هشتمین کنفرانس ملی
مطالعات مدیریت و اقتصاد در علوم انسانی
8th National Conference on
Management Studies and Economics in the Humanities
www.spconf.ir

نقش اهداف و تصمیمات با معماری نرم افزار ارتباطی در مدیریت خدمات شهری

حسین رضازاده[۱]، سهیلا محمدی[۲]

۱.رییس ناحیه سه شهرداری شیراز، h.rezazadeh1023@gmail.com
۲.رییس اداره ارتباطات و فرهنگی شهرداری منطقه ۹ شیراز، So.mohamadi1513@gmail.com

چکیده:
معماری نرم‌افزار، مجموعه‌ای از ساختارهای لازم برای استدلال در مورد یک سیستم می‌باشد که شامل عناصر نرم‌افزاری، روابط بین آنها و ویژگی‌های هردو خواهد بود. معماری نرم‌افزار یک پایه محکم برای ایجاد و توسعه نرم‌افزار را فراهم می‌کند که اکثراً با توجه به تعداد کاربر، زیرساخت، اهداف کسب و کار و شاید از همه مهم‌تر توسعه‌پذیری آن سیستم انتخاب می‌شود. نرم‌افزار اتوماسیون کسب و کار فراگستر یک سامانه کاملاً تحت وب می‌باشد که امکان بهره‌برداری از آن در سازمان‌های کوچک و متوسط تا سازمان‌های دارای گستردگی و با هزاران کاربر به راحتی ممکن بوده و برای برآورده شدن همین هدف این سیستم بر اساس الگویی از معماری طراحی و تولید گردیده است. در این مقاله مدل معماری نرم افزاری ارائه شد که به ساخت سیستم های اطلاعاتی شهری کمک می کند. این مدل با اصول سیستم های اطلاعاتی شهری سازگار است. از این رو، این مدل به سیستم اطلاعاتی کمک می کند تا چابک تر باشد و با نیازهای کاربر ، مقیاس پذیرتر و سازگارتر است. ضمن اینکه این مدل باید از طریق آزمایشات به منظور استفاده بهتر در اصول سیستم های اطلاعاتی ارزیابی شود. علاوه بر این دو سوال پاسخ داده نشده باقی می ماند. سوال اول در مورد اثر بخشی استفاده از این مدل برای معماری "سیستم های تصمیم یار هوشمند" می باشند. سوال دوم در مورد ادغام معماری سیستم های نرم افزاری برای استفاده از مدیریت ارتباط با مشتری و برنامه ریزی منابع سازمانی می باشد. این مشکلات می توانند جهت های تحقیقات آینده را مشخص کنند.

واژگان کلیدی: مدیریت ، اتوماسیون اداری، معماری نرم افزار، فناوری اطلاعات، خدمات شهری.

مقدمه

شهرنشینی باعث ایجاد ساخت و سازها و زیرساختهای بزرگ در مناطق جغرافیایی مختلف شده است. به دلیل وجود هزینه های سنگین برای ساخت پروژههای بزرگ، اکثر هزینه ها توسط دولتها و نهادهای سرمایه گذاری تامین میشوند. به همین دلیل، مشکلات جدی در این باره گزارش میشوند، مانند کیفیت پایین به دلیل محدود کردن مدیریت حرفهای ساخت و سازها، تاراج کردن بودجه ها و بروز فساد توسط رانت. تقریباً اکثر سازمانهای پیشرفته با فشارهای همیشه در حال تغییر اقتصادی، فنی، اجتماعی و سیاسی مواجه میشوند. پس آنها خود را به طور پیوسته با اولویتها، فرآیندهای محصولات، سرویسها و روابط با مشتریان و تامین کنندگان منطبق میکنند. بنابراین امروزه بعضی از سازمانها ،بسیار زیاد از فناوریهای اطلاعاتی و ارتباطی تا زمان اجرای فرآیندهای سازمانی استفاده میکنند. با این وجود، سیستمهای اطلاعاتی چهار ویژگی دارند: پیچیدگی، انطباق پذیری، قابلیت تغییر و غیر قابل مشاهده بودن به طور خاص، پیچیدگی سیستم اطلاعاتی در دو حالت ساختاری و سیستمی وجود دارد. پیچیدگی ساختاری در سیستم اطلاعاتی همراه با ویژگیهای ساختاری است. مانند: تعداد و اندازه نرم افزارهای کاربردی. اما پیچیدگی سیستمی در سیستم اطلاعاتی با تعامل بین اطلاعاتِ در حال جریان در سیستم و سرویسهای رد و بدل شده در سیستمهای اطلاعاتی خارجی مرتبط است. پیچیدگی سیستم اطلاعاتی به دلایل وجود مشکلات زیادی است. از یک طرف، دشواری در برقراری ارتباط بین سیستم اطلاعاتیِ سهامدارن که منجر به کیفیت پایین، هزینه و تاخیر زیاد می شود. از طرف دیگر، دشواری در فهمیدن همه قسمتهای نرمافزارهای کاربردی، ارتقا و نگهداری آنان میباشد. در نهایت دشواری در یک دید جهانی نسبت به سیستم اطلاعاتی که ممکن است مفهوم و تمامیت سیستم اطلاعاتی را به خطر اندازد. نتیجه این تغییرات در ارزیابی سیستم اطلاعاتی میباشد که این ارزیابی به طور پیوسته احتیاج به این دارد که سازمان را با سیستم اطلاعاتی منطبق کرد که بتوان فشارهای محیط کسب و کار بر سازمان را برآورد کرد.

مشکلات بیان شده مانع از این میشوند که یک سیستم اطلاعاتی چابک و مشتری مدار در سازمان ساخته شود که توسط سیستم اطلاعاتی باز و چابک پشتیبانی شود که این سیستم اطلاعاتی بتواند در حالت موثر و امن با سیستم اطلاعاتی مشتریها و تامین کنندگان ادغام شود. با وجود این نقاط ضعف، مدلهای معماری اختصاصی برای تعریف راه حلهای نرم افزاری سازگار با اصول و قوانین سیستمهای اطلاعات شهری وجود ندارد. هم چنین مدلهای معماری نرم افزاری معروف، منابع برای رفع نیازها و محدودیتهای سیستم اطلاعات شهری را فراهم نمیکنند. از این رو در این مقاله مدل "معماری نرم افزاری ارتباطی" را ارائه شده است. این مدل به سازمانها برای ساخت راه حلیهای نرم افزاری شهری کمک میکند. هم چنین این مدل، مدلهای معماری نرم افزاری را بهبود میبخشد و به آنها اجازه میدهد که با مدلهای معماری جدید ادغام شوند. در ادامه مدل"معماری نرم افزاری ارتباطی" را بیان میگردد. همچنین روش تحقیق بیان میشود و در آخر نتیجه‌گیری ارائه خواهد شد.

۱. معماری نرم افزار چیست؟

معماری نرم افزار به ساختارهای بنیادی یک سیستم نرم افزاری و نظم ایجاد چنین ساختارها و سیستم هایی اشاره دارد .هر ساختار در نرم افزار شامل عناصر نرم افزاری، روابط بین آن ها و خصوصیات هر دو عنصر و روابط است .در واقع معماری یک سیستم، اجزای اصلی آن، روابط آن ها (ساختارها) و نحوه تعامل آن ها با یکدیگر را توصیف می کند .معماری یک سیستم نرم افزاری مشابه معماری یک ساختمان است .این معماری به عنوان یک طرح اصلی برای سیستم و پروژه در حال توسعه عمل می کند و وظایفی را که باید توسط تیم های طراحی انجام شود، مشخص می کند .معماری نرم افزار در واقع انتخاب ساختاری اساسی است که تغییر آن پس از اجرا پر هزینه است .گزینه های معماری نرم افزار شامل گزینه های ساختاری خاص از امکانات موجود در طراحی نرم افزار است .به عنوان مثال، سیستم هایی که وسیله نقلیه پرتاب شاتل فضایی را کنترل می کنند باید بسیار سریع و بسیار قابل اعتماد باشند.بنابراین، باید یک زبان محاسباتی در زمان واقعی مناسب انتخاب شود .علاوه بر این برای برآوردن نیاز قابلیت اطمینان بودن می توان چندین نسخه اضافی و مستقل از برنامه تولید کرد واین نسخه ها را روی سخت افزار مستقل اجرا کرد و نتایج را بررسی کرد.

۲.معماری فناوری اطلاعات

معماری فناوری اطلاعات شرح مفصلی از دارایی های مختلف پردازش اطلاعات مورد نیاز برای دستیابی به اهداف تجاری، قوانین حاکم بر آنها و اطلاعات مرتبط با آنها است. گارتنر، تحلیلگر بازار فناوری اطلاعات، معماری فناوری اطلاعات را اینگونه تعریف می‌کند: «مجموعه‌ای از اصول، دستورالعمل‌ها یا قوانینی است که توسط یک شرکت برای هدایت فرآیند کسب، ساخت، اصلاح و ارتباط منابع IT در سراسر سازمان استفاده می‌شود.» یک معمار فناوری اطلاعات مانند یک نقشه برای یک شهر است. شبکه خیابان، سیستم آب، سیستم فاضلاب، خدمات اضطراری و شبکه های برق را می سازد. این چارچوب اساسی برای تمام رایانه ها، سیستم ها و مدیریت اطلاعات است که از خدمات سازمانی پشتیبانی می کند. مانند طرح شهر، معماری فناوری اطلاعات پیچیده است و با پشتیبانی از خدمات بیشتر و استفاده از فناوری های مختلف، این پیچیدگی در حال افزایش است.

انواع معماری فناوری اطلاعات

نکته دیگری که باعث سردرگمی افراد می شود این است که انواع مختلفی از معماری های فناوری اطلاعات وجود دارد. واقعیت این است که مانند زمانی که در مورد طراحی یک ساختمان به عنوان نمونه موازی صحبت می شود، سطوح مختلفی از معماری مانند معماری کلی ساختمان، پلان طبقات برای هر طبقه و طراحی های داخلی وجود دارد. همین امر در مورد معماری فناوری اطلاعات نیز صدق می کند و ما می توانیم سه نوع معماری فناوری اطلاعات را تشخیص دهیم: معماری سازمانی(EA) معماری راه حل(SA) معماری فناوری(TA) معماری سازمانی (EA) نقشه کل شرکت است و معماری کل شرکت را تعریف می کند. این شامل تمام برنامه‌ها و سیستم‌های فناوری اطلاعات است که در داخل شرکت و توسط بخش‌های شرکت‌های مختلف از جمله همه برنامه‌ها، پلت‌فرم‌های یکپارچه‌سازی، وب، پورتال و برنامه‌های تلفن همراه، ابزارهای تحلیل داده‌ها استفاده می‌شوند. انبار داده و دریاچه داده، ابزارهای عملیاتی و توسعه، امنیت، و برنامه های مشترک (مانند ایمیل، چت، فایل های سیستمی) و غیره .طرح EA تمام سیستم های IT را در یک نقشه منطقی در بر می گیرد.

READ راه حل کاهش تورم

هدف یک معماری سازمانی تمرکز بر روی کارآمد ساختن فناوری اطلاعات برای کل شرکت و تجارت و متناسب با اهداف شرکت ها و کسب و کار است. به استراتژی کسب‌وکار نگاه می‌کند و راه‌حل‌های فناوری اطلاعات را برای کارکرد آن از جمله نوآوری و توانمندسازی دیجیتال پیدا می‌کند. بنابراین، یک معمار سازمانی باید بتواند هم فناوری اطلاعات و هم تجارت را در سطح بالایی درک کند. معماری راه حل توصیف می کند که یک سیستم خاص باید چه عملکردهایی را انجام دهد. این یک توصیف دقیق از عملکردهای مورد نیاز برای دستیابی به اهداف تجاری، منطق حاکم بر آنها و اطلاعات مرتبط با آنها است. همچنین به عنوان معماری عملکردی یک برنامه یا سیستم توصیف می شود.

یک معماری راه حل معمولاً برای یک پروژه یا انتشار پروژه به کار می رود و ترجمه الزامات را به چشم انداز راه حل، کسب و کار سطح بالا و/یا مشخصات سیستم فناوری اطلاعات تسهیل می کند. این طرح از تیم معماری سازمانی از نظر کسب و کار شرکتی، اطلاعات و راهنمایی فنی جهت گیری می کند. تفاوت بین یک معماری راه حل و معماری سازمانی دراین است که نتیجه آن به یک راه حل خاص منتهی می شود نه کل یک شرکت.

معماری فناوری شرح مفصلی از اجزای مختلف فناوری مورد نیاز برای دستیابی به اهداف تجاری، منطق حاکم بر آنها و داده های مرتبط با آنها است. به طور خلاصه، معماری فناوری اطلاعات معماری نرم‌افزار و سخت‌افزار را نشان می‌دهد و کمتر با استراتژی کلی کسب‌وکار و شرکت مرتبط است، اما بیشتر بر این تمرکز دارد که چگونه راه‌حل خاصی می‌تواند توسط این پلتفرم ارائه شود.

معماران فناوری بر نحوه طراحی و ساخت اجزا برای کمک به شما در یافتن راه حل های نرم افزاری و سخت افزاری قوی و مقرون به صرفه تمرکز می کنند. آنها به عنوان دروازه بین تیم توسعه نرم افزار و کسب و کار عمل می کنند تا مطمئن شوند که نیازهای تجاری برآورده شده است.معماری فناوری اجزای برنامه را از معماری برنامه با مؤلفه های فناوری که مؤلفه های نرم افزار و سخت افزار را نشان می دهد، مرتبط می کند. اجزای آن عموماً در بازار به دست می آیند و می توانند مونتاژ و پیکربندی شوند تا زیرساخت های فناوری شرکت را تشکیل دهند. معماری فناوری دیدگاه ملموس تری از روشی که در آن اجزای برنامه کاربردی تحقق می یابند و به کار می روند، ارائه می دهد. این امکان را فراهم می کند که مشکلاتی که می توانند بین مراحل مختلف مسیر تکامل IS ایجاد شوند، زودتر مطالعه شوند. ابزار دقیق‌تری برای ارزیابی

پاسخ‌ها به محدودیت‌ها (الزامات غیرعملکردی) مربوط بهIS ، به ویژه با تخمین نیازهای سخت‌افزار و اندازه شبکه یا با راه‌اندازی سرور یا افزونگی ذخیره‌سازی ارائه می‌کند. معماری فناوری بر مشکلات لجستیکی و مکان یابی مربوط به مکان سخت افزاری، قابلیت های مدیریت IS، و سایت هایی که بخش های مختلف IS در آن استفاده می شود، متمرکز است. معماری فناوری همچنین تضمین می کند که اجزای برنامه ارائه شده با هم کار می کنند و تأیید می کند که یکپارچه سازی تجاری مورد نیاز پشتیبانی می شود.

رویکرد معماری سازمانی نگره جدیدی است که در آن کلیه فعالیتها و فناوریهای پشتیبان آن به صورت هرمی سلسله مراتبی و در لایه های کسب و کار اطلاعات سیستم های کاربردی و زیرساختها تدوین می شود. این رویکرد بسیار فراتر از متدولوژی های ایجاد سیستم های اطلاعاتی عمل کرده و دامنه آن کل منابع اطلاعاتی و فرایندهای کسب و کار سازمان را دربرمی گیرد. این رویکرد بیشتر در سازمانهای بزرگ و دارای سیستم های موروثی کاربردیافته که در آن سعی بر ساماندهی کلیه منابع اطلاعاتی سازمان همراستا با استراتژی ها و اهداف کلان تجاری بنگاه است. اهمیت کاربریهای فناوری اطلاعات در عصر جدید و قابلیتها و مزایای معماری سازمانی چشم انداز گسترده تر و فراگیری بیشتر این رویکرد را نوید می دهد.

۳-تصمیمات در معماری نرم افزار

معماری نرم افزار شامل مجموعه ای از تصمیمات مهم در مورد سازمان مربوط به توسعه نرم افزار است و هر یک از این تصمیمات می تواند تاثیر قابل توجهی بر کیفیت، نگهداری، عملکرد و موفقیت کلی محصول نهایی داشته باشد.

این تصمیمات شامل موارد زیر است:

- انتخاب عناصر ساختاری و رابط هایی که سیستم توسط آن ها ساخته شده است
- رفتار در همکاری بین این عناصر
- ترکیب این عناصر ساختاری و رفتاری به زیر سیستم بزرگ
- همسو بودن تصمیمات معماری با اهداف تجاری
- سبک های معماری سازمان را راهنمایی کند

۴-اهداف معماری نرم افزار

هدف اصلی این معماری شناسایی نیازهایی است که بر ساختار برنامه تأثیر می گذارد.

یک معماری خوب خطرات تجاری مرتبط با ایجاد راه حل فنی را کاهش می دهد و پلی را بین نیازهای فنی و تجاری ایجاد می کند.

برخی از اهداف معماری نرم افزار عبارتند از:

۱. ساختار سیستم را در معرض دید قرار دهد اما جزئیات اجرای آن را مخفی کند.
۲. تحقق بخشیدن به تمام موارد استفاده و سناریوها
۳. سعی در رسیدگی به الزامات ذینفعان مختلف
۴. رسیدگی به هر دو الزامات عملکردی و کیفیت
۵. کاهش هدف مالکیت و بهبود موقعیت سازمان در بازار
۶. بهبود کیفیت و قابلیت های ارائه شده توسط سیستم
۷. بهبود اعتماد خارجی به هر دو سازمان و سیستم

۵-محدودیت های معماری نرم افزار

معماری نرم افزار هنوز یک رشته در حال ظهور در مهندسی نرم افزار است که محدودیت های زیر را دارد:

- کمبود ابزار و روش های استاندارد برای نمایش معماری
- کمبود روش تجزیه و تحلیل پیش بینی اینکه آیا معماری منجر به پیاده سازی مطابق با الزامات خواهد شد یا خیر.
- عدم آگاهی از اهمیت طراحی معماری در توسعه نرم افزار

- عدم درک نقش معمار نرم افزار توسط ذینفعان و ارتباط ضعیف بین آن ها
- عدم درک فرآیند طراحی، تجربه طراحی و ارزیابی طرح

۶-نقش معمار نرم افزار چیست؟

یک معمار نرم افزار راه حلی را ارائه می دهد که تیم فنی می تواند آن را در کل برنامه ایجاد و طراحی استفاده کند.

خروجی طرحی که از معمار نرم افزار دریافت می شود شامل موارد زیر می شود:

- مجموعه ای واضح، کامل، سازگار و قابل دستیابی از اهداف عملکردی
- شرح عملکرد سیستم، حداقل با دو لایه تجزیه
- مفهومی برای سیستم
- طرحی به شکل سیستم، با حداقل دو لایه تجزیه
- ایده ای از زمان کلی، ویژگی های اپراتور و برنامه های اجرا و بهره برداری
- یک سند یا فرآیند که تضمین می کند تجزیه عملکردی دنبال شده، و نوع رابط کنترل می شود

برای رسیدن به نتایج بالا یک معمار نرم افزار باید در زمینه های زیر تخصص داشته باشد:

۶-۱-تخصص طراحی

- متخصص در طراحی نرم افزار، از جمله روش ها و رویکردهای متنوع مانند طراحی شی گرا، طراحی رویداد محور و غیره
- رهبری تیم توسعه و هماهنگی تلاش های توسعه برای یکپارچگی طراحی
- باید قادر به بررسی پیشنهادات طراحی و سبک و سنگین کردن آن ها جهت استفاده باشد

۶-۲-تخصص دامنه

- متخصص روی سیستم در حال توسعه و برنامه ریزی برای توسعه نرم افزار باشد.
- کمک در روند بررسی نیاز، اطمینان از کامل بودن و ثبات
- هماهنگ سازی در تعریف مدل دامنه برای سیستم در حال توسعه

۶-۳- تخصص فناوری

- متخصص در فناوری های موجود که در اجرای سیستم کمک می کند.
- هماهنگ سازی انتخاب زبان برنامه نویسی، چارچوب، سیستم عامل ها، پایگاه داده و غیره

۶-۴-تخصص روش شناسی

- متخصص روش های توسعه نرم افزار که ممکن است در طول) SDLC چرخه عمر توسعه نرم افزار) اتخاذ شود.
- رویکردها و روش های مناسبی را برای پیشرفت انتخاب کند که به کل تیم کمک کند.

۶-۵- نقش پنهان معمار نرم افزار

- تسهیل کار فنی در میان اعضای تیم و تقویت رابطه اعتماد در تیم
- متخصص اطلاعات که دانش مشترک دارد و تجربه زیادی دارد
- از اعضای تیم در برابر نیروهای خارجی که باعث حواس پرتی آن ها می شود و ارزش کمتری برای پروژه دارند محافظت می کند.

۷- ویژگی های کیفی در معماری نرم افزار

کیفیت معیاری برای تعالی یا عاری بودن از کمبود و نقص است.

ویژگی های کیفی خصوصیاتی از سیستم هستند که از عملکرد سیستم جدا هستند.

پیاده سازی ویژگی های کیفیت، تمایز یک سیستم خوب از یک سیستم بد را آسان می کند.

این ویژگی ها به طور کلی عواملی هستند که بر رفتار زمان اجرا، طراحی سیستم و تجربه کاربر تاثیر می گذارند.
ویژگی های کیفی را می توان در موارد زیر طبقه بندی کرد:
۷-۱ویژگی های کیفیت استاتیک
انعکاس ساختار یک سیستم و سازمان که مستقیماً با معماری، طراحی و کد منبع مرتبط است.
این موارد برای کاربر نهایی نامرئی هستند اما بر هزینه توسعه و نگهداری تاثیر می گذارند.
به عنوان مثال: ماژولار بودن، قابلیت تست، قابلیت نگهداری و غیره
۷-۲ ویژگی های کیفیت پویا
این ویژگی ها بازتاب رفتار سیستم در هنگام اجرای آن است که مستقیما با معماری، طراحی، کد منبع، پیکربندی، پارامترهای استقرار، محیط و بستر سیستم ارتباط دارند.
این موارد برای کاربر نهایی قابل مشاهده هستند و در زمان اجرا وجود دارند.
به عنوان مثال توان عملیاتی، قدرت، مقیاس پذیری و غیره

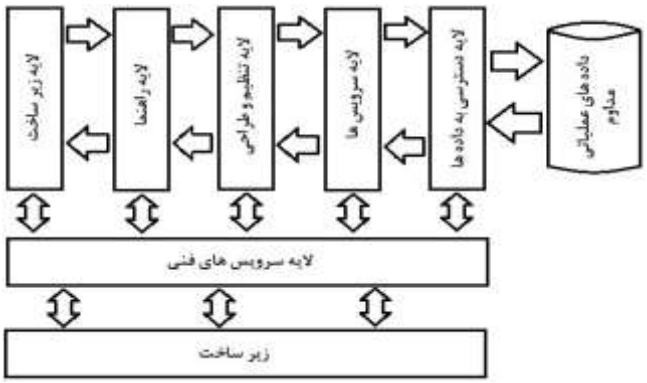
۸-سناریوهای کیفی در معماری نرم افزار

سناریوهای کیفی نحوه جلوگیری از خطا در زمان خرابی را مشخص می کنند.
آن ها را می توان بر اساس مشخصات ویژگی آن ها به شش بخش تقسیم کرد:

- منبع :(Source) موجودیتی داخلی یا خارجی مانند افراد، سخت افزار، نرم افزار یا زیرساخت های فیزیکی
- محرک :(Stimulus) شرایطی که هنگام ورود به سیستم باید مورد توجه قرار گیرد.
- محیط :(Environment) محرک هایی که با شرایطی خاص در محیط ایجاد می شوند.
- سازه ها :(Artifact) کل سیستم یا بخشی از آن مانند پردازنده ها، کانال های ارتباطی، ذخیره سازی مداوم، پردازش ها و غیره
- واکنش :(Response) فعالیتی که پس از رسیدن محرک انجام می شود.مانند شناسایی خطاها، بهبود خطا، غیرفعال کردن منبع رویداد و غیره
- اندازه گیری پاسخ :(Response measure) باید پاسخ های رخ داده را اندازه گیری کرد تا بتوان الزامات را آزمایش کرد.

۹-معماری نرم افزاری ارتباطی

این مدل، معماری سیستم های نرم افزاری که متعلق به سیستم های اطلاعاتی شهری است را توضیح می دهد .مدل معماری نرم افزاری ارتباطی از شش لایه تشکیل شده است. این شش لایه عبارت اند از: زیرساخت، راهنما، تنظیم، طراحی، سرویس، دسترسی به داده در شکل زیر، این لایه ها مشخص شده اند:



شکل ۱- لایه های معماری نرم افزاری ارتباطی

روش تحقیق

در این تحقیق دو راهکار به نام های "شش لایه ای" و "توسعه چارچوب پروژه تطبیقی" ارائه شده است. ابتدا به شرح راهکار "شش لایه ای" می پردازیم: لایه های "زیرساخت" ، "راهنما" ، "تنظیم" ، "طراحی" ، "سرویس ها" و "دسترسی به داده" وجود دارند.

لایه زیرساخت:

نقش ها:

- مدیریت تعامل با کاربر نهایی برای هر یک از کانال های ارتباطی فنی
- مدیریت گرافیکی رابط ماشین و انسان
- کامپوننت ها: نمایش، ویرایش و قالب بندی المان ها

توابع پشتیبانی:

- مدیریت ارائه
- نمایش محتوا
- کنترل قوانین داده ها
- کنترل سطح ورودی داده ها
- کمک کردن آنلاین

مخزن های لایه:

- نمایش رنگ ها
- برچسب های پیام ها
- نمایش زبان های مرتبط

قانون معماری:

- فقط ماژول لایه رابط می تواند با کاربر نهایی تعامل داشته باشد.

لایه راهنما:

نقش ها:

- نمایش پیشرفت در صفحه نمایش حرکتی
- مدیریت داده های خاص که در تعامل با سیستم نرم افزاری و کاربر نهایی هستند
- مدیریت زمینه مرتبط با تبادل جریان داده ها با لایه تنظیم و طراحی
- کامپوننت ها: کانال های فنیِ ارتباطی.

توابع پشتیبانی:

- صفحه نمایش هویت شناسی برای عملکرد وظیفه ها
- تماس با لایه تنظیم و طراحی برای حمل کنترلهای مرتبط با فرآیندهای سازمانی پشتیبانی شده توسط سیستم نرم افزاری

مخزن های لایه:

- لیست نمایش مرتبط با وظیفه ها

قانون معماری:
- فقط ماژول های لایه راهنما می توانند ماژول های لایه تنظیم و طراحی را صدا بزنند.

لایه تنظیم و طراحی:
نقش ها:
- شناسایی فرآیندهای فعالیت سازمانی پشتیبانی شده توسط سیستم نرم افزاری
- مدیریت دنباله ای وظیفه های پشتیبانی شده توسط سیستم نرم افزاری
- شرح نقش های کاربر نهایی
- کنترل کردن تبادل اطلاعات و سرویس ها با سیستم های نرم افزاری دیگر
- مدیریت یک زمینه مربوط به وظایف در حال اجرا به منظور اجازه وقفه بدون انتشار اطلاعات
توابع پشتیبانی:
- شروع و اتمام دنباله ای از وظیفه ها
- نمایش دادن سرویس ها به سایر سیستم های نرم افزاری
- فراخوانی سرویس های نمایش داده شده توسط سیستم های نرم افزاری دیگر
- دنباله ای از فعال کردن سرویس ها) یا دنباله ای از آغاز به کار کردن سرویس ها(
- سرویس های تنظیم
- سرویس های طراحی

مخزن های لایه:
- فهرست وظایفی از ترکیب کردن)درست کردن(مورد استفاده
- فهرستی از سرویس هایِ استفاده شده توسط مورد استفاده) use case(
- شرح دنباله ای از وظایِف ترکیب کردنِکردن مورد استفاده

قانون های معماری:
- قانون اول: داده های تجاری پردازش شده در طول اجرای پروسه ، در متن)زمینه(پروسه مدیریت می شوند. این زمینه منحصراً توسط ماژول "لایه تنظیم و طراحی" مدیریت می شود. هیچ ماژولی از لایه های دیگر نمی تواند به این لایه دسترسی داشته باشد حتی اگر دسترسی "فقط خواندن" داشته باشد.
- قانون دوم: فقط ماژول "لایه تنظیم و طراحی" می تواند جریان اطلاعات و سرویس ها در سایر سیستم های نرم افزاری را تبادل کند.
- قانون سوم: فقط "لایه تنظیم و طراحی" می تواند سرویس ها را برای سایر سیستم های نرم افزاری نمایش دهد.

لایه سرویس ها:
نقش ها:
- میزبانی قانون های مناسب برای موِجودیت های سیستم اطلاعاتی نرم افزاری
- پیاده سازی توابع در حال پردازش موجودیت های سیستم اطلاعاتی نرم افزاری
توابع پشتیبانی:
- شرح دولت)دولت چین(از موجودیت های اطلاعاتی نرم افزاری

- انجام ساده و کنترل های پیچیده

- پردازش داده ها:
- شرح سرویس های پِشتیبانی شده توسط سیستم نرم افزاری
- ضبط کردن وضعیت تغییراتِ موجودیت های سیستم اطلاعاتی نرم افزاری

مخزن های لایه:
- فهرست وظایفی از ترکیب کردن)درست کردن(مورد استفاده
- قوانین محاسبات

قانون معماری:
- لایه سرویس ها، ثبات فرآیند بین کسب و کارها را از طریق بررسی کردن موجودیت های سیستم نرم افزاری تضمین می کند.

لایه دسترسی داده:
نقش:
- فراهم کردن دسترسی مداوم به داده های عملیاتی که وابسته)متعلق(به سیستم نرم افزاری هستند، با مطمئن بودن از وابستگی مستقل بین مدل در حال پردازش و مدل فیزیکی و انجام کنترل های یکپارچگی داده
توابع پشتیبانی:
- انتخاب داده
- به روز رسانی داده
- حذف کردن داده
- ساخت داده
- ویرایش داده
- پیوستن جدول
- کنترل یکپارچگی داده

مخزن لایه:
- فهرستی از جدول هایی که در نگهداری اطلاعات مرتبط با موجودیت های اطلاعاتی استفاده شده اند .

قانون های معماری:
- قانون اول: فقط ماژول "لایه دسترسی داده" ، داده ی خواندن و نوشتن سرویس ها ، در انجام مداوم داده ی عملیاتی متعلق به سیستم نرم افزاری را فراهم می کند.
- قانون دوم: خواندن و نوشتن سرویس های نمایش داده شده ، توسط "لایه دسترسی داده" در طول مدل "داده منطقی" در سیستم نرم افزاری ، تعریف می شوند.

حال به شرح راهکار "توسعه چارچوب پروژه تطبیقی" پرداخته میشود .این راهکار برای پروژه هایی طراحی شده که هدف واضحی ندارند و یک راه حل توسعه ای مشتری محور برای مشتری هایی که نتیجه ها و بازخورد را به موقع ارائه می دهند، به صورت پیوسته توسعه را بهبود می دهند و راه حلها و روش های خود را انعطاف پذیر می کنند. راه حل های این پروژه ها کاملاً جلوتر از زمان نیستند ولی فقط می توانند از طریق انجام پروژه شناخته شوند.

نتیجه گیری

مفهوم سیستمهای اطلاعات شهری از سال 1990 میلادی به منظور کمک به سازمانها برای ساخت سیستمهای اطلاعاتی چابکتر مطرح شده است. به طور خاص، مدلهای معماری اختصاصی برای تعریف راه حلهای نرم افزاری سازگار با اصول و قوانین سیستمهای اطلاعات شهری مفید نیستند. علاوه براین، مدلهای معماری نرم افزاری معروف، رسیدگی به نیازها و محدودیتهای سیستمهای اطلاعات شهری را فراهم نمیکنند. در این مقاله، مدل "معماری نرم افزاری ارتباطی" را ارائه میگردد. این مدل برای سازگاری، انعطاف پذیری، پاسخگویی و توسعه سریعتر با اصول سیستمهای اطلاعات شهری ارائه شده است. از این رو، مدل "معماری نرم افزاری ارتباطی" با نیازهای سازمان، مقیاس پذیرتر و سازگارتر است. سیستم‌های اطلاعاتی بر کیفیت زندگی شخصی و کاری تأثیر گذاشته‌اند. در محل کار، می‌توان سیستم‌های اطلاعاتی را برای از بین بردن کارهای خسته کننده و استقلال بیشتر کارمندان به کار گرفت یا از آن‌ها برای حذف کارهایی که به تفکر نیاز ندارند و فراهم کردن فضایی سالم با مدیریت صحیح برای نیروی کار، استفاده کرد.

منابع و مأخذ:

[1]International Conference on Business Management and Electronic Information (BMEI). IEEE, pp. 99 Lu, Luo, S., Y., Peng, Y., Le, Y. Developing cost control information system in the complex project. 2011 –105, 2011.
[2] F. P. Brooks, Jr., The Mythical Man-Month, Boston, MA, USA: Addison-Wesley Longman Inc., 1995.
[3] pp. 10F. P. Brooks, “No Silver Bullet: Essence and Accidents of Software Engineering,” IEEE Computer, vol. 20, no. 4, -19, April, 1987.
[4] P. P. Lemberger, Managing the Complexity of Information Systems: The Value of Simplicity, New York, NY: Wiley & Sons, 2011.
[5] Yin, Y., Yan, L., Yan, M., Wang, Y., Yin, L., Su, Y. Innovation of Centralized Management Mode for Government Investment Projects (in Chinese). Tianjin University Press, Tianjin. .0102
[6] Complex Projects. John Wiley & Sons. 2011.Wysocki, R.K. Executive's Guide to Project Management: Organizational Processes and Practices for Supporting

هشتمین کنفرانس ملی
مطالعات مدیریت و اقتصاد در علوم انسانی
8th National Conference on
Management Studies and Economics in the Humanities
www.spconf.ir

The role of goals and decisions with communication software architecture in urban service management

Hossein Rezazadeh[1], Soheila Mohammadi[2]

1. Municipality of Shiraz, h.rezazadeh1023@gmail.com
2. Head of Communication and Cultural Department of Shiraz District 9 Municipality, So.mohamadi1513@gmail.com

Abstract
Software architecture is a set of structures necessary to reason about a system, which includes software elements, the relationships between them, and the characteristics of both. Software architecture provides a solid foundation for creating and developing software, which is often chosen based on the number of users, infrastructure, business goals, and perhaps most importantly, the scalability of that system. Fragastar business automation software is a completely web-based system that can be easily used in small and medium-sized organizations to large organizations with thousands of users, and to fulfill this goal, this system is based on a model of design and production architecture. has been In this article, a software architecture model was presented that helps to build urban information systems. This model is compatible with the principles of urban information systems. Hence, this model helps the information system to be more agile and more scalable and compatible with user needs. In addition, this model should be evaluated through experiments in order to be better used in the principles of information systems. In addition, two questions remain unanswered. The first question is about the effectiveness of using this model for the architecture of "intelligent decision-making systems". The second question is about the integration of software systems architecture for the use of customer relationship management and enterprise resource planning. These problems can determine future research directions.

Keywords: management, office automation, software architecture, information technology, urban services.