

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

"На правах рукопису"
УДК 004.4'6

«До захисту допущено»
Науковий керівник кафедри

_____ Дичка І.А.
(підпис)

“ ____ ” _____ 2017 р.

Магістерська дисертація

зі спеціальності 8.05010301 “ Програмне забезпечення систем”

на тему: МЕТОД АНАЛІЗУ СТРАТЕГІЇ НЕПЕРЕРВНОЇ ІНТЕГРАЦІЇ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ВИКОРИСТАННІ
АРХІТЕКТУРИ МІКРОСЕРВІСІВ

Виконав: студент 6 курсу, групи КП-51м

Кисельов Євгеній Леонідович

_____ (підпис)

Науковий керівник доц., к.т.н. Заболотня Т.М.

_____ (підпис)

Рецензент доц., к.т.н., доц. Дідковська М. В.

_____ (підпис)

Рецензент доц., к.т.н., доц. Марченко О. І.

_____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2017

Реферат

Актуальність теми. Кожного дня кількість користувачів програмного забезпечення стрімко збільшується, розмір програм та їх складність також невпинно зростає. Це призводить до збільшення кількості людей, задіяних у процесі розроблення програмного забезпечення, та їх впливу на роботу один одного. На сьогоднішній день існує багато підходів до процесу неперервної інтеграції програмного забезпечення, проте не всі з них підходять для мікросервісної архітектури, яка в свою чергу є одним з обов'язкових підходів для підвищення простоти підтримки та розгортання систем. Разом з цим збільшення кількості мікросервісів прямо пропорційно збільшує складність управління системою загалом, а також ускладнює процеси, пов'язані з розгортанням окремого сервісу, перевірки його роботи відносно інших сервісів, виконання інтеграційного тестування та можливості незалежної інтеграції виправлень у сховище коду. З цими стратегічними та технологічними проблемами сьогодні стикаються всі розробники, які починають розгортати системи неперервної інтеграції для власного проекту з мікросервісною архітектурою.

Об'єктом даного дослідження є процес розроблення програмного забезпечення.

Предметом дослідження є методи аналізу стратегій неперервної інтеграції програмного забезпечення при використанні архітектури мікросервісів, а також методи оптимізації процесу розроблення програмного забезпечення.

Мета роботи. Метою дослідження є зменшення часу, затраченого на інтеграцію та перевірку коду розробника у процесі неперервної інтеграції за рахунок розроблення та реалізації комбінованого методу неперервної інтеграції в архітектурі мікросервісів та програмних засобів.

Відповідно до вказаної мети в роботі поставлені і розв'язані такі задачі:

- аналіз існуючих методів та засобів неперервної інтеграції в архітектурі мікросервісів;

- формування переліку критеріїв ефективності методів неперервної інтеграції в мікросервісній архітектурі;
- розроблення методу управління інструментами неперервної інтеграції, який дозволить розробникам витратити менше часу на інтеграцію;
- дослідження ефективності запропонованого методу на практиці відповідно до розробленого переліку критеріїв.

Методи дослідження. В роботі використовуються емпіричні методи, метод порівняння середніх значень та аналіз швидкодії роботи системи.

Наукова новизна роботи полягає в наступному:

1. Сформовано перелік критеріїв, що дозволяють визначити ефективність методів неперервної інтеграції та дозволяють виділити ключові параметри, що прискорюють процес неперервної інтеграції та підвищують його стабільність.
2. Створено комбінований метод неперервної інтеграції, що згідно з критеріями оцінки є більш ефективним в умовах падіння сервісу у мікросервісній архітектурі та зменшує час, затрачений на його інтеграційне тестування та розгортання, що відрізняється від існуючих паралельним запуском фаз та використанням контейнерів як інструменту версіонування сервісів.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований метод дозволив збільшити швидкодію проходження фаз збірки під час процесу неперервної інтеграції, а також значно скоротив час повторної перевірки, запуску окремого сервісу та введення його у роботу.

Здійснено програмну реалізацію, яка дозволяє масштабувати збірку в залежності від розміру системи, незалежно розгортати окремі сервіси у разі їх падіння та виконувати інтеграційні тести на різних версіях сервісів. Розроблене програмне забезпечення та конфігурація існуючих інструментів враховує сучасні потреби до неперервної інтеграції та поставки.

Апробація роботи. Основні теоретичні положення та результати роботи були представлені та обговорювались на ІХ науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг»

ПМК-2017 (Київ, 19–21 квітня 2017 р.) та опубліковані у збірнику тез за результатами конференції.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п'яти розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень.

У першому розділі розглянуто загальні підходи до побудови процесу неперервної інтеграції програмного забезпечення; розглянуто існуючі програмні рішення та інструменти, що дозволяють побудувати систему для неперервної інтеграції; проведено огляд існуючих шаблонів побудови мікросервісної архітектури та зазначено детальну характеристику основних відмінностей між ними.

У другому розділі обрано шаблон мікросервісної архітектури, згідно з яким розроблена стратегія неперервної інтеграції; встановлені критерії оцінки неперервної інтеграції та наведений їх опис; сформульований метод аналізу неперервної інтеграції в архітектурі мікросервісів; описана теоретична частина запропонованого комбінованого методу на основі результатів аналізу.

У третьому розділі запропоновано засоби реалізації для кожного з етапів методу, описаний процес підготовки середовища, розгортання необхідних інструментів; описаний процес побудови кожної з фаз збірки; наведені етапи тестування системи та перевірки основних сценаріїв запропонованого підходу.

У четвертому розділі наведено результати аналізу доцільності використання запропонованого підходу; описані головні переваги та недоліки в залежності від різних стратегій інтеграції програмного забезпечення; проаналізовані таблиці з тестовими результатами швидкодії різних підходів; описана відповідність системи встановленим критеріям оцінки.

У п'ятому розділі описані можливі шляхи для створення стартапу на основі запропонованого методу.

У висновках проаналізовано отримані результати роботи.

У додатках наведено фрагменти програмної реалізації запропонованого методу та копії графічних матеріалів.

Робота виконана на 100 аркушах, містить 2 додатки та посилання на список використаних літературних джерел. У роботі наведено 17 рисунків та 2 таблиці.

Ключові слова: збірка, інтеграція, контейнер, неперервна інтеграція, мікросервісна архітектура.

Abstract

Topic actuality. Every day number of software users is rapidly increasing, the size of programs and their complexity is also steadily increasing. This leads to increasing the number of people who involved in the software development process and their impact on each other's work. In our days there are many approaches of continuous integration, but not all of them are suitable for microservice architecture, which is one of the mandatory approaches for increasing systems support ease and deployment. However, increasing microservices number directly increases complexity of system managing, as well as complicates the processes associated with separate service deployment, checking its integration with other services and the ability to independently integrate patches into the code repository. Today, all developers who are beginning deploy continuous integration system for their own project with micro-service architecture are facing these strategic and technological issues.

The object of this research is the software development process.

The subject of the research is the methods of strategies analysis for the continuous integration using the microservices architecture, as well as methods for optimizing software development process.

Work purpose. The purpose of the study is to reduce time spent on developer's code integration and verification in the continuous integration process through the development and implementation of a continuous integration combined method in microservices architecture and software.

In accordance with the stated work purpose there were set and solved the following tasks:

- analysis of existing methods and means of continuous integration

in the microservices architecture;

- formation of a list of criteria of efficiency of methods of continuous integration in the microservices architecture;
- developing a method for managing continuous integration tools that will allow developers to spend less time integrating;
- studying the effectiveness of the proposed method in practice in accordance with the developed list of criteria.

Research methods. The paper uses empirical methods, a method for comparing averages and analyzing the system's performance.

The scientific novelty of the work is as follows:

1. A list of criteria has been formed that allows us to determine the effectiveness of continuous integration methods and to distinguish key parameters that accelerate the process of continuous integration and increase its stability.
2. A composite continuous integration method is created that according to the evaluation criteria is more effective in the context of the fall in service in the micro-service architecture and reduces the time spent on its integration testing and deployment, which differs from the existing parallel run of phases and the use of containers as a tool for service versioning.

The practical value of the results obtained in the work is that the proposed method has allowed to increase the speed of passing phases of the assembly during the process of continuous integration, as well as significantly reducing the time of re-verification, the launch of a separate service and putting it into work.

The program implementation is implemented, which allows to scale the assembly according to the size of the system, independently deploy separate

services in case of their fall and perform integration tests on different versions of services. The developed software and the configuration of the existing tools take into account the current needs for continuous integration and delivery.

Approbation of research. The main theoretical positions and results of work were presented and discussed at the IX scientific conference of masters and postgraduates "Applied Mathematics and Computer", PMK-2017 (Kyiv, April 19-21, 2017) and published in the abstracts on the results of the conference.

Structure and scope of work. The master's thesis consists of an introduction, five sections, conclusions and appendices.

The introduction gives a general description of the work, an assessment of the current state of the problem is performed, the relevance of the research direction is substantiated.

The first section discusses general approaches to building a process of continuous software integration; Existing software solutions and tools that allow to build a system for continuous integration are considered; An overview of the existing patterns of the construction of the micro-service architecture has been reviewed and a detailed description of the main differences between them is given.

In the second section the template of the micro-service architecture was selected, according to which the strategy of continuous integration was developed; The criteria for the assessment of continuous integration are established and their description is given; The method of analysis of continuous integration in the architecture of microservices is formulated; The theoretical part of the proposed combined method is described on the basis of the results of the analysis.

In the third section, the means of implementation for each stage of the

method are described, the process of preparation of the environment is described, the deployment of the necessary tools; Describes the process of constructing each of the phases of the assembly; The stages of testing the system and checking the main scenarios of the proposed approach are presented.

The fourth section presents the results of an analysis of the feasibility of using the proposed approach; Describes the main advantages and disadvantages depending on various software integration strategies; Analyzed tables with test results of the speed of different approaches; The system's compliance with the established criteria of assessment is described.

The fifth section describes the possible ways to create a startup based on the proposed method.

The conclusions are analyzed the results of the work.

The annexes contain fragments of software implementation of the proposed method and copies of graphic materials.

The work is made on 87 sheets, contains 2 attachments and a link to the list of used literary sources. The paper contains 37 figures and 3 tables.

Keywords: assembly, integration, container, continuous integration, microservice architecture.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Неперервна інтеграція [Електронний ресурс] — Режим доступу : https://uk.wikipedia.org/wiki/Неперервна_інтеграція.
2. *Мікросервіси* [Електронний ресурс] — Режим доступу : <https://uk.wikipedia.org/wiki/Мікросервіси>.
3. Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, Mike Amundsen. *Microservice Architecture: Aligning Principles, Practices, and Culture*, 2016. — p. 17–48.
4. Newman S. *Building Microservices*, 2015. — 123-273 с.
5. Duvall P., Matyas S., Glover A. *Continuous Integration. Improving Software Quality and Reducing Risk*, 2007. — 17-164 .
6. Humble J., Farley D. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, 2011. — 86-112 .
7. Кисельов. Є.Л. Комбінований метод неперервної інтеграції мікросервісів. — Конференція «Прикладна математика та комп'ютинг». — Київ, 2017.