

دانشگاه پیام نور مشهد

# جزوه درس مهندسی نرم افزار ۲



مصطفی جهانگیر

[www.mjahangir.ir](http://www.mjahangir.ir)

[mjahangirf@gmail.com](mailto:mjahangirf@gmail.com)

به نام آن که جان را فکرت آموخت

جزوه درس

# مهندسی نرم افزار ۲

مصطفی جهانگیر

mjahangirf@gmail.com

www.mjahangir.ir

اردیبهشت ۱۳۹۳

## فهرست مطالب

|                                         |           |
|-----------------------------------------|-----------|
| <b>جلسه ۲- طراحی برنامه های تحت وب</b>  | <b>۵</b>  |
| ۱. کیفیت طراحی برنامه های تحت وب        | ۵         |
| ۲. شاخص های کیفیت برنامه های تحت وب     | ۵         |
| ۳. اهداف طراحی برنامه تحت وب            | ۵         |
| ۴. هرم طراحی برنامه های تحت وب          | ۶         |
| ۵. طراحی واسط برنامه تحت وب             | ۶         |
| ۶. طراحی زیبایی شناسانه (گرافیکی)       | ۷         |
| ۷. طراحی محتوا                          | ۷         |
| ۸. طراحی معماری                         | ۸         |
| ۹. طراحی گشت و گذار                     | ۱۰        |
| <b>جلسه ۳- مفاهیم کیفیتی</b>            | <b>۱۲</b> |
| ۱. مساله چیست؟                          | ۱۲        |
| ۲. کیفیت چیست؟                          | ۱۲        |
| ۳. کیفیت نرم افزار                      | ۱۲        |
| ۴. ابعاد کیفیتی                         | ۱۳        |
| ۵. معضل کیفیت نرم افزار                 | ۱۴        |
| ۶. دستیابی به کیفیت نرم افزار           | ۱۵        |
| <b>جلسه ۴- تکنیک های مرور نرم افزار</b> | <b>۱۷</b> |
| ۱. مقدمه                                | ۱۷        |
| ۲. تأثیر نقایص نرم افزار بر هزینه ها    | ۱۷        |
| ۳. تشدید نقایص و حذف آن ها              | ۱۷        |
| ۴. معیارهای مرور و کاربردهای آن ها      | ۱۹        |
| ۵. مرورها: یک طیف رسمیت                 | ۲۰        |
| ۶. مرورهای غیر رسمی                     | ۲۰        |
| ۷. مرورهای فنی رسمی                     | ۲۱        |
| <b>جلسه ۵- تضمین کیفیت نرم افزار</b>    | <b>۲۲</b> |
| ۱. مقدمه                                | ۲۲        |
| ۲. عناصر تضمین کیفیت نرم افزار          | ۲۲        |
| ۳. وظایف، اهداف و معیارهای SQA          | ۲۳        |
| ۴. تضمین کیفیت آماری نرم افزار          | ۲۴        |

|                                                |                                             |    |
|------------------------------------------------|---------------------------------------------|----|
| ۵.                                             | قابلیت اطمینان نرم افزار                    | ۲۵ |
| ۶.                                             | استاندارد کیفیتی ISO 9001                   | ۲۵ |
| ۷.                                             | طرح SQA                                     | ۲۵ |
| <b>جلسه ۶- راهبردهای آزمون نرم افزار</b>       |                                             |    |
| ۱.                                             | مقدمه                                       | ۲۶ |
| ۲.                                             | روشی راهبردی برای آزمون نرم افزار           | ۲۶ |
| ۳.                                             | راهبردهای آزمون برای نرم افزارهای سنتی      | ۲۸ |
| ۴.                                             | راهبردهای آزمون برای نرم افزارهای شیء گرا   | ۲۸ |
| ۵.                                             | راهبردهای آزمون برای نرم افزارهای تحت وب    | ۲۸ |
| ۶.                                             | آزمون اعتبار سنجی                           | ۲۸ |
| ۷.                                             | آزمون سیستم                                 | ۲۸ |
| ۸.                                             | هنر اشکال زدایی                             | ۲۹ |
| <b>جلسه ۷- آزمون برنامه های کاربردی سنتی</b>   |                                             |    |
| ۱.                                             | مقدمه                                       | ۳۰ |
| ۲.                                             | مبانی آزمون نرم افزار                       | ۳۰ |
| ۳.                                             | دیدگاه های درونی و بیرونی نسبت به آزمون     | ۳۰ |
| ۴.                                             | آزمون جعبه سفید-جعبه شیشه ای                | ۳۱ |
| ۵.                                             | آزمون جعبه سیاه                             | ۳۱ |
| ۶.                                             | آزمون محیط ها و معماری ها                   | ۳۲ |
| <b>جلسه ۸- آزمون برنامه های شیء گرا</b>        |                                             |    |
| ۱.                                             | مقدمه                                       | ۳۳ |
| ۲.                                             | وسعت بخشیدن به دیدگاه آزمون                 | ۳۳ |
| ۳.                                             | آزمون مدل های تحلیل شیء گرا و طراحی شیء گرا | ۳۳ |
| ۴.                                             | راهبردهای آزمون شیء گرا                     | ۳۳ |
| ۵.                                             | روش های آزمون شیء گرا                       | ۳۴ |
| ۶.                                             | روش های آزمون قابل اجرا در سطح کلاس ها      | ۳۴ |
| <b>جلسه ۹- آزمون برنامه های کاربردی تحت وب</b> |                                             |    |
| ۱.                                             | طرح مساله                                   | ۳۵ |
| ۲.                                             | مفاهیم آزمون برای برنامه های تحت وب         | ۳۵ |
| ۳.                                             | راهبرد و فرایند آزمون                       | ۳۶ |
| ۴.                                             | راهبرد آزمون واسط                           | ۳۶ |
| ۵.                                             | آزمون در سطح مولفه ها                       | ۳۷ |

|    |                                           |    |
|----|-------------------------------------------|----|
| ۳۷ | آزمون پیکربندی                            | ۶  |
| ۳۸ | آزمون امنیت و کارایی                      | ۷  |
| ۴۰ | <b>جلسه ۱۰- طراحی مبتنی بر الگوها</b>     |    |
| ۴۰ | مقدمه                                     | ۱  |
| ۴۰ | الگوهای طراحی                             | ۲  |
| ۴۰ | انواع الگوها                              | ۳  |
| ۴۱ | چارچوب ها                                 | ۴  |
| ۴۱ | توصیف الگوها                              | ۵  |
| ۴۲ | مخازن و زبان های الگو                     | ۶  |
| ۴۲ | طراحی مبتنی بر الگوها در حیطه             | ۷  |
| ۴۲ | وظایف طراحی                               | ۸  |
| ۴۳ | ساخت جداول سازمان دهی الگو                | ۹  |
| ۴۳ | الگوها                                    | ۱۰ |
| ۴۵ | <b>جلسه ۱۱- مدل سازی و واریسی رسمی</b>    |    |
| ۴۵ | مقدمه                                     | ۱  |
| ۴۶ | راهبرد اتاق تمیز                          | ۲  |
| ۴۸ | مشخصات عملیاتی                            | ۳  |
| ۴۹ | طراحی و آزمون اتاق تمیز                   | ۴  |
| ۵۰ | مفاهیم روش های رسمی                       | ۵  |
| ۵۰ | زبان های تعیین مشخصات رسمی                | ۶  |
| ۵۲ | <b>جلسه ۱۲- مدیریت پیکربندی نرم افزار</b> |    |
| ۵۲ | مقدمه                                     | ۱  |
| ۵۲ | مدیریت پیکربندی نرم افزار                 | ۲  |
| ۵۳ | مخزن SCM                                  | ۳  |
| ۵۴ | ویژگی های SCM                             | ۴  |
| ۵۵ | فرایند SCM                                | ۵  |
| ۵۶ | <b>جلسه ۱۳- معیارهای محصول</b>            |    |
| ۵۶ | نگاهی گذرا                                | ۱  |
| ۵۶ | چارچوبی برای معیارهای تکنیکی محصول        | ۲  |
| ۵۷ | معیارهایی برای مدل خواسته ها              | ۳  |
| ۵۷ | معیارهایی برای مدل طراحی                  | ۴  |

## جلسه ۲- طراحی برنامه های تحت وب

### ۱. کیفیت طراحی برنامه های تحت وب

#### ۱.۱. کیفیت طراحی برنامه های تحت وب

- طراحی، یک فعالیت مهندسی است که به ایجاد محصول با کیفیت بالا می انجامد.
- اگر محصولات طوری طراحی شوند که بهتر با تمایلات رفتاری انسان ها همخوانی داشته باشند، در آن صورت رضایت مندی بیشتر می شود. (سوزان واین شنک)

### ۲. شاخص های کیفیت برنامه های تحت وب

#### ۲.۱. کیفیت طراحی برنامه های تحت وب

| توضیح                                                                                                                                                                             | شاخص های کیفیتی    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| قابلیت درک سایت در تمام جهان - ویژگی های زیبایی شناختی و واسط کاربر - ویژگی های راهنما                                                                                            | قابلیت استفاده     |
| قابلیت جستجو و بازیابی - ویژگی های گشت و گذار و مرورگری                                                                                                                           | قابلیت عملیاتی     |
| رهایی از وضعیت خطا - پردازش صحیح - اعتبارسنجی و بازیابی ورودی کاربر                                                                                                               | قابلیت اطمینان     |
| کارایی زمان پاسخ - سرعت لود صفحات - سرعت لود اقلام گرافیکی                                                                                                                        | بازدهی             |
| سهولت تصحیح - انطباق پذیری - بسط پذیری                                                                                                                                            | قابلیت نگهداری     |
| رد دستیابی غیر مجاز - جلوگیری از حمله نفوذگران                                                                                                                                    | امنیت              |
| میزان زمان در دسترس بودن برنامه تحت وب (۳۶۵/۷/۲۴) - قابل دسترس در همه مرورگرها                                                                                                    | دسترس پذیری        |
| ارایه سرویس به تعداد زیادتری از کاربران                                                                                                                                           | گسترش پذیری        |
| نخستین برنامه ای که یک بخش مشخصی از بازار را پاسخ دهد، کاربران بیشتری جذب می کند.                                                                                                 | زمان عرضه به بازار |
| برآورده کردن نیازهای محتوایی کاربران - زمینه علمی و سوابق نویسندگان محتوا - به روز بودن محتوا<br>منحصر به فرد بودن محتوا و تکراری نبودن آن - ارزشمند بودن محتوا - سازماندهی محتوا | محتوا              |

### ۳. اهداف طراحی برنامه تحت وب

#### ۳.۱. اهداف طراحی

| توضیح                                                                                     | اهداف             |
|-------------------------------------------------------------------------------------------|-------------------|
| ○ اعتدال در حجم محتوای سایت. محتوای زیادی، تصاویر فراوان، پویانمایی های بیجا نامناسب است. | سادگی             |
| ○ استفاده از قابلیت های برنامه باید آسان و درک آن ها ساده باشد.                           | Simplicity        |
| ○ محتوا باید به صورت سازگار و هماهنگ در برنامه ساخته شود. (فونت، رنگ و ...)               | سازگاری           |
| ○ طراحی گرافیکی، طراحی معماری، طراحی واسط و سازوکارهای گشت و گذار هماهنگ باشند.           | (Consistency)     |
| ○ سازگاری طراحی سایت با موضوع آن. (طراحی سایت موسیقی با سایت بانک متفاوت است.)            | هویت (Identity)   |
| ○ گشت و گذار باید ساده و سازگار باشد.                                                     | قابلیت گشت و گذار |
| ○ طراحی مبتنی بر حدس و قابل پیش بینی باشد.                                                | (Navigation)      |

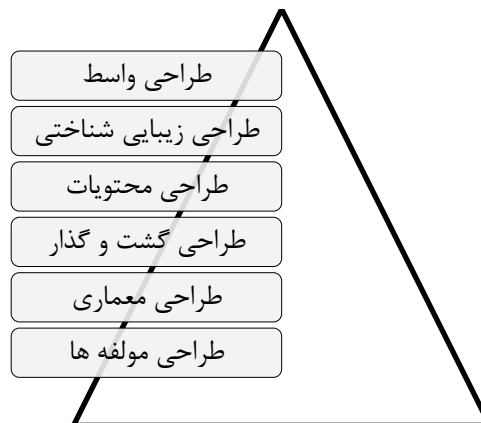
نکته: اگر سایتی کاملاً قابل استفاده باشد، ولی فاقد ظرافت و سبک طراحی مناسب باشد، شکست خواهد خورد.

## ۴. هرم طراحی برنامه های تحت وب

### ۴.۱. مهارت های مورد نیاز برنامه های تحت وب



### ۴.۲. هرم طراحی برنامه های تحت وب



## ۵. طراحی واسط برنامه تحت وب

### ۵.۱. اهداف واسط برنامه های تحت وب

- ساخت پنجره ای سازگار فراروی محتوای سایت
- راهنمایی کاربر
- سازماندهی گزینه های گشت و گذار
- منوهای عمودی یا افقی به صورت سلسله مراتبی
- آیکون های گرافیکی شامل دکمه ها، سویچ ها و تصاویر گرافیکی

|                                                                                      |              |                                   |              |                              |                 |
|--------------------------------------------------------------------------------------|--------------|-----------------------------------|--------------|------------------------------|-----------------|
| فراموشی کلمه عبور؟<br>ثبت نام<br>* روشهای پرداخت                                     |              | پرتال جدید نو نگار راه اندازی شد. |              |                              |                 |
| صفحه اصلی                                                                            | درباره ما    | خدمات                             | محمولات      | نمونه کارها                  | تعرفه ها        |
| تبلیغ در Google                                                                      | مقالات مرتبط | خدمات وب                          | جستجوی دامین | گواهینامه ها                 | نرم افزار Lemon |
| شرکت نو نگار پردازش خدمات طراحی سا<br>خود را، با توجه به نیازهای کارفرمایان، نوع فعا |              | خدمات طراحی صفحات وب<br>نرم افزار |              | از کارفرمایان ما :<br>پاکنام |                 |



## ۶. طراحی زیبایی شناسانه (گرافیکی)

### ۶.۱. مسائل مربوط به چیدمان صفحات وب

- از فضای خالی نترسید. (آشفته‌گی بصری، چشم نواز نیست)
- بر محتوا تاکید کنید. (هر صفحه وب ۸۰٪ مطلب و ۲۰٪ سایر امکانات)
- عناصر را از راست به چپ و از بالا به پایین سازماندهی کنید.
- گشت و گذار، محتوا و قابلیت‌های سایت را بطور مکانی گروه‌بندی کنید.
- محتوای خود را با نوارهای جابجایی (Scroll Bar) توسعه ندهید.
- کاربران جابجایی عمودی صفحه را بیش از جابجایی افقی تحمل می‌کنند.

### ۶.۲. مسائل طراحی گرافیک وب

- آدم‌هایی را می‌بینیم که یک سایت را به سرعت و تنها از روی طراحی بصری آن قضاوت می‌کنند.
- هر طراح وبی، استعداد هنرمندانه ندارد. برای کار طراحی زیبایی‌شناسی یک طراح گرافیک کارآموده را بکار بگیرید.

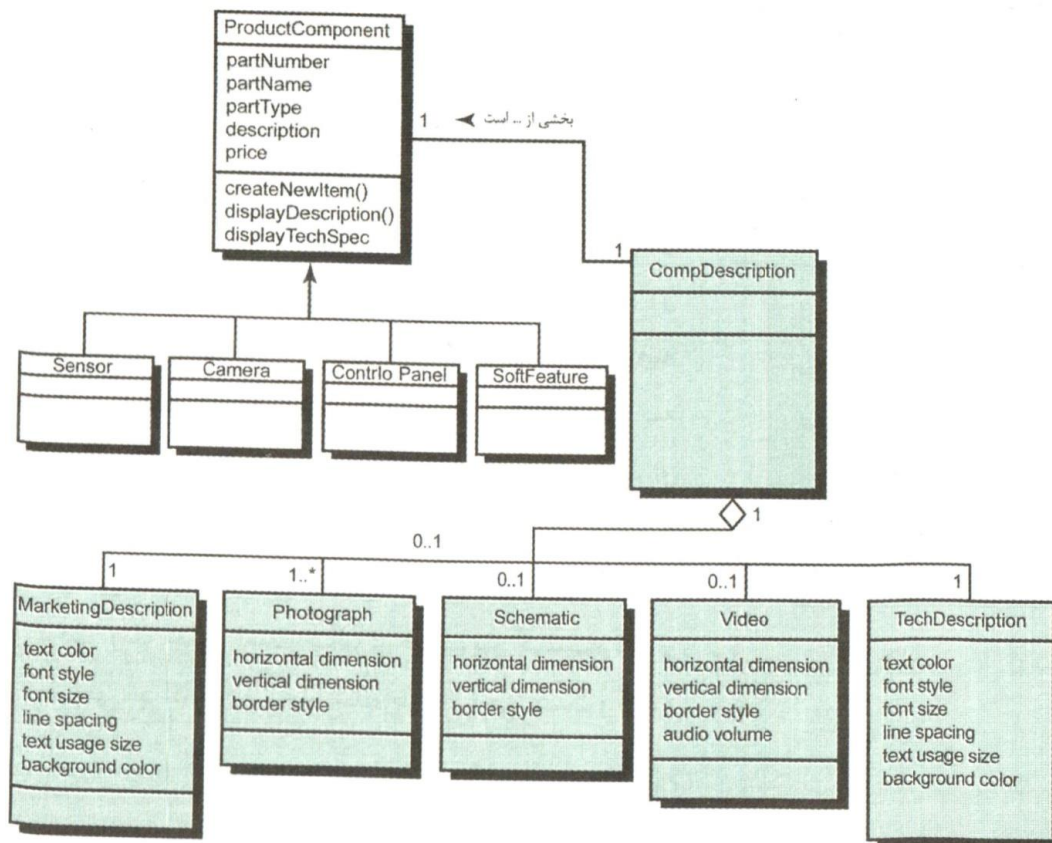
|        |     |           |      |     |       |           |
|--------|-----|-----------|------|-----|-------|-----------|
| چیدمان | رنگ | اندازه‌ها | فونت | صدا | تصویر | پویانمایی |
|--------|-----|-----------|------|-----|-------|-----------|

## ۷. طراحی محتوا

### ۷.۱. وظایف طراحی محتوا

- یک نمایش طراحی اشیای محتوایی (Content Object) و سازوکارهای لازم برای ایجاد رابطه میان آن‌ها توسعه می‌یابد.
- اطلاعات درون هر شی محتوایی خاص ایجاد می‌شود.





۳-۱۳ نمایش طراحی اشیای محتوایی.

## ۷.۲. مسائل طراحی محتوا

- مدل سازی اشیای محتوایی
- مدیریت اطلاعات هر شی محتوایی
- فرمت بندی اطلاعات اشیای محتوایی برای همخوانی با نیازهای مشتری
- دسته بندی اشیای محتوایی برای تشکیل صفحات وب
- طراحان خوب می توانند از آشفته گی، نظم بیافرینند؛ آن ها می توانند به وضوح ایده ها را از طریق سازمان دهی و دستکاری واژه ها و تصاویر منتقل کنند.

## ۸. طراحی معماری

### ۸.۱. طراحی معماری با چه چیزهایی ارتباط دارد؟

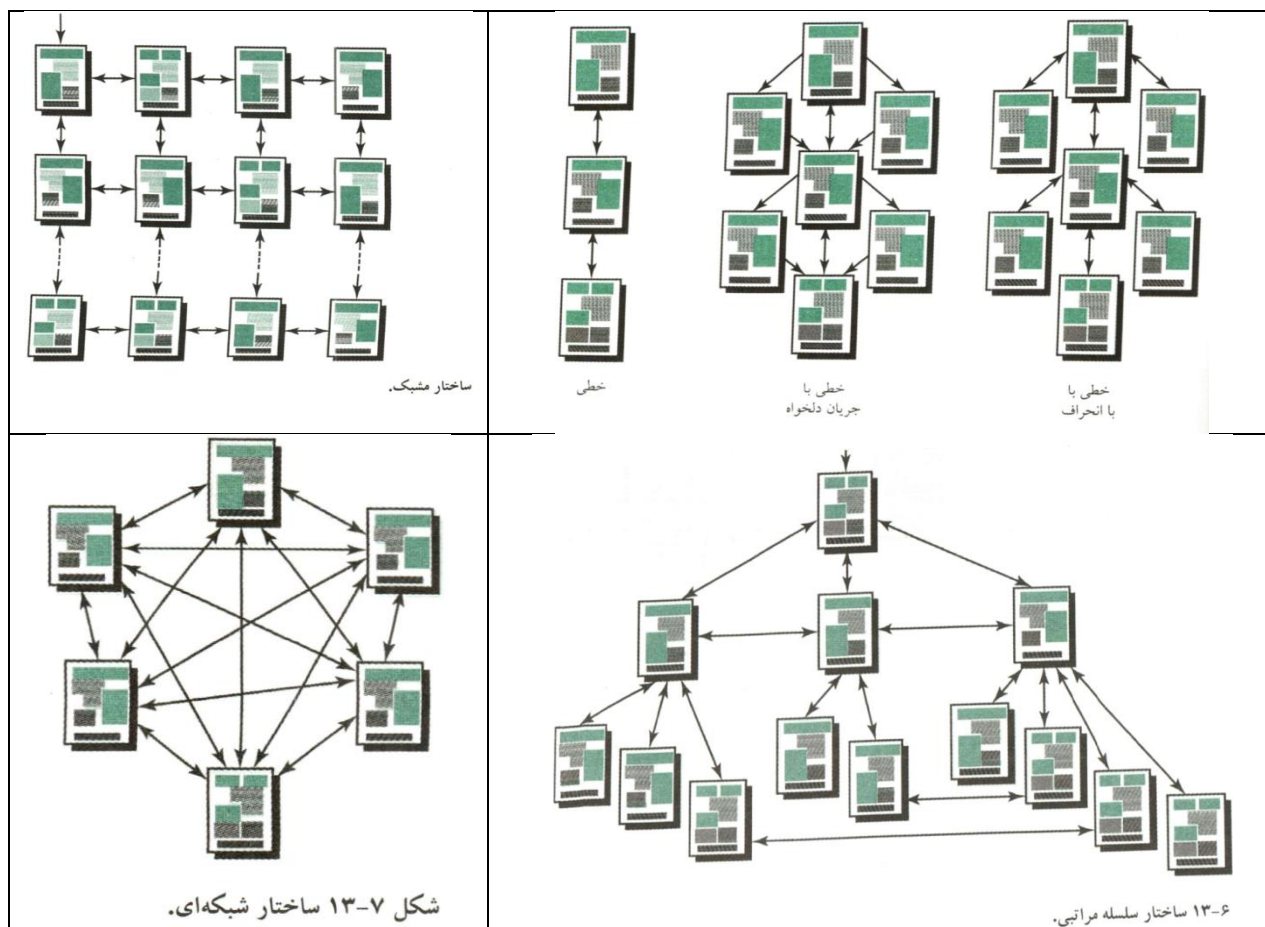
- اهداف تعیین شده برای برنامه تحت وب
- محتوایی که قرار است ارائه شود.
- کاربران بازدیدکننده از برنامه تحت وب
- فلسفه تعیین شده برای گشت و گذار

## ۸.۲. انواع معماری

| انواع معماری                        | توضیح                                                                                                                                                                      |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| معماری محتوا (Content Architecture) | <ul style="list-style-type: none"> <li>○ شیوه ساختاردهی اشیای محتوایی یا صفحات وب برای عرضه و گشت و گذار</li> <li>○ تعریف ساختار ابر رسانه ای کلی برنامه تحت وب</li> </ul> |
| معماری برنامه تحت وب                | <ul style="list-style-type: none"> <li>○ شیوه ساختاردهی به برنامه تحت وب برای مدیریت تعامل با کاربر، اداره وظایف پردازشی، گشت و گذار موثر و ارایه محتوا</li> </ul>         |

## ۸.۳. چهار ساختار معماری محتوا

| ساختارهای معماری                      | توضیح                                                                                                                |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| خطی (Linear Structure)                | ○ نمایش خودآموز، سفارش محصولات، ارسال پیامک اینترنتی                                                                 |
| مشبک (Grid Structure)                 | ○ سایت تجارت الکترونیک فروش تلویزیون دارای دو بعد.<br>○ بعد افقی: نوع تلویزیون - بعد عمودی: شرکت های سازنده تلویزیون |
| سلسله مراتبی (Hierarchical Structure) | ○ ساختار درختی                                                                                                       |
| شبکه ای (Networked Structure)         | ○ ساختار پیچیده                                                                                                      |
| مرکب                                  | ○ تلفیق چهار نوع ساختار معماری گفته شده                                                                              |

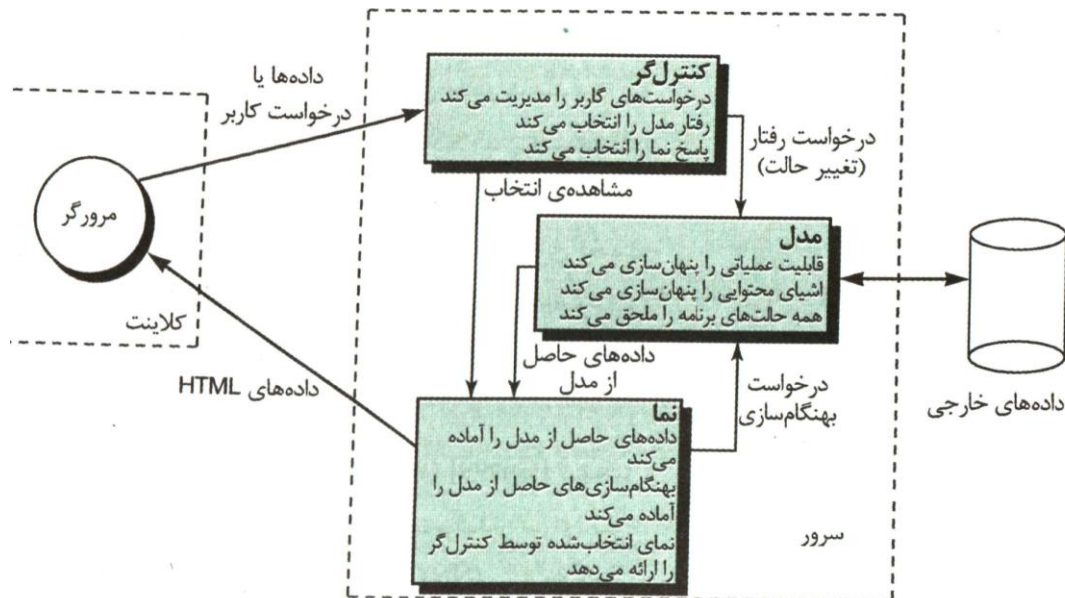


## ۸.۴. معماری برنامه های تحت وب (معماری سه لایه MVC)

- برنامه های کاربردی را باید با استفاده از لایه هایی ساخت که در آن ها دغدغه های متفاوت به حساب آورده می شوند.

- معماری سه لایه (MVC): واسط را از گشت و گذار و از رفتار و قابلیت های عملیاتی برنامه کاربردی منفک می کنند.

| معماری سه لایه        | توضیح                                                                                            |
|-----------------------|--------------------------------------------------------------------------------------------------|
| مدل (Model)           | شامل همه محتوای خاص برنامه و منطق پردازش، اشیای محتوایی، منابع داده ها و قابلیت های عملیاتی است. |
| نما (View)            | توسط کنترل گر و با داده های به دست آمده از مدل، بر اساس ورودی کاربر، به هنگام می شود.            |
| کنترل گر (Controller) | دستیابی به مدل و نما را مدیریت می کند و جریان داده ها را میان آن ها هماهنگ می سازد.              |



شکل ۸-۱۳ معماری MVC.

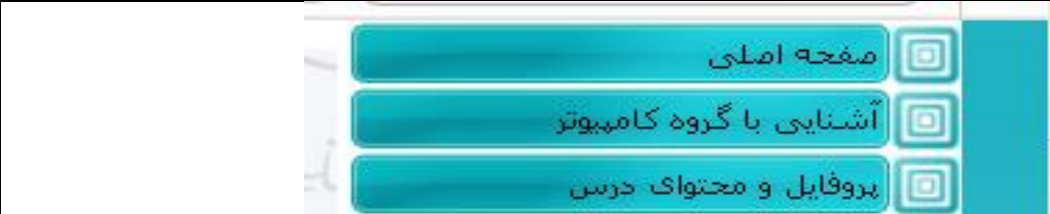
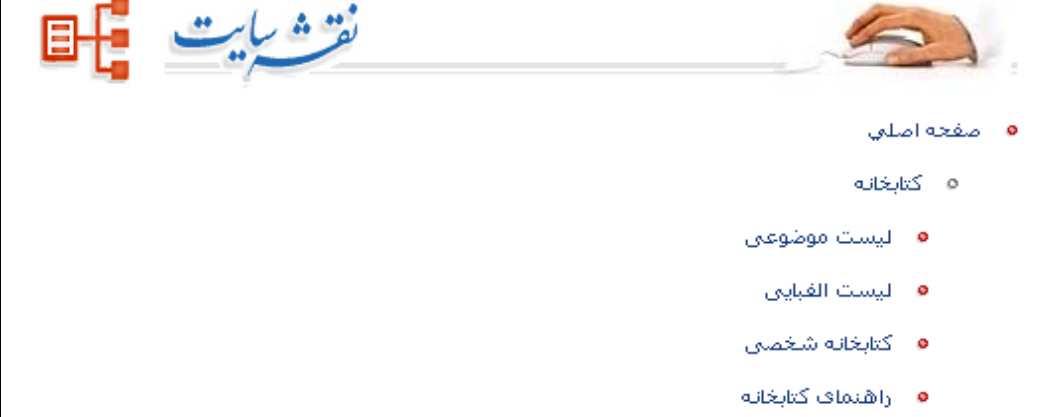
## ۹. طراحی گشت و گذار

### ۹.۱. معناسازی گشت و گذار (Navigation Semantics)

کارکرد گشت و گذار وابسته به کاربر است.

| واحد | توضیح                                                                                                                                                                                                                                             |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NSU  | <ul style="list-style-type: none"> <li>واحد های معناسازی گشت و گذار</li> <li>مجموعه ای از اطلاعات و ساختارهای گشت و گذار مرتبط که با همکاری یکدیگر، زیرمجموعه ای از خواسته های مرتبط با کاربر را برآورده می سازند.</li> </ul>                     |
| WON  | <ul style="list-style-type: none"> <li>هر NSU از مجموعه ای عناصر گشت و گذار موسوم به راه های گشت و گذار (WON) تشکیل می شود.</li> <li>یک WON، نشان گر بهترین مسیر گشت و گذار برای دستیابی به هدف گشت و گذار برای نوع خاصی از کاربر است.</li> </ul> |

## ۹.۲. قالب نحوی گشت و گذار (Navigation Syntax)

| توضیح                                                                                | قالب                        |
|--------------------------------------------------------------------------------------|-----------------------------|
|    | پیوند گشت و گذار<br>انفرادی |
|    | منوی افقی                   |
|   | برگه ها                     |
|  | منوی عمودی                  |
|  | نقشه سایت                   |

## جلسه ۳ – مفاهیم کیفیتی

### ۱. مساله چیست؟

- تا دهه ۱۹۹۰ شرکت های عظیم به این نتیجه رسیدند که سالانه میلیاردها دلار بیهوده صرف نرم افزارهایی می شود که ویژگی ها و قابلیت های وعده داده شده را تحویل نمی دهند.
- شرکت های تجاری در آمریکا میلیاردها دلار صرف نرم افزارهایی می کنند که آن چه را که قرار است انجام دهند، انجام نمی دهند.
- نرم افزارهای بد تقریباً هر سازمانی را که از کامپیوتر استفاده می کند به آشوب می کشد و باعث می شوند در مدت زمانی که کامپیوترها از کار می افتند، ساعت ها وقت کارمندان هدر رود، داده ها از بین رود یا مخدوش شود، فرصت های فروش از بین برود و از رضایت مشتری کاسته شود.

### ۲. کیفیت چیست؟

#### ۲.۱. توصیف کیفیت از پنج دیدگاه متفاوت

| دیدگاه    | توضیح                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| متعالی    | کیفیت چیزی است که بلافاصله آن را تشخیص می دهید ولی نمی توانید به صراحت آن را تعریف کنید.                                 |
| کاربری    | کیفیت بر حسب اهداف خاص کاربر نهایی دیده می شود. اگر محصولی این اهداف را برآورده سازد، از خود کیفیت نشان می دهد.          |
| سازندگان  | کیفیت بر حسب مشخصات اولیه محصول تعریف می شود. اگر محصول با مشخصات تعیین شده مطابقت داشته باشد، از خود کیفیت نشان می دهد. |
| محصولی    | کیفیت را می توان به خصوصیات ذاتی از قبیل قابلیت ها و ویژگی های محصول ربط داد                                             |
| ارزش محور | کیفیت بر اساس میزان پولی سنجیده می شود که مشتری حاضر به پرداخت برای محصول است.                                           |

- تحویل در زمان بندی و بودجه تعیین شده + کیفیت خوب + محصول مطابق با استاندارد = رضایت کاربر

### ۳. کیفیت نرم افزار

#### ۳.۱. کیفیت نرم افزار چیست؟

- یک فرایند نرم افزاری موثر که به شیوه ای به کاربرده می شود. که محصولی مفید ایجاد می کند. تا ارزشی قابل سنجش برای سازندگان این محصول و استفاده کنندگان از آن ایجاد کند.
- فرایند نرم افزار اثربخش زیرساختی را بنا می کند که هرگونه تلاش برای ساخت یک محصول نرم افزاری با کیفیت بالا را پشتیبانی می کند.
- محصول مفید، محتویات، قابلیت ها و ویژگی هایی را تحویل می دهد که مطلوب کاربر نهایی هستند، ولی تحویل این موارد به شیوه ای مطمئن و عاری از خطا نیز به همان اندازه دارای اهمیت است.

- یک نرم افزار با کیفیت بالا با افزودن ارزش برای تولیدکننده و کاربر این محصول نرم افزاری هم برای سازمان نرم افزاری و هم برای جامعه کاربران نهایی مزیت فراهم می کند.
- سازمان نرم افزاری از آن رو ارزش افزوده کسب می کند که، نرم افزار با کیفیت بالا به تلاش کمتر برای نگهداری، اشکال زدایی کمتر و و پشتیبانی کمتر برای مشتری نیاز دارد.
- این به مهندسان نرم افزار امکان می دهد تا وقت بیشتری را صرف ایجاد برنامه های کاربردی جدید کنند و کمتر به دوباره کاری بپردازند.
- جامعه کاربران از آن رو ارزش افزوده کسب می کنند که برنامه یک قابلیت مفید فراهم می سازد. به طوری که یک فرایند تجاری خاص با سرعت بیشتری انجام شود.

### ۳.۲. نتیجه نهایی

- درآمد بیشتر برای محصول نرم افزاری،
- منفعت بهتر هنگام پشتیبانی یک برنامه کاربردی از یک فرایند تجاری
- و یا بهبود دسترسی به اطلاعات حیاتی برای شرکت تجاری خواهد بود.

## ۴. ابعاد کیفیتی

### ۴.۱. ابعاد کیفیتی گاروین

| ابعاد            | توضیح                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| کیفیت کارایی     | آیا نرم افزار همه محتویات، قابلیت ها و ویژگی های مشخص شده در مدل خواسته ها را طوری تحویل می دهد که برای کاربر نهایی ایجاد ارزش کند؟                     |
| کیفیت ویژگی ها   | آیا نرم افزار ویژگی هایی فراهم می آورد که کاربران نهایی را در نخستین استفاده از نرم افزار شگفت زده و خشنود کند؟                                         |
| قابلیت اطمینان   | آیا نرم افزار، همه ویژگی ها و قابلیت ها را بدون شکست تحویل می دهد؟ آیا هنگام نیاز در دسترس هست؟ آیا قابلیت های عملیاتی را عاری از خطا تحویل می دهد؟     |
| متابعت           | آیا نرم افزار از استانداردهای محلی و خارجی که به کاربرد مورد نظر مربوط می شوند، پیروی می کند؟ آیا با قراردادهای طراحی و کد نویسی غیر رسمی همخوانی دارد؟ |
| دوام             | آیا نرم افزار را می توان نگهداری کرد (تغییر داد) یا تصحیح (اشکال زدایی) کرد بدون این که اثرات جانبی ایجاد شود؟                                          |
| قابلیت سرویس دهی | آیا نرم افزار را می توان در زمانی کوتاه نگهداری کرد (تغییر داد) یا تصحیح (اشکال زدایی) کرد؟                                                             |

### ۴.۲. عوامل کیفیتی مک کال

| شاخص های کیفیتی | توضیح                                                                               |
|-----------------|-------------------------------------------------------------------------------------|
| قابلیت اطمینان  | حدی که می توان از برنامه انتظار داشت تا عملکردهای موردنظر را با دقت لازم ارائه دهد. |
| بازدهی          | مقدار منابع کامپیوتری و کد لازم برای آن که برنامه قادر به اجرای عملکردهای خود باشد. |
| انسجام          | حد کنترل دستیابی افراد غیر مجاز به نرم افزار یا داده ها                             |
| قابلیت استفاده  | کار لازم برای فراگیری راه اندازی آماده کردن ورودی و تفسیر خروجی برنامه              |
| قابلیت نگهداری  | کار لازم برای یافتن و تصحیح خطاهای برنامه                                           |



|                     |                                                                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| آزمون پذیری         | کار لازم برای آزمودن برنامه برای اطمینان یافتن از این که عملکرد مورد نظر رابه خوبی اجرا می کند.                                                    |
| حمل پذیری           | کار لازم برای انتقال دادن نرم افزار از یک سخت افزار و یا محیط سیستم نرم افزاری به دیگری                                                            |
| قابلیت استفاده مجدد | حدی که می توان یک برنامه (یا بخش هایی از برنامه) را دوباره در کاربردهای دیگر مرتبط با پکیج سازی و دامنه عملیاتی که برنامه اجرا می کند استفاده کرد. |
| قابلیت کار متقابل   | کار لازم برای جفت کردن یک سیستم به سیستم دیگر.                                                                                                     |

## ۴.۳. عوامل کیفیتی ایزو ۹۱۲۶

| شاخص های کیفیتی | توضیح                                                                          |
|-----------------|--------------------------------------------------------------------------------|
| قابلیت عملیاتی  | حدی که نرم افزار نیازهای ذکر شده براساس این صفات را برآورده می کند.            |
| قابلیت استفاده  | حد سهولت استفاده از نرم افزار بر اساس این صفات                                 |
| قابلیت درک      | قابلیت فراگیری کار با آن                                                       |
| بازدهی          | حدی که نرم افزار بر اساس این صفات از منابع سیستم استفاده بهینه به عمل می آورد. |
| قابلیت نگهداری  | سهولت ترمیم نرم افزار                                                          |
| حمل پذیری       | سهولت انتقال نرم افزار از محیطی به محیط دیگر                                   |

## ۴.۴. عوامل کیفیتی هدفمند

| شاخص های کیفیتی | توضیح                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------|
| بصیرت گرایی     | میزان پیروی واسط از الگوهای کاربرد مورد انتظار به طوری که حتی یک کاربر تازه کار بتواند بدون نیاز به آموزش زیاد از آن استفاده کند. |
| بازدهی          | میزانی از امکان یافتن عملیات ها و اطلاعات یا استفاده از آن ها                                                                     |
| استحکام         | میزان اداره کردن داده های ورودی یا تعامل نامناسب کاربر توسط نرم افزار                                                             |
| غنا             | میزان ارائه مجموعه ای از ویژگی ها به وسیله واسط                                                                                   |

## ۵. معضل کیفیت نرم افزار

## ۵.۱. هزینه کیفیت

- هزینه کیفیت شامل همه هزینه هایی می شود که در جستجوی کیفیت، یا اجرای فعالیت های مرتبط با کیفیت و هزینه های ناشی از فقدان کیفیت تحصیل می شوند.
- برای شناخت این هزینه ها سازمان باید معیارهایی جمع آوری کند که: بستری برای هزینه جاری کیفیت شناسایی فرصت ها برای کاهش دادن این هزینه ها و فراهم ساختن مبنایی بهنجار جهت مقایسه به دست دهد.

هزینه کیفیت را می توان به هزینه های مرتبط با موارد زیر تقسیم کرد:

| هزینه    | توضیح                                                                                                                                                                                                                                                      |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| پیش گیری | عبارتند از فعالیت های مدیریتی مورد نیاز برای برنامه ریزی و هماهنگ کردن کلیه فعالیت های تضمین کیفیت:<br>○ هزینه فعالیت های فنی برای توسعه و تکمیل مدل خواسته هاومدل طراحی<br>○ هزینه های برنامه ریزی آزمون<br>○ هزینه همه ی آموزش های مرتبط بااین فعالیت ها |
| ارزیابی  | شامل فعالیت های انجام شده برای به دست آوردن دیدی از وضعیت محصول در نخستین گذر از هر فرایند:                                                                                                                                                                |

|                                                                                                                                                                                                                                                                                                                                                                                                     |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <ul style="list-style-type: none"> <li>○ هزینه اجرای بازبینی های فنی برای محصولات کاری مهندسی نرم افزار</li> <li>○ هزینه جمع آوری داده ها و ارزیابی معیارها</li> <li>○ هزینه آزمون و اشکال زدایی</li> </ul>                                                                                                                                                                                         |            |
| <p>هنگامی تحصیل می شوند که در محصول و قبل از رسیدن آن به مشتری کشف می شوند:</p> <ul style="list-style-type: none"> <li>○ هزینه لازم برای اجرای دوباره کاری (ترمیم) برای تصحیح خطا</li> <li>○ هزینه ناشی از اثرات جانبی که در اثر دوباره کاری ها ایجاد می شود.</li> <li>○ هزینه های مربوط به جمع آوری معیارهای کیفیتی که به سازمان این امکان را می دهند تا به حالت های شکست دست پیدا کند.</li> </ul> | شکست داخلی |
| <p>به نقایصی مربوط می شود که پس از رسیدن محصول به دست مشتری کشف می شوند.</p> <p>همانطور که انتظار می رود هزینه های نسبی برای یافتن و ترمیم خطاها یا نقایص با رفتن از پیش گیری به سوی کشف خطا و سپس شکست داخلی و سرانجام شکست خارجی به شدت افزایش می یابد.</p>                                                                                                                                       | شکست خارجی |

## ۵.۲. ریسک

- منظور این است که نرم افزارهای باکیفیت پایین هم برای سازنده و هم برای کاربر نهایی ایجاد ریسک می کنند.
- کیفیت ضعیف به ریسک هایی منجر می شود که برخی از آن ها بسیار جدی اند.

## ۵.۳. کیفیت و امنیت

- نفوذ در نرم افزاری که کیفیت بالایی از خود نشان ندهد راحت تر است.
- و در نتیجه نرم افزارهای باکیفیت پایین می توانند به طور غیر مستقیم ریسک امنیتی را با تمام هزینه ها و مشکلات مربوط به آن افزایش دهند.

## ۵.۴. تاثیر کنش های مدیریتی

- کیفیت نرم افزار غالباً به همان اندازه که از تصمیم گیری های فناوری تاثیر می پذیرد از تصمیم گیری های مدیریتی نیز تاثیر پذیر است.
- حتی بهترین کارهای مهندسی نرم افزار ممکن است با تصمیم گیری های تجاری ضعیف و کنش های مدیریت پروژه ضعیف به بیراهه کشیده شود.
- تصمیم گیری برآوردی
- تصمیم گیری های زمان بندی
- تصمیم گیری های مربوط به ریسک

## ۶. دستیابی به کیفیت نرم افزار

### ۶.۱. روش های مهندسی نرم افزار

- کیفیت نرم افزار چیزی نیست که یک باره ظاهر شود.
- نتیجه مدیریت خوب پروژه و کار مهندسی نرم افزار مستحکم است.



- مدیریت و کار در حیطه چهار فعالیت گسترده است که تیم نرم افزاری را در دستیابی به کیفیت بالای نرم افزاری یاری می دهد.

## ۶,۲. تکنیک های مدیریت پروژه

- مدیر پروژه از برآوردها استفاده کند تا ببیند آیا تاریخ های تحویل قابل تحقق هستند.
- وابستگی های زمان بندی درک شده و تیم در برابر وسوسه استفاده از میان برها مقاومت کند.
- برنامه ریزی برای ریسک انجام شده باشد به طوری که مسائل تولید آشوب نکنند کیفیت نرم افزار را می توان به نحوی مثبت تحت تاثیر قرار داد.

## ۶,۳. کنترل کیفیت

- کنترل کیفیت شامل مجموعه ای از کنش های مدیریت نرم افزاری شود که به کمک آن ها می توان اطمینان حاصل کرد که هر محصول کاری اهداف کیفیتی اش را برآورده ساخته است.

## ۶,۴. تضمین کیفیت

- تضمین کیفیت زیر ساختی را تعیین می کند که روش های مهندسی نرم افزار، مدیریت پروژه موجه و کنش های کنترل کیفیت را که همگی در ساخت نرم افزارهای باکیفیت بالا اهمیت محوری دارند پشتیبانی می کند.
- به علاوه تضمین کیفیت شامل یک مجموعه وظایف ممیزی و گزارش دهی می شود که اثربخش بودن و کامل بودن کنش های کنترل را ارزیابی می کنند.

## جلسه ۴- تکنیک های مرور نرم افزار

### ۱. مقدمه

#### ۱.۱. مرور چیست؟

- شما هنگام توسعه محصولات کاری مهندسی نرم افزار، مرتکب اشتباه می شوید.
- مرورهای فنی، اثربخش ترین سازوکار برای یافتن زود هنگام خطاها در فرایند نرم افزار به شمار می روند.
- مرورهای نرم افزار به مثابه فیلترهایی برای فرایند مهندسی نرم افزار عمل می کنند. یعنی در نقاط گوناگونی از توسعه نرم افزار اعمال می شوند و به کشف خطاها و نقایصی که قابل رفع باشند کمک می کنند.

#### ۱.۲. چرا اهمیت دارد؟

- اگر خطای موجود در فرایند را زود هنگام بیابید، تصحیح آن، هزینه کمتری در بر خواهد داشت.
- مرورها با کاستن از دوباره کاری ها، باعث صرفه جویی در زمان خواهد شد.

#### ۱.۳. مراحل کار کدام است؟

|             |            |                    |           |               |                           |
|-------------|------------|--------------------|-----------|---------------|---------------------------|
| برنامه ریزی | آماده سازی | سازماندهی به جلسات | ذکر خطاها | انجام تصحیحات | وارسی درستی انجام تصحیحات |
|-------------|------------|--------------------|-----------|---------------|---------------------------|

### ۲. تأثیر نقایص نرم افزار بر هزینه ها

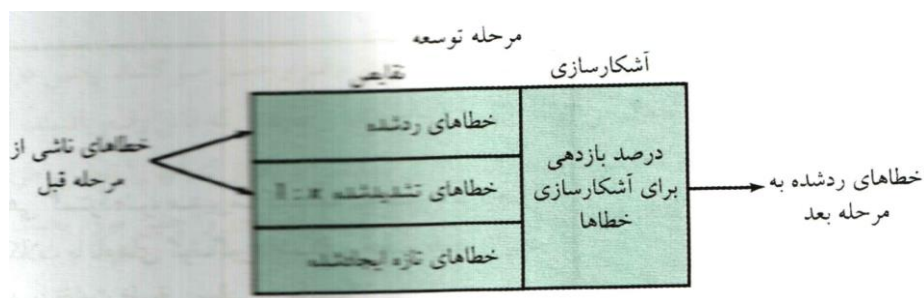
- نقص: مشکلی که پس از ارائه نرم افزار به کاربر نهایی کشف می شود.
- هدف اصلی مرورهای فنی رسمی یافتن خطاها در اثنای فرایند است به طوری که پس از ارائه نرم افزار به نقص تبدیل نشوند.
- مزیت مرورهای فنی رسمی، کشف زود هنگام خطاهاست.
- تکنیک های مرور رسمی تا ۷۵ درصد در کشف معایب طراحی مؤثر واقع می شوند.

### ۳. تشدید نقایص و حذف آن ها

#### ۳.۱. تشدید نقایص و حذف آن ها

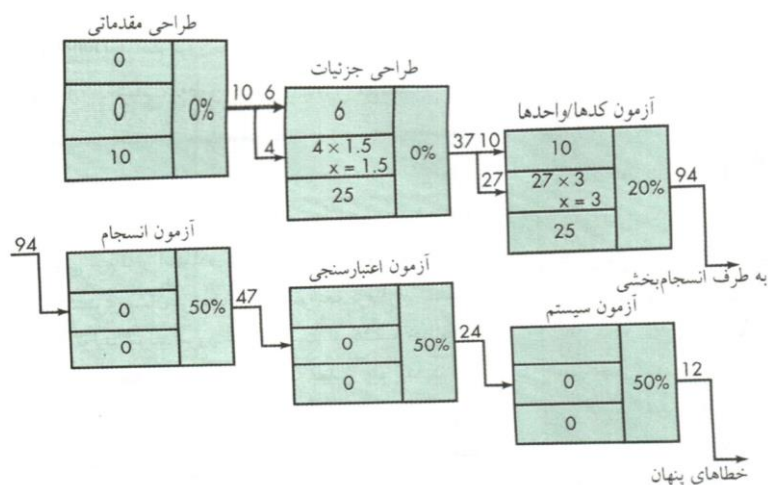
از مدل تشدید نقص می توان برای نمایش تولید و یافتن خطاها طی طراحی مقدماتی، طراحی مشروع و مراحل کدنویسی در فرایند مهندسی نرم افزار استفاده کرد.

#### ۳.۲. مدل تشدید نقایص



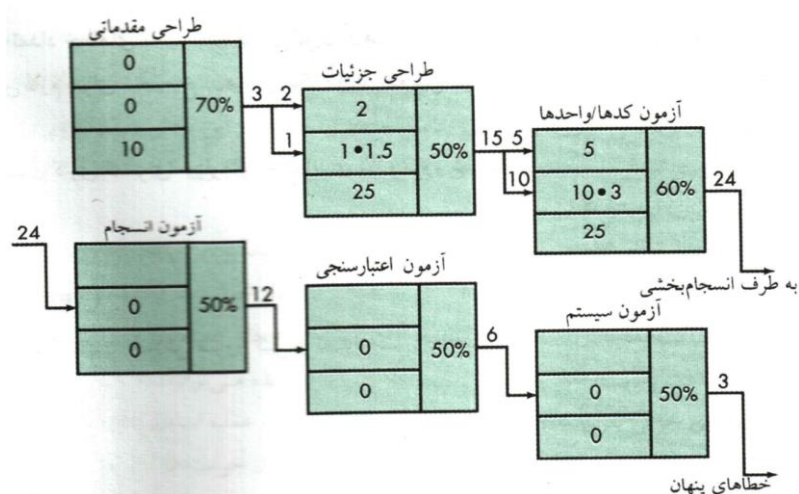
### ۳,۳. تشدید نقایص - بدون مرور

در شکل زیر یک مثال فرضی از تشدید نقص برای فرایند توسعه نرم افزار نشان داده شده است که در آن هیچ مروری صورت نمی پذیرد.



### ۳,۴. تشدید نقایص - مرورهای انجام شده

در شکل زیر شرایط شکل قبل فرض می شود با این تفاوت که مرورهای طراحی و کدها به عنوان بخشی از هر مرحله توسعه انجام می شود.



### ۳,۵. نتیجه گیری

در مثال های قبل هزینه کل توسعه و نگهداری در هنگام اجرای مرورها، ۷۸۳ واحد می شود. هنگامی که هیچ مروری صورت نپذیرد، هزینه کل ۲۱۷۷ واحد می شود که تقریباً سه برابر هزینه برمی دارد.

## ۴. معیارهای مرور و کاربردهای آنها

### ۴.۱. معیارهای مرور و کاربردهای آنها

معیارهای مرور زیر را می توان برای هر کدام از مرورهای اجرا شده جمع آوری کرد.

| معیارها                                       | توضیح                                                                                                                                      |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| تلاش آماده سازی (Ep)                          | تلاش و کار لازم برحسب نفر ساعت برای مرور یک محصول کاری قبل از جلسه مرور واقعی                                                              |
| تلاش ارزیابی (Ea)                             | تلاش صرف شده بر حسب نفر ساعت طی مرور واقعی                                                                                                 |
| تلاش دوباره کاری (Er)                         | تلاش اختصاص داده شده برحسب نفر ساعت به تصحیح آن دسته از خطاهایی که طی مرور کشف می شوند.                                                    |
| اندازه محصول کاری (WPS)                       | میزانی از اندازه محصول کاری که مرور شده است. مثلاً تعداد خطوط کد                                                                           |
| خطاهای جزئی یافته شده (Err <sub>minor</sub> ) | تعداد خطاهای یافته شده که می توان آنها را در زمره خطاهای جزئی تقسیم بندی کرد (تلاش لازم برای تصحیح آنها از یک مقدار تعیین شده کوچکتر است). |
| خطاهای عمده یافته شده (Err <sub>major</sub> ) | تعداد خطاهای یافته شده که می توان آنها را در زمره خطاهای عمده دسته بندی کرد. (تلاش لازم برای تصحیح آنها از یک مقدار تعیین شده بزرگتر است). |

### ۴.۲. تحلیل معیارها

- پیش از شروع تحلیل چند محاسبه ساده نیاز است. تلاش مرور کد و تعداد کل خطاهای کشف شده به صورت زیر تعریف می شوند.

$$E_{\text{review}} = E_p + E_a + E_r$$

$$\text{Err}_{\text{tot}} = \text{Err}_{\text{minor}} + \text{Err}_{\text{major}}$$

- چگالی خطا نشانگر خطاهای یافته شده به ازای واحد محصول کاری مرور شده است.

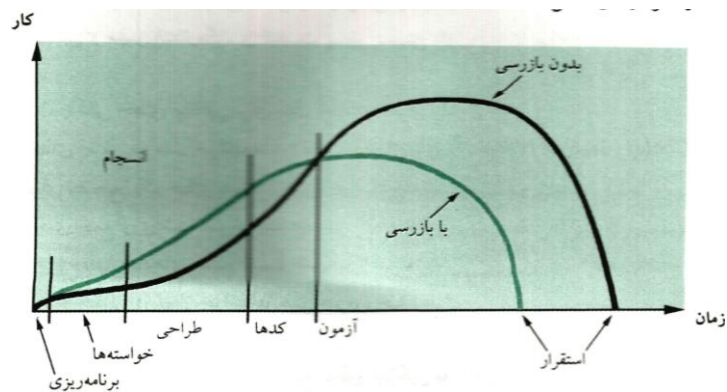
$$\text{چگالی خطا} = \frac{\text{Err}_{\text{tot}}}{\text{WPS}}$$

### ۴.۳. اثربخشی هزینه مرورها

اثربخشی مرورها و عایدی حاصل از صرف هزینه در این بخش را تنها پس از پایان مرورها، جمع آوری معیارها، محاسبه داده های میانگین و سپس اندازه گیری کیفیت پایین دستی از طریق انجام آزمون ارزیابی کرد.

$$\text{تلاش صرفه جویی شده به ازای هر خطا} = E_{\text{testing}} - E_{\text{reviews}}$$

### ۴.۴. تلاش های صرف شده با مرور و بدون مرور



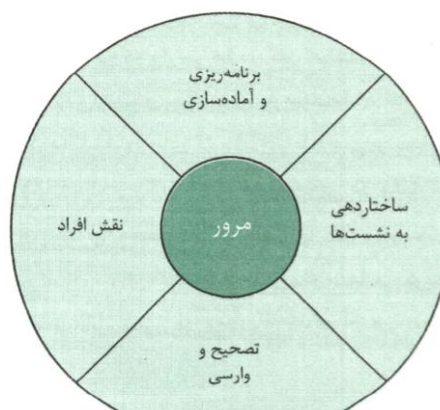
## ۵. مرورها: یک طیف رسمیت

### ۵.۱. مرورها: یک طیف رسمیت

مرورهای فنی را باید با سطحی از رسمیت به کار برد که با محصولی که قرار است ساخته شود، خط زمانی پروژه و کسانی که کار را انجام می دهند تناسب داشته باشند.

### ۵.۲. مدل مرجع برای مرورهای فنی

مدل مرجع زیر چهار خصوصیت سهیم در تعیین سطح رسمیت اجرای مرور را مشخص می کند.



## ۶. مرورهای غیر رسمی

### ۶.۱. مرورهای غیر رسمی

- مرورهای غیر رسمی شامل بررسی ساده یک محصول کاری مهندسی نرم افزار می شوند.
- بررسی ساده یک مرور است ولی از آنجا که هیچ برنامه ریزی یا آماده سازی قبلی وجود ندارد دستور کار یا ساختاری برای نشست تنظیم نمی شود و برای خطاهای کشف شده هیچ پیگیری در کار نیست.
- اثربخشی این گونه مرورها به طور چشمگیری کوچکتر از رویکردهای رسمی است. ولی این روش می تواند به کشف خطاهایی منجر شود.

- یک راه برای بهبود بخشیدن به بازدهی مرورهای ساده تهیه مجموعه ای از چک لیست های مرور ساده برای هر محصول کاری است که توسط تیم نرم افزاری ایجاد می شود.

## ۷. مرورهای فنی رسمی

### ۷.۱. اهداف مرورهای فنی رسمی

- کشف خطاها در عملکرد، منطق یا پیاده سازی هر نمایی از نرم افزار
- تصدیق اینکه نرم افزار مورد مرور، خواسته های خود را برآورده می سازد.
- حصول اطمینان از اینکه نرم افزار طبق استانداردهای از پیش تعیین شده ارائه شده است.
- رسیدن به نرم افزاری که به شیوه ای یکنواخت توسعه یافته است.
- قابل اداره کردن پروژه ها

### ۷.۲. نشست مرور

- هر قالب مرور فنی رسمی که انتخاب شود در کلیه نشست های مرور باید شرایط زیر را رعایت کرد.
- بین سه تا پنج نفر باید در نشست حضور یابند.
- آمادگی قبلی لازم است ولی نباید بیش از دو ساعت از وقت هر نفر را بگیرد.
- مدت زمان جلسه باید کمتر از ۲ ساعت باشد.

### ۷.۳. گزارش مرور و ثبت امور

- در اثنای مرور فنی رسمی مسئول ثبت، فعالانه همه مسائلی را که مطرح شده است ثبت می کند. گزارش خلاصه مرور سه پرسش زیر را پاسخ می دهد.
- چه چیزی مرور شده است؟ - چه کسانی آن را مرور کرده اند؟ - یافته ها و نتایج کدامند؟

### ۷.۴. دستورالعمل های مرور

- مرور محصول نه تولید کننده محصول
- تهیه یک دستور کار و رعایت آن
- محدود کردن بحث و مشاجره
- بیان واضح بخش های مشکل دار و نه کوشش برای حل همه مشکلات ذکر شده
- یادداشت برداری
- محدود کردن تعداد شرکت کنندگان و اصرار بر آمادگی قبلی
- تهیه چک لیستی برای هر محصولی که احتمال مرور آن می رود.
- تخصیص منابع و زمانبندی برای مرور
- اجرای آموزش معنادار برای همه مسئولان مرور

## جلسه ۵- تضمین کیفیت نرم افزار

### ۱. مقدمه

#### ۱.۱. مساله چیست؟

- فقط گفتن این جمله که کیفیت نرم افزار اهمیت دارد کافی نیست، بلکه دقیقاً باید مشخص کنیم منظورمان از کیفیت نرم افزار چیست؟
- باید مجموعه ای از فعالیت ها را ایجاد کنید که به کمک آنها بتوان اطمینان حاصل کرد هر کار مهندسی نرم افزار کیفیت بالایی از خود نشان می دهد.
- اهمیت آن در این است که منجر به کاهش هزینه ها و تسریع در زمان تحویل به بازار می شود.

#### ۱.۲. تضمین کیفیت نرم افزار چیست؟

تضمین کیفیت نرم افزار (SQA) شامل موارد:

- یک فرایند SQA
- وظایف خاص تضمین کیفیت شامل مرورهای فنی و آزمون
- کار مهندسی نرم افزار اثر بخش
- کنترل محصولات کاری نرم افزاری
- رویه ای برای حصول اطمینان از مطابقت با استانداردهای نرم افزار
- سازوکارهای اندازه گیری و گزارش دهی

#### ۱.۳. مسائل پس زمینه

- تضمین کیفیت فعالیت های ضروری برای هر شرکتی است که کار آن تولید محصولاتی برای استفاده توسط دیگران است.
- کسانی که SQA را انجام می دهند باید از دیدگاه مشتری به کیفیت نرم افزار بنگرند.

### ۲. عناصر تضمین کیفیت نرم افزار

| عناصر                  | توضیح                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| استانداردها            | <ul style="list-style-type: none"> <li>○ سازمان های IEEE و ISO و سایر سازمان ها، آرایه وسیعی از استانداردهای مهندسی نرم افزار و مستندات وابسته به آن را جمع آوری کرده اند.</li> <li>○ استاندارد ها را ممکن است سازمان مهندسی نرم افزار، داوطلبانه بپذیرد یا مشتری یا یک طرف ذینفع دیگر الزامی کند.</li> <li>○ وظیفه SQA حصول اطمینان از رعایت استانداردهای پذیرفته شده و مطابقت تمامی محصولات کاری با این استانداردهاست.</li> </ul> |
| مرورهای فنی و ممیزی ها | <ul style="list-style-type: none"> <li>○ مرورهای فنی یک نوع فعالیت کنترل کیفیت هستند که توسط مهندسان نرم افزار اجرا می شوند. هدف از انجام آن ها کشف خطاهاست.</li> </ul>                                                                                                                                                                                                                                                             |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| ○ ممیزی ها نوعی از مروری هستند که توسط پرسنل SQA و به قصد حصول اطمینان از رعایت دستورالعمل های کیفیتی برای کار مهندسی نرم افزار دنبال می شوند.                                                                                                                                                                                                                                                                                                                                                     |                                |
| ○ آزمون نرم افزار یک هدف اصلی یعنی یافتن خطاها را دنبال می کند.<br>○ وظیفه SQA حصول اطمینان از طرح ریزی درست و مناسب آزمون ها و اجرای اثر بخش آن هاست به طوری که احتمال دستیابی به هدف اصلی آن (یافتن خطاها) به حداکثر برسد.                                                                                                                                                                                                                                                                       | آزمون برای یافتن خطاها         |
| ○ تنها راه بهبود بخشیدن، سنجیدن عملکرد است.<br>○ SQA داده های مربوط به خطاها و نقایص را جمع آوری و تحلیل می کند تا بهتر معلوم شود که خطاها چگونه وارد می شوند و کدام فعالیت های مهندسی نرم افزار برای حذف آن ها از همه مناسب تر است.                                                                                                                                                                                                                                                               | جمع آوری و تحلیل خطاها و نقایص |
| ○ تغییر، یکی از مخرب ترین جنبه های هر پروژه نرم افزار است و اگر خوب مدیریت نشود می تواند به سردرگمی منجر شود و سردرگمی همیشه به کیفیت ضعیف منجر می شود                                                                                                                                                                                                                                                                                                                                             | مدیریت تغییرات                 |
| ○ هر سازمان نرم افزاری مایل است کارهای مهندسی نرم افزار خود را بهبود بخشد.<br>○ یک عامل کلیدی سهیم در بهبود بخشی، آموزش مهندسان نرم افزار، مدیران آن ها و سایر طرف های ذی نفع است.<br>○ سازمان SQA در حمایت از برنامه های آموزشی نقش کلیدی دارد.                                                                                                                                                                                                                                                   | آموزش                          |
| ○ شامل پکیج های بسته بندی شده (مثل Microsoft Office)<br>○ قطعات نیمه آماده (ساختار اسکلتی پایه را فراهم می آورند و می توان آن را مطابق با میل و سلیقه خریدار تکمیل کرد).<br>○ نرم افزار های قراردادی (از روی مشخصات ارائه شده توسط سازمان مشتری، طراحی و ساخته می شوند).<br>○ وظیفه سازمان SQA این است که اطمینان حاصل کند با پیشنهاد اقدامات کیفیتی خاص که منبع خرید باید رعایت کند، نرم افزار با کیفیت بالا نتیجه می شود و الزامات کیفیتی خاصی را در هر قرارداد منعقد شده با منبع خرید بگنجانند. | مدیریت منابع خرید              |
| ○ هر سازمان نرم افزار باید خط مشی ها و سیاست هایی را نهادینه سازد که داده ها را در تمامی سطوح محافظت کنند.<br>○ با SQA می توان مطمئن شد که فرایند و فناوری مناسب در دستیابی به کیفیت نرم افزار به کار رفته است.                                                                                                                                                                                                                                                                                    | مدیریت امنیت                   |
| ○ از آن جا که نرم افزار ها تقریباً همیشه یک جزء محوری در سیستم های در خدمت انسان (مثل خودروها و هواپیماها) هستند، تاثیر نقایص پنهان می تواند مصیبت بار باشد.<br>○ SQA ممکن است مسوول ارزیابی تاثیر شکست نرم افزار و شروع مراحل لازم برای کاهش ریسک باشد.                                                                                                                                                                                                                                           | ایمنی                          |
| ○ سازمان SQA باید اطمینان حاصل کند فعالیت های مدیریت ریسک به طور مناسب اجرا می شوند و طرح های مربوط به احتمال بروز ریسک تدوین شده اند.                                                                                                                                                                                                                                                                                                                                                             | مدیریت ریسک                    |

### ۳. وظایف، اهداف و معیارهای SQA

#### ۳.۱. وظایف SQA

- تهیه یک طرح SQA برای پروژه
- شرکت در توسعه توصیف فرایند نرم افزاری پروژه



- بازبینی فعالیت های مهندسی نرم افزار برای واری مطابقت با فرایند نرم افزاری مشخص
- بازرسی محصولات کاری برای واری مطابقت با محصولات تعیین شده به عنوان بخشی از فرایند نرم افزار
- حصول اطمینان از مستند سازی انحرافات در کار نرم افزار و محصولات کاری و مقابله با آنها بر اساس رویه مستند سازی شده
- ثبت هر گونه عدم مطابقت و گزارش به مدیریت ارشد

### ۳.۲. اهداف، صفات و معیارها

| عناصر               | توضیح                                                                                                                                                                                                                                                                    |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| کیفیت خواسته ها     | ○ صحت، سازگاری مدل و کامل بودن تاثیری قوی بر کیفیت همه محصولات کاری بعدی می گذارد.                                                                                                                                                                                       |
| کیفیت طراحی         | ○ دو عنصر از مدل طراحی باید توسط تیم نرم افزاری ارزیابی شود که کیفیت داشته باشد و طراحی با خواسته ها مطابقت دارد یا نه؟!<br>○ کد منبع و محصولات کاری مرتبط با آن باید با استاندارد های محلی کد نویسی مطابقت داشته باشد و خصوصیتی داشته باشد که قابلیت نگهداری بالا ببرد. |
| کیفیت کدها          | ○ تیم نرم افزاری باید منابع محدود را طوری به کار بگیرد که احتمال دستیابی به نتیجه با کیفیت بالاتر زیاد شود.                                                                                                                                                              |
| اثربخشی کنترل کیفیت | ○ در اثر بخشی آزمون ها معیار ما تعداد خطاها و اهمیت آن ها و تلاش لازم برای تصحیح خطاها است.<br>○ SQA تخصیص منابع به مرورها و آزمون ها را چک میکند تا مشخص شود درست تخصیص داده شدند یا نه؟!<br>○                                                                          |

## ۴. تضمین کیفیت آماری نرم افزار

### ۴.۱. تضمین کیفیت آماری نرم افزار

- اطلاعات مربوط به نقایص نرم افزار جمع آوری و گروه بندی می شود.
- کوشش می شود رد هر نقص تا علت اصلی آن پیگیری شود.
- استفاده از اصل پارتو (ریشه ۸۲٪ نقایص، ۲۲٪ همه علل ممکن است) آن ۲۲٪ جدا می شود.
- هنگامی که چند علت حیاتی شناسایی شدند حرکت برای تصحیح مشکلاتی که باعث این نقایص شدند آغاز می شود.

### ۴.۲. شش سیگما برای مهندسی نرم افزار

- شش سیگما پرکاربردترین راهبرد برای تضمین کیفیت آماری در صنایع کنونی است.
- اصطلاح شش سیگما از شش برابر مقدار انحراف معیار به دست آمده است که معادل با ۳,۴ نمونه معیوب به ازای هر یک میلیون نمونه است و این حد اعلا استانداردهای کیفیتی است.

### ۴.۳. مراحل شش سیگما

- تعریف خواسته های مشتری، محصولات قابل تحویل و اهداف پروژه از طریق روش های کاملاً مشخص برای برقراری ارتباط با مشتری
- اندازه گیری فرایند موجود و خروجی آن برای تعیین کیفیت فعلی (جمع آوری معیارهای نقص)
- تحلیل معیارهای نقص و تعیین چند علت حیاتی

## ۵. قابلیت اطمینان نرم افزار

### ۵.۱. قابلیت اطمینان نرم افزار

- قابلیت اطمینان به زمان آماری: احتمال عملکرد بدون شکست یک برنامه کامپیوتری در محیطی مشخص برای یک زمان معین.
- مثلاً: برنامه ای در ۸ ساعت قابلیت اطمینان ۰.۹۹۹ دارد، یعنی در عرض ۸ ساعت پردازشگر ۱۰۰۰ بار اجرا کند ۹۹۹ بار درست کار می کند.

### ۵.۲. ایمنی نرم افزار

- ایمنی نرم افزار یکی از فعالیت های تضمین کیفیت است که بر شناسایی و سنجش ریسک های بالقوه ای تاکید دارد که ممکن است تاثیری منفی بر نرم افزار داشته و منجر به شکست کل سیستم شود.
- اگر بتوان ریسک ها را در همان ابتدای فرآیند نرم افزار شناسایی کرد می توان ویژگی هایی را در طراحی نرم افزار مشخص کرد که ریسک های بالقوه را حذف یا کنترل کند.

## ۶. استاندارد کیفیتی ISO 9001

### ۶.۱. استاندارد کیفیتی ISO 9001

- سیستم تضمین کیفیت را می توان به عنوان ساختار سازمانی، مسئولیت ها، روال ها، فرایندها و منابع پیاده سازی مدیریت کیفیت تعریف کرد.
- سیستم های تضمین کیفیت به سازمان کمک می کند تا اطمینان حاصل کند رضایت مشتری حاصل شده است.

## ۷. طرح SQA

راهنمایی برای نهادینه کردن تضمین کیفیت نرم افزار فراهم می آورد، ساختار آن بر اساس استاندارد IEEE عبارت است از:

- هدف و دامنه کاربرد طرح
- توصیفی از همه محصولات کاری مهندسی نرم افزار
- همه ی استانداردهای قابل استفاده در طول فرآیند نرم افزار
- وظایف و کنش های SQA از جمله مرورها و ممیزی ها
- ابزارها و روش هایی که از وظایف و کنش هایی که SQA پشتیبانی می کند.
- روش های مونتاژ ایمن سازی و نگهداری از کلیه سوابق مرتبط با SQA
- نقش ها و مسئولیت های سازمانی در قبال کیفیت محصول

## جلسه ۶- راهبردهای آزمون نرم افزار

### ۱. مقدمه

#### ۱.۱. راهبرد آزمون چیست؟

برای پیدا کردن خطاهای سهوی نرم افزار که در مراحل طراحی و پیاده سازی ایجاد شده اند.

#### ۱.۲. چرا اهمیت دارد؟

اگر آزمون بدون برنامه ریزی اجرا شود، زمان هدر می رود، کار بیهوده صرف می شود، خطاها کشف نشده باقی می مانند. بنابراین باید یک راهبرد برای آزمون نرم افزار وضع کرد.

#### ۱.۳. مراحل کار کدام است؟

آزمون از ابعاد کوچک آغاز می شود و به ابعاد بزرگ پیشرفت می کند.

#### ۱.۴. چه مواردی را توصیف می کند؟

راهبرد آزمون نرم افزار، نقشه راهنمایی فراهم می آورد که موارد زیر را توصیف می کند:

- مراحل اجرای آزمون
- زمان برنامه ریزی و میزان کار
- زمان و منابع لازم

### ۲. روشی راهبردی برای آزمون نرم افزار

آزمون مجموعه ای از فعالیت هاست که می توان آن ها را از قبل برنامه ریزی کرد و به طور سیستماتیک اجرا نمود. به همین دلیل، باید الگویی برای آزمون نرم افزار در فرایند نرم افزار تعریف شود.

#### ۲.۱. چند راهبرد آزمون

- انجام بازبینی ها و مرورهای فنی اثربخش
- شروع آزمون در سطح مولفه ها تا کل سیستم
- انجام آزمون توسط یک گروه آزمون گر مستقل

#### ۲.۲. واریسی و اعتبار سنجی

آزمون یکی از عناصر موضوع وسیع تر واریسی و اعتبار سنجی است.

| عناصر                       | توضیح                                                                                                                                                                                                    |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| وارسی<br>(Verification)     | <ul style="list-style-type: none"> <li>○ یک مجموعه از فعالیتهاست که پیاده سازی صحیح یک عملکرد خاص توسط نرم افزار را تضمین می کند.</li> <li>○ واری: آیا محصول را درست ساخته ایم؟</li> </ul>               |
| اعتبار سنجی<br>(Validation) | <ul style="list-style-type: none"> <li>○ مجموعه ای از فعالیت هاست که تضمین می کند که نرم افزار ساخته شده با خواسته های مشتری مطابقت دارد.</li> <li>○ اعتبارسنجی: آیا محصول درست را ساخته ایم؟</li> </ul> |

وارسی و اعتبارسنجی شامل موارد زیر هستند:

|                 |                      |                 |                                |                              |
|-----------------|----------------------|-----------------|--------------------------------|------------------------------|
| بازبینی مستندات | مطالعه امکان<br>سنجی | نظارت بر کارایی | ممیزی های<br>کیفیتی و پیکربندی | بازبینی های فنی<br>رسمی      |
| آزمون نصب       | آزمون قابلیت ها      | آزمون توسعه     | تحلیل الگوریتم ها              | بازبینی بانک های<br>اطلاعاتی |

### ۲.۳. گروه آزمون مستقل (ITG)

- نقش گروه آزمون مستقل، برطرف کردن مشکلات ذاتی است که در واگذاری آزمون به شخص سازنده وجود دارد.
- سازنده نرم افزار و ITG تعامل سازنده ای با یکدیگر دارند.

### ۲.۴. راهبردی برای آزمون نرم افزار

| آزمون            | توضیح                                                      |
|------------------|------------------------------------------------------------|
| آزمون واحدها     | بر هر واحد یا مولفه از نرم افزار تمرکز دارد.               |
| آزمون انسجام     | بر طراحی و معماری نرم افزار تاکید می شود.                  |
| آزمون اعتبارسنجی | نرم افزار بر اساس خواسته های مشتری اعتبارسنجی می شود.      |
| آزمون سیستم      | نرم افزار و دیگر عناصر سیستمی به صورت یک کل آزمون می شوند. |

### ۲.۵. مسائل راهبردی

- خواسته ها را به شیوه ای کمیت پذیر، مدت ها قبل از شروع آزمون مشخص کنید.
- اهداف و آزمون را به وضوح بیان کنید.
- کاربران نرم افزار را بشناسید و برای هر گروه از کاربران یک سابقه تهیه کنید.
- طرح آزمونی تهیه کنید که بر آزمون چرخه سریع تاکید داشته باشد.
- نرم افزاری پرقدرت بسازید که برای آزمودن خودش طراحی شده باشد.
- از بازبینی های فنی رسمی به عنوان فیلتر قبل از آزمون استفاده کنید.

## ۳. راهبردهای آزمون برای نرم افزارهای سنتی

| آزمون                              | توضیح                                                                                                                                                                              |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| آزمون واحدها (Unit Testing)        | ○ تلاش های واریسی مربوط به کوچکترین واحد طراحی نرم افزار یعنی مولفه ها و پیمانه ها را کانون توجه قرار می دهد.                                                                      |
| آزمون انسجام (Integration Testing) | ○ تکنیکی سیستماتیک برای ایجاد ساختار برنامه و در عین حال اجرای آزمون هایی جهت کشف خطاها در ایجاد واسط ها است.<br>○ انواع آزمون انسجام: انسجام بالا به پایین - انسجام پایین به بالا |

## ۴. راهبردهای آزمون برای نرم افزارهای شیءگرا

| آزمون                              | توضیح                                                                                                                                                                                                                                                                                                       |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| آزمون واحدها (Unit Testing)        | ○ کلاس ها و اشیاء<br>○ آزمون کلاس ها برای نرم افزارهای OO مشابه با آزمون پیمانه ها برای نرم افزارهای سنتی است.                                                                                                                                                                                              |
| آزمون انسجام (Integration Testing) | ○ از آن جا که نرم افزار های شیء گرا فاقد ساختار کنترلی سلسله مراتبی اند، راهبردهای سنتی بالا به پایین و بالعکس معنی ندارد. راهبردهای آزمون انسجام عبارتند از:<br>○ آزمون نخ ها (Threads): مجموعه ای از کلاس های لازم که به یک ورودی یا رویداد پاسخ می دهند.<br>○ آزمون مبتنی بر کاربرد برای کلاس های مستقل. |

## ۵. راهبردهای آزمون برای نرم افزارهای تحت وب

|                          |                   |                  |                                          |             |
|--------------------------|-------------------|------------------|------------------------------------------|-------------|
| گشت و گذار               | مولفه های عملیاتی | واسط کاربر       | مدل طراحی                                | مدل محتویات |
| آزمون توسط کاربران نهایی | آزمون های کارایی  | آزمون های امنیتی | آزمون در انواع پیکربندی های محیطی متفاوت |             |

## ۶. آزمون اعتبار سنجی

- در پایان آزمون انسجام آغاز می شود.
- اعتبارسنجی هنگامی موفق است که نرم افزار براساس انتظار مشتری عمل کند.
- اعتبار سنجی نرم افزار از طریق آزمون های جعبه سیاه انجام میشود که نشان دهنده ی مطابقت آن با خواسته ها هستند.
- آزمون آلفا: در مکان سازنده نرم افزار توسط مشتری انجام می شود. آزمونهای آلفا در محیطی کنترل شده اجرا می شوند.
- آزمون بتا: در یک یا چند مکان متعلق به مشتری انجام می شود و در آن سازنده حضور ندارد.

## ۷. آزمون سیستم

نرم افزار فقط عنصری از یک سیستم کامپیوتری بزرگ تر به شمار می رود.

| آزمون                              | توضیح                                                                                                                          |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| آزمون ترمیم ( Recovery Testing)    | ○ سیستم های کامپیوتری باید خود را در شرایط نقص ترمیم کنند و پردازش را ادامه دهند و در برخی موارد در برابر نقص تحمل داشته باشد. |
| آزمون امنیت ( Security Testing)    | ○ کوشش می کند تا واریسی کند که راهکارهای محافظ تعبیه شده در داخل سیستم واقعا آن را از نفوذ نامناسب حفظ می کند.                 |
| آزمون فشار ( Stress Testing)       | ○ برای مقابله با شرایط غیر عادی طراحی می شود. آزمون فشار سیستم رابه شیوه ای اجرا می کند که منابع را به میزان غیر عادی طلب کند. |
| آزمون کارایی (Performance Testing) | ○ به منظور آزمون کارایی نرم افزار در زمان اجرا در حیطه ی یک سیستم انسجام یافته طراحی می شود.                                   |
| آزمون استقرار (Deployment Testing) | ○ نرم افزار را در هر کدام از محیط هایی که قرار است در آن عمل کند تمرین می دهد.                                                 |

## ۸. هنر اشکال زدایی

- در نتیجه آزمون موفق برای رفع خطا رخ می دهد. اشکال زدایی آزمون نیست بلکه پس از آزمون رخ می دهد.
- هدف اشکال زدایی: یافتن و تصحیح علت یک خطای نرم افزاری.

| آزمون                        | توضیح                                                                                                                                                              |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| جستجوی جامع ( Brute Force)   | ○ وقتی استفاده می شود که همه ی روش های دیگر با شکست مواجه می شود. متداول ترین و کم اثربخش ترین روش                                                                 |
| عقب گرد (Backtracking)       | ○ در برنامه های کوچک اغلب موفق است و از محل خطا ردگیری شده تا علت خطا پیدا شود.                                                                                    |
| حذف علت ( Cause Elimination) | ○ داده های مرتبط با رخداد خطا سازماندهی می شوند تا علت های بالقوه مشخص شوند. یک فرضیه علت عنوان می شود و از داده های فوق برای اثبات یا انکار فرضیه استفاده می شود. |

## جلسه ۷- آزمون برنامه های کاربردی سنتی

### ۱. مقدمه

#### ۱.۱. آزمون نرم افزار چیست؟

هنگامی که کد تولید شد نرم افزار باید آزموده شود تا بتوان هر تعدادی از خطاها را قبل از تحویل به مشتری کشف کرد.

#### ۱.۲. چرا اهمیت دارد؟

- مرورها و فعالیت های SQA می توانند خطاها را کشف کنند ولی کافی نیستند.
- برای یافتن حداکثر تعداد ممکن خطاها، باید آزمون هایی را به طور سیستماتیک اجرا کرد.

### ۲. مبانی آزمون نرم افزار

- هدف آزمون یافتن خطاهاست.
- آزمون خوب آزمونی است که احتمال یافتن خطا را بالا می برد.
- آزمون پذیری: یک برنامه را چقدر می توان آسان آزمایش کرد.

#### ۲.۱. خصوصیات نرم افزار آزمون پذیر

- قابلیت کارکردن (Operability): هرچه بهتر کار کند آزمون آن اثر بخش تر است.
- قابلیت مشاهده (Visibility): آنچه می بینید همان است که آزمایش می کنید.
- کنترل پذیری (Controllability): هرچه بهتر بتوان نرم افزار را کنترل کرد آزمون را بیشتر می توان خودکار و بهینه کرد.
- پایداری (Stability): هرچه تعداد تغییرات کمتر باشد آزمون کمتر با مانع مواجه می شود.
- سادگی (Simplicity): هر چه مورد آزمون کوچکتر باشد، سریع تر می توان آن را آزمود.
- درک پذیری (Understandability): هر چه اطلاعات بیشتری داشته باشیم، آزمون هوشمندانه تر انجام می شود.

#### ۲.۲. خصوصیات آزمون خوب

- با احتمال زیادی خطاها را می یابد.
- دارای زوایا نیست.
- باید بهترین باشد.
- نباید بیش از حد ساده و نه بیش از حد پیچیده باشد.

### ۳. دیدگاه های درونی و بیرونی نسبت به آزمون

هرمحصول را می توان به یکی از دو شیوه زیر آزمود:

- دیدگاه بیرونی: با دانستن قابلیت عملیاتی مشخصی که یک محصول برای ارائه آن طراحی شده است. (آزمون جعبه سیاه)
- دیدگاه درونی: با دانستن کارکرد درونی یک محصول. (آزمون جعبه سفید) (آزمون جعبه شیشه ای)

#### ۴. آزمون جعبه سفید-جعبه شیشه ای

- برای به دست آوردن موارد آزمون، از ساختار کنترلی در برنامه استفاده می کنند.
- تضمین این که همه مسیرهای مستقل در یک پیمانه، حداقل یک بار امتحان شده اند.
- همه تصمیم گیری های منطقی را در دو بخش درست و غلط امتحان کنند.
- همه حلقه ها را در مرزهای و در داخل مرزهای عملیاتی آن ها اجرا کنند.
- ساختمان داده های داخلی را امتحان کنند تا اعتبار آن ها ثابت شود.

##### ۴.۱. آزمون مسیرهای پایه

- یک تکنیک آزمون جعبه سفید است.
- این روش طراح آزمون را قادر می سازد، تا میزانی منطقی از پیچیدگی رویه ای به دست آورد و از این میزان به عنوان راهنمایی جهت تعریف یک مجموعه پایه از مسیرهای اجرا استفاده کند.
- نماد گذاری گراف جریان برای نمایش جریان کنترل استفاده می شود.

##### ۴.۲. پیچیدگی سیکلوماتیک

- معیاری مفید برای پیش بینی پیمانه هایی است که احتمال مستعد خطا بودن آن ها بیشتر است که از آن برای برنامه ریزی آزمون ها و نیز طراحی موارد آزمون استفاده می شود.

##### ۴.۳. آزمون ساختار کنترلی

- آزمون شرط ها (Condition Testing): یک روش طراحی آزمون است که شرط های منطقی موجود در یک پیمانه برنامه را امتحان می کند.
- هر شرط ساده، یک متغیر بولی یا یک عبارت رابطه ای است که ممکن است قبل از آن یک NOT وجود داشته باشد.
- عبارت رابطه ای: E1 AND E2
- آزمون جریان داده ها (Data Flow Testing): در این روش مسیرهای آزمون یک برنامه، طبق موقعیت تعاریف و کاربردهای متغیرها در برنامه انتخاب می شوند.
- آزمون حلقه ها (Loop Testing): برای آزمون ساختمان حلقه استفاده می شود.
- ۴ دسته حلقه: حلقه های ساده، حلقه های تودرتو، حلقه های تسلسلی و حلقه های غیر ساخت یافته

#### ۵. آزمون جعبه سیاه

- آزمون رفتاری نیز خوانده می شود و بر خواسته های عملیاتی نرم افزار تکیه دارد. روش مکمل آزمون جعبه سفید است در مراحل آخر آزمون به کار می رود.



**۵.۱. روش های آزمون مبتنی بر گراف:**

- نخستین گام در آزمون جعبه سیاه شناخت اشیائی ست که در نرم افزار مدل سازی می شوند و نیز روابط میان این اشیاء است.
- گراف: نشان گر روابط میان اشیای داده و اشیای برنامه ای است و به کمک آن می توانید موارد آزموننی به دست آورید که خطاهای مرتبط با این روابط را جستجو می کنند.

**۵.۲. افراز هم ارزی (Equivalence Partitioning):**

- یکی از روش های آزمون جعبه سیاه است که دامنه ورودی یک برنامه را به دسته هایی از داده ها تقسیم می کند و موارد آزمون را می توان از روی آن به دست آورد.

**۵.۳. آزمون مبتنی بر مدل**

- یک تکنیک آزمون جعبه سیاه است و که از اطلاعات موجود در مدل خواسته ها به عنوان مبنایی برای تولید موارد آزمون بهره می برد.

**۶. آزمون محیط ها و معماری ها****۶.۱. آزمون واسط گرافیکی کاربر**

- آزموننی برای مولفه ها و اجزای واسط گرافیکی کاربر

**۶.۲. آزمون معماری های کلاینت-سرور**

- آزمون عملکردهای برنامه کاربردی
- آزمون سرور
- آزمون بانک اطلاعاتی
- آزمون تراکنش ها
- آزمون ارتباطات شبکه

**۶.۳. آزمون مستندات و تسهیلات راهنما**

- آزموننی برای یافتن خطاهای موجود در مستندات برنامه

**۶.۴. آزمون های مربوط به سیستم های بی درنگ**

- آزمون وظایف
- آزمون رفتاری
- آزمون سیستم

## جلسه ۸ – آزمون برنامه های شیء گرا

### ۱. مقدمه

#### ۱.۱. آزمون برنامه شیء گرا

- معماری نرم افزار های شیء گرا منجر به زیر سیستم های لایه ای می شود که کلاس های همکار را بسته بندی می کنند.
- لازم است سیستم شیء گرا را در چند سطح متفاوت مورد آزمون قرار دهید تا خطاهایی را که ممکن است در نتیجه ارتباط کلاس ها رخ دهد کشف کنیم.

### ۲. وسعت بخشیدن به دیدگاه آزمون

#### ۲.۱. وسعت بخشیدن به دیدگاه آزمون

- ساخت نرم افزار های شیء گرا با ایجاد مدل خواسته ها و طراحی آغاز می شود.
- در هر مرحله می توان این مدل ها را مورد آزمون قرار داد تا خطاها کشف شوند.

### ۳. آزمون مدل های تحلیل شیء گرا و طراحی شیء گرا

مدل های تحلیل و طراحی را نمی توان به صورت سنتی آزمود. زیرا آن ها را نمی توان اجرا کرد ولی مرورهای فنی رسمی را می توان برای بررسی درستی و سازگاری مدل های تحلیل و نیز مدل های طراحی به کار برد.

#### ۳.۱. درستی مدل ها

- براساس استفاده درست از نمادها قضاوت می شود.
- مدل ها مرور می شود تا اطمینان حاصل شود که قراردادهای متعارف مدل سازی رعایت شده اند.

#### ۳.۲. سازگاری مدل های شیء گرا

- این رامی توان با در نظر گرفتن روابط میان موجودیت های مدل مورد قضاوت قرار داد.

### ۴. راهبردهای آزمون شیء گرا

#### ۴.۱. آزمون واحدها (آزمون کلاس ها)

- در نرم افزارهای شیء گرا مفهوم واحد فرق می کند.
- به جای آزمون یک پیمانه منفرد، کوچکترین واحد قابل آزمون، کلاس یا شیء است.

#### ۴.۲. آزمون انسجام

- از آن جا که نرم افزارهای شی گرا فاقد ساختار کنترلی سلسله مراتبی اند، راهبردهای سنتی بالا به پایین و پایین به بالا چندان معنایی ندارند.

#### ۴.۲. دو راهبرد برای آزمون انسجام سیستم های شی گرا

- آزمون نخ ها
- آزمون مبتنی بر کاربرد

### ۵. روش های آزمون شی گرا

#### ۵.۱. طراحی موارد آزمون در مفاهیم شی گرا

- هدف نهایی طراحی موارد آزمون، کلاس های شی گرا هستند.
- موارد آزمون: بسته بندی کلاس: صفات و عملیات کلاس - وراثت - وراثت چندگانه

#### ۵.۲. قابلیت به کارگیری روش های سنتی در طراحی موارد آزمون

- روش های آزمون جعبه سفید را می توان برای عملیات تعریف شده در یک کلاس به کار برد.
- مسیر پایه، آزمون حلقه یا تکنیک های جریان داده ها می توانند اطمینان حاصل کنند که همه دستورهای یک عملیات، مورد آزمون قرار گرفته اند.
- روش های آزمون جعبه سیاه، برای سیستم های شی گرا همانند سیستم های سنتی مناسب هستند.

#### ۵.۳. آزمون مبتنی بر خطا

- هدف از آزمون مبتنی بر خطا در یک سیستم شی گرا، طراحی موارد آزمونی است که احتمال بالایی در کشف خطاهای ممکن دارد.
- در این راهبرد، مجموعه ای از خطاهای محتمل فرض می شود و سپس آزمون هایی برای اثبات هر فرض به دست می آید.

### ۶. روش های آزمون قابل اجرا در سطح کلاس ها

#### ۶.۱. روش های آزمون قابل اجرا در سطح کلاس ها

- آزمون تصادفی برای کلاس های شی گرا
- آزمون افراز در سطح کلاس ها
- طراحی موارد آزمون بین کلاس ها

## جلسه ۹- آزمون برنامه های کاربردی تحت وب

### ۱. طرح مساله

#### ۱.۱. طرح مساله

- اضطراری وجود دارد که همواره پروژه های تحت وب را در بر می گیرد.
- ذینفعان فشار می آورند تا برنامه را هر چه زودتر آنلاین کنند، به دلیل رقابت با سایر برنامه های تحت وب، فشار تقاضای کاربران و نگرانی از این که زمان مناسب در بازار از دست برود.
- در نتیجه برای آزمون برنامه فرصت کوتاهی داده می شود. که ممکن است منجر به برنامه ای بی کیفیت شود.
- آزمون نباید منتظر بماند تا پروژه به پایان برسد، آزمون از همان ابتدا شروع شود.

### ۲. مفاهیم آزمون برای برنامه های تحت وب

#### ۲.۱. مفهوم آزمون برای برنامه های تحت وب

- آزمون: فرایند تمرین دادن نرم افزار با هدف یافتن خطاهاست.
- برای درک اهداف آزمون در حیطه مهندسی وب باید ابعاد بسیاری از کیفیت برنامه های تحت وب را در نظر گرفت.

#### ۲.۲. ابعاد کیفیتی

- کیفیت در یک برنامه تحت وب، نتیجه طراحی خوب است. هم مرورها و هم آزمون ها، ابعاد کیفیتی را بررسی می کنند.
- محتوا در هر دو سطح نحوی و معنایی ارزیابی می شود. در سطح لغوی: املا، واژه، علامت گذاری، گرامر و ... و در سطح معنایی: درستی، سازگاری و فقدان ابهام بررسی می شود.

| مدل            | توضیح                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| قابلیت عملیاتی | ○ تا خطاهایی که عدم مطابقت با خواسته های مشتری را نشان می دهد، بر ملا شوند.                                                         |
| قابلیت استفاده | ○ تا اطمینان حاصل شود که هر گروه از کاربران به وسیله واسط، پشتیبانی می شود.                                                         |
| کارایی         | ○ تا اطمینان حاصل شود که سیستم، پاسخگوی تعامل های کاربر هست و حداکثر ازدحام بار را بدون تنزل غیر قابل قبول در عملکرد فراهم می سازد. |

#### ۲.۳. خطاهای موجود در یک محیط برنامه تحت وب

- چون بسیاری از آزمون های برنامه تحت وب مشکلاتی را آشکار می کنند که نخستین بار به چشم کلاینت می آید، غالباً نشانه ای از خطا را می بینند، نه خود خطا را.
- گرچه برخی خطاها نتیجه طراحی نادرست هستند ولی بسیاری از خطاها را می توان تا پیکر بندی برنامه تحت وب دنبال کرد.
- برخی خطاها از محیط عملیاتی ایستا ناشی می شود در حالی که خطاها را می توان تا پیکر بندی برنامه تحت وب دنبال کرد.

### ۳. راهبرد و فرایند آزمون

#### ۳.۱. راهبرد آزمون

- راهبرد آزمون: همان اصول پایه ای مربوط به آزمون همه نرم افزارها مدنظر است.
- که به طور خلاصه، مدل محتوای برنامه تحت وب مرور می شود تا خطاها برملا گردد.
- مدل واسط مرور می شود تا اطمینان حاصل شود که همه پرونده های کاربرد قابل پاسخگویی است.
- گشت و گذار در سراسر معماری آزمایش می شود.
- برنامه ریزی آزمون: برخی سازندگان برنامه ریزی نمی کنند آن ها فقط شروع می کنند به این امید که شاید یک شاهکار ایجاد کنند.
- ولی برنامه ریزی بسیار ضروری است.

#### ۳.۲. فرایند آزمون

- فرایند آزمون: با آزمون هایی آغاز می شود که محتوای عملیات را که به چشم کاربر نهایی می آید، تمرین می دهد.
- آزمون محتوا: در آزمون محتوا دو بخش مرور و تولید موارد آزمون قابل اجرا با هم ترکیب می شود.
- اهداف آزمون محتوا: کشف خطاهای نحوی، کشف خطاهای معنایی و خطاهای موجود در سازماندهی
- آزمون بانک اطلاعاتی: در این آزمون اشیای محتوایی پویاست.
- آزمون واسط کاربر: در سه نقطه متمایز رخ می دهد. طی مرحله تحلیل خواسته ها، طی طراحی و طی آزمون.

### ۴. راهبرد آزمون واسط

#### ۴.۱. راهبرد آزمون واسط

- راهبرد آزمون واسط: سازوکارهای تعامل را تمرین می دهد.
- جنبه های زیبایی شناختی واسط کاربر را اعتبار سنجی می کند.
- راهبرد کلی عبارت است از: کشف خطای مرتبط با سازو کارهای خاص واسط و کشف خطاهای موجود در روش پیاده سازی معنانشناسی گشت و گذار
- آزمون سازو کارهای واسط: هنگامی که یک کاربری با یک برنامه تحت وب تعامل می کند، این تعامل از طریق یک یا چند ساز و کار واسط رخ می دهد.
- آزمون معنا شناختی واسط: وقتی که هر کدام از سازوکارهای واسط، مورد آزمون واحد قرار گرفت، کانون توجه آزمون به معنانشناسی واسط تغییر می کند. و این را تعیین می کند که طراحی تا چه حد، کاربران را در نظر دارد، راهنمایی های روشنی را ارائه می دهد و بازخوردها را تحویل می دهد.

#### ۴.۲. آزمون های قابلیت استفاده

- آزمون های قابلیت استفاده مشابه با آزمایش معنا شناختی واسط است.
- از این لحاظ که این آزمون نیز میزان اثر بخش تعامل کاربران با برنامه تحت وب ارزیابی می شود.

- مرورها و آزمون های قابلیت استفاده به این منظور طراحی شوند که تعیین کنند واسط برنامه تحت وب تا چه میزان کارها را برای کاربر آسان می کند.

#### ۴.۳. آزمون های سازگاری

- کامپیوترها، دستگاه نمایش، سیستم عامل، مرورگرها می توانند تاثیر چشمگیری بر عملکرد برنامه تحت وب بگذارند.
- در برخی موارد مسائل سازگاری کوچک هیچ مشکل چشمگیری به بار نمی آورد. ولی در سایر موارد خطاهای جدی ممکن است مشاهده شود. مثلاً سرعت داندلود ممکن است غیر قابل قبول باشد. این مشکلات را قبل از آنلاین شدن برنامه تحت وب کشف کند.
- هدف این آزمون، کشف خطاها یا مشکلاتی در اجراست تا اختلاف های پیکربندی قابل رد گیری باشند.

#### ۴.۴. آزمون گشت و گذار

- نخستین مرحله از آزمون گشت و گذار، در واقع طی آزمون واسط آغاز می شود.
- سازوکارهای گشت و گذار، آزمایش می شوند تا اطمینان حاصل شود که هر کدام وظیفه خاص خود را اجرا می کند.
- آزمون معنا شناختی گشت و گذار: واحد معنا شناختی گشت و گذار به عنوان مجموعه ای از اطلاعات و ساختارهای گشت و گذاری مرتبط دانستیم که با همکاری یکدیگر، زیر مجموعه ای از خواسته های مرتبط کاربر را برآورده می سازند.
- آزمون گشت و گذار، همانند آزمون واسط و قابلیت استفاده، باید توسط هر تعداد ممکن از هر هیئت ها اجرا شوند.

### ۵. آزمون در سطح مولفه ها

#### ۵.۱. آزمون در سطح مولفه ها

- گاهی آزمون توابع نامیده می شود که مجموعه ای از آزمون ها را کانون توجه می دهد که سعی در کشف خطاهای موجود در توابع برنامه های تحت وب دارند.
- موارد آزمون در سطح مولفه ها به وسیله ورودی در سطح فرم ها طراحی می شود. هنگامی که داده های فرم ها تعیین شدند، کاربر یک دکمه یا سازوکار کنترلی دیگری را برای شروع اجرا انتخاب می کند.
- هر مورد آزمون در سطح مولفه ها کلیه مقادیر ورودی و خروجی مورد انتظار از مولفه ها را مشخص می کند.
- خروجی واقعی که به عنوان نتیجه ای از آزمون تولید می شود برای ارجاع های بعدی در اثنای پشتیبانی و نگهداری ثبت می شود.

### ۶. آزمون پیکربندی

#### ۶.۱. آزمون پیکربندی

- تغییر پذیری و ناپایداری، عوامل مهمی هستند که آزمون برنامه تحت وب را به چالش می کشد.
- وظیفه آزمون پیکر بندی، تمرین دادن هر پیکر بندی ممکن در طرف کلاینت نیست.

- در عوض باید مجموعه ای از پیکر بندی های محتمل در طرف کلاینت و در طرف سرور را بیازماید، تا اطمینان حاصل شود که تجربه ی کاربر، روی همه آن ها یکسان خواهد بود و خطاهایی را که ممکن است خاصی یک پیکربندی ویژه باشند، جدا کند.

## ۶.۲. مسائل طرف سرور

موارد آزمون پیکربندی در طرف سرور طوری طراحی می شوند که وارد می کنند آیا اطلاعات پیش بینی شده برای سرور می توانند بدون خطا برنامه تحت وب را پشتیبانی کنند. از جمله پرسش هایی که باید طی آزمون پیکر بندی طرف سرور پرسید عبارتند از:

- آیا برنامه تحت وب به طور کامل با سیستم عامل سرور سازگار است؟
- آیا برنامه تحت وب از انسجام مناسب با نرم افزار بانک اطلاعاتی برخوردار هستند؟
- آیا برنامه تحت وب به تفاوت نسخه در نرم افزار بانک اطلاعاتی حساس است؟

## ۶.۳. مسائل طرف کلاینت

در طرف کلاینت آزمون های پیکر بندی، بر سازگاری با پیکربندی های حاوی یک یا چند ترکیب جایگشتی از مولفه های زیر تاکید دارند:

- حافظه
- دیسک ها و دستگاه چاپ
- سخت افزار CPU
- یک سیستم عامل تلفن همراه
- سیستم عامل Linux و windows
- نرم افزار مرورگر Chrome و Firefox

## ۷. آزمون امنیت و کارایی

### ۷.۱. آزمون امنیت

- امنیت برنامه تحت وب موضوعی پیچیده است که باید پیش از دستیابی به آزمون امنیتی اثر بخش آن را به طور کامل درک کرد.
- آزمون های امنیتی طوری طراحی می شوند که آسیب پذیری های محیط طرف کلاینت ارتباطات شبکه ای که در تبادل اطلاعات میان سرور و کلاینت رخ می دهند و محیط سرور را بر ملا سازند.
- به هر کدام از دامنه ها می توان حمله کرد و کشف نقاط ضعفی که توسط افراد سودجو ممکن است مورد سوء استفاده قرار گیرد.

### ۷.۲. آزمون کارایی

- هیچ چیز ناراحت کننده تر از این که برنامه تحت وب نیست که چند دقیقه وقت صرف دانلود محتوا کند،

- در حالی که سایت های رقیب همان محتوا را در عرض چند ثانیه دانلود می کنند.
- از آزمون کارایی برای کشف مسائل کارایی استفاده می شود
- که ممکن است در اثر فقدان منابع طرف سرور، پهنای باند نامناسب برای شبکه و سایر مسائل نرم افزار که ممکن است به کاهش کارایی کلانیت - سرور بینجامد، رخ می دهد.

### ۷.۳. اهداف آزمون کارایی

آزمون کارایی برای شبیه سازی شرایط ازدحام بار در جهان واقعی طراحی می شوند. آزمون کارایی به پرسش های زیر پاسخ می دهد.

- کدام مولفه سیستم مسئول تنزل کارایی اند.
  - آیا تنزل کارایی بر امنیت سیستم تاثیر دارد
  - آیا تنزل کارایی بر درآمد شرکت تاثیر دارد.
- هدف آزمون ازدحام بار، تعیین چگونگی پاسخگویی برنامه های تحت وب و محیط سرور به انواع شرایط ازدحام بار است.

### ۷.۴. آزمون فشار

آزمون فشار ادامه آزمون ازدحام بار است. هدف از این آزمون ها پاسخ دادن به هر کدام از پرسش های زیر است:

- آیا با فراتر رفتن از ظرفیت ها، سیستم به ملایمت تنزل می یابد یا سرور از کار می افتد؟
- آیا با فرا رفتن از حد ظرفیت، تراکنش ها از بین می روند؟
- آیا با فرا رفتن از حد ظرفیت، انسجام داده ها تاثیر می پذیرد؟



## جلسه ۱۰ - طراحی مبتنی بر الگوها

### ۱. مقدمه

#### ۱.۱. طراحی مبتنی بر الگوها چیست؟

- در این شیوه، نرم افزار جدید، با یافتن مجموعه ای از راهکارهای اثبات شده در یک مجموعه مسائل ایجاد می شود.
- هر مساله به وسیله یک الگوی طراحی توصیف می شود که توسط سایر مهندسان نرم افزاری بررسی شده است، که هنگام طراحی برنامه های دیگر با این مساله مواجه شده اند و راهکاری برای آن پیاده سازی کرده اند.
- هر الگوی طراحی برای بخشی از مساله که قرار است حل شود، یک رویکرد و روش اثبات شده در اختیار شما می گذارد.

### ۲. الگوهای طراحی

#### ۲.۱. الگوهای طراحی چه هستند؟

- الگوهای طراحی را می توان یک قاعده سه بخشی دانست که واسطه میان یک حیطه معین، یک مساله و یک راهکار را بیان می کند.
- حیطه: این امکان را می دهد تا محیطی را که مساله در آن جای دارد درک کند و در یابد چه راهکارهایی ممکن است در این محیط مناسب باشد.
- مجموعه ای از خواسته ها، از جمله محدودیت ها و قید و بندها، به عنوان سیستم نیروهای تاثیرگذار بر شیوه تفسیر مساله در حیطه اش و چگونگی بکارگیری موثر آن راهکار عمل می کند.
- مفهوم اثبات شده است یعنی الگوها راهکارهایی را به دست می آورند که دارای سابقه باشند.
- الگوهای طراحی، اگر به طور موثر استفاده شود امکان طراحی نرم افزار بهتر را فراهم می کنند.

### ۳. انواع الگوها

#### ۳.۱. الگوها چه انواعی دارند؟

- الگوی مولد: علاوه بر توصیف مساله، حیطه و نیروها راهکاری عمل گرا شرح می دهد. یعنی الگویی که جنبه ای مهم و تکرارپذیر از سیستم را توصیف میکند و شیوه ساخت آن جنبه را ارائه میدهد.
- الگوی غیرمولد: تنها حیطه و مساله را توصیف می کند.
- یکی از دلایل علاقه به الگوهای طراحی آن است که انسان ها ذاتاً در تشخیص الگوها مهارت دارند.
- در جهان واقعی الگوهایی را می شناسیم که با گذر زمان و از طریق تجربه بدست می آیند.
- ما این الگوها را بلافاصله تشخیص می دهیم و ذاتاً می فهمیم که چه معنایی دارند و چگونه از آنها می توان استفاده کرد.
- در جستجوی شناسایی و مستندسازی الگوهای مولد هستیم. یعنی الگویی را شناسایی می کنیم که جنبه ای مهم و تکرارپذیر از سیستم را توصیف می کند.

| انواع الگو           | توضیح                                                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| الگوهای طراحی        | ○ شامل طیف گسترده ای از انتزاع ها و کاربردها                                                                                          |
| الگوهای معماری       | ○ مسائل طراحی گسترده ای را توصیف می کنند که با بکارگیری یک رویکرد ساختاری حل می شوند.                                                 |
| الگوهای داده ای      | ○ مسائل داده گرای تکراری و مسائل مدل سازی داده ها را توصیف می کنند که در حل این مسائل قابل استفاده اند.                               |
| الگوهای مولفه ای     | ○ به مسائل مرتبط با توسعه زیر سیستم ها و مولفه ها ، شیوه برقراری ارتباط آنها با یکدیگر و تعیین مکان آنها در معماری بزرگتر می پردازند. |
| الگوی های طراحی واسط | ○ مسائل واسط کاربری متداول و راهکار آنها را توصیف می کنند.                                                                            |
| الگوهای تحت وب       | ○ به مسائلی اختصاص دارند که هنگام ساخت این نوع برنامه ها مشاهده می شوند.                                                              |
| الگوی ایجاد          | ○ ایجاد، ترکیب و نمایش اشیاست.                                                                                                        |
| الگوی ساختاری        | ○ مسائل و راهکارهای مرتبط با چگونگی سازمان دهی و انسجام بخشیدن به اشیاء برای ایجاد ساختاری بزرگتر                                     |
| الگوی رفتاری         | ○ مسائل مرتبط با تقسیم مسئولیت ها میان اشیاء و شیوه تاثیر گرفتن ارتباط میان اشیاء.                                                    |

## ۴. چارچوب ها

### ۴.۱. چارچوب ها چه هستند؟

- در برخی موارد ممکن است برای طراحی، نیاز به فراهم آوردن زیرساخت مختص یک پیاده سازی باشد که به آن چارچوب گفته می شود
- چارچوب اسکلتی است با مجموعه ای از نقاط اتصال (یا قلاب ها ) که به کمک آنها می توان این اسکلت را بر یک دامنه مساله خاص تطبیق داد.
- این نقاط اتصال به شما امکان می دهند تا کلاس و قابلیت های عملیاتی خاص مساله را در این اسکلت انسجام بخشید.

### ۴.۲. الگوها و چارچوب ها چه تفاوتی دارند؟

- الگوهای طراحی، انتزاعی از چارچوب ها هستند.
- الگوهای طراحی، عناصر معماری کوچکتر از چارچوب ها هستند.
- تخصص یافتگی الگوهای طراحی کمتر از چارچوب ها هستند.

## ۵. توصیف الگوها

### ۵.۱. چگونه الگوها را توصیف کنیم؟

- طراحی مبتنی بر الگو، با شناسایی الگوها آغاز، با جستجو برای تعیین این که آیا دیگران به این الگو پرداخته اند ادامه می یابد و با بکارگیری الگوی مناسب برای مساله پایان می یابد.
- نام الگوها باید با احتیاط انتخاب گردد، یکی از مسائل فنی در طراحی مبتنی بر الگوها ناتوانی یافتن الگوها در میان صدها یا هزاران الگوست.

- قالب الگو، ابزاری استاندارد برای توصیف الگوی طراحی فراهم می آورد.

### قالب الگوی طراحی

|                                                                                                          |                                                                                             |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>نام الگو:</b> توصیف جوهره الگو با نامی کوتاه، انتخاب نامی بامعنا به جستجو دنبال الگو کمک بسیاری میکند |                                                                                             |
| <b>حیطه:</b>                                                                                             | <b>مساله</b>                                                                                |
| توصیف محیط مساله و دامنه کاربرد                                                                          | <b>انگیزش:</b> مثالی از مساله                                                               |
|                                                                                                          | <b>نیروها:</b> فهرست نیروهای تاثیرگذار بر شیوه حل مساله شامل محدودیت ها و قید و بندها       |
|                                                                                                          | <b>راهکار:</b> توصیف راهکار                                                                 |
|                                                                                                          | <b>هدف:</b> توصیف الگو و آنچه انجام می دهد.                                                 |
| <b>الگوهای مرتبط:</b>                                                                                    | <b>همکاری ها:</b> چگونگی مشارکت سایر الگوها در راهکار                                       |
|                                                                                                          | <b>پیامدها:</b> توازن های بالقوه که در پیاده سازی باید در نظر گرفت و پیامدهای استفاده از آن |
|                                                                                                          | <b>پیاده سازی:</b> مسائلی که هنگام پیاده سازی که باید در نظر گرفته شود.                     |
|                                                                                                          | <b>موارد الگوهای شناخته شده:</b> مثالهایی از موارد استفاده واقعی الگو در برنامه های کاربردی |

## ۶. مخازن و زبان های الگو

### ۶.۱. مخازن و زبان های الگو

- زبان الگوها شامل مجموعه ای از الگوها می شود که هر یک با به کارگیری یک قالب استاندارد شده توصیف می شود و با سایر الگوهای مربوطه ارتباط داده می شود تا مسائل موجود در یک دامنه کاربرد را با همکاری یکدیگر حل کنند
- زبان الگوها مشابه با یک جزوه راهنمای ابرمتنی است که برای حل مساله در یک دامنه خاص بکار می رود .
- ده ها زبان الگو برای طراحی نرم افزار پیشنهاد شده است در اکثر موارد، الگوهای طراحی که بخشی از زبان الگوها هستند در مخزنی قرار داده می شوند که از طریق وب قابل دستیابی است و این مخزن نمایه ای از تمامی الگوهای طراحی فراهم می آورد.

## ۷. طراحی مبتنی بر الگوها در حیطه

### ۷.۱. طراحی مبتنی بر الگوها در حیطه چیست؟

- طراحی مبتنی بر الگو در خلا به کاربرده نمی شود.
- مفاهیم و تکنیک های بحث شده برای طراحی معماری، طراحی در سطح مولفه ها و طراحی واسط در ارتباط با رویکرد مبتنی بر الگو به کار برده می شوند.
- هنگامی که کار خود را به عنوان طراح شروع کردید، همواره باید صفات کیفیتی را مدنظر داشته باشید.
- این صفات راهی برای ارزیابی کیفیت نرم افزار تعیین می کنند ولی در دستیابی به آن کمک چندانی ارائه نمی دهند.

## ۸. وظایف طراحی

**۸.۱. وظایف طراحی**

- مدل خواسته ها و توسعه سلسله مراتبی از مسائل را بررسی کنید.
- تعیین کنید آیا زبان الگوی مناسبی برای مساله وجود دارد یا خیر.
- با شروع از یک مساله گسترده تعیین کنید آیا یک یا چند الگوی معماری بر آن در دسترس است یا خیر.

**۹. ساخت جداول سازمان دهی الگو****۹.۱. ساخت جداول سازمان دهی الگو**

| زیرساخت ها | پیاده سازی | کاربرد     | بانک اطلاعاتی |                   |
|------------|------------|------------|---------------|-------------------|
|            |            |            |               | داده ها (محتویات) |
|            |            |            | نام الگوها    | صورت مساله        |
|            | نام الگوها |            |               | صورت مساله        |
|            |            |            |               | معماری            |
|            |            | نام الگوها |               | صورت مساله        |
|            |            |            |               | در سطح مولفه      |
|            | نام الگوها | نام الگوها |               | صورت مساله        |
|            |            |            |               | واسط کاربر        |
|            | نام الگوها | نام الگوها |               | صورت مساله        |

**۱۰. الگوها****۱۰.۱. الگوهای معماری**

- هر معماری نرم افزار ممکن است چند الگوی معماری داشته باشد که به مسائل گوناگون از قبیل همروندی، ماندگاری و توزیع مربوط می شوند.
- چند دامنه الگوی معماری: کنترل دستیابی، همروندی، توزیع و ماندگاری

**۱۰.۲. الگوهای طراحی در سطح مولفه ها**

- الگوهای طراحی در سطح مولفه ها راهکارهای اثبات شده در اختیار شما قرار می دهند که به یک یا چند مساله فرضی استخراج شده از مدل خواسته های می پردازند.
- در بسیاری موارد، الگوهای طراحی از این نوع یک عنصر از سیستم را کانون توجه قرار می دهند.

### ۱۰,۳. الگوهای طراحی واسط کاربر

- واسط کاربر کامل: راهنمایی برای طراحی ساختار سطح بالا و گشت و گذار در سرتاسر واسط فراهم می آورد.
- الگوی Top Level Navigation: هنگامی استفاده می شود که یک سایت یا برنامه کاربردی چند قابلیت عملیاتی عمده را پیاده سازی کند. یک منوی سطح بالا فراهم می آورد که با لوگو یا آرمی همراه است که گشت و گذار مستقیم در هر کدام از قابلیت های عمده را فراهم می سازد.

### ۱۰,۴. الگوهای طراحی برای برنامه های تحت وب

- هنگام پرداختن به مسائل طراحی مرتبط با ساخت برنامه های تحت وب، در نظر گرفتن گروه های الگو با در نظر گرفتن دو بعد می تواند مفید واقع گردد:
- کانون طراحی الگو: مشخص می کند که کدام جنبه از مدل طراحی مد نظر است.
- سطح دانه بندی: سطح انتزاع را مشخص می کند.

### ۱۰,۵. کانون طراحی

- در سطح بالایی از انتزاع آغاز می شود، و به تدریج به جزئیات و مشخصات آن ها افزوده می شود.
- به بیان دیگر، کانون طراحی با نزدیکتر شدن به طراحی باریک تر می شود.
- مسائلی که هنگام طراحی معماری اطلاعاتی برای برنامه تحت وب به آنها بر می خورید با مسائل که هنگام اجرای طراحی واسط مشاهده می کنید، تفاوت دارد.
- الگوهای برنامه تحت وب را می توان بر اساس سطح کانون طراحی گروه بندی کرد: الگوی معماری اطلاعاتی، الگوهای گشت و گذار، الگوهای ارائه و الگوهای عملیاتی.

### ۱۰,۶. دانه بندی طراحی

- از نظر سطح دانه بندی الگوها را می توان در سطوح زیر تعریف کرد: الگوهای معماری، الگوهای طراحی و الگوهای مولفه ها

## جلسه ۱۱- مدل سازی و واری رسمی

### ۱. مقدمه

#### ۱.۱. مساله چیست؟

- کار را همان اول درست انجام بده.
- در ساخت نرم افزار اگر این روال را پیش بگیرید، تلاش کمتری صرف دوباره کاری بیهوده خواهد شد.
- دو روش مهندسی نرم افزار رسمی: اتاق تمیز و روش های رسمی
- با فراهم ساختن رویکرد ریاضی-محور برای برنامه ریزی، مدل سازی و توانایی واری مدل حاصل، به تیم کمک می کنند تا کار را در همان بار نخست درست انجام دهد.

#### ۱.۲. مدل سازی و واری رسمی چیست؟

- مهندسی نرم افزار اتاق تمیز بر واری ریاضی پیش از شروع ساخت برنامه و بر تایید قابلیت اطمینان به عنوان بخشی از فعالیت آزمون تاکید دارد.
- در روش های رسمی از نظریه مجموعه ها و نماد گذاری منطقی برای ایجاد بیان واضحی از حقایق (خواسته ها) استفاده می شود.
- که می توان آن ها را برای بهبود بخشیدن به صحت و درستی (و حتی اثبات آن) تحلیل کرد.
- خط مبنای هر دو روش، ایجاد نرم افزاری با مقادیر بسیار پایین خطاست.
- انجام دهنده: یک مهندس نرم افزار که آموزش های خاص را در این زمینه دیده باشد.

#### ۱.۳. چرا اهمیت دارد؟

- اشتباهات باعث دوباره کاری می شوند.
- دوباره کاری زمان می برد و هزینه ها را افزایش می دهد.
- بهتر نیست اگر بتوانیم تعداد اشتباهات (خطاها) را در زمان طراحی و ساخت نرم افزار کاهش دهیم؟
- مدل سازی رسمی نوید بخش همین مزیت است.

#### ۱.۴. مراحل کار کدام است؟

- مدل های خواسته ها و طراحی با به کارگیری یک نمادگذاری تخصصی ایجاد می شوند که قابلیت واری ریاضی را داشته باشند.
- مهندسی نرم افزار اتاق تمیز، از نمایش ساختارهای چهارگوش استفاده می کند که سیستم (یا جنبه ای از سیستم) را در سطح مشخصی از انتزاع کپسوله می کنند.
- واری، هنگامی به کار برده می شود که طراحی ساختارهای چهارگوش کامل باشد.
- هنگامی که صحت برای هر ساختار چهارگوش به تایید رسید، آزمون کاربرد آماری آغاز می شود.

- روش های رسمی، با به کارگیری نمادها و مفاهیم نظریه مجموعه ها برای تعریف داده های ثابت، حالت ها و عملیات های مربوط به یک وظیفه سیستمی، خواسته های نرم افزار را به نمایشی رسمی تر ترجمه می کنند.

### ۱.۵. محصول کاری چیست؟

- مدلی تخصص یافته و رسمی از خواسته ها تهیه می شود
- نتایج واریسی ها و آزمون های کاربرد آماری ثبت می شود.
- آزمون کاربرد آماری، سناریوهای کاربرد را تمرین می دهد تا اطمینان حاصل شود که خطاهای موجود در قابلیت عملیاتی کاربر کشف و تصحیح شوند.

### ۱.۶. روش های مدل سازی رسمی

- بر خلاف مرور و آزمون که پس از توسعه ی مدل ها و کدها آغاز می شوند، مدل سازی و واریسی رسمی شامل روش های مدل سازی تخصص یافته ای می شود که در کنار رویکردهای واریسی تجویزی به کار برده می شوند.
- بدون رویکردهای مدل سازی مناسب، واریسی را نمی توان به انجام رساند.
- هر دو روش اتاق تمیز و روش های رسمی، کاملاً دشوارند و هیچ یک از آن ها به طور گسترده توسط جامعه مهندسی نرم افزار به کار گرفته نمی شوند.
- ولی اگر قصد دارید یک نرم افزار ضد گلوله بسازید، این روش ها شما را یاری می کنند.

## ۲. راهبرد اتاق تمیز

### ۲.۱. راهبرد اتاق تمیز

- مهندسی نرم افزار افزار اتاق تمیز از یک نسخه تخصص یافته از مدل نرم افزار افزایشی معرفی شد.
- تیم نرم افزاری لوله ای از نسخه های نرم افزار را توسعه می دهند.
- هر نسخه پس از تایید به کل سیستم افزوده می شود.
- از این رو قابلیت عملیاتی سیستم با گذر زمان رشد می کند.

### ۲.۲. وظایف لوله نسخه های اتاق تمیز

در داخل لوله نسخه های اتاق تمیز، وظایف زیر باید به انجام برسد:

|            |                           |                                     |                                 |
|------------|---------------------------|-------------------------------------|---------------------------------|
| طراحی رسمی | مشخصه ی ساختار<br>چهارگوش | جمع آوری خواسته<br>ها               | برنامه ریزی برای<br>نسخه ها     |
| صدور گواهی | آزمون کاربرد آماری        | برنامه ریزی برای<br>آزمون های آماری | تولید، بازرسی و<br>واریسی کدها. |

## ۲.۳. برنامه ریزی برای نسخه ها

- یک طرح پروژه تهیه می شود که راهبرد افزایشی در آن انتخاب می شود.
- قابلیت عملیاتی هر نسخه، اندازه پیش بینی شده برای آن و یک زمان بندی برای توسعه اتاق تمیز ایجاد می شود.
- مراقبت ویژه ای باید به عمل آید تا اطمینان حاصل شود که نسخه هایی تایید شده، سر وقت منسجم می شوند.

## ۲.۴. جمع آوری خواسته ها و مشخصه ساختار چهارگوش

جمع آوری خواسته ها:

- با استفاده از تکنیک هایی، توصیف مشروحتری از خواسته ها در سطح مشتری برای هر کدام از نسخه ها تهیه می شود.
- مشخصه ساختار چهارگوش:
- یک روش تعیین مشخصات که در آن از ساختارهای چهارگوش برای توصیف مشخصات عملیاتی استفاده می شود.
- ساختارهای چهارگوش تعریف خلاقانه رفتار، داده ها، و روال های موجود در هر سطح از پالایش را جداسازی و تفکیک می نماید.

## ۲.۵. طراحی رسمی

- با استفاده از رویکرد ساختارهای چهارگوش، طراحی اتاق تمیز بسطی طبیعی و یکپارچه از مشخصات است.
- گرچه می توان میان این دو فعالیت، تمایز قائل شد، مشخصات (که چهارگوش سیاه نامیده می شوند) طی چند دور تکرار (در یک نسخه) پالایش می شوند تا به طراحی های معماری یا در سطح مولفه ها (موسوم به چهارگوش های حالت و چهارگوش های شفاف) شباهت پیدا کند.

## ۲.۶. واریسی

- تیم اتاق تمیز یک سری فعالیت های شدید واریسی را روی طراحی و سپس روی کدها اجرا می کند.
- واریسی با بالاترین سطح از ساختارهای چهارگوش (مشخصات) آغاز می شود و به سوی جزئیات طراحی و کد حرکت می کند.
- نخستین سطح از واریسی با بکارگیری مجموعه ای از پرسش های صحت آغاز می شود.

## ۲.۷. تولید، بازرسی و واریسی کدها

- مشخصات ساختارهای چهارگوش که به زبانی تخصص یافته ارائه می شوند، به زبان برنامه نویسی مناسب ترجمه می شوند.
- سپس از مروری فنی برای حصول اطمینان از مطابقت با ساختارهای چهارگوش و کدها و صحت نحوی کدها استفاده می شود.
- پس از آن، واریسی برای کد منبع انجام می شود.

## ۲.۸. برنامه ریزی برای آزمون های آماری

- کاربرد پیش بینی شده برای نرم افزار تحلیل می شود و مجموعه ای از موارد آزمون، برنامه ریزی و طراحی می شود تا توزیع احتمالی از این کاربرد را تمرین دهد.



- این فعالیت اتاق تمیز به موازات تعیین مشخصات، واریسی و تولید کد اجرا می شود.

## ۲.۹. آزمون کاربرد آماری و صدور گواهی

- آزمون کاربرد آماری: در تکنیک های کاربرد آماری یک سری آزمون اجرا می شود که از یک نمونه آماری از کلیه اجراهای ممکن برنامه توسط کلیه کاربران جمعیت هدفمند به دست می آید.
- صدور گواهی: هنگامی که واریسی، بازرسی و آزمون کاربرد به انجام رسید و همه خطاها تصحیح شد، آماده بودن نسخه برای انسجام یافتن به سیستم به تایید می رسد.

## ۲.۱۰. مشخصات عملیاتی

- در رویکرد مدل سازی در مهندسی نرم افزار اتاق تمیز، از روش موسوم به تعیین مشخصات ساختارهای چهارگوش استفاده می شود.
- یک چهارگوش سیستم (یا جنبه ای از سیستم) را در سطحی از جزئیات پنهان سازی می کند.
- از طریق فرایند پالایش مرحله به مرحله و پرداختن به جزئیات، چهارگوش ها به صورت یک سلسله مراتب پالایش می شوند که در آن هر چهارگوش دارای شفافیت ارجاعی است.

## ۳. مشخصات عملیاتی

### ۳.۱. مشخصات عملیاتی

- در اتاق تمیز، از روش تعیین مشخصات ساختارهای چهارگوش استفاده می شود.
- یک چهارگوش سیستم را در سطحی از جزئیات پنهان سازی می کند.
- از طریق فرایند پالایش مرحله به مرحله و پرداختن به جزئیات، چهارگوش ها به صورت یک سلسله مراتب پالایش می شوند که در آن هر چهارگوش دارای شفافیت ارجاعی است.
- یعنی محتوای اطلاعاتی هر مشخصه برای تعریف پالایش آن کفایت می کند، بدون این که به پیاده سازی هیچ جعبه دیگری وابسته باشد.
- به این ترتیب، تحلیل گر می تواند سیستم را به صورت سلسله مراتبی با حرکت از نمایش اساسی در بالا تا جزئیات خاص پیاده سازی در پایین، افراز کند.

### ۳.۲. از سه نوع چهارگوش استفاده می شود.

| توضیح                                                                                                                                                                                                                                           | چهارگوش      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <ul style="list-style-type: none"> <li>○ رفتار یک سیستم یا بخشی از یک سیستم را مشخص می کند.</li> <li>○ سیستم با بکارگیری مجموعه ای از قواعد انتقال که محرک را به پاسخ تبدیل می کنند به محرک ها (رویدادها) پاسخ می دهند.</li> </ul>              | چهارگوش سیاه |
| <ul style="list-style-type: none"> <li>○ سرویس ها (عملیات ها) و داده های حالت را به شیوه ای مشابه با اشیا کپسوله می کنند.</li> <li>○ در این نما از تعیین مشخصات، ورودی های چهارگوش حالت (محرک ها) و خروجی ها (پاسخ ها) عرضه می شوند.</li> </ul> | چهارگوش حالت |

|                                                                                 |              |
|---------------------------------------------------------------------------------|--------------|
| ○ چهارگوش حالت، تاریخچه محرک چهارگوش سیاه را نیز نشان می دهد.                   |              |
| ○ یعنی داده های کپسوله شده در چهارگوش حالت که باید بین گذارهای موجود حفظ شوند.  |              |
| ○ توابع گذاری که توسط چهارگوش حالت مشخص می شوند، در چهارگوش شفاف تعریف می شوند. | چهارگوش شفاف |
| ○ یک چهارگوش شفاف حاوی طراحی روالی برای چهارگوش حالت است.                       |              |

## ۴. طراحی و آزمون اتاق تمیز

### ۴.۱. طراحی اتاق تمیز

- در مهندسی نرم افزار اتاق تمیز، از فلسفه برنامه نویسی ساخت یافته، استفاده ی گسترده به عمل می آید.
- ولی در این مورد، برنامه نویسی ساخت یافته بسیار شدیدتر استفاده می شود.
- توابع پردازشی پایه که طی پالایش های اولیه مشخصات توصیف می شوند،
- با استفاده از یک روش بسط مرحله ای توابع ریاضی به ساختارهای زیر تابع ها و ارتباط های منطقی پالایش می شوند،
- که در آن، این بسط، آن قدر ادامه می یابد، تا همه زیر تابع های شناسایی شده را بتوان مستقیماً به زبان برنامه نویسی به کار رفته برای پیاده سازی بیان کرد

### ۴.۲. آزمون اتاق تمیز

- راهبرد و تاکتیک های آزمون اتاق تمیز با رویکردهای سنتی آزمون تفاوت بنیادی دارند.
- در روش های سنتی یک مجموعه موارد آزمون به دست می آید که خطاهای طراحی و کد نویسی را کشف می کنند.
- هدف آزمون اتاق تمیز، اعتبار سنجی خواسته های نرم افزار است،
- و برای این منظور نشان می دهد که یک نمونه آماری از موارد آزمون با موفقیت اجرا شده است.

### ۴.۳. آزمون کاربرد آماری

- عبارت است از آزمودن نرم افزار به شیوه ای که کاربران تمایل به استفاده از آن دارند.
- تیم های آزمون اتاق تمیز (تیم های تایید) برای نیل به این مقصود، باید یک توزیع احتمال کاربرد برای نرم افزار تعیین کنند.
- مشخصات (چهارگوش سیاه) برای هر نسخه از نرم افزار تحلیل می شود تا مجموعه ای از محرک ها (رویدادها یا ورودی ها) که باعث تغییر رفتار نرم افزار می شوند، تعریف شوند.
- برای هر مجموعه از محرک ها مطابق با توزیع احتمال کاربرد، موارد آزمونی ایجاد می شود.

### ۴.۴. صدور گواهی

تایید به این معناست که قابلیت اطمینان را که توسط زمان میانگین شکست سنجیده می شود برای هر یک از مولفه ها می توان مشخص کرد. رویکرد تایید شامل ۵ مرحله است:

- ایجاد سناریوهای کاربرد
- مشخص شدن پروفایل کاربرد
- ایجاد موارد آزمون از روی پروفایل

- اجرای آزمون ها و ثبت و تحلیل داده های مربوط به شکست ها
- محاسبه و تایید قابلیت اطمینان

## ۵. مفاهیم روش های رسمی

### ۵.۱. تعریف روش های رسمی

- تکنیک هایی با اساس و پایه ریاضی برای توصیف خواص سیستم.
- چارچوب هایی فراهم می آورند که از طریق آن ها می توان سیستم ها را به شیوه ای سیستماتیک و پیش بینی شده مشخص کرد، توسعه داد و واریسی کرد.

### ۵.۲. چرا روش های رسمی؟

- سازگاری، کامل بودن و فقدان ابهام روش های تعیین مشخصات رسمی
- تفسیر خواسته ها یا طراحی را در جهت حذف ابهامی که غالبا هنگام تفسیر یک زبان طبیعی (فارسی) یا نمادگذاری گرافیکی (UML) توسط خواننده رخ می دهد، میسر می کند.
- به کمک امکانات توصیفی نظریه مجموعه ها و نمادگذاری منطقی، می توان خواسته ها را به وضوح و روشنی بیان کرد.

### ۵.۳. مفاهیم روش های رسمی

| مفاهیم      | توضیح                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| جدول نمادها | <ul style="list-style-type: none"> <li>این جدول شامل مجموعه ای از آیتم ها بدون هر گونه تکرار می شود.</li> <li>ثابت داده ای: شرطی است که در سرتاسر اجرای سیستمی حاوی مجموعه ای از داده ها، درست است.</li> </ul>                                                                                                                                                                                                                            |
| حالت        | <ul style="list-style-type: none"> <li>بر اساس تعریف زبان رسمی OCL: سیستم می تواند در یکی از چند حالت خود به سر برد که هر کدام یک شیوه رفتاری قابل مشاهده از بیرون را از خود نشان می دهند.</li> <li>بر اساس تعریف زبان رسمی Z: حالت یک سیستم با داده های ذخیره شده در آن نمایش داده می شود. Z تعداد حالت های بسیار بیشتری را پیشنهاد می دهد که هر کدام از پیکربندی های ممکن برای داده ها را نشان می دهد. حالت جدول نمادها است.</li> </ul> |
| عملیات      | <ul style="list-style-type: none"> <li>فعالیتی است که در داخل سیستم انجام می شود و داده ها را می خواند یا می نویسد.</li> <li>مثال: عملیات add() برای افزودن نام مشخصی به جدول نمادها و عملیات remove() برای حذف یک نام از جدول.</li> </ul>                                                                                                                                                                                                |

## ۶. زبان های تعیین مشخصات رسمی

### ۶.۱. مولفه های زبان های تعیین مشخصات رسمی

- قالب نحوی که تعیین می کند مشخصات با چه نمادگذاری خاص باید ارائه شود. مبتنی بر نمادگذاری نظریه مجموعه ها و جبر گزاره ها

- معناساختی: برای کمک به تعریف مجموعه اشیای مرجع که برای توصیف سیستم بکار می رود. نشان می دهد این زبان چگونه خواسته های سیستم را نمایش می دهد. مجموعه ای از روابط که قواعدی را تعریف می کنند که نشان می دهند کدام اشیا به طور مناسب در مشخصه صدق می کنند.
- زبان های رسمی: OCL - Z - VDM - LARCH

## جلسه ۱۲ – مدیریت پیکربندی نرم افزار

### ۱. مقدمه

- زمانی که یک نرم افزار ساخته می شود، تغییراتی در آن داده می شود که در جهت کنترل این تغییرات از مدیریت پیکربندی نرم افزار، استفاده می شود.
- وظیفه SCM: شناسایی اصلاحات، کنترل اصلاحات، ممیزی اصلاحات و گزارش اصلاحات

### ۲. مدیریت پیکربندی نرم افزار

#### ۲.۱. مدیریت پیکربندی نرم افزار

- خروجی فرآیند نرم افزار، اطلاعاتی است که به سه گروه عمده قابل تقسیم است: برنامه های کامپیوتری، مستندات، داده ها، که در مجموع پیکربندی نرم افزار نام دارند.
- با پیشرفت فرآیند نرم افزار، تعداد آیتم های پیکربندی نرم افزار (SCI) به سرعت رشد می کند.

#### ۲.۲. منابع اصلی تغییر

- چهار منبع اصلی برای این تغییرات ذکر شده است.
- شرایط بازاری، تجاری جدید
- نیازهای جدیدی که نفع که اصلاحاتی را در داده تولید شده توسط سیستم های اطلاعاتی طلب می کنند.
- سازماندهی مجدد یا رشد.
- قید و بندهای بودجه ای یا زمان بندی که باعث تعریف مجدد محصول می شود.

#### ۲.۳. سناریوی SCM

- یک سناریوی SCM شامل این اجزا است: مدیر پروژه، مدیر پیکربندی، مهندسان نرم افزار و مشتری

#### ۲.۴. مدیر پروژه

- مسئولیت گروه نرم افزار را به عهده دارد.
- هدف مدیر حصول اطمینان از این که محصول در یک چارچوب زمانی معین توسعه می یابد. از این رو مدیر مشکلات را تشخیص می دهد.
- این کار با ایجاد و تحلیل گزارش در خصوص وضعیت سیستم نرم افزار انجام می شود.

#### ۲.۵. مدیر پیکربندی

- مسئولیت روال ها و خط مشی های SCM را بر عهده دارد.
- ساز و کارهایی را برای مجاز ساختن تغییرات معرفی می کند.

**۲.۶. مهندسان نرم افزار**

- مسئول تولید و نگهداری نرم افزار هستند که هدف آنان کار کردن اثر بخش است.
- مهندسان برای ایجاد تغییر، آزمون و منسجم ساختن کدها فضای کاری خاصی خود را خواهند داشت.
- در نقطه ای معین از کدها یک خط مبنا ساخته می شود که توسعه بیش تر بر اساس آن خط مبنا است.

**۲.۷. مشتری**

- از محصول استفاده می کند و روال های رسمی مربوط به تغییرات را در محصول دنبال می کنند.

**۲.۸. عناصر سیستم مدیریت پیکر بندی**

| عناصر          | توضیح                                                                            |
|----------------|----------------------------------------------------------------------------------|
| عناصر مولفه ای | ○ مجموعه ای از ابزارهای نهاد شده در داخل یک سیستم مدیریت فایل                    |
| عناصر پردازی   | ○ مجموعه ای از کنش ها و وظایف که رویکردی اثر بخش برای تغییر دادن فعالیت ها دارد. |
| عناصر ساختمانی | ○ مجموعه ای از ابزارها که ساخت خود کار نرم افزار را امکان پذیر می نماید.         |
| عناصر انسانی   | ○ مجموعه ای از ابزارها و ویژگی های پردازی که مهندس نرم افزار به کار می گیرد.     |

**۲.۹. خط مبنا (Baseline)**

- هنگامی که یک شی پیکر بندی توسعه یافت و مرور شد به خط مبنا تبدیل می شود.
- تغییراتی که در یک خط مبنا اعمال می شود، منجر به ایجاد نسخه جدیدی از آن شی می شود.
- خط مبنا یک نقطه عطف در توسعه نرم افزار است.
- وظیفه مهندس نرم افزار این است که یک یا چند SCI را تولید کرده و در بانک اطلاعاتی نگهداری کند.
- هنگامی که یک عضو از تیم یکی از SCI های خط مبنا را اصلاح کند آن را از بانک اطلاعاتی به فضای کاری خود اختصاص داده است.

**۲.۱۰. آیتم های پیکر بندی نرم افزار**

- آیتم های پیکر بندی نرم افزار را پیش از این به عنوان اطلاعاتی تعریف کردیم که به عنوان بخشی از فرآیند مهندسی نرم افزار ایجاد می شود.
- در حالتی واقع بینانه تر SCI یک سند، مجموعه کاملی از موارد آزمون یا یک قطعه برنامه است.
- بسیاری از سازمان های مهندسی نرم افزار، علاوه بر SCI های به دست آمده از محصولات کاری نرم افزار، ابزار های نرم افزاری تحت کنترل پیکر بندی را قرار می دهند.
- ابزارها در تولید مستندات، داده ها و کدهای منبع به کار می روند.
- از این رو می توانند به عنوان خط مبنا برای بخشی از یک فرآیند جامع پیکر بندی عمل کند.
- در واقع SCI ها برای تشکیل اشیای پیکربندی سازماندهی می شوند.

**۳. مخزن SCM**

### ۳.۱. مخزن SCM

در نخستین روزهای مهندسی نرم افزار آیتم های پیکربندی نرم افزار به صورت مستندات کاغذی در کابینت های فلزی نگهداری می شد که به ۴ دلیل مشکل آفرین بود.

- یافتن یک آیتم دشوار بود.
  - تعیین اینکه کدام آیتم چه هنگام و توسط چه کسی تغییر کرده، ایجاد چالش می کرد.
  - ایجاد نسخه جدید از یک برنامه وقت گیر و در معرض خطا بود.
  - توصیف روابط غیر ممکن بود ولی امروزه، SCI ها در بانک اطلاعاتی یا مخزن نگهداری می شوند.
- مخزن محلی برای انباشتن یا ذخیره سازی تعریف می شود. در روزهای اولیه از برنامه نویس استفاده می شد که برای به خاطر سپردن و بازسازی اطلاعات کار آمد نبود و بعد از یک چیز به نام مخزن که یک بانک اطلاعات است استفاده کردند.

### ۳.۲. نقش مخزن

- مخزن SCM، مجموعه ای از سازوکارها و ساختمان داده هاست که تیم نرم افزاری را به صورت اثر بخش مدیریت می کند.
- مخزن SCM به عنوان مرکزی برای انسجام بخشیدن، به ابزارهای نرم افزاری عمل می کند، که سبب یکنواختی و فرمت بندی محصولات نرم افزاری می شود.
- برای دستیابی به این قابلیت ها مخزن بر حسب یک شبه مدل عمل می کند، که چگونگی دسترسی، نگهداری و مشاهده و حفظ امنیت به همراه پاسخ گویی به نیازهای جدید را تعیین می کند.

### ۳.۳. محتوا و ویژگی های عمومی مخزن

- ویژگی و محتویات مخزن رو از دو دیدگاه می توان شناخت: آن چه که قرار است در مخزن انباشته شود و خدمات ویژه ای که توسط مخزن فراهم خواهد آمد.
- یک مخزن پر قدرت دو دسته خدمات ارائه می دهد شامل: انواع خدمات مورد انتظار از هر سیستم و خدمات مختص.
- مخزن که به تیم مهندسی نرم افزار خدمات می دهد هم چنین باید: قابلیت مدیریت منسجم یا مستقیماً پشتیبانی کند، قواعد ویژه حاکم بر مخزن را پشتیبانی کند، فراهم ساختن واسطی با سایر ابزار های مهندسی نرم افزار و ذخیره سازی اشیاء داده ای پیچیده

## ۴. ویژگی های SCM

### ۴.۱. ویژگی های SCM

- برای پشتیبانی SCM، مخزن باید ابزارهایی داشته باشد که ویژگی های زیر را پشتیبانی کند.
- ایجاد نسخه ها: به موازات پیش رفتن پروژه و وجود نسخه های فراوان، مخزن توانایی ذخیره نسخه ها را داشته باشد.
- مدیریت تغییرات و ردگیری وابستگی ها: مخزن گستره وسیعی از روابط در خود مدیریت می کند. این روابط میان فرآیندها، موجودیت ها و مولفه های طراحی و غیره است.

- که این روابط از دو نوع وابستگی و اجباری هستند. توانایی ردگیری روابط و ذخیره اطلاعات در مخزن و محصولات قابل تحویل، اهمیت حیاتی دارد.
- ردگیری خواسته ها: این وظیفه خاص به مدیریت پیوندها مربوط می شود و توانایی ردگیری مولفه های طراحی، ساخت و محصولات قابل تحویل را فراهم می سازد.
- مدیریت پیکر بندی: یک تسهیلات مدیریت پیکر بندی، اطلاعات پیکر بندی را نگهداری می کند.
- جلسات ممیزی: در جلسه ممیزی، اطلاعات اضافی در بازه زمان، علت و عامل انجام دهنده تغییرات فراهم می آید.
- اطلاعات مربوط به مخزن را می توان به عنوان صفت وارد مخزن کرد.

## ۵. فرایند SCM

فرایند SCM چهار هدف اصلی را دنبال می کند:

- شناسایی همه آیتم ها که پیکر بندی نرم افزار را تعریف می کند.
- مدیریت تغییرات به عمل آمده در آیتم ها
- تسهیل در ایجاد نسخه های متفاوتی از یک برنامه کاربردی
- حصول اطمینان از اینکه کیفیت نرم افزار با تکامل پیکر بندی به مرور زمان حفظ می شود.



## جلسه ۱۳ – معیارهای محصول

### ۱. نگاهی گذرا

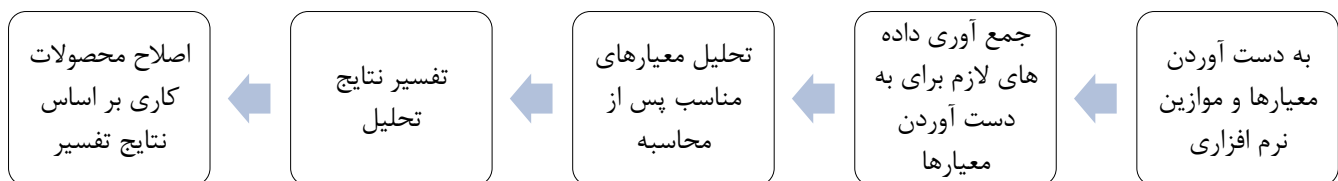
#### ۱.۱. معیارهای نرم افزار چیست؟

- مهندسی رشته ای با ماهیت کمی
- کمک گرفتن از اعداد برای ارزیابی محصول

#### ۱.۲. چرا اهمیت دارد؟

- کافی نبودن ارزیابی کیفی
- نیاز به ملاک های عینی در طراحی داده ها، معماری، واسطه ها و مولفه ها

#### ۱.۳. مراحل کار کدام است؟



### ۲. چارچوبی برای معیارهای تکنیکی محصول

#### ۲.۱. موازین، معیارها و شاخص ها

- **میزان:** کمیتی که نشان گر حد، مقدار، ابعاد، ظرفیت یا اندازه صفتی از محصول یا فرایند است. مثل: تعداد خطاهای کشف شده در مرور یک پیمانه
- **اندازه گیری:** عمل تعیین میزان
- **معیار:** میزانی کمی از حدی که یک سیستم، مولفه یا فرایند می تواند دارای یک صفت مفروض باشد. مثل: تعداد میانگین خطاهای یافت شده به ازای هر مرور
- **شاخص:** معیار یا ترکیبی از معیارهاست که درکی از فرایند نرم افزار یا محصول به دست می دهد.

#### ۲.۲. اصول اندازه گیری

- معیار باید خواص ریاضی مطلوب داشته باشد.
- هنگامی که یک معیار، خصوصیتی از نرم افزار را نمایش می دهد که با رخ دادن صفت مثبت، افزایش می یابد یا با مشاهده صفت نامطلوب، کاهش می یابد، ارزش معیار به همان شیوه افزایش یا کاهش می یابد.
- هر معیاری باید به صورت تجربی اعتبارسنجی گردد و بعدا منتشر شود.

### ۲.۳. سنجش هدف گرای نرم افزار

- الگوی هدف/پرسش/معیار (GQM) به عنوان تکنیکی برای شناسایی معیارهای بامعنی در بخشی از فرایند نرم افزار
- تعیین یک هدف اندازه گیری که ویژه خصوصیتی از محصول یا فعالیت فرایندی است که قرار است ارزیابی شود.
- تعریف مجموعه ای از پرسش ها که باید برای دستیابی به هدف به آن ها پاسخ داده شود.
- شناسایی معیارهایی با تدوین مناسب که به پاسخ گفتن به این پرسش ها کمک می کند.

### ۲.۴. صفات معیارهای نرم افزاری اثر بخش

|                                        |                             |                                            |                         |                                            |                            |
|----------------------------------------|-----------------------------|--------------------------------------------|-------------------------|--------------------------------------------|----------------------------|
| سازوکار موثری<br>برای بازخورد<br>کیفیت | استقلال زبان<br>برنامه سازی | سازگاری در<br>استفاده از واحدها<br>و ابعاد | سازگار و هدفمند<br>باشد | از نظر تجربی و<br>شهودی قانع<br>کننده باشد | ساده و قابل<br>محاسبه باشد |
|----------------------------------------|-----------------------------|--------------------------------------------|-------------------------|--------------------------------------------|----------------------------|

## ۳. معیارهایی برای مدل خواسته ها

### ۳.۱. معیارهای مبتنی بر عملکرد

- معیار امتیاز عملکرد (FP): ابزاری برای اندازه گیری عملکردهای ارائه شده توسط سیستم.

### ۳.۲. معیارهایی برای کیفیت مشخصات

|                          |               |             |               |                |           |
|--------------------------|---------------|-------------|---------------|----------------|-----------|
| سازگاری داخلی<br>و خارجی | قابلیت واریسی | قابلیت درک  | درست بودن     | کامل بودن      | مشخص بودن |
| قابلیت استفاده<br>مجدد   | دقت           | اصلاح پذیری | قابلیت ردگیری | قابلیت بایگانی |           |

## ۴. معیارهایی برای مدل طراحی

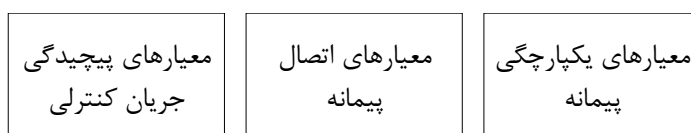
### ۴.۱. معیارهای طراحی معماری

- در معیارهای طراحی معماری، خصوصیات معماری برنامه و اثربخشی پیمانه ها بررسی می شود.
- این معیارها، از نوع جعبه سیاه هستند.
- پیچیدگی ساختاری: برای معماری های سلسله مراتبی (فراخوانی و بازگشت) شاخصی از پیچیدگی در توان خروجی پیمانه
- پیچیدگی داده ای: شاخصی از پیچیدگی در واسط داخلی پیمانه
- پیچیدگی سیستمی: مجموع پیچیدگی داده ای و ساختاری

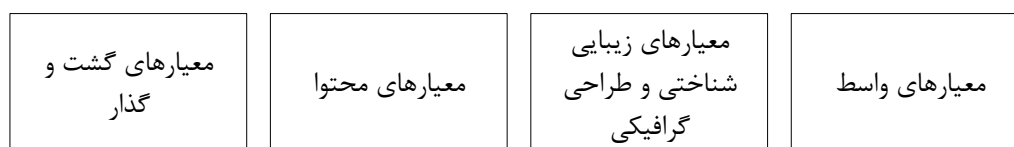
### ۴.۲. معیارهایی برای طراحی شی گرا



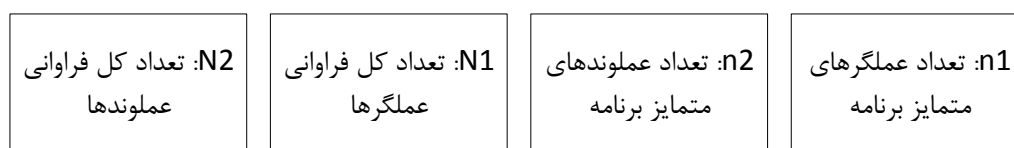
#### ۴,۳. معیارهای طراحی در سطح مولفه ها



#### ۴,۴. معیارهای طراحی برای برنامه های تحت وب



#### ۴,۵. معیارهایی برای سورس کد



#### ۴,۶. معیارهایی برای آزمون

- کاربرد معیارهای هالستد در آزمون: تعیین کننده مقدار کار لازم برای آزمون است.

#### ۴,۷. معیارهایی برای نگهداری

- شاخص بلوغ نرم افزار (SMI): بر اساس تغییراتی که برای هر بار ارائه نسخه جدیدی از محصول رخ می دهد نشان گر پایداری محصول نرم افزاری است.

