

РОЗРОБКА ТА ДОСЛІДЖЕННЯ АРХІТЕКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

*д-р. техн. наук, проф. В.І. Носков, магістр О.В. Лазебний,
Національний технічний університет "Харківський політехнічний
інститут", м. Харків*

Автоматизація ефективно застосовується на сучасному етапі розвитку людства з метою досягнення зростання показників ресурсозбереження, поліпшення екології навколишнього середовища, якості, надійності та швидкої розробки продукції. В зв'язку з бурхливим розвитком ІТ сфери, методологія DevOps дає змогу використовувати найдосконаліші методи автоматизації та поставки програмного продукту. Нове програмне забезпечення дозволяє впроваджувати комп'ютерні технології у нетрадиційній сфері діяльності підприємства, що проявляється в інтеграції виробничих процесів та управління ними.

Протягом останніх десяти років стався значний зсув у підході, який кардинально змінив увесь ІТ-ландшафт. Розробники, втомлені від постійного очікування зворотного зв'язку, почали писати програмне забезпечення, яке автоматизує операційні завдання. Водночас фахівці ІТ-підрозділів почали делегувати свої глибокі знання й багатий досвід у програмне забезпечення, яке написано розробниками. Сьогодні, коли дуже чіткі межі між розробкою та експлуатацією почали зникати, це призвело до прискорення всього життєвого циклу програмного забезпечення, більш коротких сесій забезпечення якості та багатьох інших змін. З'являються нові процеси, такі як управління без перервною інтеграцією й безперервним розгортанням. Багато великих програмних рішень, які розроблені за допомогою методології DevOps, розгортаються по кілька разів на день, а не кілька разів на рік, як було раніше.

В роботі проведено аналіз існуючих підходів до розробки архітектур систем автоматизованого безперервного розгортання програмного забезпечення. На основі цього аналізу в якості базової методології було обрано DevOps. Виходячи поставленої задачі було взято наступний список інструментів DevOps: **Code:** Git; **Build:** Maven; **Test:** Junit, Selenium; **Release:** Jenkins; **Deploy:** Docker, AWS, Nexus; **Operate:** Chef, Zabbix, Grafana, Keycloak. Запропоновано архітектуру взаємодії сучасних компонентів методології DevOps, яка на відміну від існуючих архітектурних рішень, використовує сучасні компоненти, що дозволяє скоротити час на розробку, тестування та розгортання програмного забезпечення.