

بررسی تعامل RUP با معماری سرویس گرا امکان -

سنجی، چگونگی تحقق، فرصت‌ها و چالش‌ها

منا شریفیان^۱، عباس رسول‌زادگان^۲

^۱دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه فردوسی مشهد،

Mona.sharifian65@gmail.com

^۲استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه فردوسی مشهد،

Rasoolzadegan@um.ac.ir

چکیده

معماری سرویس‌گرا مجموعه‌ای از سرویس‌های توزیع‌شده با حداقل وابستگی می‌باشد که به منظور رفع نیازهای کسب و کار باهم بر روی فعالیت‌های تعیین شده‌ای در تعامل هستند. یکی از چالش‌های تحقیقاتی در حوزه‌ی معماری سرویس‌گرا، برخورداری از یک فرآیند تولید نرم‌افزار است که شامل کلیه ملاحظات و اصول این معماری می‌باشد تا بستر لازم را برای تولید سیستم‌های سرویس‌گرا با کیفیت مطلوب فراهم نماید. RUP چارچوبی برای تعریف فرآیند نرم‌افزار است. سفارشی‌سازی RUP بر مبنای نیازمندی‌ها، محدودیت‌ها و اهداف یک پروژه نرم‌افزاری منتج به تعریف فرآیند مناسب برای تولید نرم‌افزار مذکور می‌گردد. با توجه به محبوبیت و کاربرد گسترده RUP در تعریف فرآیند مناسب و کارا برای طیف وسیعی از سیستم‌های نرم‌افزاری، بررسی تعامل RUP با معماری سرویس‌گرا، می‌تواند راهکاری در جهت رفع چالش پیش‌گفته به شمار آید. در این مقاله، ابتدا به بررسی اجمالی RUP و معماری سرویس‌گرا می‌پردازیم. همزمان معایب و مزایای هر یک از آن‌ها را مورد تحلیل قرار می‌دهیم. در ادامه اهداف فازهای مختلف RUP را با اصول و ملاحظات معماری سرویس‌گرا مقایسه می‌نماییم و بر مبنای آن به سنجش و نحوه تعامل RUP با معماری سرویس‌گرا می‌پردازیم. در نهایت مزایا و محدودیت‌های تعامل مذکور را مورد بررسی قرار می‌دهیم.

واژه‌های کلیدی: معماری سرویس‌گرا، فرآیند تولید نرم‌افزار، کیفیت، RUP



بسمه تعالی



بدینوسیله گواهی میشود

مقاله با عنوان:

بررسی تعامل RUP با معماری سرویس گرا امکان سنجی،

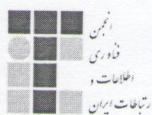
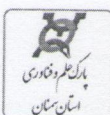
چگونگی تحقق، فرصت‌ها و چالش‌ها

منا شریفیان، عباس رسولزادگان

در ششمین کنفرانس فناوری اطلاعات و دانش ارائه شده است.

دبیر کنفرانس

دکتر حیدر طوسی‌ان شاندیز



بررسی تعامل RUP با معماری سرویس‌گرا: امکان‌سنجی، چگونگی تحقق، فرصت‌ها و چالش‌ها

منا شریفیان^۱، عباس رسول‌زادگان^۲

^۱ دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه فردوسی مشهد، Mona.sharifian65@gmail.com

^۲ استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه فردوسی مشهد، Rasoolzadegan@um.ac.ir

چکیده

معماری سرویس‌گرا مجموعه‌ای از سرویس‌های توزیع‌شده با حداقل وابستگی می‌باشد که به منظور رفع نیازهای کسب و کار باهم بر روی فعالیت‌های تعیین شده‌ای در تعامل هستند. یکی از چالش‌های تحقیقاتی در حوزه‌ی معماری سرویس‌گرا، برخورداری از یک فرآیند تولید نرم افزار است که شامل کلیه ملاحظات و اصول این معماری می‌باشد تا بستر لازم را برای تولید سیستم‌های سرویس‌گرا با کیفیت مطلوب فراهم نماید. RUP چارچوبی برای تعریف فرآیند نرم‌افزار است. سفارشی‌سازی RUP بر مبنای نیازمندی‌ها، محدودیت‌ها و اهداف یک پروژه نرم‌افزاری منتج به تعریف فرآیند مناسب برای تولید نرم‌افزار مذکور می‌گردد. با توجه به محبوبیت و کاربرد گسترده RUP در تعریف فرآیند مناسب و کارا برای طیف وسیعی از سیستم‌های نرم‌افزاری، بررسی تعامل RUP با معماری سرویس‌گرا، می‌تواند راهکاری در جهت رفع چالش پیش‌گفته به شمار آید. در این مقاله، ابتدا به بررسی اجمالی RUP و معماری سرویس‌گرا می‌پردازیم. همزمان معایب و مزایای هر یک از آن‌ها را مورد تحلیل قرار می‌دهیم. در ادامه اهداف فازهای مختلف RUP را با اصول و ملاحظات معماری سرویس‌گرا مقایسه می‌نماییم و بر مبنای آن به سنجش و نحوه تعامل RUP با معماری سرویس‌گرا می‌پردازیم. در نهایت مزایا و محدودیت‌های تعامل مذکور را مورد بررسی قرار می‌دهیم.

واژه‌های کلیدی

معماری سرویس‌گرا، فرآیند تولید نرم‌افزار، کیفیت، RUP

۱- مقدمه

پیاده‌سازی معماری سرویس‌گرا ارائه گردیده است، اما هیچ یک به طور کامل، ملاحظات و اصول این معماری را پوشش نمی‌دهد.

هدف این مقاله امکان‌سنجی و تحلیل نحوه تعامل RUP و معماری سرویس‌گرا است. دلیل انتخاب RUP برای پوشش ملاحظات و اصول معماری سرویس‌گرا، محبوبیت و کاربرد گسترده‌ی آن در حوزه دانشگاه و صنعت برای تعریف فرآیندهای تولید مناسب برای طیف گسترده‌ای از سیستم‌های نرم‌افزاری می‌باشد. سایر متدولوژی‌های موجود دارای ناسازگاری‌های متعددی با معماری سرویس‌گرا هستند که برخی از آن‌ها عبارتند از: (۱) معماری سرویس‌گرا بر مبنای ارتباطات ارزش‌افزین است، در حالیکه متدولوژی‌های مذکور بر ارتباطات هزینه بر استوار هستند، (۲) معماری سرویس‌گرا مبتنی بر فرآیند است، در حالیکه متدولوژی‌های مذکور مبتنی بر کارکرد هستند، (۳) معماری سرویس‌گرا برای تغییر ساخته می‌شود، در حالیکه متدولوژی‌های مذکور برای بقاء می‌سازند، (۴) معماری سرویس‌گرا بر خلاف متدولوژی‌های مذکور بر مبنای توسعه تدریجی است [۵]. برای نیل به این مقصود، در این مقاله، پس از کسب

معماری سرویس‌گرا^۱ به دلیل مزایایی نظیر تقویت محیط کسب و کار با استفاده از زیرساخت انعطاف‌پذیر، افزایش فرآیندهای کسب و کار خودکار، توانایی استفاده‌ی مجدد از سرویس‌های مستقل و ساخت یک بستر اطلاعاتی جامع برای سرویس‌های سازمان، مورد توجه قرار گرفته است [۱]. یکی از چالش‌های تحقیقاتی در معماری سرویس‌گرا یافتن یک فرآیند کارای تولید نرم‌افزار است که تمامی ملاحظات و اصول این معماری را در نظر بگیرد تا بستر لازم را برای تولید برنامه‌های کاربردی مؤثر و با کیفیت فراهم سازد. از آن‌جا که معماری سرویس‌گرا یک سبک معماری است، سازمان‌های مختلف می‌توانند براساس نیازهای متفاوت خود، آن را طراحی و پیاده‌سازی نمایند و به همین جهت انتظار نمی‌رود که تنها یک فرآیند خاص برای پشتیبانی این سبک معماری ارائه شود [۲]. امروزه متدولوژی‌های مختلفی توسط محققان و شرکت‌ها برای

^۱ Service Oriented Architecture (SOA)

آنچه از اطلاعات ارائه شده در جدول ۱ می‌توان نتیجه گرفت آن است که اجتناب و گذر از محدودیت‌های معماری سرویس‌گرا نیازمند یک متدولوژی نرم‌افزاری قاعده‌مند و جامع می‌باشد. استفاده از یک فرآیند توسعه‌ی مناسب می‌تواند احتمال موفقیت معماری سرویس‌گرا را از طریق ایجاد ارتباط دقیق بین افراد تیم توسعه، تعریف مکانیزم شفاف و قاعده‌مند کنترل و مدیریت تغییرات و ارائه سیاست‌های پشتیبانی و مهندسی نیازمندی‌ها افزایش دهد [۱]، [۴]، [۵]، [۷].

همانطور که در ادامه نیز به آن پرداخته خواهد شد، RUP با فراهم سازی امکان تعریف فرآیندهای توسعه‌ی مناسب برای طیف وسیعی از سیستم‌های نرم‌افزاری به صورت بالقوه دارای قابلیت پوشش کمبودهای معماری سرویس‌گرا است.

۲-۲ RUP

RUP، از یک منظر، رویکردی است برای تولید نرم‌افزار که ویژگی‌های برجسته‌ای نظیر تکرار شونده‌گی، تمرکز بر معماری و توسعه مبتنی بر موارد کاربرد را در بر می‌گیرد.

از منظر دیگر، RUP یک فرآیند خوش‌تعریف و سازمان‌یافته در مهندسی نرم‌افزار می‌باشد که اجزاء اصلی آن شامل نقش‌های تعریف شده، فعالیت‌ها، دستاوردها^۲، دیسپلین‌ها^۳ و جریان‌های کاری^۴ می‌باشد و ساختار مناسبی را برای در کنار هم قرار دادن این اجزاء در اختیار قرار می‌دهد.

همچنین RUP را می‌توان یک محصول مهندسی قلمداد کرد که دربرگیرنده چارچوب و قالب کلی برای تعریف فرآیندهای تولید سیستم‌های نرم‌افزاری است [۸]، [۹]، [۲]. در واقع با سفارشی‌سازی RUP بر اساس نیازمندی، محدودیت، ملاحظات و اهداف یک پروژه، فرآیند توسعه مناسب آن پروژه حاصل می‌گردد.

RUP گنجینه‌ای ارزشمند از دانش‌ها، راه‌حل‌ها، راه‌کارها و تجارب موفق مهندسی قابل تکرار را در اختیار ما قرار می‌دهد. بنابراین توصیه‌های RUP برای گستره وسیعی از پروژه‌ها، از کوچک تا بزرگ مفید است. هر پروژه با توجه به نیازها، محدودیت‌ها و ملاحظات خود می‌تواند RUP را سفارشی نماید. برخی از مزایا و محدودیت‌های RUP در جدول ۲ به صورت خلاصه ارائه شده است. شایان ذکر است که در این جدول مزایای RUP با تمرکز ویژه بر شناسایی نقاط بالقوه تعامل RUP با معماری سرویس‌گرا ارائه شده است [۱۰]، [۸]. در ادامه به بررسی قابلیت‌های RUP در جهت برآوردن نیازمندی‌های معماری سرویس‌گرا در قالب جدول ۳ می‌پردازیم.

شناخت کافی از RUP و معماری سرویس‌گرا و تحلیل نقاط ضعف و قوت هر یک به مقایسه اهداف و وظایف فازهای RUP با اصول و ملاحظات معماری سرویس‌گرا می‌پردازیم. نتایج حاصل از مقایسه مذکور مبنای سنجش امکان و نحوه تعامل RUP با معماری سرویس‌گرا قرار می‌گیرد. بررسی مزایا و محدودیت‌های این تعامل آخرین قدم از پژوهش جاری را تشکیل می‌دهد.

این مقاله در چهار بخش سازماندهی شده است: در بخش دوم به معرفی اجمالی معماری سرویس‌گرا و RUP می‌پردازیم. در این بخش همچنین مزایا و محدودیت‌های هر یک نیز مورد بررسی قرار می‌گیرد. در بخش سوم به امکان‌سنجی و تحلیل چگونگی تعامل RUP با معماری سرویس‌گرا پرداخته می‌شود. همچنین، مزایا و محدودیت‌های چنین تعاملی در این بخش مورد بررسی قرار می‌گیرد. در نهایت در بخش چهارم به نتیجه‌گیری و معرفی کارهای آینده می‌پردازیم.

۲- معرفی و تحلیل اجمالی معماری سرویس‌گرا و RUP

در این بخش به اختصار به معرفی معماری سرویس‌گرا و RUP می‌پردازیم. علاوه، مزایا و محدودیت‌های هر یک را نیز تحلیل می‌نماییم.

۲-۱ معماری سرویس‌گرا

معماری سرویس‌گرا رهیافتی شامل ساخت سیستم‌های توزیع شده که کارکردهای نرم‌افزاری را در قالب مجموعه‌ای از سرویس‌های توزیع شده در اندازه‌های مورد نیاز سازمان ارائه می‌نماید. این سرویس‌ها با کمک تعریف یک واسط استاندارد مستقل از سکو از پیاده‌سازی مجزا شده‌اند [۱]. معماری سرویس‌گرا، سبکی از معماری سیستم‌های اطلاعاتی است که هدف آن دستیابی به اتصال سست در ارتباط مؤلفه‌های نرم‌افزاری با یکدیگر است. اتصال سست بین مؤلفه‌های نرم‌افزاری باعث افزایش قابلیت استفاده مجدد آن‌ها می‌شود [۳]. معماری سرویس‌گرا شامل مزایا و معایبی می‌باشد که بطور خلاصه در جدول ۱ ارائه شده است.

جدول ۱ - مزایا و معایب معماری سرویس‌گرا

مزایا	معایب
- قابلیت استفاده مجدد از سرویس	- پیچیدگی که موجب افزایش ریسک شکست می‌شود
- قابلیت ترکیب‌پذیری سرویس	- تحمل‌پذیری اشکال کم با وجود هزینه بالای پیاده‌سازی
- خود مختاری سرویس‌ها	- عدم توانایی در کنترل و مدیریت نیازمندی‌های کسب و کار پویا
- پنهان‌سازی پیاده‌سازی داخلی	
- قابلیت شناسایی و کشف	
- استاندارد و شفاف‌بودن ورودی و خروجی	

² Achievements

³ Discipline

⁴ Workflow

جدول ۲- مزایا و معایب RUP

محدودیت‌ها	مزایا (نقاط بالقوه برای تعامل با معماری سرویس‌گرا)
۱- RUP نتوانسته خود را با گسترش روز افزون رهیافت‌های توسعه و تولید نرم افزار وفق دهد و عموماً تابع تغییرات بوده تا مبداء تغییرات.	۱- RUP معماری محور است و از معماری شیء‌گرا استفاده می‌کند.
۲- به دلیل وابستگی بسیار زیاد RUP به موارد کاربرد، در صورت وجود هرگونه نقص در موارد کاربرد این نقص به سایر بخش‌های پروژه نیز سرایت می‌نماید.	۲- RUP مبتنی بر موارد کاربرد است که می‌توان آن‌ها را به عنوان سرویس‌هایی در نظر گرفت که کاربر علاقه‌مند است سیستم نرم‌افزاری به او ارائه کند.
۳- RUP وابستگی زیادی به UML دارد و استفاده از UML در RUP بعضاً با پیچیدگی بسیاری مواجه است و این پیچیدگی با بزرگتر شدن ابعاد پروژه بیشتر نیز می‌شود.	۳- توسعه در RUP بر مبنای مدل تکرار و تکامل است که محدودیت‌های سایر مدل‌های فرایند را بر طرف می‌نماید.
۴- به نظر می‌رسد RUP به بعضی نمودارها نیازمند است که هنوز ارائه نشده‌اند.	۴- ویژگی برجسته RUP نحوه‌ی سازماندهی گام‌های فرآیند به دو صورت پویا و ایستا می‌باشد.
	۵- جامعیت و کاربرد گسترده‌ی RUP که امکان فرآیند تولید مناسب را برای طیف گسترده‌ای از تیم‌های نرم‌افزاری فراهم می‌نماید.

بنابراین با توجه به ماهیت ملاحظات و نیازمندی‌های معماری سرویس‌گرا و شرح وظایف هر یک از فازهای چهارگانه RUP (آغازین، تشریح (معماری)، ساخت و انتقال) [۱۱]، [۱۲]، برای پوشش ملاحظات معماری سرویس‌گرا معمولاً کافی است فازهای اول و دو RUP توسعه یابند.

آن چه از این مقایسه بر می‌آید آن است که RUP در حال حاضر می‌تواند با عنوان چارچوبی برای تعریف فرآیندهای مختلف تولید نرم‌افزار، بستر مناسبی را برای توسعه معماری سرویس‌گرا فراهم آورد.

جدول ۳- نگاشت قابلیت‌های RUP با نیازمندی‌های معماری سرویس‌گرا

نیازمندی‌های معماری سرویس‌گرا	قابلیت‌های RUP
تیم‌های توسعه بزرگ با تخصص‌های چندگانه	مدیریت کارا و قاعده‌مند و سازماندهی تیم‌های توسعه بزرگ
مستندسازی بهینه و قاعده‌مند نیازمندی‌ها	شامل دیسیپلینی به منظور مهندسی نیازمندی‌ها
فریند توسعه طولانی، مخصوصاً در زمان طراحی معماری	فراهم‌سازی امکان تعریف فرآیند تولید مفصل و جامع
استفاده در سازمان‌های بزرگ	عدم وجود محدودیت در اندازه سازمان مقصد
تعریف کارکردها در قالب سرویس	طرح‌ریزی بر مبنای اصول شیء‌گرایی و مفهوم شیء که یاده‌سازی مفهوم سرویس را میسر می‌سازد

ملاحظات و اصول معماری سرویس‌گرا در قالب جدول ۴ می‌پردازیم.

بنابراین در ادامه به توسعه هر یک از دو فاز مذکور و همچنین به تعریف خروجی‌های جدید هر یک از آن‌ها به منظور پوشش

جدول ۴- توسعه فازهای آغازین و معماری RUP بر مبنای معماری سرویس‌گرا

فاز	توسعه هر فاز بر مبنای معماری سرویس‌گرا	خروجی‌های جدید هر فاز بر مبنای معماری سرویس‌گرا
۱- آغازین	اعضاء تیم متوجه مفاهیم معماری سرویس‌گرا خواهند شد، در این فاز با استفاده از مدل بلوغ معماری سرویس‌گرا سطح بلوغ و جامعیت سازمان سنجیده شده و عوامل هدایت پروژه به سمت معماری سرویس‌گرا تعیین می‌گردد.	<ol style="list-style-type: none"> چشم‌انداز و اهداف معماری سرویس‌گرا استراتژی معماری سرویس‌گرا تجزیه و تحلیل بازگشت سرمایه تشریح چگونگی ارتباط تیم معماری سرویس‌گرا با سایر اعضا و کاربران پروژه موارد کاربرد و تجزیه و تحلیل آن‌ها نرم‌افزارها و سخت‌افزارهای مورد نیاز برای طراحی معماری سرویس‌گرا
۲- تشریح (معماری)	<ol style="list-style-type: none"> مشارکت تیم معماری سرویس‌گرا با سایر اعضا پروژه جمع‌آوری، تعریف و تحلیل نیازمندی‌ها طرح‌ریزی و زمان‌بندی پروژه تعریف زیرساخت فنی تعریف و تکمیل موارد کاربرد با توجه ویژه به مفهوم سرویس تعریف و مستندسازی معماری سرویس‌گرا اندازه‌گیری کمی کیفیت سرویس‌های مورد نیاز تحويل معماری سرویس‌گرا به منظور استفاده در مرحله تولید پروژه 	<ol style="list-style-type: none"> سند نیازمندی‌های عملیاتی شامل همه فرآیندهای کسب و کار در قالب سرویس‌ها سند نیازمندی‌های غیرعملیاتی (کیفی) شامل اندازه کمی کیفیات مورد انتظار از سرویس‌ها و توافق‌نامه سطح خدمات و نیازمندی‌های زیر ساخت سند عملی معماری سرویس‌گرا سند تعریف زیر ساخت سند طراحی معماری سرویس‌گرا با جزئیات برنامه تست و تضمین کیفیت سرویس‌ها

۳- تحلیل مزایا و کمبودهای تعامل RUP با معماری سرویس‌گرا

وجود نقاط مثبت و منفی در هر تعامل، امری غیرقابل‌انکار می‌باشد. تعامل RUP و معماری سرویس‌گرا نیز از این مقوله مستثنا نیست. از جمله ویژگی‌های RUP در تعامل با معماری سرویس‌گرا که باعث ایجاد بستری مناسب در جهت توسعه این معماری می‌گردد، عبارتند از:

- امکان گنجاندن ملاحظات معماری سرویس‌گرا در فازهای آغازین و معماری با تمرکز ویژه بر دسیپلین‌های تحلیل و طراحی سرویس‌ها
- امکان سفارشی‌سازی فعالیت‌های مختلف تعریف شده در RUP بر مبنای معماری سرویس‌گرا

اما باید توجه کرد که RUP در تعامل با معماری سرویس‌گرا با مشکلاتی نیز مواجه است:

(۱) همانطور که پیش از این نیز بیان شد، RUP بر مبنای شیء‌گرایی است. اگرچه امکانات و مفاهیم موجود در شیء‌گرایی مدل‌سازی و پیاده‌سازی سرویس را میسر می‌سازد، اما قادر به پوشش کلیه جنبه‌ها، ملاحظات و قیود سرویس‌گرایی نیست؛ به عنوان مثال تجمع سرویس‌ها برای ایجاد سرویس ترکیبی که خود می‌تواند مانند یک سرویس عمل کند.

(۲) فقدان چابکی در مقایسه با دیگر متدولوژی‌ها؛ این متدولوژی در هر تکرار، نیازمند کار بیشتری نسبت به دیگر متدولوژی‌ها می‌باشد. هر چند این موضوع یک مشکل اساسی محسوب نمی‌شود، اما هنگامی که سریعاً نیاز به یک برنامه کاربردی مبتنی بر معماری سرویس‌گرا است، یک مدل فرآیند چابک‌تر مورد نیاز است [۶]، [۱۳].

مزایا و محدودیت‌های RUP در تعامل با معماری سرویس‌گرا در جدول ۵ سازماندهی شده است.

جدول ۵- مزایا و محدودیت‌های RUP در تعامل با معماری سرویس‌گرا و تغییرات مورد نیاز برای دستیابی به سازگاری بیشتر

محدودیت‌ها	مزایا
عدم پوشش کلیه جنبه‌های سرویس‌گرایی توسط شیء‌گرایی	قابل استفاده در پروژه‌های بزرگ
عدم برخورداری از چابکی	برخورداری از ساختار خوش‌تعریف
	امکان توسعه فازها و یسپیلین‌ها بر مبنای سرویس‌گرایی

۴- نتیجه‌گیری و کارهای آینده:

در سال‌های اخیر، معماری سرویس‌گرا بیش از پیش مورد توجه قرار گرفته است، بنابراین، نیاز به یک فرآیند تولید نرم‌افزار که دربردارنده کلیه ملاحظات و اصول این معماری باشد، ضروری به نظر می‌رسد. تاکنون متدولوژی‌های متعددی برای تعریف فرآیند نرم‌افزار ارائه شده است. ارزیابی‌ها حاکی از ناتوانی متدولوژی‌های موجود در پوشش معماری سرویس‌گرا است. شایان ذکر است که در حال حاضر متدولوژی خاصی برای معماری سرویس‌گرا وجود ندارد. از بین متدولوژی‌های مذکور تنها برای برای تعامل با معماری سرویس‌گرا مساعد به نظر می‌رسد.

در این مقاله با تحلیل مزایا و نقاط قوت RUP با توجه به نیازمندی‌های معماری سرویس‌گرا به شناسایی نقطه بالقوه RUP برای تعامل با سرویس‌گرایی پرداختیم. نتایج حاکی از آن است که اگرچه ملاحظات و اصول RUP کاملاً منطبق بر معماری سرویس‌گرا نمی‌باشد، اما بیشتر نیازمندی‌های معماری سرویس‌گرا را پوشش می‌دهد. در این مقاله، همچنین، به محدودیت‌های موجود در تعامل RUP با معماری سرویس‌گرا پرداخته شد؛ این محدودیت‌ها عبارتند از: طرح‌ریزی RUP بر مبنای شیء‌گرایی و عدم توانایی شیء‌گرایی در پوشش جنبه‌های مختلف سرویس‌گرایی و همچنین عدم چابکی RUP.

با توجه به اینکه RUP در حال حاضر مناسب‌ترین متدولوژی برای پوشش نیازمندی‌های معماری سرویس‌گرا به شمار می‌آید، لذا بر طرف کردن محدودیت‌های فوق در تعامل RUP و سرویس‌گرایی کاری ضروری و ارزشمند برای آینده تلقی می‌گردد.

منابع:

- [۸] رسول‌زادگان، عباس، مدل‌سازی نرم‌افزار به کمک UML با رویکرد RUP، بابل، علوم رایانه، زمستان ۱۳۹۱.
- [۱۰] برآنی، احمد، مرجع کاربردی متدولوژی RUP برای تولید و توسعه سیستم‌های نرم‌افزاری، انتشارات دانشگاه اصفهان، پاییز ۱۳۸۷.
- [1] Erl T., "Service-Oriented Architecture: Concepts Technology and Design," Prentice Hall, 2005.
- [2] N. Dastranj Mamaghani, F. Mousavi, F. Hakamizadeh and M. Sadeghi, "Proposed Combined Framework of SOA and RUP," 3rd International Conference on Information Sciences and Interaction Sciences (ICIS), 2010.
- [3] Sprott, D., Wilkes, L., Understanding SOA, CBDI Forum, 2003
- [4] Khoshafian, S., Service Oriented Enterprises, Auerbach, 2008
- [5] Knipple, R., Service Oriented Enterprises Architecture, MS Thesis, IT-University of Copenhagen, 2005.
- [6] Guillermo A. Callahan, "Suitability of Extreme Programming and RUP Software Development Methodologies for SOA Applications," Seminar on Enterprise Information Systems, 2006.
- [7] Arsanjani, A., "Service-Oriented Modeling and Architecture (SOMA)," IBM® developerWorks®, 2004
- [9] S. Hussain, Sheikh Muhammad Saqib, "Mapping of SOA and RUP: DOA as Case Study," Journal Of Computing, Volume 2, Issue 1, January 2010, ISSN 2151-9617.
- [11] Jim Msden, "Modeling SOA: Part 3. Service Realization," 2007.
- [12] Zhang, Liang-Jie, "SOA Solution Reference Architecture," 2007.
- [13] FilippoSantas, "Mitigating Service Orientation Risks with RUP," Service Technology Magazine Issue LIV September 2011.