**Домашнее задание к занятию «1.1. Введение в DevOps – Алексей Гуценко»**

**Задание №2 - Описание жизненного цикла задачи (разработки нового функционала)**

Описанию жизненного цикла задачи хорошо соответствует так называемая общепринятая модель жизненного цикла программного обеспечения, согласно которой программные системы проходят в своем развитии две фазы:

* разработка
* сопровождение

Фазы разбиваются на ряд этапов (рис. 1)

|  |
| --- |
|  |

Разработка начинается с идентификации потребности в новом функционале, а заканчивается передачей продукта разработки в эксплуатацию.

Первым этапом фазы разработки является постановка задачи и определение требований. Определение требований включает описание общего контекста задачи, ожидаемых функций системы и её ограничений. На этом этапе заказчик совместно с разработчиками принимают решение о создании системы.

В случае положительного решения начинается этап спецификации системы в соответсвии с требованиями. Назначение этих спецификаций – описывать внешнее поведение разрабатываемой системы, а не её внутреннюю организацию, т.е. отвечать на вопрос, что она должна делать , а не как это будет реализовано. Задача этапа спецификаций состоит в том , чтобы описание программы выстроить в виде логически выверенной системы, понятной как для заказчика , будущих пользователей , так и для исполнителей проекта.

Разработка проектных решений , отвечающих на вопрос , как должна быть реализована система, чтобы она могла удовлетворять специфицированным требованиям, выполняется на этапе проектирования. Поскольку сложность системы в целом может быть очень большой, главной задачей этого этапа является разбитие процесса разработки до уровня очевидно реализуемых модулей (спринтов).

На следующем этапе реализации , или кодирования каждый из этих модулей программируется на наиболее подходящем для данного приложения языке.

Фаза разработки заканчивается этапом тестирования (автономного и комплексного) и передачей системы в эксплуатацию.

Фаза эксплуатации и сопровождения включает в себя всю деятельность по обеспечению нормального функционирования ПО , в том числе фиксирование вскрытых во время исполнения программ ошибок (багов) , поиск их причин и исправление, повышение эксплуатационных характеристик системы.

DevOps инженер помогает автоматизировать все перечисленные этапы создания программного обеспечения: от разработки , сборки и тестирования до развертывания, выпуска и доставки пользователям. За счёт этого команда работает эффективнее и выпускает более качественные цифровые продукты. DevOps автоматизирует многие рутинные операции вроде тестирования и выкатки продукта в продакшн. В итоге команда разработки тратит меньше времени на рутину и может сфокусироваться на стратегических задачах.

Основные задачи DevOps - инженера:

* планирование , проектирование , создание сценариев и инструментов разработки;
* внедрение автоматизации;
* участие в разработке приложений и инфраструктуры;
* административная деятельность , включая управление конфликтами внутри команды или с клиентом (заказчиком);
* мониторинг работы продукта после релиза.

Основные цели DevOps - инженера:

* уменьшение времени, затраченного командой на разработку ПО;
* повышение безопасности разработки;
* минимизация рисков ошибок в новых релизах;
* поиск багов и своевременное их исправление;
* координация работы команды и автоматизация процессов разработки выхода ПО в продакшн.