آزمون نرم افزار

تمرین شماره 2 9 آبان 1402

محمدرضا بخشایش - 810199381

محمد امین باهنر – 810199548

سوال 1)

سوال 2)

سوال3)

درباره تفاوت این دو روش میتوان گفت:

|  |  |
| --- | --- |
| روش Classic | روش Mockist |
| * برای شبیه سازی DOC ها ترجیحا از Stub ها و نمونه های واقعی استفاده می کند. در صورت عدم امکان سراغ ساخت Mock ها و یا Stub های دوگان شده می رود | * حدالامکان از Mock های دوگان شده استفاده می کند و از DOC های واقعی استفاده نمیکند |
| * بیشتر از State Verification استفاده می کند | * بیشتر از Behavior Verification استفاده می کند |
| * امکان استفاده از رویکرد "وسط به بیرون" را فراهم می کند؛ یعنی در توسعه هر قابلیت، از کلاس های تعریف شده در مدل دامنه شروع کرده و توسعه را تا پایان لایه UI ادامه می دهیم. در محیط هایی که وفاداری به مدل دامنه مهم است، این روش مفید است | * امکان استفاده از رویکرد "بیرون به درون" را فراهم میکند؛ یعنی توسعه نرم افزار از بیرونی ترین لایه (مثلا UI) به کمک یک Mock از لایه داخلی تر برنامه شروع شده و به مرور زمان لایه های داخلی تر توسعه داده میشوند |
| * قبل از توسعه هر قابلیت، باید بخشی از تعریف DOC های مربوط به آن انجام شده باشد | * برای توسعه هر قابلیت، نیازی به پیاده شدن هیچ کدام از وابستگی ها نداریم |
| * برای اجرای تست، نیاز به ساخت نمونه از تعداد زیادی از اشیاء وابسته داریم | * برای اجرای تست، صرفا نیاز به ساخت یک Mock حداقلی از اشیاء وابسته داریم |
| * ممکن است خطا های موجود در هر ماژول، روی تست های ماژول های دیگر نیز اثر بگذارد | * نتیجه هر تست صرفا به عملکرد SUT بستگی دارد |
| * به دلیل استفاده از State Verification، تست تا حد خوبی از پیاده سازی مستقل می شود | * به دلیل استفاده از Behavior Verification، تست تا حدی به پیاده سازی وابسته می شود |
| * نیاز به اضافه کردن توابع Getter برای Verify کردن State اشیاء بعد تست داریم | * نیازی به تغییر در Interface کلاس ها برای پیاده سازی تست نداریم |
| * از آنجا که تست ها با کمک اشیاء واقعی انجام می شوند، بهتر میتوان به آنها اعتماد کرد | * برای برنامه های با ساختار لایه ای مناسب تر است. |

به عنوان جمع بندی میتوان گفت هیچ کدام از این دو روش لزوما به دیگری برتری ندارد و انتخاب از بین این دو بیشتر به سلیقه و عادات توسعه ای تیم بستگی دارد.