.md

│ ├── Подпроект\_Киберполигон/

│ └── Подпроект\_Геймдев/

├── Проект\_X/

│ └── docs/

└── README.md (общая инструкция)

- Преимущества:

- Простота управления: все документы в одном месте.

- Легко клонировать/скачивать целиком: git clone, затем переход в нужную папку.

- Возможность использовать GitHub Pages для автоматической публикации (например, через ветку gh-pages).

- Удобство для проектов с общими шаблонами или стандартами.

- Как работать:

- Участники вносят изменения в свои подпапки через Pull Requests.

- Для скачивания:

- Весь репозиторий: git clone <repo-url>.

- Конкретная папка: использовать инструменты вроде [DownGit](https://downgit.github.io/) или git sparse-checkout.

- Минусы:

- Нет изоляции прав доступа (все видят всё).

- Большой размер репозитория при множестве файлов.

2. Отдельные репозитории для каждого проекта/подпроекта

- Структура:

- Основной репозиторий (например, Web-Studio) с README, ссылающимся на подпроекты.

- Отдельные репозитории для каждого подпроекта:

Web-Studio-Gostdoc, Web-Studio-Cyberpolygon, Project-X.

- Преимущества:

- Полная изоляция: настройки доступа, Issues, Wiki для каждого подпроекта.

- Удобно для команд, которые работают независимо.

- Как работать:

- Участники клонируют только нужный репозиторий:

git clone https://github.com/org/Web-Studio-Gostdoc.git.

- Для общего списка проектов можно создать Organization на GitHub или мета-репозиторий с ссылками.

- Минусы:

- Сложнее синхронизировать общие изменения (например, обновление шаблонов).

- Множество репозиториев может запутать новичков.

3. GitHub Pages + Статические генераторы

- Идея: Создать сайт документации, где каждый проект/подпроект — отдельный раздел.

- Инструменты:

- [MkDocs](https://www.mkdocs.org/) или [Docusaurus](https://docusaurus.io/) для структурирования.

- Хостинг на GitHub Pages.

- Структура:

docs/

├── web-studio/

│ ├── gostdoc/

│ │ └── index.md

│ └── cyberpolygon/

├── project-x/

└── index.md

- Преимущества:

- Красивый и удобный интерфейс для чтения.

- Возможность поиска, навигации и версионирования.

4. GitHub Wiki

- Для каждого репозитория можно активировать Wiki и организовать страницы по проектам:

- Home

- Web-Studio

- Gostdoc

- Cyberpolygon

- Project-X

- Плюсы: Встроенная вики, простая Markdown-редакция.

- Минусы: Не подходит для хранения бинарных файлов (PDF, изображения).

Рекомендации

1. Если команды небольшие и доверяют друг другу:

- Используйте единый репозиторий с папками.

- Добавьте GitHub Pages для красивого отображения.

- Настройте GitHub Actions для автоматической проверки изменений.

2. Если нужна изоляция между подпроектами:

- Создайте отдельные репозитории.

- Объедините их через GitHub Organization и мета-репозиторий с ссылками.

3. Для не-технических пользователей:

- Используйте интерфейс GitHub с кнопкой Download ZIP для конкретных папок.

- Напишите инструкцию в README по работе с Git.

Пример workflow для участника из Гостдока

1. Клонировать репозиторий:

git clone https://github.com/your-org/docs.git.

2. Перейти в свою папку:

cd docs/web-studio/gostdoc.

3. Добавить/изменить файлы, сделать commit и push.

4. Создать Pull Request для проверки.

Для скачивания:

- Через браузер: зайти в папку web-studio/gostdoc → Download ZIP.

- Через Git: git sparse-checkout init --cone && git sparse-checkout set web-studio/gostdoc.

Этот подход обеспечит структурированное хранение, простоту доступа и минимальные накладные расходы.