**استراتژی تست** جعبه سفید:

حال تصور کنید که شما خود یک توسعه دهنده نرم‌افزار هستید، پس شما می توانید به سورس، طراحی و منابع دیگر نرم‌افزار دسترسی داشته باشید. در این حالت سیستم را می توان به یک جعبه شیشه ای ([جعبه سفید](http://fa.wikipedia.org/w/index.php?title=%D8%AC%D8%B9%D8%A8%D9%87_%D8%B3%D9%81%DB%8C%D8%AF&action=edit&redlink=1&preload=%D8%A7%D9%84%DA%AF%D9%88:%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%D9%85%D9%82%D8%A7%D9%84%D9%87/%D8%A7%D8%B3%D8%AA%D8%AE%D9%88%D8%A7%D9%86%E2%80%8C%D8%A8%D9%86%D8%AF%DB%8C&editintro=%D8%A7%D9%84%DA%AF%D9%88:%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%D9%85%D9%82%D8%A7%D9%84%D9%87/%D8%A7%D8%AF%DB%8C%D8%AA%E2%80%8C%D9%86%D9%88%D8%AA%DB%8C%D8%B3&summary=%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%DB%8C%DA%A9+%D9%85%D9%82%D8%A7%D9%84%D9%87+%D9%86%D9%88+%D8%A7%D8%B2+%D8%B7%D8%B1%DB%8C%D9%82+%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF%DA%AF%D8%B1&nosummary=&prefix=&minor=&create=%D8%AF%D8%B1%D8%B3%D8%AA+%DA%A9%D8%B1%D8%AF%D9%86+%D9%85%D9%82%D8%A7%D9%84%D9%87+%D8%AC%D8%AF%DB%8C%D8%AF&withJS=MediaWiki:Intro-Welcome-NewUsers.js)) تشبیه کرد که شما می توانید براحتی محتویات داخل و نحوه عملکرد آنرا مشاهده کنید. آزمایش جعبه سفید نیز دقیقاً از دیدگاه توسعه دهنده نرم‌افزار را مورد آزمایش قرار می دهد یعنی با این فرض که شما به منطق داخلی و ساختار کد برنامه دسترسی و احاطه دارید و می دانید که سیستم چگونه پیاده سازی شده است. شما با دانستن این موارد می توانید مشخص کنید که آیا اعمال داخلی بر طبق مشخصه‌ها نجام می‌شود و یا نه.

در یک استراتژی آزمایش جعبه سفید ما عموماً موارد زیر را مورد توجه و بررسی قرار می دهیم:

۱. بررسی سطر به سطر کد (Code coverage): در این حالت باید سیستم را به گونه ای اجراء و بررسی کنیم که مطمئن شویم سطر به سطر کد برنامه حداقل یکبار اجراء شده است.

۲. بررسی همه انشعاب‌ها در کد برنامه (branch) : در کد برنامه باید تمام عبارت‌های شرطی ( if elseها و Switch case ها) را تک به تک مورد بررسی قرار داد. به این صورت که در یک عبارت if else هم فسمت if و هم قسمت else هر کدام بصورت مجزا یکبار اجراء شوند.

۳. بررسی همه حلقه‌ها در برنامه : حلقه‌ها در نرم‌افزار نقش اساسی دارند، چون می تونند با اشتباه جزئی مقدار زیادی از منابع را مصرف کرد برای مثال شرط خروج از حلقه به اشتباه هیچ وقت True نشود. برای نمونه حلقه‌ها را با ورودی بزرگتر از شرط خروج حلقه چک کنید یعنی حلقه اصلاً اجر نشود. تستی طراحی کنید که حلقه دقیقاً یکبار اجراء شود، تستی طراحی کنید که حلقه در یک بازه خاص اجراء شود و ....

۴. مدیریت خطای مطلوب : برسی اینکه اگر به یک متد یک ورودی نامعتبر وارد شود، نحوه آگاه سازی و نمایش مطلوب خطا برای کاربر چگونه باشد؟

5.بررسی امنیت : سیستم را از این جهت که چگونه در برابر دسترسی‌های غیرمجاز، هک، کرک و هر چیز دیگر که می تواند به آن آسیب برساند مورد بررسی قرار می دهد. در اینجا ما باید مکانهای از کد را که داده‌ها را اعتبارسنجی و مدیریت می کنند، دسترسی به منابع یا عملیات مهم و حیاتی را انجام می دهند را بررسی کنیم.

**استراتژی تست جعبه سیاه:**

جعبه سیاه از جمله استراتژی های تست است که به همراه تست جعبه سفید و جعبه طوسی در کتب و منابع مختلف مهندسی نرم افزار در موردش بحث شده است. در این روش تست کننده اطلاعاتی در زمینه منطق درونی و جزئیات مربوط به نرم افزار که شامل سورس کد و مستندات دیگر خواهد بود را در اختیار ندارد. بصورت ساده می توانیم بگوییم که وی تنها یک فایل اجرایی را در اختیار داشته و فقط به لایه خارجی نرم افزار دسترسی خواهد داشت، همان چیزی که از عنوانش واضح است، یعنی نمیتوان به محتویات داخل آن دسترسی داشت.

سوالی که در این راستا مطرح می گردد این است که با وجود در اختیار نداشتن سورس کد چگونه میتوان نرم افزار را تست نمود؟ پاسخ این سوال ورودی و خروجی هاست، بله در واقع تست کننده در اینجا نقش یک کاربر نهایی را بازی خواهد کرد، بدینصورت که میتواند داده های طراحی شده خود را وارد و نتایج مورد نظر را از خروجی کسب نماید. ما در چند پست پیش طی یک مثال عمومی به [بررسی الگوی یک تست ساده نرم افزار](http://softeng.blogsky.com/1390/03/02/post-10/) پرداختیم، در واقع می توانیم بگوییم آن مثال شیوه ای ساده از پیاده سازی تست جعبه سیاه بود. در مثال مذکور طی مراحلی به تهیه داده های تست پرداختیم و پس از آن منتظر خروجی ماندیم، خروجی ها در مراحل مختلف نتایج ما را تشکیل می دهند، نتایجی که میتوانیم آنها را با هم مقایسه کرده و سنجش نماییم.