**DEVOPS МИФЫ И ЗАБЛУЖДЕНИЯ**

Как вы считаете, что такое «DevOps»? Кто-то скажет, что это некое маркетинговое изобретение и ничего больше. А кто-то может думать, что это определенный «сборник» полезных инструментов для разработки. Исходя из этого, получаем огромное множество различных трактовок, порождающие в дальнейшем устойчивые мифы. Что же это за зверь и с чем его едят, мы разберемся в материале, состоящем из двух частей.

**Миф первый: DevOps – это вещь, для которой нужен DevOps-отдел со своими инженерами**

Итак, мы подошли к первому и самому яркому примеру непонимания термина «DevOps». Почему-то, многие руководители компаний считают, что для модернизации методов **управления ИТ проектами** нужно создавать отдельный отдел. И назвать его нужно «DevOps-отделом», куда нанимаются соответствующие инженеры.

Мы не будем объяснять, почему DevOps-инженеров и DevOps-отделов не существует. Потому что в России все равно продолжают следовать этой практике и судя по количеству вакансий на вышеуказанную должность, тенденция не будет снижаться.  
  
Основная проблема заключается в том, что ради **автоматизации процессов разработки ПО,** руководители начинают искать что-то новое. И как только они сталкиваются с новомодными словами Agile или DevOps, то сразу бегут создавать новые отделы с соответствующими названиями. Вы можете зайти в любую ИТ-компанию в России и увидеть, как из отдела по контролю за качеством сформировали DevOps-отдел.

Более того, сами сотрудники не совсем понимают, что такое DevOps, обращаясь за просветлением в Википедию. Ведь по сути, это метод, который должен помочь компании перейти на непрерывность разработки, объединив в себе все этапы.

Если вы не верите тому, что компании неправильно используют метод **управления разработкой ПО** и не понимают, что это такое, то просто загляните на доску объявлений. Абсолютно все вакансии на место DevOps-инженера разнятся до неузнаваемости!

Небольшое просветление наблюдается у небольших компаний, которые не могут позволить себе большой штат, а потому нуждаются в наиболее универсальных сотрудниках. Таким образом, неосознанно, они реализуют у себя принцип непрерывной поставки ПО, нанимая DevOps-инженеров. И то, что работодатели часто пишут о технологиях с ошибками, лишний раз подтверждает это.

Конечно, создание DevOps-отдела – право каждого. Главное, чтобы сама суть метода сохранялась и ей следовали абсолютно все сотрудники, работающие ради разработки качественного ПО.

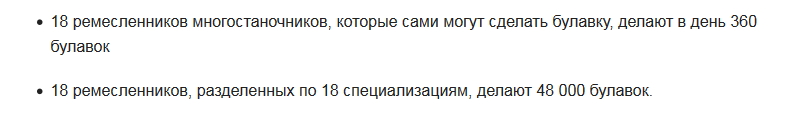
Стоит отметить, что впервые о DevOps сказал Патрик Дебуа,который создал целое сообщество для приверженцев этой идеи: решения проблем производства ПО. Но в России почему-то поняли это не так, считая, что DevOps – это человек, который предоставляет решение, опираясь на данный метод.

И на заметку: если вам нужен DevOps-инженер, лучше наймите обычного. Почему? Посмотрите на ценник равнозначных специалистов, но с различной приставкой. Обычный инженер запросит около 90 000 рублей, когда второй — целых 150 000. Профит! А если вам захотелось организовать DevOps-отдел, то подумайте сначала о том, чем конкретно он будет заниматься и не усложнит ли вам жизнь.

**Миф второй: С DevOps методом смогут работать только инженеры-универсалы**

Начнем с того, что в самом начале, многие так и работали. Разработчик выступал не только в той роли, ради которой он и учился, но и во многих других: он сам настраивал, сам анализировал, сам подключал абсолютно другие технологии. А все потому, что никто и подумать не мог, что разделение на профили поможет ускорить разработку и дать возможность компаниям **автоматизировать процесс разработки ПО**.   
  
Но нанимать универсальных инженеров – это ошибка!

Вот вам простой пример:



В этой простой истине кроется ключевой аспект в **управление разработкой ПО**: для успешного производства нудна кросс-функциональная команда, а не универсальные специалисты. В обратном случае, вам попросту не будет хватать людей для больших объёмов работ.  
  
Но у нас устоялось мнение, что универсальный специалист, этакий мастер на все руки, это панацея от всех бед. Потому что все привыкли ставить задачи, но мало кто может решить их. В частности, это относится к DevOps. Как понять, какой инженер и какой специализации может понадобиться в этом направлении? Попробуем разобраться:

* **Разработчик-тестировщик, который способен анализировать архитектуру и работу софта в целом**

Вы можете подумать, что это специалист, который немногим отличается от выпускника технического ВУЗа. Но на самом деле, такой инженер может не только разработать алгоритм, но и воплотить его в жизнь для конкретной задачи.

Руководители заблуждаются в самом понимании определения «готовый продукт» и считают, что сама схема работы – это и есть решение. Но это не так. DevOps потребует от такого инженера способностей к постановке и разъяснению конкретных задач, которые должны выполнить остальные, а также дать им четкий алгоритм для работы. После этого, все смогут сопровождать продукт в продакшене, так как будут иметь свои распределенные роли.

* **Инженер инфраструктуры, который будет создавать платформу для разработки**

Довольно разнообразная должность, на которой инженер должен будет реализовывать платформу для разработки, хорошо разбираясь в системах управления конфигурациями. То есть, по сути, он управляет созданием ПО, предоставляя для этого соответствующую площадку.   
  
С помощью знаний о Docker или Kubernetes, инженер сможет организовать **автоматизацию процесса разработки ПО**, делая его непрерывным и передавая следующий этап разработчикам .

* **Инженер-разработчик для инфраструктурных сервисов**

Речь идет о понимании DBaS, Logging as Service и т.д. Это своего рода готовый продукт, с которым продолжают работу сами другие разработчики.

По сути, реализацию такого распределения ролей можно наблюдать у «Амазон». Они продают решения, которые вы можете создать и сами.

* **Менеджер проектов**

Нет, это не тот инженер, который будет организовывать **управление ИТ-проектами.** Это специалист, который будет контролировать зависимости между этапами. Более того, в его обязанности должны входить контроль версий, он будет фасилитировать группы людей для интеграции, а также следить за непрерывной поставкой ПО.

**Миф третий: У нас корпоративная IT-система, которая использует DevOps с 1995 года**

Многие читают, что они следуют принципам DevOps уже много лет. А новомодные течения – это лишь интерпретация того, что придумано давно.

Но самое интересное то, что в таких компаниях зачастую действительно используется процесс **управления разработкой ПО,** но проявляется он лишь в критичных случаях. Как же так вышло?

Проблема в том, что руководители считают непрерывный процесс разработки инструментом, который будет давать им некоторое количество релизов в определенный отрезок времени. И да, чем чаще выпускается продукт, тем лучше. Но это не совсем DevOps.

Существует множество циклов разработки в различных группах инженеров. Релиз конечного ПО может занимать месяцы, а сам процесс деливера может происходить и каждую неделю. Чувствуете разницу? Понятия спутали.  
  
DevOps – это метод для обеспечения **time-to-market**. И это не означает конкретное количество доставок продукта за единицу времени. С точностью до наоборот. Это время, которое необходимо для конечного выпуска ПО, начиная с момента создания первой строки кода.

Именно time-to-market обеспечивает рост популярности DevOps. Но, к сожалению, у нас его понимают по-другому. Приведем правильные способы реализации данного метода:

* **Стратегия релиза**

Нужно задумываться над способом релиза и когда он будет происходить. Если решить выпускать продукт частично для 0,1% юзеров, это приведет к необходимости создавать микросервисную архитектуру. И хорошо, если вы подумали об этом заранее.

* **Цикличный мониторинг**

Монитринг должен выступать не только, как инструмент для анализа, но и как самостоятельный способ для тестирования продукта. Разработчик сам определяет необходимые части ПО, которые будут нуждаться в этом и сможет наблюдать за релизом и дальнейшее работой продукта.   
  
Опять же, если вы решили выкатить софт для 0,1% юзеров, то будьте готовы мониторить его состояние. Иначе, смысл ограничивать выпуск?

* **Сквозное логирование**

Сквозное логирование необходимо для **автоматизации процесса разработки ПО,** так как помогает быстро находить причины неполадок, за счет записи каждого шага пользователя и действий самой программы. Только представьте, что бы вам пришлось делать, если бы все 0,1% юзеров начали сообщать об ошибках, а у вас нет логирования.

* **Автоматическое тестирование**

Это не только способ улучшить качество выпускаемого ПО, но и форма договоренности между командами. То есть, группы разработчиков смогут изолировать себя от необходимости проводить тестирование созданного ими кода. Они не будут терять время на лишние проверки, потому что автоматизировали этот процесс заранее!

## Миф четвертый: DevOps — это набор нужных инструментов

Что мы подразумеваем под словом «нужный» или «эффективный»? DevOps – это не определенная формула к успеху и не набор полезных инструментов. Каждая задача нуждается в определенном подходе.

Например, автоматические тесты – это результат договоренности между командами для ускорения и автоматизации. Но это никакой не инструмент. Да и тесты бывают разные, а про способы их написания мы вообще молчим.  
  
Ещё один яркий пример – это Kubernetes, с помощью которого многие компании пытаются внушить себе, что используют DevOps. Но позже они сталкиваются с тем, что ПО не всегда стабильно, оно сложное, да и вообще, не так уж и необходимо.

Так что, DevOps — это не набор нужных инструментов. Скорее, существуют правильные инструменты для реализации данного метода. А их, кстати, можно найти в сообществах DevOps.

**Миф пятый: DevOps — это исключительно западная идеология, которая не может быть реализована на отечественном рынке**

Да-да, и такие мнения имеют место. Но правда в том, что любая технология где-то берет свое начало. Она рождается в коллективе. Но это не значит, что она неприменима где-то ещё.

Возможно, некоторые принципы работы не только DevOps, но и других методов, больше подходят реалиям западного мира. Но не стоит забывать, что DevOps – это не только идеология. Существуют эффективные инструменты, которые создавались и создаются для его реализации.

**Управление разработкой ПО** никак не зависит от культуры. Даже если что-то нужно адаптировать, то надо с чего-то начать. Если не пробовать построить работающий механизм для себя, то он так и останется исключительным для особенных. Философия – это не суть DevOps, а лишь её часть, которая помогает людям лучше понять принцип работы.

**Миф шестой: ITIL включает в себя DevOps, поэтому внедрять его как нечто новое – это глупость.**

С одной стороны, так и есть, но если следовать такой логике, то ITIL можно отнести к любому производству на заводе. Везде есть свой инцидент-менеджмент или организация управлением изменениями. По сути, это лишь обобщение, которое является основополагающим принципом в инженерии.   
  
Но правда в том, что современный рынок IT, который включает в себя именно разработку цифровых продуктов, модернизировал вышеуказанный принцип под себя. То есть, оболочка та же, но содержимое изменилось.   
  
С другой стороны, ITIL изначально создавался немного для других целей, поэтому не может использоваться для эффективного управления ИТ-проектами.

**Миф седьмой: DevOps — это лишь маркетинг**

Возможно, так начали думать из-за массовой «болезни» руководителей, который вдруг решили заDevOps’нуть все, что видят. В результате, мы получили огромное количество соответствующих отделов и специалистов.

Но на самом деле, в мире регулярно проводятся DevOps-собрания, целью которых является модернизация метода **автоматизации процесса разработки ПО**. Это не собрание людей ради сетевого маркетинга.

Люди, которые собираются на тех же DevOps Days, ищут способ решить ту или иную проблему. Не только с точки зрения менеджмента, но и со стороны разработки. Какой эе это маркетинг?

Когда будет желание, загляните на эти русскоязычные ресурсы, и вы поймете, что хайп, маркетинг и вендор – это не про DevOps: