**به نام خدا**



پاسخنامه آزمون «معماری نرم افزار» \_ 1115280

**دانشجو: مریم سادات موردگر- 990190426 استاد : دکتر سید علی رضوی ابراهیمی**

C:\Users\masood\Desktop\Untitled.png**رشته و مقطع تحصیلی: ارشد مهندسی نرم افزار 19/10/1400**

|  |
| --- |
| (پاسخ آزمون در گیت هاب به آدرس نیز می باشد) |

**سؤالات میان ترم**

**١- معماری نرم افزار چیست و نحوه ارزیابی یک معماری طرح شده را شرح دهید؟**

**(راهنمایی: یکی از روش های کیفی ارزیابی معماری را به زبان ساده شرح دهید.)**

معماري نرم‌افزار يك برنامه يا يك سيستم محاسباتي ساختار يا ساختارهايي از سيستم است كه عناصر نرم‌افزاري، خصوصيات قابل مشاهده از بيرون آن عناصر، و ارتباطات بين آنها را شامل مي‌شود.

معماری نرم افزار در مورد ساختار کلی سیستم است. معماری یک سیستم اجزای اصلی آن و روابط آن ها و نحوه تعامل آن ها با یکدیگر را توصیف می کند. نقش معماري نقش اساسي در ايجاد و نگهداري نر م افزار است.

use case ها و موارد كاربردي در معماري مورد بحث هستند. در معماري صفت، كيفيت، چگونگي انجام كار، سختي يا آساني و نيازمندي هاي عملياتي ( use case ها) و غير عملياتي و نيازمندي هاي سيستمي مطرح است. معماری نرم افزار، ابزار طراحي صفات كيفي است . معماری نرم افزار می تواند تاثیر زیادی بر روی صفات کیفی داشته باشد. نرم افزاری که مجهز به فضای ذخیره سازی مناسب نیست؛ اتلاف انرژی بالایی دارد. معماري نرم افزار ساختار ها و روابط آنها را توجه مي كند و مجموعه اي از ساختارهايي است كه لازم است در طراحي و نگهداري و ... نرم افزار مورد توجه داشته باشيم . مثلا ساختاري كه با پايگاه داده نرم افزار مورد در ارتباط است، رابطه بين آن ها و خصوصيات آن ها را بحث مي كند.

به زبان ساده تر می توان گفت نیازمندی ها در حالت کلی به دو دسته نیازمندی های عملیاتی و نیازمندی های غیرعملیاتی تقسیم می‌شوند. نیازمندی های عملیاتی، عبارتست از توانایی سیستم در انجام کاری که برای آن ایجاد شده است. نیازمندی های غیرعملیاتی که تحت عنوان مشخصه‌های کیفی از آنها یاد می‌شود، هر آنچه که غیر از نیازمندی های عملیاتی سیستم باشد، در این دسته قرار می‌گیرند. مانند کارایی، امنیت، هزینه ساخت و… . کیفیت نرم‌افزار به صورت مستقیم با توانایی یک سیستم در قبال نحوه انجام نیازمندی‌های عملیاتی و غیرعملیاتی آن در ارتباط می‌باشد. یک سیستم می‌تواند شامل مشخصه‌های زیادی همچون کارایی، قابلیت نگهداری، امنیت و… باشد. کیفیت هر یک از مشخصه‌های موجود، بر کیفیت کل سیستم تاثیر دارد. یعنی کیفیت کل سیستم تابعی از کیفیت تک‌تک این مشخصه‌ها می‌باشد. لازم به ذکر است که همیشه کیفیت این مشخصه‌ها قابل اندازه‌گیری نیست.

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.

براي ارزيابي يك معماري مي شود از روش تجزيه و تحليل تجارت معماری (ATAM) استفاده كرد كه به شرح زير مي باشد:

روش تجزيه و تحليل تجارت معماري (ATAM) بيش از يك دهه است كه براي ارزيابي معماري نرم افزار در حوزه هايي از خودرو تا مالي تا دفاع ي مورد استفاده قرار مي گيرد. ATAM به گونه اي طراحي شده است كه ارزيابي كنندگان نيازي به آشنايي با معماري يا اهداف تجاري آن ندارند يا سيستم هنوز نيازي به ساخت ندارد و ممكن است تعداد زيادي از ذينفعان وجود داشته باشند. اين روش ٩ مرحله دارد كه به ترتيب زير مي باشد:

١ - ارائه ATAM - ارائه مفهوم ATAM به ذينفعان، و پاسخ به هر گونه سوال در مورد روند كار.

٢ -ارائه محرك هاي تجاري - همه افراد در اين فرآيند، محرك هاي تجاري سيستم مورد نظر را ارائه و ارزيابي می كنند.

٣ -ارائه معماري - معمار با "سطح مناسبي از جزئيات" معماري سطح بالا را به تيم ارائه مي دهد .

٤ - شناسايي رويكرد هاي معماري - روش هاي مختلف معماري براي سيستم توسط تيم ارائه شده و در مورد آنها بحث مي شود.

٥ - توليد درخت سودمندي صفت كيفي - در اين مرحله الزامات اصلي تجاري و فن ي سيستم را تعريف مي شود و آنها را با ويژگي مناسب معماري ترسيم مي كنند. سناريويي نيز براي اين نياز ارائه مي شود.

٦ -تحليل رويكردهاي معمار ي - تجزيه و تحليل هر سناريو ، رتبه بندي آنها بر اساس اولويت و سپس ارزيابي معماري در برابر هر سناريو.

٧ -طوفان فكري و اولويت بندي سناريو ها - در ميان گروه ذ ينفعان بزرگتري سناريوهاي موجود ارائه مي شود و گسترش پيدا مي كند.

٨ - تجزيه و تحليل رويكردهاي معماري – مرحله ٦ با اطلاعات اضافه شده گروه بزرگتر ذينفعان دوباره انجام مي شود.

٩ - ارائه نتايج – كليه اسناد و مدارك در اختيار ذينفعان قرار مي گيرد.

اين مراحل در دو فاز از هم تفكيك مي شوند: فاز ١ شامل مراحل ١ تا ٦ است و پس از اين فاز ، وضعيت و زمينه پروژه ، نيازهاي معماري محرك و وضعيت اسناد معماري مشخص مي شود. فاز ٢ كه شامل مراحل ٧ تا ٩ است و ارزيابي در آن به پايان مي رسد.

سؤالات پایان ترم

**1- مدل یک صفت کیفی را بطور کامل شرح دهید.**

قابلیت دسترسی Availability

خصوصيت كيفي قابليت دسترسي در ارتباط با شكست سيستم و پيامدهاي همراه با آن است. شكست سيستم زماني رخ مي دهد كه سيستم سرويس هاي پيش فرض خود را به طور مناسب ارائه نكند. همچنين وقتي شكست روي مي دهد نتايج آن توسط كاربران سيستم قابل مشاهده است. حوزه هاي مرتبط با اين خصوصيت كيفي عبارتند از :

* چگونه شكست سيستم تشخيص داده مي شود.
* چگونه شكست سيستم رخ مي دهد.
* زماني كه يك شكست رخ مي دهد چه اتفاقي مي افتد.
* تا چه مدت سيستم در حالت عملياتي نيست (در صورت شكست).
* چگونه ميتوان جلوي شكست را گرفت.
* زماني كه يك شكست رخ مي دهد چه نوع اخطارهاي لازم است.

بايد توجه كرد كه ميان دو مفهوم شكست و نقص تفاوت وجود دارد. يك نقص در صورت اينكه مورد بررسي قرار نگيرد و برطرف نشود منجر به شكست مي شود. يك شكست توسط كاربر سيستم قابل مشاهده است در حالي كه نقص توسط كاربر قابل مشاهده نيست. زماني كه يك نقص توسط كاربر قابل مشاهده مي شود در واقع آن تبديل به شكست شده است. به عنوان مثال، يك خرابي مي تواند انتخاب الگوريتم اشتباه براي يك محاسبه باشد كه نهايتا نتيجه اشتباه حاصل شده توسط الگوريتم اشتباه منجر به شكست مي شود.

يكي از مهمترين مفاهيم در زمانيكه سيستم دچار شكست مي شود مدت زمان لازم براي تعمير است. از آنجايي كه شكست سيستم توسط كاربران قابل مشاهده است بنابراين زمان برطرف كردن شكست برابر است با زمان بين مشاهده شدن شكست توسط كاربر تا زماني كه ديگر شكست قابل مشاهده نيست (سيستم تعمير مي شود).

تفاوت بين شكست و نقص ارائه كننده مفهوم راهبردهاي تعمير خودكار است. معني راهبرد تعمير خودكار اينكه در زمان اجراي سيستم اگر نقصي روي دهد، سيستم بدون آنكه شكست را نشان دهد (دچار شكست شود) آن نقص را اصلاح كند. بنابراين با اين راهبرد، ديگر در سيستم شكست وجود نخواهد داشت.

در دسترس بودن يك سيستم برابر است با احتمال آنكه سيستم در زمان مورد نياز قابل استفاده باشد.

سناريوهاي عمومي قابليت دسترسي

**منبع تحريك:** بين تحريك هاي داخلي و خارجي مشخص كننده شكست و نقص، تفاوت وجود دارد زيرا پاسخ سيستم به هر يك از اين تحري كها ميتواند متفاوت باشد.

**محرك:** يك نقص كه ميتواند محرك در نظر گرفته شود در يكي از دسته هاي زير قرار مي گيرد:

Omission : يك مولفه شكست خورده كه نمي تواند به ورودي پاسخ دهد.

Crash : مولفه مكررا دچار نقص از نوع omission مي شود.

Timing : يك مولفه پاسخ ميدهد ولي پاسخ زود يا دير توليد مي شود.

Response : يك مولفه پاسخ مي دهد ولي پاسخ آن صحيح نيست.

**فرآورده:** فرآورده مشخص كننده منبعي است (مانند پردازشگر، كانال ارتباطي، فرآيند يا انباره) كه نيازدارد تا حد بالايي قابل دسترس باشد.

**محيط:** وضعيت سيستم وقتي نقص يا شكست اتفاق مي افتد ممكن است پاسخ سيستم را تحت تاثير قرار دهد. به عنوان مثال اگر سيستم از قبل دچار چندين نقص شده و در حال اجرا در يك حالت غيرطبيعي است احتمالا بهتر است كه سيستم را كاملا خاموش كرد اما اگر اولين نقص مشاهده شده باشد مي توان بعضي ملاحظات را در مورد زمان پاسخ يا نحوه عملكرد سيستم، در نظر گرفت.

**پاسخ:** براي يك شكست واكنش هاي متفاوتي مي تواند وجود داشته باشد. اين واكنش ها مي نتوا ند ثبت كردن شكست، هشدار دادن به كاربر يا ديگر سيستم ها، خاموش كردن سيستم هاي خارجي و يا دسترس پذير نبودن در طول تعمير در نظر گرفته شوند.

**معيار پاسخ:** معيار پاسخ مي تواند درصد قابليت دسترسي، مدت زمان تعمير و زمان ها يا دوره هايي كه سيستم در آنها قابل دسترس است، در نظر گرفته شود.

**2- یک صفت کیفی تخیلی را طراحی و جدول سناریوهای عمومی آن و یک نمونه سناریوی معین برای آن تعریف نمائید. (مثلا: «دوست داشتنی» یا «دانشجویی!» یا هر صفت کیفی تخیلی و ابتکاری دیگر)**

صفت کیفی مُسِن پسند

در دنیای امروز که برای انجام بسیاری از کارهای روزمره نیاز به استفاده از اپلیکیشن ها و نرم افزارهای مختلف است، لازم است برخی از نرم افزارها به گونه ای باشند که افراد مُسِن و کم سواد جامعه نیز بتوانند از آن ها استفاده کنند. مانند نرم افزارهای بانک یا تهیه بلیط و یا بیمه ها و ...

این صفت می خواهد که یک نرم افزار خیلی ساده و واضح باشد تا همه افراد بتوانند از آن استفاده کنند. از منوهای کمتر استفاده شود در نتیجه کارهای قابل اجرای کمتری نیز باید در برنامه گنجانده شود و نرم افزار کارهای متفاوتی را انجام ندهد.

مثلا می توان از نرم افزار موجود بانک ها استفاده کرد و آن ها را برای این هدف بهینه کرد تا در اختیار کاربران این گروه قرار بگیرد.

سناریوهای عمومی مُسِن پسند:

منبع: کاربر نهایی (افراد مُسِن و کم سواد)

محرک: یادگیری ویژگی های سیستم – احساس راحتی هنگام کار با سیستم

فرآورده: نرم افزار مورد استفاده

محیط: رابط کاربری

پاسخ: حمايت ازيادگيري ويژگي هاي سيستم مثل واسط آشنا به كاربر و واسط كاربري كه در يك زمينه ناآشنا قابل استفاده است. حمايت از استفاده كارا از سيستم مثل استفاده مجدد از داده ها و فرمان هاي از قبل وارد شده و پرهیز از فیلدهای اضافه و گیج کننده، استفاده از خط واضح و خوانا برای نوشته ها و گرفتن تایید نهایی برای انجام عملیات

معیار پاسخ: سرعت بارگذاری، سرعت واکنش به درخواست، رضايت كاربر مُسِن

مثال نرم افزار بانک عملیات انتقال موجودی

 سناریوی عمومی

**در صورت وارد نکردن شماره کارت مقصد یا اشتباه بودن آن خطا می دهد.**

**صفت مُسِن پسند**

**قابلیت ذخیره شماره کارت کاربر**

**قابلیت ذخیره شماره کارت مقصد**

**قابلیت جایگذاری خودکار رمز دوم پس از درخواست آن**

**کاربر مُسِن**

سناریوی معین:

حالت خاص: اگر رمز اشتباه وارد شود بدون پاک شدن اطلاعات وارد شده دیگر تا سه بار اجازه ورود مجدد رمز را بدهد.

**رمز اشتباه است و بدون پاک کردن دیگر فیلدهای وارد شده اجازه ورود مجدد رمز را می دهد.**

**کاربر مُسِن**