



آزمون نرم افزار

تدوین و گردآوری:

دکتر سید محمد حسین هاشمی نژاد

عضو هیئت علمی دانشگاه الزهراء (س)

مهندس مرضیه سادات اسماعیلی

کارشناس ارشد مهندسی کامپیوتر

سرشناسه: هاشمی نژاد، سیدمحمدحسین، ۱۳۶۴-
عنوان و نام پدیدآور: آزمون نرم افزار / تدوین و گردآوری سیدمحمدحسین هاشمی نژاد، مرضیه سادات اسماعیلی.
مشخصات نشر: تهران: دانشگاه الزهرا (س)، انتشارات، ۱۳۹۹.
مشخصات ظاهری: ۲۸۰ ص. شابک: 978-622-6114-69-1
وضعیت فهرست نویسی: فیبا.
یادداشت: کتابنامه.
شناسه افزوده: اسماعیلی، مرضیه سادات، ۱۳۷۴-
(س). انتشارات شناسه افزوده: Alzahra University press
رده بندی کنگره: QAV6/76 رده بندی دیویی: ۰۰۵/۱۴ شماره کتابشناسی ملی: ۷۵۴۷۶۹۹



آزمون نرم افزار

تدوین و گردآوری:

دکتر سیدمحمدحسین هاشمی نژاد عضو هیئت علمی دانشگاه الزهرا (س)
مهندس مرضیه سادات اسماعیلی کارشناس ارشد مهندسی کامپیوتر

ناشر: دانشگاه الزهراء (س)

طراح جلد: محمد احمدی

چاپ و صحافی: مرکز چاپ و نشر دانشگاه پیام نور
شمارگان: ۳۰۰ نسخه
قیمت: ۹۰۰۰۰۰ ریال

شابک: ۹۷۸-۶۲۲-۶۱۱۴-۶۹-۱

مرکز پخش:

ونک، دانشگاه الزهرا (س) / تلفن: ۸۵۶۹۲۸۳۹-۴۸۹۳۴-۸۸۰

وبگاه فروش اینترنتی: <https://book.alzahra.ac.ir>

فهرست:

| | |
|----|---|
| ۶ | سخن مولفان |
| ۹ | پیشگفتار |
| ۱۱ | مقدمه |
| ۱۳ | چه کسی آزمون می‌کند؟ |
| ۱۳ | چه زمانی آزمون شروع می‌شود؟ |
| ۱۴ | چه زمانی آزمون پایان می‌یابد؟ |
| ۱۵ | فصل اول - تعریف آزمون نرم افزار |
| ۱۷ | ۱.۱. مقدمه |
| ۱۹ | ۲.۱. مفاهیم شکست، خطا، نقص و اشکال |
| ۲۴ | ۳.۱. رابطه درستی |
| ۲۷ | ۴.۱. نقص مشروط و قطعی |
| ۳۷ | ۵.۱. مدیریت نقص |
| ۳۹ | ۶.۱. مدیریت اشکال |
| ۴۱ | ۷.۱. تعیین اهمیت اشکالها |
| ۴۵ | ۸.۱. نتیجه گیری |
| ۴۷ | فصل ۲ - اهداف و مدل بلوغ آزمون پذیری |
| ۴۹ | ۱.۲. مقدمه |
| ۴۹ | ۲.۲. صنعت نرم افزار - مدل بلوغ قابلیت |
| ۵۲ | ۳.۲. مدل بلوغ آزمون پذیری |
| ۵۸ | ۴.۲. اهداف و سیاست‌های آزمون |
| ۶۳ | ۵.۲. طرح آزمون |
| ۷۳ | ۶.۲. نتیجه گیری |
| ۷۵ | فصل ۳ - چرخه عمر آزمون و صفت کیفی آزمون پذیری |
| ۷۷ | ۱.۳. مقدمه |

۴ / آزمون نرم افزار

| | |
|-----|---|
| ۷۷ | ۲.۳. چرخه عمر پروژه |
| ۸۵ | ۳.۳. تفاوت تایید و اعتبارسنجی |
| ۸۶ | ۴.۳. چرخه عمر آزمون نرم افزار |
| ۹۲ | ۵.۳. مدل V و مدل W برای آزمون نرم افزار |
| ۹۴ | ۶.۳. نظم آزمون در دیدگاه متدولوژی RUP |
| ۹۸ | ۷.۳. صفت کیفی آزمون پذیری نرم افزار |
| ۱۰۴ | ۸.۳. نتیجه گیری |
| ۱۰۵ | فصل ۴ - سطوح آزمون |
| ۱۰۷ | ۱.۴. مقدمه |
| ۱۰۹ | ۲.۴. طراحی پیشگو |
| ۱۱۶ | ۳.۴. آزمون واحد |
| ۱۲۳ | ۴.۴. آزمون یکپارچگی |
| ۱۲۹ | ۵.۴. آزمون سیستم |
| ۱۳۱ | ۶.۴. آزمون پذیرش |
| ۱۴۶ | ۷.۴. آزمون های برگشت و جهش |
| ۱۵۳ | ۸.۴. نتیجه گیری |
| ۱۵۵ | فصل ۵ - آزمون های رنگی |
| ۱۵۷ | ۱.۵. مقدمه |
| ۱۵۸ | ۲.۵. آزمون جعبه سفید |
| ۲۰۱ | ۳.۵. آزمون ایستا و آزمون پویا در رویکرد جعبه سفید |
| ۲۰۴ | ۴.۵. آزمون جعبه سیاه |
| ۲۱۴ | ۵.۵. آزمون جعبه خاکستری |
| ۲۱۷ | ۶.۵. نتیجه گیری |
| ۲۱۹ | فصل ۶ - آزمون های غیرکارکردی |
| ۲۲۱ | ۱.۶. مقدمه |

| | |
|-----|--|
| ۲۲۲ | ۲.۶. معماری نرم افزار و نیازمندی های غیر کارکردی |
| ۲۲۴ | ۳.۶. نیازمندی کارایی و روش های آزمون کارایی |
| ۲۲۹ | ۴.۶. نیازمندی امنیت و روش های آزمون امنیت |
| ۲۳۴ | ۵.۶. نیازمندی کاربردی بودن و روش های آزمون کاربردی بودن |
| ۲۳۸ | ۶.۶. آزمون تطبیق کد و طراحی |
| ۲۴۳ | ۷.۶. نتیجه گیری |
| ۲۴۵ | پیوست اول - فهرست موارد آزمون عمومی برای سامانه های تحت وب |
| ۲۶۹ | لغت نامه |
| ۲۷۸ | منابع و مراجع |

سخن مؤلفان

آزمون نرم افزار یکی از مهم ترین مراحل چرخه تولید نرم افزار و وظیفه آن اطمینان از صحت و کامل بودن کدهای پیاده سازی شده است. همان طور که از نام این فعالیت برداشت می شود، این حوزه یکی از ارکان کیفیت نرم افزار محسوب می شود. در واقع، کیفیت نرم افزار دارای دو وجه کنترل و تضمین است که در آزمون نرم افزار نیز این دو وجه دیده می شود. برنامه ریزی که برای اجرای آزمون انجام می شود و تدارکاتی که می بایست در این حوزه آماده شود، مرتبط با دیدگاه تضمین کیفیت و اجرای آزمون، ارائه نتایج آن و پیگیری اشکالها، مرتبط با دیدگاه کنترل کیفیت است.

به طور مختصر هدف آزمون، اثبات بدون خطا بودن یک سامانه نیست، بلکه هدف آن یافتن اشکالهای محتمل سامانه است. در واقع با انجام آزمون نرم افزار نمی توان قسم خورد که سامانه دیگر اشکالی ندارد، زیرا در حیطه آزمون نرم افزار بررسی تمامی حالات، منجر به انفجار حالت می شود و عملاً نیاز به هزینه بسیار زیاد - حتی چندین برابر زمان صرف شده برای تولید آن سامانه - دارد. حال زمانی که به این حقیقت می رسیم، بیشتر سازمانها و شرکت های تولید نرم افزار، به دلیل هزینه های زیاد آزمون و اینکه هر چه هم هزینه کنند باز هم اشکالی ممکن است وجود داشته باشد، تصمیم می گیرند که این فعالیت را از چرخه تولید نرم افزار حذف کنند و آن را به مشتریان نهایی بسپارند.

زمانی که مرحله آزمون داخلی نرم افزار در شرکت تولید نرم افزاری انجام نگردد، این امر سبب می شود که رفت و برگشتها، برای رفع اشکال مابین مشتری نهایی و تیم پیاده ساز زیاد شود و ضمن افزایش نمایی هزینه ها - به دلیل دوباره کاریها - در این بین مشتری نهایی خسته و ناامید می شود و عملاً برند آن شرکت زیر سؤال می رود که ساده ترین موضوعات مثل ورود به سامانه نیز دارای اشکال است.

تجربه نشان می دهد، به دلیل اینکه مشتریان نهایی با سامانه های کاربردی آشنایی کمی دارند، حتی در صورتی که آزمون نرم افزار به ایشان سپرده شود و انجام آن را بپذیرند، بسیاری

از اشکال و خرابی‌ها کشف نمی‌شوند و زمانی که سامانه زیر بار برود، خود را بروز می‌دهند؛ بنابراین وجود مرحله‌ی آزمون داخلی نرم‌افزار چه از لحاظ ارتقای کیفیت و چه از لحاظ کل هزینه تمام‌شده نرم‌افزار، الزامی است، اما حال چه باید کرد که این مرحله بهینه انجام شود؟

جواب این سؤال، فلسفه وجودی این کتاب است که می‌بایست افراد آزمونگری تربیت شوند تا با آشنایی کامل به مفاهیم آزمون نرم‌افزار و روش‌های آزمون نرم‌افزار - چه خودکار و چه دستی - آزمون‌هایی را طراحی کنند که بتوانند با به‌کارگیری بیست درصد هزینه‌ی آزمون، هشتاد درصد اشکال‌های یک سامانه را کشف کنند. در آن صورت می‌توان به شرکت‌های تولید نرم‌افزار اطمینان داد که با صرف ۱۰ تا ۱۵ درصد هزینه‌ی کل پروژه، جهت اجرای آزمون نرم‌افزار توسط خبرگان این حوزه می‌توانند نرم‌افزار با کیفیت‌تری را ارائه نمایند و هزینه‌های رفت و برگشت مابین مشتریان نهایی و تیم پیاده‌ساز را به حداقل برسانند.

در این کتاب تلاش شده تا اولاً مفاهیم آزمون نرم‌افزار تبیین گردد و ثانیاً خواننده را با روش‌های مختلف آزمون نرم‌افزار و ارائه‌ی نتایج آن آشنا سازد. پیش‌نیاز خواندن این کتاب، آشنایی با مبانی برنامه‌نویسی کامپیوتر است و انتظار می‌رود که خواننده با مطالعه آن بتواند به‌عنوان فردی متخصص آزمون در یک شرکت تولید نرم‌افزاری مشغول به کار شود.

درواقع، ریشه‌ی شکست بسیاری از پروژه‌های تولید نرم‌افزار در کشورمان ایران، بحث کیفیت آن‌ها است که اگر در حوزه‌ی آزمون نرم‌افزار متخصصان متعدد و کارآزموده‌ای وجود داشته باشند، می‌توانند به ارتقای کیفیت نرم‌افزارهای تولید داخل کمک کنند، لذا امیدواریم که دانشجویان و فارغ‌التحصیلان رشته‌های مهندسی کامپیوتر و فناوری اطلاعات (و سایر رشته‌های علاقه‌مند به کامپیوتر) با مطالعه این کتاب و کسب تجربه و کار با ابزارهای دستی و خودکار آزمون نرم‌افزار بتوانند این مهم را محقق نمایند و سامانه‌هایی که در شرکت‌های نرم‌افزاری تولید می‌شوند، دارای کیفیت بیشتر و قابلیت رقابت با نرم‌افزارهای خارجی باشند.

در حوزه آزمون نرم افزار و جایگاه آن در چرخه تولید نرم افزار، هنوز مسائل باز متعددی وجود دارد که مؤلف این کتاب در خصوص مباحث مربوط به انتخاب الگوهای طراحی و مؤلفه های معماری نرم افزار با توجه به نیازمندی های غیرکارکردی مختلف، ارزیابی کمی معماری نرم افزار، آزمون تطبیق کد با طراحی انجام شده و اولویت بندی موارد آزمون با رویکرد الگوریتم های تکاملی، نوآوری هایی انجام داده که در بخش های مختلف کتاب به آن ها اشاره شده است.

نابروه باک، کیفیت میسر نمی شود مزد آن گرفت جان برادر که آزمون به کد کند

دکتر سید محمدحسین هاشمی نژاد

مهندس مرضیه سادات اسماعیلی

پیشگفتار

اهمیت درستی اجرای یک سیستم و برنامه نرم‌افزاری بر کسی پوشیده نیست. همه ما کم‌وبیش تجربه ناخوشایند مشاهده یک اشکال در زمان اجرای نرم‌افزار را داریم. خطای مشاهده شده چه کوچک و در حد یک اشکال گرافیکی خفیف باشد و چه یک خطای بزرگ که موجب برهم خوردن نتیجه وظیفه نرم‌افزار باشد؛ اگرچه از نظر اهمیت بسیار متفاوت و در سطوح مختلفی محسوب می‌شود اما از حس ناخوشایند کاربر و بی‌اعتمادی او نسبت به نرم‌افزار، برنامه‌نویس، شرکت و حوزه کاری مرتبط نمی‌کاهد. در ادامه، به اختصار موضوع هر فصل از کتاب را معرفی می‌کنیم.

در مقدمه کتاب، به بیان کلی حیطه آزمون نرم‌افزار پرداخته و مجری آزمون و زمان آغاز و پایان آن را معرفی می‌کنیم. فصل اول به تعریف آزمون نرم‌افزار، مفاهیم خطا، شکست، نقص و اشکال و بررسی آن‌ها می‌پردازد. در ادامه با مدیریت اشکال و دسته‌بندی‌ها، اولویت و اهمیت آنان آشنا می‌شوید.

موضوع فصل دو، اهداف و مدل بلوغ آزمون‌پذیری است. در این فصل این مدل به تفصیل شرح داده شده و تمامی سطوح معرفی می‌شود، سپس سیاست‌های آزمون و قوانین و مقررات حاکم بر آن در سطح برنامه‌نویس، شرکت و در سطوح کلان معرفی می‌شود. طرح و سند آزمون، معرفی‌کننده روال و اصول اجرای آزمون یک نرم‌افزار، موضوع بحث بعدی این فصل است.

چرخه عمر آزمون و صفت کیفی آزمون‌پذیری در فصل سوم بررسی می‌شود. ویژگی‌های ساختار عمومی چرخه نرم‌افزار، چرخه‌های عمر افزایشی و تکراری و انطباق‌پذیری، تأیید و اعتبارسنجی، مدل V و مدل W از مباحث این فصل است. نظم آزمون در متدولوژی **RUP** و آزمون‌پذیری نرم‌افزار در انتهای این فصل بیان شده است.

سطوح آزمون شامل آزمون واحد، آزمون یکپارچگی، آزمون سیستم و آزمون پذیرش کارفرما هستند که توضیح، تفسیر و ویژگی های این سطوح در فصل چهار بیان شده است. در ضمن معیارهای گسترده آزمون پذیرش نیز در این فصل مورد بررسی قرار می گیرد. آزمون های رنگی، نوعی دسته بندی انواع آزمون های نرم افزار است. در این دسته بندی با توجه به میزان اطلاعات و دسترسی آزمونگر به قسمت های مختلف نرم افزار و همچنین رویکرد استفاده شده برای آزمون، آزمون ها به سه دسته آزمون سفید، آزمون سیاه و آزمون خاکستری تقسیم می شوند که مبانی و اطلاعات این آزمون ها، در فصل پنجم بیان شده است.

انواع آزمون، تعاریف و ویژگی های هر کدام به اختصار، موضوع فصل ششم است و مبحث پراهمیت و مفصل معیارهای آزمون نرم افزار در فصل هفتم بررسی می شود. بدون شک مباحث این کتاب شامل تمامی مباحث موجود در حیطه آزمون نرم افزار نیست ولی تلاش شده است که نکات حائز اهمیت در این حیطه به شکل ساده و مختصر در اختیار خواننده قرار گیرد.

مقدمه

همواره یکی از مهم‌ترین بحث‌ها در طراحی و پیاده‌سازی نرم‌افزارها، کارکرد درست سیستم طراحی شده است. با رشد سیستم و توسعه چندین ساله آن توسط تیم‌های نرم‌افزار، کارکرد درست و بدون اشکال همه قسمت‌های آن همواره یکی از دغدغه‌های برنامه‌نویسان بوده است. به همین دلیل آزمون‌های مختلفی تعریف و طراحی شده‌اند که به تیم‌های نرم‌افزاری کمک می‌کنند، نرم‌افزارهای کم‌اشکال یا بی‌اشکالی را تولید نمایند.

شرکت‌های نرم‌افزاری، همگی محصولی به نام «نرم‌افزار» تولید می‌کنند، ولی چه عامل یا عواملی باعث می‌شوند که یک نرم‌افزار از نرم‌افزار مشابه دیگر متمایز و برجسته شود؟ عوامل متعددی را می‌توان نام برد که باعث این برتری و تمایز می‌شوند. یکی از این عوامل می‌تواند کیفیت محصول نهایی باشد که به بازار عرضه خواهد شد؛ بنابراین این سؤال مطرح می‌شود که برای رسیدن به این نقطه تمایز و برتری باید چگونه عمل کرد و اندیشید؟ یک پاسخ به این سؤال می‌تواند آزمون نرم‌افزار و نحوه انجام آن باشد.

حتی آزمون را می‌توان یکی از زیرمجموعه‌های مبحث کیفیت نرم‌افزار با نام «تضمین کیفیت» در نظر گرفت. در واقع، آزمون نرم‌افزار به دنبال خطایابی و عیب‌یابی محصول، قبل از تحویل به مشتری است؛ آزمون نرم‌افزار را می‌توان به صورت‌های زیر معنا کرد:

❖ کوشش در جهت عیب‌یابی و رفع آن، نه تلاش در جهت اثبات کامل صحت نرم‌افزار، زیرا این قضیه با ماهیت آزمون تفاوت دارد.

❖ آزمون نرم‌افزار یا آزمایش نرم‌افزار، تحقیق بر روی کیفیت یک محصول یا سرویس نرم‌افزاری و ارائه اطلاعات ناشی از آن به مشتری‌هایش است. این تحقیق، جستجوی نرم‌افزار را به منظور یافتن خطاها نیز در برمی‌گیرد ولی به آن محدود نمی‌شود.

❖ آزمون نرم‌افزار مجموعه‌ای از سؤال و جواب‌هایی است که نرم‌افزار را با آن امتحان می‌کنیم درحالی‌که از برنامه انتظار داریم با توجه به ورودی‌هایی که با استفاده از سؤالات وارد می‌کنیم، پاسخ‌های صحیحی را به‌عنوان خروجی ارائه دهد.

- ❖ آزمون نرم افزار حیطة وسیعی از فعالیت‌های مربوط به تولید برنامه‌های کامپیوتری از آزمون کد برنامه توسط برنامه‌نویس گرفته تا نشان دادن عملکرد درست یک سیستم اطلاعاتی بزرگ به مشتری را در برمی‌گیرد.
 - ❖ آزمون نرم افزار شامل فرآیند اجراهای متعدد برنامه با هدف یافتن اشکال‌های نرم‌افزاری است اما محدود به آن نیست. تضمین اینکه تا چه حد نیازمندی‌های موجود را برآورده می‌کند و آیا با انتظارات مشتری سازگار است یا خیر نیز توسط فرآیند آزمون مشخص می‌شود.
 - ❖ آزمون نرم افزار به دفعات تکرار خواهد شد تا مشکلات هر چه بیشتر نمایان شوند. استاندارد آزمون نرم افزار فرآیندهایی را در جهت آشکارسازی و برطرف کردن عیوب سیستم دنبال خواهد کرد.
 - ❖ آزمون نرم افزار از وظایف توسعه‌دهندگان نرم افزار است و نه از وظایف تحویل گیرندگان آن.
 - ❖ دایجکسترا^۱ در تعریف آزمون نرم افزار می‌گوید [۱]: آزمون نرم افزار فقط می‌تواند وجود خطاها را نشان دهد، نه عدم وجود آنها را!
 - ❖ آزمون نرم افزار به طور رسمی جزئی از بازمینی و اعتبارسنجی نرم افزار است که این دو واژه در ادامه تعریف و باهم مقایسه می‌شوند.
 - واری: آیا محصول را به درستی می‌سازیم؟
 - اعتبارسنجی: آیا محصول درستی را می‌سازیم؟
- واری، بررسی می‌کند که آیا نرم افزار از مشخصاتش پیروی می‌کند یا خیر. اعتبارسنجی، باید تضمین کند که نرم افزار انتظارات مشتری را برآورده می‌سازد یا خیر. توجه کنید که آنچه در مشخصات می‌آید ممکن است دقیقاً خواسته‌های مشتری را برآورده نسازد. در واقع هدف فرآیند آزمون در یک جمله «**ایجاد اعتماد**

¹ Edsger W. Dijkstra

نسبت به سیستم است؛ اما چطور می‌توان نسبت به سیستم ایجاد اعتماد نمود؟ عوامل مختلفی در این امر دخیل هستند اما نقش آزمون پررنگ‌تر است. به همین دلیل آزمون دو هدف اصلی را دنبال می‌کند:

۱- آزمون اعتبارسنجی: موفقیت آن در درست کار کردن سیستم است.

۲- آزمون نقص: موفقیت آن در نمایان‌سازی خطاهایی است که موجب کارکرد نادرست سیستم می‌شود.

چه کسی آزمون می‌کند؟

در صنعت فناوری اطلاعات، شرکت‌های بزرگ، تیمی را برای بررسی نرم‌افزارهای تولیدی بر اساس چارچوب الزامات و فعالیت‌های شرکت در نظر دارند که این تیم، تیم آزمون است. در اغلب موارد آزمونگر در یکی از بخش‌های زیر قرار می‌گیرد [۲]:

➤ بخش آزمون نرم‌افزار

➤ بخش توسعه نرم‌افزار

شرکت‌ها، طراحی‌های متفاوت، وظایف و نقش‌های متفاوتی برای اشخاصی که آزمون نرم‌افزار را انجام می‌دهند، قائل می‌شوند. این نقش‌ها شامل آزمونگر نرم‌افزار، تضمین کیفیت نرم‌افزار^۱، تحلیل‌گر و غیره هستند.

چه زمانی آزمون شروع می‌شود؟

در طول چرخه عمر توسعه نرم‌افزار^۲ که به اختصار SDLC گفته می‌شود [۳]، فرآیند آزمون از فاز تحلیل آغاز شده و تا استقرار نرم‌افزار به طول می‌انجامد. با این حال تمامی این آزمون‌ها بستگی به مدل توسعه‌ای دارد که شرکت‌ها انجام می‌دهند. به‌طور مثال در مدل آبشاری، آزمون در پایان مرحله پیاده‌سازی نرم‌افزار انجام می‌شود؛ اما در مدل افزایشی، آزمون در پایان هر افزایش یا تغییر تکرار می‌شود و در پایان پیاده‌سازی نرم‌افزار هم دوباره

^۱ QA (Quality Assurance)

^۲ Software Development Life Cycle

آزمون انجام می شود. در هر مرحله از SDLC ، تجزیه و تحلیل و تأییدیه های مورد نیاز نیز برای آزمون در نظر گرفته می شود. بررسی طراحی در مرحله طراحی محصول نیز به قصد بهبود طراحی در حوزه آزمون در نظر گرفته می شود. انجام آزمون توسط یک توسعه دهنده پس از اتمام کد نیز به عنوان آزمون واحد^۱ طبقه بندی می شود. انجام آزمون در طول SDLC شامل مزایای ذیل است:

- ۱ - کاهش زمان تولید
- ۲ - کاهش هزینه ها
- ۳ - کاهش زمان دوباره کاری ها
- ۴ - کاهش خطاهای نرم افزاری
- ۵ - افزایش بازدهی
- ۶ - افزایش کیفیت نرم افزار
- ۷ - تحویل به موقع پروژه به کارفرما
- ۸ - افزایش رضایت مشتری

چه زمانی آزمون پایان می یابد؟

برخلاف آن که می دانیم چه زمانی آزمون را آغاز کنیم، تعیین زمان پایان آزمون بسیار دشوار است [۲] و آزمون فرآیندی بی پایان و تعیین زمانی برای توقف آن بسیار دشوار است و نمی توان با اطمینان گفت که نرم افزار تولیدی صد درصد آزمون شده است. مواردی که نمی توان به صورت دقیق برای آزمون در نظر گرفت شامل تعیین مهلت و اطمینان از کامل بودن آزمون است.

^۱ Unit Test