

# SOA Manifesto

Background, discussions and result



Herbjörn Wilhelmsen

# Background

Community discussions

What is



SOA Manifesto presented @



Thomas Erl



Dagstuhl

We need a manifesto!

Gregor Hohpe (Google)

SOA is dead  
<http://apsblog.burtongroup.com/2009/01/soa-is-dead-long-live-services.html>  
Anne Thomas Manes

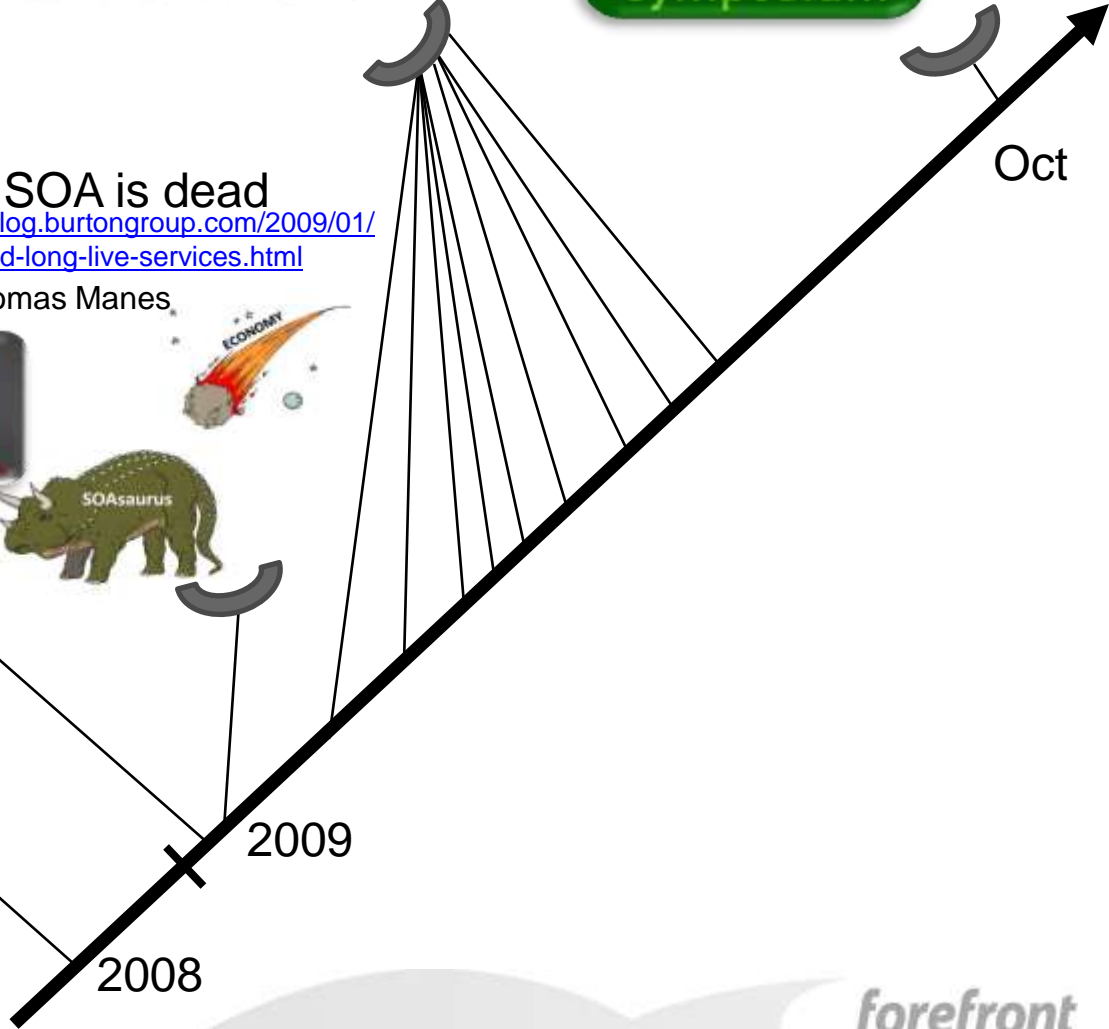


SOA has matured  
[http://en.wikipedia.org/wiki/Hype\\_cycle](http://en.wikipedia.org/wiki/Hype_cycle)  
Gartner hype cycle



2009-11-27

2



# The model: Agile Manifesto

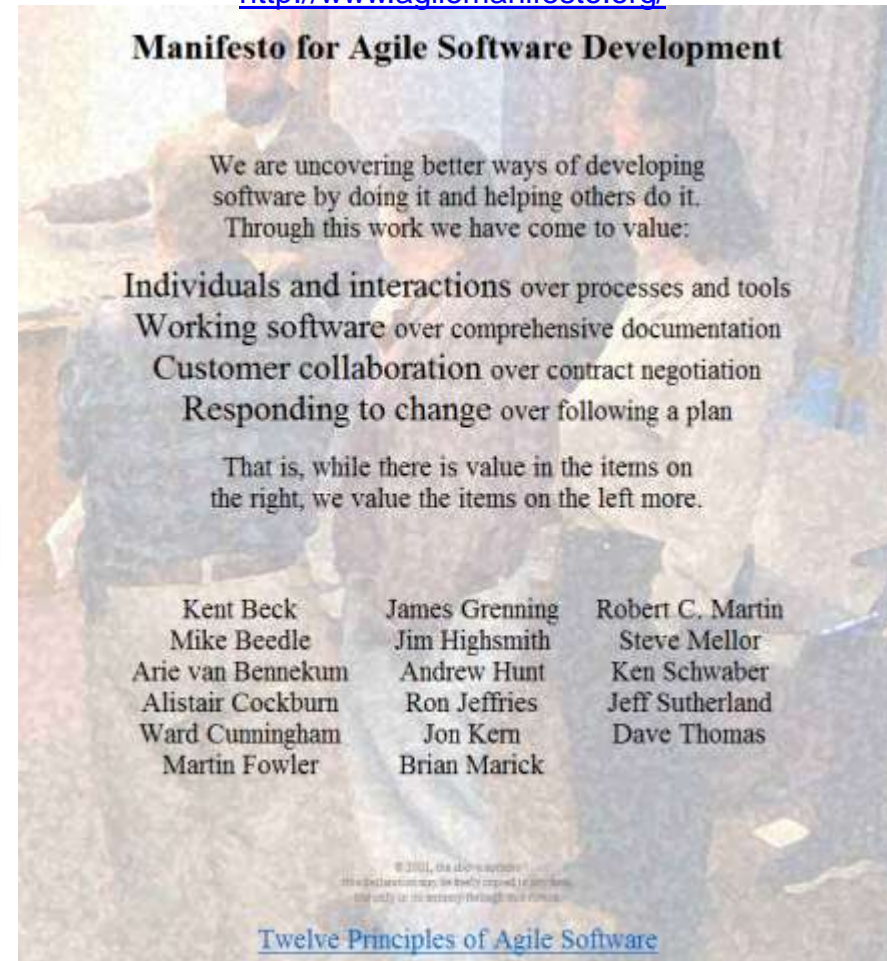
- Has an interesting format:

- Short
- Defines core values
- Highlights hard choices
- Principles builds and expands on values

<http://www.ddj.com/architect/184414755>

- Spurred a lot of discussion in the agile world

<http://www.agilemanifesto.org/>





# Discussions



# SOA Manifesto - preamble

” Service orientation is a paradigm that frames what you do.  
Service-oriented architecture (SOA) is a type of architecture  
that results from applying service orientation. “

SOA is the result of working with design. You cannot buy SOA

“We have been applying service orientation to help  
organizations consistently deliver sustainable business value,  
with increased agility and cost effectiveness, in line with  
changing business needs.”

SOA can improve Business Agility. It is key to the SOA philosophy.

# SOA Manifesto - value statements (1 of 2)

- **Business value** over technical strategy
  - technical strategy must help you to provide business value or be ~~abandoned~~ modified, building the perfect framework is not the goal of SOA, SOA may not be the best way to accommodate all business needs for all organizations
- **Strategic goals** over project-specific benefits
  - the needs of the many over the needs of the few, over-arching goals are more important than that of just one project (if we need to choose), create business value but do not use short-cuts that messes up your strategy, implementations are on a project level – but SOA requires an enterprise perspective for project success to evolve into SOA success
- **Intrinsic interoperability** over custom integration
  - designing for intrinsic interoperability will reduce cost and development speed over time, services needs to be compatible (service may consume another service, consumer may consume several services) and this requires a design effort, existing IT assets may need modifications in order to be leveraged, integration should be a first class citizen – not an afterthought

# SOA Manifesto - value statements (2 of 2)

- **Shared services** over specific-purpose implementations
  - avoid spreading related functionality, build upon what is already there, keep IT support for a process together in one service, reuse (beware of cultural issues)
- **Flexibility** over optimization
  - to deliver sustainable business value when requirements change, optimizations very often complicates design and therefore makes changes more difficult, services should be flexible in the way they can be composed, optimizations often makes assumptions about the context of use
- **Evolutionary refinement** over pursuit of initial perfection
  - it is good to get it right the first time, but understand that before we know the real usage we will not get it completely right, avoid analysis paralysis, deliver value early, we can deliver value without perfection

The primary benefit of SOA

# SOA Manifesto – principles (1 of 3)

- **Respect the social and power structure of the organization**
  - We need support from those in authority to understand how and get support for changing the way business is automated, we have to work within cultural constraints
- **Recognize that SOA ultimately demands change on many levels**
  - Collaboration across boundaries (business as well as technology), new roles and responsibilities , modifying existing systems
- **The scope of SOA adoption can vary. Keep efforts manageable and within meaningful boundaries**
  - Be pragmatic when it comes to organizational changes, technology changes, deliver value early but avoid building new silos
- **Products and standards alone will neither give you SOA nor apply the service orientation paradigm for you**
  - Modern products (ESB's / SOA Infrastructure) cannot alone give you SOA, standards (WSDL, WS-\*, REST, SAML ...) cannot ensure that your inventory is service-oriented, only service-oriented design can do that



# SOA Manifesto – principles (2 of 3)

- **SOA can be realized through a variety of technologies and standards**
  - You can build service-oriented solutions with WSDL, WS-\*, REST, CORBA, MQ or Mainframes; SOA is a kind of distributed computing but is technology -and vendor- agnostic; If you build your SOA in a technology agnostic way or not should be based on business reasons
- **Establish a uniform set of enterprise standards and policies based on industry, de facto, and community standards**
  - Use a set of standards consistently to allow for interoperability; data models and policy must also be consistent; enforce design standards
- **Pursue uniformity on the outside while allowing diversity on the inside**
  - Diversity hinders interoperability, but diversity is a fact of life that is hard to fight for large enterprises; while heterogenous environments may be inevitable we can aim for uniformity in what consumers are exposed to through the proper use of encapsulation; this allows for “best-of-breed”
- **Identify services through collaboration with business and technology stakeholders**
  - Technology solutions should be business-driven, business people can explain business goals; technology-experts can provide pragmatic advice on suitability, granularity, compatibility, definitions and other requirements + new ideas of technology adoption that may affect business
- **Maximize service usage by considering the current and future scope of utilization**
  - Assign functional borders to services; consider granularity, autonomy, scalability and how generic the functionality is

# SOA Manifesto – principles (3 of 3)

- **Verify that services satisfy business requirements and goals**
  - Monitor services to ensure they fulfill expectations; verify that they truly fulfill business needs
- **Evolve services and their organization in response to real use**
  - Services needs to evolve as requirements change; a service or a set of services may need to be refactored or reorganized; the alignment between business and technology must be maintained
- **Separate the different aspects of a system that change at different rates**
  - Separate process-specific and process-agnostic logic; isolates changes ; decreases complexity; increases maintainability
- **Reduce implicit dependencies and publish all external dependencies to increase robustness and reduce the impact of change**
  - Loose coupling; allow for moving resources, changing implementation; eliminate the need (or possibility) to understand the internals of a service
- **At every level of abstraction, organize each service around a cohesive and manageable unit of functionality**
  - Create cohesive and manageable services with well defined functional contexts; even for services that compose other services; getting the right granularity is key to manageability

# Thanks for listening

<http://www.soa-manifesto.org/>

## SOA Manifesto

Service orientation is a paradigm that frames what you do.  
Service-oriented architecture (SOA) is a type of architecture that results from applying service orientation.  
We have been applying service orientation to help organizations consistently deliver sustainable business value, with increased agility and cost effectiveness, in line with changing business needs.

Through our work we have come to prioritize:

**Business value** over technical strategy

**Strategic goals** over project-specific benefits

**Intrinsic interoperability** over custom integration

**Shared services** over specific-purpose implementations

**Flexibility** over optimization

**Evolutionary refinement** over pursuit of initial perfection

That is, while we value the items on the right, we value the items on the left more.

## Guiding Principles

*We follow these principles:*

Respect the social and power structure of the organization.

Recognize that SOA ultimately demands change on many levels.

The scope of SOA adoption can vary. Keep efforts manageable and within meaningful boundaries.

Products and standards alone will neither give you SOA nor apply the service-oriented paradigm for you.

SOA can be realized through a variety of technologies and standards.

Establish a uniform set of enterprise standards and policies based on industry, de facto, and community standards.

Pursue uniformity on the outside while allowing diversity on the inside.

Identify services through collaboration with business and technology stakeholders.

Maximize service usage by considering the current and future scope of utilization.

Verify that services satisfy business requirements and goals.

Evolve services and their organization in response to real use.

Separate the different aspects of a system that change at different rates.

Reduce implicit dependencies and publish all external dependencies to increase robustness and reduce the impact of change.

At every level of abstraction, organize each service around a cohesive and manageable unit of functionality.

## Authors

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

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

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



**forefront**  
consulting group



Herbjörn Wilhelmsen

[herbjorn@wilhelmsen.se](mailto:herbjorn@wilhelmsen.se)

<http://se.linkedin.com/in/herbjorn>

© 2009, the above authors, Creative Commons Attribution-Share Alike 3.0 Unported License