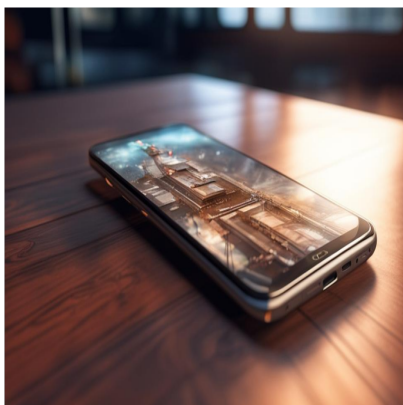




Что такое управление релизами: углубленный взгляд на процесс, внедрение и достижения

## **Описание**

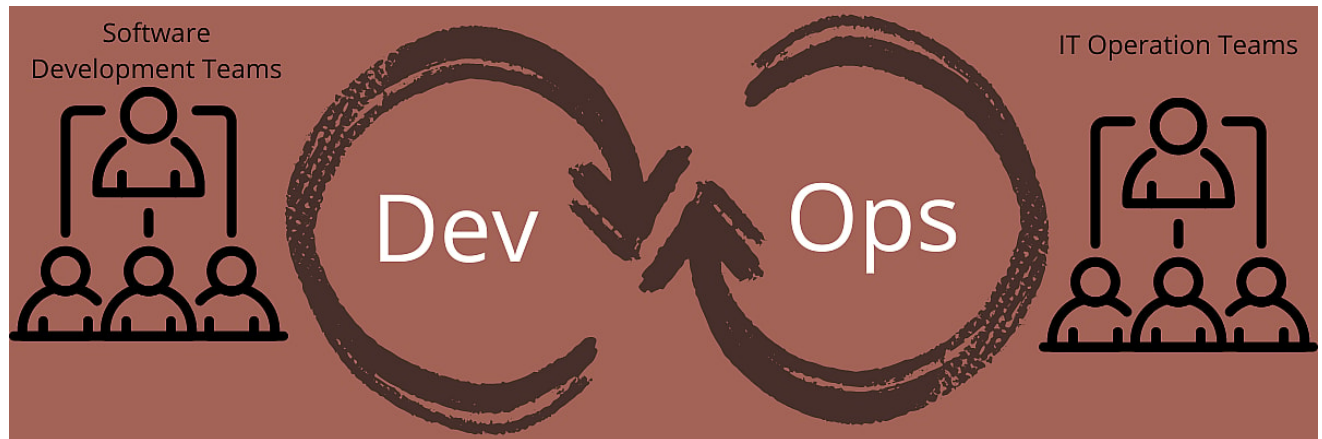
Управление выпуском – важнейший процесс, требующий эффективного планирования и исполнения, чтобы обеспечить выполнение требований и удовлетворить конечного пользователя. Однако организациям приходится работать над множеством релизов, чтобы поддерживать свои программные продукты в актуальном состоянии с новыми возможностями и функциями. При отсутствии должного подхода это может привести к путанице и перегрузке. Поэтому вам необходимо оптимизировать процесс управления релизами, чтобы работать над каждым релизом с полной ясностью и эффективностью. Давайте разберемся, как это сделать.

## **Что вы подразумеваете под управлением выпуском?**



В области разработки программного обеспечения и ИТ управление выпуском относится к процессу, в ходе которого программная система создается, развертывается и передается пользователям. Этот процесс включает в себя все этапы, от планирования и разработки до тестирования и развертывания. (Кстати, если вас интересует только процесс развертывания, мы также подготовили замечательный список лучших платформ автоматизации развертывания, на которые вам стоит обратить внимание). Управление релизами – это система, которой следуют производственные команды для управления всем жизненным циклом поставки программного обеспечения. Она также направлена на обеспечение оптимального опыта для конечных пользователей и согласование приоритетов бизнеса. Кроме того, управление релизами может помочь вам оптимизировать и визуализировать процессы разработки и развертывания программного обеспечения, последовательно удовлетворять потребности в поставках, управлять рисками при поставке программного обеспечения, управлять ИТ-ресурсами и обеспечивать рентабельность при предоставлении ценности. Разработчики используют процесс управления релизами при создании нового программного продукта или создают новые версии, внося некоторые изменения в существующую версию ПО. Наличие четкой цели позволит им двигаться по одному пути, устраняя путаницу и повышая эффективность.

## Управление выпусками и DevOps



DevOps делает акцент на расширении коммуникаций и сотрудничества между командами разработчиков программного обеспечения и операционными группами ИТ. Методология DevOps направлена на сокращение изолированности в работе над проектами и позволяет командам не забывать о важных действиях. Это также способствует сокращению циклов обратной связи, что позволяет вашей команде быстрее выпускать продукт и сокращать количество сложностей. Поэтому для эффективного выпуска кода в производство релиз-менеджеры полагаются на три основные вещи – автоматизацию, непрерывную интеграцию (CI) и DevOps. Они могут автоматизировать тесты и постоянно собирать, интегрировать и обновлять свой код. Используя менталитет DevOps, они также могут улучшить координацию между командами разработки и эксплуатации. В результате ошибки обнаруживаются на ранней стадии, что позволяет легко их исправить и ускорить процесс разработки и выпуска.

## Процесс управления выпуском

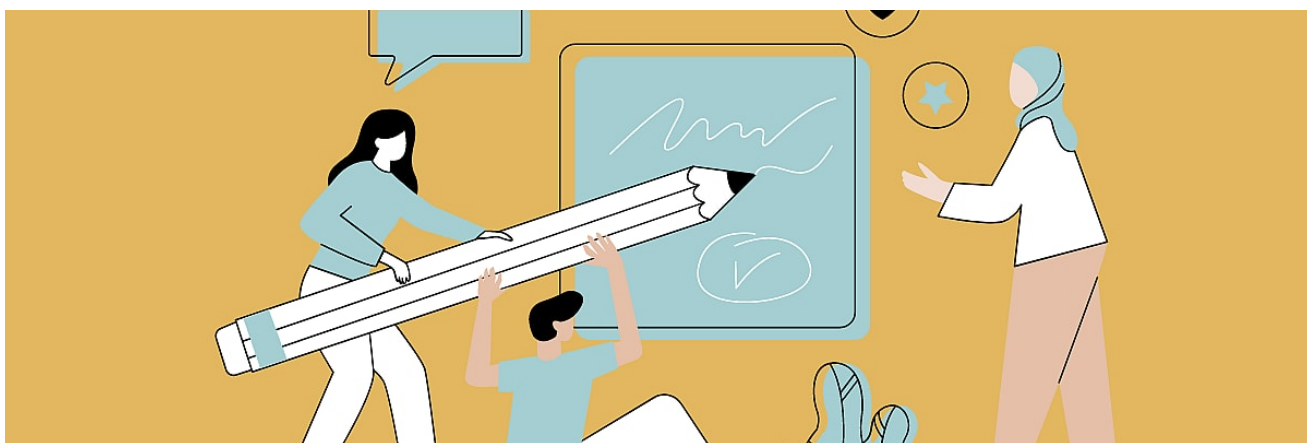
В жизненном цикле управления релизами есть разные этапы. Этот процесс может отличаться от команды к команде и от организации к организации, поскольку требования к проекту могут быть разными. Однако есть несколько общих шагов, которым должны следовать организации и команды любого размера, чтобы свести концы с концами и предоставить пользователям качественное решение. Итак, вот как выглядит типичный процесс управления релизами.

### Понимание требований

Независимо от того, планируете ли вы создать новый продукт или добавить несколько новых функций и возможностей к существующему продукту, понимание

требований имеет решающее значение. Поэтому прислушайтесь к своему клиенту и к тому, что он хочет добавить в свой проект, например, в мобильное приложение. Например, они могут попросить вас добавить определенный раздел в их мобильное приложение. Это потребует от вас проведения встречи с ними, чтобы понять их требования и ожидания, а также то, почему они хотят получить то, что хотят. Аналогично, если у вас есть сайт и вы хотите добавить раздел блога, посетители смогут читать ваши статьи и узнавать больше о ваших предложениях. Какой бы ни была цель, внимательно изучите ее. Если есть какие-то сомнения, обсудите их с командой или заказчиком, а затем приступайте к разработке подходящего плана выпуска.

## Планирование



После полного понимания требований к выпуску следует планирование. Создание и выпуск того, что вы собираетесь сделать, требует тщательного планирования и стратегий, основанных на требованиях. Ваше планирование должно быть осуществимым и практичным с точки зрения технологий, сроков, рабочей силы и ресурсов. Например, если вы хотите выпустить новую версию своего приложения, вы должны убедиться, что оно эффективно работает на всех устройствах – мобильных, ноутбуках, планшетах и т. д. Для планирования тесно общайтесь с клиентом. Вы можете обсудить график (его можно составить с помощью программы для составления графиков) проекта и то, когда они могут ожидать выхода продукта. Нельзя обещать невыполнимые сроки. Поэтому при утверждении сроков не забывайте о таких ресурсах, как бюджет, время и люди. Кроме того, спланируйте технологии, которые вы будете применять для выпуска. Определите, насколько эффективно они обосновывают требования, укладываются ли в ваш бюджет и соответствуют ли навыкам ваших сотрудников. Выберите подходящие

технологии, которые помогут вам создать эффективный продукт, выпустить его в срок и легко адаптировать для ваших сотрудников. Планирование также требует эффективного распределения и использования имеющихся ресурсов, чтобы избежать потерь и обеспечить эффективное строительство вашего проекта. Чтобы создать надежный план, проведите встречу с командой разработчиков и операторов, чтобы обсудить требования, проблемы, способы борьбы с ними и пути эффективного достижения цели.

## Разработка

После того как вы окончательно определились с планом, наступает следующий этап – проектирование и разработка продукта. Это время для реализации ваших планов и стратегий на основе установленных требований. Этот процесс требует от ваших разработчиков написания кода, который может быть воплощен в тех возможностях или функциях, которые вы собираетесь добавить в свое программное обеспечение. Этот этап может повторяться много раз в течение всего цикла выпуска, как, например, в DevOps с непрерывной разработкой. Это связано с тем, что после того, как разработчик написал код, в нем может быть несколько проблем, ошибок и багов, которые необходимо протестировать. Прежде чем код будет утвержден, он пройдет множество раундов тестирования. Разработчикам будет предоставлен весь документ с проблемами, которые они должны решить и оптимизировать код, чтобы он работал так, как должен, и был утвержден.

## Тестирование



Как уже говорилось выше, код нуждается в тестировании, чтобы убедиться в отсутствии ошибок и багов, которые могут повлиять на удобство использования, производительность или безопасность программного обеспечения.

Тестирование программного обеспечения или приложений может быть функциональным или нефункциональным, например, интеграционное тестирование, тестирование удобства использования, нагрузочное тестирование, тестирование производительности, тестирование принятия пользователем и другие. После обнаружения проблем код будет отправлен обратно команде разработчиков для их устранения и повторной отправки улучшенного кода. Теперь программное обеспечение будет передано пользователям, чтобы они определили, соответствует ли оно требованиям и ведет ли себя так, как им нужно, с помощью приемочного тестирования. Если пользователь одобряет его, необходимо выполнить следующие шаги. В противном случае пользователь получает обратную связь, чтобы еще раз улучшить код, протестировать его, а затем развернуть.

## **Развертывание релиза**

После того как команда разработчиков убедится, что программное обеспечение создано в соответствии с требованиями и что в нем нет никаких проблем. Они готовятся к выпуску или развертыванию на рынке или передают его клиенту. Команда QA также проведет финальные тесты, чтобы убедиться, что продукт соответствует бизнес-требованиям и минимальным стандартам, определенным в плане выпуска продукта. Затем продукт будет рассмотрен владельцем продукта или менеджером для утверждения отправки. В это время составляется необходимая документация, чтобы помочь другим разработчикам разобраться в программном обеспечении и понять, как его использовать. Команды также занимаются окончательным оформлением документов, чтобы передать продукт клиенту. Организации также рассматривают возможность обучения своих пользователей или сотрудников работе с новым продуктом, чтобы они могли работать с ним без проблем.

## **Обслуживание релиза**





Независимо от того, создавали ли вы релиз для своей команды или для своих клиентов, ваши обязанности не заканчиваются на этапе развертывания. Независимо от того, насколько эффективно и превосходно работает ваше программное обеспечение в данный момент, оно нуждается в периодическом обслуживании для поддержания оптимальной производительности. Кроме того, никогда нельзя предугадать, когда может возникнуть проблема безопасности. И когда это произойдет, это может серьезно повлиять на ваш бизнес и репутацию. Существует множество факторов, которые могут повлиять на ваше программное обеспечение, привести к замедлению работы, сбоям, уязвимостям в системе безопасности, проблемам с удобством использования и так далее. Поэтому вы всегда должны следить за своим программным обеспечением даже после того, как оно будет выпущено для пользователей. Вы должны уделять время проверке его производительности, безопасности, удобства использования и стабильности, чтобы найти проблемы и устранить их до того, как они могут повлиять на пользователей. Вот как выглядит процесс управления релизами – от планирования до развертывания и сопровождения, а также все, что между ними.

## Управление выпуском и управление изменениями

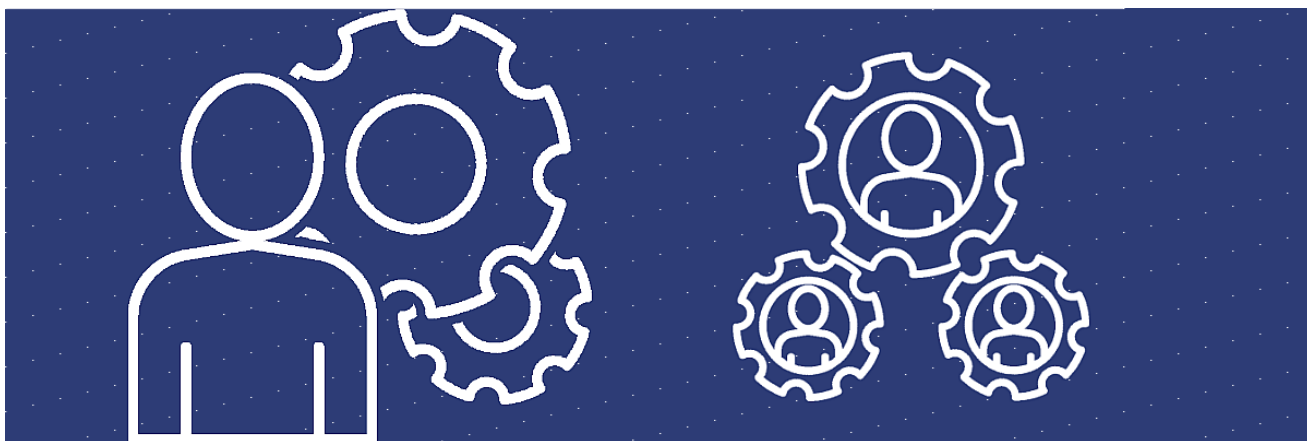
Иногда управление релизами может выглядеть как управление изменениями, поскольку вы вносите изменения в свое программное обеспечение и оптимизируете общий процесс с помощью некоторых стратегий, позволяющих управлять всем. Но управление релизами и управление изменениями – это не одно и то же. Управление изменениями – это шаг вперед по сравнению с управлением выпуском. Оно управляет всеми действиями до и после выпуска, включая окончательный анализ того, как изменения были реализованы. Однако управление выпуском – это лишь

отдельная часть процесса управления изменениями.

## Какие роли важны для управления выпуском?

Управление выпуском требует участия в этом процессе нескольких человек. К числу основных из них относятся:

### Владелец продукта



Владелец продукта отвечает за определение требований к релизу и стандартов приемки, которые должны быть соблюдены, чтобы релиз был утвержден. Владелец продукта возглавляет начальные этапы жизненного цикла управления релизом, на которых обсуждаются требования и осуществляется планирование.

### Команда DevOps

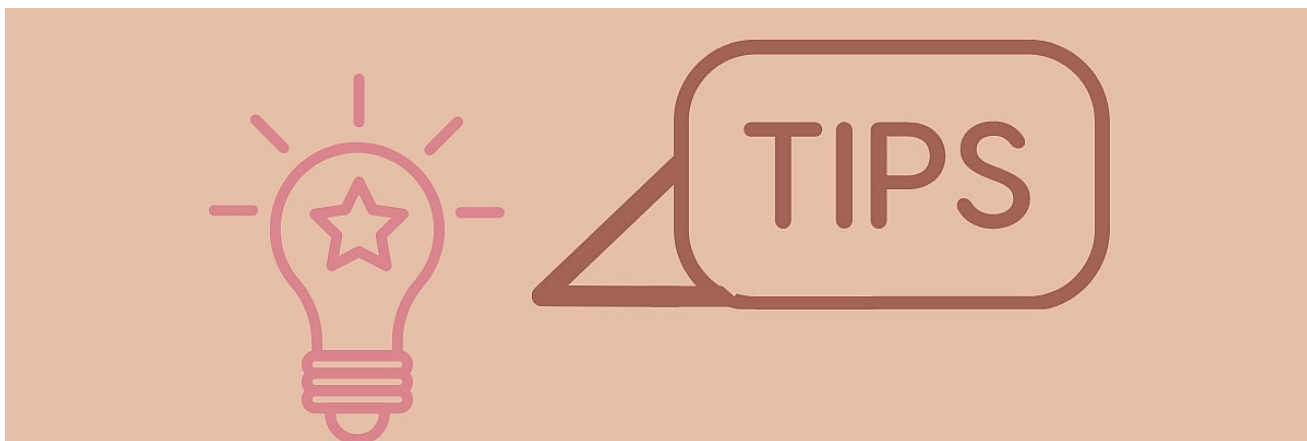
Управление выпуском использует подход DevOps для разработки, развертывания и сопровождения программного проекта. Это позволяет объединить усилия команды разработчиков и операторов для тесного сотрудничества, обсуждения проекта, понимания рисков и способов их устранения, чтобы создать качественное программное обеспечение для выпуска. Команда DevOps поддерживает стабильную среду постановки, чтобы разработчики могли эффективно работать. В идеале эта среда должна быть ближе к производственной среде, чтобы после проведения всех тестов можно было быстро перенести программное обеспечение в производство. При этом они также заботятся о том, чтобы минимизировать время простоя. Таким образом, вы можете спокойно выпускать программное обеспечение, не испытывая его на пользователях.



## Менеджер по качеству

Менеджер по качеству отвечает за определение того, выполнены ли критерии приемки и создано ли программное обеспечение в соответствии с требованиями, установленными владельцем продукта. Эта роль необходима для получения одобрения продукта владельцем продукта. Менеджер по продукту также следит за тем, чтобы тесты проводились без ложных отрицательных или положительных результатов, и определяет, все ли проблемы решены.

## Лучшие практики управления релизами и советы по улучшению процесса



Если вы хотите улучшить процесс управления релизами, воспользуйтесь этими советами и лучшими практиками.

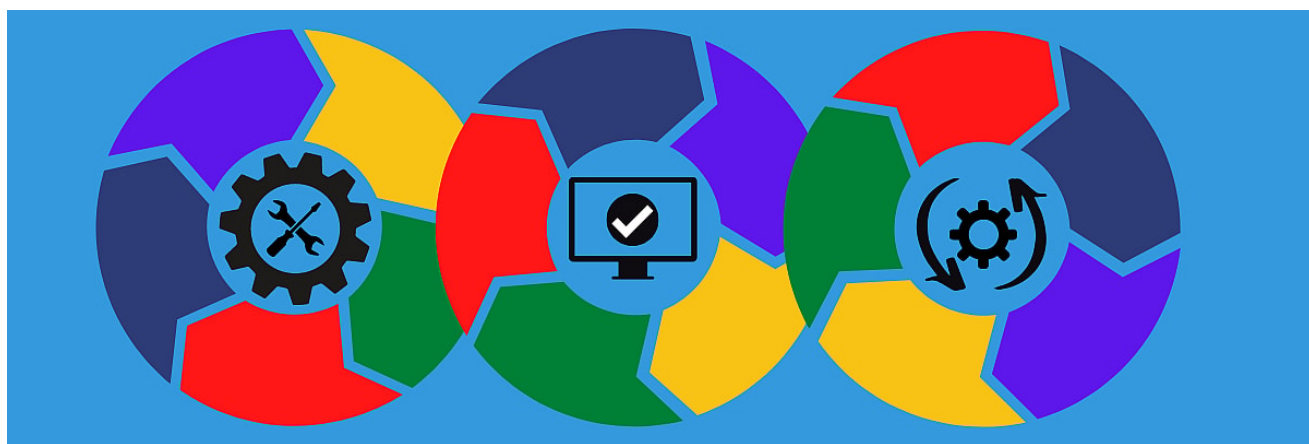
- **Устраните проблему первого кода:** Первый код пишется первым разработчиком и используется следующей командой. Поэтому, когда этот человек покидает компанию, другим становится сложнее понять или адаптировать код, который другой человек написал с особым умыслом. Таким образом, все, что вы делаете, должно быть доступно всей команде, и все должны работать в сотрудничестве, чтобы избежать этой проблемы. Именно поэтому DevOps считается одним из лучших подходов к разработке программного обеспечения.
- **Автоматизация тестирования программного обеспечения:** Используйте инструменты тестирования, чтобы автоматизировать процесс поиска ошибок. Это также повышает точность по сравнению с тем, чего вы можете достичь

при ручном кодировании. В этом вам помогут такие инструменты, как Selenium, Watir и т. д.

- **Инфраструктура как код:** Используйте IaaS везде, где можно ускорить процесс и сделать его более масштабируемым и менее затратным.
- **Централизованное управление релизами:** Управляйте каждым релизом централизованно, а не передавайте его одному человеку или системе. Это повышает безопасность и улучшает взаимодействие, не полагаясь во всем на одного человека или систему.
- **Интеграция ITIL и DevOps:** Использование DevOps и ITIL в управлении релизами дает множество преимуществ. Это улучшает сотрудничество, правильное использование ресурсов и эффективность.
- **Постоянно добавляйте новые обновления:** Вместо того чтобы изменять обновления, вы можете регулярно создавать новые. Это связано с тем, что многократное изменение конфигурации может привести к ошибкам и недочетам. Однако если вы будете создавать новые обновления, ваши релизы будут более безопасными и надежными, обеспечивая лучший пользовательский опыт.
- **Обновите среду постановки:** Обновляйте среду постановки и приближайте ее к производственной среде, чтобы после устранения ошибок можно было быстро перейти к работе.
- **Четко определите требования и критерии приемки:** Работа “под водой” может привести к ошибкам, путанице и недовольству клиентов. Поэтому важно выслушать пользователей или клиентов об их требованиях, чтобы создать то, что им действительно нужно. Кроме того, убедитесь, что определенные критерии приемки также соблюдены, чтобы повысить шансы на одобрение и уменьшить количество переделок.
- **Минимизируйте воздействие на пользователей:** при внедрении анти-релиза убедитесь, что вы оказываете минимальное воздействие на конечных пользователей. Для этого планируйте сокращение времени простоя и вовремя предупреждайте пользователей, чтобы они не испытывали сильного воздействия.

- **Автоматизация:** Автоматизация – это ключ к выполнению большего объема работы за меньшее время и с максимальной эффективностью. Поэтому автоматизируйте процесс везде, где это возможно, чтобы ускорить работу, повысить производительность и сэкономить время. Вы можете использовать инструменты автоматизации, доступные на разных этапах цикла управления релизами.

## Популярные инструменты управления выпусками



Как уже говорилось выше, автоматизация помогает сэкономить время и повысить эффективность. Вот некоторые из популярных инструментов, облегчающих управление релизами.

- **GitLab:** Это платформа с открытым исходным кодом, которая поможет вам в управлении всеми процессами выпуска релизов, от планирования до производства, улучшая при этом время выпуска.
- **Ansible:** Ansible – это популярная платформа автоматизации, помогающая создавать и развертывать системы. В ней есть инструменты для автоматизации в масштабах предприятия для предоставления ресурсов, ИТ-среды и т. д.
- **Liquibase:** Liquibase – это инструмент автоматизации изменения схемы базы данных, который поможет вам быстро и безопасно выпустить программное обеспечение, легко включив изменения базы данных в текущую автоматизацию CI/CD.
- **AWS CodePipeline:** AWS CodePipeline – это платформа непрерывной доставки (CD) для автоматизации конвейеров выпуска. Она позволяет быстрее и надежнее обновлять инфраструктуру и приложения.

- Конвейеры Azure: Azure Pipelines помогает автоматизировать сборки и релизы. С его помощью можно собирать, тестировать и выпускать приложения, написанные на Node.js, Java, Python, PHP, C/C++, Ruby и .NET, а также приложения для iOS и Android.
- Digital.ai Release: Digital.ai Release – это эффективный инструмент управления релизами, который поможет вам отслеживать и контролировать релизы, оптимизировать процессы, а также повысить уровень безопасности и соответствия стандартам в ваших конвейерах релизов.
- Chef: Chef – это полный набор инструментов автоматизации, которые помогут вам быстро и эффективно создавать релизы.
- Spinnaker: Spinnaker – это платформа непрерывной доставки (CD) с открытым исходным кодом, которая включает в себя множество функций управления кластерами и развертыванием.
- Octopus Deploy: Octopus Deploy – это инструмент автоматизированного управления релизами, который может интегрироваться с вашим CI-сервером и предоставляет возможности автоматизации релизов и операций.
- Jenkins: Jenkins – это популярный инструмент автоматизации с открытым исходным кодом, который можно использовать для быстрой и оперативной сборки, тестирования и выпуска программного обеспечения.

## **Заключение**

Планируйте и осуществляйте управление релизами в своей организации с помощью вышеупомянутых стратегий и инструментов и сами увидите преимущества. Это поможет вам улучшить цикл выпуска релизов, повысить эффективность и удовлетворенность пользователей.

## **Дата Создания**

17.04.2024