



دانشگاه پیام نور  
دانشکده فنی و مهندسی

پایان نامه

برای دریافت درجه کارشناسی ارشد  
رشته مهندسی کامپیوتر - گرایش نرم افزار  
گروه فناوری اطلاعات و ارتباطات

## **ارائه چارچوب مناسب برای ایجاد نرم افزارهای جامع دانشگاهی بر مبنای معماری سرویس گرا**

مرضیه عمادی

استاد راهنما:

**دکتر محمد داورپناه جزی**

استاد مشاور:

**دکتر رضا عسکری مقدم**

آبان ۱۳۹۱



دانشگاه پیام نور  
دانشکده فنی و مهندسی  
دانشگاه پیام نور مرکز شمیرانات

پایان نامه  
برای دریافت درجه کارشناسی ارشد  
رشته مهندسی کامپیوتر - گرایش نرم افزار  
گروه فناوری اطلاعات و ارتباطات

## **ارائه چارچوب مناسب برای ایجاد نرم افزارهای جامع دانشگاهی بر مبنای معماری سرویس گرا**

**مرضیه عمادی**

استاد راهنما:

**دکتر محمد داورپناه جزی**

استاد مشاور:

**دکتر رضا عسکری مقدم**

آبان ۱۳۹۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



تقدیم به:

پدر، مادر و همسر عزیزم

## تشکر و قدردانی

خداوند مهربان را سپاس که به من نعمت زیستن، تفکر، هدایت، آموختن و شکر کردن را عطا نمود. صمیمانه‌ترین سپاسگزاری‌ها را از راهنمایی‌ها و نظرات جناب آقای دکتر محمد داورپناه جزی داشته که در تمام مراحل انجام این امر بنده را یاری نمودند و همچنین از زحمات جناب آقای دکتر رضا عسکری مقدم سپاسگزارم. ضمناً لازم است از کارشناسان محترم در شرکت‌های سامانه‌ساز مروارید، مدارگسترش فناوری اطلاعات، نوپرداز اصفهان، داده‌پردازان معاصر و سما سامانه به دلیل در اختیار گذاشتن منابع شرکت بسیار سپاسگزاری نمایم. در پایان از کسانی که هرگز پرسش‌های بی‌شمارم را در طول انجام این تحقیق بی‌پاسخ نگذاشتند قدردانی می‌نمایم.

## چکیده

امروزه به دلیل پیچیده‌تر شدن سیستم‌های نرم‌افزاری مورد استفاده در سازمان‌ها، روش‌های قدیمی تولید نرم‌افزار دیگر پاسخگو نخواهد بود به همین جهت از معماری نرم‌افزار که باعث مشخص شدن نیازهای سیستم و همچنین واضح و معنی‌دار نمودن روابط بین اجزاء آن خواهد شد استفاده می‌گردد. معماری نرم‌افزار دارای سبک‌های مختلفی است که براساس نیاز سازمان مورد نظر، سبک مناسب انتخاب می‌شود. با توجه به ویژگی‌هایی از قبیل ایجاد یکپارچگی در زیرسیستم‌ها، استقلال از سکوها‌های نرم‌افزاری و سخت‌افزاری، بهبود قابلیت استفاده مجدد و انعطاف‌پذیری اجزاء، بهبود تعامل‌پذیری و خاصیت اتصال سست سرویس‌ها در معماری سرویس‌گرا این سبک بیشتر از سبک‌های دیگر مورد توجه قرار گرفته است. روش‌های متفاوتی از این نوع معماری توسط افراد و شرکت‌های مختلف ابداع شده است. در این تحقیق چارچوبی برای طراحی سیستم‌های جامع دانشگاهی براساس مزایا و معایب برخی از روش‌های معماری سرویس‌گرا ارائه شده است و سعی گردیده معایب روش‌های دیگر را پوشش دهد. این چارچوب برای قسمت پذیرش و ثبت نام دانشجویان جدیدالورود دانشگاه پیام نور به عنوان مطالعه موردی تحلیل شده و نشان می‌دهد که به تشخیص و تحلیل نیازمندی‌ها، استفاده مجدد از سرویس‌ها و نگهداری از سیستم پس از تولید، تاکید داشته است.

**واژه‌های کلیدی:** معماری نرم‌افزار، معماری سرویس‌گرا، روش‌ها، چارچوب، سیستم‌های جامع دانشگاهی.

## فهرست مطالب

صفحه	عنوان
۱	فصل ۱ مقدمه
۲	۱-۱ مقدمه.....
۳	۲-۱ تعریف مساله و سؤالات اصلی تحقیق.....
۳	۳-۱ سابقه و ضرورت انجام تحقیق.....
۵	۴-۱ فرضیه‌ها.....
۶	۵-۱ اهداف تحقیق.....
۶	۶-۱ روش تحقیق.....
۶	۷-۱ مراحل انجام تحقیق.....
۷	۸-۱ ساختار پایان‌نامه.....
۹	فصل ۲ معماری نرم افزار و سبک‌های آن
۱۰	۱-۲ مقدمه.....
۱۰	۲-۲ تاریخچه معماری نرم‌افزار.....
۱۱	۳-۲ آشنایی با معماری نرم‌افزار.....
۱۴	۴-۲ تعریف سبک معماری نرم‌افزار.....
۱۴	۵-۲ دسته‌بندی سبک‌ها در سال ۱۹۹۴.....
۱۵	۱-۵-۲ سبک‌های مبتنی بر فراخوانی.....
۱۹	۲-۵-۲ سبک‌های متمرکز روی داده.....
۲۱	۳-۵-۲ سبک‌های اجزای مستقل.....
۲۴	۴-۵-۲ سبک جریان داده.....
۲۷	۵-۵-۲ ماشین مجازی.....
۲۹	۶-۲ دسته‌بندی سبک‌ها در سال‌های ۱۹۹۶ تا ۲۰۰۳.....
۳۷	۷-۲ معرفی سبک‌های جدید.....
۳۸	۱-۷-۲ سبک مبتنی بر مولفه.....
۳۹	۲-۷-۲ سیستم برنامه‌ریزی منابع سازمان.....
۴۰	۳-۷-۲ سبک مبتنی بر عامل.....
۴۱	۴-۷-۲ سبک مبتنی بر عامل‌های هوشمند.....
۴۴	۵-۷-۲ معماری سرویس‌گرا.....



۴۶	۶-۷-۲ سبک مبتنی بر استفاده مجدد.....
۴۷	۷-۷-۲ سبک جنبه‌گرا.....
۴۹	۸-۲ تعریف ویژگی‌های سبک‌ها و مقایسه براساس آنها.....
۵۴	۹-۲ انتخاب سبک مناسب برای سیستم‌های جامع دانشگاهی.....
۵۶	۱۰-۲ جمع‌بندی و نتیجه‌گیری.....

### ۵۷ فصل ۳ معماری سرویس‌گرا و روش‌های آن

۵۸	۱-۳ مقدمه.....
۵۸	۲-۳ تعریف معماری سرویس‌گرا.....
۵۹	۳-۳ علت استفاده از معماری سرویس‌گرا.....
۶۰	۴-۳ مقایسه سرویس‌گرایی و مولفه‌گرایی.....
۶۰	۵-۳ تعریف سرویس.....
۶۱	۶-۳ تعریف وب سرویس‌ها.....
۶۲	۷-۳ پروتکل‌ها و استانداردها.....
۶۳	۸-۳ اتصال سست.....
۶۳	۹-۳ هم‌نوا سازی و هم‌خوانی.....
۶۴	۱۰-۳ مزایای معماری سرویس‌گرا.....
۶۶	۱۱-۳ روش‌های معماری سرویس‌گرا.....
۶۶	۱-۱۱-۳ تحلیل و طراحی سرویس‌گرا (SOAD).....
۶۹	۲-۱۱-۳ معماری و مدلسازی سرویس‌گرا (SOMA).....
۷۵	۳-۱۱-۳ کیفیت تکرارپذیر SOA (RQ).....
۷۶	۴-۱۱-۳ روش CBDI-SAE.....
۷۸	۵-۱۱-۳ چارچوب معماری سرویس‌گرا (SOAF).....
۸۰	۶-۱۱-۳ فرایند یکنواخت سرویس‌گرا (SOUP).....
۸۶	۷-۱۱-۳ روش پایازگلو.....
۸۸	۸-۱۱-۳ روش استیون جونز.....
۸۹	۹-۱۱-۳ روش RUP for SOA.....
۹۱	۱۰-۱۱-۳ روش MSOAM.....
۹۸	۱۲-۳ خصوصیات روش‌ها و مقایسه بر مبنای معیارهای مشخص.....
۱۰۴	۱۳-۳ جمع‌بندی و نتیجه‌گیری.....

### ۱۰۹ فصل ۴ سیستم‌های جامع دانشگاهی

۱۱۰	۱-۴ مقدمه.....
۱۱۰	۲-۴ علت نیاز به سیستم جامع دانشگاهی.....
۱۱۱	۳-۴ قابلیت‌های سیستم‌های جامع دانشگاهی.....
۱۱۵	۴-۴ آشنایی با تعدادی از سیستم‌های موجود.....
۱۱۵	۱-۴-۴ نرم‌افزار مروارید.....
۱۱۶	۲-۴-۴ نرم‌افزار ناد.....

۱۱۷.....	۳-۴-۴ نرم افزار گلستان.....
۱۱۹.....	۴-۴-۴ سیستم سدف.....
۱۲۰.....	۵-۴-۴ سیستم سما.....
۱۲۱.....	۵-۴ مقایسه سیستم‌های نرم‌افزاری دانشگاهی.....
۱۲۶.....	۶-۴ جمع‌بندی و نتیجه‌گیری.....

## فصل ۵ ارائه چارچوب بهبود یافته

۱۲۷	
۱۲۸.....	۱-۵ مقدمه.....
۱۲۸.....	۲-۵ اصول اصلی معماری سرویس‌گرا.....
۱۳۱.....	۳-۵ قابلیت‌های سیستم‌های نرم‌افزاری موجود کشور.....
۱۳۱.....	۴-۵ فازهای چارچوب بهبود یافته.....
۱۴۳.....	۵-۵ مقایسه فازهای چارچوب بهبود یافته با فاز متناظر سایر روش‌ها.....
۱۴۷.....	۶-۵ ارائه یک نمونه.....
۱۷۷.....	۷-۵ جمع‌بندی و نتیجه‌گیری.....

## فصل ۶ ارزیابی و جمع‌بندی

۱۷۹	
۱۸۰.....	۱-۶ مقدمه.....
۱۸۰.....	۲-۶ یافته‌های تحقیق.....
۱۸۲.....	۳-۶ ارزیابی چارچوب بهبود یافته.....
۱۸۵.....	۱-۳-۶ نحوه تکمیل پرسشنامه.....
۱۸۶.....	۲-۳-۶ تحلیل آماری.....
۱۹۵.....	۴-۶ نوآوری تحقیق.....
۱۹۶.....	۵-۶ پیشنهادها.....

## مراجع

۱۹۷

## واژه‌نامه

۲۰۱

## ضمائم

۲۱۲

۲۱۳.....	پیوست ۱ پرسشنامه خام.....
۲۲۸.....	پیوست ۲ نمونه پرسشنامه تکمیل شده.....

صفحه	عنوان
۱	فصل ۱ مقدمه
۹	<b>فصل ۲ معماری نرم افزار و سبکهای آن</b>
۱۱	شکل ۲-۱ سیر مطالعات مربوط به معماری نرم افزار (کروچتن و همکاران، ۲۰۰۶).....
۱۲	شکل ۲-۲ نقش معماری نرم افزار (کلمنتز و همکاران، ۲۰۰۳).....
۱۵	شکل ۲-۳ دسته بندی سبکها توسط گارلن و شاو (گارلن و شاو، ۱۹۹۴).....
۱۶	شکل ۲-۴ سبک لایه ای (گارلن و شاو، ۱۹۹۴).....
۱۸	شکل ۲-۵ سبک برنامه اصلی و زیر روال (باس و همکاران، ۲۰۰۳).....
۱۹	شکل ۲-۶ سبک مخزن (باس و همکاران، ۲۰۰۳).....
۲۰	شکل ۲-۷ سبک تخته سیاه (گارلن و شاو، ۱۹۹۴).....
۲۲	شکل ۲-۸ احضار ضمنی (لند، ۲۰۰۱).....
۲۴	شکل ۲-۹ سبک پردازش دسته ای متوالی (باس و همکاران، ۲۰۰۳).....
۲۵	شکل ۲-۱۰ سبک لوله ها و فیلترها (گارلن و شاو، ۱۹۹۴).....
۲۶	شکل ۲-۱۱ موازی کار کردن سبک لوله ها و فیلترها (لند، ۲۰۰۱).....
۳۰	شکل ۲-۱۲ دسته بندی سبکها توسط بوسچمن و همکاران (بوسچمن و همکاران، ۱۹۹۶).....
۳۱	شکل ۲-۱۳ دسته بندی سبکها توسط فیندینگ (فیندینگ، ۲۰۰۰).....
۳۲	شکل ۲-۱۴ دسته بندی سبکها توسط باس و همکاران (باس و همکاران، ۲۰۰۳).....
۵۷	<b>فصل ۳ معماری سرویس گرا و روش های آن</b>
۶۷	شکل ۳-۱ جایگاه EA، BPM، OOAD در معماری سرویس گرا (زیمرن و همکاران، ۲۰۰۴).....
۷۱	شکل ۳-۲ فازهای روش SOMA (آرسنجی و همکاران، ۲۰۰۸).....
۷۴	شکل ۳-۳ مراحل انجام فعالیت های SOMA (آرسنجی و همکاران، ۲۰۰۸).....
۷۷	شکل ۳-۴ چارچوب CBDI-SAE (آلن، ۲۰۰۷).....
۷۹	شکل ۳-۵ روش SOAF (اردی، ۲۰۰۶).....
۸۰	شکل ۳-۶ فازهای روش SOUP (میتال، ۲۰۰۶).....
۸۱	شکل ۳-۷ تطبیق روش RUP و SOUP (میتال، ۲۰۰۶).....
۸۵	شکل ۳-۸ فعالیت های XP نسبت به SOUP (میتال، ۲۰۰۶).....
۸۶	شکل ۳-۹ روش پاپازگلو (پاپازگلو و واندين هیول، ۲۰۰۶).....
۹۲	شکل ۳-۱۰ فرایند عمومی توسعه بر اساس MSOAM (ارل، ۲۰۰۵).....
۹۳	شکل ۳-۱۱ اجرای MSOAM با استفاده از روش بالا به پایین (ارل، ۲۰۰۵).....
۹۵	شکل ۳-۱۲ اجرای MSOAM با استفاده از روش پایین به بالا (ارل، ۲۰۰۵).....

شکل ۳-۱۳ اجرای MSOAM با استفاده از روش چابک (ارل، ۲۰۰۵)..... ۹۶

۱۰۹

## فصل ۴ سیستم های جامع دانشگاهی

۱۲۷

### فصل ۵ ارائه چارچوب بهبود یافته

شکل ۵-۱ نمونه فرم شناسایی سرویس..... ۱۳۷

شکل ۵-۲ چارچوب بهبود یافته..... ۱۴۲

شکل ۵-۳ چارت سازمانی سازمان مرکزی دانشگاه پیام نور..... ۱۴۸

شکل ۵-۴ ساختار بخش های دانشگاه پیام نور..... ۱۴۸

شکل ۵-۵ عملیات قابل انجام توسط دانشجو..... ۱۵۱

شکل ۵-۶ فرم تکمیل و تایید اطلاعات شخصی..... ۱۵۲

شکل ۵-۷ نمونه پرسشنامه جهت جمع آوری نیازمندی ها..... ۱۵۳

شکل ۵-۸ فرایندهای موجود برای پذیرش و انتخاب واحد..... ۱۵۵

شکل ۵-۹ نمودار ارتباط اجزاء برای عملکردهای پایه ای..... ۱۵۶

شکل ۵-۱۰ اجزاء داخلی و خارجی برای انجام فرایندها..... ۱۵۶

شکل ۵-۱۱ نمودار فرایند پذیرش دانشجوی جدیدالورود..... ۱۵۷

شکل ۵-۱۲ نمودار ایجاد و تکمیل پرونده الکترونیکی دانشجو..... ۱۵۸

شکل ۵-۱۳ نمودار فرایند انتخاب واحد..... ۱۵۹

شکل ۵-۱۴ نمودار فرایند معادسازی دروس..... ۱۵۹

شکل ۵-۱۵ نمودار اشیاء فرایند انتخاب واحد..... ۱۶۰

شکل ۵-۱۶ نمودار مدیریت پروژه..... ۱۶۴

شکل ۵-۱۷ نمودار میله ای مدیریت پروژه..... ۱۶۵

شکل ۵-۱۸ طبقه بندی لایه ها..... ۱۷۳

شکل ۵-۱۹ نمودار توالی اعتبارسنجی دانشجو..... ۱۷۴

شکل ۵-۲۰ نمودار توالی انتخاب واحد..... ۱۷۵

شکل ۵-۲۱ نمودار فعالیت انتخاب واحد..... ۱۷۶

شکل ۵-۲۲ معماری سرویس های مربوط به پذیرش و انتخاب واحد..... ۱۷۷

۱۷۹

### فصل ۶ ارزیابی و جمع بندی

شکل ۶-۱ نمودار سطح تحویلات پاسخ دهندگان..... ۱۸۷

شکل ۶-۲ نمودار نوع تخصص پاسخ دهندگان..... ۱۸۷

شکل ۶-۳ نمودار رشته تحصیلی پاسخ دهندگان..... ۱۸۸

شکل ۶-۴ نمودار گروه سنی پاسخ دهندگان..... ۱۸۸

شکل ۶-۵ نمودار سابقه کار پاسخ دهندگان..... ۱۸۹

شکل ۶-۶ نمودار میانگین روش ها..... ۱۹۰

شکل ۶-۷ نمودار میانگین پاسخهای روش ها در سوال ۱ از گروه T..... ۱۹۳

- شکل ۶-۸ نمودار میانگین پاسخهای روش‌ها در سوال ۸ از گروه F..... ۱۹۳
- شکل ۶-۹ نمودار میانگین پاسخهای روش‌ها در سوال ۱۹ از گروه S..... ۱۹۴
- شکل ۶-۱۰ نمودار میانگین پاسخهای روش‌ها در سوال ۲۰ از گروه R..... ۱۹۴
- شکل ۶-۱۱ نمودار میانگین پاسخهای روش‌ها در سوال ۲۴ از گروه M..... ۱۹۵

۱۹۷	مراجع
۲۰۱	واژه‌نامه
۲۱۲	ضمائم

## فهرست جداول

صفحه	عنوان
۱	فصل ۱ مقدمه
۹	فصل ۲ معماری نرم افزار و سبک های آن
۱۷	جدول ۱-۲ مزایا و معایب سبک لایه ای
۱۷	جدول ۲-۲ مزایا و معایب سبک شیء گرا
۱۸	جدول ۳-۲ مزایا و معایب سبک برنامه اصلی و زیر روال
۲۰	جدول ۴-۲ مزایا و معایب سبک مخزن
۲۱	جدول ۵-۲ مزایا و معایب سبک تخته سیاه
۲۳	جدول ۶-۲ مزایا و معایب سبک مبتنی بر رویداد
۲۷	جدول ۷-۲ مزایا و معایب سبک لوله ها و فیلترها
۲۸	جدول ۸-۲ مزایا و معایب مفسرها
۳۴	جدول ۹-۲ مزایا و معایب سبک کلاینت/سرور
۳۷	جدول ۱۰-۲ مزایا و معایب سبک دسترسی داده از راه دور
۳۹	جدول ۱۱-۲ مزایا و معایب سبک مبتنی بر مولفه
۴۰	جدول ۱۲-۲ مزایا و معایب سیستم برنامه ریزی منابع سازمان
۴۱	جدول ۱۳-۲ مزایا و معایب سبک های مبتنی بر عامل
۴۳	جدول ۱۴-۲ مزایا و معایب سبک های مبتنی بر عامل های هوشمند
۴۶	جدول ۱۵-۲ مزایا و معایب معماری سرویس گرا
۴۷	جدول ۱۶-۲ مزایا و معایب سبک مبتنی بر استفاده مجدد
۴۹	جدول ۱۷-۲ مزایا و معایب سبک جنبه گرا
۵۳	جدول ۱۸-۲ مقایسه سبک های معماری نرم افزار (علامت- نشان دهنده ضعف آن قابلیت، +/ نشان دهنده حد متوسط بین ضعف و قدرت آن ویژگی و نشان دهنده قدرت آن قابلیت در سبک مورد نظر می باشد و در صورت عدم وجود اطلاعات علامت $\odot$ ثبت شده است)
۵۷	فصل ۳ معماری سرویس گرا و روش های آن
۶۹	جدول ۱-۳ مزایا و معایب روش تحلیل و طراحی سرویس گرا
۷۵	جدول ۲-۳ مزایا و معایب روش معماری و مدلسازی سرویس گرا
۷۶	جدول ۳-۳ مزایا و معایب روش RQ

۷۷.....	جدول ۳-۴ مزایا و معایب روش CBI-DAE
۸۰.....	جدول ۳-۵ مزایا و معایب روش SOAF
۸۶.....	جدول ۳-۶ مزایا و معایب روش SOUP
۸۸.....	جدول ۳-۷ مزایا و معایب روش پاپازگلو
۸۹.....	جدول ۳-۸ مزایا و معایب روش استیون جونز
۹۱.....	جدول ۳-۹ مزایا و معایب روش RUP for SOA
۹۷.....	جدول ۳-۱۰ مزایا و معایب روش MSOAM
۱۰۳.....	جدول ۳-۱۱ مقایسه روش‌های معماری سرویس‌گرا

## فصل ۴ سیستم‌های جامع دانشگاهی ۱۰۹

۱۲۳.....	جدول ۴-۱ مقایسه روش‌های معماری سرویس‌گرا (علامت ✓ به منزله وجود آن ویژگی و علامت ✗ نشان‌دهنده وجود این ویژگی در برخی از بخش‌های سیستم و علامت ✗ نشان‌دهنده عدم وجود آن ویژگی می‌باشد و در صورت عدم وجود اطلاعات علامت ⊙ ثبت شده است)
----------	--

## فصل ۵ روش بهبود یافته ۱۲۷

۱۶۲.....	جدول ۵-۱ ریسک‌های فناوری
۱۶۳.....	جدول ۵-۲ ریسک‌های فرایندی
۱۶۳.....	جدول ۵-۳ ریسک‌های محیطی
۱۷۰.....	جدول ۵-۴ مشخصات سرویس تولید شماره دانشجویی

## فصل ۶ ارزیابی و جمع‌بندی ۱۷۹

۱۸۵.....	جدول ۶-۱ تحلیل پایایی پرسشنامه
۱۸۶.....	جدول ۶-۲ محاسبه آلفای کرونباخ
۱۹۰.....	جدول ۶-۳ مقایسه میانگین و انحراف معیار
۱۹۲.....	جدول ۶-۴ مقایسه چهار روش

## مراجع ۱۹۷

## واژه‌نامه ۲۰۱

## ضمائم ۲۱۲

## فهرست علائم اختصاری

BPM	Business Process Management	مدیریت فرایندهای کسب و کار
C\$SS	Client-Cache-Stateless-Server	کلاینت/سرور بدون حالت دارای حافظه پنهان
CBD	Component-Based Design	طراحی مبتنی بر مولفه
CRM	Customer Relationship Management	مدیریت ارتباط با مشتری
CS	Client-Server	کلاینت/سرور
CSS	Client-Stateless-Server	کلاینت/سرور بدون حالت
DNS	Domain Name System	سرویس دهنده نام
EA	Enterprise Architecture	معماری سازمانی
EBI	Event Based Integration	مجموع سازی بر پایه رویداد
ER	Relationship Entity	ارتباط موجودیت
ERP	Enterprise Resource Planning	سیستم‌های برنامه‌ریزی منابع سازمان
KS	Knowledge Source	منبع دانش
LC\$SS	Layered-Client-Cache-Stateless-Server	کلاینت/سرور بدون حالت دارای حافظه پنهان لایه‌ای
LCODC\$SS	Layered-Code-On-Demand-Client-Cache-Stateless-Server	لایه درخواست کد کلاینت/سرور بدون حالت دارای حافظه پنهان لایه‌ای
LCS	Layered-Client-Server	کلاینت/سرور لایه‌ای
MA	Mobile Agent	عامل متحرک
OOAD	Object Oriented Analysis And Design	تحلیل و طراحی شیء‌گرا
PF	Pipe& Filter	لوله و فیلتر
QA	Quality Assurance	تضمین کیفیت
RDA	Remote Data Access	دسترسی داده از راه دور
REV	Remote Evaluation	ارزیابی از راه دور
RF	Reference Framework	چارچوب مرجع
ROI	Return On Investment	نرخ بازگشت سرمایه



## فهرست علائم اختصاری

---

RQ	Repeatable Quality	کیفیت تکرارپذیر
RS	Remote Session	جلسه از راه دور
RUP	Rational Unified Process	فرایند یکپارچه رشنال
SCM	Supply Chain Management	مدیریت زنجیره تامین
SOA	Service Oriented Architecture	معماری سرویس‌گرا
SOAD	Service-Oriented Analysis and Design	تحلیل و طراحی سرویس‌گرا
SOAF	Service Oriented Architecture Framework	چارچوب معماری سرویس‌گرا
SOAP	Simple Object Access Protocol	پروتکل دسترسی ساده به شیء
SOMA	Service Oriented Modeling and Architecture	معماری و مدل‌سازی سرویس‌گرا
SOUP	Service Oriented Unified Process	فرایند یکنواخت سرویس‌گرا
TCP/IP	Transmission Control Protocol/ Internet Protocol	پروتکل کنترل انتقال/پروتکل اینترنت
UDDI	Universal Description Discovery and Integration	تجمیع و کشف توصیفات عمومی
UML	Unified Modeling Language	زبان مدل‌سازی یکنواخت
UPF	Uniform Pipe& Filter	لوله و فیلترهای یکنواخت
VM	Virtual Machine	ماشین مجازی
WSDL	Web Services Description Language	زمان توصیف وب سرویس‌ها

# فصل ۱

## مقدمه

## ۱-۱ مقدمه

یکی از مهم‌ترین ویژگی‌های سیستم‌های نرم‌افزاری امروزی پیچیدگی و توزیع‌شدگی آنها است که تولید و پشتیبانی را مشکل کرده است پس لازم است از سبک‌های<sup>۱</sup> از قبل طراحی شده در تولید این سیستم‌ها استفاده نمود تا سیستم در فرایند طراحی و پشتیبانی با شکست مواجه نگردد. وظیفه اصلی معماری نرم‌افزار<sup>۲</sup> ایجاد دید کامل از سیستم است و در هر یک از سبک‌های آن اجزاء سیستم به شیوه‌های مختلفی در نظر گرفته شده است. جهت انتخاب سبک مناسب برای سیستم مورد نظر باید یک دید کلی نسبت به سبک‌های متداول معماری نرم‌افزار داشت تا مناسب‌ترین آن برای سازمان مورد نظر انتخاب گردد. در انتهای فصل دوم معیارهایی جهت مقایسه سبک‌ها انتخاب و نتیجه در قالب یک جدول جمع‌آوری و خلاصه شده است. براساس تحقیقات به عمل آمده سبک معماری سرویس‌گرا<sup>۳</sup> به دلایلی از قبیل استقلال از سکو<sup>۴</sup>، ایجاد یکپارچگی، قابلیت مقیاس‌پذیری<sup>۵</sup>، قابلیت گسترش<sup>۶</sup>، پشتیبانی مناسب و قابلیت استفاده مجدد برای سیستم‌های جامع دانشگاهی انتخاب می‌شود. این معماری وظایف و توانایی‌های سیستم را در قالب واحدهای مجزایی به نام سرویس تقسیم‌بندی کرده که این سرویس‌ها از قابلیت ویژه‌ای به نام اتصالات سست استفاده می‌نمایند. این سبک معماری دارای روش‌های<sup>۷</sup> مختلفی است که هر یک از آنها از ویژگی‌های خاصی بهره‌مند هستند. هدف از این

---

<sup>1</sup> Styles

<sup>2</sup> Software Architecture

<sup>3</sup> Service Oriented Architecture (SOA)

<sup>4</sup> Platform

<sup>5</sup> Scalability

<sup>6</sup> Deployment

<sup>7</sup> Methods

تحقیق ارائه چارچوب بهبودیافته‌ای از ترکیب روش‌های بررسی شده است تا بتوان برای طراحی سیستم‌های نرم‌افزاری جامع دانشگاهی در آن استفاده نمود.

## ۲-۱ تعریف مساله و سئوالات اصلی تحقیق

در راستای استفاده از معماری نرم‌افزار و انتخاب معماری سرویس‌گرا برای سیستم‌های جامع دانشگاهی سئوالات زیر مطرح است:

- انتخاب سبک معماری برای سیستم‌های مورد نظر ما چقدر تأثیرگذار است؟
- سبک مناسب برای سیستم‌های جامع دانشگاهی کدام است؟
- نیازهای دانشگاه‌ها برای تولید سیستم‌های نرم‌افزاری جامع دانشگاهی کدام است؟
- چه فرآیندهایی را باید برای شناسایی و تعریف سرویس‌ها انجام داد؟
- روش‌ها و چارچوب مناسب برای پیاده‌سازی سرویس‌های فوق در این سیستم‌ها به چه صورت می‌باشد؟
- جهت نگهداری<sup>۱</sup> بهتر و کم‌هزینه‌تر از سیستم پس از تولید، چه راهکارهایی ارائه خواهد شد؟
- معایب و مزایای روش‌های معماری سرویس‌گرا چیست؟

## ۳-۱ سابقه و ضرورت انجام تحقیق

در روش‌های قدیمی، نرم‌افزارها از طریق شیوه تحلیل بالا به پایین<sup>۲</sup> تولید می‌شدند به صورتی که پس از تحلیل، سیستم طراحی و پیاده‌سازی شده سپس به اجرا در می‌آمد و در صورت نیاز، نرم‌افزار به اجزا کوچک‌تر تقسیم شده و هر جزء تست می‌گردید، تا در صورت لزوم ارتقا یابد که این روش سنتی برای سیستم‌های امروزی که از پیچیدگی بیشتری برخوردار هستند جوابگو نخواهد بود و هزینه‌های بسیار زیادی را

<sup>۱</sup> Maintenance

<sup>۲</sup> Top-Down