

A SOA Model & what to look for

Abstract

In many enterprises, business-IT alignment is a challenge that requires continuous attention. There is considerable literature on measuring and improving such alignment, but it falls short when applied to collaborative networked organizations. In order to facilitate managing business-IT alignment in organizations, we need a *maturity model* that allows collaborating organizations to assess the current state of alignment and take appropriate action to improve it where needed.

In this paper we propose such a model, a framework that enables benchmarking the effectiveness of a Service-oriented Architecture (SOA) implementation and its alignment with business. The SOA Maturity Model described in this paper could be viewed as a tool that defines a set of criteria and measurable parameters to measure and benchmark the effectiveness of a SOA implementation. This model is based on the Software Engineering Institute's Capability Maturity Model (CMM) developed to define specific levels of improvement in the software development process.

Table of Contents

1.	INTRODUCTION	3
2.	MATURITY LEVELS AND DIMENSIONS	3
	LEVEL DEFINITIONS	4
	MATURITY DIMENSIONS	4
3.	SOA MATURITY MODEL MATRIX.....	5
4.	CONCLUSION	8
5.	REFERENCE	9

1. Introduction

Maturity Models describe the evolution of a specific entity over time. Commonly, this entity is an organizational area or function. Maturity Models have been developed to assess specific areas against a norm. Based on maturity assessments, organizations know the extent to which activities in such areas are predictable. That is, organizations can be aware of whether a specific area is sufficiently refined and documented so the activities in this area now have the potential to achieve desired outcomes. Maturity Models apply a life-cycle approach where an area develops over time until it reaches its highest maturity level.

Essentially, Maturity Models make it easier for organizations to establish goals for process improvement and identify opportunities for optimization, since these models describe basic attributes that are expected to characterize a particular area for each maturity level. By comparing an organization's characteristics and attributes with a Maturity Model, an organization would be able to identify how mature it is in order to increase its process capability: first, establishing goals for the improvement of processes and then, taking actions to achieve them.

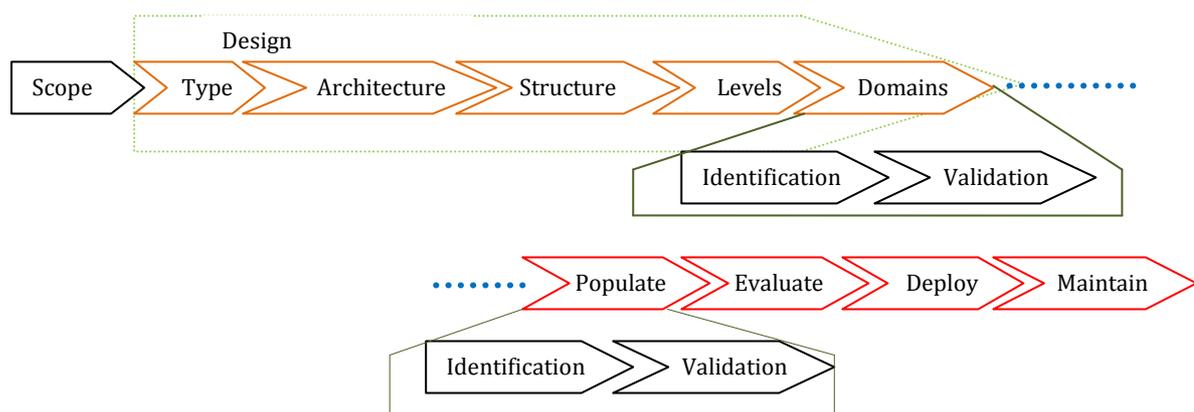
While designing a typical maturity model, the nature / type of model would need to be ascertained. Maturity models can be classified as:

1. Assessment maturity models: Consists of normative models which serve as assessment tools that target certification, and help improve the organization's image as a reliable partner
2. Development maturity models: Includes development tools that organizations could use as guides for implementing best practices that, ultimately, lead to improvements and better results

In this paper, a development maturity model is discussed that describes the various maturity levels in a SOA Adoption Journey and the ideal state of an organization at each level measured against a standard set of maturity dimensions. The resulting matrix represents the SOA Maturity Model.

2. Maturity Levels and Dimensions

Achieving a certain level of maturity as per the dimensions and levels defined below is typically a process that needs to be performed continuously and cannot be considered steady state. The development of a maturity model consists of some typical steps as depicted below:



The first step in a maturity model is of course to determine the *scope* by setting the boundaries for the model’s application and use, and to define the purpose of the model. The next steps would be to design the model and then populate it with assessment criteria. Then evaluate it for its relevance and rigor before deploying it. The maturity model (like all programs) would need to be kept up-to-date through a maintenance activity.

The SOA Maturity Model was developed using the process described above. The *scope* in this case is ‘effective SOA implementation’. Note that there is no single way to determine this and there are several SOA Maturity Models proposed by different organizations to determine the maturity of SOA in an enterprise. This particular model is based on CMM standards for development and defines 5 levels of maturity as below:

Level Definitions

1 – Primitive	This level indicates that the organization has adopted SOA, but is in a very early stage. There is not much reuse of assets and needs overall direction in achieving the goals of SOA. At this level, organizations cannot fully reap the benefits of SOA
2 – Standardized Services	This is the second level of maturity and indicates that re-use exists to a certain extent, developed through the use of standardized services and well defined best practices.
3 – Manageable Services	This level of maturity indicates that services are standardized and the organization has better control in terms of overall manageability. This level is not very far from realizing the full benefits of SOA.
4 – Measurable Services	Just like in the SEI-CMM standards, metrics and measurements play an important role in SOA. This level of maturity indicates greater control over the initiative, as metrics are available from both a technical and a commercial perspective.
5 – Agile Enterprise	This is the ultimate goal organizations aspire to achieve. This indicates close alignment between the IT and business teams so that changes are addressed quickly at low costs, thereby gaining business agility and the ability to remain competitive.

Maturity Dimensions

Each of the above levels are assessed as described below, based on collectively considering the following well-known dimensions - Organizational Factors, Tools and Processes, Architecture, Services, Governance Support and Operations.

3.SOA Maturity Model Matrix

The following table details the complete maturity model levels and their criteria

	1. Primitive	2. Standardized Services	3. Manageable Services	4. Measurable Services	5. Agile Enterprise
Organization	<p>Management has no clear vision on SOA</p> <p>IT groups are not aligned to SOA vision</p> <p>Inadequate staffing for different SOA teams (:architects, governance, service factory, business analysts)</p> <p>Management has limited budget for the SOA initiative</p>	<p>Different teams are in place and their roles and responsibilities are defined</p> <p>Setting up organization-wide SOA Center of Excellence</p>	<p>SOA Programme Manager is designated for organization-wide steer</p> <p>Costing models for Service development and re-use are identified</p> <p>Adequate Budget is allocated for the initial cost of launching SOA</p> <p>Knowledge Management Plan in place</p>	<p>IT groups are trained on the metrics</p> <p>ROI Measurement plans and strategy in place</p> <p>Based on measurement, encourages projects to reuse services with rewards and incentives</p>	<p>Management enforces trainings/workshops across the board on SOA and its benefits</p> <p>IT groups are aligned and committed to SOA vision</p> <p>Business and IT are fully aligned; IT can respond faster to changing business needs with lower costs thereby fulfilling the vision of an 'Agile Enterprise'</p>
Tools & Processes	<p>SOA Repository and Registry tools are not used</p> <p>Service Lifecycle states are not identified</p> <p>Review processes are not identified</p> <p>Service Development Process is not identified</p> <p>Service principles, versioning guidelines are not defined</p>	<p>Service design and development best practices are defined</p> <p>Interfaces are standardized</p> <p>Service designs and deliveries are reviewed thoroughly</p> <p>Service and Project dependencies are not managed</p> <p>Service deployment plan does not exist</p>	<p>SOA Repository and Registry tools are used</p> <p>Service Lifecycle states are managed using Repository</p> <p>Detailed and automated reviews of service design</p> <p>Service-level Dependency tracking is automated and project-level dependencies are managed</p>	<p>Tools are used effectively to collect service usage and performance metrics</p> <p>Runtime Governance and Management tools are used to automate usage measurement, SLA compliance reports, etc</p>	<p>Processes are fully automated</p> <p>BAM (Business Activity Monitoring) is used to understand service usage behavior for Business Intelligence and decision making</p> <p>Runtime-Governance tools (RTG) feeds data into design-time tools such as real-time service performance characteristics</p>

	1. Primitive	2. Standardized Services	3. Manageable Services	4. Measurable Services	5. Agile Enterprise
Architecture	<p>Reference Architecture does not exist</p> <p>SOA Principles for the organization is not defined</p> <p>Architectural Best practices are not defined</p> <p>Design standards are not defined</p> <p>Security Policy is not defined</p>	<p>SOA Reference Architecture is defined</p> <p>Right products are evaluated and procured to align with the reference architecture</p> <p>SOA Principles are defined</p> <p>Architectural deviations are not approved/tracked</p>	<p>Services are verified for compliance with best practices and principles</p> <p>Architectural deviations are approved and tracked.</p> <p>Architectural goals are defined</p>	<p>Architecture is tweaked to facilitate automated measurement</p> <p>Automated tools for runtime governance, BAM, etc is planned and considered as an architectural element</p> <p>Capacity planning and dynamic scalable architectures are considered to facilitate SOA adoption</p> <p>Architectural goals are measured and constantly improved</p>	<p>Robust architecture enterprise-wide where capacity is dynamically managed with cloud computing</p> <p>Architecturally, support changes to Business Processes to facilitate changing business needs</p>
Services	<p>Capabilities are tightly coupled within applications</p> <p>Different teams may need shared capabilities thereby re-building them every time.</p> <p>Services may be exposed but is highly subjective to a single project with non-standard interfaces</p> <p>No SLAs</p>	<p>Services have standard and uniform interface</p> <p>Service provider contract is defined and documented with all characteristics such as security, performance and availability</p> <p>Services are loosely coupled and designed with the right level of granularity</p> <p>Service Stubs are not used</p> <p>Services and entities are not well documented</p>	<p>Service Roadmap is defined and published using a Repository</p> <p>SLAs for services are defined and published with the service contract</p> <p>Stubs are used for service virtualization thereby mitigating dependency issues</p> <p>Services are versioned with concurrent version support and aligned to the versioning guidelines</p> <p>Services are well documented using a Repository</p>	<p>SLAs are measured using Runtime governance tools</p> <p>Service policies are enforced non-programmatically using RTG tools</p> <p>Service invocation patterns/usage patterns are measured and optimized with RTG tools</p> <p>Service quality parameters are defined and measured</p>	<p>Services are designed with right granularity to be composed in a business process</p> <p>Domain-wise grouping of services to facilitate easy composition</p> <p>Continuous improvement practices are followed and business processes eventually evolve</p>

	1. Primitive	2. Standardized Services	3. Manageable Services	4. Measurable Services	5. Agile Enterprise
Governance	<p>Governance team is non-existent</p> <p>Governance team is not empowered to make decisions</p> <p>Governance team has no control over organization's departments</p> <p>Governance team is not comprised of senior SOA architects</p>	<p>Governance team is empowered and has full support of senior management</p> <p>Senior SOA Architects within Governance team provide technical steer and assist in service identification</p> <p>Engages in review of service design to ensure compliance with architectural principles</p>	<p>Different teams within the organization recognize and realize the importance of Governance</p> <p>Governance team involves in technical decision making in frameworks and practices</p> <p>Governance team reviews and approves any architectural deviations for proven reasons</p> <p>Participates in all project initial design stages to ensure proper use of SOA</p>	<p>Governance Architects define the service measurable parameters and strategy to measure them</p> <p>Defines quality parameters and recommends service providers/consumers for incentives</p> <p>Ensures that all technical decisions /strategies are communicated to various stakeholders</p>	<p>Creates a positive influence across the organization on the overall SOA initiative</p> <p>Monitors the SOA adoption success within the organization using metrics and engages in continuous improvement</p> <p>Influences all future projects of the organization to be aligned to SOA</p>
Support and Operations	<p>Service Live Support and Operations team does not exist</p> <p>Hardware infrastructure is not monitored</p> <p>Deployments are not properly managed and is mostly done on an ad-hoc basis</p>	<p>Operations Procedures are standardized</p> <p>Service Operations Guide is documented and published</p> <p>Live Service version/build details are not tracked</p> <p>Issue resolution process is not defined</p>	<p>Service deployments are controlled using strict and disciplined processes</p> <p>Service build numbers/versions are tracked and release notes are maintained</p> <p>Service deployments are automated using scripts</p> <p>Deployment roll-back strategy is in place</p> <p>Hot-deployment strategy is in place</p>	<p>Hardware and OS resources are measured in production environments</p> <p>Service health checks and monitoring is done</p> <p>Network and legacy system connectivity from the SOA platform is monitored</p> <p>Hardware Capacity limit is monitored on a continuous basis</p>	<p>Operations team uses state-of-the-art tools for monitoring all technical resources to ensure high availability and quality of service</p> <p>A problem record is maintained for quick resolution of known problems</p> <p>SLA violations, security violations are automatically captured using RTG tools</p>

4. Conclusion

The SOA maturity assessment process helps organizations evaluate each parameter mentioned above and arrive at a specific maturity level. The assessment process is not meant to find faults with SOA implementations, but rather to measure the level of success and to identify those aspects that worked well and the ones that didn't. Most organizations start at Level 3 and work constantly towards achieving levels 4 and 5. Based on the inputs gathered during the assessment stage, existing technical and process gaps are identified. Eventually, the overall SOA strategy of the organization is revised based on the established maturity level.

Torry Harris Business Solutions Inc, a US based services provider with a large base of technologists located in the UK, India and China has provided cost effective solutions at a design, development and support level to a variety of enterprise clients across the world since 1998. The company specializes in integration, distributed computing, and its focus on SOA is a result of nearly a decade of expertise gathered in the middleware space. The company has partnerships with almost all the leading SOA and integration product vendors. SOA, involving the creation of autonomous parts of a solution, lends itself admirably to the cost effective model of offshore service collaboration. A separate white paper entitled "SOA Implementation with an offshore partner" available for download, explores this model in a more detailed manner.

For more information, write to us at soa@thbs.com

Centre for Telematics and Information Technology, University of Twente

Centre for Telematics and Information Technology, University of Twente CTIT (Centre for Telematics and Information Technology) of the University of Twente in the Netherlands is one of the largest academic ICT research institutes in Europe. Over 475 researchers actively participate in the research programme.

One of the focal areas of CTIT is Applied Science of Services for Informations Society Technologies (ASSIST), which addresses research on service architectures and software infrastructures fostering innovative, open and cost-effective solutions for health and other application areas. The CTIT affiliated authors of this article are members of the Information Systems (IS) group, participating in CTIT/ASSIST.

More information is available at <http://www.ctit.utwente.nl/sro/assist>

5. Reference

1. Blue-Crow. Business process modelling and analysis, 2007.
2. M. Broadbent and E. Kitzis. Interweaving business-driven IT strategy and execution: Four foundation factors. *Ivey Business Journal*, 69(3):1-6, 2005.
3. L. M. Camarinha-Matos and H. Afsarmanesh. The emerging discipline of collaborative networks. In *Virtual Enterprises and Collaborative Networks*, pages 3-16. Kluwer Academic Publishers, 2004.
4. Y. E. Chan. Why haven't we mastered alignment? the importance of the informal organization structure. *MIS Quarterly Executive*, 1(21):76-112, 2002.
5. R.G. Santana Tapia, M. Daneva, P.A.T. van Eck, R.J. Wieringa (2008) Towards a business-IT alignment maturity model for collaborative networked organizations. In: *EDOC 2009 Workshops - International Workshop on Enterprise Interoperability (IWEI 2008)*, IEEE Computer Society, pp. 276-287.