

Что такое DevOps?

6 февраля 2026 г.

Правдин Иван Дмитриевич

- Выпускник ИБАС 2024 г.
- 4 года опыта в DevOps

Жизненный цикл ПО раньше

- Разработка (Development)
- Эксплуатация (Operations)

Жизненный цикл ПО сейчас

- Разработка (Development)
- DevOps
- Эксплуатация (Operations)

Между разработкой и эксплуатацией появился новый «слой» — он в центре нашего курса

Индустрия разработки ПО изменилась:

- Программы стали **сложнее**
- Работают как **веб-сервисы 24/7**
- Бизнес требует **быстрых обновлений**

Традиционные подходы не справляются с этими вызовами

DevOps - способ организации работы IT команд

Включает:

- Автоматизацию ручных процессов
- Сотрудничество разработчиков и админов
- Культуру совместной ответственности

Цель DevOps: быстрая и надежная доставка ПО

Зачем вам это?

1. Требуется работодатель
2. Делает вас эффективнее

Кейс 1: Пятничный релиз

Пятница. Вечер. В конце рабочего дня разработчик вручную устанавливает новую версию на прод. Через час пользователи жалуются на баги, а восстановить прошлую версию невозможно: она только на устройстве разработчика, который уже уехал.

Что делать?

1. Нужно заставить лучше тестировать
2. Надо было подождать до понедельника
3. Не хватает автоматизации
4. Баги — это нормально, в понедельник починим

Кейс 2: Медленная работа после релиза

Администратор устанавливает новую версию приложения на прод. Предварительно разработчик и тестировщик проверили эту версию на тестовом стенде. После релиза пользователи жалуются: приложение начало работать заметно медленнее. Оказывается, дело в новом функционале.

Что делать?

1. Нужно лучше тестировать, а лучше нанять ещё 2 тестировщиков
2. Автоматизировать настройку серверов
3. Купить более мощные сервера для продакшена
4. Разработчик должен писать больше тестов

Кейс 3: Закончилось место на диске

Разработчик сделал новый функционал, который увеличивает использование диска. Администратор не знал про новую фичу. Через два дня приложение падает — место закончилось.

Что делать?

1. Ввести правило: все новые фичи должны согласовываться с Ops за 3 дня
2. Закупить «железо» про запас
3. Настроить мониторинг и квотирование
4. Администратор должен смотреть все changelogs перед обновлением стенда

Кейс 4: Важная фича и окно релиза

Разработчик за два дня сделал фичу, которая, предположительно, принесет много новых клиентов компании. При этом ближайшее окно релиза (окно развертывания) через 3 недели.

Что делать?

1. Разрешить релизить важные фичи вне окна релиза
2. Мобилизовать тестировщиков с других активностей
3. Ввести ежедневные планерки для пересмотра приоритетов
4. Вообще убрать окна релиза и релизить по желанию

Неожиданно сайт перестает работать. Выяснилось, что кто-то удалил таблицу users в базе данных. Резервная копия устарела.

Что делать?

1. Разрешить работать с prod базой только «роботам»
2. Поменять все пароли и сообщить их только некоторым синьорам
3. Вести список провинившихся разработчиков
4. Нанять отдельного специалиста DBA

Кейс 6: Разработчик ушел в отпуск

Разработчик завершил выпуск новой функции приложения, но на следующий день ушёл в планируемый отпуск. Остальные сотрудники не знают, как её тестировать, настраивать и использовать, из-за отсутствия документации.

Что делать?

1. Внести в общие правила пункт «Не уходить в отпуск без документации»
2. Нанять отдельного техписателя для написания документации
3. Ввести в KPI пункт «покрытие документацией»
4. Задача выполнена только тогда, когда она в prod и протестирована

Вопрос: если DevOps - это культура для команды,
то зачем существует должность
«DevOps инженер»?

DevOps — это культура и практики для всей команды

Но на практике кто-то должен:

- Настраивать и поддерживать автоматизацию
- Внедрять практики автоматизации
- Быть связующим звеном между Dev и Ops

DevOps инженер — специалист, который помогает командам применять DevOps практики

- DevOps решает проблемы современной разработки
- Это культура + технические практики
- DevOps инженер — координатор, не исполнитель

Лекции:

1. Введение в DevOps
2. Прокси и туннелирование
3. Git в DevOps
4. CI/CD: концепции и инструменты (GitHub Actions)
5. Configuration Management (Ansible)
6. Контейнеризация (Docker)
7. Оркестрация контейнеров (Docker Compose)
8. Yandex Cloud, Baked Configuration (Packer)
9. Infrastructure as Code (Terraform)

Лабораторных работы:

1. GitHub webhooks
2. GitHub Actions
3. Docker
4. Docker Compose