

ГЛАВА 5

Заблуждения и антишаблоны, относящиеся к devops

Когда идет речь о devops, будет не лишним обсудить, что не относится к этой концепции. В результате этого обсуждения выявятся некоторые очевидные заблуждения или идеи, не имеющие отношения к devops. В этой главе рассматривается дополнительный контент, касающийся devops, а также некоторые общие антишаблоны.

Общие заблуждения, связанные с devops

В индустрии разработки ПО часто встречаются неверные представления о devops. Поэтому группы сотрудников, сформированные в вашей организации, могут и должны бороться за выработку ясных и точных убеждений и ценностей, имеющих отношение к devops. В этом разделе мы рассмотрим некоторые проблемы, с которыми обычно сталкиваются при попытке внедрения devops в организации.

Devops — это лишь разработчики и системные администраторы

Несмотря на то что название «devops» произошло от слов *development* (разработка) и *operations* (эксплуатация), оно скорее является напоминанием об источниках происхождения движения devops, чем строгим определением. И хотя на конференциях devopsdays звучит слоган «Конференция, которая объединяет разработку и эксплуатацию», на самом деле концепции и идеи devops охватывают все роли в организации. Не существует единого исчерпывающего списка, регламентирующего, как и какие люди и команды должны быть вовлечены в движение devops. Также отсутствует единый универсальный подход к внедрению devops.

Идеи, которые помогают группам разработчиков и специалистам эксплуатации лучше общаться и более эффективно работать вместе, могут применяться на

уровне компании в целом. Можно повысить эффективность любой группы в составе организации, будь то служба безопасности, отдел контроля качества, отдел эксплуатации или юридический отдел. Например, эффективные devops-процессы, внедренные на уровне юридического и коммерческого отделов, позволят реализовать автоматическое формирование контрактов на основе постоянного обновляемого каталога товаров.



Обратите внимание, что благодаря использованию принципов devops играют любые две или большее число команд, поэтому не ограничивайте потенциальную аудиторию пользователей devops и не изолируйте группы сотрудников. В части III будут изложены соображения по эффективному привлечению групп сотрудников к использованию devops.

Devops — это команда

В общем случае создание специальной «devops-команды» вряд ли можно считать удачным решением. Обратите внимание, что ни создание команды под названием «devops», ни присвоение имени «devops» существующей команде не является ни необходимым, ни достаточным условием для формирования devops-культуры. Если ваша организация пребывает в таком состоянии, что члены команд разработчиков и эксплуатации не могут общаться друг с другом, создание дополнительной команды, ответственной за налаживание общения, приведет лишь к появлению дополнительных проблем. Чтобы внедрить значительные и долговременные изменения, сначала нужно решить фундаментальные проблемы.

Формирование отдельной команды в среде, в которой запускаются новые процессы и коммуникационные стратегии, может быть эффективным в тех случаях, когда проект стартует «с нуля». В больших компаниях формирование отдельной devops-команды обычно осуществляется на короткий срок. Основная задача подобной команды заключается во внедрении значительных изменений. Со временем, выполнив задачи по инициированию изменений, члены вновь созданной команды возвращаются обратно в свои группы.

Наличие в стартап-среде единственной команды, выполняющей обе упомянутые выше функции, является целесообразным в том случае, если она берет на себя ответственность за разработку ПО и выполняет миссию по его эксплуатации. При этом в команде должно быть налажено сотрудничество между участниками, позволяющее быстро решать возникающие проблемы, не прибегая к экстренному вызову отдельных профессионалов. В этом случае все равно потребуются менеджмент, позволяющий облегчить четкое распределение ролей и обязанностей. Это позволит выполнять масштабирование группы по мере роста компании в целом.

В этой книге рассматриваются различные возможности, используемые при организации команд, и стратегии, обеспечивающие коммуникации и координирование деятельности отдельных групп. В конечном счете следует помнить, что не существует

единственного верного или неправильного способа внедрения devops. И если в вашей организации уже сформирована команда, в названии которой присутствует слово «devops», и эта команда действительно работает, нет смысла менять ее на другую команду. Имейте в виду, что слово «devops» подразумевает культуру, процесс, название и структуру команд, сформированных в вашей организации.

Devops как профессия

Толкование профессии «devops-инженер» является неоднозначным. Некоторые из толкований приведены в следующем списке:

- системный администратор, который знает, как писать программный код;
- разработчик ПО, владеющий навыками системного администрирования;
- мифический десятикратный инженер (производительность этого сотрудника в 10 раз выше производительности других сотрудников, хотя это скорее образное выражение), который может за одну зарплату выполнять обязанности системного администратора и разработчика ПО с наивысшим качеством работы.

Описанная концепция devops-инженера не только нереалистична, она еще и плохо масштабируется. В новых организациях разработчики нередко развертывают код и поддерживают сопутствующую инфраструктуру, но если компания является зрелой и имеет большие размеры, целесообразно привлекать специалистов к выполнению отдельных работ.

Вряд ли имеет смысл назначать devops-директора либо любого другого человека, ответственного за внедрение devops-практик в организации. По своей сути devops является культурным движением, поэтому идеи и принципы этого движения должны внедряться на всех уровнях организации. Лишь в этом случае devops будет эффективным.

Несмотря на все, что сказано выше, должность «devops-инженер» действительно существует. В соответствии с данными, опубликованными в отчете 2015 DevOps Salary Report от Puppet (<https://puppet.com/resources/white-paper/2015-devops-salary-report>), у инженеров, в названии должности которых присутствует слово «devops», зарплата выше, чем у обычных системных администраторов. Конечно, эти оценки весьма приблизительны, поскольку в разных организациях devops-инженеры исполняют самые разные роли. Скажите на милость, кто бы отказался от звания devops-инженера за прибавку в 10 тыс. долларов к зарплате?



В отчете 2015 DevOps Salary Report отмечается, что «большинство devops-инженеров работают более 50 часов в неделю, системные инженеры работают от 41 до 50 часов в неделю, а системные администраторы работают менее 40 часов в неделю». Как видите, никто даром деньги не платит. Если вы получаете высокую зарплату, будьте готовы пожертвовать своим личным временем и здоровьем.

Методология devops связана только с веб-стартапами

Основной продукт интернет-компаний — веб-приложения, с которыми имеют дело пользователи при просмотре веб-сайта компании. Эти сайты часто могут быть реализованы без использования информации, которая хранится где-либо между сеансами взаимодействия пользователей с сайтом (то есть сеансы без отслеживания состояния). Обновление сайта можно выполнить либо поэтапно, либо путем переноса «живого» трафика на новую версию сайта. При этом облегчается процесс непрерывного развертывания.

Несложно понять, почему devops-методики имеют такое значение для интернет-компаний. Это движение помогает разрушить барьеры, которые стоят на пути разработки и развертывания программного обеспечения. Если процессы, реализованные в интернет-компаниях, настолько медленны, что требуется несколько недель на устранение мелкой ошибки, скорее всего, эта компания не сможет конкурировать с организациями, в которых используются новые гибкие подходы к разработке и развертыванию приложений.

Преимущества, обеспечиваемые улучшенным сотрудничеством, близостью и инструментами, проявляются не только в веб-стартапах. Просто повторяющиеся действия (или итерации) с групповыми структурами и процессами проще выполнять в рамках небольших стартапов. Крупные компании сопротивляются быстрым изменениям, а правительственные учреждения активно препятствуют изменениям и ограничивают их с помощью специальных законов. Тем не менее изменения возможны даже в таких учреждениях и организациях. В следующих главах книги будут рассмотрены примеры использования методик devops в крупных корпорациях и правительственных учреждениях.

Devops нужно сертифицировать

Поскольку devops-методики в большой степени связаны с культурой, сертификация в этом случае вряд ли применима. Невозможно провести 60-минутный экзамен и по его результатам сертифицировать эффективность общения с другими людьми, оценить степень налаженности командной работы в вашей организации и сделать выводы о порядке обучения в организации. Сертификацию имеет смысл применять по отношению к высоким технологиям, использование которых потребует серьезного уровня знаний о таких вещах, как специфическое программное обеспечение или оборудование. Благодаря подобной сертификации компании могут получить представление о знаниях отдельных сотрудников в данной области.

При использовании devops-методов не требуется технологическая база либо универсальные решения. На сертификационных экзаменах обычно проверяют уровень знаний, задавая вопросы, на которые существуют четко определенные ответы. Как правило, в devops таких ответов не существует. То, что хорошо работает в одной компании, вовсе не обязательно окажется оптимальным для другой компании. Не существует простого способа разработать набор вопросов,

имеющих отношение к devops, которые имели бы однозначные ответы. Сертификация в devops может либо использоваться в качестве средства заработка, либо быть связанной определенным решением поставщика, но называть ее devops-сертификацией некорректно.

Благодаря devops можно сократить персонал

Некоторые полагают, что благодаря devops можно получить разработчика ПО и инженера из службы эксплуатации в одном лице, но не платить ему двойной оклад. Мало того что это мнение ошибочно, оно еще и очень вредно. Многие стартапы предлагают сотрудникам бонусы, такие как трехразовое питание и прачечная на рабочем месте, в надежде стимулировать сотрудников подольше задерживаться на работе, но это приводит к негативным эффектам. Многие инженеры проводят на работе по 60–80 часов в неделю, люди перестают соблюдать баланс между работой и отдыхом, что ведет к переработке и переутомлению.

На самых ранних стадиях проекта стартап может только выиграть, если разработчики настолько хорошо разбираются в эксплуатации, что могут выполнять развертывание ПО, особенно при обращении к провайдерам облачных услуг и другим компаниям, могущим взять на себя выполнение большого объема оперативной работы. Но как только компания переходит в стадию зрелости и сотрудники начинают выполнять двойную работу, это может вылиться в выгорание.



Методология devops не поможет вам сэкономить деньги за счет сокращения вдвое количества инженеров, работающих в компании. Скорее она позволяет организациям улучшить качество и эффективность работы, сократить количество и длительность простоев, уменьшить время разработки и улучшить эффективность отдельных сотрудников и команд в целом.

Не существует единственно верного пути внедрения devops

Практики и специалисты по внедрению devops, работавшие во времена появления этой методологии, особенно хорошо известные в этой области, например Netflix и Etsy, часто назывались «единорогами»¹. Они монополизировали рынок «правильных» способов внедрения devops. Другие компании, стремящиеся ощутить преимущества devops-культуры, иногда стремятся подражать этой практике.

Тот факт, что компания представляет собственную успешную devops-стратегию, вовсе не означает, что те же самые процессы позволят корректно внедрить практики devops

¹ В терминологии devops единорог — это интернет-компания, которая исполняла функции практика, специалиста по внедрению инноваций и реализации devops в первые годы появления этой методологии. Не путайте с определением единорога в сфере финансов, там это стартап стоимостью более 1 млрд долларов.

в другой среде. Принудительный «вброс»¹ процессов и инструментов в среду может привести к дополнительной изоляции и к появлению сопротивления изменениям.



В devops поощряется критическое мышление по отношению к процессам, инструментам и практикам. В самообучающейся организации требуется проведение опросов и итеративный перебор процессов. При этом не принимаются на веру такие утверждения, как «единственно верный путь» либо «все уже было сделано до нас».

Не следует прислушиваться к людям, навязывающим свой способ внедрения devops. Нелишним будет повторить, что существует множество вариантов внедрения devops. Методология devops — это не набор жестко заданных правил, подобно Scrum либо ITIL. Компании, которые наиболее успешно внедрились devops, добились этого путем создания комфортных условий для обучения и итеративного поиска наиболее эффективных инструментов и процессов.

Для внедрения devops нужно X недель/месяцев

Если для внедрения devops нужно согласие руководства, то обычно задается вопрос о том, сколько времени займет подобное преобразование. Дело в том, что многие считают, что состояние внедрения devops можно как-то оценить либо измерить. И как только это состояние будет достигнуто, работу по переходу на devops можно завершить.

В действительности же devops представляет собой непрерывный процесс, это путешествие, а не конечная точка. Некоторые компоненты devops будут иметь фиксированные конечные точки (например, настройка системы управления конфигурированием и управление всеми серверами компании с помощью этой системы), но текущее обслуживание, разработка ПО и использование системы управления конфигурированием — непрерывный процесс.

Поскольку движение devops является в большей степени культурным феноменом, затруднительно предсказать, сколько будут длиться некоторые из этих изменений, сколько потребуется времени людям, чтобы отказаться от привычки работать в одиночестве и получить навыки сотрудничества. Все это довольно сложно предугадать, но пусть это не остановит вас от реализации важнейших культурных преобразований.

Методология devops — это лишь инструменты

Методология devops не требует обязательного применения каких-либо конкретных инструментов. Неверное представление об обязательности использования инструментов способствует формированию убежденности, что devops предназначен

¹ *Принудительный «вброс»* (или карго-культ) описывает практику эмуляции наблюдаемых поведений или внедрения инструментов в среде без полного понимания причин или обстоятельств, которые ведут к успеху.

исключительно для стартапов, поскольку крупные компании зачастую не склонны внедрять новые технологии.

По своей природе devops является культурным движением. Инструменты, используемые в определенной среде, также являются частью культуры. Прежде чем принять решение в пользу изменений, требуемых для внедрения devops-принципов, нужно идентифицировать инструменты, являющиеся частью существующей культуры, получить сведения об опыте использования этих инструментов отдельными сотрудниками, а также идентифицировать сходство и различие между этими инструментами. После получения этой информации вы сможете определиться с необходимыми изменениями.

Применяемые технологии влияют на скорость изменений и организационные структуры организации. Серьезные изменения, внесенные в набор используемых инструментов, могут привести к замедлению скорости развития организации в целом, особенно если эти инструменты представляют ценность для отдельных сотрудников или команды.

Принципы, рассматриваемые в книге, не требуют специального набора инструментов и могут применяться к произвольному набору технологий. Существует довольно много общего между компаниями, которые практикуют devops, и компаниями, использующими контейнеры или провайдеров облачных услуг, но это вовсе не означает, что для внедрения devops нужны эти конкретные технологии. Известны примеры компаний, которые успешно внедрили devops, не используя какие-либо инструменты. В части IV будут рассмотрены эффективные способы оценивания, выбора и внедрения инструментов.

Методология devops эквивалентна автоматизации

Благодаря совершенствованию инструментов, применяемых devops-командами, облегчается систематизация накопленной информации. Внедрение автоматизации позволяет улучшить взаимодействие между командами. Практические специалисты делают упор на инструментах, устраняющих необходимость выполнения скучных повторяющихся задач. Примеры подобных инструментов — автоматизация инфраструктуры и непрерывная интеграция. В этих случаях автоматизация является результатом применения улучшенной технологии.

Если для освобождения персонала от выполнения рутинной работы используется автоматизация, это приведет к росту эффективности труда. В некоторых случаях преимущества автоматизации очевидны. Например, создание автоматизированного сервера позволит системному администратору сэкономить драгоценное время, которое можно потратить на более интересную или интеллектуальную работу. Но если на внедрение автоматизации будет потрачено больше времени, чем сэкономлено в результате этой же автоматизации, то вряд ли игра стоит свеч (рис. 5.1).

В свое время имели место многочисленные дискуссии о роли автоматизации в разных средах и о роли человеческого фактора в выборе типа автоматизации и способах ее внедрения. Эти дискуссии можно рассматривать в качестве примера

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

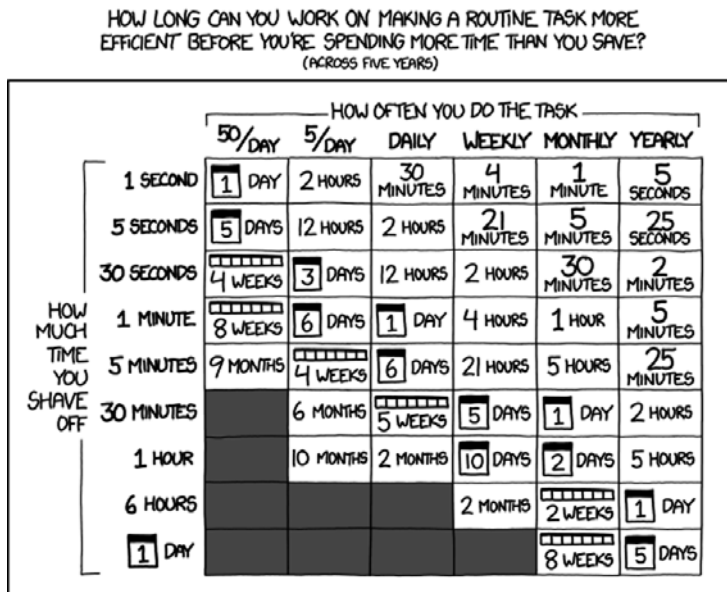


Рис. 5.1. Карикатура на потраченное и сэкономленное время согласно представлению карикатуриста из XKCD (<https://xkcd.com/1205/>); изначально была опубликована со следующим текстом: «Не забывайте о том, что на поиск диаграммы, которая позволит узнать, сколько вы сэкономите, тратится время. Также вы тратите время на чтение напоминания о потраченном времени. И тратите время на выяснение разных фактов. Помните о том, что вы тратите секунды вашей жизни на выполнение разных действий, и делаете это сейчас»

межотраслевой близости. Если даже автоматизация применяется в других областях промышленности, вы сможете перенять чужой опыт и творчески его переосмыслить.

РАННИЕ ПРИМЕРЫ АВТОМАТИЗАЦИИ В АВИАЦИИ

В 1977 году при рассмотрении состояния дел в американской авиапромышленности Комитет палаты представителей по науке и технике пришел к выводам о том, что автоматизация кабины пилотов негативно влияет на авиационную безопасность. В результате проведенных исследований выяснилось, что автоматизация кабины привела практически к полной атрофии критического мышления пилотов. Они утратили способность к ориентированию в пространстве без использования карты, отображаемой на дисплее, не могут выбрать следующее действие либо распознать сбой прибора. Как видите, применение автоматизации способно изменить поведение и способ мышления.

В июле 2013 года рейс 214 компании Asiana Airlines врезался в дамбу в международном аэропорту Сан-Франциско. В результате этого авиационного происшествия получили смертельные травмы 3 человека. В ходе расследования этого инцидента, проводимого Национальным советом по безопасности на транспорте (National Transportation Safety Board, NTSB), был выявлен ряд причин,

способствующих катастрофе. Среди этих причин — недостаточный контроль скорости воздушного судна из-за чрезмерной зависимости от автоматизированных систем, показания которых не всегда корректно интерпретировались пилотами.

В своих попытках устранить ненадежный человеческий фактор разработчики автоматизированных систем управления невольно создали возможности для появления новых типов ошибок, которые могут быть более серьезными, чем те, которых они стремятся избежать.

Джеймс Русон, Managing the Risks of Organizational Accidents

По мере усложнения систем и распространения общих услуг автоматизация приобретает критическое значение. Но если при внедрении автоматизации не используется общий контент и не учитываются интересы персонала, появляется дополнительный риск. Несомненно, благодаря автоматизации ускоряется выполнение работы, но чтобы это выполнение стало более эффективным, нужно увеличить степень прозрачности, сотрудничества и понимания.

Методология devops вышла из моды

Подобно тому как методология devops не привязана жестко к используемой технологии, инструменту или процессу, нельзя сказать, что она устарела или нуждается в замене. Движение, направленное на повышение эффективности деятельности организации и создание хорошего настроения для каждого сотрудника, может стать привычным и незаметным, но вряд ли когда-либо устареет.

Одно из основных различий между devops и такими методологиями, как ITIL и гибкая разработка (Agile), заключается в том, что последние имеют строгие определения, им присущ постепенно добавляемый контент. С другой стороны, devops — это движение, основанное на историях и идеях, которыми делятся отдельные люди, команды и организации. Это движение выливается в повторяющиеся беседы, эволюцию процессов и идей, которые приводят к росту и изменениям организаций.

В сообществе пользователей devops имела место дискуссия о том, потеряла методология devops вектор развития или нет. Критики этого движения утверждают, что оно слишком сильно зависит от отрицательных факторов, что оно уже не то, чем было ранее. Они также утверждают, что devops не является уникальным, что это всего лишь ребрендинг идей, которые существовали ранее, и от devops откажутся, как только появятся новое имя или тренд.

Многие из идей, лежащих в основе движения devops, действительно витали в воздухе в течение некоторого времени и под разными названиями, но дух devops является чем-то большим, чем простая сумма составляющих частей. Конечно же, люди выступали против функциональной изоляции еще до появления devops, высказываясь в пользу самообучающихся организаций, в защиту человеческих ценностей и автоматизации.

Движение devops объединило эти идеи и смогло достичь значимого успеха (<http://bit.ly/2015-state-of-devops>). Обобщение и использование эффективных методик devops приведет к росту и эволюции инструментов, технологий и процессов в вашей организации.

Антишаблоны devops

В этом разделе будут определены некоторые дополнительные термины, которые помогут лучше осознать суть devops. Но в отличие от терминов, определенных в предыдущей главе, эти понятия, как правило, трактуются как антишаблоны, продвигающие идеи, которые идут вразрез с концепциями devops.

Культура, основанная на обвинениях

Культура, основанная на обвинениях (упречная культура), проявляет склонность к обвинениям и наказаниям людей, совершивших ошибку (как на индивидуальном, так и на организационном уровне). В рамках этой культуры анализ первопричин не включается в постмортем или ретроспективу, используемые для идентификации причины сбоя или инцидента. Если анализу подвергаются действия человека, служащего «первопричиной» инцидента, он будет обвинен, ему будет объявлен выговор или его даже уволят из-за неприглядной роли в произошедшем инциденте. Этот вид культуры часто имеет место там, где нужно давать ответы внешним аудиторам либо нужно повысить производительность в соответствии с запланированными показателями.

В сильно изолированных или сегментированных средах, в которых уделяют недостаточно внимания прозрачности, пышным цветом расцветает культура, основанная на обвинениях. Если менеджмент стремится обвинить сотрудника или группу людей в каждом произошедшем инциденте и избавиться от «гнилого яблока», отдельные сотрудники будут пытаться переложить вину на других сотрудников или посторонних лиц. Хотя стремление к самосохранению вполне объяснимо в подобной среде, оно несовместимо с культурой открытости и сотрудничества. Более чем вероятно, что в подобной ситуации люди начнут скрывать информацию о произошедших событиях, особенно возникших вследствие их собственных действий, чтобы попытаться избежать обвинений.

Если не рассматривать реагирование на инциденты, культура, основанная на обвинениях, в попытках повышения производительности сотрудников будет способствовать формированию атмосферы враждебности между коллегами, в которой каждый сотрудник будет пытаться избежать ответственности. Если люди только и думают о том, чтобы не подвергнуться наказанию, они в меньшей степени нацелены на обучение и сотрудничество.

Барьеры

С помощью модели барьера на уровне отдела или организации описывается менталитет команд, которые не делятся своими знаниями с другими командами внутри одной и той же организации. Вместо того чтобы иметь общие цели и обязанности, группы, находящиеся за барьерами, исполняют разные и сегрегированные роли. В сочетании с культурой, основанной на обвинениях, это может привести к накоплению секретной информации о порядке выполнения той или иной работы («Если я единственный, кто знает, как выполнить X, они не смогут меня уволить»). Подобная атмосфера может привести к затруднениям в работе или к затягиванию работы, выполняемой несколькими командами, а также к ухудшению морального климата, когда команды или люди, находящиеся за барьерами, начинают видеть друг в друге противников.

Зачастую в барьерной среде можно найти разные команды, применяющие совершенно разные инструменты или процессы для выполнения идентичных задач. В этой среде сотрудники используют несколько уровней управленческой цепочки команд для получения ресурсов или информации от членов другой группы и достаточно часто перекладывают ответственность на другую команду.

Для устранения проблем и решения вопросов, связанных с организационными барьерами, потребуются время, усилия и культурные изменения. Наличие отдельных барьеров для разработчиков программ, системных администраторов и инженеров из группы эксплуатации, а также предпринимаемые попытки исправления ошибок, допущенных при разработке программного обеспечения, привели к возникновению движения devops. Основная причина появления барьеров заключается не в тотальном разделении обязанностей, а в отсутствии общения и сотрудничества между командами. Именно эти проблемы способно устранить движение devops.

Анализ первопричин

Анализ первопричин (Root Cause Analysis; RCA) — это метод идентификации причин возникших событий либо опасных ситуаций и выработки комплекса мер, направленных на предотвращение рецидивов. Это повторяющийся процесс, который продолжается до тех пор, пока не будут идентифицированы все организационные факторы либо пока не будут исчерпаны все данные. Организационный фактор — это фактор, который осуществляет контроль над системой на любом этапе жизненного цикла: проектирование, разработка, тестирование, техническое обслуживание, эксплуатация и списание.

Один из методов идентификации первопричин известен под названием «5 почему». При использовании этого способа вопросы «почему» задаются до тех пор, пока не будут идентифицированы основные причины. При этом требуется, чтобы лица, отвечающие на вопрос «почему», располагали объемом информации, достаточным для ответа на вопрос должным образом. Второй, более систематический подход заключается в создании *диаграммы Ишикавы*. Разработанная в 1968 году

Каору Ишикавой, эта причинно-следственная диаграмма помогает командам визуализировать и группировать первопричины в основные категории, чтобы идентифицировать причины изменчивости, выявлять взаимосвязи между источниками и обеспечивать понимание процесса.

Зачастую метод RCA применяется для идентификации единственной первопричины. Инструменты, которые поддерживают управление событиями, часто допускают единственное назначение ответственности. Это ограничивает степень полезности метода RCA, поскольку внимание акцентируется на непосредственных причинах, а дополнительные элементы упускаются из виду. В методе анализа первопричин имеет место неявное предположение, что сбой или успех системы происходит в соответствии с линейным законом, что некорректно в случае достаточно сложных систем.

Человеческие ошибки

При использовании метода анализа первопричин в качестве основной причины сбоя указывается человеческая ошибка. Зачастую за этим утверждением следует вывод, что другой человек не совершил бы подобной ошибки. Это типично для культур, основанных на обвинениях, когда кому-то нужно объявить выговор за роль в произошедшем инциденте. Опять же, это слишком упрощенное представление, используемое в качестве точки останова при проведении расследования. Предполагается, что человеческие ошибки совершаются из-за простой небрежности, усталости или некомпетентности. При этом из виду упускаются мириады факторов, которые способствовали принятию решения либо выполнению действия.

В культуре, основанной на обвинениях, обсуждение завершается после того, как будет найден человек, совершивший ошибку. При этом упор делается на самой ошибке и ее последствиях. В безупречной культуре или в самообучающейся организации человеческая ошибка рассматривается в качестве отправной точки, а не завершающего этапа. Появление ошибки приводит к дискуссии о контексте, связанном с принятием ошибочного решения, и о его смысле в данной ситуации.

Выводы

Знакомство с терминами, рассмотренными в главе, обеспечит вам углубленное понимание остального материала книги. Вооруженные основной терминологией и концепциями, рассмотренными в предыдущей главе, а также сведениями из главы 3, вы получите более четкое представление о современном состоянии devops. На основе этого представления можно определить и более подробно рассмотреть четыре столпа эффективных devops-методов, что и будет сделано в следующей главе.