Рекомендации по созданию ИТ-ландшафта. Команда Dependonyou

1. Общие рекомендации

При замене устаревших частей архитектуры необходимо руководствоваться принципами унификации, масштабируемости, безопасности и производительности, помимо этого при выборе решений необходимо учитывать их стоимость, функциональность, надежность и простоту обслуживания.

2. Конкретные решения

Монолитные приложения:

- Замена на микросервисную архитектуру.
- Использование контейнеров и Kubernetes для оркестрации микросервисов.

Устаревшие СУБД:

- Миграция на более современные СУБД, такие как PostgreSQL, MySQL, MongoDB.
- Использование облачных СУБД.

Неэффективная инфраструктура:

- Переход на облачную инфраструктуру.
- Использование инструментов автоматизации для управления инфраструктурой.

Устаревшие языки программирования:

- Переход на более современные языки программирования, такие как Python, Java, Go.
- Использование фреймворков и библиотек для ускорения разработки.

3. Примеры

1. Монолитные приложения:

- Пример 1: Монолитное приложение "Интернет-банк" было заменено на аналогичное на основе микросервисной архитектур. Это позволило:
    - Разделить приложение на независимые сервисы, что упростило разработку и поддержку.
    - Масштабировать сервисы по отдельности, что повысило производительность и отказоустойчивость.
    - Ускорить внедрение новых функций, что позволило банку быстрее реагировать на изменения рынка.
- Пример 2: Монолитная система "Процессинг" была переработана с использованием API-шлюза. Это позволило:
    - Предоставить доступ к функциональности системы внешним разработчикам.
    - Создать новые приложения и сервисы на основе данных системы.
    - Повысить интеграцию системы с другими ИТ-системами банка.

2. Устаревшие СУБД:

- Пример 1: СУБД "Oracle" в системе "ФО-система" была мигрирована на PostgreSQL. Это позволило:
    - Снизить расходы на лицензирование и поддержку СУБД.
    - Повысить производительность системы за счет использования более современных функций PostgreSQL.
    - Улучшить масштабируемость системы за счет возможности горизонтального масштабирования PostgreSQL.
- Пример 2: СУБД "MS SQL Server" в системе "КХД" была заменена на MongoDB. Это позволило:
    - Увеличить скорость работы системы за счет использования NoSQL-подхода
    - Повысить масштабируемость системы за счет возможности горизонтального масштабирования MongoDB
    - Упростить разработку и внедрение новых функций системы.

3. Неэффективная инфраструктура:

- Пример 1: Инфраструктура банка была перенесена в облако AWS. Это позволило:
    - Снизить расходы на содержание собственного ЦОД.
    - Повысить масштабируемость инфраструктуры за счет возможности быстрого выделения и освобождения ресурсов.
    - Повысить отказоустойчивость инфраструктуры за счет использования облачных сервисов AWS.
- Пример 2. Внедрена система мониторинга и управления инфраструктурой Zabbix. Это позволило:
    - Повысить прозрачность ИТ-инфраструктуры.
    - Своевременно выявлять и устранять проблемы в работе ИТ-систем.
    - Оптимизировать использование ИТ-ресурсов.

4. Устаревшие языки программирования:

- Пример 1: Язык программирования "Delphi" в системе "АБС" **может быть заменен** на Python. Это позволило:
    - Ускорить разработку и внедрение новых функций системы
    - Улучшить читаемость и сопровождаемость кода системы
    - Снизить расходы на поддержку системы.
- Пример 2: Язык программирования "1С" в системе "Управление складом" был заменен на Java. Это позволило:
    - Повысить производительность системы.
    - Улучшить масштабируемость системы.
    - Повысить безопасность системы.