

# Модуль 1. Введение



Каждый день на рынке появляется огромное количество новых цифровых продуктов.

Однако существенно сократить time-to-market позволяют не только современные инструменты для разработки, в том числе [облачные решения](#), но и эффективные методологии. Рассказываем простыми словами, что такое DevOps и как этот подход помогает компаниям автоматизировать создание приложений.

## Что такое DevOps

До появления DevOps процесс разработки строился по принципу каскада, в котором все этапы были четко зафиксированы. На старте определялись требования к продукту, затем наступал черед проектирования, после него — написание и сборка кода, тестирование и запуск.

- Определение требований
- Проектирование
- Написание и сборка кода
- Тестирование
- Инсталляция
- Поддержка

Сегодня каскадная модель уже потеряла свою эффективность. Вот основные причины:

- **Низкая скорость разработки**

Перейти на следующий этап можно только по завершении предыдущего. Любая ошибка — и все надо начинать заново.

- **Неэффективное взаимодействие между разработчиками, тестировщиками и администраторами**

При каскадном подходе разработчики могут вообще не знать, что происходит с их кодом после сборки. Тестировщики далеко не всегда уверены, что работают с актуальной версией приложения. А администраторам и эксплуатация зачастую не хватает четких данных по требованиям к конфигурации серверов.

- **Высокий уровень зависимости отделов друг от друга**

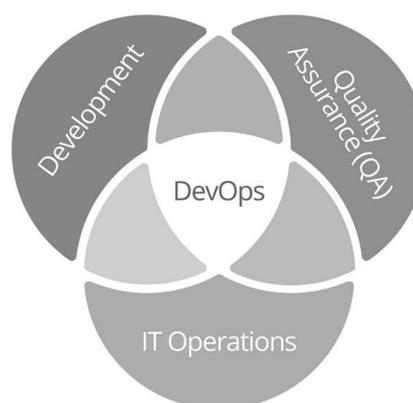
Нередко возникают ситуации, когда одна команда вынуждена сидеть без работы, так как другая еще не закрыла свои задачи. Например, пока разработчикам не передадут обновленные требования пользователей, они не могут работать над новой версией продукта.

Решить эти проблемы и повысить эффективность процесса разработки призвана методология DevOps. В ее основе лежит идея, что разработка, тестирование и эксплуатация продукта — отнюдь не разрозненные этапы, а единый циклический процесс.

## Методология DevOps

Призвана повысить эффективность процесса разработки

**В ее основе лежит простая идея:** разработка, тестирование и эксплуатация продукта — не разрозненные этапы, а **единый циклический процесс.**



При каскадной модели команды неизбежно участвовали в цепочке «разработка новых фич → тестирование → устранение багов». Обновление приложения занимало очень много времени, и команды просто не успевали своевременно реагировать на запросы пользователей и потребности рынка.

Методология DevOps позволяет создавать цифровые продукты в разы быстрее: все шесть этапов идут параллельно друг с другом.

### **1. Проектирование**

Владелец продукта указывает требования к приложению, а команда разработки определяет, какой должна быть его структура. Каждое подразделение фиксирует этапы, в которых принимает участие, и время, которое на них будет затрачено.

### **2. Разработка**

Под руководством DevOps-инженера (которого нередко называют просто «девопс») команда создает окружение и конвейер непрерывной разработки и доставки (CI/CD), где и будет идти разработка продукта. Пишутся скрипты и части кода. Отдел эксплуатации и тестировщики настраивают отдельные кластеры для разработки, тестов и продакшена.

### **3. Запуск конвейера CI/CD**

На этом этапе запускают в работу скрипты, которые автоматизируют все рутинные задачи и работу с инфраструктурой: компиляцию кода, его передачу на тестирование и многое другое.

### **4. Тестирование**

Процесс полностью автоматизирован и идет непрерывно. После успешного прохождения автотестов приложение отправляется на продакшен-сервер.

### **5. Развертывание**

Как и тестирование, развертывание происходит непрерывно. Конфигурации, успешно прошедшие тесты, автоматически или полуавтоматически (в зависимости от подхода) отправляются на продуктив. Благодаря тому что все этапы заскриптованы, развертывание может происходить несколько раз в день без ручного вмешательства команды.

### **6. Непрерывный мониторинг**

Когда приложение попадает в продакшен-среду, к нему подключаются выбранные системы мониторинга. Они круглосуточно отслеживают работоспособность продукта, а если что-то идет не так — оповещают разработчиков.

## **Чем занимается DevOps-инженер**

Ключевую роль во внедрении методологии DevOps и обеспечении ее работы играет DevOps-инженер. В его задачи входит:

- участие в определении инфраструктуры приложения, выборе системы оркестрации и принципов его масштабирования;

- настройка серверов, проверка среды, автоматизация передачи кода;
- автоматизация процесса тестирования и деплоя.

Кроме всего этого DevOps-инженер берет на себя десяток (а то и несколько десятков) задач, которые помогают разработчикам, тестировщикам и администраторам эффективно работать сообща.

## Почему все больше компаний переходят на DevOps

Мир не стоит на месте, а ИТ — тем более. Компании стремятся все быстрее реагировать на изменения рынка и новые потребности клиентов. И DevOps — верный путь к этой цели.

### Кому будет полезно

DevOps упрощает жизнь всех участников процесса разработки цифрового продукта:

- Разработчик может целиком сконцентрироваться на написании кода и внедрении новых функций, не отвлекаясь на настройку инфраструктуры.
- Тестировщик получает больше возможностей при проверке приложения на различных конфигурациях и не тратит время на автотесты: они уже автоматизированы.
- С администраторов снимаются непрофильные задачи по развертыванию приложения и мониторингу его работоспособности.
- Бизнес сократит затраты на разработку и сможет эффективнее использовать бюджет, например перенаправить его на маркетинг и продвижение продукта.