



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

1399/8/18



تمرین سوم



مهندسی نرم افزار 2

گروه {#}

اعضاء گروه:

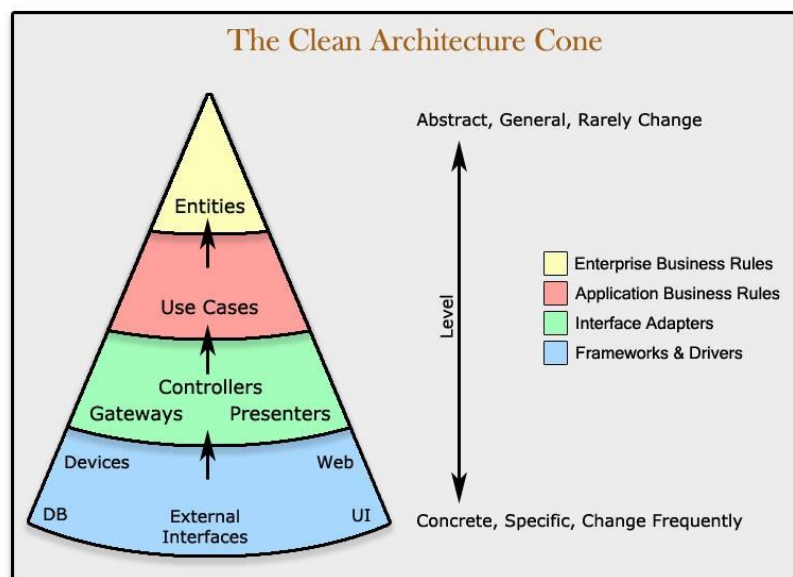
- 1) علی خرمی پور
- 2) امیررضا شیرمست
- 3) علیرضا صدیقی مقدم



1) معماری مبتنی بر مولفه¹ و معماری سرویس گرا² چه تفاوت هایی دارند؟
در معماری سرویس گرا تعدادی سیستم وجود دارد که هر یک سرویسی را در اختیار ما قرار می دهند و مجموع این سیستم ها خود یک سیستم کارآمد می شود.
در معماری مبتنی بر مولفه یک سیستم به چند component تقسیم می شوند و این مولفه ها هدف سیستم را برآورده می کنند.

در معماری سرویس گرا هر یک از مولفه های سطح پایین در قالب یک سیستم سطح بالا در آمده است. به بیان دیگر، در معماری سرویس گرا هر یک از سررویس دهنده ها یک سیستم جدا هستند که به تنهایی میتوانند مورد استفاده قرار بگیرند ولی در معماری مبتنی بر مولفه، اگرچه هر مولفه یک استقلال و حدودا نسبی دارد ولی با تمام اینها در داخل یک سیستم مفهوم پیدا می کنند.

2) در مورد “معماری تمیز”³ که توسط Robert C. Martin مطرح شده است، مطالعه کنید.⁴
الف) مهم ترین اصول و قواعد این معماری را معرفی کنید.



¹ Component-Based Architecture

² Service-Oriented Architecture

³ Clean Architecture

⁴ برای مطالعه بیشتر: <https://blog.cleancoder.com> و <https://cutt.ly/lgJsQTE>



در لایه web تعاملات سیستم با سایر سیستم ها، کتابخانه ها، چهارچوب ها و کاربران شکل می گیرد. در لایه بعد عملا یک ارتباط بین لایه web و لایه ها Logic برقرار می شود. در نهایت در لایه های درونی که مربوط به منطق برنامه است دو لایه در نظر گرفته شده است که یکی به جنبه های سطح بالای منطق و دیگری به جنبه های کاربردی آن اشاره دارد.

ب) کاربرد اصل Dependency Inversion (از اصول SOLID) در این معماری چیست؟
چون معماری لایه ایست پس می توان بخش های logic برنامه را جدا از interface, database, ... توسعه داد. زیرا هر لایه تنها از لایه ی زیرین خود سرویس می گیرد و این سرویس گرفتن با یک interface ساده انجام می شود و جزئیات هر لایه به لایه ی بالایی مربوط نیست.
ج) آیا “معماری تمیز” در معماری مبتنی بر microservice ها نیز کاربرد دارد؟ مختصرا بحث کنید.
بله.

زیرا هر لایه صرفا با یک interface به لایه پایین دسترسی دارد و از جزئیات آن یا آگاه نیست یا کاری ندارد.
پس میتوان یک لایه را در قالب تعدادی میکروسرویس در آورد و این میکروسرویس ها را در سرور های مختلف اجرا کرد و خروجی را دریافت کرد و به لایه بالایی داد.



3) اصول SOLID و YAGNI را در نظر بگیرید:

الف) آن ها را با یکدیگر مقایسه کنید.

در SOLID، بیشتر به چگونگی اضافه کردن Feature ها و در YAGNI بیشتر به زمان اضافه کردن آن ها پرداخته می شود؛ هم چنین، در SOLID، می دانیم (یا حداقل می توانیم تخمین بزنیم) که چه تغییراتی در کد رخ خواهند داد ولی در YAGNI، فرض شده که تغییرات غیرقابل پیش بینی هستند.

ب) آیا این دو رویکرد با یکدیگر در تناقض هستند؟

با توجه به YAGNI، نباید چیزهایی که لازم نداریم را به سیستم اضافه کنیم مگر اینکه حتما به آن احتیاج داشته باشیم ولی SOLID، اشاره به چگونگی تعریف کلاس ها و ارتباط آن ها با یکدیگر است. این دو رویکرد در تناقض با یکدیگر نیستند و می توانند در کنار یکدیگر استفاده شوند.

ج) تجربه یا توصیه خود در استفاده از این دو رویکرد را مختصرا شرح دهید.

استفاده از SOLID می تواند در scalability و انعطاف پذیری سیستم به ما کمک کند. هم چنین، استفاده از YAGNI باعث می شود تا در وقت و هزینه صرفه جویی شده و سیستم بهینه تری داشته باشیم.



4) درباره تحلیل معماری⁵ و روش های آن تحقیق کنید.⁶ (امتیازی)
الف) دو روش را انتخاب کرده و مختصراً معرفی کنید. (ترجیحاً با ذکر دلیل انتخاب)

با توجه به پروژه گروه، روش هایی که بیشتر به رفتار سیستم و نگهداری از آن توجه دارند مد نظر هستند.

1-Scenario-Based Architecture Analysis Method (SAAM)

هدف این روش انطباق فرضیات هنگام تعریف سیستم و خود سیستم است. همچنین، در این روش به ارزیابی ریسک هایی که معماری انتخابی ممکن است داشته باشد پرداخته می شود.
با در نظر گرفتن سناریوهای مختلف، می توان مشخصات معماری سیستم را استخراج کرد. به همین دلیل، می تواند برای پروژه گروه مناسب باشد؛ چرا که کلیت پروژه نیز بر اساس سناریوها تعریف شده است.

2-Architecture Level Prediction of Software Maintenance (ALPSM)

در این روش میزان maintainability یک سیستم نرم افزاری بر اساس تاثیر سناریوها در سطح معماری نرم افزار، اندازه گیری می شود. تنها عضو مورد نیاز برای این روش تحلیل معماری، طراح معماری است.
این روش هم با توجه به اینکه بر اساس سناریوها تعریف شده است، می تواند برای پروژه گروه مفید باشد.

⁵ Architecture Analysis

⁶ به قسمت “مطالعه بیشتر” در مدل مراجعه کنید.



ب) پس از انتخاب یا تعریف یک مجموعه معیار ارزیابی، این دو روش را با یکدیگر مقایسه کنید.
می توان روش های تحلیل را از طریق معیارهای زیر با یکدیگر مقایسه کرد:

۱- هدف خاص

۲- روش ارزیابی

۳- ذی النفعان

۴- توضیح معماری

۵- فعالیت ها

۶- قابلیت استفاده از داده های قبلی

۷- اعتبارسنجی روش

۱- هدف خاص

در روش SAAM هدف پیدا کردن نقاطی که ممکن است دچار مشکل شوند (مانند تناقض نیازمندی ها یا توضیح ناکامل طراحی). در ALPSM با استفاده از اندازه تغییرات به عنوان معیاری برای تطبیق سیستم درون یک سناریو میزان maintainability اندازه گیری می شود.

۲- روش ارزیابی

SAAM: بررسی می شود که با در نظر گرفتن هر سناریو آیا نیاز است که آن تغییر کند یا خیر؛ سناریوهایی که احتیاجی به تغییر ندارند مستقیم (direct) و آنهایی که احتیاج به تغییر دارند غیرمستقیم (indirect) نام دارند.

ALPSM: زیرمجموعه ای از سناریوها که منتج به تغییر در سیستم می شوند را استخراج کرده و براساس آن ها تغییراتی که لازم است اعمال کنیم معلوم می شوند.



۳- ذی النفعان

SAAM: تقریباً تمامی ذی النفعان در این روش شرکت دارند؛ چرا که از روش‌های مختلفی به معماری می‌پردازد.

ALPSM: تنها طراح در فعالیتهای این روش حضور دارد.

۴- توضیح معماری

SAAM: به آخرین نسخه معماری قبل از طراحی کامل اعمال می‌شود و توضیح معماری باید برای تمامی ذی النفعان قابل فهم باشد.

ALPSM: این روش به آخرین نسخه معماری اعمال می‌شود.

۵- فعالیت‌ها

SAAM: این روش یک سناریو را بر اساس این که کدام یک از اجزای معماری را تحت تأثیر قرار می‌دهد، ارزیابی می‌کند.

ALPSM: پس از دسته‌بندی taskهای لازم برای نگهداری سیستم و ساختن سناریوها، به هر کدام از آنها وزنی متناسب اختصاص داده و بر اساس آنها میزان کار لازم برای نگهداری سیستم را تخمین می‌زنیم.

۶- قابلیت استفاده از داده‌های قبلی

SAAM: ندارد.

ALPSM: می‌توان از داده‌هایی که در کاربردهای مشابه یا در نسخه‌های قبلی برنامه به دست آمده‌اند استفاده کرد.

۷- اعتبارسنجی روش



تمرین سوم

مهندسی نرم افزار 2

دکتر طارمی راد



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

SAAM: این روش بارها استفاده شده و روش کاملی است.

ALPSM: در مقایسه با SAAM، از اعتبار بالایی برخوردار نیست.