

مرحله تست اجزا (Unit Test)

اجزاء کوچکترین بلوک‌های کد یک نرم‌افزار هستند. تست اجزاء پروسه اعتبارسنجی این بلوک‌های کوچک از سیستم پیچیده کامل است که لازم است خیلی قبل‌تر از انجام هرگونه تست دیگر از زیرسیستم‌ها یا خود سیستم به صورت کامل انجام شود. تست کردن جزء روی تست واحدهایی از نرم‌افزار تمرکز می‌کند تا اطمینان حاصل شود که این اجزا همانگونه که انتظار می‌رود کار می‌کنند. به دلیل اینکه پروژه ما در مراحل ابتدایی فرایند تکمیل شدنش است، فعلاً انجام Unit Test مقدور می‌باشد. ما برای انجام این کار از ابزارهای خود Visual Studio و برای تست رابط‌های کاربری نیز از WatiN که یک open source framework برای انجام هرگونه تست در Visual Studio است و مخصوص برای محیط‌های وب طراحی شده است، استفاده می‌کنیم.

برخی از مهم‌ترین فواید تست جزء به صورت دوره‌ای به شرح زیر است:

- توانایی تست کردن برخی قسمت‌های پروژه بدون اینکه نیاز به تکمیل کامل پروژه باشد.
- توانایی پیدا کردن و رفع مشکلات نرم‌افزار از طریق چندین نفر به صورت موازی و هم زمان در تیم.
- توانایی استفاده از تعدادی تکنیک برای تست جز که شناسایی و رفع مشکلات احتمالی پروژه را بسیار سهل‌تر می‌کنند.
- سهولت دیباگینگ از طریق محدود سازی این پروسه به یک سری جزیهای کوچکتر
- توانایی تست برخی از شرایط دورن سیستمی (همانند شرایط خطا) که به راحتی از طریق ورودی‌های خارجی در سیستم‌های بزرگ به عنوان یک واحد قابل تشخیص نیستند.

۱. پروسه انجام Unit Test

یک جز قابل تست یک جز مستقل از کد است که می‌تواند جدای از سایر بخش‌ها مورد تست قرار بگیرد. تعریف یک جز بستگی به محیط برنامه نویسی دارد که ما در اینجا بسته به شرایط متدها و کلاس‌ها را مورد تست قرار می‌دهیم. در این مرحله باید کلاسی که مورد تست قرار می‌گیرد از لحاظ اینکه وابسته به کدامین لایه از ساختار برنامه است، مورد توجه قرار گیرد. سه نوع تست برای یک نرم‌افزار می‌توان متصور شد:

- Structural Testing
- Functional Testing
- Heuristic / intuitive Testing

۲. Structural Testing

تست ساختار که به تست جعبه سفید یا تست داخل سیستمی معروف است بر اساس ساختار کد صورت می‌پذیرد. برای انجام این تست یک سری مورد‌های تست ساخته می‌شود (یک سری از ورودی‌ها و خروجی‌های مورد انتظار نظیر آنها) که همه مسیرهای ممکن از تکه کدهای یک جز شامل if، while، switch و ... که در طول اجرا برخورد می‌شود را مورد ارزیابی قرار می‌دهد. مورد‌های تست گفته می‌شود که باید همه خطوط و شاخه‌های برنامه را تحت پوشش قرار دهد.

این تست کمک می‌کند تا مطمئن باشیم که همه بخش‌های کد در طول تست اجرا شده و نتایج مورد انتظار ما را برمی‌گردانند. مورد تست باید به اندازه کافی دارای مجموعه داده باشد تا اطمینان حاصل شود که :
همه متدها صدا زده شده‌اند.

همه شاخه‌های "درست" و "غلط" و if ها پوشش داده شده اند.

همه حلقه‌ها برای تعداد صفر، یک و تعداد بیشتر اجرا شده‌اند.

همانطور که گفته شد برای هر مجموعه ورودی باید یک مجموعه خروجی نظیر نیز مشخص شود. سپس برنامه با این داده‌ها اجرا می‌شود و خروجی‌های برنامه با خروجی‌های مورد انتظار مقایسه می‌شود تا از صحت برنامه اطمینان حاصل شود.

۳. Unit-level Functional Testing

تست کارایی سطح اجزا (که به عنوان تست جعبه سیاه یا تست خارجی نیز شناخته می‌شود) روی مسایلی تمرکز می‌کند که برنامه درحالت کلی برای آن جز مورد انتظار است تا انجام دهد. به این دلیل شخصی که این تست را انجام می‌دهد باید به خوبی بداند که ایده کلی این جز چیست و کلا چه ایده‌ای را پیش می‌برد. تست کننده یک مجموعه داده (ورودی و خروجی‌های مورد انتظار) را تهیه می‌کند. معمولاً باید شرایط مرزی و ورودی‌هایی که محتمل به بروز خطا هستند را در این مجموعه داده جای داد. تست ساختاری و تست کارایی به نوعی مکمل یکدیگر هستند.

۴. Heuristic / intuitive Testing

در تست Heuristic تست کننده به صورت جدا گانه یک برنامه را مرور می‌کند و تمام مشکلاتی که می‌توان پیدا کرد را اصلاح می‌کند. انجام این تست خیلی مورد توجه نیست و بهتر است در حین انجام دو بخش دیگر مورد توجه قرار گیرد.