

به نام خدا

امتحان نهایی درس : معماری نرم افزار - رشته و مقطع تحصیلی : ارشد مهندسی نرم افزار - دی ماه ۱۴۰۰

دانشجو : سمیه کرباسی کد دانشجویی : ۹۹۰۱۹۰۳۴۵

سوالات را با توجه به دو منبع و تدریس استاد و اطلاعات خودم نوشته ام . سعی کرده ام طبق نظر استاد پاسخ ها کوتاه و مفید باشد . طراحی شکل ها کاملا ابتکاری و توسط اینجانب است. ضمناً پاسخ ها ها در گیت هاب نیز قرار داده ام

http://github.com/skarbasy/PNU_4001_AR

در کلاس درس هم چنین فعال بودم و به سوالات درسی با میکروفن پاسخگو بودم.

سوالات میان ترم (۸ نمره – الزامی)

۱- معماری نرم افزار چیست و نحوه ارزیابی یک معماری طرح شده را شرح دهید؟(۸نمره)

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

برای معماری نرم افزار، تعریفی که به طور عمومی پذیرفته شده باشد وجود ندارد . تعاریفی مختلفی وجود دارد که در مواردی مفاهیم مشترکی دارند . معماری نرم افزار یک برنامه یا سیستم کامپیوتری ، ساختار یا ساختارهایی از سیستم می باشد که در برگیرنده اجزا ، صفات قابل مشاهده آن اجزا و ارتباط بین آنها باشد.

در معماری نرم افزار ساختار کلی یک سیستم مورد بحث است و ساختار بین اجزا آن سیستم مطرح است. نقش معماری نقش اساسی در ایجاد و نگهداری نرم افزار است . use case ها و موارد کاربردی در معماری مورد بحث هستند. در معماری صفت، کیفیت، چگونگی انجام کار – سختی یا آسانی و نیازمندی های عملیاتی (use case ها) و غیر عملیاتی و نیازمندی های سیستمی مطرح است. ابزار طراحی صفات کیفی معماری نرم افزار 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. معماری نرم افزار ساختار ها و روابط آنها را توجه می کند و مجموعه ای از ساختارهایی است که لازم است در طراحی و نگهداری و ... نرم افزار مورد توجه داشته باشیم . مثلاً ساختاری که با پایگاه داده نرم افزار مورد در ارتباط است . رابطه بین آن ها و خصوصیات آن ها را بحث می کند.

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 پرهزینه است. گزینه های معماری نرم افزار شامل گزینه های ساختاری خاص از امکانات موجود در طراحی نرم افزار است.

معماری، اجزای نرم افزار را تعریف می کند. در مواردی از جزئیاتی از اجزا که در نحوه استفاده و ارتباط با اجزای دیگر کاربردی ندارند صرف نظر می شود. هر سیستم نرم افزار شامل چندین ساختار می باشد و هیچ یک از این ساختارها به تنهایی معماری نرم افزار نمی باشد. بلکه این ساختارها در کنار یکدیگر معماری نرم افزار را تشکیل می دهند .

معماری نرم افزار مجموعه ای از تصمیمات مهم درباره ساختار سیستم نرم افزاری ، انتخاب اجزا ساختاری و ارتباطات بین آن ها و همچنین مشخص نمودن انواع همکاری این اجزا با یکدیگر است . وقتی این اجزا در کنار یکدیگر سیستم بزرگی را تشکیل دهند معماری نرم افزار ایجاد می شود.

هر سیستم نرم افزاری دارای یک معماری می باشد زیرا هر سیستم نرم افزاری دارای اجزایی است که به من در ارتباطند. معماری نرم افزار باید قابل ارزیابی باشد تا بتوان از روی آن تشخیص داد سیستم مورد نظر بر پایه معماری انتخاب شده نیازهای خود را برآورده خواهد کرد یا خیر

ارزیابی معماری چابک می تواند بخشی از فرایند چابک باشد . توسط متد ارزیابی موازنه های معماری (ATAM) درست شده است . ATAM سعی در تحلیل همه یا بیشتر یک معماری ندارد. این روش بر روی مجموعه ای از صفات کیفی متمرکز شده که مهمترین (اما نه همه ی) دغدغه ذینفعان را دارد. به منظور بازخورد و تحلیل سریعتر می توان یک ارزیابی سبک وزن از معماری انجام داد. این چنین ارزیابی ها را می توان برای مهیا سازی ورودی با ارزش در باز تولید و باز طراحی انجام داد. یک مثال از معماری چابک WebArrow : سیستم کنفرانس بر مبنای وب

برای برآورده کردن نیازهای ذینفعان ، معمار و توسعه دهندگان باید فکر و عمل را به صورتی متفاوت و همزمان انجام دهند :

بالا به پایین : طراحی و تحلیل ساختارهای معماری برای برآورده کردن نیازمندی های کیفی و موازنه ها

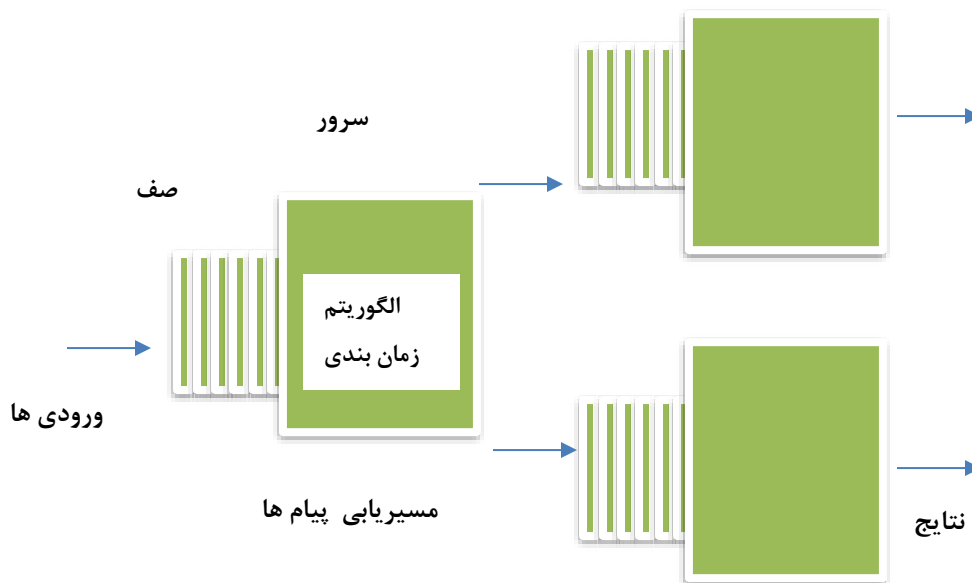
پایین به بالا : تحلیل گسترده وسیعی از قید و بندهای مربوط به پیاده سازی و محیط و ارائه ی راه حل برای آن ها

سوالات پایان ترم (۱۲ نمره-الزامی)

۱- مدل یک صفت کیفی را به طور کامل شرح دهید؟ (۲نمره)

مدل کارایی (Performance):

ابتدا شکل آن را رسم می کنم :



پارامترهای تاثیر گذار بر کارایی Performance

سرعت بخشیدن اتفاقات- انتخاب قاعده ی صف بندی یعنی از چه رابطه ای برای صف بندی ورودی ها استفاده می شود .-انتخاب الگوریتم زمان بندی- زمان پاسخ به رویدادها- توپولوژی شبکه- پهنای باند شبکه- الگوریتم مسیریابی

مزایا و معایب مدل کارایی Performance

هزینه پارامترهایی که قبلا تعیین کردیم باید حساب کنیم . - سود را می توان به طور دقیقی حساب کرد - هر چه قدر پارامترها را دقیق تر تخمین بزنیم محاسبه راحت تر است . در جایی که کارایی اهمیت نداشته باشد نیاز به مدل سازی ندارد.

همچنین موارد دیگر را نیز می توان مورد بررسی قرار داد مانند :

Modifiability

Security

Testability

Usability

۲- یک صفت کیفی تخیلی را طراحی و جدول سناریوهای عمومی و یک نمونه سناریوی معین تعریف کنید؟ (۱۰ نمره)

صفت کیفی دلچسب : خصوصیات یک سیستم فرضی مثل گلستان باید دلچسب باشد یعنی مشکلات رایج را نداشته باشد ، سرعت بالایی داشته باشد . پشتیبانی های لازم را داشته باشند

ما از یک قالب مشترک برای مستند سازی صفات به عنوان سناریو استفاده می کنیم .

باز نمایش صفت کیفی دلچسب شامل بخش های زیر است :

محرك- منبع تحريك- پاسخ - اندازه گیری پاسخ- محیط-محصول

Use case ها عملیاتی بودن سیستم را مدل می کنند و این صفت دلچسب میزان خوب بودن نحوه انجام عملیات را مشخص می کند. یعنی چگونگی انجام کار و میزان دقت و کیفیت را نشان می دهد .

چون ما مهندس نرم افزار هستیم باید بتوانیم این صفت را اندازه گیری کنیم.

باید سناریو های عمومی صفات کیفی (یا فقط سناریوی عمومی) که مستقل بوده و سناریو های معین صفات کیفی (با سناریوی معین) که با شرایط خاص سیستم مواجه باشد.

سناریو عمومی :

سناریوهایی که مستقل از نوع سیستم هست و از این سناریوها برای مشخص کردن بخش های کلی یک ویژگی کلی استفاده می شود.

موارد مربوط به جداول عمومی که ۶ مورد هستند و همچنین آیتم های جدید و مشابه را در نظر بگیریم . البته توجه داشته باشید که آیتم های جداول عمومی حالت نرمال و پیش فرض دارند .

بر اساس این آیتم ها سناریو ها و تاکتیک و چک لیست مناسب را تشکیل می دهیم

Modifiability

Security

Testability

Usability

availability

Performance

سناریو ویژگی کیفی از بخشهای زیر تشکیل شده :

محرك : شرایطی است که وقتی رخ دهد ، سیستم نرم افزاری باید در قبال آن عملی شود

محیط: محیطی که محرك در آن رخ می دهد . بسته به شرایط سیستم محیط متفاوت است که در اینجا ما سناریوهای عمومی و خاص را در نظر گرفته شدیم .

محصول نرم افزاری : که در اینجا سیستم نرم افزاری گلستان دانشگاه پیام نور است .

ارزیابی معماری نرم افزار یک عمل مهم در مهندسی نرم افزار جهت توسعه کیفیت است . ارزیابی در واقع هزینه های توسعه نرم افزار را بشکل مفیدی کاهش می دهد . روش های ارزیابی معماری در دسته بیندیی های مختلفی قرار می گیرد. یکی از دسته بندی های مختلفی قرار می گیرد . یک از دسته بندی های مهم روش های مبتنی بر سناریو است .

ویژگی هایی کیفی که در صفت دلچسب مورد نظر است :

زمان دستیابی به بازار

هزینه و سود

زمان انجام پروژه و ماندگاری پروژه

بازار هدف و جا افتادن بین اساتید و دانشجویان

یکپارچگی مفهومی

صحیح بودن و کامل بودن

تاکتیک های معماری و چک لیست ها به نحو دقیق برای صفت کیفی دلچسب طراحی می شود .

برای دستیابی به ویژگی کیفی دلچسب ، باید تصمیماتی مربوط به نحوه طراحی معماری اتخاذ نمود.

ابتدا پیشبرنده های معماری از مجموعه سناریوهای ویژگی کیفی دلچسب و نیازهای کارکردی انتخاب می شود . الگوهای معماری که

برآورده کننده معماری مورد نظر است انتخاب می شود .

مشاهده نمره :



در صورت نمره
پایین خطا دهد

صفت دلچسب

دیدن نمرات

بازشدن راحت سایت

امکان تغییر رمز و بازبایی رمز راحت

باز شدن در هر پلت فرم



کاربر

سناریو معین :

در بالا و در قسمت سناریو عمومی موارد لازم بحث شد اینجا موضوع را بیشتر شرح می دهیم . معیارهایی علاوه بر معیارهای فقط باید

مورد توجه باشد :

قابلیت انطباق با فرهنگ های مختلف

یکپارچگی داده ای

قابلیت نگهداری بالا

قابلیت سلامت (safety)

قابلیت مدیریت

ویژگی های کیفی سیستم ، نیازهای غیر کارکردی می باشند که بر روی کارکردهای سیستم اثرگذار باشند

مثلا اگر شهریه پرداخت نشده باشد . به دانشجو در بازه معین فرصت پرداخت راحت و بدون خطا بدهد.

خطا شهریه می دهد و به راحتی
امکان پرداخت باشد .



در بالا سناریو عمومی و معین برای صفت خیالی دلچسب طراحی شده است .

دست یابی به صفات کیفی با استفاده از تکنیک ها :

مجموعه ای از تصمیمات طراحی اولیه وجود دارد که می توان از آنها برای دست یابی به صفت کیفی دست یافت

برای خاصیت دلچسب طراحی اولیه باید حتی الامکان بدون عیب باشد و روی تست زیاد کار شود .

می توان دانشجویها و یا اساتید که از سیستم فعلی رضایت کامل ندارند برای تست سیستم و رسیدن به صفت کیفی دلچسب استفاده کرد.

تاکتیک ها مانند الگوهای طراحی، تکنیک هایی هستند که طراحان و معماران سیستم از سال ها قبل از آن استفاده می کردند. دقیقا مورد جدیدی وجود ندارد

فقط باید ایرادات حل شده و تمام نقطه ضعف هایی که اساتید و دانشجویان مطرح می کنند برطرف شده و خصوصیت کیفی دلچسب محقق شود

ماژول های مورد نظر ایجاد شده و کارکردهای لازم برای هر ماژول با توجه به کاربرد آن ها اختصاص داده می شوند.

برای زیر ماژول ها واسط هایی انتخاب می شوند و مستند سازی و چک لیست تهیه می شود

در این طراحی صفت کیفی دلچسب از ABAS استفاده می کنم. این یک معماری همزمان است معماری لوله و فیلتر موازی است که سرعت و دقت را بالا برده و در هزینه صرفه جویی می کند .

مشخص نمودن راهبردهای معماری و اثر هر راهبرد بر پاسخ سناریوها

محاسبه سودمندی راهبرد معماری

تخصیص وظایف - مدل هماهنگی- مدل داده - مدیریت منابع - نگاشت بین عناصر معماری- تصمیمات زمان انقیاد و انتخاب تکنولوژی بسیار مهم و حیاتی است و نیاز به دقت دارد .

برای انتخاب تکنولوژی اول باید بررسی کنیم چه تکنولوژی هایی وجود دارد و بعد کمبودها را شناسایی و اقدام به طراحی و پیاده سازی و تست کرد و در نهایت سیستم را راه اندازی کرد

ضمنا پشتیبانی و نگهداری سیستم جهت تحقق صفت کیفی دلچسب بسیار مهم است.