

## بررسی و ارزیابی متدولوژی های معماری سرویس گرا – SOMA, SOA+RUP

امیر مهجوریان – [mahjoorian@esoa.ir](mailto:mahjoorian@esoa.ir)

در این نوشته، متدولوژی های ارائه شده برای معماری سرویس گرا معرفی و ارزیابی شده و کاربرد هر کدام به تفصیل بررسی می گردد. مجموعه مطالب ارائه شده حاصل ترکیب تجربیات نگارنده و همچنین مطالعه مستندات و مقالات مربوط به این متدولوژی ها است. تعدادی از کتاب ها و وب سایتهای مفید در این زمینه در پایان مقاله معرفی شده اند

### مقدمه:

معماری سرویس گرا<sup>1</sup> (SOA) به عنوان رهیافت برتر در حوزه معماری سیستم های اطلاعاتی به سرعت در حال رشد و فراگیر شدن است، ابزارهای مبتنی بر این معماری امکانات قابل قبولی ارائه داده اند و شرکت ها و سازمانها نیز طرح های خود را بر مبنای گذار به این رهیافت تنظیم نموده اند. در میان همه نقاط امیدوار کننده تنها موضوعی که نیاز به توجه بیشتر دارد مبحث "متدولوژی" است، دلیل این امر نیز روشن است: برای داشتن یک متدولوژی کارآمد و خوش تعریف نیاز به انجام تعداد زیادی تجارب موفق است تا نتایج حاصل از این تجارب در قالب روش ها و گامهایی شکل گرفته و نهایتا متدولوژی انجام کار که شامل گامها، محصولات، راهنماییها، فرایندها، استانداردها و .. است تهیه شود و از آنجاکه استانداردهای معماری سرویس گرا و ابزارهای پیاده سازی آن از حدود سالهای ۲۰۰۱ و ۲۰۰۲ ارائه شده اند لذا هنوز متدولوژی های این رهیافت به بلوغ و تکامل نهائی (در حدی مانند RUP<sup>۲</sup>) نرسیده اند. در این نوشته متدولوژی ها و روش های ارائه شده موجود برای معماری سرویس گرا را مورد بررسی و ارزیابی قرار می دهیم و پیشنهاداتی برای سازمانهایی که قصد انجام پروژه هایی با چنین متدولوژی هایی دارند ارائه می دهیم.

### متدولوژی های موجود

با نگاهی به مقالات و کتابهای منتشر شده در حوزه معماری سرویس گرا متوجه می شویم که در بحث متدولوژی، بیشترین فعالیت و مطالب مربوط به IBM است. دو دلیل عمده این موضوع یکی سابقه این شرکت در ارائه و پشتیبانی متدولوژی معروف و بی رقیب RUP است و دلیل دوم آن پیشگامی و کیفیت برتر این شرکت در حوزه معماری سیستم های اطلاعاتی است. از آنجا که معماری سرویس گرا یک "سبک" معماری است و لذا در سازمانهای مختلف و بر حسب نیازمندیهای متفاوت می تواند به شیوه های گوناگونی طراحی و پیاده سازی شود، در نتیجه نباید انتظار داشت که تنها "یک نوع" متدولوژی مشخص برای پشتیبانی از این رهیافت ارائه شود. شرکت IBM تا کنون سه متد (روش) مختلف برای معماری سرویس گرا ارائه نموده که هر کدام قابلیت ها و شرایط خاص خود را دارند و برای نیازمندیهای مخصوص خود تهیه شده اند. این سه روش عبارتند از:

- متدولوژی RUP گسترش یافته برای معماری سرویس گرا
- معماری و مدلسازی سرویس گرا (SOMA<sup>۳</sup>)
- مدرن کردن سیستم های موروثی به کمک معماری سرویس گرا

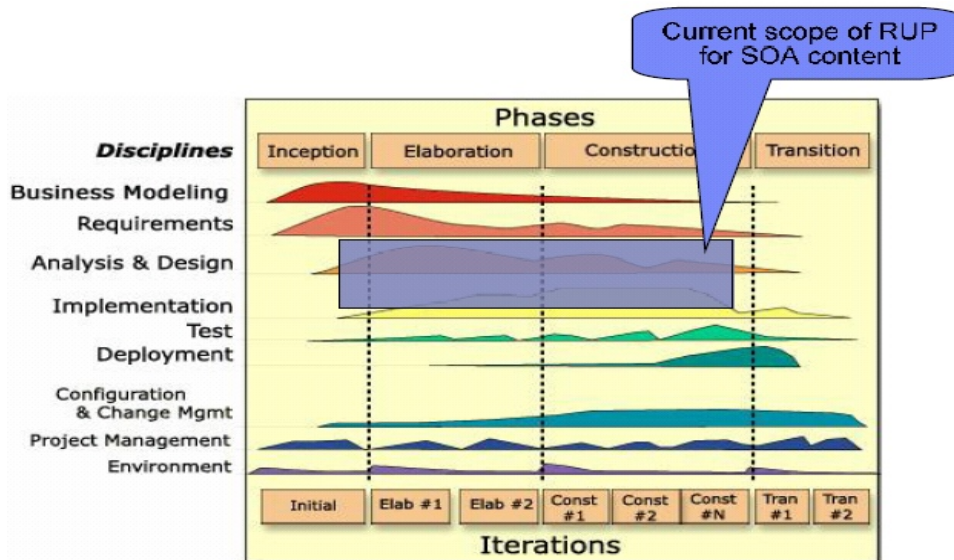
اهمیت این سه متد در انجام است که کاملترین مجموعه ارائه شده در حوزه معماری سرویس گرا می باشند و دیگر متدهای ارائه شده (MSOMA<sup>۴</sup>, OASIS<sup>۵</sup> Adoption Blueprint, Agile SOA) زیرمجموعه و مشتق شده از متدهای IBM هستند. در ادامه سه متد ارائه شده توسط IBM بررسی می شوند و کاربرد هر یک به تفصیل تشریح می گردد.

### ۱. RUP update for SOA

این متدولوژی حاصل بسط RUP برای پوشش به نیازهای معماری سرویس گرا است. جدیدترین نسخه ای از آن که تاکنون توسط IBM ارائه شده است، نسخه 2.X است و بصورت Plug-in نصب می شود. ویژگیهای جدید این نسخه به نسبت RUP پایه شامل این موارد است:

- New Activity and Workflow Details
- New Concepts
- Complete Step-by-Step Example
- New Guidelines
- Artifact Based Upon UML Profile

میزان و حوزه تاثیر گذاری معماری سرویس گرا بر متدولوژی RUP پایه در شکل زیر نشان داده شده است:

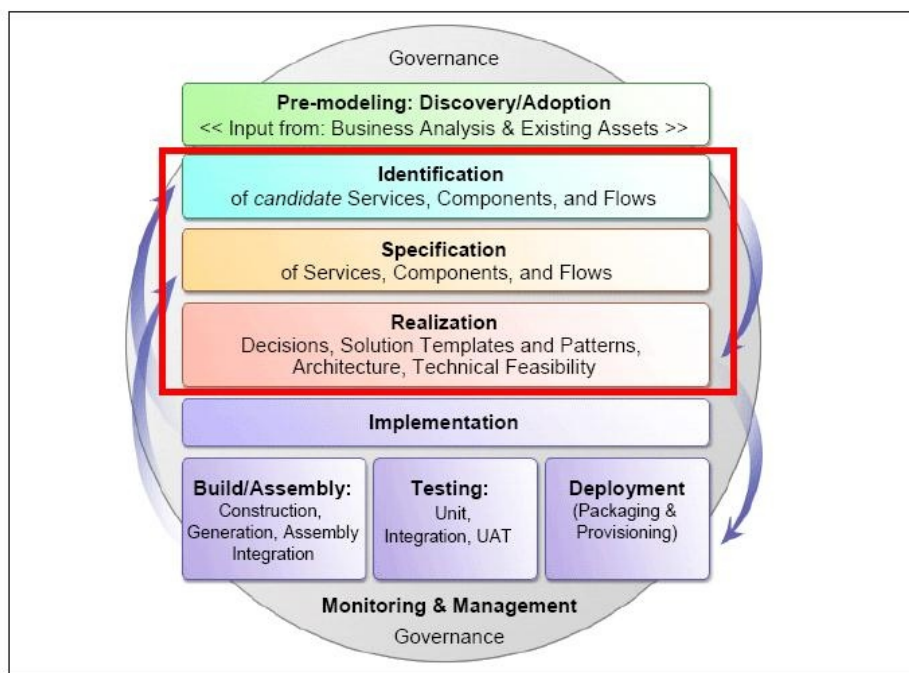


مشکل عمده این متدولوژی مربوط به ساختار RUP است که اساساً "تک پروژه محور" است، درحالیکه معماری سرویس گرا فراتر از یک پروژه نرم افزاری مجرد است و حوزه آن شامل چندین سیستم و حتی کل یک سازمان می شود. از این رو قابلیت ها و محصولات RUP برای پوشش به پروژه های گسترده و سازمانی مناسب نیست و با امکانات تحلیل و طراحی شیء گرا نیز نمی توان تمامی نیازمندیهای سازمان (فرایندها، ساختار سازمانی، نیازمندیهای غیر وظیفه ای، وظایف، قوانین حرفه، ذینفعان، تعاملات بین سازمانی و ...) را به خوبی مدلسازی و تحلیل نمود. مجموعه محصولات و فرآورده های RUP پایه برای مدلسازی کسب و کار سازمان محدود به چند مدل نه چندان قوی همچون Business Actor, Business UseCase, .. می شود که این موارد بسیار کمتر از نیازمندیهای لازم برای انجام پروژه های سازمانی می باشد. در جمع بندی می توان گفت آنچه به عنوان بسطی به RUP اضافه شده تا نیازمندیهای معماری سرویس گرا را بپوشاند اگرچه لازم و مفید ولی ناکافی است. اما برای پروژه های معماری سرویس گرائی که حوزه آن محدود و معین بوده و تعاملات بین سامانه ای نیز کم رنگ است، استفاده از این متدولوژی مناسب و مفید می باشد. IBM در تلاش است تا بزودی یکپارچگی بیشتری بین این متدولوژی با SOMA به وجود آورد و نهایتاً این دو را در هم ادغام نماید.

### ۲. Service Oriented Modeling and Architecture (SOMA)

این متدولوژی از ترکیب دو متد غیر رسمی IBM حاصل شده است. یکی متد داخلی شرکت برای معماری سیستم های اطلاعاتی است و دیگری متدها و دستورات عملی های مربوط به کاربرد UML 2.0 در مدلسازی سیستم های اطلاعاتی است. اما متاسفانه IBM همه اسناد و راهنمایی های مربوط به SOMA را بصورت عمومی منتشر

نکرده است و لذا این متدولوژی به عنوان متدولوژی داخلی IBM برای انجام پروژه های معماری سرویس گرا شناخته می شود. بیشتر اطلاعاتی که در خصوص این متدولوژی منتشر شده در غالب چند مقاله فنی و خصوصا فصلهائی از کتاب "Patterns: Service-Oriented Architecture and Web Services" و تا حدودی کتاب "Building SOA Solutions Using the Rational SDP" بوده و لذا تحلیل گرانی که می خواهند از این متدولوژی برای شرکت خود استفاده کنند با ابهام و مشکل مواجه می شوند. در یک نگاه کلان SOMA را می توان بصورت شکل زیر نشان داد.



سه فاز اصلی این متدولوژی که مربوط به شناسائی و طراحی سرویس هستند در ادامه بررسی شده اند.

### ➤ فاز شناسائی سرویس ها:

این فاز دارای دو تحلیل بالا به پائین و پائین به بالا برای شناسائی سرویس هاست. در روش بالا به پائین با بررسی حوزه های کلان کسب و کار سازمان، فرایندها شناسائی و سپس شکسته می شوند تا سرویس های ارکستریشنی و پایه استخراج گردند و سرانجام سرویس های دانه خردتر شناسائی می شوند. در سوی دیگر با تحلیل پائین به بالا نسبت به تحلیل امکانات و قابلیت های سیستم های موروثی سازمان اقدام می شود تا سرویس های اتمیک و نرم افزاری قابل استخراج از این سیستم ها شناسائی شوند. بدین ترتیب با انجام هر دو تحلیل بالا به پائین و پائین به بالا، مجموعه سرویس های سازمانی شناسائی شده و در قالب چند سطحی (سرویس های فرایندی/ میانی/ اتمیک) سازماندهی می شوند.

### ➤ فاز توصیف سرویس ها:

در این فاز ویژگیها و قابلیت های سرویس ها تشریح می گردد. عمده سئوالهایی که به آنها پرداخته می شود شامل این موارد است: سرویس چه خدمتی ارائه می دهد؟ منطق داخلی سرویس ها چیست؟ نیازمندیهای غیر وظیفه ای هر سرویس چیست؟ فرمت پیامهای ورودی/خروجی سرویس ها چگونه است؟ هر سرویس مرکب چه سرویسهای دیگری را فراخوانی می کند؟ چگونگی کاربرد سرویس ها در حوزه های دیگر چگونه است؟

### ➤ فاز عینیت بخشیدن به سرویس ها:

تمرکز این فاز بر "چگونگی" و جزئیات طراحی سرویس هاست و خروجی آن مدل تفصیلی طراحی سرویس است. جزئیات جریان داخلی سرویس ها، داده های اطلاعاتی مرتبط، و ... در اینجا مستند و مدلسازی می شوند تا در مراحل بعد پیاده سازی شوند. همچنین مباحث مربوط به "امکان سنجی فنی" پیاده سازی یا استخراج سرویس ها از سیستم های موروثی موجود در این فاز تعیین می شود.

در جمع بندی از مشخصات SOMA می توان گفت این متدولوژی برای پروژه های بزرگ مدیریت فرایندهای حرفه (BPM<sup>۲</sup>) و معماری سرویس گرای سازمانی (گسترده) مناسب بوده و کاملترین متدولوژی موجود در این حوزه می باشد اما نقطه ضعف آن عدم ارائه تمامی مستندات و پشتیبانی های فنی توسط IBM است. البته به نظر می رسد این شرکت قصد دارد به تدریج اسناد و راهنمائیهای فنی آن را در قالب یک متدولوژی کامل به بازار عرضه نموده و راهی که با RUP شروع نمود را ادامه دهد.

### ۳. مدرن کردن سیستم های موروثی به کمک معماری سرویس گرا

اگر چه این متد به گستردگی و خوش تعریفی دو مورد قبلی نیست و شاید نتوان از آن با عنوان یک "متدولوژی تحلیل و طراحی" نام برد، اما در برخی سازمانها که نیاز به مدرن نمودن سیستم های موروثی دارند می توان از این روش که در قالب مجموعه ای از مقالات و راهنمائیها توسط IBM ارائه شده استفاده نمود.

مشکلات و موانع مربوط به مدرن کردن سیستم های اطلاعاتی موروثی یکی از چالش های اصلی سازمانها در حوزه فناوری اطلاعات است. سرعت رشد و پیشرفت فناوری های جدید و همچنین تغییرات سریع بازار دو عامل اصلی هستند که سازمانها را به مدرن کردن و نوآوری در سیستم های اطلاعاتی ملزم می کنند. اما مشکل از انجا شروع می شود که بخشی از سیستم های اطلاعاتی قدیمی در مدتی که در سازمان مستقر بوده اند جزو فرهنگ و دارایی های سازمان شده و حذف یا جایگزینی آن به دلایل مختلف کار ساده ای نخواهد بود. این سیستم های قدیمی از یک طرف بخش عمده ای از منطق حرفه و داده های اطلاعاتی را در خود نگه داشته اند و از طرف دیگر کارمندان به استفاده از آنها عادت کرده و قابلیت ها و محیط کاربری این سیستم ها جزو مهارت های کارمندان شده است، لذا این سیستم ها را می توان "موروثی" نامید چراکه جزئی از اموال و دارائیهای با ارزش سازمان هستند که از گذشته

به ارث رسیده و اکنون نمی توان آنها را دور انداخت. اما همانطور که گفتیم مشکل این سیستم ها قدیمی بودن و عدم انطباق با فناوری های جدید است و این سیستم ها باید به روز شوند. این چالش اساس مباحث مربوط به مدرن نمودن سیستم های موروثی است که یکی از راه حل های موثر برای حل آن استفاده از معماری سرویس گرا برای ارائه (استخراج) کارکردهای با ارزش این سیستم ها در قالب "سرویس" است، حال این سرویس ها که بصورت استاندارد ارائه شده اند توسط دیگر سیستم های اطلاعاتی سازمان قابل استفاده اند و به تدریج اکثر کارکردهای سیستم موروثی در قالب سرویس های استاندارد استخراج می شوند تا سرانجام سیستم مدرن جدیدی که به کمک مجموعه سرویس های استخراج شده طراحی شده جای سیستم قدیمی را بگیرد. بدین ترتیب اولاً سیستم قدیمی به تدریج کنار گذاشته می شود و سیستم مدرن و مناسب جای آن را می گیرد و دوماً در مدت زمان این جایگزینی وقفه ای در کارکرد سیستم موروثی یا تعامل آن با دیگر سیستم های سازمان رخ نخواهد داد.

### نتیجه گیری:

حاصل و جمع بندی بررسی های انجام گرفته در این نوشته را می توان به سه نتیجه اصلی خلاصه نمود:  
نتیجه اول: معماری سرویس گرا نوعی "سبک" معماری است و نه یک تکنولوژی (Platform Or Technology) و بر حسب نیاز سازمانهای مختلف به صورتهای متنوعی قابل پیاده سازی است، لذا انتظار ارائه "یک نوع متدولوژی" مشخص برای آن به دور از واقعیت بوده و بر همین اساس نیز IBM حداقل سه رویکرد (روش) مختلف در این حوزه ارائه نموده است.

نتیجه دوم: از سه متد بررسی شده در این نوشته کاملترین آنها متدولوژی SOMA است که هنوز همه راهنماها/مستندات مفید آن توسط IBM منتشر نشده اما امید است به زودی این عمل محقق شود. بسط متدولوژی RUP برای پوشش به معماری سرویس گرا برای حوزه هائی که دامنه معماری سرویس گرا محدود و در حد یک پروژه متوسط است مناسب است اما برای پروژه های گسترده سازمانی اصلاً توصیه نمی شود. راهنمائیها یا شبه متدهائی که IBM برای بعضی کاربردهای خاص دیگر از جمله "مدرن نمودن سیستم های موروثی موجود" ارائه نموده نیز در حوزه خود مفید و کارگشا هستند.

نتیجه سوم: اگرچه متدولوژی های ارائه شده توسط IBM برتر و کاملتر از سایرین است اما از آنجا که متدولوژی های معماری سرویس گرا هنوز به بلوغ نهائی خود نرسیده اند، می توان گفت این متدولوژی ها به تنهایی نیاز سازمانها را برآورده نمی کنند و لازم است شرکت هائی که قصد استفاده از این متدولوژی های را دارند نسبت به تکمیل و دوخت و دوز (Customize) آن با توجه به شرایط پروژه/ نیازمندیها/ امکانات/ تخصص ها/... اقدام نمایند.

### 1 منابع و مراجع

1. Iran's Information Architecture committee: <http://esoa.ir>
2. SOA Methodology website: [www.soamethodology.com](http://www.soamethodology.com)
3. Service Oriented Architecture Weblog: [www.soea.blogfa.com](http://www.soea.blogfa.com)
4. An Evaluation of SOA Frameworks, 2007, Ramakrishna Raju.
5. Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Erl, Thomas, 2005.
6. RUP Plug-In for SOA webpage: [http://www-128.ibm.com/developerworks/rational/library/05/510\\_soaplug/](http://www-128.ibm.com/developerworks/rational/library/05/510_soaplug/)
7. Ali Arsanjani webpage; Chief Architect for the SOA in IBM Global Services:  
<http://www.ibm.com/developerworks/blogs/page/AliArsanjani>
8. Modernize legacy systems using an SOA approach, Skill Level: Advanced, Jan 2008, Jean-Louis Maréchaux.
9. BEA WebLogic Platform, SOA Platform: [www.bea.com/products/weblogic/platform/index.shtml](http://www.bea.com/products/weblogic/platform/index.shtml)
10. Microsoft BizTalk Homepage: <http://www.microsoft.com/biztalk>
11. OASIS Reference Model for Service Oriented Architecture, 2006.
12. Building SOA Solutions Using the Rational SDP (IBM), 2007, IBM Corp.
13. Case Study: SOA Design Scenario, 2008, IBM Corp.
14. Patterns: Service-Oriented Architecture and Web Services, IBM Corp.
15. Case Study: Business Process Management SOA Scenario, 2008, IBM Corp.

---

<sup>1</sup> Service Oriented Architecture

<sup>2</sup> Rational Unified Process

<sup>3</sup> Service Oriented Modeling and Architecture

<sup>4</sup> Mainstream Service Oriented Architecture Methodology

<sup>5</sup> Organization for the Advancement of Structured Information Standards

<sup>6</sup> Business Process Management