

3. Morin, E.; Gabella, M. Radar-based quantitative precipitation estimation over Mediterranean and dry climate regimes. *J. Geophys. Res.* 2007, 112, 1–13.
4. Gianfranco Vulpiani, Mario Montopoli, Luca Delli Passeri, Antonio G. Gioia, Pietro Giordano, and Frank S. Marzano. On the use of dual-polarized c-band radar for operational rainfall retrieval in mountainous areas. *Journal of Applied Meteorology and Climatology*, 51(2):405–425, Feb 2012. doi:10.1175/JAMC-D-10-05024.1.
5. Инамова В.М., Шутяев В.П., «Алгоритмы и задачи ассимиляции данных для моделей динамики атмосферы и океана. Научно-образовательный курс». [Электронный ресурс]. – Режим доступа: <https://mipt.ru/education/chair/mathematics/upload/99f/algssaasimulation.pdf>.
6. Wang, H., Huang, X.-Y., Xu, D. and Liu, J. (2014) A scale-dependent blending scheme for WRFDA: impact on regional weather forecasting. *Geoscientific Model Development*, 7, 1819–1828.

**РАЗРАБОТКА ЛОКАЛЬНЫХ РЕПОЗИТОРИЕВ ДЛЯ СОЗДАНИЯ
И ОПТИМИЗАЦИИ ВЕБ-РЕСУРСОВ ВИЗУАЛИЗАЦИИ И МОНИТОРИНГА
ДЛЯ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ЗАМКНУТОЙ
ВОДНОЙ СРЕДОЙ В ГИДРО- И АКВОПОННЫХ СИСТЕМАХ
DEVELOPMENT OF LOCAL REPOSITORIES FOR CREATION
AND OPTIMIZATION OF WEB RESOURCES OF VISUALIZATION
AND MONITORING FOR AUTOMATED CONTROL OF CLOSED AQUATIC
ENVIRONMENT IN BOTH HYDRO- AND AQUAPONIC SYSTEM**

**А. И. Каркоцкая, А. А. Антонович, Е. В. Кот
A. Karkotskaya, A. A. Antonovich, E. V. Kot**

*Белорусский государственный университет, МГЭИ им. А. Д. Сахарова БГУ,
г. Минск, Республика Беларусь
anzhelika201777@gmail.com
Belarusian State University, ISEI BSU, Minsk, Republic of Belarus*

Рассматриваются проблемы при работе над одним проектом нескольких разработчиков для решения данных проблем, разработки и оптимизации web-ресурсов для осуществления визуализации и мониторинга для автоматизированного управления замкнутой водной средой в гидро- и аквAPONных системах.

This report examines the problem of working on one project by several developers to solve these problems and development and optimization of web-resources for implementation of visualization and monitoring for the automated control of closed aquatic environment in hydro- and aquaponic systems.

Ключевые слова: система управления версиями, Git, GitHub, TortoiseGit.

Keywords: version control system, Git, GitHub, TortoiseGit.

<https://doi.org/10.46646/SAKH-2020-2-312-315>

Система управления версиями (от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.[1]

Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако, они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов. В частности, системы управления версиями применяются в САПР (Система автоматизированного проектирования — автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности), обычно в составе систем управления данными об изделии (PDM). Управление версиями используется в инструментах конфигурационного управления (Software Configuration Management Tools).

Существует три типа СКВ: локальная, централизованная и распределённая.

Локальные системы контроля версий (далее – ЛСКВ)[2]:

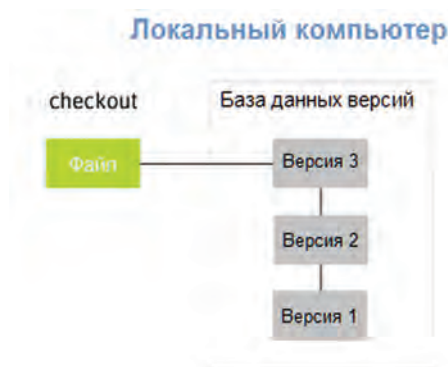


Рисунок 1 – Локальная система контроля версий

В качестве метода контроля версий применяется копирование файлов в отдельную директорию, возможно даже в директорию с отметкой по времени для большего контроля. Данный подход очень популярен и распространен. Изменения сохраняются в виде наборов патчей, где каждый патч датируется и получает отметку времени. Таким образом, если код перестает работать, наборы патчей можно совместить, чтобы получить исходное состояние файла.

Централизованная система контроля версий



Централизованные системы контроля версий (далее – ЦСКВ)[2]:

Рисунок 2 – Централизованная система контроля версий

ЦСКВ были созданы для решения проблемы взаимодействия с другими разработчиками. Такие системы имеют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов, которые получают файлы из этого централизованного хранилища и там же их сохраняют. Тем не менее, такой подход имеет существенный недостаток — выход сервера из строя обернется потерей всех данных. Минус таких систем — это затрудненность одновременной работы нескольких разработчиков над одним файлом.

Распределенные системы контроля версий (далее – РСКВ)[2]:

Распределённая система контроля версий

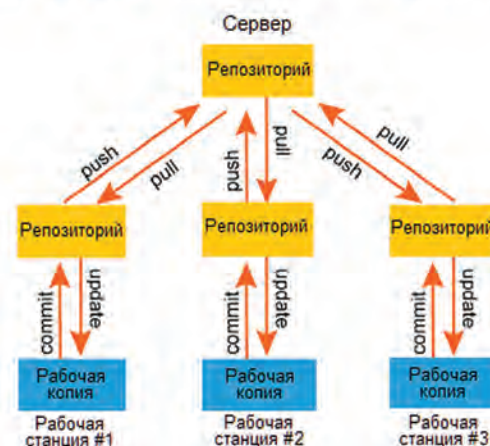


Рисунок 3 – Распределенная система контроля версий

Недостаток ЦСКВ был исправлен в РСКВ, клиенты которых не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени), а полностью копируют репозиторий (хранилище объектов баз данных). Это значит, что у каждого клиента есть копия всего исходного кода и внесённых изменений. В этом случае, если один из серверов выйдет из строя, любой клиентский репозиторий может быть скопирован на другой сервер для продолжения работы. Ещё одним преимуществом РСКВ является то, что они могут одновременно взаимодействовать с несколькими удалёнными репозиториями, что означает, что вы можете параллельно работать над несколькими проектами.

Проанализировав данные виды, для проекта была выбрана распределенная система контроля версий, в частности самая популярная — Git, а также GitHub. Git — это инструмент, позволяющий реализовать распределённую систему контроля версий, а GitHub — это сервис для проектов, использующих Git.

GitHub — сервис онлайн-хостинга репозитория, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

При создании репозитория на github, называемого далее удалённым репозиторием, является создание локальной копии этого репозитория на своем компьютере. Особенностью git является наличие на локальном компьютере полной копии репозитория со всей информацией об истории изменений. Для создания и работы с локальной копией на своем компьютере мы рассматривали 2 популярных подхода: работа с GitHub через командную строку и работу с GitHub с помощью различных приложений. В нашем случае с помощью TortoiseGit.

TortoiseGit — визуальный клиент системы управления исходными кодами программ git для ОС Microsoft Windows. Имеет открытый исходный код и может быть полностью построен с помощью свободно доступного программного обеспечения. Поскольку это не интеграция для конкретной IDE, такой, как Microsoft Visual Studio, Eclipse или другие, вы можете иметь возможность использовать ее с любыми инструментами разработки, которые необходимы в ходе работы, и с любым типом файлов. Основным взаимодействием с TortoiseGit будет использование контекстного меню проводника Windows. TortoiseGit поддерживает вас обычными задачами, такими как фиксация, показ логов, различие двух версий, создание веток и тегов, создание патчей и т. д.

TortoiseGit позволяет выполнять самые необходимые задачи при работе с локальным репозиторием :

1. Обновление текущей ветки из центрального репозитория;
2. Отправка текущей ветки в центральный репозиторий;
3. Переключение на некоторую ветку;
4. Создание новой ветки;
5. Удаление веток;
6. Слияние ветки с текущей;
7. Просмотр и сохранение изменений.

Для упрощения создания разработки web-ресурсов в данном проекте применяется связывание CSS-переменных и JavaScript кода. Это делается для оптимизации загрузки страницы, ведь чем больше число внешних ресурсов, к которым идет обращение, тем больше время требуется для загрузки страницы. В данном примере все файлы CSS и JavaScript объединяются, чтобы уменьшить число ресурсов до двух.

Это делается с помощью `document.querySelector()`. `Document` метод `querySelector()` возвращает первый элемент (Element) документа, который соответствует указанному селектору или группе селекторов. Если совпадений не найдено, возвращает значение `null`.

То есть в JavaScript коде мы прописываем данную функцию, которая возвращает значение данного селекта из CSS переменных.[3]

Например, код: `var usernamePage = document.querySelector('#username-page')`. Возвращает данную CSS переменную:

```
#username-page  
{ text-align: center; }.
```

Это позволяет аккуратно передавать информацию между JavaScript и стилями.

Таким образом, говоря о разработке веб-сайта несколькими разработчиками, которые работают удаленно друг от друга можно утверждать, что распределенная система контроля версиями более удобна и пластична, чем традиционная «офисная» разработка проектов, когда группа разработчиков относительно невелика и целиком находится на одной территории, в пределах единой локальной компьютерной сети, с постоянно доступными серверами. Используя РСКВ, в частности Git, вы получаете высокую производительность, развитые средства интеграции с другими VCS, продуманную систему команд, приятный для работы веб-интерфейс, абсолютный контроль над репозиторием и быстроту работы. А с правильно продуманной структурой файлов и их содержимого, можно получить оптимизированный, понятный к чтению и изменения веб-ресурс.

ЛИТЕРАТУРА

1. Система управления версиями [Электронный ресурс]. URL: https://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%B. (дата обновления: 10.03.2020).

2. Git и GitHub: что это такое и в чём разница [Электронный ресурс]. URL: <https://tproger.ru/translations/difference-between-git-and-github/> (дата обновления: 10.03.2020).

3. Взаимодействие между Javascript и CSS с помощью CSS-переменных [Электронный ресурс]. URL: <https://css-live.ru/articles/vzaimodejstvie-mezhdu-javascript-i-css-s-pomoshhyu-css-peremennykh.html> (дата обновления: 10.03.2020).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПОСТРОЕНИЯ МНОЖЕСТВ МОДЕЛЕЙ ВЫЖИВАЕМОСТИ НА ВЫБОРКАХ ОГРАНИЧЕННОГО ОБЪЁМА В МЕДИЦИНСКИХ ИССЛЕДОВАНИЯХ

SOFTWARE FOR BUILDING OF SETS OF SURVIVAL MODELS ON SAMPLES OF LIMITED SIZE IN MEDICINE RESEARCHES

***А. В. Копыцкий, В. Н. Хильманович, Т. Н. Сакович, А. К. Пашко, В. М. Завадская
A. Kapytski, V. Khilmanovich, T. Sakovich, A. Pashko, V. Zavadskaya***

*Гродненский государственный медицинский университет,
г. Гродно, Республика Беларусь
fizika@grsmu.by
Grodno State Medical University, Grodno, Republic of Belarus*

Анализ выживаемости – один из важных аспектов современной медицинской науки. Однако, зачастую объёмы выборок в медицинских исследованиях невелики, а число показателей, измеренных у пациентов, наоборот, велико. В таких случаях, построение моделей выживаемости при одновременном включении или при пошаговом включении (исключении) переменных-предикторов оказывается невозможным. Кроме этого, многие показатели состояния здоровья пациента могут быть статистически значимо связаны, что приводит к проблеме мультиколлинеарности предикторов в модели. Одновременное решение обеих проблем: малого объёма и мультиколлинеарности – прямой перебор множеств моделей выживаемости, построенных на всех сочетаниях предикторов из определённого подмножества. Нами разработано программное решение на языке «R», позволяющее проводить перебор моделей выживаемости, построенных на отфильтрованных сочетаниях предикторов из их некоторого подмножества.

Survival analysis is one of the important aspects of modern medical science. However, often sample sizes in medical researches are small, but the number of health indicator measured in patients vice versa is high. In such cases, the construction of survival models with the total inclusion of predictors or forward (backward) stepwise methods is impossible. Moreover, many of patients' health indicators can be statistically significantly related, what leads to the problem of multicollinearity of model predictors. The simultaneous solution of both problems: small sample size and multicollinearity is a direct enumeration of the sets of survival models built on all combinations of predictors from a certain subset. We developed a software solution based on the “R” programming language, which allows enumerating survival models built on filtered combinations of predictors from a certain subset of them.

Ключевые слова: программное обеспечение, медицинские исследования, выборки малого объёма, модели выживаемости.

Keywords: software, medicine researches, small samples sizes, survival models.

<https://doi.org/10.46646/SAKH-2020-2-315-318>

Одной из актуальных проблем современной медицинской науки является проблема прогнозирования сроков наступления какого-либо события: смерти пациента, наступления рецидива заболевания, возобновления роста опухоли и т.д. На сегодняшний день существует несколько вариантов построения прогнозов. Среди них можно выделить две группы методов: методы машинного обучения и методы моделирования. Из методов машинного обучения обычно используются: метод опорных векторов, случайный лес, бустинг, нейронные сети [1]. Однако, наряду с более высокой (по сравнению с методами моделирования) точностью предсказания длительности жизни, эти методы отличаются худшей содержательной интерпретацией; используя данные методы, исследователь часто не может логически объяснить влияние факторов и ковариат на зависимую переменную. Кроме этого,