

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра системного анализа и автоматического управления

**РАЗРАБОТКА БРАУЗЕРНОЙ ИГРЫ „WATER SORT“ С  
ИСПОЛЬЗОВАНИЕМ МЕТОДОЛОГИИ РАЗРАБОТКИ СИ/СД**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 5 курса 551 группы  
направления 09.03.04 — Программная инженерия  
факультета КНиИТ  
Говорухина Владимира Алексеевича

Научный руководитель  
старший преподаватель

\_\_\_\_\_

М. В. Белоконь

Заведующий кафедрой  
к. ф.-м. н., доцент

\_\_\_\_\_

И. Е. Тананко

Саратов 2021

## ВВЕДЕНИЕ

**Актуальность темы.** Непрерывная интеграция (Continuous Integration, CI) и непрерывная поставка (Continuous Delivery, CD) представляют собой культуру, набор принципов и практик, которые позволяют разработчикам чаще и надежнее развертывать изменения программного обеспечения.

**Цель бакалаврской работы** – разработка браузерной игры „Water Sort“ с использованием методологии разработки CI/CD.

Поставленная цель определила **следующие задачи:**

- изучить принципы работы методологии;
- ознакомиться с инструментами и технологиями, которые позволяют реализовать эти принципы;
- определить набор инструментов, который потребуется для реализации;
- применить CI/CD методологию при разработке игры.

**Практическая значимость бакалаврской работы.** Ориентироваться на CI/CD полезно как инфраструктурным инженерам, так и разработчикам.

Можно выделить пять основных причин, почему:

1. экономия времени благодаря переиспользованию кода и быстрому развертыванию проектов;
2. получение ожидаемого результата от деплоя;
3. независимость проекта от окружения;
4. иммутабельная инфраструктура и легкая миграция;
5. возможности подхода „Инфраструктура как код“.

**Структура и объём работы.** Бакалаврская работа состоит из введения, семи разделов, заключения, списка использованных источников и пятнадцати приложений. Общий объём работы – 64 страницы, из них 33 страницы – основное содержание, включая 18 рисунков, 23 страницы приложения, список использованных источников информации – 22 наименования.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Первый раздел «Описание разрабатываемого программного продукта»** посвящен описанию разработки веб-приложения, которое представляет из себя небольшую браузерную игру, относящуюся к жанру легких головоломок. На каждом уровне перед пользователем предстает некоторое

количество колб (бутылок), в каждой из которой находятся несколько разноцветных жидкостей и пара пустых колб. Жидкости не смешиваются и наливааются послойно (как вода и масло).

Цель игры заключается в сортировке всех жидкостей по разным колбам. Когда все цвета будут рассортированы по колбам, уровень посчитается пройденным, игрок получит за него балы и для него откроется доступ к следующему уровню.

**Второй раздел «CI/CD»** посвящен описанию деталей CI/CD методологии, непрерывной интеграции и непрерывной поставки.

**Третий раздел «Git»** посвящен описанию инструмента Git – это консольная утилита для отслеживания и ведения истории изменения файлов в проекте. Чаще всего его используют для кода, но можно и для других файлов. Например, для изображений – полезно для дизайнеров.

С помощью Git можно откатить проект до более старой версии, сравнивать, анализировать или сливать свои изменения в репозиторий.

Репозиторием называют хранилище кода и истории его изменений. Git работает локально и все репозитории хранятся в определенных папках на жестком диске.

**Четвертый раздел «Docker»** посвящен описанию инструмента Docker, который представляет собой систему управления контейнерами. Она позволяет „упаковать“ приложение или веб-сайт со всем его окружением и зависимостями в контейнер, которым в дальнейшем можно легко и просто управлять: переносить на другой сервер, масштабировать, обновлять.

Важной частью экосистемы Docker является Docker Hub – открытый репозиторий образов контейнеров. В нем можно найти десятки готовых приложений от официальных разработчиков. Среди них – nginx, MySQL, Apache, Gitlab, Redmine, Elasticsearch, Jenkins и другие.

В подразделе 4.1 описываются Преимущества Docker и что используя Docker, можно оптимизировать работу и сэкономить время.

**Пятый раздел «Jenkins»** посвящен описанию инструмента Jenkins, который позволяет автоматизировать процессы. Jenkins используется для непрерывной сборки и тестирования программных проектов, что облегчает разработчикам интеграцию изменений в проект и облегчает пользователям получение новой сборки. Это также позволяет непрерывно поставлять про-

граммное обеспечение, интегрируя с большим количеством технологий тестирования и развертывания.

С помощью Jenkins организации могут ускорить процесс разработки программного обеспечения за счет автоматизации. Jenkins объединяет процессы жизненного цикла разработки всех видов, включая сборку, документацию, тестирование, пакет, этап, развертывание, статический анализ и многое другое.

В подразделе 5.1 описывается использование Jenkins в непрерывной интеграции.

**Шестой раздел «Реализация CI/CD-конвейера для игры „Water Sort“»** посвящен описанию процесса разработки игры с использованием методологии CI/CD.

Приложение состоит из двух частей: графический интерфейс (frontend) и серверная часть (backend). Разработка каждой части ведется в двух разных github-репозиториях с использованием различных языков программирования и технологий. Графический интерфейс (frontend), с которым взаимодействует пользователь, реализован с помощью языка разметки документа – HTML, формального языка описания внешнего вида документа – CSS и мультипарадигменного языка программирования – JavaScripts. Внутренняя или серверная часть (backend), которая отвечает за основные вычисления, реализована с помощью скриптового языка PHP и свободной реляционной системой управления базами данных – MySQL.

Для того, чтобы пользовательский интерфейс и программно-аппаратная часть могли взаимодействовать есть оговоренные эндпоинты (адреса), на которые и происходит взаимодействие – frontend опрашивает их, а backend предоставляет необходимую информацию.

Для работы приложения необходим сервер, на котором будет производиться запуск приложений, а так же база данных MySQL, которая арендуется у сторонней компании. В роли сервера будет выступать, также арендованный, виртуальный сервер с операционной системой Ubuntu 20.04, доступ к которому, для управления им, будет осуществляться по протоколу SSH. Для удобства передачи новых версий на сервер на нем установлен Docker, а запуск и работа приложений будут происходить в контейнерах Docker.

В подразделе 6.1 описывается участие GIT в CI/CD конвейере, как ис-

пользуются ветки и коммиты в методологии CI/CD.

В подразделе 6.2 описывается участие Jenkins в CI/CD конвейере, что он является главным центром конвейера по автоматизации, который и позволяет объединить все составляющие.

**Седьмой раздел «Потенциал CI/CD»** посвящен описанию преимуществ методологии CI/CD, что это современная методика, которая постоянно дополняется новыми технологиями, инструментами и идеями. Ее можно расширять и подстраивать под любые задачи благодаря огромному количеству инструментов. В данном разделе будут приведены некоторые из них.

В подразделе 7.1 описывается гибкость Jenkins, что в базовой комплектации Jenkins позволяет добавлять действия блоками, то есть для каждого нового действия создается сначала блок „сервер“, который указывает, на каком сервере то или иное действие выполняется, а в нем уже создаются блоки самого действия.

В подразделе 7.2 описываются Облачные сервисы, что это совокупность программ и различных платформ, которые „живут“ и работают на серверах облачных провайдеров. Они позволяют быстро и легко решить проблему нехватки серверных мощностей – расширить уже существующую архитектуру или создать ее с нуля.

В подразделе 7.3 описывается подход к инфраструктуре IaC и инструмент Terraform. Что идея, стоящая за IaC, заключается в том, что для определения, развертывания, обновления и удаления инфраструктуры нужно писать и выполнять код. И что Terraform — это популярный инструмент, который позволяет управлять облачной инфраструктурой в парадигме Infrastructure as a Code на очень дружелюбном, легко читаемом декларативном языке. Его применение может обеспечить единый вид ресурсов и использование практик работы с кодом для управления инфраструктурой, которые уже выработаны сообществом разработчиков.

В подразделе 7.4 описывается программное обеспечение для централизованного управления конфигурациями – Ansible, которое позволяет обеспечить централизованное управление конфигурациями (другими операционными системами и установленными на них программами).

В подразделе 7.5 описываются стратегии доставки, чем они отличаются и как выбрать среди них.

## ЗАКЛЮЧЕНИЕ

Методология CI/CD – это отличная инструмент для автоматизации процессов разработки программного обеспечения, который позволит ускорить исправление ошибок в коде и выход новых версий продукта. Но тем не менее, будет ошибкой принимать методологию как панацею и стремиться внедрить ее во все проекты. При недостатке опыта это приведет к усложнению работ по IT-продуктам.

Также нужно уделить внимание и организации взаимодействия между проектными группами, так как CI/CD сильно завязан на человеческий фактор. Инженеры, scrum-специалисты, аналитики, dev-группы и другие участники проекта должны работать в единой экосистеме с адекватным руководством и проектным управлением.

В ходе написания бакалаврской работы была изучена методология CI/CD и настроен конвейер разработки веб-приложения, разработана и программно реализована браузерная игра „Water Sort“.

Таким образом, поставленная цель достигнута, все задачи выполнены.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Что такое CI/CD? Разбираемся с непрерывной интеграцией и непрерывной поставкой [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/company/otus/blog/515078/> (Дата обращения 14.03.2021). Загл. с экрана. Яз. рус.
- 2 Что такое методология разработки CI/CD [Электронный ресурс] : [сайт]. URL: <https://selectel.ru/blog/what-is-ci-cd/> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 3 CI/CD: принципы, внедрение, инструменты [Электронный ресурс] : [сайт]. URL: <https://medium.com/southbridge/ci-cd-principi-vnedrenie-instrumenti-f0626b9994c8> (Дата обращения 24.04.2021). Загл. с экрана. Яз. рус.
- 4 *Chacon, S.* Pro Git / S. Chacon. — Apress, 2014.
- 5 Git и GitHub: что это такое и в чём разница [Электронный ресурс] : [сайт]. URL: <https://tproger.ru/translations/>

- [difference-between-git-and-github/](#) (Дата обращения 04.01.2021).  
Загл. с экрана. Яз. рус.
- 6 Что такое Git? [Электронный ресурс] : [сайт]. URL: <https://www.atlassian.com/ru/git/tutorials/what-is-git> (Дата обращения 15.05.2021). Загл. с экрана. Яз. рус.
- 7 Что такое Docker, с чем его едят и почему он похож на консоли Dendy и PlayStation [Электронный ресурс] : [сайт]. URL: <https://vc.ru/dev/156389-что-такое-docker-s-chem-ego-edyat> (Дата обращения 23.02.2021). Загл. с экрана. Яз. рус.
- 8 Полное практическое руководство по Docker: с нуля до кластера на AWS [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/post/310460/> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 9 Что такое Docker: для чего он нужен и где используется [Электронный ресурс] : [сайт]. URL: <https://selectel.ru/blog/what-is-docker/> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 10 What is Jenkins? The CI server explained [Электронный ресурс] : [сайт]. URL: <https://www.infoworld.com/article/3239666/what-is-jenkins-the-ci-server-explained.html> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 11 Учимся разворачивать микросервисы. Часть 4. Jenkins [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/post/493580/> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 12 Что такое Jenkins? [Электронный ресурс] : [сайт]. URL: <https://dataengineer.ru/?p=1647> (Дата обращения 13.03.2021). Загл. с экрана. Яз. рус.
- 13 htmlbook.ru | Для тех, кто делает сайты [Электронный ресурс] : [сайт]. URL: <http://htmlbook.ru/> (Дата обращения 13.02.2021). Загл. с экрана. Яз. рус.
- 14 *Дебольт, В.* HTML и CSS. Совместное использование / В. Дебольт. — Пресс, 2006.

- 15 Современный учебник JavaScript [Электронный ресурс] : [сайт]. URL: <https://learn.javascript.ru/> (Дата обращения 13.02.2021). Загл. с экрана. Яз. рус.
- 16 *Гарнаев, А. К.* WEB-программирование на Java и JavaScript / А. К. Гарнаев. — БХВ-Петербург, 2005.
- 17 *Дронов, В. А.* JavaScript в Web-дизайне / В. А. Дронов. — БХВ, 2001.
- 18 Groovy за 15 минут – краткий обзор [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/post/122127/> (Дата обращения 14.04.2021). Загл. с экрана. Яз. рус.
- 19 *Брикман, Е.* Terraform: Up & Running: Writing Infrastructure as Code / Е. Брикман. — Wiley, 2017.
- 20 Как мы переложили управление инфраструктурой на Terraform — и начали жить [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/company/dins/blog/470543/> (Дата обращения 04.01.2021). Загл. с экрана. Яз. рус.
- 21 *Р. Мозер, Л. Х.* Запускаем Ansible / Л. Х. Р. Мозер. — ДМК Пресс, 2018.
- 22 Стратегии деплоя в Kubernetes: rolling, recreate, blue/green, canary, dark (А/В-тестирование) [Электронный ресурс] : [сайт]. URL: <https://habr.com/ru/company/flant/blog/471620/> (Дата обращения 04.01.2021). Загл. с экрана. Яз. рус.