

УДК 004.416

***ВНЕДРЕНИЕ ПРАКТИК DEVOPS В ПРОЦЕССЫ
РАЗРАБОТКИ ПО В РАМКАХ IIOT***

Поклонский А.Ю.

аспирант,

Национальный исследовательский ядерный институт «МИФИ»

Москва, Россия

Титов А.Д.

магистр,

Национальный исследовательский ядерный институт «МИФИ»,

Москва, Россия

Аннотация:

Для оптимизации процессов разработки программного обеспечения ряд практик, которые получили название непрерывного развертывания. В данной работе проводится изучение развития процессов непрерывной интеграции и доставки в разработке IoT. В исследовании производится оценка пригодности существующих решений в сфере непрерывного развертывания для решения задач разработки аппаратно-программных комплексов в рамках интернета вещей систем и, в частности, промышленного интернета вещей.

Ключевые слова: промышленный интернет вещей, непрерывная интеграция, непрерывная доставка, функциональное тестирование, гибкие методологии разработки.

***IMPLEMENTATION OF DEVOPS PRACTICES IN IIOT
DEVELOPMENT PROCESSES***

Poklonskiy A. Y.

graduate student,

National Research Nuclear Institute MEPHI,

Moscow, Russia

Titov A. D.

Master's student

National Research Nuclear Institute MEPHI,

Moscow, Russia

Annotation:

To optimize software development processes, a number of practices are used, which are called continuous deployment. In this paper, we study the development of continuous integration and delivery processes in the development of IoT. The study assesses the suitability of existing solutions in the field of continuous deployment for solving the problems of developing hardware and software systems within the Internet of things systems and in particular the industrial Internet of things.

Keywords: Industrial Internet of Things, Continuous Integration, Continuous Delivery, functional testing, Agile software development.

Практики DevOps на сегодняшний день стали настолько распространены в IT компаниях, что использование соответствующих технологических стеков уже является обыденностью для разработчиков. Использование механизмов непрерывного развертывания

зарекомендовало себя как метод гарантированной оптимизации процесса разработки. DevOps — это набор методик, с помощью которых можно автоматизировать процессы между командами разработчиков и IT-специалистов, чтобы они могли быстрее и надежнее собирать, тестировать и выпускать релизы программного обеспечения. Среди преимуществ — рост уровня доверия, повышение скорости выпуска релизов программного обеспечения, быстрое устранение критических неполадок и готовность лучше справляться с внеплановой работой. Движение DevOps начало формироваться в 2007–2008 годах, когда сообщества специалистов по IT-операциям и разработчиков программного обеспечения, наконец, заговорили о серьезнейших проблемах, существующих в отрасли [1]. Интернет вещей держится несколько обособленно от обычного процесса разработки программного обеспечения – ведь это слияние кода и физических объектов, данных и «вещей». Следовательно, и процессы создания подобных систем несколько отличаются устоявшихся концепций разработки. Работа в сфере Интернета вещей накладывает некоторые ограничения на процесс разработки – а значит и на доступные инструменты оптимизации рабочих процессов. Целью данного исследования является выявления принципов применения некоторых практик DevOps в разработке систем Интернета вещей. В рамках исследования изучаются существующие подходы к внедрению DevOps в IoT и PoT, а также определяются проблемы внедрения рассматриваемых технологий и их дальнейшего развития.

Развитие практик автоматизации рабочих процессов в рамках разработки и развертывания программного обеспечения связано в первую очередь с ростом объема программного кода, который требуется для написания программ. С увеличением сложности проекта возникает потребность в интеграции и частой проверке слаженной работы

компонентов программного обеспечения. Со стороны конечного пользователя можно отнести данные тезисы к следующим ключевым проблемам:

- Длинный цикл релиза
- Низкая скорость расследования инцидентов
- Ошибки на производственной среде
- Недоступность сервиса

Эти факторы поспособствовали развитию практик, получивших название непрерывного развертывания (Continuous Deployment). Процессы непрерывного развертывания состоят из 3 основных компонент – непрерывная интеграция (Continuous Integration, CI), непрерывная доставка (Continuous Delivery, CD) и автоматизированное развертывание на production-сервере (рис. 1). Наиболее распространена модель, состоящая только из непрерывной интеграции и непрерывной доставки, так называемой практике CI/CD, так чаще всего эти этапы полностью удовлетворяют основным потребностям бизнеса и просты во внедрении.

В своей статье Мартин Фаулер описывает CI так: “... практика разработки программного обеспечения, когда участники группы осуществляют частую интеграцию своих работ” [2].

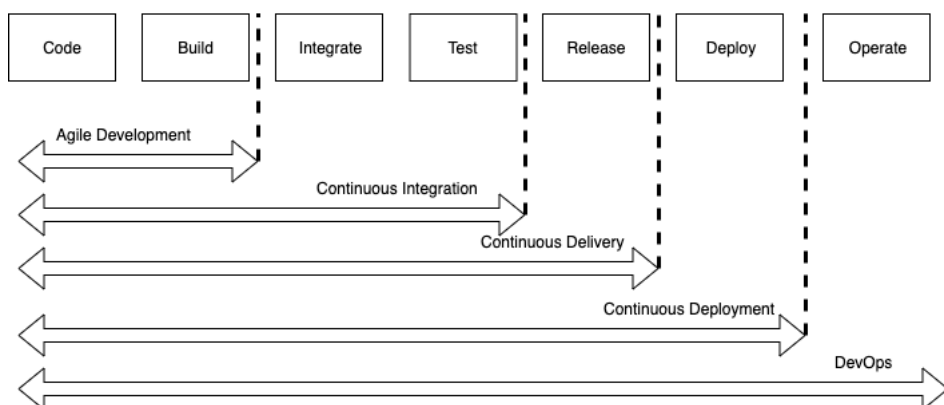


Рис. 1. Процесс внедрения нового функционала в рамках CI/CD процессов

Непрерывная интеграция является ключевым компонентом практики CI/CD и Agile Development в целом. Основой данной практики является постоянное попадание кода в центральный репозиторий после успешного запуска тестов. При непрерывной интеграции разработчики часто заливают свои изменения в центральный репозиторий, выполняя до этого модульное тестирование (юнит-тесты). Система контроля версий автоматически проверяет код на возможность безопасной интеграции с существующим в репозитории.

Непрерывная поставка — расширение непрерывной интеграции, позволяющее сохранять уверенность в том, что вы можете быстро и предсказуемо вносить изменения в продукт для ваших клиентов. Это подразумевает, что производится автоматизация не только процесса тестирования, но и процесс выпуска продукта, а потому возможно развертывание приложения в любой момент времени, одним нажатием кнопки. Непрерывная поставка позволяет снизить риски релизов, сократить время поставки новой функциональности, путем автоматизации большинства операций, таких как развертывание, настройка окружения, тестирование, повысить качество создаваемого программного обеспечения за счет того, что команда может больше времени уделять на исследовательское тестирование, юзабилити тестированию, тестированию производительности и безопасности. Инвестируя в сборку, тестирование и развертывание, разработчики добиваются снижения стоимости создания и доставки инкрементальных изменений программного обеспечения, устраняя многие постоянные затраты, связанные с процессом выпуска ПО. Для организации процесса непрерывного развертывания на выделенном сервере запускается служба, в задачи которой входят:

- получение исходного кода из репозитория;

- сборка проекта;
- выполнение тестов;
- развёртывание готового проекта;
- отправка отчетов.

Интернет вещей (Internet of Things, IoT) — концепция вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключая из части действий и операций необходимость участия человека. Годом зарождения Интернета вещей стоит назвать 2009 год – в этот момент количество подключенных устройств превысило численность населения Земли [5]. Интернет вещей – один из ключевых трендов развития технологий и одна из самых быстрорастущих реализаций применения аппаратно-программных комплексов, к 2020 году по прогнозам ожидается подключение порядка 50 миллиардов устройств [3]. Архитектура экосистемы IoT состоит из 4 главных компонент: датчиков, данных, облачных центров обработки данных (ЦОД) и соединения типа M2M (Machine-To-Machine) [8]. При этом каждая из компонент так или иначе связана с непосредственным написанием программного кода: описание протокола передачи и обработки данных от датчиков, характер взаимодействия датчиков с MQTT-брокером, настройка работы ЦОД на макро и микроуровне, передача команд к «вещам».

DevOps технологии, которые были описаны ранее, на первый взгляд, вписываются в формат быстрорастущей отрасли Интернета вещей, так как одним из ключевых преимуществ CI/CD практик является масштабируемость. Однако процесс разработки в рамках Интернета

вещей несколько отличается от разработки иных аппаратно-программных комплексов. В рамках IoT для каждого отправленного в репозиторий изменения необходимо автоматизировать:

- сборку, чтобы код мог предоставить валидный бинарный файл для устройства;
- тестирование, чтобы убедиться, что очередные изменения не сломали приложение и оно работает в производственной среде так, как ожидается;
- развертывание, чтобы имелась возможность обновлять миллионы устройств в любой точке мира;
- создание истории коммитов, логирование и мониторинг, чтобы отслеживать все изменения или отклонения в продуктовой среде;

В серии статей Костас Пипилас рассмотрел наиболее популярные CI/CD платформы и возможность их применения в рамках Интернета вещей [4]. Существует множество инструментов и сервисов CI / CD, таких как Jenkins, TeamCity, Travis CI, Gitlab CI, CircleCI, Bamboo, Codefresh, Codeship и другие. Все они обеспечивают конфигурацию CI/CD процесса, предоставляют способ запуска тестов, ограниченных API взаимодействия устройства с остальным миром. Имеются в том числе и решения, которые напрямую созданы для управления и конфигурирование IoT проектов (например решения компании Valena, в котором хорошо проработана CD составляющая процесса непрерывного развертывания). Однако автоматизирование функционального тестирования аппаратно-программных комплексов IoT в данных инструментах не заложено, следовательно, необходимо каждый раз проводить функциональное тестирование вручную, что является большой тратой денежных и временных ресурсов. Чаще всего для внедрения CI/CD на производстве IoT использовались несколько независимых программных модулей,

отвечающих за некоторый отдельный процесс в общем макропроцессе непрерывного развертывания. Полноценное решение, удовлетворяющее ранее указанным требованиям и решающее проблему функционального тестирования, появилось в конце 2018 года. Платформа SpannerCI позволяет производить сборку, полный цикл тестирования, мониторинг, логирование, а также реализовать проведение тестов функций готового IoT решения при помощи программируемого аппаратного модуля собственной разработки [6]. Основой решения SpannerCI, позволяющего проводить функциональные тесты является программируемый модуль – Spanner Test Board. Данный элемент представляет собой подобие платы, на которой расположен программируемый модуль и к которой можно присоединить «вход» и «выход» испытуемого образца. Для написания тестов используются популярные веб-сервисы для написания апплетов, например If This Then That (IFTTT), Amazon Web Services (AWS) и другие. Spanner Test Board выступает цифровым преобразователем исходных сигналов устройства в значения, которые могут быть использованы для разного рода функциональных тестов. Но подобное решение еще будет требовать доработок. Например, как взаимодействовать с подобной тестовой панелью очень маленьким устройствам или что делать, если результат реализации функционала устройства связан не напрямую с устройством, а, например, с окружающей средой или некоторым иным неподключенным объектом. Хорошим примером является концепция «умной пыли» - самоорганизующихся крошечных устройств (групповых роботов), обменивающихся беспроводными сигналами и работающих как единая система.

Особняком в вопросе применения комплексов непрерывной интеграции и доставки стоит индустриальное применение концепции

Интернета вещей – промышленный Интернет вещей (Industrial Internet of Things, IIoT) [7]. Говоря о комплексах IIoT нужно понимать, что речь идет о высоконагруженных отказоустойчивых системах, для которых каждое изменение системы – это довольно крупное событие. Релиз обновлений должен проходить без какого-либо влияния на рабочий процесс, ведь остановка предприятия на время релиза – это колоссальные убытки, не говоря о неудавшемся релизе и сбоях в процессе установки. В рамках IIoT CI/CD нужен в первую очередь для того, чтобы уметь противостоять новым проблемам безопасности, о которых еще год назад и подумать не могли. Темп развития технологий обязывает разработчиков создавать решения, которые будут способны отвечать не только уже существующим, но и новейшим методам хакерских атак, а значит системы следует разрабатывать при помощи методик, позволяющих адаптировать решение под изменившиеся требования, в том числе по безопасности, с минимальной потерей в сроках релиза версии. При подходе, аналогичном используемому в SpannerCI, запуск функционального тестирования в процессе разработки на этапе выхода релиза на production-сервер может оказаться дорогим или вообще невозможным на момент разработки программного обеспечения. Довольно сложно представить возможность выделения некоторого «входа» и «выхода» для крупного машиностроительного производства или по крайней мере станка. Кроме того, встает вопрос о действительном запуске станка при реализации подобного тестирования, что может являться нецелесообразным. Однако оставлять функциональное тестирование последним этапом разработки перед приемочным тестированием заказчиком – довольно рискованный шаг, который может увеличить сроки разработки при обнаружении какого-то рода ошибки.

Таким образом, для развития CI/CD в IoT и, в частности, в IIoT, нужно не только развитие технического прогресса, но и в том числе переосмысление самой концепции разработки подобных систем. Каскадная модель разработки позволяет качественно прописать техническое задание, проработать архитектуру и детали реализации проекта, тем самым обработав все предполагаемые критические узлы программного комплекса. Разработанный монолит не так часто подвергается изменениям, тем самым становясь уязвимым для новых атак по уязвимым местам IIoT системы. Agile Development, в свою очередь, концентрирует внимание разработки на возможность изменений в том числе по причине влияния внешних факторов, однако разработка с использованием Agile в большинстве своём не имеет конкретных сроков и таймлайнов, что непозволительно ввиду дороговизны создания и внедрения аппаратно-программных комплексов IIoT. Ключевая технологическая и методологическая проблема – как реализовывать функциональное тестирование аппаратно-программных модулей промышленного Интернета вещей, чтобы минимизировать затраты на тестирование с целью возможности внедрения CI/CD технологий и полноценного непрерывного развертывания на производстве. Ответ на этот вопрос предстоит дать специалистам в ближайшее время – так как темпы развития IIoT требуют масштабирования разработки.

Таким образом, вопрос внедрения практик непрерывного развёртывания в IIoT очень важен, поскольку разработка таких программных комплексов занимает много времени, а значит вероятность устаревания некоторых программных решений может привести к созданию уязвимых точек в отказоустойчивом комплексе. На данный момент каких-то готовых решений, аналогичных SpannerCI не предвидится ввиду высокой сложности создания подобного

программируемого модуля для использования на промышленном объекте, а также дороговизны проведения подобных функциональных тестов. Можно утверждать, что в процессах разработки IoT систем проблема внедрения непрерывного развёртывания упирается не только в технологический, но и в методологический аспект, так как ни каскадная, ни гибкая методология не могут в полном объеме удовлетворять специфике разработки в рамках IoT.

Библиографический список

1. 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr [Электронный ресурс] // O'Reilly Velocity: Web Performance and Operations Conference, URL: <https://conferences.oreilly.com/velocity/velocity2009/public/schedule/detail/7641> (дата обращения: 13.05.2019)
2. Continuous Integration [Электронный ресурс] // Martin Fowler.com, URL: <https://martinfowler.com/articles/continuousIntegration.html> (дата обращения: 15.03.2019)
3. The most promising Internet of Things trends for 2018 [Электронный ресурс] // Hackernoon, URL: <https://hackernoon.com/the-most-promising-internet-of-things-trends-for-2018-10a852ccd189> (дата обращения 08.04.2019)
4. Hands-On IoT DevOps: Day 1 [Электронный ресурс] // DZone, URL: <https://dzone.com/articles/iot-devops-hands-on-day-1?fromrel=true> (дата обращения: 17.04.2019)
5. The Internet of Things. How the Next Evolution of the Internet Is Changing Everything [Электронный ресурс] // CISCO, URL: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (дата обращения: 07.04.2019)

6. SpannerCI. Finally. Continuous Integration for IoT. [Электронный ресурс] // SpannerCI, URL: <https://www.spannerci.com/> (дата обращения: 18.04.2019)
7. Цифровые платформы управления жизненным циклом комплексных систем / ред. В.А.Тупчиенко. - М.: Научный консультант, 2018. - 439 с.
8. Куприяновский В.П. Интернет Вещей на промышленных предприятиях / В.П. Куприяновский, Д.Е. Намиот, В.И. Дрожжинов, Ю.В.Куприяновская, М.О. Иванов // International Journal of Open Information Technologies. - 2016. - №12. - URL: <https://cyberleninka.ru/article/n/internet-veschey-na-promyshlennyh-predpriyatiyah> (дата обращения: 13.04.2019).

Оригинальность 76%