

SOA-Manifest

Service-Orientierung ist ein Paradigma, das den Rahmen für unser Handeln vorgibt.

Service-orientierte Architektur (SOA) ist ein Architekturtyp,
der aus der Anwendung von Service-Orientierung entsteht.

Wir haben Service-Orientierung angewendet, um Organisationen zu helfen, kontinuierlich nachhaltigen Geschäftswert zu liefern mit höherer Agilität und Kosteneffizienz und im Einklang mit den sich ändernden fachlichen Bedürfnissen.

Im Rahmen unserer Arbeit sind wir zu folgender Priorisierung gekommen:
Geschäftswert über technische Strategie

Strategische Ziele über projektspezifischen Nutzen

Immanente Interoperabilität über maßgeschneiderte Integration

Gemeinsam verwendete Services über zweckgebundene Implementierungen

Flexibilität über Optimierung

Evolutionäre Vervollkommnung über Streben nach anfänglicher Perfektion

Das heißt, obwohl wir die Werte auf der rechten Seite schätzen,
sind uns die Werte auf der linken Seite wichtiger.

Leitprinzipien

Wir folgen den folgenden Prinzipien:

Respektiere die Sozial- und Machtstruktur der Organisation.

Erkenne an, dass SOA letztlich Veränderung auf vielen Ebenen bedeutet.

Der Bereich, in dem SOA eingeführt wird, kann unterschiedlich ausfallen.
Halte die Aufwände in einem überschaubaren
und sinnvollen Rahmen.

Produkte und Standards alleine werden weder SOA liefern noch das Service-orientierte Paradigma umsetzen.

SOA kann mit unterschiedlichen Technologien und Standards umgesetzt werden.

Etabliere einheitliche Unternehmensstandards und -richtlinien auf der Basis von Industrie- und De-facto-Standards sowie Standards der SOA-Gemeinde.

Strebe nach außen Einheitlichkeit an, aber lasse nach innen Vielfalt zu.

Identifiziere Services durch Zusammenarbeit zwischen fachlichen und technischen Interessenvertretern.

Maximiere die Anwendbarkeit von Services durch Berücksichtigung der derzeitigen und zukünftigen Anwendungsgebiete.

Stelle sicher, dass Services fachlichen Anforderungen und Zielen dienen.

Services und deren Ausgestaltung sollten sich anhand der Art und Weise, wie sie wirklich genutzt werden, entwickeln.

Trenne die verschiedenen Aspekte eines Systems, die sich unterschiedlich häufig ändern.

Reduziere implizite Abhängigkeiten und publiziere alle externen Abhängigkeiten, um Robustheit zu fördern und die Auswirkungen von Veränderungen zu reduzieren.

Organisiere jeden Service auf jeder Abstraktionsebene in oder anhand einer zusammenhängenden und überschaubaren Funktionseinheit.

Autoren

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

Übersetzer

Nicolai Josuttis
Dirk Krafzig
Stefan Tilkov
Clemens Utschig-Utschig