

به نام خدا

اراعه مهندسی نرم افزار ۲

مریم انگوتی گلنگدري

۳۹۸۱۲۳۱۰۰۹

آزمون نرم افزار

عناوین

(۱) آزمون نرم افزار چیست

(۲) test-case

(۳) دلایل آزمون نرم افزار

(۴) مراحل تست

(۵) چه کسانی تست نرم افزار را انجام میدهند

(۶) رویکرد های تست

آزمون نرم افزار چیست؟

آزمون نرم افزار ، مجموعه ای از فعالیت هاست که می تواند به طور پیشرفته، برنامه ریزی شده و به طور سیستمی هدایت شود. برای این منظور، باید برای فرآیند تست نرم افزار، یعنی مجموعه گام ها که بتوانیم **test-case** های خاص و فنون طراحی روش های تست را جایگزین کنیم، الگویی تعریف شود.



Test-case

یک Test Case جهت ممیزی یک امکان (Feature یا Functionality خاص از یک اپلیکیشن نرم‌افزاری، به صورت مجموعه‌ای از Action‌های قابل اجرا تعریف می‌شود. یک Test Case، جزئی ضروری تست نرم افزار است که باعث تأیید اعتبار می‌شود.

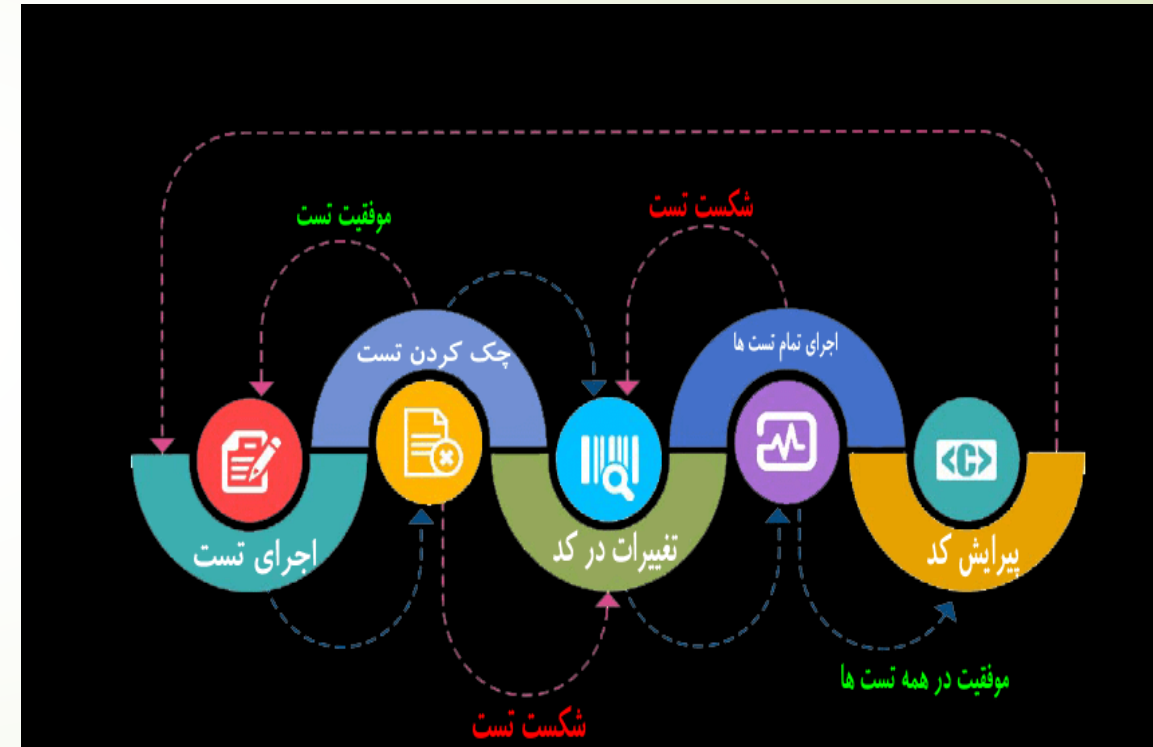
➤ Test Case 1: بررسی نتایج در حالت ورود User ID معتبر و Password معتبر

➤ Test Case 2: بررسی نتایج در حالت ورود User ID نامعتبر و Password نامعتبر

➤ Test Case 3: بررسی نتایج در حالتیکه User ID خالی باشد و دکمه لاگین فشرده شود

دلایل آزمون نرم افزار

وقتی کدی را می نویسیم، ممکن است به همه نکات آن توجه نکنیم؛ البته این موضوع با توجه به تجربه برنامه نویس متفاوت است؛ ولی هنگامی که شما برای کدتان تست می نویسید، مجبور می شوید به خیلی از جزئیاتی که در حالت معمول به آن توجه نمی کنید، توجه داشته باشید. به طور معمول، تست، بیشترین تلاش را نسبت به دیگر فعالیت های مهندسان، نیاز دارد.



اگر فرایند تست بدون دقت هدایت
شود، باعث هدر رفتن زمان، تلاش
غیر ضروری، بدتر شدن اوضاع و عدم
تشخیص خطاهای پنهان می شود.
بنابراین، برپا کردن یک راهبرد
سیستمی برای تست نرم افزار، معقول
به نظر می رسد.



مراحل تست

- تست واحد
- تست جامعیت
- Functional test/ Acceptance test
- Performance test
- System test



➤ تست واحد

در این نوع تست، قسمت های کوچک برنامه را تست می کنیم؛ مثلاً توابع یا کلاس ها در زبان های object oriented. در این تست، با کل برنامه کاری نداریم و هدف ما، اطمینان از کارکرد قسمت های کوچک برنامه است. تست واحد، در سطح کد است و یک ابزار مناسب برای آن JUnit است. این ابزار، تست خودکار انجام می دهد که در مرکز تحقیقات کامپیوتری علوم اسلامی (نور) زبان هایی مثل: C++، C# و Java وجود دارد که برنامه نویسان تست های واحد را در برنامه های خود می نویسند و در آینده، همه کدهای موجود در مرکز تست های واحد خواهند داشت.



object oriented: برنامه نویسی شی گرا

➤ تست جامعیت

این تست برای این است که مطمئن شویم کامپوننت های مختلف برنامه با هم درست کار می کنند.



Namnak.com

Namnak.com

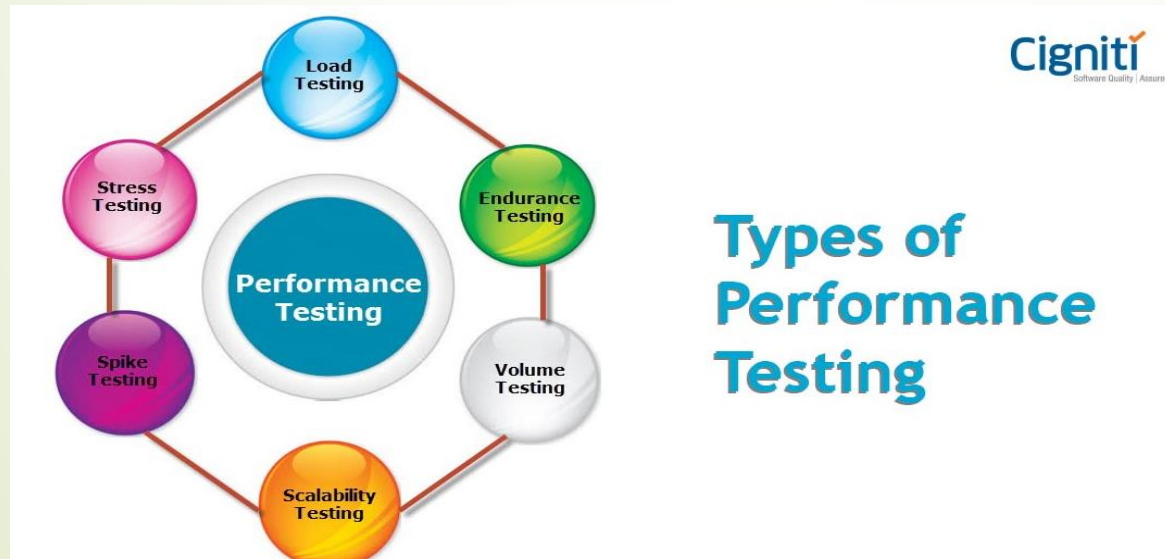


➤ Functional test/ Acceptance test

رای این است که مطمئن شویم آیا نرم افزار کامل هست و به درستی کار می کند؟ این نوع تست، یک تست سطح بالا محسوب می شود و ابزار این تست، **Fitness** می باشد

➤ Performance test

در این نوع تست، کارایی کامپوننت های برنامه مورد بررسی قرار می گیرد. تست کارایی در معاونت فنی نور توسط برنامه نویسان، و در معاونت پژوهش توسط قسمت بررسی نرم افزار انجام می شود.



چه کسانی تست نرم افزار را انجام می دهند؟

- تخصصان و یا ابزارهایی که روند تست نرم افزار را انجام می دهند

- مدیر پروژه: برای تأیید یک فاز پروژه و رفتن به فازهای بعدی، انجام می دهد. تست مدیر پروژه، بعد از بررسی های دقیق کارشناسان تست و تست برنامه نویسان است.

- مهندسان نرم افزار: مهندسان تست، برای هر قسمت از برنامه، کد تست می نویسند که از این نوع تست ها، به عنوان تست های واحد تعبیر می شود. یا هنگامی که چند واحد برنامه را کنار هم قرار می دهند، برای تست اینکه متوجه بشوند چند واحد یکپارچه شده در کنار همدیگر درست کار می کنند یا نه، بررسی هایی انجام می دهند و تست هایی تعریف می نمایند که به این نوع تست، تست یکپارچگی گفته می شود.

- متخصصان تست: افرادی هستند که در تمام مراحل تولید برنامه، روش هایی برای تست تعریف می کنند و سعی دارند این تست ها را انجام دهند؛

رویکردهای تست

➤ تست ایستا

➤ تست پویا

➤ تست جعبه سفید

➤ تست جعبه سیاه

➤ تست جعبه خاکستری

➤ **تست ایستا:** نوعی تست است که در آن، از نرم افزار استفاده نمی شود. در این تست، وارد جزییات نمی شویم و مثلاً در آن منطق برنامه، الگوریتم و داکيومنت های آن بررسی می شوند؛ در این نوع تست، کد را می خوانیم و به دنبال اشکال ها می گردیم که معمولاً خود توسعه دهنده برنامه، این کارها را انجام می دهد.

➤ **تست پویا:** رفتار نرم افزار را به ورودی هایی که به مرور زمان تغییر می کنند، نشان می دهد؛ یعنی تستی که در آن برنامه اجرا می شود و با ابزاری دیگر به آن ورودی های مختلف داده می شود و رفتار یا خروجی آن را می بیند و تحلیل می کند.

➤ **تست جعبه سفید:** در این تست، کارکرد ساختارهای داخلی برنامه بررسی می شود و اینکه منطقاً به کد احتیاج هست تا مثلاً متدهای مختلف بررسی شوند. فنی که در این نوع تست استفاده می شود، به این شرح است:

✓ تست API: در تست جعبه سفید، هر API را به صورت جدا گانه تست می کنیم.

✓ بررسی سطر به سطر کد: یعنی بررسی خطوط کد، به صورت خط به خط و جزء به جزء؛ به صورتی که خطوط کد و مسیرهای مستقل داخل یک پیمانه حداقل یک بار اجرا و تست شوند.

✓ متدهای Fault injection: طی این متد، یک کد خطا در کد تزریق می کنیم تا خطایی رخ دهد و بتوانیم موردی را تست کنیم.

✓ متدهای mutaion test: برای تست نرم افزار، خودمان یک خطا را در کد ایجاد می نماییم.

➤ **تست جعبه سیاه:** در این نوع تست، به اجزای داخلی نرم افزار کاری نداریم و به عنوان یک جعبه سیاه نگریسته، رفتار برنامه را با دادن ورودی بررسی می کنیم

➤ **تست جعبه خاکستری:** در این نوع تست، در مورد ساختار داخلی نرم افزار و الگوریتم آن اطلاعاتی داریم و از آن برای طراحی تست استفاده می کنیم؛ درحالی که در تست هایی که طراحی می کنیم، هیچ دسترسی به ساختار داخلی نرم افزار نداریم و مانند یک جعبه سیاه به آن می نگریم.



پایان