

آزمون نرم افزار

خانم دکتر فرشته جدیدی میاندشتی

علیرضا سلطانی نشان

۱ مهر ۱۴۰۳

فهرست مطالب

۱	معرفی
۲	تعاریف اولیه
۲	۱.۲ نیازمندی های آزمون نرم افزار یا Test Requirements
۲	۲.۲ معیار آزمون نرم افزار یا Test Criterion
۲	۳.۲ مورد آزمایش یا Test Case
۲	۴.۲ نقاط مرزی آزمایش یا Test Boundry
۲	۵.۲ معیارها برای آزمون نرم افزار مبتنی بر ساختار
۳	۶.۲ منظور از Bypass یا کنار گذاشتن

مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLV۳ منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می دهد.

۱ معرفی

از روش هایی برای آزمون نرم افزار استفاده می کنیم که بالاترین ضریب اطمینان را به توسعه دهنده و صاحب محصول نرم افزاری می دهد. اگر برای آزمون نرم افزار در فرایند توسعه هزینه نکنیم (زمان و نیروی مناسب برای آن در نظر نگیریم)، در هنگام Production و استقرار نهایی مطمئناً چندین و چند برابرش را خواهیم پرداخت.

نکته

- آزمون نرم افزار کاملاً هزینه بر است.
- ضریب اطمینان هیچ چیز در امنیت به طور ۱۰۰ درصد تامین نمی شود.
- برای سیستم های مهم و بحرانی بایستی به طور کامل تمام استثناها را کنترل و مدیریت کرد (اشاره به تکنیک Exception handling).

- بودجه‌ای که برای آزمون نرم‌افزار استفاده می‌کنیم ممکن است کم باشد ولی باید در نظر داشت که کار عاقلانه‌ای است که بودجه پروژه را به صورت متعادل بین توسعه و آزمون فرایندها تقسیم کرد تا از تمام رخدادها از قبل جلوگیری کرد.
 - بیشتر اوقات نسخه آفا را کاربر آزمایش می‌کند و پیش‌پا افتاده‌ترین مشکلات و باگ‌ها را کاربران عادی متوجه می‌شوند و ممکن است به مدیر سیستم و نگهداری آن را گزارش دهند.
 - برای آزمون نرم‌افزار 2^n راه را می‌توانیم بررسی کنیم اما هزینه و زمان زیادی را می‌تواند در بر گیرد.
- زمان زیاد منجر به افزایش هزینه‌ها می‌شود.

در این درس با روش‌های معمول آشنا می‌شویم که بالاترین ضریب اطمینان را برای آزمون نرم‌افزار مورد نظر خود بدست آوریم. در حقیقت می‌توان گفت با انواع روش‌های آزمون نرم‌افزار آشنا می‌شویم و سپس آزمونی را انتخاب می‌کنیم که بیشترین پوشش نرم‌افزاری یا Coverage را دارا می‌باشد.

۲ تعاریف اولیه

۱.۲ نیازمندی‌های آزمون نرم‌افزار یا Test Requirements

دقیقاً چه بخش‌هایی را می‌خواهیم مورد آزمون قرار دهیم؟ چه نیازی می‌خواهیم برآورده شود؟ بطور کلی منظور موارد خاصی است که بایستی در طول آزمایش رعایت شوند یا پوشش داده شوند. برای مثال قبل از هر گونه آزمایش بایستی Best practice های آن بخش پوشش داده شود و سپس به دنبال آزمایش آن‌ها برویم تا از نظر استاندارد پیاده‌سازی مناسبی داشته باشیم.

برای مثال یک مگا اپلیکیشن به دلیل کمبود زمان در طول توسعه، امکانش نیست که همه دستورات و شاخه‌هایش آزموده شوند. پس سعی می‌کنیم قسمت‌های تصمیم‌گیری و بخش‌های حیاتی و مهم نرم‌افزار را به طور کامل آزمایش کنیم تا مشکلی از بابت در دسترس نبود یا به خطر افتادن نرم‌افزار را تهدید نکند. مثلاً فقط حلقه‌ها آزموده شود یا فقط شروط بررسی شود که چه خروجی دارند. چون بررسی کردیم که احتمال خطا در آن‌ها زیادتر است.

۲.۲ معیار آزمون نرم‌افزار یا Test Criterion

یک مجموعه‌ای از قوانین و فرایندهایی که نیازمندی‌های آزمون نرم‌افزار را مشخص می‌کنند. به عبارتی دیگر راه و عملیاتی که می‌خواهیم برای آزمون نرم‌افزار پیش بگیریم بایستی براساس معیاری باشد.

۳.۲ مورد آزمایش یا Test Case

چه چیزی را می‌خواهیم مورد آزمون قرار دهیم؟ چه ورودیهایی را برای آن انتخاب می‌کنیم؟ چه انتظاری از خروجی‌ها داریم؟ (اشاره به Assertions).

۴.۲ نقاط مرزی آزمایش یا Test Boundry

تمام نقاط مرزی و رعایت استثناهای برنامه را بررسی می‌کنیم. برای مثال برنامه‌ای را توسعه داده‌ایم که مجموع اعداد را حساب کند و تقسیم بر n کند. این برنامه دقیقاً چه زمانی به زمین می‌خورد؟ چه شرطی وجود دارد که برنامه با ورودی نامناسب به شکست دچار شود؟

۵.۲ معیارها برای آزمون نرم‌افزار مبتنی بر ساختار

۱. استفاده از گراف: برای منبع برنامه و طراحی usecase ها

۲. استفاده از عبارات منطقی: (Not x or not y) and A and B

۳. استفاده از ویژگی‌هایی که در دامنه ورودی وجود دارد:

A: 0, 1, > 1 (آ)

B: 600, 700, 800 (ب)

C: swe, cs, isa, infs (ج)

۴. Syntactic structures: $\text{if } (x > y): z = x - y \text{ else: } z = 2 * x$

گاهی اوقات قبل از هرگونه آزمایش نرم‌افزاری از روش نوشتن عبارات منطقی برای آزمایش سناریوهای نرم‌افزاری استفاده می‌شود.

۶.۲ منظور از Bypass یا کنار گذاشتن

گاهی در هنگام آزمون بخشی از نرم‌افزار نیاز است که از بررسی برخی قسمت‌ها عملیات Bypass را در پیش گیریم تا سرعت فرایند آزمایش نرم‌افزار افزایش پیدا کند.

