

CI/CD/CD

№ уроку: 37 **Курс:** Manual QA

Засоби навчання: Браузер, Microsoft Office

Огляд, мета та призначення уроку

Мета даного уроку – отримати загальну картину CI/CD/CD процесів на проєкті та роль QA у них.

Вивчивши матеріал даного заняття, учень зможе:

- Розбиратися у базовій теорії CI/CD процесу.
- Вільно орієнтуватися та брати участь у DevOps області програми.
- Налаштовувати Quality Gates.

Зміст уроку

- Головні принципи
 - Основи CI/CD/CD
 - Поняття Continuous testing
- Структура оточення проєкту (Dev->Test->UAT->Prod)
- Quality Gates
 - У CD процесі
 - У QA процесі
 - У Test Automation процесі

Резюме

- **Система контролю версій (Version Control System, VCS)** – представляє собою програмне забезпечення, яке дозволяє відстежувати зміни в документах, при необхідності проводити їх відкат, визначати, хто і коли вніс виправлення.
- **Безперервна інтеграція (CI)** — первинний, базовий процес оновлення ПЗ, у якого всі зміни лише на рівні коду вносяться у єдиний центральний репозиторій. Таке внесення прийнято називати злиттям. Після кожного злиття (що проходить по кілька разів на день) у змінній системі відбувається автоматичне складання та тестування (перевірка конкретних модулів коду, UI, продуктивності, надійності API). Таким чином, розробники страхуються від занадто пізніх виявлень проблем в оновленнях.
- **Безперервна доставка (CD)** — CI + CD. Наступний після CI рівень. Тепер нова версія не тільки створюється і тестується при кожній зміні коду, що реєструється в репозиторії, але і може бути оперативно запущена по одному натисканні розгортання кнопки. Однак запуск розгортання все ще відбувається вручну - ту саму кнопку все ж таки треба комусь натиснути. Цей метод дозволяє випускати зміни невеликими партіями, які легко змінити чи усунути у разі потреби.
- **Безперервне розгортання (CD)** — CI + CD + CD. Після автоматизації релізу залишається один ручний етап: схвалення та запуск розгортання у продакшен (кнопка). Практика безперервного розгортання скасовує і це, не вимагаючи безпосереднього затвердження з боку розробника. Усі зміни розгортаються автоматично.
- **Основні принципи CI/CD:**
 - **Сегрегація відповідальності зацікавлених сторін.** Учасники процесу розробки та споживачі готового проєкту поділяють між собою відповідальність за ту чи іншу стадію життєвого циклу продукту. Розробники та дизайнери проєктують

бізнес-логіку, а також забезпечують позитивний досвід користувача взаємодії з готовою системою. Інженери за якістю вводять наскрізні функції та приймальні тести, DevOps-інженери організують логістику коду, а користувачі дають зворотний зв'язок за результатами використання системи..

- **Зниження ризику.** Кожна група учасників процесу розробки мінімізує можливі ризики при проходженні продукту через стадії життєвого циклу (контроль цілісності бізнес-логіки, досвіду користувача, оптимізація зберігання та обробки даних, міграції та ін.).
- **Короткий цикл зворотного зв'язку.** У комерційної розробки швидкість реакцію виникнення помилок, чи запитів нового функціоналу закладає основу конкурентоспроможності майбутньої системи. Щоб додавати в продукт новий функціонал швидше за конкурентів, необхідно прагнути до автоматизації складання та тестування коду. Однак, у ситуаціях, коли для вирішення потрібна участь людини, автоматизація може лише нашкодити. Для таких ситуацій рекомендується скорочувати кількість інформаційних посередників, забезпечуючи короткий цикл зворотного зв'язку.
- **Оточення (Environment).** У розробці програмного забезпечення під середовищем розуміється рівень сервера, призначений для певної стадії процесу випуску.
 - У розробці програмного забезпечення є в основному 4 середовища, а саме: **Середовище розробки (Development)** – тут розробляється програмне забезпечення. У деяких ситуаціях це може бути робочий стіл розробника, в інших випадках це може бути сервер, який спільно використовується кількома розробниками, що працюють разом над одним проектом. Це середовище має максимально схоже на Production середовище, щоб запобігти проблемам, при яких програмне забезпечення в Production середовищі діє інакше.
 - **Тестове середовище (Test)** - після того, як програма була розроблена до узгодженої стадії, вона випускається в тестове середовище. Тут тестувальники перевіряють якість програми, виявляють помилки та переглядають виправлені помилки. Це середовище має точно нагадувати Production середовище, тому що це останнє безпечне місце для пошуку та виправлення помилок, пов'язаних із середовищем.
 - **Проміжне середовище (Staging).** Може називатися Pre-production або UAT. Використовується як останній рубіж тестування та огляду нових версій веб-сайту перед його запуском у Production. На цій стадії програмне забезпечення часто тестується на обладнанні, яке відображає обладнання, що використовується в середовищі Production.
 - **Виробниче середовище (Production)** – тут програма виходить у світ і стає доступною для користувачів.
- **Quality Gates** — набір автоматизованих перевірок, вбудованих у devops-пайплайн кожної системи.
- **Мета Quality Gates** – гарантувати, що кожен проект слід набору певних правил і передових методів, щоб запобігти загальним ризикам проекту та збільшити шанси на успіх.
- **Безперервне тестування (Continuous Testing)** — це підхід, який спрямований на забезпечення якості в умовах підвищених швидкостей. Він має на увазі максимальне покриття функціональними автоматичними тестами, які запускаються максимально часто.

Закріплення матеріалу

- Що таке CI/CD?
- Навіщо потрібно налаштувати CI?
- У чому різниця між двома останніми CD в CI/CD/CD?

- Яка задача UAT оточення?
- Які ризики зростають за відсутності будь-яких Quality Gates ?

Самостійна діяльність учня

Завдання 1

Намалюйте найповнішу схему постачання продукту, яку зможете вигадати, починаючи комп'ютером розробника і закінчуючи Production оточенням. Не забудьте про Quality Gates.

Завдання 2

Наведіть приклади переваг та недоліків "Безперервної доставки (CD)" та "Безперервного розгортання (CD)"? Чи є ситуації, коли краще обмежитися першим етапом і не модифікувати процес до безперервного розгортання?

Рекомендовані ресурси

Continuous Testing

<https://www.tricentis.com/products/what-is-continuous-testing/#:~:text=Continuous%20Testing%20is%20the%20process,candidate%20as%20rapidly%20as%20possible.>

Continuous Testing

<https://www.broadcom.com/products/software/continuous-testing>