

УДК 004.4

Автоматическая генерация пайплайнов CI/CD на основе метаданных проекта: новый подход к ускорению разработки программного обеспечения

Дьяченко Никита Иванович — магистрант 1-го курса направления 09.04.02 «Информационные системы и технологии». Научные интересы: информационные системы, базы данных.
E-mail: dacenkonikita439@gmail.com

Забродин Андрей Владимирович — канд. истор. наук, доцент кафедры «Информационные и вычислительные системы». Научные интересы: информационные технологии.
E-mail: zabrodin@pgups.ru

Петербургский государственный университет путей сообщения Императора Александра I, Россия, 190031, Санкт-Петербург, Московский пр., 9

Для цитирования: Дьяченко Н. И., Забродин А. В. Автоматическая генерация пайплайнов CI/CD на основе метаданных проекта: новый подход к ускорению разработки программного обеспечения // Интеллектуальные технологии на транспорте. 2025. № 1 (41). С. 74–82. DOI: 10.20295/2413-2527-2025-141-74-82

Аннотация. В статье рассматривается проблема автоматизации процессов конфигурации CI/CD пайплайнов для разработки и развертывания программного обеспечения. Авторы анализируют существующие сложности ручной настройки и предлагают инновационный подход, основанный на использовании метаданных проекта для автоматического создания пайплайнов. **Цель:** разработка метода автоматической генерации пайплайнов на основе метаданных, а также оценка влияния этого подхода на процесс разработки программного обеспечения в рамках практик CI/CD. Для достижения цели разработан сервис, реализующий данный подход. **Результаты:** автоматизация настройки пайплайнов повышает эффективность процессов разработки и сокращает время доставки программных продуктов. **Практическая значимость:** заключается в возможности широкого внедрения предложенного метода в различных сферах разработки ПО, что способствует повышению стандартизации и снижению трудозатрат. В обсуждении высказываются рекомендации по дальнейшему совершенствованию разработанного сервиса, выявляются перспективы его применения и освещаются вопросы, требующие дальнейших исследований и разработок.

Ключевые слова: CI/CD, автоматизация, пайплайн, DevOps, разработка программного обеспечения

2.3.6 — методы и системы защиты информации, информационная безопасность (технические науки);
2.3.5 — математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей (технические науки)

ВВЕДЕНИЕ

В современном мире информационных технологий создание программного обеспечения становится все более сложным и трудоемким процессом. Важнейшим условием любой успешной разработки остается эффективное управление данными и внутренними процессами проекта.

Автоматизация этих процессов может достигаться путем использования пайплайнов (англ. pipeline — трубопровод), которые представляют собой автоматизированные последовательности действий по интеграции, тестированию и развертыванию программного обеспечения.

В данной статье предлагается инновационный подход, который позволяет создавать пайплайны с использованием метаданных проекта. Этот метод представляет собой перспективное решение в области разработки ПО, так как пайплайны позволяют стандартизировать процессы сборки, тестирования и развертывания и предотвращают возможные ошибки. В рамках подхода используются метаданные проекта, которые включают используемые технологии, систему сборки проекта и другие ключевые характеристики проекта.

Пайплайны используются в рамках практик непрерывной интеграции (Continuous Integration, CI) и непрерывной доставки/развертывания (Continuous Delivery/Deployment, CD), что в результате сокращает время между изменением кода и его появлением в производственной среде.

Несмотря на значительные достижения в области CI/CD и использование метаданных, многие вопросы остаются нерешенными. Существующие программные решения часто требуют от разработчиков ручной настройки пайплайнов, что может приводить к ошибкам и увеличению времени настройки процессов. Также стоит отметить, что большинство существующих инструментов ориентированы на конкретные технологии, что ограничивает их применимость. Предлагаемый в статье подход позволит устранить эти проблемы, повысить гибкость и ускорить процессы разработки и развертывания программного обеспечения.

Объектом исследования в работе являются процессы автоматизации, такие как интеграция, тестирование и развертывание. Новизна работы заключается в предложении подхода для создания гибких и адаптируемых пайплайнов, исполь-

зование которых позволит значительно улучшить скорость и качество разработки программных продуктов.

Методологии разработки программного обеспечения CI/CD

Непрерывная интеграция представляет собой методологию разработки программного обеспечения, которая основана на регулярном внесении изменений в общий репозиторий всеми участниками разработки. Каждый добавленный фрагмент кода автоматически проверяется на совместимость с основной кодовой базой [1]. Так как над проектом обычно работает несколько разработчиков, необходимо централизованно хранить все изменения, чтобы обеспечить согласованность кода. Процесс слияния кода в идеале должен выполняться несколько раз в день в автоматическом режиме. Главной целью непрерывной интеграции является поддержание стабильности разработки и выпуска программного обеспечения за счет автоматизации, командного взаимодействия и своевременной проверки внесенных изменений.

Внедрение подхода непрерывной интеграции начинается с регулярной отправки обновлений в систему управления версиями. Это позволяет всем разработчикам работать с актуальным кодом. Каждый коммит автоматически запускает процесс сборки и тестирования, который проверяет корректность работы программы и предотвращает возможные ошибки. Такой подход позволяет реализовать первый этап создания полного CI/CD-пайплайна [2].

На рис. 1 представлена схема процесса CI.

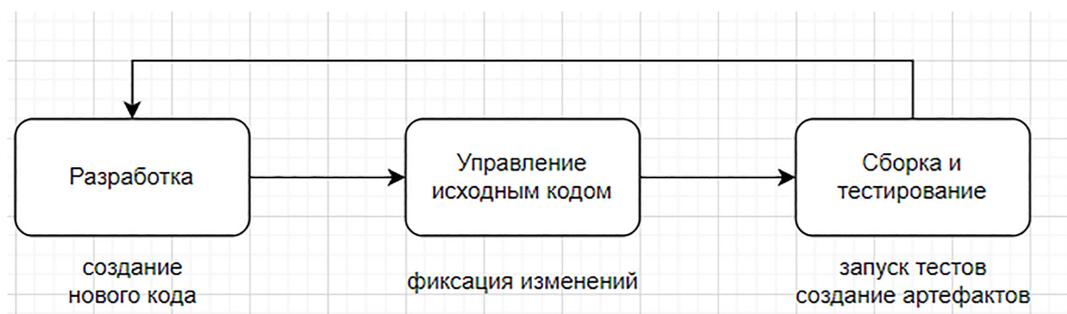


Рис. 1. Схема процесса непрерывной интеграции

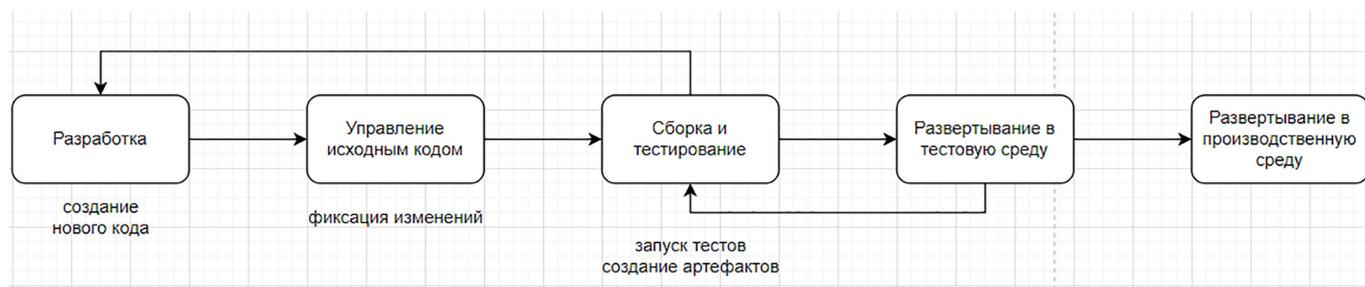


Рис. 2. Схема процесса непрерывной доставки и развертывания

На рис. 2 представлена схема процесса непрерывной доставки и развертывания.

Непрерывная доставка/развертывание является логическим продолжением процесса непрерывной интеграции и предполагает автоматизацию ручных шагов, необходимых для сборки и выпуска ПО. Основной целью этого процесса является обеспечение постоянной готовности кода проекта к развертыванию [3]. Для этого в рабочий процесс доставки необходимо встроить комплекс автоматических тестов. Использование этого метода позволяет добиться ускорения доставки версий программного обеспечения за счет автоматизации ряда задач, благодаря чему разработчики могут сосредоточиться на более сложных задачах.

Задачей CD является ускорение процесса разработки и повышение его надежности путем сокращения времени на доставку версии программного продукта и обеспечения обратной связи от пользователей [4].

Последовательное выполнение процессов CI и CD составляет CI/CD-пайплайн, благодаря которому обеспечивается интеграция, тестирование и доставка новых версий программного обеспечения. Основной задачей пайплайна является ускорение и автоматизация процессов разработки и доставки ПО, что обеспечивает быстрое обновление с минимальными рисками [5].

Основные функции CI/CD-пайплайнов:

1) периодическая интеграция кода в репозиторий проекта, что позволяет отслеживать возможные ошибки в коде;

2) запуск модульных, интеграционных и функциональных тестов, что позволяет выявлять ошибки и проблемы на ранних стадиях разработки;

3) сборка программного продукта и его упаковка в пакеты или контейнеры. Это позволяет создавать релизы приложения и подготавливать его к развертыванию;

4) мониторинг работы приложения, а также отображение отчетов о прохождении тестов, что помогает получить данные о работе приложения [5].

Пайплайны CI/CD описываются текстом с помощью специальных форматов данных. Наиболее часто используемыми из них являются форматы данных YAML и JSON. Эти форматы отличаются читаемым синтаксисом, который позволяет облегчить создание и поддержку пайплайнов. Ниже подробнее рассмотрен формат представления YAML.

YAML (англ. Yet Another Markup Language — «еще один язык разметки») представляет собой формат сериализации данных, который разработан для удобства чтения и записи. YAML хранит данные в структурированном формате (в виде списка) [6].

Конфигурационные файлы в формате YAML применяются в CI/CD следующим образом:

1. *Определение структуры пайплайна.* В конфигурационном файле описываются этапы, шаги и задания пайплайна, что позволяет определять порядок выполнения этапов.

2. *Определение инструкций для сборки и развертывания.* YAML-файл используется для задания последовательности действий на различных этапах пайплайна. В нем указываются команды, необходимые для компиляции кода, выполнения тестов и развертывания приложения в нужной среде.

3. *Конфигурация переменных окружения и зависимостей.* В конфигурационном файле задаются переменные окружения и зависимости, необходимые для корректного функционирования пайплайна. Сюда входит информация о подключении к

базам данных, API-ключи, а также внешние библиотеки, используемые в проекте.

4. *Код-ревью и утверждение кода.* YAML-файл можно использовать для определения рабочих процессов рецензирования и утверждения кода, чтобы гарантировать, что все изменения будут проверены перед слиянием кода с основной веткой.

В настоящее время существует множество программных решений, которые позволяют упростить процесс создания и конфигурации пайплайнов. Далее рассмотрены некоторые из них.

1. *Jenkins.* Программное решение, написанное на языке программирования Java, которое предназначено для обеспечения процесса непрерывной интеграции программного обеспечения. Платформа поддерживает инструменты систем управления версиями, включая AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase и RTC. В приложении существует возможность запуска пайплайна по событию фиксации изменения программного кода в системе управления версиями, а также по расписанию [7]. Вокруг Jenkins образовалось обширное сообщество разработчиков, готовых предоставить поддержку [8].

2. *GitLab CI/CD.* Встроенная в GitLab платформа, которая позволяет автоматизировать сборку, тестирования и развертывания приложений [9]. GitLab CI/CD использует файл конфигурации `.gitlab-ci.yml`, в котором задаются шаги для выполнения пайплайнов. Данная платформа позволяет в некоторой степени автоматизировать создания пайплайнов за счет использования готовых шаблонов, но в большинстве случаев требуется ручная настройка.

3. *GitHub Actions.* Платформа, встроенная в хостинг проектов GitHub, предназначенная для автоматизации рабочих процессов, непрерывной интеграции и развертывания [10]. В GitHub Actions существует возможность запуска пайплайнов на основе определенных событий в репозитории.

Все представленные выше программные решения не позволяют избежать ручной настройки пайплайнов, поэтому возникла необходимость в разработке сервиса, позволяющего если не полностью, то в некоторой степени автоматизировать этот процесс.

Процесс автоматической генерации пайплайнов на основе метаданных проекта

Автоматизация процесса создания пайплайнов имеет большое значение, так как позволяет, во-первых, освободить разработчиков от рутинной работы и повторяющихся задач, во-вторых, уменьшается вероятность человеческих ошибок при настройке пайплайнов. Также автоматизация позволит стандартизировать процесс разработки и интеграции, что способствует улучшению качества и предсказуемости поставки программного обеспечения.

Автоматическая генерация пайплайнов на основе метаданных представляет собой инновационный метод, который использует различные характеристики автоматизируемого проекта для построения пайплайнов CI/CD. Ниже рассмотрены основные аспекты данного метода и приведен пример пайплайна, созданного на его основе:

1. Определение метаданных проекта. На данном этапе программа определяет ключевые атрибуты проекта, такие как его название, язык программирования, система сборки и используемые технологии, и сохраняет их в виде переменных.

2. Генерация пайплайна. Далее автоматически создается пайплайн CI/CD в виде конфигурационного файла формата YAML. Во время создания в него интегрируются полученные от проекта данные. В рамках данного процесса определяются этапы сборки, тестирования, развертывания и других операций, адаптированные к уникальным характеристикам проекта. Для описания всех этапов и параметров пайплайна в конфигурационном файле используется формат YAML.

3. Сгенерированный пайплайн интегрируется в систему CI/CD, где он становится частью цикла непрерывной доставки и развертывания. На данном этапе определяются события триггера (например, отправка изменений (push) в ветку master удаленного репозитория), которые запускают пайплайны.

На рис. 3 представлена часть пайплайна, созданного автоматически с помощью разработанного сервиса.

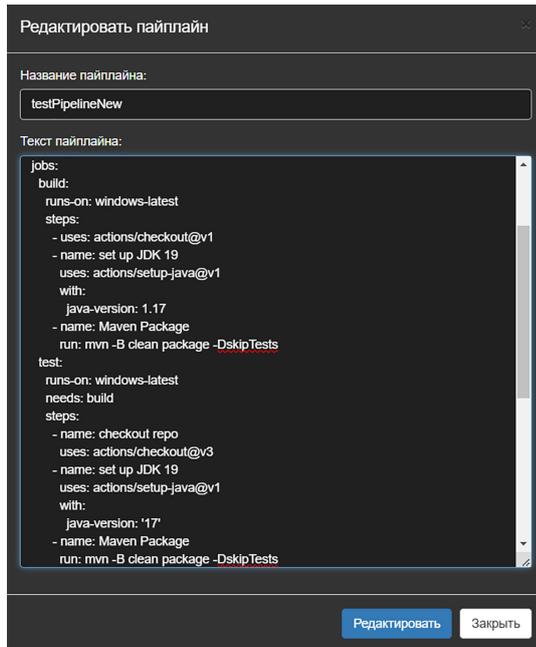


Рис. 3. Пример сгенерированного пайплайна

Из рис. 3 видно, что пайплайн включает в себя этапы `build` (сборка проекта) и `test` (тестирование), в рамках которых происходит выполнение всех необходимых команд. Метаданные, которые были получены в результате анализа проекта: версия JDK, версия Java, система сборки (в данном случае Maven). Эти данные интегрируются в пайплайн на этапе его создания. Параметр `needs` определяет последовательность выполнения этапов. Без успешного выполнения этапа сборки этап тестирования выполняться не будет. Использование данного параметра гарантирует, что последние этапы пай-

плайна будут запущены только после успешного выполнения всех предыдущих этапов.

Далее в статье подробнее рассматривается разработанный сервис, реализующий предложенный подход к созданию пайплайнов.

Разработанный сервис для автоматической генерации пайплайнов

Сервис, использующий представленный в статье подход к созданию пайплайнов, представляет собой веб-сервис, написанный на языке программирования Java с использованием фреймворка Spring и его надстройки Spring Boot. На рис. 4 представлена архитектура приложения.

Из рис. 4 видно, что приложение использует паттерн проектирования MVC (Model-View-Controller), в рамках которого реализуется принцип разделения ответственности между 3 компонентами:

- 1) контроллер — отвечает за обработку запросов на создание, изменение и удаление пайплайнов;
- 2) модель — отвечает за данные приложения и методы работы с ними;
- 3) представление — отвечает за визуализацию данных, полученных из моделей.

Разработанный сервис обеспечивает быстрое создание пайплайнов с акцентом на минимизацию изменений, которые вносит пользователь для обеспечения работоспособности. На рис. 5 представлено окно создания пайплайна.

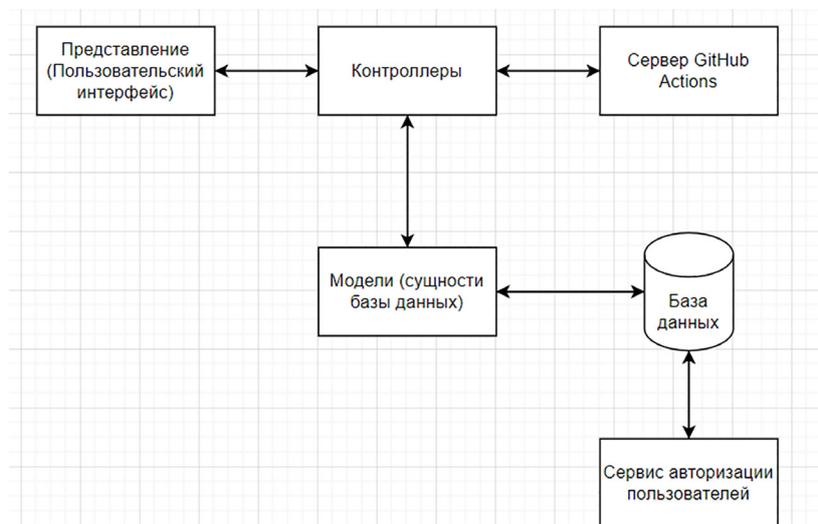


Рис. 4. Архитектура разработанного приложения

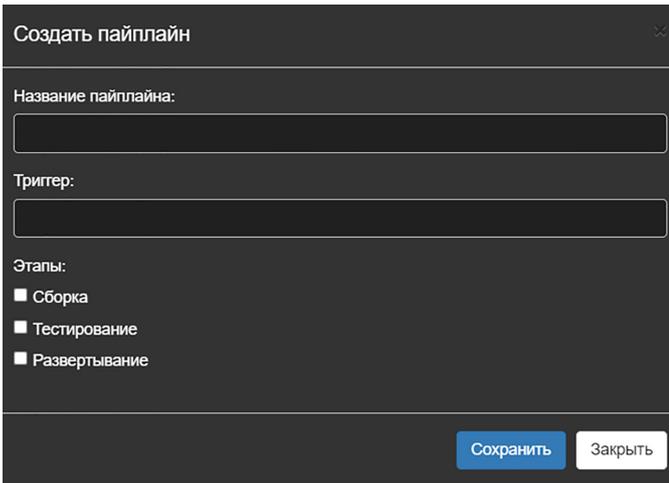


Рис. 5. Окно создания пайплайна

Для создания пайплайна пользователь указывает название, триггер (условие срабатывания) и этапы. Для проверки работоспособности сервиса был выбран проект, который представляет собой чат, написанный на Spring с использованием технологии WebSocket. Чтобы наглядно продемонстрировать работу пайплайна и в целом преимущество методологии CI/CD, необходимо показать, как выглядел развертываемый проект до и как он выглядит после внесения изменений и сколько времени занимает доставка новой версии приложения до конечного пользователя. На рис. 6 представлено окно проекта до внесения в его программный код изменений.

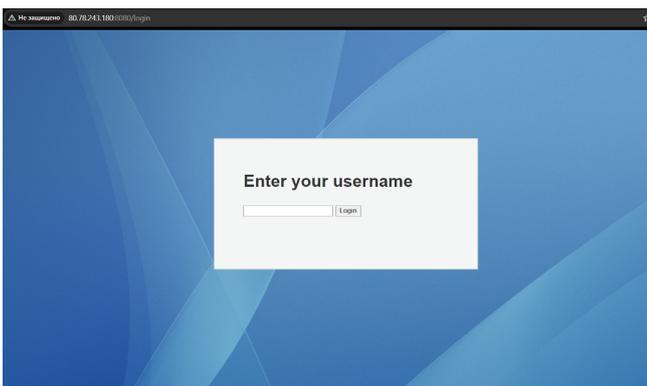


Рис. 6. Окно проекта до внесения изменений

В нем была изменена надпись Enter your username на «Введите ваше имя» и надпись кнопки входа с Login до «Войти». На рис. 7 показан код страницы входа с внесенным изменением.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Login</title>
5    <link rel="stylesheet" href="/css/style2.css" />
6  </head>
7  <body>
8    <div id="login-container">
9      <h1 class="title">Введите ваше имя</h1>
10     <form id="loginForm" name="loginForm" method="POST">
11       <input type="text" name="username" />
12       <button type="submit">Войти</button>
13     </form>
14   </div>
15 </body>
16 </html>

```

Рис. 7. Внесенное изменение

После фиксации и отправки изменения на удаленный репозиторий пайплайн автоматически запускается (рис. 8).

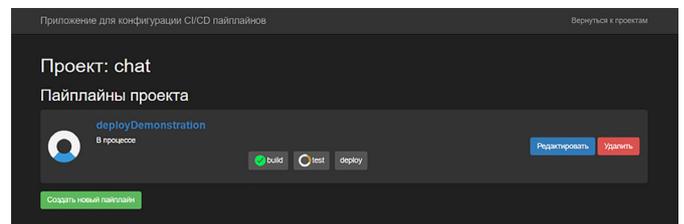


Рис. 8. Запущенный пайплайн

Приложение периодически отправляет запросы для определения состояния проекта. Таким образом отслеживаются изменения, которые разработчики вносят в программный код.

После выполнения всех этапов новая версия приложения развернута на сервере и готова к работе (рис. 9).

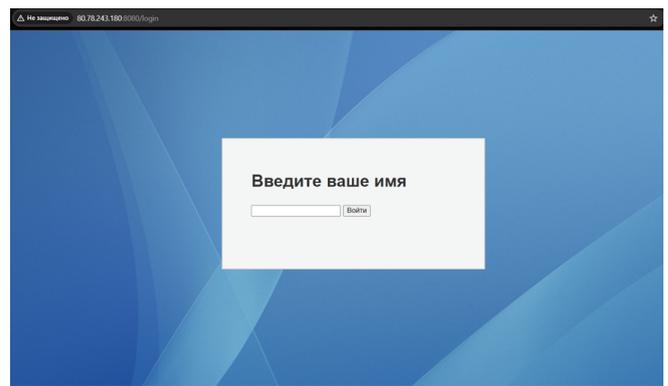


Рис. 9. Окно проекта после внесения изменений

Рис. 9 показывает развернутую на сервере новую версию приложения. Весь процесс от внесения изменения в код до появления новой версии приложения на сервере занял 5 минут 37 секунд,

тогда как при ручном написании пайплайна этот процесс занял бы гораздо большее время. При этом работа приложения на сервере не прерывалась и процесс интеграции новой версии был для пользователей незаметным. И каждое дальнейшее изменение, которое будет вноситься разработчиками в программный код, будет интегрироваться сразу на сервер.

Заключение

Проведенное исследование демонстрирует, что использование метаданных для автоматической генерации CI/CD пайплайнов значительно упрощает процессы настройки и уменьшает вероятность ошибок. В рамках работы был предложен подход, который позволяет значительно упростить процесс создания пайплайнов, повысить их адаптивность и уменьшить вероятность ошибок, связанных с ручной конфигурацией.

В рамках исследования был разработан сервис, реализующий предложенный подход. Он доказал свою эффективность на практике, сократив

время от внесения изменений до их появления в производственной среде. Полученные результаты подтверждают достижение поставленной цели исследования: предложенный метод успешно интегрирует метаданные в процесс создания пайплайнов. Научная новизна работы заключается в использовании метаданных проекта для создания пайплайнов, что ранее не применялось в таком широком формате. Это открывает возможности для дальнейшего усовершенствования подхода, включая адаптацию подхода для различных технологий и языков программирования. В перспективе возможна интеграция дополнительных функций, таких как анализ производительности пайплайнов и их оптимизация в реальном времени.

Использование предложенного метода может быть полезным как для крупных команд разработчиков, так и для небольших проектов. Таким образом, проведенное исследование имеет значение для развития практик CI/CD и способствует улучшению процессов разработки и развертывания программного обеспечения.

СПИСОК ИСТОЧНИКОВ

1. Уилсон К. Грокаем Continuous Delivery / пер. с англ. Н. Ю. Григорьевой. СПб.: Питер, 2024. 400 с.
2. Что такое непрерывная интеграция? // JetBrains. URL: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/continuous-integration> (дата обращения: 29.11.2024).
3. CI/CD системы / Р. А. Мухутдинов, А. А. Мухутдинов, Р. И. Гильмуллин, С. В. Чернова // Актуальные научные исследования в современном мире. 2020. № 12-2 (68). С. 82–85.
4. Что такое непрерывная доставка? // JetBrains. URL: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/continuous-delivery> (дата обращения: 29.11.2024).
5. Что такое CI/CD-пайплайн // Руцентр. URL: http://www.nic.ru/help/что-такое-cicd-pajplajn_11681.html (дата обращения: 30.12.2024).
6. Что такое YAML? // JetBrains. URL: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/faq/yaml> (дата обращения: 04.01.2025).
7. Jenkins (software) // Wikipedia. Обновлено 29.11.2024. URL: [http://en.wikipedia.org/wiki/Jenkins_\(software\)](http://en.wikipedia.org/wiki/Jenkins_(software)) (дата обращения: 07.01.2025).
8. Что такое Jenkins // AppTractor. URL: <http://apptractor.ru/info/articles/jenkins.html> (дата обращения: 08.01.2025).
9. Мулдакаев М. А. Развертывание и непрерывная интеграция приложений с помощью GitLab CI/CD // Научный аспект. 2024. № 5, Т. 13. С. 1704–1708.
10. Туманова М. Б., Мороз А. В. GitHub Actions: автоматизация рабочих процессов для более эффективной разработки // Современные информационные технологии в образовании, науке и промышленности: XXV Международная конференция, XXIII Международный конкурс научных и научно-методических работ, IX Международный конкурс «Научное школьное сообщество»: сборник трудов (Мытищи, Россия, 27–28 апреля 2023 г.). М.: Экон-Информ, 2023. С. 124–128.

Дата поступления: 17.02.2025

Решение о публикации: 18.02.2025

Automatic Generation of CI/CD Pipelines Based on Project Metadata: A New Approach to Accelerating Software Development

Nikita I. Dyachenko — 1st year Master's Degree Student of the 09.04.02 direction "Information systems and technologies".
Research interests: information systems, databases. E-mail: dacenkonikita439@gmail.com

Andrey V. Zabrodin — PhD in History, Associate Professor of the Information and Computing Systems Department.
Research interests: information technology. E-mail: zabrodin@pgups.ru

Emperor Alexander I St. Petersburg State Transport University, 9, Moskovsky ave., Saint Petersburg, 190031, Russia

For citation: Dyachenko N. I., Zabrodin A. V. Automatic Generation of CI/CD Pipelines Based on Project Metadata: A New Approach to Accelerating Software Development. *Intellectual Technologies on Transport*, 2025, No. 1 (41), pp. 74–82. DOI: 10.20295/2413-2527-2025-141-74-82. (In Russian)

Abstract. *The article addresses the issue of automating the configuration processes of CI/CD pipelines for software development and deployment. The authors analyze the existing challenges of manual configuration and propose an innovative approach based on utilizing project metadata for automatic pipeline generation. Purpose: to develop a method for automatic pipeline generation using metadata and to assess the impact of this approach on software development within CI/CD practices. To achieve this goal, a service implementing the proposed approach has been developed. Results: to demonstrate that automating pipeline configuration enhances development efficiency and reduces software delivery time. Practical significance: a potentially wide application of the proposed method across various software development domains would contribute to increased standardization and reduce labor costs. The discussion presents recommendations for further improvements to the developed service, explores its application prospects, and highlights areas requiring further research and development.*

Keywords: *CI/CD, automation, pipeline, DevOps, software development*

REFERENCES

1. Wilson C. Grokaem Continuous Delivery [Grokking Continuous Delivery]. Saint Petersburg, Piter Publishing House, 2024, 400 p. (In Russian)
2. Chto takoe nepreryvnaya integratsiya? [What is Continuous Integration?], *JetBrains*. Available at: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/continuous-integration> (accessed: 29.11.2024). (In Russian)
3. Mukhutdinov R. A., Mukhutdinov A. A., Gilmullin R. I., Chernova S. V. CI/CD sistemy [CI/CD Systems], *Aktualnye nauchnye issledovaniya v sovremennom mire [Actual Scientific Research in the Modern World]*, 2020, No. 12-2 (68), Pp. 82–85. (In Russian)
4. Chto takoe nepreryvnaya dostavka? [What is Continuous Delivery?], *JetBrains*. Available at: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/continuous-delivery> (accessed: 29.11.2024). (In Russian)
5. Chto takoe CI/CD-payplayn [What is CI/CD-pipeline], *Rutsentr [Rucenter]*. Available at: http://www.nic.ru/help/chto-takoe-cicd-pajplajn_11681.html (accessed: 30.12.2024). (In Russian)
6. Chto takoe YAML? [What is YAML?], *JetBrains*. Available at: <http://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/faq/yaml> (accessed: 04.01.2025). (In Russian)
7. Jenkins (software), *Wikipedia*. Last updated November 29, 2024. Available at: [http://en.wikipedia.org/wiki/Jenkins_\(software\)](http://en.wikipedia.org/wiki/Jenkins_(software)) (accessed: 07.01.2025).

8. Chto takoe Jenkins [What is Jenkins], *AppTractor*. Available at: <http://apptractor.ru/info/articles/jenkins.html> (accessed 08 Jan 2025). (In Russian)

9. Muldakaev M. A. Razvertyvanie i nepreryvnaya integratsiya prilozheniy s pomoshchyu GitLab CI/CD [Deploying and Continuously Integrating Applications with GitLab CI/CD], *Nauchnyy Aspekt*, 2024, No. 5, Vol. 13, Pp. 1704–1708. (In Russian)

10. Tumanova M. B., Moroz A. V. GitHub Actions: avtomatizatsiya rabochikh protsessov dlya bolee effektivnoy razrabotki [GitHub Actions: Workflow Automation for More Efficient Development], *Sovremennye informatsionnye tekhnologii v obrazovanii, nauke i promyshlennosti: XXV Mezhdunarodnaya konferentsiya, XXIII Mezhdunarodnyy konkurs nauchnykh i nauchno-metodicheskikh rabot, IX Mezhdunarodnyy konkurs «Nauchnoe shkolnoe soobshchestvo»: sbornik trudov* [Modern Information Technologies in Education, Science and Industry: Proceedings of the XXV International Conference, XXIII International Competition of Scientific and Scientific and Methodological Works, IX International Competition “Scientific School Community”], Mytishchi, Russia, April 27–28, 2023. Moscow, Econ-Inform Publishing House, 2023, Pp. 124–128. (In Russian)

Received: 17.02.2025

Accepted: 18.02.2025