

Манифест COA

Сервис-ориентация -это парадигма, которая описывает то, что вы делаете.

Сервис-ориентированная архитектура (COA) -это такой тип архитектуры, который приводит к конечному результату за счет применения сервис-ориентации.

Мы применяем сервис-ориентацию, чтобы помочь компаниям последовательно достичь устойчивой значимости бизнеса, повышенной гибкости и эффективной стоимости, в соответствии с изменяющимися потребностями бизнеса.

В процессе работы мы установили следующие приоритеты:

Значимость бизнеса перед технической стратегией.

Стратегические цели перед выгодами, специфичными для данного проекта.

Свойственная способность к взаимодействию перед специально достигнутой интеграцией.

Совместно используемые сервисы перед разработками с узкими, специальными целями.

Гибкость перед оптимизацией.

Эволюционные усовершенствования перед попыткой достичь изначального совершенства.

Это значит, что мы ценим значение выше перечисленных понятий справа, однако ещё более ценим понятия слева.

Руководящие Принципы

Мы следуем пересчисленным ниже принципам:

Уважать социальную и руководящую структуру компании.

Признать то, что COA, в конечном счете, требует внесение изменений на многих уровнях.

Масштабы внедрения COA могут варьироваться. Держать усилия в этом направлении под контролем и в разумных рамках.

[Программные] продукты и стандарты сами по себе не дадут вам COA и не применят сервис-ориентацию за вас.

COA может быть реализована путём применения различных технологий и стандартов.

Установить единый комплекс стандартов и положений предприятия, основанных на отраслевых стандартах, общественных стандартах и стандартах, принятых "по- факту".

Преследовать внешнее единообразие, одновременно позволяя внутреннее многообразие.

Идентифицировать сервисы путём взаимодействия с заинтересованными представителями бизнеса и технологии.

Максимизировать использование сервиса путём рассмотрения текущей и будущей сферы применения.

Контролировать, чтобы сервисы удовлетворяли требованиям и целям бизнеса.

Развивать сервисы и их организацию в ответ на их реальное использование.

Разделять различные аспекты системы, которые изменяются различными темпами.

Сокращать неявные зависимости и публиковать все явные зависимости с целью увеличения прочности и снижения последствий изменения.

Организовать каждый сервис в соответствии с единой, управляемой единицей функциональности на каждом уровне абстракции.

Авторы

Ali Arsanjani
Grady Booch
Toufic Boubez
Paul C. Brown
David Chappell
John deVadoss

Thomas Erl
Nicolai Josuttis
Dirk Krafzig
Mark Little
Brian Loesgen
Anne Thomas Manes

Joe McKendrick
Steve Ross-Talbot
Stefan Tilkov
Clemens Utschig-Utschig
Herbjörn Wilhelmsen

Переводчик

Леонид Феликсон (Leonid Felikson)