

сигналов с датчиков и отправки управляющих сигналов на Arduino – это цель выполнения лабораторной работы.

Таким образом, разработана концепция построения информационной системы удаленного доступа к лабораторному оборудованию. На кафедре инженерной психологии и эргономики БГУИР реализован пилотный проект по удалённому доступу к лабораторному стенду «Робот-автомобиль». Применение данной разработки в учебном процессе позволит получить студентами дистанционного обучения практических навыков и умений в работе с лабораторным оборудованием.

Список литературы

1. Arduino [Электронный ресурс] – Режим доступа : <http://arduinoblog.com/what-is-arduino/>.
2. Спецификация RPLidar [Электронный ресурс] – Режим доступа : <https://www.robotshop.com/media/files/pdf/datasheet-rplidar.pdf>.
3. Hector SLAM4 [Электронный ресурс] – Режим доступа : <http://www.sim.informatik.tu-darmstadt>.

REALIZATION OF REMOTE ACCESS TO LABORATORY STAND «ROBOT-CAR»

Malinina T.A., Osipovich V.S., Krishtopova E.A.

Belarusian State University of Informatics and Radioelectronics

Abstract. The laboratory stand "Robot car" was made. The concept of software for controlling the movement of a vehicle along a ring road has been developed.

Keywords: robotics, machine learning, laboratory stand, remote access.

УДК 004.89

DEVOPS ИНЖЕНЕРЫ КАК ПЕРСПЕКТИВНОЕ НАПРАВЛЕНИЕ ОБУЧЕНИЯ

Малич К.В., Куликов С.С.

Белорусский государственный университет информатики и радиоэлектроники

Аннотация. В статье рассмотрено явление DevOps - набор практик, нацеленных на активное взаимодействие специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимную интеграцию их рабочих процессов друг в друга. Показаны перспективы изучения DevOps как части формирования компетенций молодого специалиста.

Ключевые слова: изучение современных технологий, интеграция процессов, управление проектами.

Движение DevOps начало формироваться в 2007-2008 годах, когда сообщества специалистов по ИТ-операциям и разработчиков программного обеспечения, наконец, заговорили о серьёзнейших проблемах, существующих в отрасли.

Они выступали против традиционной модели разработки программного обеспечения, предписывающей организационное и функциональное разделение между теми, кто пишет код, и теми, кто выполняет развёртывание и поддержку этого кода.

В настоящее время все больше и больше приложений сталкиваются с проблемами производительности при высоких нагрузках. Порой не всегда достаточно оптимизировать программный код для достижения быстрого действия системы. В таких случаях применяют архитектурные и инфраструктурные подходы, используют различные сервисы для повышения производительности и сокращения времени отклика системы.

Далеко не каждый специалист в IT сфере способен грамотно заложить архитектуру системы таким образом, чтобы избежать указанных выше проблем в дальнейшем; также не всегда ему известны различные сервисы, которые позволят оптимизировать рабочие процессы во время разработки. Для решения таких задач привлекают DevOps инженеров, имеющих багаж знаний в этой области.

Поскольку DevOps - это командная работа (между сотрудниками, занимающимися разработкой, операциями и тестированием), нет единого инструмента DevOps: это скорее набор, состоящий из нескольких инструментов. Как правило, инструменты DevOps вписываются в одну или несколько из этих категорий, что отражает ключевые аспекты разработки и доставки программного обеспечения.

Такие инструменты, как Docker (контейнеризация), Jenkins (непрерывная интеграция), Puppet (инфраструктура как код) и Vagrant (платформа виртуализации) - и многие другие - часто используются и часто упоминаются в дискуссиях по инструментам DevOps.

Однако большинство DevOps инженеров получили свои знания в процессе самообучения и самообразования, путём проб и ошибок. В данной работе раскрывается важность обучения специалистов этой сферы.

Обучение должно быть направлено на изучение таких сервисов, как Azure и Amazon и всех их возможностей, реляционных и нереляционных баз данных, различных веб серверов, Docker и других контейнеров и многих других инструментов, необходимых для автоматизации процесса разработки [2].

Конкретные цели DevOps охватывают весь процесс поставки программного обеспечения. Они включают: сокращение времени для выхода на рынок, снижение частоты отказов новых релизов, сокращение времени выполнения исправлений, уменьшение количества времени на восстановление (в случае сбоя новой версии или иного отключения текущей системы).

Интеграция DevOps предназначена для доставки продукта, непрерывного тестирования, тестирования качества, разработки функций и обновлений обслуживания для повышения надёжности и безопасности и обеспечения более быстрого цикла разработки и развёртывания.

DevOps даёт преимущества в управлении выпуском программного обеспечения для организации путём стандартизации среды разработки. События можно более легко отслеживать, а также разрешать документированные процессы управления и подробные отчёты. Подход DevOps предоставляет разработчикам больше контроля над средой, предоставляя инфраструктуре более ориентированное на приложения понимание.

Компании, которые используют DevOps, сообщили о значительных преимуществах, в том числе: значительном сокращении времени выхода на рынок, улучшении удовлетворённости клиентов, улучшении качества продукции, более надёжных выпусках, повышении производительности и эффективности, а также увеличении способности создавать правильный продукт путём быстрого экспериментирования.

Однако, исследование, выпущенное в январе 2017 года компанией "F5 Labs", на основе опроса почти 2200 ИТ-руководителей и специалистов отрасли, показало, что только один из пяти опрошенных полагает, что DevOps оказывает стратегическое влияние на их организацию, несмотря на рост использования. В том же исследовании было установлено, что только 17 % определили DevOps как ключевой инструмент [3].

Чтобы эффективно использовать DevOps, прикладные программы должны соответствовать набору архитектурно значимых требований (ASR), таких как: возможность развёртывания, изменяемость, тестируемость и возможности мониторинга.

Хотя в принципе можно использовать DevOps с любым архитектурным стилем, архитектурный стиль микросервисов становится стандартом для построения постоянно развёрнутых систем. Поскольку размер каждой услуги невелик, он позволяет создавать архитектуру отдельного сервиса посредством непрерывного рефакторинга, что уменьшает необходимость в большом предварительном дизайне и позволяет выпускать программное обеспечение на ранней стадии непрерывно.

Необходимые качества DevOps инженера: аналитический склад ума, стрессоустойчивость, умение не сдаваться даже в безвыходных ситуациях. Возможные карьерные пути DevOps инженера: расти как DevOps специалист; углубляться в специализацию и осваивать смежные технологии; переквалифицироваться в разработчики, если начинали как системный администратор; переквалифицироваться в сисадмины, если начинали как разработчик (если интересно больше работать с инфраструктурой, чем с разработкой); переквалифицироваться в инженеры по IT-безопасности; также открыты пути в системные архитекторы, тестировщики (в том числе автоматизаторы), проектные менеджеры.

Как видно из выше сказанного, в настоящее время обучение и подготовка DevOps инженеров имеет высокую важность как для построения рабочих процессов во время разработки, так и для автоматизации ряда действий во время разработки. Задача DevOps - сделать процесс разработки и поставки программного обеспечения согласованным с эксплуатацией, часто эти задачи решаются при поддержке автоматических средств.

Литература

1. Mitesh Soni: DevOps for Web Development. Packt Publishing, 2016 (24 October 2016).
2. Christopher Weller: Devops Handbook: Simple Step by Step Instructions to Devops. CreateSpace Independent Publishing Platform, 2017 (7 August 2017).
3. Sricharan Vadapalli: DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive into the core DevOps strategies. Packt Publishing Ltd, 2018 (13 March 2018)

DEVOPS ENGINEERS AS A PERSPECTIVE TRAINING DIRECTION

Malich K.V., Kulikov S.S.

Belarusian State University of Informatics and Radioelectronics

Abstract. The article describes the DevOps phenomenon - a set of practices aimed at the active interaction of development specialists with information technology service specialists and the mutual integration of their work processes into each other. The prospects of studying DevOps as part of the formation of competencies of a young specialist are shown.

Keywords: study of modern technologies, integration of processes, project management.

УДК 101+008

ФИЛОСОФИЯ КАК ИНТЕЛЛЕКТУАЛЬНЫЙ РЕСУРС МЕЖКУЛЬТУРНОЙ КОММУНИКАЦИИ

Малыхина Г.И.

*Белорусский государственный университет
информатики и радиоэлектроники*

Аннотация. Исследуется интернациональное значение философского образования. Проводится сравнительный анализ интеллектуальных и культурных традиций Востока и Запада. Конкретизируется роль средневековой арабоязычной философии в рецепции и дальнейшем развитии философского наследия Античности.

Ключевые слова: философия, интеллектуальная традиция, античная философия, арабское средневековье, диалог культур, межкультурная коммуникация.

Одной из форм интернационализации образования является изучение мировой философии. Творчество того или иного философа интересно не только (и не столько) само по себе, сколько в исторической панораме мировой философии, в контексте сравнения и диалога философских культур Запада и Востока с целью нахождения путей к взаимопониманию и взаимообогащению.