**فصل 7**

**طراحی معماری نرم افزار**

**7-1 معماری چیست؟**

وقتی در مورد معماری یک ساختمان یا بنا صحبت میکنیم نگرش های متفاوتی به ذهنمان میرسد. در ساده ترین سطح ،شکل کلی و فیزیکی ساختمان را در نظر میگیریم.اما در واقع معماری آن بسیار پیچیده تر از این هاست.این حالتی است که دران اجزای مختلف ساختمانی به صورتی با هم ترکیب میشوند که یک کل سازمان یافته و منسجم را تشکیل دهند.این شیوه ای است که با ان ساختمان،همراه با ساختمانهای مجاور خوددر محیط سازگاری می یابد و هماهنگ می شود.دراینجا ساختار سیستم تا حدی هدف مشخص شده خود را نشان داده و نیازهای صاحبش را بر طرف میکند.در اینجا حس زیبایی شناسی ساختمان مانند تأثیر بصری ساختمان و شیوه ترکیب بافت ها،رنگ ها و مواد برای خلق نما و محیط زنده درونی مطرح است .در اینجا جزئیات کوچک مثل طراحی تسهیلات نور،نوع کف پوش و...فهرستی بلند بالا را به وجود می اورندو نهایتأ موضوع اصلی هنر است.

د رمورد ساختار و معماری نرم افزار چه میتوان گفت؟ Bassو همکارانش این عبارت دشوار ودیر فهم را به شکل زیر تعریف کرده اند:

معمار ی نرم افزار در یک برنامه یا سیستم محاسباتی عبارت است از ساختار یا ساختارهای سیستم که شامل اجزای نرم افزاری ،مشخصه های مشهود برونی این اجزا و ارتباطات میان آن ها است.

**7-2 چرا معماری**

معماری یک نرم افزار نمودی است که مهندس نرم افزار را قادر میکند:

- میزان تأثیر طرح را در مرتفع کردن نیازهای بیان شده،تحلیل کند.

- معماری جایگزین را در مرحله ای که تغییر طرح هنوز نسبتأ اسان است،بررسی کند.

- خطرات مربوط به ساخت نرم افزار را کاهش دهد.

طراحی داده باعث نمایش مولفه داده ای معماری و طراحی معماری بر تمرکز نمایش ساختار مولفه های نرم افزار،خواص انها و ارتباط بین انها قرار دارد.

**7-3 طراحی داده ها**

طراحی داده ها،همچون دیگر فعالیت های مهندسی نرم افزار (که گاهی به ان معماری داده ها نیز میگویند ) مدلی از داده ها یااطلاعات را در سطح بالایی از انتزاع ،ایجاد میکند .

سپس مدل داده ای به صورت باز نمای خاص پیا ده سازی در می اید که میتوان ان را بر سیسیتم مبتنی بر رایانه پردازش کرد.در بسیاری از برنامه های کاربردی ،معماری داده ها تأثیر شگرفی برمعماری نرم افزار دارد که باید ان راپردازش کند،بنابراین مدل داده ای :

- اشیای داده ای را پالایش کرده و مجموعه ای از انتزاعات داده ای را توسعه میدهد.

- صفات اشیا داده ای را یک یا چندساختمان داده ای پیاده سازی می کند .

- ساختمان داده ای برای اطمینان از ارتباط های مناسب باز نگری میشود .

- ساختمان داده ها تا حد ممکن ساده سازی میشود.

طراحی مبتنی بر داده ها یک روش طراحی معماری نرم افزار است که به راحتی امکان گذر از مرحله تحلیل نیاز به طراحی توصیف ساختمان برنامه را می دهد.

**7-3-1مدل ساز ی داده ،ساختار داده،پایگاه داده و انبار داده**

اشیای داده ای که در طول تحلیل نیازهای نرم افزاری تعریف شده اند،با استفاده از نمودارهای ارتباط –موجودیت و فرهنگ داده ها مدل سازی میشوند.کار طراحی داده ها، عناصر مورد نیاز مدل نیاز ها را در سطح مولفه نرم افزاری به ساختمان داده ها وقتی لازم باشد،معماری پایگاه داده ها را در سطح برنامه کاربردی ،تبدیل میکند(در این مورد به مباحث مربوط به پایگاه داده ها مراحعه کنید).مجموعه ای از داده های موضوع گرا،یکپارچه و وابسته به زمان که برای پشتیبانی تصمیم گیری های مدیریتی مورد استفاده قرار میگیرد،انبار داده ها نامیده میشود.

**7-3-2 طراحی تفصیلی داده ها**

طراحی داده ها در این سطح روی ارائه و نمایش ساختمان داده هایی متمرکز میشود که مستقیمأ توسط یک یا چند مولفه نرم افزاری قابل دسترسی هستند.Wasserman مجموعه اصولی را برای مشخصه های داده ای پیشنهاد کرده است که ممکن است برای مشخص کردن و طراحی چنین ساختارهایی به کار روند.توجه داشته باشید که مدل های تحلیل نیازها و طراحی اغلب هم پوشانی دارند.

اصول تحلیل نظام مند به کاررفته در مورد عملکرد و رفتارباید در مورد داده ها نیز به کاررود.

تمتم عملیات و ساختمان داده ها باید شناسایی شوندد درطراحی یک ساختار داده ای کارامد باید عملیاتی را در نظر گرفت که روی ساختار داده ها صورت میگیرد.

یک فرهنگ داده ای به وجود اورده واز ان برای تعریف طراحی داده ها و برنامه استفاده کنید.

تصمیمات مربوط به سطوح پاین طراحی داده ها را تا اواخر فرایند طراحی به تعویق بیندازید.

نمودار ساختاری داده ها تنها باید برای پیمانه هایی شناخته شده باشد که مستقیمأ از داده های موجود در ساختار استفاده میکنند.

کتابخانه ای از ساختارهای داده ای مفید و عملیاتی که ممکن است درمورد انها به کاررود،ایجاد کنید.

طراحی نرم افزار و زبان برنامه نویسی باید خصوصیات و شناسایی انواع داده های انتزاعی را پشتیبانی کند.

**7-4سبک های معماری نرم افزار**

**معماری مبتنی بر مخزن داده ها :** یک مخزن داده ای در مرکز این معماری قرار دارد و اغلب توسط دیگر اجزایی که به هنگام سازی ،افزودن ،حذف یا کارهای دیگر اصلاحی را در مورد مخزن انجام میدهند،قابل دسترسی است.اغلب این معماری را معماری داده گرا نیز می نامند.

**معماری مبتنی بر جریان داده:** این معماری وقتی به کار گرفته میشود که قرار است داده های ورودی از طریق اجرای یکسری پردازش های محاسباتی و اعمال تغییرات ،به داده ای خروجی تبدیل شوند.

**معماری فراخوانی و بازگشت:** این سبک از معماری طراح نرم افزار را قادر میکند تا به ساختار برنامه ای دست یابد که از نظر اصلاح و ارزیابی نسبتأ ساده است.

**معماری های شی ءگرا** : اجزای یک سیستم در بر گیرنده داده ها و عملیاتی هستند که باید برای تغییر داده ها مورد استفاده قرار گیرند.ارتباط و هماهنگی بین اجزا از طریق عبور پیام ها حاصل میشود.

**معماری های لایه ای** : تعدادی لایه مختلف تعریف شده اند که هر کدام به عملیاتی دست می یابند که به طور گسترده ای به مجموعه دستورات ماشین نزدیکتر میشوند.در لایه خارجی تر اجزا در خدمت عملیات واسط کاربر هستند.درلایه داخلی تر اجزا وظیفه ارتباط با سیستم عامل را انجام میدهند.لایه های میانی خدمات استفاده،بهره برداری و عملیات کارکردی نرم افزار را مهیا میکنند.

**7-5 فرایند طراحی معماری**

فرایند طراحی معماری نرم افزاربا انجام فعالیت های تحلیل و طراحی به صورت تکراری و به شرح زیر صورت میگیرد:

**جمع اوری (سناریوها)طرح ها:** مجموعه ای از مورد های کاربرد که نماینگر عملکرد سیستم از دیدگاه کاربر باشند.

**به دست اوردن نیازها،،محدودیت ها و توصیف محیط:** این اطلاعات به عنوان بخشی از مهندسی نیازها محسوب شده و برای اطمینان از اینکه تمام نگرانیهای مشتری،کاربرو سهامداران مورد توجه قرار گرفته اند،جمع اوری و تحلیل میشوند.

**توصیف سبکها /الگوهای معماری انتخاب شده برای بررسی طرح ها ونیازها :** این سبک ها را باید با استفاده از دیدگاه های معماری زیر تشریح کرد :

* دید گاه پیمانه
* دید گاه پردازشی
* دید گاه جریان داده ای

**ارزیابی صفات خاصه کیفی با در نظر گرفتن هر یک به طور جداگانه:** صفات کیفی برای ارزیابی طراحی معماری شامل قابلیت اطمینان،کارایی، امنیت، قابلیت نگهداری، انعطاف پذیری،ازمون پذیری، قابلیت حمل، قابلیت استفاده مجدد و قابلیت کارکرد تعاملی مولفه هاست که با توجه به زمان و میزان تأثیر هر یک بر سیستم مورد ارزیابی قرار میگیرند.

**شناسایی حساسیت صفات خاصه بر معماری در یک سبک به خصوص :** این کار با ایجاد تغییرات کوچکی در ساختا رو تعیین چگونگی حساسیت یک صفت خاصه کیفی مثلا کارایی نسبت به تغییر ،انجام میگیرد.صفت خاصه ای که تا حد زیادی تحت تأثیر تنوع معماری قرار دارد ،نقاط حساسیت نامیده میشود.

**نقد معماریهای پیشنهادی با استفاده از تحلیل حساسیت(مرحله 5) و تعیین معماری نهایی.**

**7-5-1 پیچیدگی معماری**

یک روش مفید برای ارزیابی پیچیدگی کلی معماری پیشنهادی ،عبارت است از در نظر گرفتن وابستگی میان اجزای درونی معماری .

این وابستگی ها ناشی از جریان کنترل/ اطلاعات درون سیستم هستند.

**Zhao سه نوع وابستگی را ارائه میدهد:**

* **وابستگی های مشترک** : نمایانگر ارتباطات وابسته در میان مصرف کنندگانی است که از منبعی یکسان استفاده کرده یا تولیدکنندگانی که برای مصرف کنندگانی یکسان تولید میکنند.مثلا در مورد دو جزء u و v اگر u و v هردو به یک سری داده های سرتاسری یکسانی رجوع کنند،یک واسطه وابستگی مشترک بین u و v وجود دارد.
* **وابستگی های جریان** : نمایانگر ارتباطات وابستگی میان تولید کننده و مصرف کننده های منبع است.در مورد دو جزء u وv اگر باید u قبل از جریان یافتن کنترل در v تکمیل شودیا اگر u به وسیله پارامترهایی با v ارتباط برقرار کند،در این صورت یک جریان وابستگی بین این دو وجود دارد.
* **وابستگی های محدود شده** : نمایانگر محدودیت ها و قیودی در جریان نسبی کنترل میان مجموعه ای از فعالیت ها هستند. مثلا در مورد دو جزء u و v انها نمیتوانند در یک اجرا شوند،پس یک واسطه وابستگی محدود و مقید شده بین انها وجود دارد.

وابستگی های مشترک و جریان که مورد توجه Zaho قرار گرفته است از بعضی جهات شبیه مفهوم انسجام و اتصال هستند.

**7-6 نگاشت نیازها به معماری نرم افزار**

طراحی ساخت یافته اغلب به عنوان یک روش طراحی مبتنی بر جریان داده معرفی میشود،زیرا به کمک ان میتوان براحتی از نمودار جریان داده به معماری نرم افزار دست یافت . گذار از جریان داده ها به ساختار برنامه با انجام یک فرایند شش مرحله ای زیر صورت میگیرد :

* تعیین نوع جریان اطلاعات
* مشخص کردن مرز جریان
* نگاشت نمودار جریان داده به ساختمان برنامه
* تعیین سلسله مراتب کنترل
* پالایش و بازنگری معماری حاصل با استفاده از معیارها ،اصول و قواعد طراحی
* توصیف و تدوین معماری نهایی

**7-6-1 تعیین نوع جریان اطلاعات**

دو نوع جریان تبدیلی و تراکنشی اطلاعات وجود دارد که در ادامه به نحوه نگاشت انها به معماری نرم افزار می پردازیم.

**جریان تبدیلی :** در جریان تبدیلی روال به صورت شکل 5-7 نشان داده شده است.

**جریان تراکنشی :** در جریان تراکنشی ،جریانی وارد سیستم میشود و سیستم انتخاب میکند که از کدام شاخه خارج شود.

**فصل 8**

**طراحی واسط کاربر**

نقشه يكخانه بدون نشان دادن درها پنجرهها مسيرهاي لوله كشي سيم كشي و گاز كشي كامل نمي شود.در نرم افزارهايرايانه هاي طراحي واسط ها حكم حكم طراحي درها و پنجره ها و اتصالات دارد .

طراحي واسط ها بر سه حوزه به شرح زير تاكيد دارد:

طراحي واسط بين اجزاي نرم افزار

طراحي واسط بيننرم افزار و ساير توليد كنندگان ومصرف كنندگان و اطلاعات غير انساني

طراحي واسط بين انسان و رايانه

1-8 قواعد طلايي memdal

Mendal در كتاب خود با عنوان طراحي واسطسه قانون طلايي زي را مطرح مي كند:

اين قوانين طلايي عمدتا به عنوان مجموعه اي از اصول طراحي واسط كاربر مطرح هستند كه فعاليت مهم طراحي واسط نرم افزاري را هدايت مي كند.

1**-1-8 واگذاري كنترل به كاربر**

در اين قانون رعايت نكات زير الزامي است:

* تعیین شیوه های تعاملی به نحوی که کاربر را مجبور به اعمال غیر ضروری یا نامطلوب نکند.
* ایجاد تعامل انعطاف پذیر در ارتباط با کاربر
* امکان ایجاد وقفه و خنثی سازی(بازگشت) در تعامل با کاربر
* کارامد ساختن تعامل همراه با پیشرفت سطوح مهارتی و امکان سفارشی کردن
* مخفی کردن موارد فنی داخلی از دید کاربران عادی
* طراحی تعامل مستقیم با اشیایی که روی صفحه نمایش ظاهر میشوند.

**8-2-1 کاستن از بار حافظه کاربر**

در این قانون نیز رعایت نکات زیر الزامی است:

* کاهش باردرحافظه کوتاه مدت
* ایجاد پیش گزیده های معنی دار
* تعیین میانبرهای شهودی
* اشکارسازی اطلاعات به شیوه ای تدریجی
* طرح بصری واسط باید بر اساس استعاره جهان واقعی استوار باشد.

مثال رایج در بسیاری از برنامه های کاربردی واژه پردازی ،عمل "خط زیر " است.این کارکرد یکی از چندین کارکردی است که در پنجره قالب بندی متن قرار دارد.هر چند که تمامی امکانات "خط زیر" فهرست نمیشوند ولی کاربر باید ابتدا "خط زیر" را انتخاب کند و سپس تمامی امکانات "خط زیر" (خط زیر تک خطی ،دو خطی و نقطه چین)به نمایش در می ایند.

**8-3-1 سازگاری واسط**

در این قانون نیز رعایت نکات زیر الزامی است :

قرار دادن عمل فعلی در یک بافت معنی دارتوسط کاربر.بسیاری از واسط ها ،لایه های پیچیده تعاملی را با تصاویر زیادی در صفحه نمایش پیاده سازی میکنند .

تهیه نشانگرها (مثل عناوین پنجره ،شمایل های گرافیکی کد گذاری ثابت رنگ)،از این نظر که کاربر را قادر کند تا محیط کاری موجود را بشناسد ،اهمیت دارند.

حفظ ثبات در خانواده برنامه های کاربردی. مجموعه ای از برنامه های کاربردی (یا محصولات)باید همگی قوانین یکسان طراحی را پیاده سازی کنندوبه نحوی که سازگاری در تمامی تعاملات و محاوره ها حفظ شود.

**اگر مدل های تعاملی پیشین انتظاراتی رادر کاربر به وجود آورده اند،تا زمانی که دلیل قانع کننده ای نداریذ از انجام تغییرات خودداری کنید.** پس از تبدیل یک ترتیب خاص تعاملی یا یک استاندارد عملی (مثل کاربردAlt+s برای ذخیره فایل) **از** تغییر ان خودداری کنید.زیرا کاربردرمواجهه با هر برنامه کاربردی دیگر همین انتظاررا دارد و انجام یک تغییر سبب اشفتگی و سردر گمی او میشود.

اصول طراحی واسط راهنمای اصلی فعالیت های یک مهندس نرم افزار به شمار میروند.

**8-2 طراحی واسط کاربر**

طراحی واسط کاربر با ایجاد مدل های مختلف کارکرد سیستم اغاز می شود.سپس وظایف انسانی و رایانه ای لازم برای تحقق کارکرد سیستم توصیف میشوند.موضوعات طراحی که در تمام طراحی های واسط کاربر دارند مد نظر قرار می گیرند.برای الگو سازی و پیاده سازی نهایی مدل طراحی واسط ها ،ابزارهایی به کار می روند و نتیجه از لحاظ کیفی ارزیابی می شود.

**8-2-1 مدل های طراحی واسط**

هنگام طراحی یک واسط کاربر، چهار مدل مختلف مطرح هستند :

مهندسی نرم افزار مدل طراحی را ایجاد می کند.

مهندس فاکتورهای انسانی (یا مهندس نرم افزار)مدل کاربر را تعیین میکند.

کاربر نهایی یک تصویر ذهنی می سازد که غالبأ مدل ذهنی کاربر را به وجود می اورد.

پیاده کنندگان سیستم نیز یک تصویر سیستم ایجاد می کنند.

متأسفانه ممکن است هر یک از این مدل ها تفاوت های قابل ملاحظه ای با یکدیگر داشته باشند .نقش طراح واسط رفع اختلافات و به دست اوردن یک نمایش منسجم و سازگار از واسط هاست.

**مدل طراحی** کل سیستم، تلفیقی از نمایش داده ها ، معماری ،واسط و بازنمایی رویه ای نرم افزار است .تعیین نیازها ممکن است محدودیت های خاصی را مطرح کند.که به تعیین کاربرسیستم کمک میکنند.اما طراحی واسط اغلب با مدل طراحی یکسان میشود.

**مدل کاربر،**نمایی از کابران نهایی سیستم را ترسیم میکند. برای ساخت یک واسط کاربر،موثر،تمام کار طراحی باید با درک درستی از کاربران مورد نظر از جمله مشخصات سن ،جنسیت،توانایی های جسمی،سابقه تحصیلی،فرهنگی یا قومی ،انگیزه اهداف و شخصیت انها اغاز شود.به علاوه کاربران را میتوان در گروه های زیر طبقه بندی کرد :

**کابران مبتدی.ا**از دانش نحوی سیستم برخوردار نیستند و دانش معنایی انها از برنامه کاربردی یا کاربرد رایانه به طور کلی اندک است.

**کاربران مطلع و دایمی** .دانش نحوی و معنایی مناسب دارند که اغلب به مشخصه کاربرماهر منجر میشود.یعنی کابرانی که به دنبال میانبرها و حالت های اختصاری تعامل هستند.

**مدل ذهنی کاربر(برداشت سیستم)،**تصویری از سیستم است که کاربر نهایی در ذهن خود ایجاد میکند.مثلا اگر از کاربر یک واژه پرداز خاص بخواهیم تا کارکرد ان را توصیف کند،پاسخ او بر اساس درک او از سیستم است .صحت توصیف بستگی به شرح حال کاربر و اشنایی کلی با نرم افزار در حیطه برنامه کاربردی دارد.

**تصویر سیستم،** ترکیبی از نمود بیرونی سیستم رایانه ای همراه تمام اطلاعات پشتیبان است که نحو و معنای شناسایی سیستم را توصیف میکنند.

زمانی که مدل های تصویر سیستمی و برداشت سیستمی یکسان باشند ،عمومأ کاربران با نرم افزار احساس راحتی کرده و به طور موثر ان را به کار می برند.

برای دستیابی به ترکیب این مدل ها ،مدل طراحی باید طوری توسعه یابد که با اطلاعات موجود در مدل کابر سازگارباشد و تصویر سیستم نیز درستی اطلاعات نحوی و معنایی مربط به واسط را دقیقأ منعکس کند.

**8-3 فرایند طراحی واسط کاربر**

فرایند طراحی واسط های کاربر ،یک فرایند تکراری است و با استفاده از مدل حلزونی ،قابل ارائه است .

**8-3-1 تحلیل نیازهای کاربر،وظایف،محیط و مدل سازی**

تحلیل اولیه بر سابقه کاربرانی تکیه دارد که با سیستم تعامل میکنند.سطح مهارت افراد،شناخت حرفه و سازمان و سطح امادگی انان شناسایی و تحلیل می گردد و گروه های کاربر متفاوت تعریف می شود.در واقع مهندس سیستم سعی میکند تا برداشت سیستم را برای هر طبقه از کاربران شناسایی کند. تحلیل محیط کاربر بر محیط کار فیزیکی تکیه دارد .اطلاعات حاصل از این فعالیت به منظور ایجاد یک مدل تحلیل برای واسط کاربر استفاده میشود.

**8-3-2 طراحی واسط**

هدف طراحی واسط،تعریف یک مجموعه از اشیا و عملیات واسط و نمایش انها در صفحه نمایش است که کاربر را قادر به انجام تمام وظایف تعریف شده می سازد،به طوری که همه اهداف قابلیت استفاده را برای یک سیستم براورده سازد.

**8-3-3 پیاده سازی و ساخت واسط**

این فعالیت با ایجاد یک نمونه اولیه اغاز میشود که ارزیابی سناریوهای کاربر را امکان پذیر میسازد.

**8-3-4 اعتبار سنجی واسط**

اعتبار سنجی برموارد زیر تکیه دارد :

* توانایی واسط در پیاده سازی تمامی وظایف ،انجام تمام وظایف و دستیابی به کلیه خواسته های عمومی کاربر،
* میزان سهولت استفاده و فراگیری واسط،
* برداشت کاربران از واسط به عنوان یک ابزار مفید در کارهای انان.

**8-4 مسایل طراحی واسط ها**

در حین تکمیل طراحی واسط کاربر،چهار مسئله معمول طراحی تقریبأ همیشه سطحی تلقی میشوند که باید به انها توجه کرد:

- زمان پاسخگویی سیستم

- تسهیلات کمکی کاربر

- خطا گردانی اطلاعات

- برچسب گذاری فرمان

متأسفانه بسیاری از طراحان با تأخیر این مسایل را در فرایند طراحی مد نظر قرار میدهند و همین امر موجب بروز مشکلات عدیده میشود.تکرار غیرضروری ،تأخیرهای پروژه و نارضایتی مشتری، اغلب از پیامدهای حاصله است.پس بهتر ان است که هر یک از مسایل در اغاز طراحی نرم افزار و هنگام انجام راحت تغییرات و پایین بودن هزینه ها مورد توجه قرار گیرند.

**زمان پاسخگویی سیستم** : از زمانی که کاربر یک عمل کنترلی را انجام میدهد تا زمان پاسخگویی نرم افزار با یک اقدام یا خروجی مطلوب اندازه گیری می شود.

زمان پاسخگویی سیستم دو ویژگی مهم دارد:**طول و تغییر پذیری** . اگر طول پاسخگویی سیستم بسیار طولانی باشد،ناامیدی و فشار روی کاربر،نتیجه ای اجتناب ناپذیر است. البته زمان پاسخگویی بسیار کوتاه نیز میتواند مضر باشد. پاسخگویی سریع ممکن است موجب عجله کاربر و بنابراین باعث انجام اشتباه از سوی او شود.تغییر پذیری به انحراف از زمان میانگین پاسخگویی اشاره داشته و از خیلی جهات ،مهم ترین مشخصه زمان پاسخگویی به شمار می رود.

تغییر پذیری کم،حتی در صورت طولانی بودن زمان پاسخگویی ،به کاربر امکان میدهد تا به طریق مناسب تعامل را با سیستم برقرار کند.به عنوان مثال ،پاسخ یک ثانیه ای به یک فرمان ،به پاسخگویی متغییر بین یک تا سه ثانیه ،ترجیح دارد.

**تسهیلات کمکی کاربر :** تقریبأ تمامی کاربران یک سیستم تعاملی رایانه ای گه گاه به کمک نیاز دارند.دو نوع امکانات کمکی عبارتند از : یکپارچه و افزودنی .

**تسهیلات کمکی یکپارچه :** از اغازدر داخل نرم افزار طراحی میشود، این نوع تسهیلات کمکی غالبأ به متن حساس هستندو کاربر را قادر میکنند تااز میان موضوعات مرتبط با اعمال در حال اجرا اقدام به انتخاب کند واضح است که این امر زمان لازم برای دریافت کمک توسط کاربر را کاهش داده و کارپسندی واسط را افزایش می دهد.

**تسهیلات کمکی افزودنی:** پس از ساخت سیستم ،این گونه تسهیلات کمکی به نرم افزار افزوده می شود ،ممکن است کاربر مجبور شود برای یافتن راهنمایی صحیح و مناسب فهرستی با صدها موضوع را جستجو کند و اغلب با شروع نادرست اطلاعات غیر مرتبط را دریافت کند .

شکی نیست که امکانات کمکی یکپارچه بر نوع افزودنی ان برتری دارد.

رعایت چند مسئله طراحی در ارائه تسهیلات راهنما به کاربر به شرح زیر الزامی است :

* ایا راهنما برای تمام عملکردهای سیستم ودر همه اوقات تعامل ،در دسترس است ؟
* کاربر چگونه درخواست کمک و راهنمایی میکند؟
* راهنما چگونه ارائه میشود؟
* کاربر چگونه به تعامل عادی باز می گردد؟
* اطلاعات راهنما چه ساختاری دارند؟

**خطا گردانی اطلاعات**: هنگام بروز خطا،پیغام های خطا و هشدار ،اخبار بدی است که به کاربران سیستم های تعاملی ارائه می شود.در بدترین حالت پیغام های خطا و اخطارها ،اطلاعات بی فایده یا گمراه کننده را منتقل کرده و تنها باعث تشدید ناکامی کاربر می شوند.تعداد کاربران رایانه که با خطایی از نوع زیر مواجه نشده باشند ،بسیار اندک است.

به طور کلی هر پیغام خطا یا هشدار تولید شده توسط یک سیستم محاوره ای باید داری ویژگی های زیر باشد :

* پیام باید مشکل را به زبانی شرح دهد که کاربر قادر به درک ان باشد.
* پیام باید حاوی یک توصیه سازنده برای رهایی از وضعیت خطا باسد.
* پیام باید هرگونه تبعات منفی خطا (مثلا فایل های داده ای مخدوش شده )را خاطر نشان کند تا کاربر بتواند انها را کنترل کند.
* پیام باید با یک نشانه سمعی یا بصری همراه باشد.
* پیام باید قضاوت گونه باشد.

**برچسب گذاری فرمانها :** فرمان تایپ شده زمانی رایج ترین شیوه محاوره بین کاربر و نرم افزار بود ودر هرنوع برنامه کاربردی به طور معمول به کار میرفت.امروزه استفاده از واسط های پنجره ای ،اشاره و انتخاب ،کاربرد فرامین تایپ شده را کاهش داده است.اما بسیاری از کاربران ماهر همچنان شیوه ارتباطی مبتنی بر فرمان را ترجیح می دهند.هنگام انتخاب فرامین تایپ شده به عنوان نوعی شیوه محاوره برخی مسایل طراحی باید مورد توجه قرار گیرند :

* ایا هر یک از گزینه های موجود در پنجره یک فرمان متناظر دارند؟
* فرامین چه شکلی خواهند داشت؟ امکانات موجود عبارتند از :توالی کنترل (مثل Alt+p ) کلیدهای تابعی،واژه تایپ شده.
* یادگیری و به خاطر سپردن فرامین تا چه حد دشوار است؟در صورت فراموشی یک فرمان چه می توان کرد؟
* ایا کاربر میتواند فرمان هارا به سلیقه خویش مختصر ،کوتاه و به دلخواه خود تنظیم کند؟

**8-5 ابزارهای پباده سازی واسط**

برای انجام روش تکراری طراحی واسط های نرم افزاری ،انواع ابزارهای طراحی واسط و ساخت نمونه اولیه وجود دارند.

این ابزارها بادعنوان سیستم توسعه واسط کاربر نامیده میشوند.این سیستم با استفاده از مولفه های نرم افزاری ،راهکاری برای موارد زیر فراهم می اورد :

* مدیریت دستگاه های ورودی (مثل ماوس یا صفحه کلید)
* اعتبار سنجی ورودی کاربر
* کنترل خطا و نمایش پیام های خطا
* فراهم اوردن بازخورد(مثل بازتاب ورودی خودکار)
* فراهم اوردن راهنما و پیغام ها
* کنترل پنجره ها و فیلدها ،حرکت در داخل پنجره ها
* برقراری ارتباط میان نرم افزار کاربردی و واسط
* مجاز کردن کاربر به سفارشی کردن واسط

**8-6 ارزیابی طراحی واسط**

پس از ایجاد الگوی عملی واسط کاربر،این مدل باید موردارزیابی قرار گیرد تا معلوم شودایا نیازهای کاربر را برطرف می کند یا خیر؟ارزیابی می تواند در یک طیف رسمی صورت گیرد که گستره ان با انجام ازمونی غیر رسمی که در ضمن ان کاربر بازتابی بدون فکر قبلی دارد شروع شود و به مطالعه رسمی ختم می گردد که از روش های اماری برای ارزیابی پرسشنامه تکمیل شده توسط کاربران نهایی استفاده می کند .

برخی از معیارهای ارزیابی را هنگام بررسی های اولیه طراحی می توان اعمال کرد :

* **طول و پیچیدگی مشخصات مکتوب سیستم وواسط ان:**بیانگرزمان یادگیری لازم توسطکاربران سیستم است.
* **تعداد وظایف تعیین شده کاربر و میانگین اعمال درهر کار**:نشان دهنده محاوره و کارایی کلی سیستم است.
* **تعداد اعمال،وظایف و ضعیت های سیستم که در مدل طراحی تعیین** شده،به بار حافظه کاربران سیستم دلالت دارد.
* **روش رتباط واسط،امکانات کمکی و خطا گردانی**،در کل بیانگر پیچیدگی واسط و میزان پذیرش از سوی کاربر است.

پس از ساخته شدن اولین مدل طراح میتواند مجموعه ای از داده های کیفی و کمی را که به ارزیابی واسط کمک میکنند جمع اوری کند.ارزش ویژگی های مورد نظر را میتوان از طریق پرسشنامه هایی که بین کاربران توزیع می شود تعیین کرد.

نمونه ای از سوالات نیز به صورت زیر هستند :

* ایا نمادها خود توصیف هستند؟اگر نیستند کدام نماد نیاز به توصیف دارد؟
* ایا عملیات را به اسانی میتوان به یاد سپرد و اجرا کرد ؟
* تاکنون از چند عمل متفاوت استفاده کرده اید ؟سهولت فراگیری عملیات اصلی سیستم تا چه حدی است؟
* در مقایسه با واسط های دیگری که استفاده کرده اید ،به این واسط چه امتیازی میدهید؟
* جمع اوری این اطلاعات ،مبنایی را برای تحلیل و جمع بندی نهایی نظرات کاربران در مورد واسطه های نرم افزار فراهم میکند.