انواع نيازمندي هاي نرم افزار را نام ببريد ؟

1-نیاز مندی های وظیفه مندی )Requirements Functional) :وظايفي كه سيستم بايد انجام بدهد را توصيف مينمايد .

2-نیاز مندی های غیروظیفه مندی: (Requirements Functional-Non ) خصوصیات کیفی نرم افزار که نحوه انجام وظایف سیستم را توصیف می‌نماید .

چه وقت نیاز های غیروظیفه مندی به نرم افزار افزوده می شوند؟

یک روش قدیمی

ابتدا نرم افزاری می سازیم که نیازهای وظیفه مندی را برآورده سازد و سپس خصوصیات غیر وظیفه مندی را به آن اضافه یا کم می کنیم. θ از دست دادن منابع و زمان θ کیفیت پائین نرم افزار بخاطر عدم تاثیر این دو نیازمندی بریکدیگر

روش جدید

در توسعه سیستم های نرم افزاری این دو را همزمان باید پیش برد.

نسبت وظیفه مندی و خصوصیات کیفی چگونه است ؟

وظیفه مندی و خصوصیات کیفی متعامدند، یعنی در رابطه تنگاتنگ با یکدیگر هستند .

**آیا هر سطح از وظیفه مندی قابل دستیابی با هر سطح از خصوصیات کیفی است ؟**

اما هر سطح از وظیفه مندی )عملکرد( قابل دستیابی با هر سطح از خصوصیات کیفی نیست

مثالً، مرتب کردن )sort )بانک های اطالعاتی حجیم ممکن است با کارآیی باال انجام نشود )از این رو: معمار، یک سطح مناسبی از کیفیت را برای هر خصوصیت بر می گزیند(

**دلایل طراحی مجدد سیستم ها چیست؟**

طراحی مجدد سیستم ها همیشه به دلیل پوشش نیازهای وظیفه مندی نیست،بلکه بیشتر به دلیل مشکالتی در زمینه نگهداشت پذیری، مقیاس پذیری، سرعت پاسخ گویی و امنیت آن ها است.

**معماری چگونه ابزار است ؟**

معماری ابزاری است برای دستیابی به خصوصیات کیفی )این کار با ساختاردهی به وظیفه مندی در قالب مولفه ها/ماژول ها انجام می شود(.

**دستیابی به خصوصیات کیفی مرتبط با معماری مربوط به چه مراحلی است ؟**

دستیابی به خصوصیات کیفی مرتبط با معماری )نیازهای غیروظیفه مندی( مربوط به کلیه مراحل طراحی، پیاده سازی و استقرار سیستم است. یعنی: این نوع خصوصیات کیفی، مربوط به تنها یک مرحله از تولید نرم افزار مثالً طراحی یا پیاده سازی نیست. در واقع، خصوصیات کیفی هم دارای جنبه های معمارانه و هم غیرمعمارانه هستند.

**آیا خصوصیات کیفی مستقل هستند ؟**

خصوصیات کیفی مستقل نیستند و نمی توانند به تنهایی انجام شوند )لزوم ایجاد توازن بین آن ها(

همبستگی مثبت: قابلیت اصلاح در مقابل قابلیت ساخت )در بیشتر موارد(. ν

همبستگی منفی: قابلیت اطمینان در مقابل امنیت ،کارایی در مقابل بقیه خصوصیات کیفی

**انواع خصوصیات کیفی را نام ببرید؟**

1-خصوصیات کیفی سیستمی شامل:

الف) قابلیت دسترسی ب) کارایی ث) امنیت پ)قابلیت جابجایی ج)قابلیت استفاده چ)قابلیت تغییر ح)عملکرد خ)قابلیت استفاده مجدد ر)قابلیت تجمیع پدیری ز) قابلیت آزمون

2-خصوصیات کیفی حرفه(معماری بر آنها تاثیر می‌گذارد) شامل :

الف) زمان ارائه به بازار ب)سود و هزینه ت) طول عمر پیش بینی شده . ع) زمان بندی راه اندازی ف) تجمیع با سیستم های موروثی غ) بازار هدف

3-خصوصیات کیفی معماری (معماری را تحت تاثیر قرار میدهند.شامل:

یکپارچگی مفهومی ، قابلیت ساخت ، صحت و تکامل

**طبقه بندی خصوصیات کیفی برحسب اجرا چیست ؟**

1. قابل مشاهده از طریق اجرا )نرم افزار باید اجرا شود تا مشخص شود چنین خصوصیاتی فراهم شده است یاخیر؟( مانند کارایی و امنیت
2. غیر قابل مشاهده از طریق اجرا )این خصوصیات در زمان اجرا قابل رویت و ارزیابی نیستند(. مانند قابلیت اصالح، قابلیت آزمون و قابلیت استفاده مجدد θ خصوصیات غیر قابلمشاهده برخی اوقات بسیار مهم می شوند. )این خصوصیات نشان می دهد که تحلیل، طراحی و آزمایش سیستم با چه میزان سهولت و راحتی انجام می گیرد (

**مشکلات گدشته خصوصیات کیفی چیست ؟**

1)‌ تعاریف ارائه شده برای یک خصوصیت کیفی، عملیاتی نیست. مثال: امنیت با توجه به کدام جنبه سیستم؟ در مورد خطای سیستم یا نفوذ پذیری سیستم

2) یک جنبه خاص به کدام خصوصیت کیفی تعلق دارد. ν آیا خطای سیستم یکی از جنبه های کارایی، امنیت یا قابلیت استفاده است؟

3) هر خصوصیت کیفی فرهنگ اصطالحات خاص خود را دارد . خصوصیت کارایی شامل مفهوم »رویداد« است که در سیستم رخ می دهد، ν خصوصیت امنیت دربردارنده مفهوم »حمله« است، ν خصوصیت قابلیت دسترسی شامل مفهوم »خرابی« است.

**آیا راه حلی برای مشکلات بالا وجود دارد ؟**

برای مشکالت اول و دوم می توان: ♣ از سناریوهای خصوصیات کیفی استفاده نمود )به عنوان ابزاری برای مشخص کردن خصوصیات کیفی(. θ برای مشکل سوم می توان: ♣ توصیف کوتاهی برای مفهوم هر خصوصیت کیفی ارائه نمود. θ سناریوها می توانند عمومی یا منحصربفرد )برای سیستم خاص( باشند.

سناریوی خصوصیات کیفی چیست و شامل چه مواردی است ؟

هر سناریوی خصوصیات کیفی، یک نیازمندی مرتبط با خصوصیات کیفی است که شامل شش قسمت است:

1. منبع تحریک (stimulus of Source ): موجودیتی که تحریک را ایجاد می کند )مانند: انسان، سیستم کامپیوتری یا هر محرک دیگری که مولد یک تحریک است(.
2. محرک (Stimulus ) : شرط یا شرایطی است که وقتی وارد یک سیستم می شود باید مورد توجه قرار گیرد. )مانند: خطا، یا پاسخ یک مولفه با یک مقدار نادرست(. θ
3. محیط )Environment :)تحریک در وضعیت ها و شرایط خاص و معینی روی می دهد. )مثالً وقتی یک تحریک رخ می دهد، سیستم ممکن است در حال اجرا با شرایط اضافه بار )overload )باشد(. θ
4. فرآورده )Artifact :)قسمت هایی از سیستم که نیاز است در دسترس باشند، به عبارت دیگر فرآورده موجودیتی است که هدف تحریک است- فرآورده ممکن است کل سیستم یا بخشی از آن باشد )مانند پردازنده، کانال های ارتباطی، فرایند، ذخیره سازی و ...( θ
5. پاسخ )Response:)واکنش مورد نظر سیستم به یک محرک )ثبت خرابی، اطالع رسانی به اپراتور سیستم و غیره( θ
6. اندازه گیری پاسخ )measure Response :)پاسخ باید با استفاده از روش هایی قابل اندازه گیری باشد تا نیازمندی بتواند آزمایش شود

**بخش های اصلی سنارویها خصوصیت کیفی؟**

سناریوهای عمومی )مستقل از سیستم بوده و می توانند در ارتباط با هر نوع سیستمی باشند(

. ♣ سناریوهای عینی )برای سیستم های خاص استفاده می شوند(

**سناریوهای عمومی چه چیزی را فراهم می کنند؟**

سناریوهای عمومی چارچوبی را فراهم می کنند برای تولید تعدادی از سناریوهای مستقل از سیستم.

**برای اینکه سناریوی عمومی برای یک سیستم خاص مفید باشد باید چه کار کرد ؟**

برای اینکه سناریوی عمومی برای یک سیستم خاص مفید باشد، ♦ باید آنها را خاص سیستم موردنظر ) specific-system (تعریف نمود.

**specific-system نمودن یک سناریوی عمومی، یعنی چه؟**

Specific-system نمودن یک سناریوی عمومی، یعنی ترجمه آن به عبارات و اصطالحات کامالً مرتبط با یک سیستم خاص.

**هر سناریوی عمومی می تواند چه تعداد نسخه سناریوی specific-system داشته باشد؟**

هر سناریوی عمومی می تواند چندین نسخه سناریوی specific-system داشته باشد.

مثال )سناریوی عمومی(: درخواستی برای ایجاد یک تغییر در یک وظیفه مندی دریافت می شود و تغییر باید در یک بازه زمانی مشخص در داخل فرایند اِعمال گردد.

مثال1( یک نسخه خاص سیستم(: درخواستی برای اضافه کردن یک Browser جدید به سیستم مبتنی بر وب دریافت می شود وآن تغییر باید در بازه زمانی دو هفته انجام شود.

مثال2( یک نسخه خاص سیستم(: سیستمی که از مرورگر جدید حمایت می کند، باید از یک رسانه جدید نیز حمایت نماید.

**قابلیت دسترسی با چه چیزی مرتبط است و چه زمانی اتفاق میافتد و مهمترین مسئله در زمان این رخداد چیست؟**

قابلیت دسترسی با شکست سیستم )failure system )و پیامدهای آن مرتبط است.

شکست سیستم وقتی اتفاق می افتد که سیستم قادر به ارائه سرویس های پیش فرض خود نیست.

در زمان شکست سیستم، مهمترین مسئله مدت زمان الزم برای تعمیر آن است.

**استراتژی تعمیر اتومات سیستم چیست ؟**

:اگر در زمان اجرای سیستم نقصی روی دهد، سیستم بدون اینکه دچار شکست شود، نقص را اصالح کند )با این راهبرد، شکست در سیستم وجود نخواهد داشت(.

**تفاوت شکست و نقص را توضیح دهید ؟**

نقص می تواند به شکست تبدیل شود اگر اصالح یا پوشش داده نشود. θ

شکست توسط کاربران قابل مشاهده است ولی نقص قابل مشاهده نیست.

θ نقص وقتی قابل مشاهده می گردد که به شکست تبدیل می شود.

**قابلیت دسترسی چیست؟**

احتمال اینکه وقتی به سیستم نیاز داریم، سیستم در حالت عملیاتی باشد.

**قابلیت اصالح دربارة چیست؟.**

قابلیت اصالح دربارة هزینة تغییرات است.

**قابلیت اصلاح چه نگرانی های را به وجود میآورد ؟**

1 -چه چیزی می تواند تغییر کند ؟

تغییر می تواند در هر یک از جنبه های سیستم باشد مانند: ν

1. وظایفی که سیستم انجام می دهد )متداول ترین تغییرات(.
2. سکویی که سیستم روی آن قرار دارد )سخت افزار، سیستم عامل و ...(. ν
3. محیطی که سیستم در آن اجرا می شود )سیستمی که سیستم موردنظر باید با آن ارتباط داشته باشد(. ν
4. خصوصیات کیفی که سیستم ارائه می دهد )کارآیی، قابلیت اطمینان(. ν
5. ظرفیت سیستم )تعداد کاربران پشتیبانی شده، تعداد عملیات همزمان(.

2 -چه زمانی و چه کسی تغییر را انجام می دهد ؟

یک تغییر ممکن است توسط: کاربر، توسعه دهنده یا مدیر سیستم انجام گیرد

تغییرات می تواند در کد برنامه در مواقع زیر اعمال گردد:

1. در زمان کامپایل،
2. در زمان ساخت،
3. در زمان استقرار،
4. در زمان پیکره بندی نصب،
5. یا در طول اجرای سیستم

**کارایی با چه موضوعی در ارتباط است ؟**

کارایی با زمان در ارتباط است.

رویدادهایی (وقفه ها، پیام ها، درخواست های کاربران) در سیستم رخ می دهند و سیستم باید به آنها پاسخ دهد.

وقتی رویدادی اتفاق می افتد چه مدت طول می کشد تا سیستم پاسخ دهد.

**چه چیزی باعث پیچیدگی کارایی میشود ؟**

تعداد منابع رویداد، ترتیب وقوع آنها و الگوهای ورودی (دوره ای مثالً هر 10 میلی ثانیه یا آماری با توجه به یک تابع توزیع احتمال(.

**رویداد ها از چه چیزهایی ممکن است صادر شوند؟**

رویدادها ممکن است از درخواست های کاربران، از سیستم های دیگر یا از داخل سیستم صادر شوند**.**

**ویژگی های اغلب سناریوهای کاربری را توضیح دهید ؟**

1. با درخواست برای یک سرویس در سیستم شروع می شوند.
2. برآورده نمودن درخواست با مصرف منابع همراه است
3. معموالً رویدادها به صورت موازی به کار می روند (همزمان وارد سیستم می شوند(

**انواع الگوهای رسیدن رویداد ها را نام ببرید؟**

دوره ای: اغلب در سیستم های بالدرنگ دیده می شوند.

احتمالی )آماری(: رویدادها بر طبق تابع توزیع احتمالی خاص دریافت می شوند.

منفرد )پراکنده(: الگویی که نمی تواند با دیگر الگوها نمایش داده شود.

**امنیت چه معیاری است ؟**

امنیت معیاری برای توانایی سیستم به منظور:

مقاومت در مقابل استفاده غیرمجاز است در حالیکه سیستم خدمات خود را برای کاربران مجاز ارائه می دهد.

**حمله چیست ؟**

هر تالشی برای شکستن امنیت، حمله نامیده می شود. حمله می تواند برای اهداف زیر باشد:

1. دستیابی غیرمجاز به داده ها یا خدمات،
2. تغییر آنها،
3. از کار انداختن سرویس دهنده برای کاربران مجاز.

**امنیت بیان گر چه صفتی است ؟**

امنیت می تواند به عنوان صفت برای سیستمی بیان شود که فراهم کنندة محرمانگی، تمامیت، عدم انکار، قابلیت دسترسی و ثبت وقایع است

**قابلیت آزمون چیست ؟**

به میزان راحتی آزمون نرم افزار برای نمایش خطاهای آن، اطالق می شود

**قابلیت آزمون چه چیزی را بیان می دارد ؟**

در عمل، قابلیت آزمون، به احتمال اینکه نرم افزار حداقل دارای یک خطا باشد، اشاره دارد که در اجرای بعدی بروز خواهد نمود. البته، محاسبه این احتمال آسان نیست ونیازمند به کارگیری الگوریتم های پیچیده ای است

.

**چگونه یک سیستم باید قابلیت درست آزمون کردن را داشته باشد ؟**

باید بتوان حالت داخلی و ورودی های هر مولفه را کنترل نموده و سپس خروجی را مشاهده کرد**.**

**چه چیزهای در آزمون مورد ارزیابی قرار می گیرد؟**

ممکن است بخش های از کد، طراحی یا کل سیستم مورد آزمون قرار گیرد

**اندازه گیری پاسخ برای قابلیت آزمون با چه مواردی سر و کار دارد؟**

اندازه گیری پاسخ برای قابلیت آزمون با موارد زیر سروکار دارد:

چگونگی اثربخشی آزمون ها در کشف خطاها،

ν مدت زمان آزمون ها برای سطح مناسبی از پوشش

**خصوصیت کیفی قابلیت استفاده چه چیزی را بیان میدارد؟**

چقدر برای یک کاربر آسان است که کار (وظیفه ) دلخواه خود را انجام دهد.

**سیستم برای انجام کارها، از کاربران چه حمایت هایی می نماید؟**

یادگیری ویژگی های سیستم )features system Learning)

**چه نکاتی در مورد سناریوهای عمومی وجود دارد ؟**

نقش سناریوهای عینی ( Concrete )برای نیازمندی های خصوصیات کیفی همانند: نقش موارد کاربری (Cases Use )برای نیازمندی های وظیفه مندی است. ν

مجموعه ای از سناریوهای عینی می توانند: برای نمایش نیازمندی های خصوصیات کیفی یک سیستم استفاده شوند.

ν یکی از کاربردهای سناریوهای عمومی: برقراری ارتباط بین ذینفعان است.

**مراحل اجرای خصوصیات کیفی را نام ببرید؟**

1. نیازمندی های حرفه
2. خصوصیات کیفی
3. انتخاب تاکتیک
4. پیاده سازی تاکتیک: الگوهای طراحی و الگوهای معماری

**تاکتیک چیست ؟**

تاکتیک یک تصمیم طراحی است که کنترل پاسخ خصوصیت کیفی را تحت تاثیر قرار می دهد.

هر تاکتیک یک گزینه طراحی برای معمار است (مثال: تاکتیک افزونگی یکی از گزینه های طراحی معمار برای افزایش »قابلیت دسترسی« است(.

**راهبرد معماری چیست ؟**

مجموعه ای از تاکتیک ها، یک راهبرد معماری ( strategies Architectural )را تعریف می کنند.

**الگو چیست ؟**

الگوها، تاکتیک ها را بسته بندی می کنند ( مجموعه ای از تاکتیک ها هستند(. الگویی که از »قابلیت دسترسی« حمایت می کند، مطمئناً دربرگیرنده هر دو نوع »تاکتیک افزونگی« و »تاکتیک همگام سازی« است.

**تاکتیک های معماری از چه طریقی فراهم می شوند ؟**

تاکتیک هایی که معمار برای طراحی یک سیستم استفاده می کند، از طریق : الگوهای طراحی، الگوهای معماری یا راهبردهای معماری فراهم می شوند.

**معمار پس از انتخاب خصوصیات کیفی، برای اجرای آنها چه کاری می کند؟**

معمار پس از انتخاب خصوصیات کیفی، برای اجرای آنهااز الگوها یا سبک های معماری استفاده می نماید. هر الگو یا سبک، مجموعه ای از تاکتیک ها است.

**هدف از تاکتیک های قابلیت دسترسی چیست ؟**

1. عدم تبدیل نقص به شکست : شکست سببِ عدم توانایی سیستم در ارائه خدمات می شود.
2. کاهش اثرات نقص : نقص کمترین اثر را بر عملکرد کاربران داشته باشد.
3. اصالح نقص : وضعیت سیستم به حالت قبل از نقص برگردد.

**تاکتیک های قابلیت دسترسی دارای چه ویژگی هایی است ؟**

همه تاکتیک های »قابلیت دسترسی« دارای روش هایی از نوع زیر هستند:

1. افزونگی،
2. کنترل صحت سیستم برای شناسایی نقص )سیستم کنترل و آزمون شود تا تقسیم بر صفر اتفاق نیفتد(،
3. بازیافت )ترمیم( سیستم وقتی شکست اتفاق می افتد.

**تاکتیک های دسته بندی را نام ببرید؟**

1. تشخیص نقص )Fault Detection(
2. بازیافت (ترمیم )نقص )Recovery Fault )
3. جلوگیری از نقص )Prevention Fault )

**تاکتیک های تشخیص نقص را نام ببرید؟**

1. انعکاس/صدا Ping/echo )): یک مولفه، پیامی را ارسال میکند و انتظار دارد تا در بازه زمانی مشخصی پاسخ را دریافت نماید. عدم دریافت پاسخ به معنای خطا تلقی می شود )رابطه بین سرویس گیرنده ها و سرویس دهنده ها(
2. ضربان قلب ( Heartbeat ) : یک مولفه، به صورت دوره ای یک پیام )ضربان قلب( را در بازه زمانی مشخصی منتشر می کند و مولفه دیگر آنها را دریافت می کند. اگر مولفه دریافت کننده پیام، قطع شدن پیام را متوجه شود به معنای وجود خطا است )در اینصورت مولفه اصالح نقص را فراخوانی می کند(.
3. Exceptions (استثنائات ) : وقتی خطایی رخ می دهد یک استثناء اتفاق می افتد و مدیر استثنائات آن را تشخیص و رسیدگی به آن را آغاز می نماید )در نهایت حالت خود را به حالت قبل از وقوع خطا برمی گرداند(

**تاکتیک های بازیافت نقص (ترمیم ) را نام ببرید ؟**

1. رای گیری (voting ) : فرآیندهایی که بصورت افزونه روی پردازنده های مجزا اجرا می شوند، ورودی یکسان دریافت می کنند و خروجی خود را به یک رای دهنده )Voter )ارسال می کنند. رای دهنده اگر انحرافی یا اشتباهی در رفتار هر پردازنده دریافت کند آن را به عنوان یک نقص در نظر می گیرد. کاربرد: در سیستم های کنترلی- برای اصالح نقص یک الگوریتم یا خطای یک پردازنده. Voter از الگوریتم »قانون اکثریت« یا »مولفه های برتر« استفاده می کند و پاسخ درست را انتخاب می کند.
2. :)افزونگی فعال )Active Redundancy : همه مولفه های افزونه، به صورت موازی به رویدادها پاسخ می گویند )وضعیت مولفه ها مشابه یعنی همه به روز هستند(. ν در این تاکتیک، تنها اولین پاسخ مورد استفاده قرار می گیرد و مابقی نادیده گرفته می شوند. در صورت بروز نقص، زمان دردسترس نبودن سیستم بسیار کوتاه )چند میلی ثانیه( است، زیرا مولفه پشتیبان فعال شده و سرویس های مربوطه را پاسخ می دهد )مولفه پشتیبان در آن لحظه فعال است( . این تاکتیک اغلب در سیستم های سرویس گیرنده/سرویس دهنده به کار گرفته می شود )مانند سیستم های مدیریت پایگاه داده(
3. افزونگی غیر فعال (passive redundancy ) :

یک مولفه )به عنوان نسخه اصلی( به رویدادها پاسخ می دهد و دیگر مولفه ها را مطلع می سازد تا حالت خود را به روزرسانی کنند )تغییرات را در خود اعمال نمایند(

وقتی نقص اتفاق می افتد، ابتدا سیستم مطمئن می شود که نسخه های پشتیبان به روز هستند و سپس آنها را استفاده می کند )انتقال از مولفه اصلی به مولفه پشتیبان(

همگام سازی بین مولفه های پشتیبان وظیفه مولفه اصلی است.

این تاکتیک در سیستم های کنترلی استفاده می شود )به ویژه حالتی که ورودی از طریق کانال های ارتباطی یا سنسورها است و ضرورت ایجاب می کند در صورت بروز نقص، انتقال از مولفه اصلی به مولفه پشتیبان صورت گیرد(

1. Spare(یدک) :

در این تاکتیک، یک سکوی محاسباتی یدکی جانشین، جایگزین مولفه های دارای خطا می شود .

حالت های متفاوتی از سیستم )یک نقطه وارسی از وضعیت سیستم( به صورت دوره ای در یک رسانه ماندگار ثبت می شود )تمامی تغییرات در وضعیت سیستم در آن رسانه ثبت می شود(

هنگام وقوع خطا، حالت مناسب تشخیص داده شده و یدک جایگزین آن می شود )این عمل چند دقیقه طول می کشد(

1. عملیات در سایه ( Shadow Operation ):

مولفه ای که قبالً خطا داده است، برای مدت کوتاهی در حالت سایه کار می کند تا مطمئن شود همان رفتاری را دارد که که قبل از حالت ترمیم داشت )برگشت مولفه به سیستم پس از اصالح(. یعنی مولفه خطادار از حالت اجرا خارج می شود و بازیافت روی آن اعمال می گردد و پس از مدت کوتاه در صورت اصالح رفتار، مجددا به سیستم بر می گردد.

1. همگام سازی مجدد حالت ) resynchronization State ) :

تاکتیک های افزونگی فعال و غیرفعال نیاز به روشی دارند که مولفه جایگزین قبل از شروع سرویس خود، به روز باشد، لذا همزمانی ضرورت دارد )همزمانی اِعمال تغییرات در مولفه های پشتیبان(

1. نقطه وارسی / عقبگرد ( checkpoint /rollback )

نقطه وارسی یک ذخیره سازی از وضعیت سازگار سیستم در رسانه ماندگار است.

ثبت حالت سازگار، به صورت دوره ای یا موقع پاسخ به یک رویداد خاص انجام می شود.

هنگام وقوع خطا، سیستم به آخرین حالت پایدار با توجه به آخرین نقطه وارسی برگردانده می شود.

**تاکتیک های جلوگیری از نقص را نام ببرید؟**

1. خارج کردن از سرویس (removal from service ):

مولفه ای که دارای مشکل است از عملیات خارج می شود تا سبب خطا نشود.

اینکار می تواند دستی یا خودکار انجام شود )خودکار: نیاز به یک راهبرد معماری دارد(.

1. تراکنش ها ( transactions ) :

تراکنش مجموعه ای از دستورات هستند که یا همه باهم اجرا می شوند و یا هیچیک اجرا نمی شوند

اگر خطایی در یک دستور درون تراکنش اتفاق افتد، کلیه دستورات دیگر اعم از اجراشده یا نشده، نادیده گرفته می شوند

1. نظارت بر فرایند (process monitor ) :

کنترل کننده فرآیندها با مشاهده نقص، می تواند فرآیند را حذف نماید و یا یک نمونه جدید از آن ایجاد کرده و آن را شروع مجدد نماید.

**هدف از تاکتیک قابلیت اصلاح چیست ؟**

کنترل زمان و هزینه پیاده سازی، آزمایش و استقرار تغییرات است .

**دسته بندی تاکتیک های قابلیت اصلاح را نام ببرید ؟**

1. محلی سازی تغییرات )modifications Localize:

هدف، کاهش تعداد ماژول هایی است که در اثر یک تغییر تحت تاثیر قرار می گیرند( یعنی مسئولیت آن ها به طور مستقیم تغییر می کند). این عمل باعث کاهش هزینه تغییرات می شود.

1. جلوگیری از تاثیرات موجی )Effects Ripple Prevent:

هدف، ایجاد تغییر در ماژولی است که به طور مستقیم تحت تاثیر آن تغییر نیست.

یعنی مسئولیت ماژول تغییر نمی کند ولی برای دیده شدن اثر تغییر در یک ماژول، پیاده سازی آن باید تغییر کند.

1. به تاخیر انداختن زمان انقیاد )Time Binding Defer:

هدف، کنترل زمان و هزینه استقرار است

به کاربر یا مدیر سیستم اجازه می دهد تغییراتی را ایجاد کنند و یا ورودی را فراهم کنند تا رفتار سیستم را تحت تاثیر قرار دهد.

**تاکتیک های محلی سازی استقرار را نام ببرید ؟**

1. حفظ انسجام معنایی :

پیوستگی معنایی به ارتباط بین مسئولیت ها در یک ماژول اشاره دارد.

هدف از این تاکتیک، نگه داشتن تمام وظایف مرتبط در یک ماژول بدون استفاده از ماژول دیگر است. )معیارهای اندازه گیری انسجام معنایی عبارت است از: Coupling و Cohesion.)

1. پیش بینی تغییرات مورد انتظار:

در این تاکتیک، تغییرات پیش بینی شده در تقسیم بندی ماژول ها موثر است. ν

این تاکتیک با حداقل کردن اثرات تغییرات در ماژول ها مرتبط است.

1. عمومی سازی ماژول ها

عمومی تر نمودن ماژول سبب می شود که توابع بیشتری از آن استفاده نمایند**.**

1. محدود کردن گزینه ها ی ممکن

محدود کردن گزینه ها سبب کاهش تغییرات و اصلاحات خواهد شد. ν مثال: در یک خط تولید اگر اجازه تغییر پردازنده داده شود، محدود کردن استفاده از پردازنده های هم خانواده، گزینه های ممکن برای تغییرات و لذا اثرات ناشی از آن تغییرات را محدود می کند.

**انواع تاکتیک های جلوگیری از تاثیرات موجی را نام ببرید .**

1-‌ پنهان سازی اطلاعات (hide information ):

تجزیه وظایف یک ماژول به قسمت های کوچکتر و انتخاب اینکه کدام یک از این قسمت ها عمومی و کدام خصوصی باشند

هدف: تغییرات در داخل یک ماژول ایزوله شود و از انتشار به سایر ماژول ها جلوگیری گردد.

2- نگه داری رابط های موجود (maintain exiting interfaces )

اگر ماژول B وابسته به نام و امضای یکی از رابط های ماژول A باشد، ثابت نگه داشتن این رابط اجازه می دهد که B نیز بدون تغییر باقی بماند )از طریق جدا کردن واسط از پیاده سازی(.

3-‌ محدود کردن مسیرهای ارتباطی (restrict communication paths )

کاهش تعداد ماژول هایی که داده ی تولید شده یک ماژول را مصرف می کنند و نیز کاهش تعداد ماژول هایی که داده تولید می کنند )کاهش تعداد ماژول های تولید کننده و مصرف کننده داده)

این عمل تاثیر موجی را کاهش می دهد، زیرا تولید و مصرف داده، وابستگی تعریف می کند و باعث انتشار موجی می شود.

4-‌ استفاده از یک میانجی/واسطه (use an intermediary )

در صورتی که دو ماژول به یکدیگر وابسته باشند، استفاده از یک میانجی کمک می کند تا فعالیت های بین ماژول ها را مدیریت نمود

. ν انواع مختلفی از میانجی ها وجود دارند که از آن جمله می توان به میانجی های زیر اشاره کرد: θ

داده یا مخزن )سبک blackboard و passive – )رابط بین مصرف کننده و تولید کننده داده θ

مخزن ها، می توانند گرامر تولید شده بوسیله A را به فرضیات B تبدیل کنند

. θ سرویس )الگوهای facade( نمای خارجی(، bridge ،mediator ،strategy )و ... رابطی فراهم می کنند که گرامر یک سرویس را به دیگری تبدیل می کنند

**تاکتیک های به تاخیر انداختن زمان انقیاد را نام ببرید ؟**

1-‌ ثبت نام زمان اجرا (runtime registration ):

ماژول ها در زمان اجرا و به صورت play and plug ثبت و مورداستفاده قرار می گیرند )الگوی Subscribe/Publish هم در زمان اجرا قابل پیاده سازی است(

2- فایل های پیکربندی (configuration files )

فایل های پیکره بندی تمایل دارند که در زمان شروع، پارامترها را مقداردهی کنند.

3- ‌چند ریختی (polymorphism )

اجازه انقیاد دیر هنگام فراخوانی متدها را می دهد .

4-‌ جایگزینی مولفه ها (component replacement )

اجازه بارگذاری مولفه ها در زمان اجرا را می دهد.

1. پایبندی به پروتوکول های تعریف شده :

اجازه انقیاد فرآیندهای مستقل در زمان اجرا را می دهد.

**هدف تاکتیک های کارایی چیست ؟**

هدف تاکتیک های کارایی ارائه پاسخ به یک رویداد ورودی در زمان تعیین شده است .

**رویداد ها چگونه میتوانند باشند و نمونه ای از رویداد ها را نام ببرید ؟**

رویدادها می توانند به صورت جریانی از رویدادها یا منفرد باشند (شروع کننده یک درخواست( و مونه ای از رویدادها: رسیدن یک پیام، انقضای بازه زمانی، تشخیص یک تغییر وضعیت. با وقوع یک رویداد، سیستم رویداد مربوطه را پردازش کرده و پاسخ مناسب را تولید می کند.

**تاکتیک های کارایی چه چیز را کنترل می کنند ؟**

مدت زمان تولید پاسخ برای یک رویداد را کنترل می کنند.

**زمان تاخیر چیست ؟**

زمان بین رسیدن یک رویداد تا تولید یک پاسخ است.

**دو مفهوم مهم که بر زمان پاسخ تاثیر دارند را نام ببرید ؟**

1. )Resource consumption( مصرف منابع

پردازنده، رسانه های ذخیره سازی داده، پهنای باند شبکه ارتباطی، حافظه و

2- زمان مسدود شدن (blocked time )

زمانی که یک فرآیند بنا به دالیل ذیل مسدود می شود و باعث تاخیر در زمان پاسخ می شود:

o رقابت بین منابع )رقابت فرایندها برای دسترسی به یک منبع(. O

قابل دسترس نبودن منابع )به دلیل برون خط )offline )بودن منبع، خطا در مولفه های آن و ...(. o

وابستگی یک محاسبه به نتایج دیگر محاسبات )که هنوز تولید نشده اند

**دسته بندی های مختلف تاکتیک های خصوصیت کیفی (کارایی ) را نام ببرید ؟**

1. تقاضای منابع (resource demand )
2. مدیریت منابع (resource management )
3. داوری منابع (resource arbitration )

**سرچشمه تقاضای منابع چیست ؟**

جریان رویدادها سرچشمه و منشاء تقاضای منابع هستند.

**دو خصوصیت (ویژگی ) تقاضای منابع به شرح زیر است :**

1. زمان بین رویداد ها در یک جریان چقدر است .
2. چقدر منبع برای هر درخواست مصرف میشود .

**تاکتیک های خصوصیت کیفی (امنیت )**

تاکتیک های امنیت به سه گروه زیر تقسیم می شوند :

1. مقاومت در برابر حملات (resisting attacks )
2. تشخیص حملات (detecting attacks )
3. بازیابی از حملات ( recovering from attacks )
4. مثل: قراردادن قفل بر روی درب خانه، تاکتیک مقاومت در برابر حمله است. قراردادن یک سنسور حرکتی در داخل خانه، تاکتیک شناسایی حمله است و داشتن بیمه نامه خانه، تاکتیک بازیابی از حمله است

**مجموعه تاکتیک های مقاومت resisting attacks) ): در برابر حملات را نام ببرید ؟**

1. احراز هویت کاربران(authenticate users) :

:احراز هویت کاربران با استفاده از کلمات عبور، کلمات عبور دو مرحله ای، گواهینامه های دیجیتال، شناسه های بایومتریک و غیره

2-‌ انتساب حق دسترسی به کاربان (authorize users ):

انتساب حق دسترسی به داده ها و سرویس های مختلف پس از تصدیق کاربران

3- ‌حفظ محرمانگی داده ها (maintain data confidentiality )

:استفاده از الگوریتم های رمزگذاری روی داده ها و پیوندهای ارتباطی )VPN یا SSL )برای حفظ محرمانگی و دسترسی های نامعتبر

4- حفظ یک پارچگی (maintain integrity )

تحویل داده ها به همان شکلی که از ابتدا مدنظر بوده با استفاده از کدگذاری آن ها

داده می تواند اطالعات اضافی رمزگذاری شده در درون خود داشته باشد )مانند checksums یا نتایج hash( )checksums به منزله اثر انگشت فایل شما است(

5- نمایش محدود (limit exposure )

سرویس هایی که مورد نیاز کاربران نیستند، نباید توسط کاربران دیده شوند. ν

معمار می تواند با محدود نمودن سرویس های قابل رویت توسط کاربران، امکان حمله به آنها را کاهش دهد.

6-دسترسی محدود (limit access )

استفاده از دیواره آتش )Firewall )می تواند دستیابی کاربران غیرمجاز به سرویس ها و داده ها را کاهش دهد

**تاکتیک های تشخیص حملات را نام ببرید ؟**

1-‌ تشخیص نفوذ (intrusion detection ):

سیستم های تشخیص نفوذ می توانند برای شناسایی حمالت با استفاده از روش هایی نظیر الگوبرداری از ترافیک شبکه و ... مورد استفاده قرار گیرند )مثالً اگر دسترسی یک فرد به شبکه در خارج از ساعت اداری باشد، می تواند یک نفوذ تلقی شود(.

**تاکتیک های بازیابی پس از حمله را نام ببرید ؟**

1. رد ممیزی (audit trial )

تاکتیک موردنیاز برای شناسایی حمله کننده، نگهداری یک رد ممیزی است. ν

رد ممیزی، ثبت تمام تراکنش های انجام شده بر روی داده ها به همراه اطالعات تشخیص هویت است. ν

ردگیری فعالیت های حمله کننده و جلوگیری از عدم انکار وی و پشتیبانی از بازیابی سیستم است. )بازگرداندن سیستم به وضعیت قبل از حمله(

**هدف از آزمون سیستم چیست ؟**

کشف خطا(نقص) های آن است .

**هدف از تاکتیک های قابلیت آزمون چیست ؟**

ساده تر کردن آزمون نرم افزار هنگام کامل شدن یک نسخه است.

**چرا آزمون نرم افزار اهمیت دارد ؟**

به دلیل اینکه آزمون نرم افزار می تواند سبب کاهش هزینه توسعه نرم افزار شود، قابلیت آزمون اهمیت فراوانی دارد.

**دسته بندی تاکتیک های قابلیت آزمون چیست ؟**

این تاکتیک ها، به دو دسته »مدیریت ورودی و خروجی« و »نظارت داخلی« تقسیم می شوند.

**دسته بندی تاکتیک های مدیریت ورودی و خروجی** (manage input/output )

**را نام ببرید ؟**

1-ثبت /برگشت به عقب(record playback )

ثبت اطالعات مبادله شده در رابط ها و استفاده از آنها به عنوان ورودی روال های آزمایش سیستم.

2-جدا کردن رابط از پیاده سازی( separate interface from implementation )

جداسازی، امکان جایگزینی پیاده سازی های مختلف با اهداف آزمون متفاوت را فراهم می نماید.

3-اختصاصی کردن مسیرها/رابط دسترسی (specialize access routes /interfaces )

داشتن رابط های آزمون اختصاصی امکان ضبط (دریافت اطالعات) یا تعیین مقادیر متغیر برای یک مولفه را از طریق تحت کنترل درآوردن آزمون، مستقل از اجرای نرمال آن فراهم می کند.

استفاده از یک مسیر یا رابط های ویژه برای آزمایش، اجازه دریافت اطالعات از منابع مختلف و بکارگیری آنها به منظور تحت کنترل درآوردن آزمایش را فراهم می کند (ایجاد سلسله مراتبی از رابط های آزمون(.

مسیرهای دسترسی/واسط های ویژه باید از واسط های وظیفه مندی های سیستم جدا نگهداری شوند )به این واسط ها، واسط های کنترلی سیستم گویند. مثل واسط هایی برای نمایش درجه حرارت یا فشار یک دیگ بخار در یک واحد صنعتی)

**تاکتیک های نظارت داخلی(internal monitoring ) را نام ببرید ؟**

1. نظارت های از پیش تعیین شده ( **built in monitoring )**

استفاده از یک مولفه کنترلی برای: کنترل کارآیی، ظرفیت، امنیت و سایر اطالعات قابل دسترس از طریق رابط )مثل(noc )

ثبت رویدادها توسط مولفه کنترل کننده (ضمن ثبت رویدادها، اطالعات وضعیت سیستم روی رابط های خاص نمایش داده می شوند(.

**قابلیت استفاده با چه چیزی مرتبط است ؟**

قابلیت استفاده مرتبط است با میزان راحتی کاربر در انجام یک عمل مشخص و پشتیبانی هایی که یک سیستم برای کاربر فراهم می کند.

**انواع تاکتیک های قابلیت استفاده را نام ببرید ؟**

1. زمان اجرا: کمک به کاربر در زمان اجرای نرم افزار.
2. زمان طراحی: کمک به توسعه دهنده رابط کاربری در زمان طراحی نرم افزار

**هدف از تاکتیک های زمان اجرا چیست و انواع آن را نام ببرید ؟**

هدف از تاکتیک های قابلیت استفاده در زمان اجرا، جواب و کمک مناسب به کاربران نرم افزار در زمان اجرا است.

1. حمایت از ابتکار عمل کاربر Initiative User) )

کاربر در انجام عملیات اصلی پیش قدم است (شروع کننده کاربر است و سیستم نیز به آن پاسخ می دهد ) مانند: استفاده از Cancel ،Undo و A aggregate

1. حمایت از ابتکار عمل سیستم Initiative System)) مدل کاربر، مدل سیستم و مدل وظیفه )Model Task)

نگهداری یک مدل از کاربر: دانش کاربر از سیستم و رفتارهای او را در قبال سیستم مشخص می کند. مثال: صفحه ای که توسط کاربر در حال خواندن است، سریع و قبل از خوانده شدن ناپدید نشود.

نگهداری یک مدل از سیستم: ارایه بازخوردهای مناسب به کاربر توسط سیستم را مشخص می کند. مثال: پیش بینی زمان به پایان رسیدن فعالیت جاری در سیستم )از طریق bar task.) ν

نگهداری یک مدل از وظیفه: مربوط به وظیفه ای است که در سیستم درحال اجرا شدن است. به سیستم کمک می کند راهنمایی هایی برای کاربر فراهم کند )مثال: با دانستن شروع هر جمله با حروف بزرگ، حروف کوچک در هر موقعیت را اصالح می کند(. مثال یک سیستم اتوماسیون اداری ضمن اجازه ایجاد یک نامه به عنوان یک task مشخص، در حین تایپ جمالت، شروع هر جمله با حروف بزرگ را مدیریت می کند

**انواع تاکتیک های طراحی را نام ببرید ؟**

1. جداسازی رابط کاربری از بقیه نرم افزار

ناظر به محلی کردن تغییرات پیش بینی شده

با توجه به امکان تغییر مکرر واسط کاربری در طول توسعه، جدا کردن کد واسط از بقیه نرم افزار، باعث تمرکز تغییرات در داخل آن می شود.

الگوهای معماری مربوط به پیاده سازی این تاکتیک شامل: MVC ، PAC ، Slinky/Arch و ...

**برای اعمال خصوصیات کیفی (دستیابی به ویژگی های مورد انتظار در سیستم ) چه کار باید کرد ؟**

معمار، معمولا الگو یا مجموعه ای از الگوها را انتخاب می نماید.

هر یک از خصوصیات کیفی با چند تاکتیک درگیر هستند.

هر الگو، چندین تاکتیک را پیاده سازی می کند.

هر نوع پیاده سازی از الگوها، تاکتیک ها را شفاف و عینی می سازد )مثال: یک پیاده سازی الگو می تواند: 1 )یک ثبت وقایع )log )از درخواست ها برای اهداف بازیافت ایجاد کند، 2 )یک رد ممیزی )trail audit) را نگهداری کند 3 )و یا قابلیت آزمون را پشتیبانی نماید(

**فرایند تحلیل و طراحی شامل چه چیزی است ؟**

فرآیند تحلیل برای معمار شامل فهم همه تاکتیک هایی است که در یک پیاده سازی درگیر هستند.

فرایند طراحی شامل انتخابی خردمندانه از مجموعه ای از تاکتیک ها است تا اهداف موردنیاز معمار را برآورده سازند.

**سبک و الگوی معماری را مقایسه کنید ؟**

الگوی معماری به سبک معماری نیز مشهور است )مشابه سبک معماری در ساختمان سازی(. ν مانند سبک معماری گوتیک یا سبک یونانی

سبک معماری، دیدگاهی کالان تر نسبت به الگو دارد.

الگو دارای قالب تعریف شده و مشخص است.

هر دو )سبک و الگو( به دنبال اجرای خصوصیات کیفی هستند

**هر الگوی معماری حاوی چه چیزهایی است ؟**

اجزاء: مجموعه ای از انواع عناصر )مخزن داده، مولفه محاسباتی و غیره

اتصاالت: لایه توپولوژیکی نشان دهندة ارتباط بین مولفه ها

قوانین: مجموعة از اجبارهای معنایی )مثال: فیلترها در سبک filter-and-pipe) θ

مکانیزم تعامل: مجموعة از مکانیزم های تعاملی که هماهنگی بین مولفه ها را تضمین می نماید )مانند فراخوانی زیرروال ها(.

**سبک مولفه های مستقل چیست (independent components )؟**

همکاری بین فرآیندهای موازی (دستیابی به قابلیت اصلاح از طریق جداسازی بخش های محاسباتی سیستم(

**این سبک شامل چه زیر سبک های چی است ؟**

1. فرایند های ارتباطی
2. سیستم های برپایه رخداد
3. فراخوانی ضمنی
4. فراخوانی صریح

**هدف سبک مستقل چیست ؟**

1. قابلیت اصالح )modifiability -)اصالح یک مولفه، در دیگری تاثیر نمی گذارد.
2. مقیاس پذیری )scalability -)با بهبود عملکرد هر یک از مولفه ها، می توان عملکرد کل سیستم را بهبود بخشید تا بیشتر مقیاس پذیر باشد و با افزایش خطی درخواست ها، زمان پاسخ به طور خطی افزایش نیابد.

**نحوه کارکرد سبک فرایندهای(**communication process ) **ارتباطی چگونه است ؟**

تعامل بین فرآیندها از طریق ارسال پیام انجام می شود )عمل wake/signal در فرایندهای همزمان آسنکرون

**اجزا سبک های فرایند های ارتباطی چیست ؟**

مولفه ها : واحدهای موازی مانند فرآیند یا نخ هستند.

اتصال دهنده ها: ارسال پیام )passing Message )یا هماهنگ سازی )Synchronization )هستند. ν

نمونه کاربرد این سبک: 1-سرویس های سیستم عامل 2- سیستم های موازی

یک زیرسبک از این سبک، سبک سرویس گیرنده و سرویس دهنده است

تمرکز این زیرسبک بر روی سرویس های مختلفی است که سرویس گیرنده ها نیاز دارند.

**سبک فراخوانی ضمنی چگونه کار میکردند ؟**

تعامل بین فرآیندها یا رویه ها با استفاده از رویدادها انجام می شود )این نوع فراخوانی ها توسط خود کامپایلرها یا ماشین های مجازی مانند JVM و بدون دخالت کاربر انجام می شود(

مولفه ها در این سبک: رویه ها )Procedures)

اتصال دهنده ها: رویدادها )Events(

جریان داده: داده های مشترک در مخزن

**سبک های متمرکز بر روی داده ( data centered ) چیست ؟**

استفاده از داده های مشترک

y( حالتی که منبع داده غیرفعال است و ارتباط بین مولفه ها و منبع داده یک طرفه است( ♦ Blackboard( حالتی که منبع داده فعال است و ارتباط بین مولفه ها و منبع داده دو طرفه است(

**سبک مخزن چگونه کار میکند و اجزای ان چیست ؟**

پاسخ گویی به درخواست بازیابی دادهها را داشته باشد و مخزن داده بصورت غیرفعال عمل مینماید.

مولفه ها: واحدهای محاسباتی، کد و...

اتصال دهنده ها: رویدادهای احضارشده

نمونه های کاربرد: تشخیص صوت، پردازش سیگنال، امروزه روشن شده که این سبک برای طراحی کامپایلرها بهترین سبک موجود است.

**سبک تخته سیاه چکونه است و اجزا ان چیست ؟**

**مخزن داده با تغییر داده، مصرفکننده را مطلع می سازد)مخزن داده فعال و حساس به تغییرات است(. ν**

**مولفه ها: ♦ واحدهای محاسباتی، کد و**

**... ν اتصال دهنده ها: ♦ رویدادهای احضارشده ν جریان داده ها: ♦ داده ها در هر دو سیستم )هم مولفه ها و هم مخزن داده( ν نمونه ها: ♦ ساخت سامانه های تشخیص گفتار، پردازش سیگنال، شناسایی خودرو و ردیابی آن**

**سبک داده های مشترک چه خصوصیات کیفی را پیاده می کند ؟**

دستیابی به قابلیت تجمیع پذیری داده ها ( در مخزن مشترک(.

قابلیت اصلاح بالا (به دلیل امکان تغییر هر یک از مولفه ها بدون تاثیرگذاری روی بقیه مولفه ها(.

مقیاس پذیری (مولفه های جدید به راحتی می توانند اضافه شوند(.

فصل 7

**چه چیزی به عنوان پایه ای برای فرایند توسعه نرم افزار مطرح است ؟**

معماري به عنوان پایه اي براي فرآیند توسعه نرم افزار مطرح است.

**سازمانی که از معماری به عنوان پایه ای برای فرایند توسعه نرم افزار استفاده می کند باید چه کار کند ؟**

سازماني که از معماري به عنوان پایه اي براي فرآیند توسعه نرم افزار خود استفاده مي کند باید جایگاه معماري در چرخه حیات نرم افزار را مشخص نماید.

**جایگاه معماری در چرخه حایت تولید نرم افزار چه چیزی را تعیین می کند ؟**

جایگاه معماري در چرخه حیات تولید نرم افزار:

1. نحوة دستیابي به خصوصیات کیفي را تعیین مي کند.
2. نحوة تعامل با مشتریان براي درک نیازمندي هاي آنها را تعیین مي کند.

**یکی از مدل های چرخه حیات تولید نرم افزار چیست و چه اجازه هایی را صادر میکند ؟**

چرخه حیات تحویل تکاملی (Evolutionary Delivery life cycle ) است که

1. اجازه دریافت بازخورد از مشتریان و کاربران را مي دهد.
2. اجازه توسعه نسخه هاي مختلف Version )) قبل از نسخه نهایي را مي دهد.
3. اجازه افزودن وظیفه مندي به هر تکرار و در نسخه هاي مختلف را مي دهد.

**هدف از مدل چرخه حیات تحویل تکاملی چیست ؟**

هدف از مدل چـرخه حیات تحویل تکاملي:

1. دریافت بازخورد از مشتریان
2. اعمال نظرات مشتریان در چندین نسخه مختلف قبل از تحویل نسخه نهایی
3. امکان اضافه کردن وظیفه مندی های جدید در هر بار تکرار
4. فلش بازخورد از طراحی معماری به تحلیل نیازمندی های اولیه نشان دهنده فرایند تکراری بودن عملیات طراحی است

**چـرخه حیات تحویل تکاملي چه چیزی را نشان مي دهد؟**

چـرخه حیات تحویل تکاملي نشان مي دهد که معماري مي تواند با داشتن Shaping Requirements (شامل چند نیازمندي اولیه کسب وکار، کیفي و وظیفه مندي)شکل بگیرد (نیازمندی های کلی سیستم(.

**پیشران های معماری چی هستند ؟**

نیازهای وظیفه مندی و کیفی شکل دهنده معماری هستند .

**تمایز نیازمندي کسب وکار با نیازمندی وظیفه مندي چیست؟**

نیازمندي های وظیفه مندي: به عنوان یک مالک محصول، می خواهم مشتریان ما هر بار خرید یک محصول، یک ایمیل خودکار دریافت کنند. این قطعا یک نیاز وظیفه مندی است. ν مالک محصول واقعا در تالش برای دستیابی به چه چیزی است؟ آیا ارسال ایمیل خودکار بهترین راه برای رسیدن به هدف نهایی است؟ ν

نیازمندهاي کسب وکار: درک مسئله واقعی یا نیاز کسب و کار (Requirement Business )اطمینان حاصل می دهد که باالترین ارزش را به مشتری خود تحویل می دهید. هنگامی که نیازهای واقعی کسب و کار را درک کردیم، می توانیم بهترین روش برای دستیابی به ساخت یک راه حل )خواه اتوماسیون در آن دخیل باشد یا خیر( و نحوه اجرای آن را شروع کنیم

**نحوه یافتن پیشران های معماری را توضیح دهید ؟**

1. نیازمندی های سیستم که اولویت باالیی دارند را بیابید (این نیازمندی ها معموال کم هستند(.
2. آنهایي را انتخاب نمائید که بیشترین تاثیر را روي معماري دارند. تعداد این موارد محدود و معموال کمتر از 10 مورد است.
3. آنها را به سناریوهاي کیفي یا موارد کاربري تبدیل نمائید.
4. با استفاده از روش) ATAM روش تحلیل مصالحه ای معماری) پیشران هاي معماري به سناریوهاي کیفي تبدیل مي شوند.

**بعد از تعیین پیشران های معماری چه چیزی شروع میشود ؟**

بعد از تعیین پیشران های معماری، طراحی معماری می تواند شروع شود.

**روش طراحی مبتنی بر ویژگی چیست ؟**

1. هم نیازهای کیفی و هم نیازهای وظیفه مندی را برآورده می کند. θ
2. مي توان آن را گسترشی از دیگر روش هاي توسعه نرم افزار دانست مانند ) RUP )
3. ADDدر طراحي سطح باالی معماری در RUP مي تواند مورد استفاده قرار گیرد.
4. مبتني بر فرآیند تجزیه ( Process Decomposition )بر روی خصوصیات کیفي است.

**مراحل روش ADD چیست؟**

1. انتخاب تعدادي تاکتیک و الگوي معماري برای برآوردن مجموعه اي از سناریوهاي کیفي.
2. تخصیص وظیفه مندي به انواع ماژول هاي موجود در الگوها.

**مهم ترین خروجی ADD چیست؟**

چند سطح اول دید تجزیه ماژول و دیگر دیدهاي موردنیاز است.

**ویژگی های خروجی حاصل از ADD چیست ؟**

1. تفـاوت معمـاري حاصل از ADD و معمـاري آماده براي اجرا، در میزان جزئیات تصمیمات طراحی است.
2. همه جزئیات دیدها از طریق طراحی ویژگی رانه )ADD ،)قابل شناسایی نیست )دانه بندی آن درشت است(.
3. با این حال، برای دستیابی به خصوصیات کیفی مناسب، ضروری است.
4. نیاز است تا جزئیات به معماري حاصل از ADD افزوده شود )تکرار-افزایشی(

**مراحل ADD چیست ؟**

صفر - مجموعة پیشران هاي کلیدی معماري را انتخاب کنید (طراحی ویژگی رانه وابسته به شناسایی پیشران ها است(

1. ماژولي را براي تجزیه انتخاب نمائید (در ابتدا کل سیستم را انتخاب کنید(.
2. ماژول را به صورت زیر پاالیش کنید:

الف( پیشران هاي کلیدی معماري را از مجموعه سناریوهای کیفی عینی و نیازهای وظیفه مندی انتخاب کنید.

* مشخص کنید کدام پیشران ها اهمیت بیشتري دارند.
* پیشران هایي با اهمیت کمتر، توسط پیشران هاي با اهمیت باالتر برآورده مي شوند.

ب) الگوي معماري را انتخاب کنید که پیشران های مربوطه را برآورده کند.

* برای هر خصوصیت کیفی تاکتیک هایی وجود دارد و (الگوهایی که این تاکتیک ها را مي توانند پیاده سازی کنند – فصل 5 .)مثال: الگوی مولفه های مستقل، تامین کننده خصوصیات کیفی قابلیت اصالح و مقیاس پذیری
* زیر ماژول هاي مورد نیاز براي پیاده سازي تاکتیک ها را تعیین کنید.

ج) ماژول ها را معرفي نموده، وظیفه مندي را به آنها تخصیص دهید و با استفاده از دیدهاي مختلف نمایش دهید.

* معمولا در سیستم هاي واقعي بیش از یک ماژول وجود دارد.
* نتیجه این قسمت تجزیه قابل جمع هر ماژول خواهد بود.
* تخصیص وظیفه مندي به یک ماژول و فرزندان، سبب کشف اطالعات تبادلي موردنیاز خواهد شد.
* تجربه طراحی معماری با روش ADD نشان داده است که براي شروع،
* یک دید از هر یک از گروه هاي دید تجزیه ماژول، دید همروندي و یا دید استقرار کافي خواهد بود.

د) رابط هاي زیرماژول ها (ماژول های فرزند) را تعریف کنید.

* عمل تجزیه برحسب نوع تعامل ماژول ها، سبب معرفي تعدادي ماژول و محدودیت مي شود.
* اطالعات ماژول ها، نوع تعامالت و محدودیت ها را در مستند رابط های هر ماژول معین کنید.

ه) وظیفه مندی ها و سناریوهاي کیفي را بازبینی و پالایش کنید و محدودیت هاي آنها را به زیرماژول ها اعمال نمایید.

3- گام هاي بالا را براي هرماژولی که نیاز به تجزیه بیشتر دارد اجرا کنید.

**کلیه نیـازمندی ها برای این مـاژول باید در دسترس باشند چه هستند ؟**

کلیه نیـازمندی ها برای این مـاژول باید در دسترس باشند،

1. محدودیت هـا،
2. نیاز هاي وظیفه مندي،
3. نیازهای کیفي.

**سناریوهای کیفی به چند صورت انجام میشوند ؟**

1. با تجزیه انجام شده به طور کامل اجرا می شود.
2. با تجزیه انجام شده و با محدودیت هایی روي زیرماژول ها انجام مي شود.
3. تجزیه انجام شده اثري بر سناریوي کیفي ندارد.
4. با تجزیه انجام شده، سناریوي کیفي برآورده نمي شود.

**چرا »ساختار تیم« در یک سازمان، منعکس کننده »ساختار تجزیه ماژول« است؟**

اولا ، مخفي سازي اطالعات، درون تیم انجام مي پذیرد

. ثانیا ، هر تیم توانایي انجام وظایف در ناحیه (Domain )خاصي را دارد و هر ماژول نیز باید مربوط به ناحیة خاصي باشد

**اسکلت سیستم چیست؟**

پس از طراحی معماری و آمادگی تیم ها برای ساخت آن، یک اسکلت سیستم می تواند ایجاد گردد.

قابلیتي را فراهم مي آورد تا یک وظیفه مندي سیستم به صورت موثر پیاده سازي شود

با استفاده از معماري مي توان ترتیب و گام هاي پیاده سازي را مشخص نمود..

* بخش هایی از نرم افزار را پیاده سازي نمائید تا اجرا و تعامل مولفه هاي معماري را ببینید.
* اگر سازوکار تعاملی فرایندها برپایه میان افزار باشد، به جای پیاده سازی، آن را نصب و نحوة اجراي آن را ببینید.
* وظیفه مندي هایي که باید اضافه شوند را انتخاب نمائید.
  + انتخاب مي تواند براساس سختي مسئله، تیم موجود ، ارایه قابلیت مفید به بازار و غیره باشد.
* با توجه به ساختار uses مي توان مشخص کرد کدام مولفه ها باید به طور صحیح اجرا شوند تا بتوان یک وظیفه مندی را به طور کامل پیاده سازی کرد.

**مزایای ساخت اسکلت سیستم چیست ؟**

1. مجتمع سازي و آزمون در هیچ جاي فرآیند توسعه، کار سختي نیست )به دلیل پیاده سازی تدریجی کلیه بخش های سیستم(.
2. در هر تکرار براحتي مي توان منبع خطاها را کشف و برطرف نمود.
3. مي توان تخمین قابل قبولی از بودجه و زمانبندی داشت )زیرا هر تکرار به صورت یک سیستم کوچک درنظر گرفته می شود(

**معایب ساخت اسکلت سیستم چیست ؟**

رابط هایي که اولین تیم توسعه دهنده برای تکمیل بخشی از سیستم ایجاد مي کند، اغلب تمام سیستم ها و زیرسیستم هاي بعدي باید با آن مطابقت داشته باشند.

حل مشکل: سعي کنید دربارة نحوة ارتباط زیرسیستم ها گفتگو نمائید و آنها را مشخص کنید.