

«Книга для тех, кто не боится инноваций и хочет вывести рабочие процессы компаний на качественно новый уровень».

Елена Лукутина, директор по операционной деятельности и технологическому развитию «Неофлекс»

# Руководство по DevOps

Джен Ким  
Патрик Дебуа  
Джон Уиллис  
Джез Хамбл



Как добиться гибкости, надежности и безопасности  
мирового уровня в технологических компаниях

[Купить книгу на сайте kniga.biz.ua >>>](http://kniga.biz.ua)

# Оглавление

Предисловие кроссийскому изданию .....	9
Введение .....	11
Предисловие.....	21
Вступление     Как будет выглядеть мир, если разработка и эксплуатация пойдут по принципу DevOps.....	23
 Часть I «Три пути»	
Введение .....	49
Глава 1 Agile, непрерывная поставка и «три пути».....	55
Глава 2 Первый путь: принципы потока .....	64
Глава 3 Второй путь: принципы обратной связи.....	80
Глава 4 Третий путь: принципы непрерывного обучения и экспериментирования.....	92
 Часть II Откуда начать	
Введение .....	109
Глава 5 Как выбрать стартовый поток создания ценности .....	111
Глава 6 Основные сведения о работе в потоке создания ценности, превращении его в прозрачный и расширении на всю организацию .....	124

[<<> Купить книгу на сайте kniga.biz.ua >>>](http://kniga.biz.ua)

Глава 7	Как проектировать организацию и ее архитектуру, не забывая о законе Конвея.....	144
Глава 8	Как получить лучшие результаты, интегрируя эксплуатацию в повседневную деятельность разработчиков .....	168

### Часть III

#### Технические практики потоков создания ценности

Введение .....	185	
Глава 9	Создание основы конвейера внедрения .....	187
Глава 10	Быстрое и надежное автоматизированное тестирование.....	201
Глава 11	Запустить и практиковать непрерывную интеграцию.....	227
Глава 12	Автоматизация и запуск релизов с низким уровнем риска.....	239
Глава 13	Архитектура низкорисковых релизов.....	273

### Часть IV

#### Второй путь: методики обратной связи

Введение .....	291	
Глава 14	Создайте телеметрию, позволяющую замечать проблемы и решать их .....	293
Глава 15	Анализируйте телеметрию, чтобы лучше предсказывать проблемы и добиваться поставленных целей .....	319
Глава 16	Настройте обратную связь, чтобы разработчики и инженеры эксплуатации могли безопасно разворачивать код.....	335

Глава 17 Встройте основанную на гипотезах разработку и А/В-тестирование в свою повседневную работу ..... 353

Глава 18 Создайте процессы проверки и координации для улучшения качества текущей работы ..... 363

## Часть V

### Третий путь: методики непрерывного обучения и экспериментирования

Введение ..... 387

Глава 19 Внедрите обучение в повседневную работу ..... 389

Глава 20 Преобразуйте локальные открытия в глобальные улучшения ..... 407

Глава 21 Выделите время для обучения и улучшений ..... 422

## Часть VI

### Методики интегрирования информационной безопасности, управления изменениями и контроля над соответствием нормам и требованиям

Введение ..... 437

Глава 22 Защита информации как часть повседневной работы всех сотрудников компании ..... 439

Глава 23 Безопасность конвейера развертывания ..... 465

Призыв к действию. Заключение ..... 482

Дополнительные материалы ..... 485

Приложения ..... 487

Дополнительная литература ..... 507

# Предисловие к российскому изданию

**В**первые о DevOps заговорили в связи с переходом в эру цифровой экономики, когда скорость выпуска на рынок продуктов стала одним из ключевых конкурентных преимуществ. Технологиям, обеспечивающим стремительное развитие бизнеса, пришлось бежать со всех ног, чтобы только оставаться на месте, а для достижения дополнительных результатов, как минимум, в два раза быстрее. Компаниям понадобились инструменты для быстрого и непрерывного улучшения качества существующих процессов разработки продуктов и их максимальной автоматизации, потому что хороший продукт стал равен хорошему ИТ.

Свой путь погружения в DevOps я начала несколько лет назад, когда возглавила отдел тестирования системы подготовки регулярной банковской отчетности Neoflex Reporting, которая отличалась большим количеством параллельных веток разработки и обилием ручных процессов. В ее разработку к этому моменту уже были вложены десятки тысяч человеко-часов.

Засучив рукава, наша команда взялась за точечную автоматизацию этапов жизненного цикла продукта. В целом мы достигли неплохих результатов, но добиться слаженной и синхронной работы от всех участников процесса оказалось по-настоящему трудной задачей. Периодически возникающие «тут подкрутить», «там вручную запустить», «а это не на моей стороне», «я был на обеде», «исторически сложилось» тормозили ожидаемое от автоматизации ускорение.

Осознать, что же делать дальше, помогла книга, которую вы сейчас держите в руках. Мы прочитали её всей командой и здорово переработали текущие процессы взаимодействия в парадигме слаженности, простоты и удобства. А процессы сборки,

## ПРЕДИСЛОВИЕ К РОССИЙСКОМУ ИЗДАНИЮ

развертывания инфраструктуры, установки, тестирования и выдачи поставки объединили в непрерывный производственный конвейер, вдохновленные идеей «все, что связано с кодом — тоже код». [ 10 ] Довольно быстро были получены ошеломляющие результаты: время выпуска обновлений с одного дня сократилось до десятка минут, а работа над продуктом Neoflex Reporting стала приносить профессиональное удовольствие.

«Руководство по DevOps» — книга об эффективном ИТ настоящего. Захватывающий и понятный путеводитель, способный обобщить, разложить по нужным полочкам существующий опыт и обогатить его цennыми идеями.

В книге описаны основные шаги и принципы построения производственного взаимодействия, автоматизации процессов и развития культуры разработки ПО. Теория щедро сдобрена историями реальных людей и компаний, прошедших непростой, но интересный путь к DevOps.

Неоспоримая ценность «Руководства...» в том, что оно помогает вырваться из рутины бытия и взглянуть на текущие процессы совершенно другими глазами. Приходит осознание того, что на точечных «костылях» автоматизации далеко не уйти, появляется понимание того, как выглядит путь роста и развития, который подходит именно вашей компании, проекту, продукту.

Желаю вам приятного чтения и пусть эта книга станет для вас источником неиссякаемого вдохновения!

*Лина Чуднова, руководитель практики DevOps компании «Неофлекс»*

## ВВЕДЕНИЕ

---

# «Ага!»

**П**уть к созданию книги «Руководство по DevOps\*» был долгим. Он начался в феврале 2011 г. с еженедельных переговоров по скайпу между соавторами. Мы решали, как создать руководство с рекомендациями — дополнение к книге The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win\*\*.

Прошло пять с лишним лет. Более двух тысяч часов работы. Книга «Руководство по DevOps» наконец завершена. В результате мы вознаграждены сполна, поскольку неожиданно обрели новое знание и поняли: сфера его применения гораздо шире, чем мы первоначально предполагали. Оно обеспечивает невиданные возможности. В конце концов мы сами воскликнули: «Ага!» — и нам кажется, что многие читатели разделят наше мнение.

### Джин Ким

Мне повезло: с 1999 г. я изучал организации, использующие высокопроизводительные технологии. Вот один из моих первых выводов: решающее значение для успеха имеет перекрестное взаимодействие функциональных групп, занимающихся эксплуатацией, информационной безопасностью и разработкой. Я до сих пор помню, как впервые осознал масштабы нисходящей спирали, в которую заключена деятельность этих групп с их противоположными задачами.

\* Акроним от англ. development и operations — методология разработки программного обеспечения, нацеленная на активное взаимодействие и интеграцию специалистов по разработке и специалистов по IT-обслуживанию. *Прим. перев.*

\*\* Ким Д., Бер К., Слаффорд Дж. Проект «Феникс». Роман о том, как DevOps меняет бизнес к лучшему. М.: Эксмо, 2015. *Прим. перев.*

## ВВЕДЕНИЕ

Это было в 2006 г., и мне тогда представилась возможность поработать целую неделю с группой, решавшей отанные [ 12 ] на аутсорсинг IT-задачи, поставленные крупной службой резервирования и продаж авиабилетов. Участники группы рассказали об увеличивающихся негативных последствиях ежегодных крупных обновлений программного обеспечения: каждый раз наступал настоящий хаос, шквал неудобств как для исполнителей, так и для заказчика. Из-за простоев у пользователей им приходилось выплачивать немалые компенсации согласно договорам по сервисному обслуживанию. Увольнялись наиболее способные и опытные работники, так как, опасаясь потерять прибыль, компания вынуждала их наращивать темп, выполнять массу незапланированной работы и «тушить пожары». У оставшегося персонала не хватало сил справляться со все возрастающим потоком требований заказчиков, желавших исправления ошибок. От расторжения сервисного контракта компанию спасали только героические усилия менеджеров среднего звена, и все были уверены: у контракта нет будущего, его не продлят на следующие три года.

Отчаяние и безнадежность подтолкнули меня к тому, чтобы начать нечто вроде наступательной операции. Разработка всегда рассматривалась как часть стратегии, а эксплуатация — тактики. Нередко они частично или даже полностью отдавались на аутсорсинг, чтобы лет через пять вернуться обратно, еще более усложнившимися.

Многие годы мы размышляли, как улучшить ситуацию. Вспоминаю, как на конференции Velocity Conference 2009 с интересом следил за обсуждением фантастических результатов, достигнутых благодаря использованию принципов бизнес-архитектуры, технических методов и норм корпоративной культуры в совокупности. Теперь эта методика известна нам как DevOps. Тогда я неподдельно вззволновался: передо мной налился путь выхода из создавшейся ситуации — его-то мы так долго искали. Стремясь распространить новое знание как можно шире, я и решил выступить соавтором The Phoenix Project. Вы запросто можете представить себе, какое огромное внутреннее

удовлетворение я испытал, видя видя отзывы людей о том, как книга помогла им прийти к озарению и воскликнуть: «Ага!»

[ 13 ]

## Джез Хамбл

Мое личное «Ага!» впервые раздалось в 2000 г., в стартапе, который был моим первым местом работы после окончания обучения. Некоторое время нас, технических специалистов, было только двое. Поэтому мне приходилось заниматься всем: сетями, программированием, поддержкой пользователей, системным администрированием. Мы выпускали ПО, размещая его на FTP прямо с рабочих станций.

В 2004 г. я перешел в консалтинговую компанию ThoughtWorks, где впервые принял участие в работе над проектом в составе команды численностью около 70 человек. Я входил в группу из восьми инженеров, занимавшуюся развертыванием нашей программы в среде, приближенной к производственной. Поначалу задание вызывало у нас сильный стресс. Но спустя несколько месяцев мы перешли от режима работы вручную, занимавшего около двух недель, к автоматическому разворачиванию продолжительностью всего один час. Теперь можно было за доли секунды откатывать конфигурации назад и вперед, используя технику «Blue-Green разворачивания» в рабочее время\*.

Проект породил массу идей, изложенных как в этой книге, так и в другой — «Непрерывное развертывание ПО»\*\*. Они убедили меня и многих моих коллег: с какими трудностями нам ни пришлось бы сталкиваться, мы способны действовать с максимальной отдачей и помогать людям.

\* Техника «Blue-Green разворачивания» — стратегия установки ПО, базирующаяся на двух идентичных инсталляциях промышленной системы, одна из которых активна, и возможно мгновенное переключение между ними. Одна из них условно называется синей, ее копия же называется зеленой. *Прим. перев.*

\*\* Хамбл Д., Фарли Д. Непрерывное развертывание ПО. Автоматизация процессов сборки, тестирования и внедрения новых версий программ.. М.: Вильямс, 2011. *Прим. перев.*

## Патрик Дюбуа

[ 14 ]

Для меня это была целая цепь событий. В 2007 г. я работал в проекте по миграции data-центра совместно с несколькими Agile-командами. Я завидовал их высокой продуктивности, умению выполнять большой объем работы за ограниченное время.

Получив следующее задание, я приступил к эксперименту по внедрению методики канбан в работу группы эксплуатации и увидел: группа стала быстро меняться. Позже, на конференции Agile Toronto 2008, я представил свой доклад в IEEE\* на эту тему, но, к сожалению, он не получил широкого отклика в Agile-сообществе. Мы начали создавать административную группу для системы Agile, но тут я переоценил значение человеческого фактора.

Увидев на Velocity Conference 2009 презентацию Джона Олспоу и Пола Хаммонда «то развертываний в день», я убедился: у меня есть единомышленники. Поэтому я решил организовать первую конференцию DevOpsDays и так случайно создал новый термин — DevOps.

Энергетика мероприятия была уникальной. Участники благодарили меня, утверждая, что конференция изменила их жизнь к лучшему. Так я осознал степень воздействия концепции DevOps и с тех пор неустанно продвигаю ее.

## Джон Уиллис

В 2008 г. я продал свою консалтинговую компанию, специализировавшуюся на внедрении крупномасштабных устаревших решений в области управления конфигурациями и мониторинга (Tivoli). Тогда же я впервые встретил Люка Каниса (основателя компании PuppetLabs). Люк выступал с презентацией о Puppet на проводившейся издательством O'Reilly конференции по конфигурационному управлению (CM) на основе открытого исходного кода.

\* Институт инженеров электротехники и электроники (от англ. Institute of Electrical and Electronics Engineers) — международная некоммерческая ассоциация специалистов в области техники, мировой лидер в области разработки стандартов по радиоэлектронике, электротехнике и аппаратному обеспечению вычислительных систем и сетей. *Прим. перев.*

Поначалу я скучал на заднем ряду лекционного зала, размышляя, что нового этот двадцатилетний парень может рассказать мне об управлении конфигурациями. Ведь я занимался этим всю жизнь, помогая крупнейшим корпорациям мира разрабатывать решения в области СМ и других сферах управления эксплуатацией. Однако через пять минут после начала доклада я уже сидел на первом ряду. Я тут же понял: все, что я делал за последние 20 лет, я делал неправильно. Люк описывал то, что я сейчас называю вторым поколением СМ.

[ 15 ]

После доклада мне удалось поболтать с ним за чашечкой кофе. Я был совершенно восхищен идеей, сейчас называемой «инфраструктура как код». Люк увлекся и стал подробно объяснять, что имеет в виду. Он верит, что эксплуатация становится похожей на разработку программ. Специалисты отдела эксплуатации хотят, чтобы конфигурации проверялись системой контроля качества и чтобы в рабочий процесс были адаптированы методики обеспечения CI/CD\*. Поскольку я к тому времени уже немало проработал в области эксплуатации IT, я ответил ему примерно так: «Это то же, что пытаться заставлять управленцев петь как Led Zeppelin».

Я жестоко ошибался.

Примерно через год на другой конференции Velocity, проводившейся в 2009 г. O'Reilly, я увидел презентацию Эндрю Шефера по инфраструктуре Agile. В ней он показал ставший каноническим рисунок — метафорическую стену между разработчиками и инженерами эксплуатации, через которую они перебрасывают друг другу рабочие задания. Он назвал это «стеной неразбирахи». Идеи, высказанные в презентации, в систематизированном виде выражали то, что Люк пытался рассказать мне годом ранее. Для меня это стало откровением. В том же году меня, единственного из американцев, пригласили на первую конференцию DevOpsDays в Генте. Ко времени окончания конференции идея, ныне получившая название DevOps, полностью овладела моим разумом.

\* Continuous Integration / Continuous Deployment — непрерывная интеграция и непрерывное развертывание. *Прим. ред.*

Из всего вышесказанного ясно, что соавторы книги пришли к единому выводу, хотя шли к нему разными путями. Реальность [ 16 ] доказала, что описанная проблема существует практически везде и решения с помощью DevOps применимы повсюду.

Цель написания этой книги - показать, как воспроизвести DevOps-трансформации, частью которых мы были или которые мы наблюдали со стороны. Еще мы хотим развеять множество мифов о том, почему DevOps не будет работать в тех или иных ситуациях. Нам довелось слышать примерно следующее.

**Миф 1:** *DevOps пригоден только для стартапов.* Методы DevOps впервые были применены единорогами интернет-индустрии: Google, Amazon, Netflix и Etsy. Каждая из компаний в определенные моменты своей истории рисковала выпасть из бизнеса из-за проблем, обычно возникающих в традиционных организациях (их еще называют рабочими лошадками экономики). Это опасные релизы, приводящие компанию к катастрофическому провалу, неумение быстро проводить изменения продукта или сервиса, чтобы превзойти конкурентов в новой области, проблемы с соблюдением нормативных требований, неспособность масштабироваться, высокая степень недоверия между разработкой и эксплуатацией и так далее.

Однако каждая из названных организаций смогла преобразовать свою архитектуру, технические методы, производственную культуру и достичь выдающихся результатов благодаря DevOps. Как язвительно заметил известный американский специалист по информационной безопасности Бранден Вильямс, «пусть больше не будет разговоров о пегасах или рабочих лошадках в DevOps, пусть останутся только чистокровные скакуны, а остальные отправятся на мыловарню в качестве сырья».

**Миф 2:** *DevOps заменяет собой Agile.* Принципы и методы DevOps совмещаются с Agile, причем многие отмечают, что DevOps — логическое продолжение Agile. Agile часто оказывается эффективным катализатором DevOps, поскольку предпочитает организовывать деятельность небольших команд, непрерывно поставляющих пользователям код высокого качества.

Многие практики DevOps возникают, если мы продолжаем управлять нашей работой за пределами цели «код, потенциально пригодный для релиза» в конце каждой итерации, расширяя ее до того, чтобы наш код всегда находился в развертываемом состоянии, а разработчики ежедневно синхронизировали свои изменения с основной веткой кода и могли продемонстрировать новые изменения в окружениях, близким к реальным. [17]

**Миф 3:** *DevOps несовместим с ITIL.* Многие рассматривают DevOps как ответ на ITIL или ITSM (управление IT-инфраструктурой компании). Его описание было впервые опубликовано в 1989 г. ITIL сильно повлияла на несколько поколений практиков в области управления инфраструктурой, включая одного из авторов этой книги. Это постоянно развивающаяся библиотека методов, позволяющих кодифицировать процессы и практики, лежащие в основе признанных во всем мире способов управления IT, связывающих воедино стратегию услуг, разработку и поддержку.

Методы DevOps можно сделать совместимыми с процессами ITIL. Однако для обеспечения более короткого цикла разработки и повышения частоты развертываний, предложенных DevOps, многие области процессов ITIL должны быть полностью автоматизированы. Следует решить проблемы, связанные с процессами управления конфигурациями и релизами (например, как обеспечивать постоянную готовность базы управления и библиотеки программ). DevOps требует, чтобы возникающие ошибки быстро обнаруживались и устранялись, поэтому дисциплина применения ITIL в проектировании архитектуры, обработке сбоев, решении проблем остается актуальной, как и всегда.

**Миф 4:** *DevOps несовместим с требованиями информационной безопасности.* Отсутствие традиционных методов контроля (например, разделение ответственности, изменение процессов проверки кода, проверка безопасности вручную по окончании проекта) может вызвать тревогу у специалистов по безопасности.

Однако это не означает, что в организациях, использующих DevOps, отсутствует эффективный контроль. Вместо работ по обеспечению безопасности и проверки соответствия требований ИБ, проводящихся на завершающем этапе проекта, необходимые проверки интегрированы в каждую стадию ежедневного цикла на протяжении всего цикла разработки. В результате обеспечиваются более высокое качество и безопасность.

[ 18 ]

**Миф 5:** *DevOps означает отсутствие необходимости управления IT-эксплуатацией, то есть NoOps (дословно — «нет эксплуатации»).* Многие неправильно трактуют DevOps как полное исключение необходимости IT-эксплуатации. Однако такое утверждение редко бывает справедливо. Хотя характер оперирования может измениться, само управление остается важным как никогда. Просто оно на гораздо более ранних этапах жизненного цикла ПО взаимодействует с разработкой, продолжающей действовать параллельно с IT-эксплуатацией еще долго после того, как разработанный код развернут в производственной среде.

Вместо того чтобы отдел эксплуатации разгребал поступающие заявки вручную, DevOps дает разработчикам возможность делать большинство операций через API и самообслуживающиеся платформы, такие как: создание среды, тестирование и развертывание кода, получение и отображение метрик о ПО и т.д. Когда реализован такой подход, IT-эксплуатация становится похожей на процесс разработки (то же справедливо для управления качеством и обеспечения информационной безопасности), сцепленный с разработкой продукта, где под продуктом понимается платформа, используемая, чтобы надежно, быстро и безопасно тестировать IT-сервисы, развертывать их и запускать в производственной среде.

**Миф 6:** *DevOps — это просто реализация подхода «инфраструктура как код».* Хотя многие из практик и подходов DevOps, приведенных в этой книге, требуют автоматизации, для реализации DevOps также необходимо изменение архитектуры системы и культуры производства, дающее возможность достичь общих

целей в ходе работы по повышению создаваемой ценности ИТ. Это выходит далеко за рамки простой автоматизации. Как написал Кристофер Литл, один из самых первых летописцев DevOps, «это [19] не автоматизация, так же как астрономия — это не телескопы».

**Миф 7:** *DevOps применим только к программам с открытым исходным кодом.* Хотя многие случаи успешного внедрения DevOps действительно имели место в организациях, использовавших ПО, входившее в группу LAMP (Linux, Apache, MySQL, PHP), имелись истории успеха, и не зависевшие от использованных технологий. Например, в приложениях, написанных на Microsoft.NET, коболе, языке ассемблера мейнфреймов, а также в системах SAP и даже в коде встроенных систем (например, микропрограммное обеспечение принтеров HP LaserJet).

## «Ага!» еще несколько раз

Каждый из авторов воодушевился удивительными инновациями, реализованными сообществом DevOps, и достигнутыми результатами: были созданы надежные системы, позволившие небольшим командам быстро и независимо разрабатывать и проверять код, который может быть без риска развернут у заказчика. Учитывая нашу убежденность, что DevOps — воплощение методов создания динамичных, обучающихся организаций, постоянно укрепляющих культуру высокого доверия, представляется неизбежным, что эти организации будут продолжать инновации и выйдут победителями на рынке.

Мы искренне надеемся, что книга «Руководство по DevOps» станет ценным источником. Как руководство по проведению DevOps-трансформации. Как набор практических примеров для накопления опыта. Как летопись истории DevOps. Как средство для организации коалиции и достижения общих целей владельцев продукта, архитекторов, разработчиков, инженеров контроля качества, эксплуатации и информационной безопасности. Она подскажет, как получить максимальную поддержку

## ВВЕДЕНИЕ

со стороны руководства при внедрении инициатив DevOps, как сформировать нравственный императив для изменения способов [ 20 ] управления технологическими организациями при обеспечении высокой эффективности. Она поможет создать более оживленную и дружелюбную рабочую среду, чтобы любой участник смог учиться в течение всей жизни — это не только поможет каждому исполнителю достичь целей, но и приведет организацию к победе.