**1 слайд**

DevOps = Development Operations = деятельность в целях развития

**2 слайд**

-

**3 слайд**

### Планирование и отслеживание

Идентифицируйте и отслеживайте работу. Когда работу можно наглядно отследить, заинтересованные лица получают четкое представление о возможностях команды разработки и могут более эффективно планировать задания и определять приоритеты, избегая авральных ситуаций.

### Разработка

Пишите код, используя современные системы управления версиями, такие как Git, для безопасной непрерывной интеграции с главной ветвью. Завершив работу над компонентом, разработчик отправляет запрос на вытягивание, и после утверждения происходит слияние изменений с главной ветвью. Старая ветвь при этом удаляется.

### Сборка и тестирование

### Записывание кода после изменения в Git или другую систему управления версиями запускает процесс автоматической сборки. Код проходит тестирование и проверку — это гарантирует, что ошибки устраняются на раннем этапе разработки, пока разработчик хорошо помнит о них и их исправление менее затратно. Этот процесс автоматической сборки и тестирования называется непрерывной интеграцией (CI). Артефакт, готовый к развертыванию в рабочей среде, — это результат успешной сборки и интеграции, которые создают условия для непрерывной поставки (CD), т. е. возможности развертывания в рабочей среде в любой момент.

### Развертывание

### После тестирования и проверки каждое изменение развертывается в рабочей среде. Благодаря методике непрерывной поставки финальное развертывание в рабочей среде представляет собой бизнес-решение, которым можно управлять вручную.

### Непрерывное развертывание позволяет автоматизировать весь процесс — от фиксации кода до публикации в рабочей среде. При автоматическом развертывании кода ваши клиенты получат доступ к новым компонентам, как только они будут готовы.

### Мониторинг и эксплуатация

В рабочей среде мониторинг обеспечивает поступление информации о производительности и шаблонах использования приложения. Оперативное получение подробных данных диагностики позволяет своевременно принимать меры и достичь высокого уровня доступности. Возможность устранять неполадки, возникающие у пользователей, и собирать данные помогает принимать взвешенные бизнес-решения, касающиеся дальнейшей разработки. А благодаря автоматически применяемым политикам соответствия, в которых используется инфраструктура как код и политика как код, вы можете быть уверены, что в ваших приложениях, развернутых в рабочей среде, будут применяться конфигурации требуемого состояния, соответствующие рекомендациям по обеспечению безопасности.

**4 слайд**

DevOps должен был стать связующим звеном между командой разработки и командой эксплуатации.

**5 слайд**

Разработчики считали, что если код запустился у них локально, то нет проблем – можно запускать в продакшн. Если все же проблемы возникали, то со стороны команды эксплуатации звучало: «Да это проблемы с кодом, пусть разработчики разбираются!».

**6 слайд**

Условно, в DevOps культуре можно выделить несколько ролей, которые очень хорошо соотносятся с профессиями:

Build Engineer — человек, отвечающий за сборку кода. Подтягивание зависимостей, разбор конфликтов в коде — это все про него.

Release Engineer — отвечает за доставку кода от разработки в продакшн. Какая ветка пойдет в тестирование, какой билд попадет на продакшн, релиз-инженер занимается именно этим.

Automation Engineer — инженер по автоматизации. Автоматизирует все, что движется. А что не движется, двигает и тоже автоматизирует. Автоматическая сборка при пуше в гит, прогон тестов, деплой на staging, деплой в продакшн — это все его задачи. Ключевая роль в DevOps подходе.

Security Engineer - будет отвечать за прогон security-тестов и изучение уязвимостей в используемых компонентах.

В реальном мире все (или почти все) эти роли по отдельности обычно совмещает какой-нибудь другой человек. К примеру, роль билд инженера можно отдать в руки разработчика. Да и автоматизация настройки серверов обычно отдается системным администраторам. А DevOps инженеру остается проработать и автоматизировать процесс сборки и доставки кода от разработчика в продакшн.

**7 слайд**

AppCenter - это комплексное решение для организации цикла разработки мобильных приложений на Objective-C, Swift, Java, Xamarin и React Native. Она поможет подключить репозиторий, собрать проект в облаке и протестировать сразу на тысячах устройств. Затем, например, разослать приложения тестировщикам бета-версий и в магазины приложений, провести мониторинг работы в режиме реального времени, отследить отчеты о сбоях и аналитику.