



TISTEn

The Iranian Software Test eNginers

✓ آخرین اخبار از دنیای تست نرم افزار

بزرگ رهبران تست در دنیا

✓ ISTQB و دیگر هیچ!

✓ مدیریت آزمون...

مدیریت نیازمندی...

✓ فجایع تاریخ نرم افزار

✓ تست نفوذ

✓ چرا Manual Test

✓ تست واژه

✓ تکنیک‌های کشف Error



به نام یگانه عالمِ عالمِ آفرین

STEVE JOBS

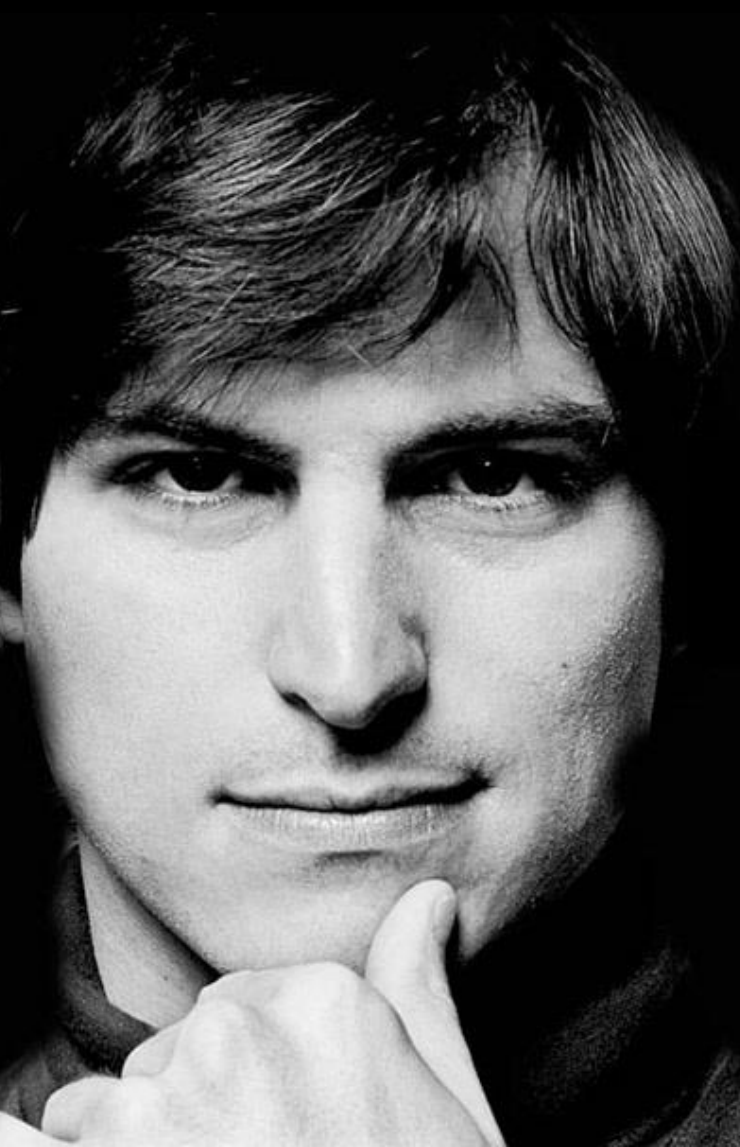
1955-2011

مردی که خدماتش را، با ارائه محصولات
باکیفیت، به دنیا معرفی کرد

کیفیت مهمتر از کمیت
است.

یک روستای آباد بهتر از
صد شهر خراب است.

Businessweek-2006



آخرین شماره نویسنده را از اینجا بخوانید



۳۱

به قلم ملیک وارطانیان



۱۶

به قلم هدی رضوی



۲

به قلم تحقیق و خیر



۳۳

به قلم هدی رضوی



۱۹

به قلم تحقیق و خیر



۱۱

به قلم ابوالفضل خواجه دیزجی



۳۵

به قلم ابوالفضل خواجه دیزجی



۳۱

به قلم بهروز منصوری



۱۲

به قلم ابوالفضل خواجه دیزجی



آخرین اخبار

از دنیای

تست نرم افزار

رهبر سرویس‌های ابری Conduit Global در راه قضاوت برای جایزه Software Testing اروپا

Conduit Global اعلام کرد که الیور اسمیت (Oliver Smith)، مدیر تضمین کیفی شرکت، برای دومین بار در جایزه Software Testing اروپا (European Software Testing Awards) به عنوان داور انتخاب شده است. این دوره در ۱۶ نوامبر ۲۰۱۶ برگزار خواهد شد. داوران این رویداد معتبر بر اساس تجربه گسترده خود در زمینه تضمین کیفیت، منصوب خواهند شد. این افراد الزاماً باید دارای تجربه طولانی در پست مدیریت ارشد باشند. انتخاب الیور اسمیت بر تعالی Conduit Global در رهبری حوزه تضمین کیفیت و تست، تأکید دارد.



الیور اسمیت بیانیه زیر را در راستای انتصاب خود در هیات داوران صادر کرد: "جایزه Software Testing در اروپا، ویرتینی پرمنزلت از ارزش فوق العاده تخصص تست را به صنعت و فناوری، افزود. این افتخاری بزرگ است که دگربار در جمع داوران برای این دوره حاضر باشم." اکنون در سال چهارم، جایزه Software Testing اروپا، تجلیلیتی از موفقیت شرکت‌ها و افراد در زمینه کیفیت. این فستیوال با حضور متخصصان صنعتی از سراسر جهان برگزار شده، و در این میان برخی از بهترین تست‌های نرم‌افزار در این فستیوال ارائه می‌شوند، که به عنوان نمادی از تعالی محسوب می‌شوند. علاوه بر

این شایان ذکر است که در طول این دوره، الیور اسمیت یافته‌های اخیر خود را در مورد روند تضمین کیفیت مربوط به صنعت تست نرم‌افزار ارائه خواهد کرد.

اهمیت Shift Left Testing

در رویکرد Shift Left، آزمون به سمت چپ چرخه توسعه نرم‌افزار حرکت کرده است. به جای اینکه آزمون در پایان چرخه تحویل قرار بگیرد، در طول مراحل اولیه انجام می‌شود. این موضوع به توسعه‌دهندگان اجازه می‌دهد تا به جای اینکه منتظر کشف نواقص در اواخر چرخه

حیات توسعه باشند، از همان ابتدا روی کیفیت به درستی تمرکز کنند. در توسعه نرم‌افزار به شیوه سنتی، نیازمندی‌های پروژه در سمت چپ برنامه (Plan) نگهداری می‌شد، و نیازمندی‌های تست و تحویل در سمت راست برنامه، که به معنی پایان پروژه است. عملکردهای توسعه سنتی نرم‌افزار قادر به اداره کردن انتظارات در حال تغییر نبودند، که این موضوع منجر به مشکلات کسب و کاری مانند افزایش هزینه، و نیز برآورده نشدن استانداردها در کیفیت و افزایش زمان رسیدن محصول به بازار می‌شد. مدل آزمون Shift Left یکی از روش‌های ممکن جهت کاهش این مشکلات است.

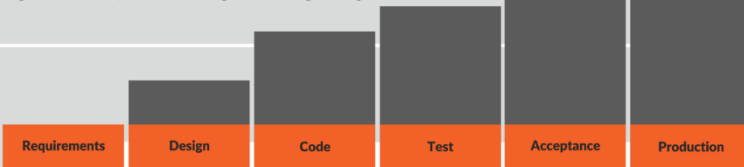
تأکید Shift Left Testing نه تنها روی انتقال تست به ابتدای فرآیند یا استفاده از اتوماسیون یا خرید یک ابزار ویژه برای پروژه است، بلکه بر ترکیب متدولوژی‌های مورد نیاز، ابزارها، Frameworkها و رویکردهای مناسبی که روی پیشگویی، پیشگیری، و شناسایی نواقص از ابتدای پروژه تا انتهای آن تأکید دارند، نیز بافشاری می‌کند. بدین ترتیب تست در سمت راست فاز (یعنی بعد فاز Requirement آغاز اجرا می‌شود. همین روال کامل چابکی و افزایش بهره‌وری، برای تست، ارائه بازخورد، و بازیابی روزهانه مهیا می‌کند. اکثر اوقات، کسب و کارها با تأخیر بسیاری در فازهای طراحی و توسعه‌ی ایلکیشن مواجه می‌شوند، که منجر به رسیدن به نقطه فشار حداکثری برای رساندن محصول در تاریخ مقرر در برنامه اولیه می‌شود. برای توضیح بهتر، یک مشتری کم‌تجربه می‌تواند منجر به تخریب شهرتی شود، که طی

سال‌ها به دست آمده است. با پذیرش امتیازات ویوکرد Left Approach به منظور تست، سازمان‌هایی مانند تجارت الکترونیک (e-Commerce) قادر به حفظ مزایای رقابتی خود با کاهش اتلاف درآمد خواهند بود. یک مطالعه در این زمینه نشان داد، که به نسبت فاز کدنویسی، اکثر نواقص در طول فاز Requirement (نیازمندی‌های) پروژه ایجاد می‌شوند. در مدل‌های سنتی، تست هنگامیکه توسعه در فاز کدنویسی قرار دارد، آغاز می‌گردد. این بدین معنیست که تست در یک فاز اشتباه از توسعه، که ممکن است در آن بسیاری از نواقص گم شوند، هدف‌گذاری شده است. مدل Shift Left به کیفیت و افزایش قدرت شناسایی و تصحیح نواقص به محض رخداد آنها توجه دارد.

تست در چرخه حیات توسعه نرم‌افزار باید تا سر حد امکان سریع آغاز گردد. از این رو، می‌توان نواقص را بسیار سریعتر شناسایی کرده و بدین ترتیب هزینه‌ها را پایین نگاه داشت. شناسایی هر چه سریعتر نواقص، به معنی کاهش ساعات زیادی از دوباره‌کاری‌های غیرضروریست.

کاهش دوباره‌کاری، زمان مورد نیاز برای تسترها جهت قرار گرفتن در زمانبندی، به منظور تحلیل دیگر نیازمندی‌ها و تمرکز بر تکنیک‌های تست مبتنی بر ریسک، را فراهم می‌آورد. این مدل، اولویت همه چیز را جهت استحصال کیفیت با دوباره‌کاری کمتر، شیف‌ت داده، و به پروژه‌ها اجازه می‌دهد تا بر مبنای زمانبندی و بودجه خود به پایان برسند. با تست فعالانه و زودهنگام و البته مرتب، تیم‌های توسعه قادر به ایزوله‌سازی به موقع نواقص جهت تصفیه سریعتر خواهند بود. این موضوع در نهایت منجر به بهبود چشمگیر زمان تحویل محصول به بازار خواهد شد.

Testing from requirement gathering stage onwards



Verticalization اپلیکیشن‌های خاص، به سمت یک پایگاه متمرکز کاربری، همچنان ادامه داشته باشد. رشد مشتری محوری در IT، کمپانی‌ها را به توسعه اپلیکیشن‌های همه‌فن‌حریف جهت تقویت حضور شرکت‌ها و تعامل با مشتریان، تشویق می‌کند.

یک کپی از این گزارش را می‌توانید از لینک زیر به دست آورید:

<http://www.marketresearchreports.biz/sample/sample/494421>

تقسیم‌بندی بوسیله End Userها و تحلیلی از بازار سرویس‌های Software Testing:

- BFSI (Banking, Financial services and Insurance) - بانکداری، سرویس‌های مالی و بیمه
- Financial Services (سرویس‌های مالی)
- Media (رسانه)
- Telecommunications (ارتباطات)
- دیگر موارد

در سال ۲۰۱۵ سرویس‌های مالی بیش از ۳۴٪ از سهم بازار را در اختیار داشتند. تقریباً کل روال توسعه و نگهداشت اپلیکیشن‌های مالی نیازمند درجه بالایی از سرویس‌های تست در سراسر چرخه حیات توسعه نرم-افزار است. رقابت شدید میان رهبران جهانی و حرکت به سمت سرویس‌های ویژه، موسسات مالی را برای گشتن به دنبال سرویس‌های تست با درجه Enterprise سوق می‌دهد.

چشم‌انداز رقابتی و فروشندگان کلیدی

حضور تعداد زیادی از ارائه دهندگان، طی این سال‌ها بازار سرویس‌های تست نرم‌افزار را به سمت رشد فوق-العاده، رانده است. کمپانی‌ها به منظور بهبود بسته‌های نرم‌افزاری خود، با بازیگران اصلی در این عرصه مشارکت کرده و به صورت مداوم نوآوری‌ها را افزایش می‌دهند. بازار آمریکای شمالی بوسیله IBM، Accenture و Cognizant قبضه شده است، در حالیکه TCS، Wipro، Infosys و قدرت خود را در EMEA (Europe, the Middle East and Africa) و APAC (Asia and Pacific) حفظ کرده است.

برترین فروشندگان در این حوزه عبارتند از:

- Accenture
- Capgemini

بازار جهانی سرویس‌های Software Testing به شدت برای سال‌های ۲۰۱۶-۲۰۲۰ رشد کرده و رونق خواهد گرفت

تحلیگر تحقیقاتی Technavios پیش‌بینی کرده است که بازار جهانی سرویس‌های Software Testing نزدیک به ۱۱٪ رشد را در "افزایش نرخ رشد متوسط Compound Annual Growth Rate (CAGR)" در طول دوره پیش‌بینی، خواهد داشت. یکی از پیش‌ران‌های کلیدی برای رشد بازار، رشد محبوبیت تست به صورت جمع‌سپاری (Crowdsourced) است. جمع‌سپاری یک تلاش جمعی از افراد مشغول به کار در نقاط مختلف روی یک پروژه آنلاین است. تست جمع‌سپاری شده محبوبیت زیادی طی این سال‌ها بدست آورده است، زیرا تنها فرآیند تستیست که بازخوردی بلادرنگ روی توسعه محصول ارائه داده، و پیشی درباره کیفیت محصول می‌دهد. علاوه بر این به نسترها برای تشخیص باگ‌ها، فراتر از چرخه تست نرم‌افزار، کمک می‌کند. سرویس‌های تست ویژه صنایع مختلف، روند و تمایل خاص دیگری را ارائه می‌دهند که پیش‌بینی می‌شود رشد بازار سرویس‌های تست نرم‌افزار را در طول دوره پیش‌بینی شده، به جلو سوق دهد. پیشرفت‌های اخیر، کمپانی‌ها را به سوی همتراز کردن سرویس-های تست در تخصص‌های عمودی (Vertical Specialization)، که در این سال‌های محبوبیت زیادی کسب کرده است، می‌راند. به عنوان نمونه، Infosys، یکی از بازیگران کلیدی در بازار جهانی توسعه نرم‌افزار، سرویس‌های تست را از طریق عملکردهای تخصصی گوناگون، مانند تست و ممیزی Healthcare، Retail (خرده‌فروشی) ارائه می‌دهد.

تحلیل و تقسیم‌بندی محصولات بازار سرویس‌های تست نرم‌افزار به صورت زیر است:

- سرویس‌های تست اپلیکیشن
- سرویس‌های تست محصول

در حال حاضر سرویس‌های تست اپلیکیشن روی بازار سرویس‌های تست نرم‌افزار با سهمی بیش از ۸۸٪ تسلط دارد. پیش‌بینی می‌شود رشد این قسمت، تا بیش از پنج سال آینده به توجه به افزایش

تویتر ظرف دو سال مبلغ \$۳۲۲,۴۲۰ به شکارچیان Bug پرداخت کرده است

سایت میکروبولگ نویسی تویتر ظرف دو سال مبلغ \$۳۲۲,۴۲۰ به محققین و شکارچیان باگ، تحت عنوان "جایزه در قبال کشف باگ" (Bug Bounty Program) پرداخت کرده است. آرکیدی تیلن (Arkady Tetelman) یکی از مهندسان برجسته تویتر این موضوع را در بلاگ اسپات منعکس کرد که: "ما یک چرخه حیات توسعه امن را توسعه داده و نگهداری می‌کنیم، که شامل: آموزش توسعه امن برای تمام کدنویسان، افرادی که فرآیند بازرسی امن را انجام می‌دهند، و نیز افرادی که بوسیله سرویس‌های داخلی و خارجی، کتابخانه‌های امنیتی را آزموده و تست استحکام (Robust test) انجام می-دهند، می‌باشد. تمام این کارها، امنیتی که ما برای مشتری فراهم می‌کنیم را به حد اکثر می-رساند." در بالاترین نقطه این اقدامات، تویتر درگیر بزرگتر کردن جامعه امنیت اطلاعات خود با استفاده از Bug Bounty Program می‌باشد. این موضوع به محققین امنیتی اجازه می‌دهد تا به شکلی مسئولانه اقدام به آشکارسازی آسیب‌پذیری‌های سیستم تویتر نمایند. بدین ترتیب کمپانی، طی این برنامه قادر خواهد بود، قبل از اینکه چنین مواردی، مورد سواستفاده دیگران قرار بگیرد، آنها را مشخص و مرتفع سازد.

تلمن اضافه می‌کند، کمپانی با استفاده از شرکت "HackerOne" از می ۲۰۱۴ برنامه‌ای را فراهم کرد تا با استفاده از آن، از منجی ارزشمند به منظور یافتن و مرتفع‌سازی آسیب‌های امنیتی استفاده نماید. وی در مطالب خود می‌افزاید: طی این دو سال کمپانی ۵,۱۷۱ مورد را از ۱,۶۶۲ محقق، در این زمینه دریافت کرده است، که بر اساس آن ۲۰٪ از باگ‌های رفع شده به این روش مشخص شده‌اند.

و اما در آخر وی اذعان می‌دارد که: "ما مجموعاً ۳۲۲,۴۲۰ دلار به این محققین پرداخت کرده‌ایم. میانگین پرداختی ما برابر با ۸۳۵ دلار بوده است. حداقل مبلغ پرداختی ما ۱۴۰ دلار و بیشینه پرداختی ما برابر با ۱۲,۰۴۰ دلار بوده است (مبلغ پرداختی همواره مضربی از ۱۴۰ است)."

تویتر ظرف دو سال مبلغ \$۳۲۲,۴۲۰ به محققین پرداخت کرده است. آرکیدی تیلن (Arkady Tetelman) یکی از مهندسان برجسته تویتر این موضوع را در بلاگ اسپات منعکس کرد که: "ما یک چرخه حیات توسعه امن را توسعه داده و نگهداری می‌کنیم، که شامل: آموزش توسعه امن برای تمام کدنویسان، افرادی که فرآیند بازرسی امن را انجام می‌دهند، و نیز افرادی که بوسیله سرویس‌های داخلی و خارجی، کتابخانه‌های امنیتی را آزموده و تست استحکام (Robust test) انجام می-دهند، می‌باشد. تمام این کارها، امنیتی که ما برای مشتری فراهم می‌کنیم را به حد اکثر می-رساند." در بالاترین نقطه این اقدامات، تویتر درگیر بزرگتر کردن جامعه امنیت اطلاعات خود با استفاده از Bug Bounty Program می‌باشد. این موضوع به محققین امنیتی اجازه می‌دهد تا به شکلی مسئولانه اقدام به آشکارسازی آسیب‌پذیری‌های سیستم تویتر نمایند. بدین ترتیب کمپانی، طی این برنامه قادر خواهد بود، قبل از اینکه چنین مواردی، مورد سواستفاده دیگران قرار بگیرد، آنها را مشخص و مرتفع سازد.



2016

برای اطلاعات بیشتر به سایت این شرکت مراجعه کنید:

www.testplant.com

QATestLab کمپانی پیشرو و مستقل در

تست نرم‌افزار، تبدیل به یک اسپانسر

عمومی برای تحقیقات "CEE QA

"Outsourcing Review 2015-16

شده است

کمپانی مستقل تست نرم‌افزار QATestLab تمام روندهای مدرن سرویس‌های QA Outsourcing را با تحقیقات جدید خود با عنوان "CEE QA Outsourcing Review 2015-16" آشکار کرد. این پژوهش به بررسی وضعیت فعلی کیفیت نرم‌افزار و عملکردهای (Practice) تست در سراسر کشورهای منطقه CEE/Central and Eastern Europe-اروپای مرکزی و شرقی، پرداخته است.

این تحقیق توسط انجمن اوکراینی HI-TECH Initiative و با حمایت انجمن Outsourcing سرویس مجارستان (Hungarian Service and HOA) (Outsourcing Association)، انجمن کارفرمایان صنعت نرم‌افزار و سرویس (Association of the Software and Services Industry (ANIS)) از رومانی، انجمن وب بلغارستان (Bulgarian Web Association) (BWA) و انجمن شرکت‌های سرویس‌های فرآیند کسب‌وکار و IT در لهستان (Association of IT & Business Process Services Companies in Hi-Tech (Poland) (ASPIRE)) و (Park) بلاروس، انجام شد.

"CEE QA Outsourcing Review 2015-16" کاتالوگ آنلاین تخصصی کامل از ارائه‌دهندگان سرویس‌های QA Outsourcing در اروپای شرقی و مرکزی راه‌اندازی کرده است - [Qalist.eu](http://qalist.eu) (<http://qalist.eu>). همه شرکت‌کنندگان در این تحقیق پروفایل‌های منتشرشده خود را در کاتالوگ ارائه دادند. در حال حاضر اطلاعات درباره ارائه‌دهندگان سرویس‌های QA Outsourcing از منطقه CEE به صورت دائم برای مشتریان بالقوه در دسترس است.

درباره QATestLab

QATestLab (<http://qatestlab.com/>) یک کمپانی پیشروی بین‌المللیست که طیف گسترده‌ای از سرویس‌های تست نرم‌افزار شامل مشاوره کیفیت نرم‌افزار، Test Planning، اجرای تست و بهبود فرآیند توسعه نرم‌افزار را ارائه می‌دهد. این کمپانی در سال ۲۰۰۵ تأسیس شد و تعداد زیادی محصول باکیفیت و مقرون به صرفه در بسیاری از صنایع مختلف ارائه کرده

UI و نیز یک محیط برای تست دستگاه‌های موبایلی، رشد خوبی داشته است". علاوه بر این گزارش به این نکته اشاره می‌کند که یکی از نقاط قوت TestPlant، توانمندی آن در پشتیبانی از Cross-Platform Testing در یک اسکریپت از طریق آگاهی از پلتفرم مورد نظر است".

علاوه بر این در این گزارش ذکر شده است که: "TestPlant تمرکز خود را روی Image-Based Recognition قرار داده است، که آترا از دیگر فروشندگان متمایز می‌نماید،" و اینکه "قابلیت‌های تست Performance و شبکه آن به سازمان‌های Development، چشم‌اندازی وسیع در رابطه با کیفیت، قابلیت اطمینان، و مقیاس‌پذیری زیرساخت Server مربوطه ایلکیشن‌هایشان ارائه می‌دهد."

آنتونی ادواردز (Antony Edwards) - معاون فنی TestPlant می‌گوید: "امروزه کسب و کار دیجیتال بصورت فزاینده‌ای روی بهبود کیفیت نرم‌افزار تمرکز کرده است، آنچنانکه تجربه کاربری، برای موفقیت و بقا مشتری، مهمتر شده است. ما خوشحالیم که در TestPlant به خاطر رویکرد منحصر به فرد، و مبتنی بر Image-Recognition، برای تست ایلکیشن‌ها و دستگاه‌های موبایلی، به رسمیت شناخته شده است. رهبران بین‌المللی مانند Wal-Mart، Bloomberg، Citi برای گارانتی کیفیت ایلکیشن‌های خود و بهبود بهره‌وری بخش‌های توسعه‌ای خود از eggplant استفاده می‌کنند."

محمول eggplant از TestPlant، سوییچی از ابزارهای سهل‌الاستفاده اتوماسیون تست نرم‌افزار را که به کمپانی‌ها در راستای تحویل سریع ایلکیشن‌های موبایل و Enterprise باکیفیت، در محیط‌های Agile و DevOps کمک می‌کند، ارائه داده است. eggplant رویکرد مبتنی بر تصویر را برای Functional Testing به نام خود ثبت کرده است، این رویکرد به مشتری‌های این کمپانی اجازه می‌دهد تا ایلکیشن‌های خود را از منظر کاربران واقعی تست کنند. محصولات Network Emulation و تست Performance قدرتمند، نیز به توسعه‌دهندگان اجازه می‌دهد تا به سادگی ایلکیشن‌هایی که تحت طیف وسیعی از شرایط هستند را تست کنند.

- IBM
- Wipro

دیگر فروشندگان در این بازار عبارتند از: Atos, Deloitte, CSC, Cognizant, Cigniti Gallop, HP, Hexaware, Solutions NTT Infosys, QualTest Group, Steria, SQS, Tech Mahindra و JST Global.

سوالات کلیدی که در این گزارش بدانها پاسخ داده می‌شود عبارتند از:

- چه اندازه‌ای از بازار و چه نرخ رشدی را در سال ۲۰۲۰ پاید متصور بود؟
- چه فاکتورهای کلیدی، بازار جهانی سرویس‌های تست نرم‌افزار را به پیش می‌رانند؟
- چه تمایلات کلیدی از بازار، رشد بازار جهانی سرویس‌های تست نرم‌افزار را متاثر می‌کند؟
- چالش‌های پیش روی رشد بازار چیست؟
- چه کسانی فروشندگان کلیدی در این بازار هستند؟
- فرصت‌ها و تهدیدهای بازار که فروشندگان حاضر در بازار جهانی سرویس‌های تست نرم‌افزار با آن مواجه می‌شوند کدام موارد هستند؟
- عوامل موثر بر سهم بازار آمریکا، APAC و EMEA
- نتایج اصلی تحلیل ۵ نیروی بازار جهانی سرویس‌های تست نرم‌افزار چیست؟

Forrester: TestPlant یک "بازیگر

قوی" در ابزارهای

End Test Automation

Image-Based

TestPlant روی

Recognition که آترا

از دیگر فروشندگان

تمایز می‌کند،

تمرکز کرده است.

TestPlant سازنده

eggplant Range از

ابزارهای کیفیت نرم‌افزار است،

و به تازگی در Forrester

Wave™ evaluation از آن به

عنوان بازیگری قوی در عرصه ابزارهای Mobile

Front-End Test Automation یاد شده است.

TestPlant از میان کمپانی‌های انتخاب شده، که

Forrester از آنها جهت شرکت در ارزیابی خود

دعوت به عمل آورده بود، برگزیده شده و منجر بدین

شد که Forrester درباره آن چنین بنویسد:

"TestPlant" به عنوان یک ابزار تست مبتنی بر تصویر



Crash های آتزا با استفاده از ایجاد لینکی بین این دو، به یکدیگر نزدیک کند". در همین راستای می افزاید: "زمانیکه یک Crash بروز می کند، TestFairy به صورت اتوماتیک ویدئوی ضبط شده از User Session و Stack Trace را به Bitbucket Repository لینک می کند، اکثر مشتریان ما که در حال حاضر راهکار را تست کرده اند، به ما می گویند بهره‌وری فعالیت آنها بسیار افزایش یافته است."

TestFairy یک پلتفرم بهبود و Mobile Testing است که به کمپانی‌ها توسط میاسازی نمایش ویدئویی که دقیقاً در خلال App Testing رخ داده و ضبط شده است، در راستای بهبود Beta Testing کمک می کند. این موضوع به شرکت‌ها اجازه می دهد تا قبل از اینکه یک App در بازار منتشر شده یا در Store مربوطه به روز گردد، Beat Testing با استفاده از ده ها هزار نفر از پرسنل کمپانی، یا نبوهی از تسترها، انجام شود. این پلتفرم توسعه دهندگان را با ویدئو و نیز Logها، گزارشات Crash، آمارهای مربوط به CPU، حافظه، باتری، Phone Reception، و Network Trace های دقیق که چگونگی ارتباط Client Server را با تجربه ویزوال کاربر مرتبط می کند، مجهز می نماید.

درباره TestFairy

TestFairy یک پلتفرم پیشرو در زمینه Mobile Beta Testing است، که راهکاری برای توسعه دهندگان App های اندروید و iOS ارائه داده است. این کمپانی در ۲۰۱۳ تاسیس شد و تا کنون بیش از یک میلیون دلار سرمایه جذب کرده است. برای اطلاعات بیشتر به آدرس زیر مراجعه کنید:

<http://www.testfairy.com>

خود را از محیط Bitbucket Pipelines شرکت Atlassian آپلود کنند. در صورت رخداد Crash، ماژول TestFairy Insight به صورت اتوماتیک جزئیات Crash را از نرم افزار JIRA و HipChat، با یک لینک مستقیم به Bitbucket ارسال می کند، تا بدین ترتیب دقیقاً Line Of Code ای که منجر به Crash شده است را مشخص نماید. این موضوع توسعه دهندگان را قادر می سازد تا بلافاصله درباره کشمکش های کاربر خود آگاه شده و همچنین بتوانند به سرعت به جزئیات Crash دست یابند. علاوه بر این، ماژول TestFairy Insight، قادر به زمینه یابی Mobile App Crash برای توسعه دهندگان موبایل، تسترها و مدیران محصول خواهد بود. در این مورد، یک ماژول ضبط ویدئویی از فعالیت های کاربر که منجر به Crash می شود، مهیا شده است.

برایانت لی (Bryant Lee) مسئول تیم های نرم افزاری و Ecosystem در Atlassian می گوید: "Atlassian از یکپارچه سازی صورت گرفته میان TestFairy با Bitbucket Pipelines بسیار هیجان زده و خوشحال است". علاوه بر این وی می افزاید: "روش TestFairy با آن Mobile App Built ها را با Atlassian Pipelines توزیع کرده و گزارشات Crash را به Bitbucket نگاهت می کند (Map)، دارای پتانسیل زیادی برای ساده تر کردن زندگی کاری توسعه دهندگان است."

گیل میگیدیش (Gil Megidish) به عنوان معاون فنی TestFairy می گوید: "بر اساس درخواست هایی که ما از سطح عظیمی از جامعه توسعه دهندگان و تسترهای خود دریافت می کردیم، ترجیح ما بر این بود که راهکاری را منتشر کنیم که Source Code و

است، از این دست صنایع می توان به این موارد اشاره کرد: صنایع تجارت الکترونیک، ارتباطات، توریسم، دولتی، تجهیزات اداری و غیره.

آژانس تحلیلی CLUTCH شرکت QA TestLab را در سال ۲۰۱۵ در زمره بهترین شرکت های Software Application Testing و آن هم به عنوان یک ارائه دهنده پیشرو در سرویس های QA Outsourcing قرار داد.

درباره CEEEO

انجمن Outsourcing اروپای مرکزی و شرقی (Central and Eastern European Outsourcing Association (<http://ceeo.org/>)) در سال ۲۰۰۸ بنیان نهاده شد. اعضای این انجمن، عبارتست از انجمن های پیشروی اروپای شرقی و مرکزی در Outsourcing و IT. یکی از اهداف اصلی CEEEO مترقی کردن منطقه اروپای شرقی و مرکزی به عنوان یک جایگزین رقابتی در مقصد برونسپاری (Offshoring) دیگر شرکت ها از سراسر جهان و نیز افزایش کیفیت و حجم سرویس های ارائه شده می باشد.

TestFairy سازگاری خود را با نرم افزار

Atlassian Jira، HipChat، Bitbucket، Bamboo و Atlassian Bitbucket Pipelines، اعلام کرد.

TestFairy یک پلتفرم پیشرو در بهبود و Mobile Testing، به تازگی سازگاری خود با محصولات شرکت Atlassian را اعلام کرده است.

راهکار جامع Mobile Beta Testing از TestFairy، به توسعه دهندگان موبایل اجازه می دهد تا به صورت اتوماتیک و به صورت مستقیم App های

Atlassian

TestFairy

در برگزیده آرای سازمان پژوهشی Gartner بوده و نباید به عنوان حجت تلقی شوند. گارتر تمام وارانته‌ها، شامل وارانته‌های تجاری یا تناسب با یک هدف خاص را با توجه به این پژوهش به صورت صریح یا ضمنی رد می‌کند.



تحقیق و خبر

با کلیک بر روی لوگوی زیر به

بزرگترین کانال تلگرامی مهندسان

تست نرم‌افزار ایران

پیوندید

SST
Scientific
Software
Testing

با استفاده از آن به کمپانی‌ها در جهت تحویل سریع و مداوم اپلیکیشن‌های با کیفیت موبایلی و Enterprise، کمک می‌کند. eggPlant رویکر مبتنی بر تصویر (Image-Based Approach) را برای Functional Testing به نام خود ثبت کرده است، که با استفاده از آن مشتریان قادر خواهند بود، اپلیکیشن‌های خود را از منظر درست و صحیح کاربر مورد تست قرار بدهند. محصولات Network Emulation و تست Performance قدرتمند، نیز به توسعه‌دهندگان اجازه می‌دهد تا به سادگی اپلیکیشن‌هایی که تحت طیف وسیعی از شرایط هستند را تست کنند.

eggPlant Functional ابزار اتوماسیون تست Functional شرکت TestPlant است، و قابلیت‌های بسیار قدرتمندی برای تست اپلیکیشن‌های موبایل و Responsive Web Design ارائه می‌دهد. در این گزارش، eggPlant Functional موفق به کسب بالاترین امتیاز در چند قابلیت حیاتی از گارتر شد. این قابلیت‌ها عبارتند از: Cross-Browser Platform Testing و Testing Real Device Testing. علاوه بر این، محصول امتیاز خوبی را در "تطبيق با مهارت‌های نقش غیرفنی" (Match To Nontechnical Role Skills) بدست آورد. این پارمتر برجسته‌ترین پارمتر در سهولت استفاده از ابزار و قابلیت‌های آن، توسط افراد مختلف و نه تنها توسعه‌دهندگان است.

George Mackintosh (جورج مکینتاش) مدیر عامل TestPlant می‌گوید: "مشتریان ما در صنعت بانکداری، رسانه و Healthcare، همگی نگران تحویل سرویس‌های موبایل قوی و Online هستند. در حال حاضر eggPlant بهترین ابزار جهت ساخت و تولید LUX (تجربه کاربری) بهتر است". وی ادامه می‌دهد: "هر کسی در TestPlant به خاطر اینکه به آنها به عنوان یک شرکت پیشرو در اتوماسیون تست نرم‌افزار نگرسته می‌شوند، خوشحال هستند".

Gartner هر فروشنده، محصول یا سرویسی را که در نشریات پژوهشی خود به تصویر می‌کشد، تأیید نمی‌کند، علاوه بر این به کاربران تکنولوژی پیشنهاد نمی‌کند تا تنها این فروشندگان، که دارای بالاترین رتبه هستند را انتخاب نمایند. نشریات پژوهشی Gartner

TestPlant بالاترین امتیاز برای موارد کاربرد اتوماسیون تست برای Responsive Web and Application Design را در گزارش بدست آورد

eggPlant Functional بالاترین امتیاز را در دو مورد از سه مورد کاربرد در گزارش ظرفیت‌های حیاتی برای اتوماسیون تست نرم‌افزار Gartner برای سال ۲۰۱۶ را بدست آورد (Gartner's 2016 Critical Capabilities for Software Test Automation Report).

TestPlant، تولید کننده eggPlant range از سری ابزارهای کیفیت نرم‌افزار، یازدهم می ۲۰۱۶ اعلام کرد که بالاترین امتیاز را در دو مورد از سه مورد کاربرد تحلیل شرکتهای Gartner با عنوان گزارش "ظرفیت‌های حیاتی برای اتوماسیون تست نرم‌افزار" به دست آورده است. این گزارش، ظرفیت فنی راه-

دهندگان ۱۲ راهکار (Solution)

اتوماسیون تست نرم‌افزار را ارزیابی کرده است، و در این میان TestPlant بالاترین امتیاز را در دسته‌بندی مورد کاربرد Mobile Applications and Responsive Design به دست آورده است. این گزارش اکنون در وبسایت www.gartner.com در دسترس عموم قرار دارد.

آنتونی ادواردز (Antony Edwards) به عنوان معاون فنی TestPlant می‌گوید "ما به خاطر به دست آوردن بالاترین امتیاز در دو مورد از سه مورد کاربرد در گزارش اخیر اتوماسیون تست نرم‌افزار Gartner، بسیار هیجان زده هستیم، و معتقدیم که این موضوع به چگونگی رسیدن ما به سطوح بالاتری از اتوماسیون برای دگرگونی-

های دیجیتال کمک می‌کند". او می‌افزاید: "ابزارهای eggPlant در DevOps، محیط‌های توسعه سنتی و Agile به منظور کمک به سازنده‌تر بودن کسب-وکارهای دیجیتال به بهبود قابل توجه در زمان رسیدن محصول به بازار استفاده می‌شوند، که تمامی این موارد منجر به کیفیت بهتر نرم‌افزار و بهبود تجربه کاربری (User Experience (UX) می‌گردند".

eggPlant range، سویی از ابزارهای سهل-الاستفاده اتوماسیون تست نرم‌افزار آماده کرده است، که



ISTQB®

International Software Testing Qualifications Board

ISTQB و دیگر هیج!

سیلابس‌های آموزشی خود را به همراه محتوا در آنها ارائه کرده است. موفقیت چنین موسساتی معمولاً با تعداد مخاطبین آنها سنجیده می‌شود. این موسسه به دلیل سیاست‌هایی که در حوزه بین‌المللی در پیش گرفته است، موفق شد که در انتهای سال ۲۰۱۳، خود را به رکورد بی سابقه آموزش و اعطای Certificate به بیش از ۳۰۰,۰۰۰ آزمونگر برساند. و این آمار اکنون به تعداد بسیار بیشتری رشد یافته است.

گزارشی از وضعیت امروز ISTQB

در اغلب کشورهای صنعتی، آزمونگر مجاز کسی که دارای Certificate است [به عنوان یک ابزار آموزشی برای آزمونگرها مورد پذیرش قرار می‌گیرد. کریس کارتر] Chris Carter، رییس و مسئول ISTQB می‌گوید: "من فکر می‌کنم طرح ISTQB موفق بوده است، چون سیلابس آموزشی ما و واژه‌نامه‌ای که ما بصورت آزادانه به عموم مردم ارائه می‌دهیم، منجر به استانداردسازی اصطلاحات حرفه‌ای می‌شود. علاوه بر این Certificate را در سطوح [Level] مختلف از مبانی [Foundation] تا پیشرفته [Advanced] و متخصص [Expert] ارائه می‌دهیم، و به این ترتیب حرفه‌ای‌های آزمون در شغل خود به درستی پشتیبانی شده و با Best Practice‌های دنیا در مبحث آزمون به‌روز می‌مانند.

انگلیسی باشد، که بسیاری از ما لافل به فراخور تاریخ سرزمینمان با آن آشنایی کافی داریم. نمونه دیگری از این دست، STBOK است که تحت لایسنس QAI Global فعالیت می‌کند و مدارک آن از سوی ISCB منتشر می‌شود.

STBOK: Software Testing Body of Knowledge
ISCB: International Software Certification Board

هر چند که این موسسه نیز به صورت بین‌المللی فعالیت می‌کند، اما اساس و پایه آن آمریکاییست و در حوزه قاره آمریکا از محبوبیت بیشتری برخوردار است. البته به خاطر دارم که یک سال و خورده‌ای پیش، توسط یکی از شرکت‌های داخلی، یکی از دوره‌های این موسسه که در نهایت با ارائه مدرک CAST به پایان می‌رسید، نیز در ایران برگزار شد، که البته از اقبال یا عدم اقبال آن اطلاعی ندارم.

اما یکی از موسساتی که طی چند سال اخیر و البته با توجه به سن کوتاه و ۱۴ ساله خود، اقبال جهانی خوبی داشته است، ISTQB می‌باشد:

ISTQB: International Software Testing Qualification Board

نقطه قوت این موسسه، همکاری مستقیم با هیأت‌های ملی آزمون نرم‌افزار در هر کشور و به رسمیت شناختن آنها و نیز تدوین دوره‌ها بر اساس نیازمندی‌هاییست که از تمامی کشورها به دست آنها می‌رسد. علاوه بر اینکه این موسسه مستقیماً اقدام به انتشار کتب می‌کند که

آشنایی با موسسات آموزشی آزمون نرم-افزار در دنیا

آیا تا به حال از خود پرسیده‌اید که دانش آزمون نرم-افزار چگونه جمع‌آوری و تدوین می‌شود، یا چگونه تبدیل به مبنایی برای عملکرد آزمونگران می‌گردد؟ آیا تا به امروز این سوال برایتان پیش نیامده، که تمام دانش‌ها مانند PMBOK، ITIL، COBIT و ... دارای متولیان هستند، آیا این قاعده برای آزمون نرم‌افزار هم صدق می‌کند؟

باید بدانید که چنین موسساتی در دنیا وجود دارند. بعضی از این موسسات قدیمی‌تر و برخی جدیدتر هستند. برخی شکل بین‌المللی دارند، برخی کشوری و برخی نیز میان چند کشور عمومیت دارند. دانشی که تمام آنها در حوزه آزمون نرم‌افزار ارائه می‌دهند، از یک بدنه و یک اصول پیروی می‌کند. اما گاهی اوقات استانداردهای مرجع آنها با یکدیگر متفاوت است.

به عنوان نمونه می‌توان از ISEB یاد کرد که مخفف کلمات زیر است:

Information Systems Examinations Board

این موسسه به عنوان قسمتی از بدنه جامعه کامپیوتر بریتانیا به ایفای نقش می‌پردازد، و البته به صورت بین‌المللی نیز فعالیت می‌کند. البته شاید یکی از علل موفقیت کمتر آن به نسبت موسسات دیگر که در این حوزه فعالیت می‌کنند، همان انحصارطلبی همیشگی

ISTQB چگونه فعالیت می‌کند؟

این موسسه برنامه‌های خود را از طریق هیأت‌های کشوری آزمون نرم‌افزار و یا نمایندگان خود در آن کشور پیش می‌برد، که خوشبختانه ایران یکی از معدود کشورهای غرب آسیاست که دارای هیأت ملی آزمون نرم‌افزار بوده، و این هیأت توسط ISTQB مورد تائید قرار گرفته است.

ممکن است هیأت‌های ملی مربوطه، نسخه‌های تکمیلی خود روی این سیلابس را ایجاد و نگهداری نمایند، و البته ممکن است چنین کاری، انحرافات کوچکی را به نسبت منابع اصلی انگلیسی شامل شود، مانند، ارجاعات به استانداردهای محلی. هیأت‌های ملی، سازگاری متقابل برنامه‌دراستی و امتحاناتشان را هماهنگ کرده و تضمین می‌کنند. در این زمینه، هیأت مسئول، هیأت بین‌المللی صلاحیت آزمون نرم‌افزار است.

امتحانات مبتنی بر نسخه جاری سیلابس، و زبان مربوط به آن برنامه درسی (مثلاً فارسی، انگلیسی و...)، در زمان امتحان است. امتحانات توسط هیأت ملی مربوطه و یا توسط مرجع منصوب شده صدور گواهینامه از طرف ISTQB، ارائه و برگزار می‌شود. اطلاعات بیشتر در مورد سیلابس و امتحانات را می‌تواند از طریق www.ISTQB.org بیاید. وبسایت ISTQB لینک به هیأت‌های ملی را نیز فراهم کرده است.

برگزاری دوره در ایران

طی چند سال گذشته دوره‌های این موسسه به صورت افتان و خیزان و توسط یکی از موسسات آموزشی و البته با اجازه هیأت ملی آزمون نرم‌افزار ایران انجام گردید، که به دلیل ضعف در برگزاری، جلوی ادامه آن گرفته شد. متأسفانه تا لحظه نگارش این مقاله هنوز خبر موسیقی مینی بر برگزاری مجدد و رسمی این دوره‌ها در ایران خبری واصل نشده است، اما طی اخباری که چند ماه پیش از سوی رییس این موسسه در ایران به دست بنده رسید، احتمال برگزاری مجدد و قریب‌الوقوع این کلاس‌ها در ایران و برگزاری امتحانات بعد از ماه مبارک رمضان وجود دارد.

چارت آموزشی ISTQB

در پایان نگاهی به چارت کلان آموزشی ISTQB در سال ۲۰۱۶ خواهیم داشت. قسمت‌هایی که زیر آنها با یک کادر آبی رنگ واژه In Development یا Beta نوشته شده است، هنوز آماده بهره‌برداری نشده‌اند.

در اینجا بیش از ۲۰،۰۰۰ آزمونگر مجاز در آلمان، بیش از ۱۰،۰۰۰ نفر در نروژ، و بیش از ۲،۰۰۰ آزمونگر مجاز سوئدی داریم. حتی در کشور کوچک آیسلند بیش از ۱۰۰ آزمونگر مجاز وجود دارد. در اکثر این کشورها، مدرک آزمونگری مجاز پیشیاز استخدام در پست آزمونگری و یا عقد قرارداد به عنوان پیمانکار آزمون است.

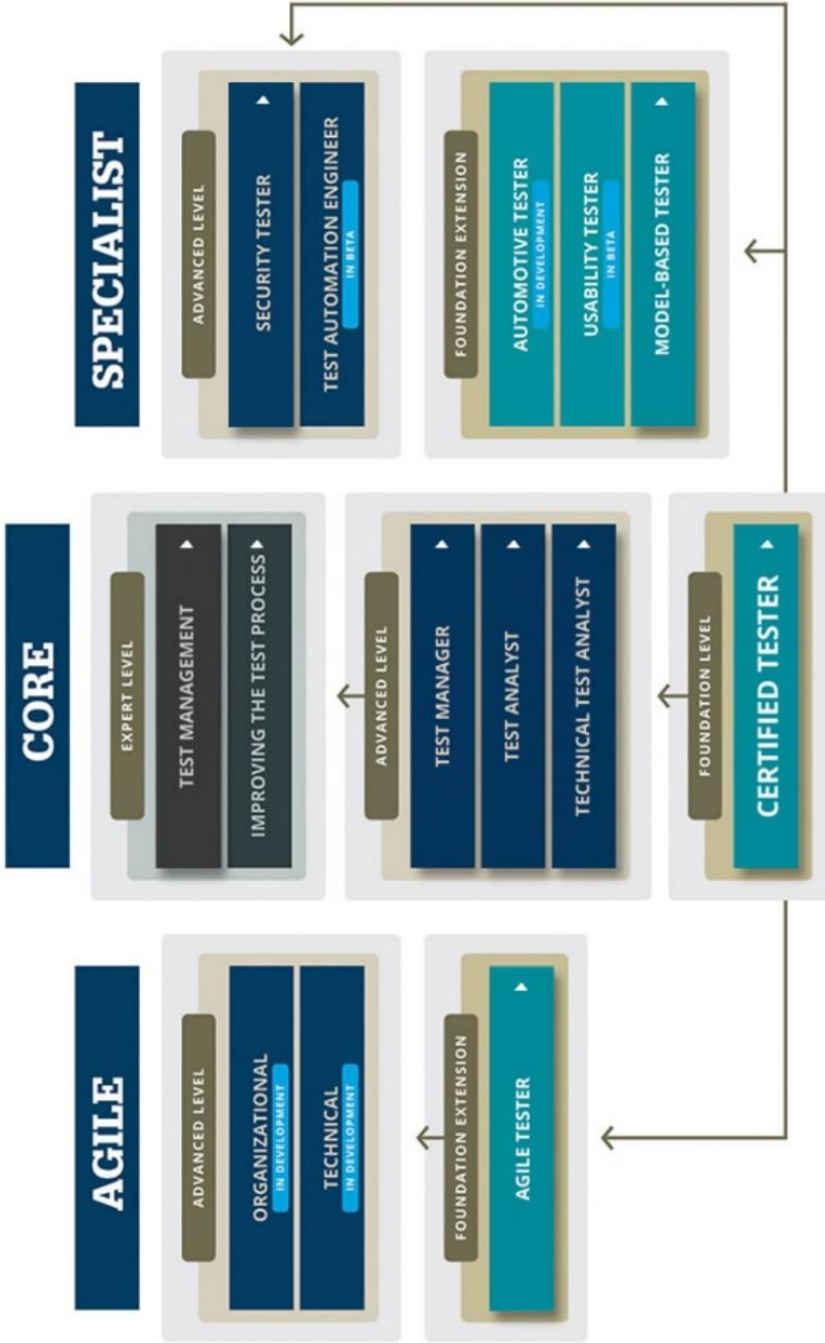
یک نظرسنجی در سال ۲۰۱۱ انجام شده در آلمان، سوییس، و اتریش نشان داد که نزدیک ۷۵٪ افراد شمایی از ISTQB را می‌دانند. بیش از ۷۰٪ آنها نیز در سطح مبتدی، Certificate دارند. در حدود ۹۰٪ نیز اذعان کردند که آموزش مفید بوده است.

پذیرش گسترده این طرح آموزشی با رشد قوی و مداوم در اعضای ISTQB [افرادی که برای اخذ گواهینامه ISTQB اقدام می‌کنند] آشکار شده است. ۴۷ هیأت آزمون در بیش از ۷۰ کشور نمایندگی دارند. ۱۲ سال قبل، ISTQB تعداد انگشت‌شماری عضو داشت. اما اکنون در همه جای دنیا نمایندگی دارد. در حال حاضر [واژه] آزمونگر مجاز رشد کرده و به یک علامت تجاری مشهور در صنعت IT دنیا تبدیل شده است، و به شکل قابل توجهی به بهینه‌سازی آزمون در فرآیند توسعه نرم‌افزار کمک کرده است.

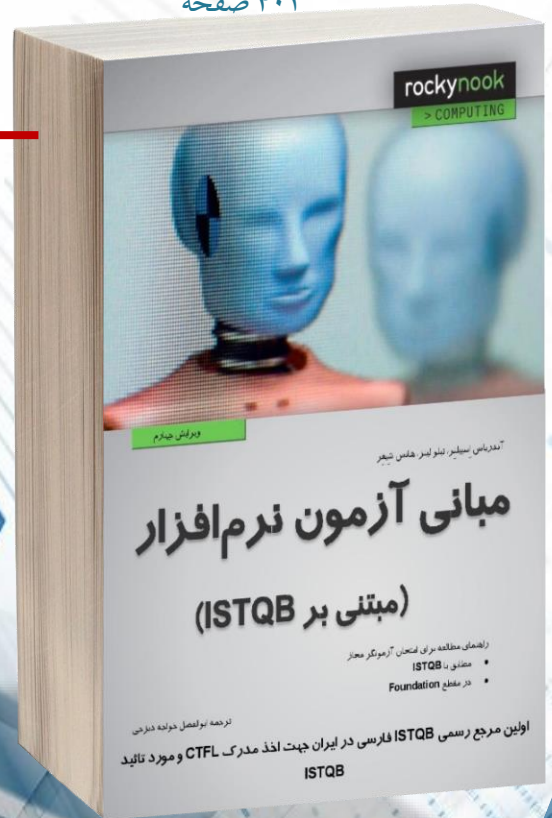
تعداد دانشکده‌هایی که در برنامه آموزشی خود به طرح آزمونگر مجاز پیوسته‌اند، قابل توجه است. دوره‌ها در مکان‌هایی مانند آخن [Aachen] و برمن در آلمان، اوسلو [Oslo] در نروژ، ریکیاویک [Reykjavik] در آیسلند، و ویسمار [Wismar] در آلمان تدریس تست تست می‌کنند. تصمیم‌گیری در این مورد که کدام دانشکده‌ها این دوره‌ها را برگزار کنند، همیشه توسط ISTQB انجام می‌شود. ارتباط آنها با بسیاری از آگهی‌های شغلی که درخواست پیشنهاد دارند، مشخص می‌شود. برای افراد مشغول در توسعه نرم‌افزار، داشتن دانش پایه‌ای آزمون نرم‌افزار کم و بیش مورد نیاز است، که بهتر است این مورد بوسیله یک گواهینامه مشخص شود. استفاده رسمی از استانداردهای بین‌المللی و نیز پیشنهاد به کار بستن برخی از این استانداردها مانند IEEE 829، ISO/IEC 25010، ISO 9126، ISO 25012، ISO 9000، و IEEE 1028، منجر به اعتبار روزافزون این موسسه شده است، و این در حالیکه که بسیاری از موسسات مشابه تمرکز زیادی روی استفاده از استانداردهای بومی خود دارند که البته در دیگر نقاط جهان به کار نمی‌آید.

ابوالفضل خواجه دیزجی





برای اولین بار در
ایران، منبع رسمی
ISTQB با ترجمه
فارسی



ترجمه: ابوالفضل خواجه دیزی
(مدیر و مشاور آزمون نرم‌افزار)

برای اطلاعات بیشتر و تهیه کتاب با استفاده از لینک زیر، به بزرگترین کانال تلگرامی مهندسان تست نرم‌افزار ایران بپیوندید



مدیریت نسبت... .

مدیریت نیازمند... .

ارتباط بین مدیریت نیازمندی‌ها و مدیریت آزمون نرم‌افزار

اما چه ارتباطی میان استخراج و مدیریت نیازمندی‌ها با مدیریت آزمون نرم‌افزار وجود دارد؟ باید گفت این ارتباط به قدری قوی و در هم تنیده است، که شاید نتوان به سادگی مرز میان آنها را تشخیص داد.

استخراج نیازمندی همواره جز اولین فعالیت‌های تولید است، و اگر مشتری در فازهای بعدی پروژه هم دخالت و همکاری داشته باشد، در راستای فهم خود از نیازمندی‌ها، وارد کار خواهد شد.

همانطور که گفتیم مشتری فقط روی رفع نیازمندی‌های خود پای می‌فشارد. یک جمله را همواره به خاطر داشته باشید:

مشتری تنها احتیاجات خود را می‌فهمد و لایبر

بنابراین باید زمانیکه محصول را مورد بهره‌برداری قرار می‌دهد، احساس کند که نیازمندی‌ها مرتفع شده است.

تنها به همین دلیل است که ارتباطی محکم میان آزمون نرم‌افزار و مدیریت نیازمندی‌ها وجود دارد.

ابتدا و انتهای کار یعنی استخراج نیازمندی‌ها و مدیریت آزمون توسط مشتری تعیین می‌شود.

تمام استانداردهای دنیا که در راستای انواع آزمون نرم-افزار شکل می‌گیرند، مانند ISO/IEC 25010 و ISO 9126، سعی در تقسیم‌بندی گونه‌های مختلف آزمون بر مبنای Best Practice‌های حاصله از دسته‌بندی نیازمندی‌های مشتریان در دنیا دارند. می‌توان گفت آزمون نرم‌افزار و مدیریت نیازمندی‌ها دارای یک DNA هستند.

به عنوان نمونه اگر به اسناد RUP مراجعه کنید خواهید دید که این Framework، آزمون نرم‌افزار را به دو قسمت Functional و Non-Functional تقسیم کرده، و سپس ۱۳ نوع آزمون مختلف در این دو دسته‌بندی جای داده است. و از همه جالبتر اینکه این آزمون‌ها را با عنوان Types Of Requirement (انواع نیازمندی) شناسایی می‌کند.

اکثریت غالب نرم‌افزارهای مدیریت آزمون امکان اتصال مستقیم به نرم‌افزارهای مدیریت نیازمندی را دارا می‌باشند، و آن دسته‌ای که این امکان را ندارند، غالباً به این دلیل از آن بی‌بهره‌اند که سازنده آنها ابزاری برای مدیریت نیازمندی‌ها مهیا نکرده است، و علاوه بر این به دلیل یکپارچه نبودن با نرم‌افزارهای دیگر تولیدکنندگان که دارای چنین ابزاری هستند، از این مهم عقب مانده‌اند.

مدیریت نیازمندی‌ها، ریشه تولید صنعتی

اولین عصری که در هر آشنایی بیشتر از همه چیز خودنمایی می‌کند این است که: "چه نیازی از من برطرف خواهد شد". این موضوع در تجارت و کسب و کار، مهمترین موضوع در برقراری پیوند میان دو طرف است، که اکثر اوقات آنها را به نام کارفرما و پیمانکار می‌شناسیم.

همیشه ابتدای آشنایی میان این دو، از بیان نیازمندی‌های کلی آغاز می‌شود، و به مرور زمان اشکال دقیقتر و پیچیده‌تری به خود می‌گیرد، آنچنانکه گاهی اوقات منجر به ایجاد سوتفاهم میان طرفین می‌شود، و اکثر اوقات یکی از طرف‌ها زیر بار نیازمندی مطروحه نمی‌رود، با این استدلال که نیازمندی‌های جدید در راستای نیازمندی‌های کلی که در روز اول مورد بحث قرار گرفته بود، نیست.

کارفرما یا همان مشتری سه چیز را برای شما به ارمغان می‌آورد:

۱. پول
۲. شهرت یا سوء شهرت
۳. نیازمندی

دو موضوع اول جز خواسته‌های شماس(البته به جز سوء شهرت). اما تنها خواسته مشتری از شما رفع نیازمندی‌ها توسط سیستمیست که در آینده مورد بهره‌برداری قرار خواهد گرفت. اگر مسیری که در آن گام بر می‌دارید، مشتری را به این موضوع مجاب کند که شما قادر به پاسخگویی به نیازمندی‌های او نیستید، نه تنها پروژه، پول و شهرت را از دست خواهید داد، حتی ممکن است در جامعه صنعتی به سوء شهرت نیز متهم شوید.

پس باید همیشه به خاطر داشته باشید که توانمندی فنی، رزومه مناسب، کار تیمی، سرعت بالا و ... هیچکدام بدون رفع نیازمندی‌های مشتری برای شما پول و شهرت را به ارمغان نمی‌آورند.

به ضرس قاطع می‌توان گفت مهمترین یا تنها عاملی که منجر به شکل‌گیری شاخه مهندسی نرم‌افزار در حوزه کامپیوتر شد، پاسخگویی به نیازهای مشتری بود.

مشتری کاری به چگونگی کارکرد شما در تیم‌تان ندارد. اکثر اوقات حتی به تکنولوژی‌های شما هم کار ندارد(مگر آنکه عدم استفاده از یک تکنولوژی خاص حال یا آینده باقی سیستم‌هایش را به مخاطره بیندازد، و به این ترتیب تکنولوژی نیز جزئی از نیازمندی‌هایش می‌شود).

تنها نیازمندی‌ها برای او مهم هستند.

که یک سوی آن فشار برای ایجاد امکانات جدید است، و یک سوی دیگر آن تغییر در سیستم است که خود یکی از مهمترین عوامل ایجاد Bug می‌باشد.

Klaus Pohl نویسنده کتاب "Requirements Engineering Fundamentals" در ابتدای فصل یک می‌گوید: "... هزینه‌های رفع نواقص برای یک نیازمندی در زمان توسعه پروژه بسیار بالاست. تلاش برای اصلاح یک نیازمندی در خلال کار برنامه‌نویسی ۲۰ برابر بزرگتر از اصلاح آن در طول مهندسی نیازمندی‌هاست. و اگر این اتفاق در زمان Acceptance Test رخ دهد هزینه از ۱۰۰ برابر هم بیشتر خواهد بود..."

تمام صحبت‌های این قسمت از مقاله فقط یک هدف داشت، و آن اینکه تاکید کنیم:

"Static Analysis را انجام دهید.

از دست دادن آن یعنی از دست دادن سرمایه‌های هنگفت مالی."

مدیریت نیازمندی در ایران

موضوع مدیریت نیازمندی‌ها در ایران تقریباً به عنوان یک کار شیک و غیرقابل انجام (به خاطر مسائلی از قبیل وقتگیر بودن و صرف منابع انسانی) نگریسته می‌شود. البته به دلیل مذکور، باید فقدان دانش آنرا نیز افزود که غالباً به علت عدم استقبال، با چنین فقری علمی همراه شده است.

مدیریت مداوم نیازمندی تقریباً عنصر فراموش شده ALM (چرخه حیات ایلکیشن) است، و اغلب با Issue Tracking و گاهی اوقات با مدیریت تغییرات اشتباه گرفته می‌شود. یکی از مشخصه‌های مدیریت نیازمندی به جز بحث بهبود کیفی نیازمندی‌ها که قبلاً ذکر شد، به Requirement Traceability معروف است. یعنی جایکه شما اصل و ریشه هر یک از نیازمندی‌ها را به صورت پشت در پشت شناسایی (از کل به جز به کل) خواهید کرد، و به عنوان یکی از مزایا می‌توانید متوجه شوید که آیا نیازمندی مطرح شده در دامنه سیستم اول که احتمالاً در پروپوزال به آن اشاره کرده- اید قرار دارد یا نه.

در چند شرکت و سازمان IT ایرانی چنین سیستمی وجود دارد، و یا با سواست کنترل می‌شود؟ متأسفانه مراحل بعدی هم اغلب، افتان و خیزان پیش می‌روند.

قطعا شما تا کنون در شرکت‌های IT کار کرده‌اید و قطعا تمام شما تا کنون با اسناد تحلیلی منسوخ شده که بعد از استخراج نیازمندی‌ها تولید می‌شوند، روبرو شده- اید. متأسفانه تحلیل هم غالب اوقات به صورت Adhoc

استخراج نیازمندی‌های گنگ است که مرتب عوض می‌شوند، و همین موضوع منجر به ایجاد نرم‌افزارهای پرتغییر می‌شود. یعنی تیم‌ها مجبور هستند به جای اجتناب از تغییر که یکی از عوالم ایجاد Defect در سیستم است، از آن استقبال کنند (البته توضیح معمول و مقدماتی در رابطه با Static Analysis- به عنوان تکنیکی در Static Test- که موارد

مذکور در اینجا را نیز در خود می- گنجاند، حداقل به اندازه یک فصل کتاب زمان نیاز دارد، که مجال آن در این مقاله

یافت نمی‌شود).

آمارهای عجیبی



که ناشی از موفقیت در این کار وجود دارد، بسیاری از شرکت‌های درجه یک دنیا را به انجام این فرآیند وادار کرده است. به عنوان نمونه، Tilo Linz در کتاب Software Testing Foundations عنوان کرده است، که: "... Static Analysis می‌تواند هزینه Defectها را در توسعه تا ۷۵٪ یا بیشتر کاهش دهد. ...". این یعنی پروژه‌ها بسته به ماهیت خود می‌توانند با این کار، ۱۰ تا ۳۵ درصد هزینه‌های کل توسعه را کاهش دهند.

از آنجاییکه استخراج نیازمندی‌ها جز اولین فعالیت‌های پروژه است، کیفیت آن تاثیر مستقیم روی مراحل بعدی از جمله تحلیل، طراحی، پیاده‌سازی و آزمون خواهد داشت.

بی‌کیفیتی در نیازمندی منجر به افزایش سرسام‌آور هزینه زمانی و مالی پروژه خواهد شد، و علاوه بر این Maintain پروژه را به دلیل درخواست‌های مکرر مشتری به شدت مشکل می‌کند. این سختی دو سو دارد

البته در بعضی از ابزارها تمام این امکانات در یک مجموعه گنجانده شده است، اما شرکت‌هایی مانند IBM که بنا به سیاست‌های تجاری خود، علاقه خاصی به تولید ابزارهای تخصصی به صورت Package دارند، معمولاً ترجیح می‌دهند که Packageها را با پروتکل‌های استاندارد خود به هم متصل نمایند.

آزمون و مدیریت نیازمندی

شاید بسیاری از ما نمای مشهور RUP که در یک تصویر فازها، دیسپلین‌ها و تکرارها را در یک حالت نمونه و معمولی

ارائه کرده بود به خاطر داشته باشیم. اگر به همان تصویر نگاه کنیم می- بینیم که Test به عنوان آخرین دیسپلین مربوط به Development. کمی بعد از دیسپلین Requirements و قبل از Analysis & Desing و Implementation آغاز شده است.

سوالی که در آن زمان ذهن بسیاری را مشغول می‌کرد، این بود که چرا Test قبل از Implementation آغاز شده است؟

و اما پاسخ:

یکی از مهمترین مسائلی که در استخراج و مدیریت نیازمندی‌ها نقش بازی می‌کند، تضمین کیفی خود نیازمندی‌های استخراج شده است، که به عنوان بخشی از فرآیند تضمین کیفیت از آن یاد می‌شود.

این موضوع نه فقط در RUP، بلکه در فرآیند آزمون نرم‌افزار و قبل از اینکه محصول برای آزمون آماده شود، لحاظ می‌گردد، که اصطلاحاً بدان Static Test نیز می‌گویند. یکی از مهمترین مراحل در این قسمت، روی افزایش کیفیت نیازمندی‌ها متمرکز می‌شود. در این مرحله از توسعه سعی می‌شود که نیازمندی‌ها تا حد امکان روشن و شفاف شوند. در موارد بسیاری مشتری جزئیات نیازمندی خود را نمی‌داند. در موارد بسیار کمی نیز ترجیح می‌دهد آنها را عنوان نکند، تا در آینده دست خود را برای مانور در تغییر برخی از امکانات پروژه باز نگذارد. در اینجا فقط کارفرماست که می‌تواند خود را از این مشخصه نجات دهد. و آن هم برگزاری جلسات متعدد با نمایندگان از مجموع ذینفعان پروژه (به عنوان منابع استخراج نیازمندی) و نگارندگان نیازمندی‌هاست، تا به این ترتیب خوراکی قابل اطمینان برای ادامه چرخه توسعه فراهم نماید، زیرا امروز ثابت شده است که یکی از عوامل اساسی بی‌کیفیتی و شاید مهمترین آن

صورت می‌پذیرد، و پس از آن یک طراحی سریع و در یک مرحله مانده به آخر، پیاده‌سازی انجام می‌گیرد. قریب اوقات هم همان تحلیلیگر، یک آزمون سریع روی امکان مربوطه انجام داده و پس از رفع نواقص، سیستم برای تحویل به مشتری آماده می‌گردد. جای تعجب نیست که با چنین فرآیندی اکثر سیستم‌های تولید داخل، مملو از ایراد و بسیار هزینه‌بر خواهند شد. باید گفت بدون مدیریت نیازمندی‌ها، هم تحلیل، هم طراحی، هم پیاده‌سازی و هم آزمون آسیب‌شدیدی خواهند دید، که رقم آن اکثر اوقات به هزینه‌ای بیش از تولید منجر خواهد شد.

نتیجه

در آخر این مقاله باید به نکته‌ای اشاره کنم، که تمام این مقاله برای رسیدن به این نتیجه مکتوب گردید و آن اینکه:

بدون مدیریت نیازمندی، مدیریت آزمون حتی اگر به دقیق‌ترین شکل هم انجام شود مانند یک بنای ناقص است. چون نمی‌توان ضمانت کرد که کیفیت حاصل شده از مدیریت آزمون، انتظار مشتری را برآورده سازد، و این همان هدف آزمون است که به این ترتیب برآورده نمی‌شود.



ابوالفضل خواجه دیز جی

تنها یک رئیس وجود ندارد: مشتری. او می‌تواند هر کسی در شرکت، از مدیرعامل گرفته تا پائین را اخراج کند. تنها با خرج کردن پولش در جایی دیگر!

Sam Walton

Founder Of Walmart

فجایع

تاریخ

نوم افزا را به

دلیل نقص

در فرآیند

تست



هم اکنون با کلیک بر روی لوگوی زیر به بزرگترین کانال تلگرامی متخصصان تست نرم‌افزار ایران پیوندید

SST
Scientific
Software
Testing



هدی رضوی

علت فنی: یک خطای برنامه‌نویسی منجر شد که تعداد خودکشی‌ها در یک سال به صورت دو برابر فرض شود، که این خود برای از بین رفتن کل مطالعه کافی بود.

ضعف مهندسی تست: ضعف استفاده از تکنیک‌های Black Box برای کشف خطا از جمله Equivalence Partitioning.

۲۰۰۰-درمان سلطان منجی به مویک

حادثه: نرم‌افزار رادیوترایی که توسط شرکت Multidata Systems International (تولید کننده محصولات رادیوترایی) در محاسبه مقدار درست رادیوگرافی دچار اشتباه می‌شد. به این ترتیب بیماران در معرض رادیوگرافی مضر و گاه بسیار خطرناک قرار

می‌گرفتند. پزشکانی که قانوناً ملزم به بررسی دقیق محاسبات نرم‌افزاری بودند، در این ماجرا به قتل متهم شدند.

هزینه: ۸ نفر کشته و ۲۰ نفر به شدت مجروح شدند.

علت فنی: نرم‌افزار مقدار اشعه را بر اساس اطلاعاتی که وارد شده بود، محاسبه می‌کرد، و گاهی اوقات، اشعه را با مقدار دو برابر می‌تاباند. اما چرا این اتفاق به صورت مکرر اتفاق می‌افتاد؟ دلایل زیادی در این راستا وجود داشت، اما علت اصلی این بود که الگوی مهندسی نرم‌افزار موجود در این سیستم‌ها منقضی شده بود؛ این الگو پاسخگویی نیازهای توسعه نرم‌افزار در قرن بیست و یکم نبود، چون کاملاً مبتنی بر تفکر خطی و اصل برهم‌نهی بود.

ضعف مهندسی تست: هر چند که علت اصلی الگو مهندسی نرم‌افزار بود، اما در سال ۲۰۰۰ هم وضعیت مهندسی تست در دنیا در نقطه‌ای قابل قبول و علمی بود. بنابراین این مشکل با هر دلیل ریشه‌ای باید در فرآیند آزمون کشف می‌شد. اما این ایراد را می‌شد با این روش‌ها کشف کرد: ۱. استفاده از تکنیک‌های Black Box مانند Cause-Effect Graphing و Decision Table. ۲. می‌شد حداقل با یک Acceptance Test به احتمال بسیار بسیار زیاد جلوی این فاجعه را گرفت.

Value Analysis ناقص در آزمون نرم‌افزار ۳. فقدان آزمون در فضای شبه‌سازی شده.

۱۹۹۸- سقوط Mars Climate Orbiter در مریخ

حادثه: در سال ۱۹۹۸ دو فضاپیما نام‌های Mars Climate و Polar Lander

Orbiter. جهت تحقیق در مورد آب و هوای و دی اکسید کربن موجود در

اتمسفر مریخ و نیز وجود آب در این سیاره به سمت آن رهسپار شدند.

پس از ۲۸۶ روز سفر از زمین Mars Climate Orbiter موتور خود را روشن کرد تا

جهت چرخش حول مدار مریخ به میدان مداری آن فشار وارد کرده و به دور آن بچرخد. اتفاقی

که افتاد این بود که، موتورهای روشن شدند، اما این

فضاپیما در اتمسفر مریخ بسیار پایین آمد و احتمالاً در سطح مریخ سقوط کرد.

هزینه: ۱۲۵ میلیون \$

علت فنی: یک ایراد بسیار پیش پا افتاده باعث بروز این فاجعه شد. در حقیقت نرم‌افزاری که پیشرفته‌های Orbiter را کنترل و هدایت می‌کرد توسط یکی از

پیمانکاران NASA آماده شده بود، که به جای استفاده از سیستم اندازه‌گیری متریک (استاندارد بین المللی

اندازه‌گیری که مورد پذیرش NASA است)، از واحدهای اندازه‌گیری بریتانیایی (Imperial) استفاده

می‌کرد. همین موضوع منجر بدین شد که اطلاعات ارسالی به کامپیوتر مرکزی فضاپیما، طبق استاندارد

متریک در نظر گرفته شده و وضعیت فضاپیما همچنان در حالت امن تصور شود. در حالیکه فضاپیما مقدار

زیادی پایین آمده بود، و دیگر امکان فرارگیری در مدار مریخ را از دست داده بود.

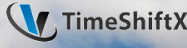
ضعف مهندسی تست: ۱. فقدان Static Analysis روی نیازمندی‌ها. ۲. فقدان آزمون در فضای شبه‌سازی شده.

۱۹۹۹- مطالعه اختفاح

حادثه: در این مورد جالب، نرم‌افزاری که برای تحلیل حوادث و فجایع مورد استفاده قرار می‌گرفت، خود یک فاجعه بود. مجله پزشکی New England از افزایش

نرخ خودکشی پس از بلایای طبیعی شدید گزارشی تهیه کرد. متأسفانه ثابت شد که این نتایج اشتباه است.

هزینه: از بین رفتن اعتبار علمی



بزرگ رہبران تست در دنیا

(قسمت اول)



Tata Consultancy Services(TCS)



+44 (0)20 7245 1800
 uk.info@tcs.com
 www.tcs.com
 4th Floor,
 33 Grosvenor Place,
 London SW1X 7HY
 UK



جوایز و افتخارات

- TCS توسط Everest PEAK Matrix Leader به عنوان به یک رهبر برای سرویس‌های آزمون مستقل، به رسمیت شناخته شد.
- Ovum, TCS را به عنوان یک رهبر در سرویس‌های آزمون برون‌سپاری شده به رسمیت شناخته است.
- Gartner, TCS را به عنوان یک رهبر در Magic Quadrant برای سرویس‌های آزمون ایلکیشن، به صورت جهانی، به رسمیت شناخته است.
- NelsonHall, TCS را به عنوان یک رهبر برای سرویس‌های سراسری آزمون نرم‌افزار به رسمیت شناخته است.
- رتبه شماره ۱ برای رضایت مشتری در سرویس‌های IT بریتانیا- Whitelane research برای سال-های ۲۰۱۵، ۲۰۱۴، ۲۰۱۳.
- دارای رتبه بالاترین کارفرما در بریتانیا در ۲۰۱۵، ۲۰۱۴-۲۰۱۳ Top Employers Institute.
- TCS به عنوان یکی از آبربرندهای سال ۲۰۱۵ شناخته شده است.

می‌دهد تا بهترین استراتژی‌هایی که با نیازهای کسب و کاری ایشان متناسب است را انتخاب نمایند. TCS رویکرد Follow The Sun را در پیش گرفته است، بدین معنی که، بدون توجه به محلی که مشتری در آنجا واقع شده است، به بالا نگاه داشتن سیستم وی به صورت 24x7 کمک می‌کند. آزمایشگاه‌های نوآوری TCS و شبکه نوآوری، به کلاینت‌ها در راستای حصول و حفظ کردن مزایای رقابتی کمک می‌کند، در حالیکه راهکارهای پیشرفته تکنولوژیک مبتنی بر تحقیقات TCS، اهداف کسب و کاری کلاینت‌ها را پشتیبانی می‌نمایند.

یکی از جامع‌ترین پورتفولیوهای قابلیت‌های مستقل آزمون ارائه شده توسط TCS این است که TCS هر دو چالش کسب و کاری و کیفی را برای کلاینت‌های جهانی خود مشخص می‌کند. این منجر می‌شود که سازمان‌ها از میان حوزه‌های مد نظر خود به منظور مخارج کلی بهینه‌سازی، متوجه اولین مزیت انگیزشی یعنی بهبود رضایت مشتری شوند. TCS در سرتاسر چرخه ارزش آزمون، سرویس‌های تضمین را پیشنهاد می‌کند، که عبارتند از: مشاوره آزمون، پیاده‌سازی سرویس‌های آزمون، و سرویس‌های مدیریت‌شده برای محیط آزمون و مدیریت داده آزمون. TCS به صورت مستمر الگوهای تضمین کیفی و آزمون، جهت کمک به کلاینت‌های خود که در آستانه یک نقطه بحرانی قرار دارند راه بازتعریف می‌کند. کتابخانه عملکردهای کسب و کاری قابل استفاده مجدد مبتنی بر دامنه در TCS، و نیز مدل تعامل و تعهد TCS که بر دو رکن کیفیت فرآیند و تولید، استوار است، این شرکت را قادر به ارائه مطمئن خدمات به کلاینت‌هایش نموده است.

واحد تضمین خدمات (TCS Assurance Services) از درآمد سالیانه شرکت را دارا بوده و بیش از ۵۰۰ مشتری در سراسر دنیا دارد. علاوه بر ۶۵ مرکز عالی آزمون حرفه‌ای (Test Centres of Excellence (TCoEs) و بیش از ۲۸۰۰ متخصص (که بیش از ۲۴۰۰۰ نفر دارای گواهینامه در ابزار تضمین کیفی هستند) در این زمینه دارد. TCoEs بیش از ۲۳ شریک Cross-Domain Strategic & Solution دارد، و جمعا مالک بیش از صد Accelerator است. همچنین TCoEs موفق به ثبت بیش از ۲۸ اختراع در زمینه تضمین نرم‌افزار شده است. شرکت‌های تحلیلی IDC, HFS Research, 451 Research، و دیگران، TCS Assurance را در زمینه دستیابی به جایگاه استراتژیک کسب و کاری در میان برندهای جهانی ستوده‌اند.

از سال ۱۹۷۵، Tata Consultancy Services(TCS) خدمات مشاوره‌ای تاتا به کسب و کارهای بریتانیا و اروپایی در راستای موثرتر بودن کمک می‌کند. TCS در بریتانیا و اروپا بیش از ۳۵۰ مشتری را سرویس‌دهی می‌کند، که ۴۴ تا از آنها در میان ۱۰۰ کمپانی برتر اروپا هستند.

- TCS در حال حاضر با موسسات زیر همکاری می‌کند:
- ۴ بانک از ۶ بانک درجه اول در حوزه سرمایه-گذاری جهانی.
- ۴ بانک از ۵ بانک اول (خرده‌فروشی) در بریتانیا.
- ۲ شرکت از ۲ شرکت جهانی و تراز اول در حوزه علوم زیستی.
- ۷ سازمان از مجموعه ۱۰ سازمان تامین زندگی و بازنشستگی.
- ۵ شرکت از بزرگان مخابرات و رسانه.
- ۸ خرده‌فروش از مجموع ۱۰ خرده‌فروش برتر در اروپا و بریتانیا.
- ۴ شرکت از مجموع ۸ شرکت بزرگ برق و گاز.
- ۲ کمپانی از شرکت‌های بزرگ سیستم‌های آب.
- ۳ شرکت از بزرگترین کمپانی‌های انرژی و منابع.
- TCS یک کارفرمای بزرگ اروپایی و بریتانیایی با بیش از ۵۰,۰۰۰ نفر پرسنل در ۵۰ مکان مختلف است. رهبران کسب و کار از سراسر اروپا از کمک‌های TCS به قاره سبز قدردانی کرده‌اند:
- ۱۴۰۰ تن از مدیران اجرایی سطح C یعنی CEO, CFO, CIO و غیره [از برترین شرکت‌های اروپایی از ۱۳ کشور، بالاترین امتیاز رضایت عمومی (۸۰٪) در صنعت راه از میان ۲۲ شرکت برتر دیگر در سرویس‌های IT، به TCS ارائه کردند.
- TCS با عنوان رتبه یک در میان ۹ کشور اروپایی رتبه‌بندی شده است، که عبارتند از: بریتانیا، فرانسه، آلمان، سوئیس، اتریش و شمال اروپا.

مدل تحویل و پورتفولیوی سرویس‌های TCS
 TCS سازمان سرویس‌های IT، مشاوره و راهکارهای کسب و کار است، و در پروژه‌های خود نتایجی واقعی و نیز سطحی از اطمینان را برای کسب و کارهای جهانی ارائه می‌دهد، که دیگر شرکت‌ها قادر به مطابقت با آن نیستند. TCS خدمات مشاوره راهنام، پورتفولیوی یکپارچه IT و زیرساخت فعال IT، مهندسی و تضمین را مهیا می‌کند. این موارد از طریق Global Network Delivery Model(GNDM) تحویل شده، و به عنوان معیار برتری و تمایز در توسعه نرم‌افزار به رسمیت شناخته می‌شود. GNDM به کلاینت‌های خود اجازه

TCS در سطح دنیا

TCS بخشی از گروه ، TATA یکی از بزرگترین و خلافتترین شرکت‌های صنعتی با ۱۰۰ شرکت، می‌باشد. این سازمان روی یک تجربه ۴۰ ساله بنیان شده است که منجر شد، TCS ارزش واقعی را از طریق Domain Expertise (تخصص دامنه‌ای)، راهکارهای اثبات شده، و سرویس‌های World-Class به سازمان‌های جهانی بیفزاید.

TCS بیش از ۳۲۰,۰۰۰ پرسنل، به نمایندگی از ۱۱۸ ملیت را در خود گنجانده است. TCS با کلاینت‌هایی از ۴۴ کشور شراکت دارد. در این میان مشتریان تکراری [که نمادی از عرضه مناسب خدمات است] به ۹۹٪ درآمد TCS کمک می‌کنند.

مشابه با Best Practice های مدیریت پروژه، Eleks مدیریت کیفیت را به منظور تسهیل در دستیابی به اهداف کیفی و تحویل راهکارهای بسیار موثر اِعمال می‌کند. مدیریت کیفیت نیازمند مشارکت مهندسان QA در تمام مراحل چرخه حیات توسعه نرم‌افزار است. Nazariy Popov، مدیر برنامه QA در Eleks می‌گوید: "امروزه انتظار می‌رود که راهکارهای نرم‌افزاری در اولین نسخه خود، کامل باشند. ما به ارائه راهکارهای دارای کیفیت بالا در سریعترین زمان، ساخت پلتفرم‌ها و ابزارهای آزمون، بهبود سودآوری، و افزایش رضایت مشتری متعهد هستیم." با بیش از ۲۰ سال تجربه در موضوع بکارگیری برنامه‌های مدیریت کیفیت، ما به مشتریان خود فراتر از انتظارات بازار کمک می‌کنیم."

زمان خود را به صورت کار داوطلبانه و حمایت از پروژه‌ها و جنبش‌های اجتماعی سپری می‌کنند.

TCS IT Futures - کار تعاملی اجتماعی که به صورت بخشی از ابتکار جهانی "Purpose4Life" انجام شد، جوایز بسیاری را نصیب شرکت کرد. این شرکت، الهام‌بخش بودن در نسل جوان، به منظور کار در خط مقدم تغییرات تکنولوژیک را برای خود هدفگذاری کرده است.

TSC Eco Futures - این طرح به منظور متعهد شدن کارکنان در سراسر TCS در بریتانیا هدفگذاری شده است، که در راستای آن TCS و مشتریان وی قادر به کاهش تاثیرات نامطلوب زیست محیطی می‌باشند. در اینجا تمرکز بر بهره‌وری انرژی، زباله، بازیافت و Business Travel است.

- برنده جایزه طلایی در "نوآوری در آموزش" ۲۰۱۴ - Learning and Performance Institute.
- دستیابی به امتیاز ۹۷٪ در "کسب و کار در جامعه" Corporate Responsibility Index توسط
- بزرگترین مشاور نرم‌افزار در بریتانیا از منظر واجد شرایط بودن برای سرمایه‌گذاران در حوزه افراد - استاندارد طلایی برای سال ۲۰۱۴.

تهدید اجتماعی TCS

TCS، سالانه به صورت میلیونی در طرح‌های شرکت-های بزرگ سرمایه‌گذاری کرده و طرح‌های بیش از ۴۰ موسسه خیریه در سراسر اروپا را حمایت می‌کند. کارکنان TCS هر سال بیش از ۱۰۰,۰۰۰ ساعت از

Eleks

Eleks یک سازمان جهانیست که توسعه محصولات نرم‌افزاری End-to-End را انجام می‌دهد. از ۱۹۹۱ راهکارهای نوآورانه شامل علوم داده (Data Science)، Mobility و Wearable. و نیز راهکارهای دیجیتال، به طور قابل توجهی به موفقیت مشتریان Eleks شامل شرکت‌های Fortune 500 کمک می‌کرد. این راهکارها به عنوان بخش ارزشمندی از Best Practice های بین المللی به رسمیت شناخته شده‌اند.

بر اساس تحقیقات Clutch، Eleks عنوان "رویکرد یکپارچه شرکت در تضمین کیفیت (QA)، تحویل امن (Delivering Secure)، کیفیت بالا، راهکارهای بدون باگ و مقرون به صرفه به مشتریان در سرتاسر دنیا" در میان ۱۰ رهبر برتر تضمین کیفیت قرار گرفت.

پرتوی‌های تضمین کیفی (QA) در Eleks:

- ۱۰ رهبر برتر در QA، بر اساس تحقیقات Clutch.
- بیش از ۲۰ سال موفقیت در QA.
- بیش از ۲۰۰ مهندس واجد شرایط.
- بیش از ۴۰ متخصص مجاز از ISTQB.
- تحویل به بیش از ۱۰۰ کلاینت بزرگ و SMB.
- QA برای چرخه کامل توسعه محصول (Full-Cycle Product Development).
- تکنیک‌های نوآورانه آزمون، دستی، اتوماتیک و موزی.
- میزبانی جامع QA.
- اعمال استانداردهای: IEEE 829، ISO 9126، ISO 27001.

eleks®

+44 (0)203 318 1274

eleksInfo@eleks.com

UK Office

5 Harbour Exchange,
South Quay, London, E14 9GE
UK



www.eleks.com/QA



amdocs

embrace challenge eXperience success

+1 314 212 7000
testing@amdocs.com
Missouri 1390 Timberlake
Manor Parkway
Chesterfield, MO 63017
USA



و بهبود می‌بخشد. Amdocs از یک مدل تحلیلی پیشرفته به منظور ارائه پیشنهادات خود استفاده می‌کند، آنتن‌کنگر هر پروژه آزمون، به طور قابل توجهی پربارتر و مقرون به صرفه‌تر خواهد شد. در این روش، Amdocs به مشتریان خود در بدست آوردن Business Goalها و یک تجربه عالی کمک می‌کند. به کار گرفتن ابزارها، متدولوژی‌ها، و Best Practiceهای اختصاصی و منحصر بفرد، باعث شده تا Amdocs Testing، در سراسر چرخه آزمون همراه با بهره‌وری و ارزش باشد. برتری آنها در بازار از حرفه‌ای‌گری، قابلیت اطمینان، و مشارکت واقعی و صحیح با مشتریان و نیز حضور جهانی منابع بسیار ماهر به منظور ضمانت یکپارچگی، تداوم پشتیبانی و استمرار "Follow The Sun" (یک نوع workflow برای انجام کارها) مشتق شده است. Amdocs Testing قسمتی از Amdocs بوده، و به عنوان یک شرکت جهانی (NASDAQ:DOX) با درآمد ۳.۶ میلیارد دلار در سال مالی ۲۰۱۴ شناخته می‌شود. Amdocs بیش از ۲۴,۰۰۰ متخصص حرفه‌ای، به منظور ارائه خدمات به مشتریان، در بیش از ۹۰ کشور و در شش قاره جهان استخدام کرده است.

Amdocs Testing یک ارائه دهنده پیشرو سرویس آزمون در جهان، برای صنعت ارتباطات است. این شرکت با بیش از ۳۰ سال تجربه در صنعت تلکام، به خوبی از چگونگی کارکرد این صنعت مطلع است. به عنوان یکی از حرفه‌ای‌های با تجربه در مخابرات، حرفه-ای‌گری آنها ترکیبی خاص و منحصر بفرد از دانش، تکنولوژی، و فرآیندهایی شامل تخصص در محیط‌های Multi-Vendor است. مخزن Best Practiceها و دانش کاری Amdocs، تماما در Frameworkای به نام Amdocs BEAT، جمع شده است. رویکرد آزمون جامع Amdocs مشتمل بر تکنولوژی، مهارت‌ها و دانشی انحصاریست که مشتریان را از نائل آمدن به یک راهکار آزمون با بهترین هزینه، سرعت فوق‌العاده، و کیفیت بالا، مطمئن می‌گرداند. این موضوع باعث می‌شود تا آنها با رساندن Business Applicationهای مشتریان خود به مرحله بهره‌برداری با سطح نقص پایین، [در حالیکه هزینه‌ها کاهش یافته و زمان رسیدن به بازار به حداقل رسیده است] به موقعیت بالاترین ارائه-دهندگان خدمات آزمون ارتقا یابند. رویکرد نوآورانه آزمون در این شرکت شامل یک Framework اختراعی و ابتکاری و مورد تایید به نام Amdocs Beat است، که فرآیند آزمون را استانداردسازی کرده

Apica

پرتال LoadTest شرکت Apica یک ابزار اسکریپت‌سازی و SaaS در آزمون Stress و Load با سطح Enterprise است که مورد اعتماد بسیاری از سازمان‌های بزرگ جهانیست. پرتال LoadTest شرکت Apica به منظور اجرای درست و دقیق آزمون-های Load با مقیاسی تقریباً نامحدود و با عمقی بی‌نظیر در جزئیات در میان ابزارهای آزمون مشابه، استفاده می‌شود. تمام سرویس‌های آزمون شرکت Apica از طریق یک پرتال مناسب، مقرون به صرفه و سهل‌الاستفاده، در دسترس است. اینکه آیا شما راه‌اندازی یک محصول جدید را در نظر دارید یا در حال برنامه‌ریزی برای افزایش ترافیک فصلی هستید، پرتال سلف‌سرویس LoadTest بر اساس SaaS به شما اجازه می‌دهد تا بدون زحمت، به هر تعدادی که می‌خواهید بر اساس تشخیص خود، قبل از ساعت پیک بار (Load)، آزمون انجام دهید. برنامه تست Load وب سایت به طور منظم و در طول زمان، حجم زیادی از تست‌ها را از ترکیب بسیاری از مکان‌های مختلف در سطح جهان انجام داده، و نتایج شما را در یک داشبورد گزارش‌گیری قوی منعکس می‌کند. این سیستم به سادگی اسکریپت‌های قدرتمند سفارشی، بدون دانش برنامه‌نویسی، ایجاد می‌کند



+46 (8) 400 273 27
sales@apicasystem.com
Apica AB, HQ
Garvargatan 9
112 21 Stockholm
Sweden



www.apicasystem.com

www.amdocs.com

کند. و چون ابزار Load Test متعلق به Apica به نام Zebraster می‌تواند بر جاواست، ابزار مستقل از پلتفرم بوده و قادر است روی Mac OS، Windows، Solaris، و سیستم‌های Linux اجرا شود. ابزار Load Test شرکت Apica از Automated Variable از Handling پشتیبانی کرده و پشتیبانی از Java Plugin را برای کاربرد اسکریپت پیچیده ارائه می‌دهد. علاوه بر این تولید اسکریپت بدون برنامه‌نویسی را نیز پشتیبانی می‌کند. استانداردهای پشتیبانی‌شده وب عبارتند از: HTTP، HTTPS، فرم‌های HTML، HTML Redirect، HTML Hyperlink، پارامترهای CGI، XML JSON Data، SOAP، و WebDAV برای استخراج و اختصاص پارامترهای دینامیک Session به صورت Runtime و بیشتر. یکپارچگی با: New Relic، AWS، TeamCity، AppDynamics، CodePipeline، Linux Agent، Windows Agent Jenkins حالت آزمایشی را با www.apicasystem.com آغاز کنید.

Apica در سال ۲۰۰۵ در استوک هولم سوئد، و سانتامونیکا کالیفرنیا به همراه دفاتری در لندن و نیویورک افتتاح شد.

توسعه دهندگان، استفاده آسان Lod Testing مبتنی بر Cloud و ابزارهای نظارت بر Performance برای Application های تستی برای ظرفیت حداکثری، Performance روزانه، زمان های Load بهبود یافته، و حفاظت از Peak Load ها را ارائه می دهد.

Apica - شریک شما در Performance
Apica موثرترین تکنولوژی را برای بهبود Performance Of Cloud و اپلیکیشن های موبایلی مهیا کرده است. Apica به طور یکسان به شرکت ها و

Centre4 Testing

- **OnSite**: راهکارهای تامین منابع، با دسترسی سریع به بهترین تسترهای حرفه ای و قراردادی در بریتانیا.
- **Cloud**: Crowd-Sourced Testers یا Testing با تست جمع سپاری شده برای برخی از بزرگترین برندهای مصرف کننده این کشور به منظور تست سازگاری یا Compatibility Testing روی دستگاه، Browser، و سیستم-عامل؛ به علاوه دسترس پذیری و تیم های تست Functional بر اساس تقاضا.
- **Grad**: تسترهای Grad، افراد باهوش، موقر، و مجاز از سوی ISTQB، که از فرآیندهای گزینشی سختی عبور کرده، و برنامه های آموزشی جامع و مداوم Centre4 Testing را مشتمل بر تکنیک های Agile، اتوماسیون و تست موبایل، آغاز کرده اند.

Centre4 Testing یکی از بزرگترین و سریعترین رشدها را در میان شرکت های سهامی خاص Software Testing و محصول محور در بریتانیا داشته است. اوایل سال ۲۰۱۵، Livingbridge، یک شرکت سرمایه گذاری خصوصی بسیار مورد توجه، پس از بیش از ۱۰۰ مشتری، مبلغ ۳.۵ میلیون پوند انگلستان در این شرکت سرمایه گذاری کرد. متخصصان مستقل تست به همراه موضوع این سرمایه گذاری، Centre4 Testing را از منظر کیفیت نفرت تا قیمت-گذاری (روی شرکت)، به رده هایی بسیار بالاتر از همسالان خود منتقل کرد.

Centre4 Testing سرویس های خود را به منظور برآوردن نیازهای گوناگون کلاینت های خود، از مشاوره Full-Service (خط سرویس OnTrack خود) تا راهکارهای تامین منابع سریع و منعطف، با استفاده از Database ای با بیش از ۱۰,۰۰۰ متخصص آزمون (ویژه بریتانیا)، مناسب سازی می نماید. جامعه کامپیوتر بریتانیا، به یکی از پروژه های مشتریان این شرکت، به عنوان یکی از فینالیست های جوایز تست نرم افزار در اروپا، جایزه اعطا کرد.

کلاینت ها از سرویس های مدیریت شده Multiple Service یا Incorporating Single Line ها متفع می شوند:

- **OneTrack**: سرویس های مشاوره ای، کلاینت ها را با مشاوره های حرفه ای تست برای کار استراتژیک، مدیریت تست، اتوماسیون تست و تست Performance مهیا می کنند. کلاینت ها از قیمت های رقابتی و تخصص فنی به عنوان یکی از نتایج همکاری نزدیک Centre4 Testing با HP، SOASTA، Smartbear، و TestPlant خوشنود هستند. همچنین این شرکت به عنوان شریک Atlassian و مشاور JIRA به رسمیت شناخته شده است.



+44 (0)20 3697 1444
ryan.hannigan@centre4testing.com
The Hop Exchange,
Southwark Street
London SE1 1TY
UK



www.centre4testing.com





+44 (0)845 270 4567
angela.vorley@hp.com
Amen Corner
Cain Road, Bracknell
Berkshire, RG12 1HN
UK



www.hp.com/go/software

در این بازارها، Maveric روی موضوعات زیر به صورت موازی متمرکز است: سرویس‌های تضمین نیازمندی‌ها، تضمین اپلیکیشن، و تضمین برنامه Maveric با هدف زدودن ریسک‌های کیفی، هزینه-ای، و زمان رسیدن به بازار، در برنامه بزرگ تحولی IT. Maveric به طور مداوم در تلاش است تا به اصول نوآوری و استقلال و تمرکز عمودی (Vertical Focus) پایبند بماند.

Maveric از سوی NelsonHall به عنوان یک متخصص تحول برای مشتریان شناخته شده است، علاوه بر این جایزه نوآوری محصول Frost and Sullivan را نیز دریافت کرده است (Sullivan Product Innovation Award). همچنین این شرکت موفق شد جایزه صنعت بانکداری خاورمیانه را برای "بهترین شریک تکنولوژی بانکی" در سال‌های ۲۰۱۳، ۲۰۱۴، و ۲۰۱۵ اخذ نماید. شایان ذکر است که Maveric در سال‌های ۲۰۱۳ و ۲۰۱۴ به عنوان یکی از فینالیست‌ها برای جایزه تست نرم افزار اروپا (European Software Testing Awards) (TESTA) نیز انتخاب شده است.

Maveric اعتبار خود را از دخیل بودن در بسیاری از محصولات بانکی، بیمه و ارتباطات از جمله

مشتریان خود در سراسر دنیا ارائه داده است. مشتریانی که کسب و کار خود را روی تحویل اپلیکیشن‌ها با سرعت بالا و کیفیت بی نظیر بنیان کرده‌اند. شایستگی اصلی HP در کیفیت است. HP از راه‌اندازی یک ایده در Palo Alto Garage به همراه بنیان-گذارانش کار را آغاز کرد. تمرکز آنها روی توانمندسازی تست کردن تکنولوژی‌هایست که پایه-های انقلاب دیجیتال خواهند بود.

میراث نرم‌افزاری HP، ارائه راهکارهای یکپارچه نرم-افزاری با هدف توسعه، تست، و مدیریت IT در طول چند دهه گذشته با محوریت نوآوری و قابلیت‌های حیاتیست، آنچه‌انکه هر ناپیوستگی جدیدی در چشم-انداز آن وارد شده است.

در حال حاضر HP روی دوازدهمین ویرایش مجموعه قدرتمند راهکارهای خود قرار دارد. HP ALM 12 نمایانگر چندین سال رهبری در تحقیق و توسعه و نوآوری به منظور تحویل نرم افزار است، که مشتریان آنرا جهت تحویل اپلیکیشن‌هایی که سبک جدید IT و رضایت مشتریان و کارکنان داخلی خود را مد نظر دارند، توانمند ساخته است.

Maveric Systems

Maveric در سال ۲۰۰۰ آغاز به کار کرد. این شرکت ارائه‌دهنده پیشرو در موضوع تضمین چرخه حیات IT (IT Lifecycle Assurance) با سرویس‌هایی در کل SDLC از Requirement تا Release است. با یک تمرکز قوی روی بخش‌های BFSI و ارتباطات، سرویس‌های Maveric به سمت تخصص‌های ویژه عمیق و راهکارهای Platform-Led سوق یافته است. Maveric به طور خاص در انگلستان روی دو زمینه زیرمتمرکز شده است:

۱. مهیا کردن ارائه‌های استراتژیک Platform-Led برای 1 Tire بانک‌ها، و شرکت‌های بیمه‌ای.
 ۲. پشتیبانی از تحولات تکنولوژیک در بانک‌ها، شرکت‌های مدیریت ثروت و تعاونی‌هایی که به تازگی مجوز فعالیت دریافت کرده‌اند.
- در بریتانیا و اروپا، موسساتی مانند Citibank, Capita, RBS, FDS1, Deutsche Bank, Validis و Ericsson, MCIS Zurich, Tesco تقاضای استفاده از سرویس‌های Maveric را دارند. Maveric حتی با موسسات پیشرو و نوظهوری مانند: Metro, Kleinworth, Benson, KBL, Raba و Bank نیز کار می‌کند.

در عقیده اقتصادی امروز، موفقیت در Business بیش از پیش بوسیله خیرگی و کارآزمودگی در دنیای دیجیتال، تعریف شده است. اغلب، اولین خط اتصال مشتری با برند و ارزش شما بواسطه نرم‌افزار رخ خواهد داد. به همین دلیل، کاربران در سراسر دنیا در تقاضای خود برای تحویل سریعتر اپلیکیشن‌های وعده شده، سخت‌گیری، بی‌رحم، و در انتظارات خود درباره کیفیت و Performance به شدت غیرمنعطف شده‌اند. این موضوع باعث می‌شود، تا HP برای تامین رضایت مشتریان در تیم‌های IT و Business Application که مسئول تحویل نرم‌افزار هستند، در موضوع ظرفیت-های تحویل اپلیکیشن در حوزه نرم‌افزار، سرمایه-گذاری‌های خوبی انجام دهد. این سرمایه‌گذاری‌ها در دو حوزه نوآوری و سهولت استفاده نیز صورت می-گیرد.

مدیریت تحویل اپلیکیشن‌های نرم‌افزاری HP (HP Software Delivery Application Management (ADM) یک پورتفولیوی یکپارچه و ثابت از اجرای نرم‌افزار با بیش از ۸,۰۰۰ Application Lifecycle و الزامات تست برای یک دهه از مشتریان HP است. HP مفتخر است که نوآورانه‌ترین، بازمترین و باثبات‌ترین راهکارها را به



+ 44 (0)20 7001 0555
info@maveric-systems.com
5 Laybourne House,
Admirals Way,
London, E14 9UH
UK



www.maveric-systems.com

این شرکت با نیروی کاری که در بیش از ۱۲۰۰ نقطه در سراسر لندن، پرینستون، دوی، سنگاپور، و چنای (Chennai) مستقر است، پروژه‌ها را با ۱۰۰٪ موفقیت اجرا و به مشتریان تحویل می‌دهد.

بوسیله End User ها به منظور پشتیبانی از مسائل کنترل سفارشی و موانع شناخته شده، در اتوماسیون تعمیم داده شود.

Axe به طور کامل از ابزارهای و سوییتهای کیفیت نرم افزار زیر پشتیبانی می‌کند:

- Selenium Webdriver (Web)
- White ,Appium(iOS And Android)
- Axe SAP ,Framework (Desktop)
- SOA ,Certified SAP GUI Testing
- Service Testing
- Microsoft- ,QC/ALM ,HP QTP/UFT
- IBM-RFT ,MTM/TF5 ,Coded UI
- Oracle OATS/OTM ,ROM

Phoenix بدست آورده است. خیرگی این شرکت در محصولاتی مانند Oracle ,Temenos T24 ,FIS ,Flexcube و TCS Bancs ,Misys به اجرای موفقیت آمیز پروژه‌های این شرکت در آینده کمک شایانی نمود.

Misys ,Flex Cube ,TEMENOS T24 Trade ,Eximbills ,Finacle ,Equation ,Fidessa ,BankTrade ,Innovation ,BSCS ,TIA ,Ingenium ,Murex ,Sungard Ethix and ,iMAL ,CS5 ,TIBCO ,Siebel

Odin

- افزایش اتوماسیون در سراسر سازمان تا ۸۰٪.
- از بین بردن سربار نگهداری اسکریپت.
- بهبود نرخ بهره‌وری در اتوماسیون.
- به کارگیری ابزارهای موجود یا Open Source.
- استانداردسازی عملکردهای اتوماسیون در سراسر سازمان.

پلتفرم اتوماسیون Enterprise و پیشرو در بازار

Axe به عنوان یک استاندارد، بوسیله بیش از ۱۰۰ سازمان در سراسر جهان، جهت یکپارچه‌سازی مهارت‌ها و رویکرد اتوماسیون تست، کاهش زمان رسیدن محصول به بازار بدون کاهش کیفیت، مورد وثوق است. Axe بسته راهکار اتوماسیون خود را با ظرفیت اجرایی اختصاصی، همراه با پشتیبانی و یکپارچه‌سازی با ابزارهای Open Source برای بازارهای تجاری پیشرو در QA Suite ها برای تست کردن اپلیکیشن‌های ERP، موبایل، وب، Desktop، SOA، و Legacy، ارائه می‌دهد.

برای رویکرد Traditional, Agile, BDD, DevOps و bi-Modal در IT

نه با متدهای Capture-Reply تاریخ گذشته، بلکه Axe برای هر نوع اپلیکیشن، به صورت منحصر به فرد، از رویکردهای متعددی برای توسعه در سراسر محیط‌های یکپارچه‌سازی مداوم و مدرن، پشتیبانی می‌کند.

تمرکز روی سناریوهای تست، کدنویسی نیست

با یک واسطه تعریف تست بر مبنای جدول و پشتیبانی روی BDD و Gherkin، افراد غیرفنی قادر به تعریف سناریوهای تست بدون نیاز به مهارت‌های کدنویسی خواهند بود. با وجود یک نوآوری، ابزار تکنولوژی تولید کد Agnostic، باعث می‌شود Axe به صورت کامل تست‌های اتوماتیک و مستندات را تولید نماید. بر خلاف دیگر ابزارهای "Scriptless"، این رویکرد منحصر به فرد بدین معنیست که Axe قادر است به طور کامل Customize شده و



+44 (0)118 370 6759
 info@odintech.com
 Atlantic House
 Imperial Way
 Reading RG2 0TD
 UK



www.odintech.com

آخرین گزارش
 آمادگی ISTQB از
 وضعیت تست در
 دنیا را از اینجا
 بدانید





+1 844 446 7587
engage@performancelabus.com
4633 Old Ironsides Drive,
Suite 220, Santa Clara,
California, 95054
USA



www.performancelabus.com

Performance Lab نقاط قوت خود و تفاوتی که این شرکت را تا اندازه یکی از رهبران صنعت تست نرم افزار بزرگ می کند را به شرح زیر ارائه می دهد:

- تجربه اثبات شده سرویس های Testing به صورت End-To-End برای صنایع مختلف.
 - تجربه اجرای موفق تست های با مقیاس بزرگ، که گاهی اوقات با بیش از ۱۰۰ متخصص آزمون همراه بوده است.
 - دسترس پذیری ترکیب مناسبی از متخصصان محلی و Offshore با حضور در سراسر جهان.
 - Framework ای اثبات شده به منظور بهبود بلوغ قابلیت های مشاوره ای اثبات شده مانند مشاوره تست.
 - مراکز تعالی (Centers of Excellence) Automation ((CoEs) تکنولوژی، مشکل از: Security CoE, Performance CoE, CoE Siebel و
 - Testing CoE، آخرین روندهای در حال مطالعه و صنعتی سازی شده در عمل و اعمال آن برای مشتریان راه تضمین می کند.
- تخصص های Performance Lab عبارتند از: Quality Assurance, Software Testing, Test Automation Tools & Services, Performance Testing Tools & Services, Mobile Testing

QASource

QASource رهبر بلامناع QA Software است، که مقر آن در Silicon Valley بوده و شرکت های بزرگ و کوچک در سراسر دنیا کار می کند. QA Source هر سال بین ۲۰٪ تا ۵۰٪ رشد داشته و تا کنون صدها پروژه موفق برای مشتریانی از قبیل IBM, Oracle, Prudential, Cisco, Mass Mutual, Ford و Luxotica انجام داده است. به عنوان یک کمپانی تست و تضمین کیفیت مستقل، QASource راهکارهای تستی سفارشی و موثری با حداکثر صرفه جویی مالی برای مشتری، ارائه می دهد. تخصص های QA در QASource عبارتند از: طراحی استراتژی های تست، اجرای تست دستی، ایجاد اتوماسیون قابل نگهداری، راه اندازی آزمایشگاه تست، مستندسازی فرآیند تست، مدیریت پروژه، Security Testing, Mobile Testing، و دیگر سرویس های تست نرم افزار.

استعداد با کیفیت

همکاری پویا میان استعداد فنی هندی و مدیریت آمریکایی راهکارهای تستی به صرفه ای را به دور از چالش های اکثر کمپانی های فراهم ساخته است. پرسنل QASource افرادی حرفه ای در حوزه QA می باشند که قادر هستند تضمین کیفیت را از فاز طراحی اولیه تا تکمیل، با فراهم کردن منابع حیاتی QA در هر گام انجام دهند. مهندسان این شرکت به طور رسمی و مستمر در حوزه متدولوژی های نوین تست، ارتباطات موثر و کارآمد و Best Practice Process ها آموزش می بینند.

کیفیتی که ارزش آفرین است

QASource، کیفیت استاندارد بین المللی، پایبندی به مهلت پروژه و رضایت مشتری را سرلوحه خود کرده است. شعار این شرکت چند ده میلیارد دلاری این است: "کیفیت ارزش می آفریند!"



تحقیق و خبر

PerformanceLab یک شرکت تخصصیست که

در سال ۲۰۰۸ برای رسیدگی به یک خلاء موجود و در حال رشد در حوزه تضمین کیفیت و Software Testing به صورت مستقل برای مشتریان Enterprise ایجاد شد. این شرکت از سال ۲۰۱۴ در ارائه این سرویس ها در بازار منطقه ای شرق اروپا پیشرو بوده است. رشد سریع این کمپانی، نتیجه موفقیت آن در میان صنایع مختلف مانند مالی، بیمه، ارتباطات، خرده فروشی، تجارت الکترونیک و مهندسی نرم افزار (مانند شرکت های لیست شده در بورس NYSE, LSE, MOEX) است. این کمپانی محدوده ای کامل از سرویس های تضمین کیفیت را ارائه داده، و تخصصی استثنایی در Performance Testing، اتوماسیون تست و Mobile Testing و نیز اتوماسیون روی دستگاه های واقعی، دارد.

تعداد پرسنل کمپانی بیش از ۵۰۰ نفر با تحصیلات و انگیزه بسیار بالا به منظور برآورده کردن نیازمندی های کلیت ها در راستای کیفیت نرم افزار با زمینه فنی و تجربی زیاد است. مدرسه Testing شرکت Performance Lab در سال ۲۰۱۰ تأسیس گردید و اکنون بیش از ۱۰۰۰ مهندس تست دارد که اخیراً بوسیله کمپانی های Fortune-500 استخدام شده اند. حضور در بازار روسیه باعث شد تا این شرکت، گزارش کیفی روسیه را در سال ۲۰۱۴ تولید نماید.



+1 (925) 271 5555
info@qasource.com
73 Ray St, Pleasanton
CA 94566
USA



www.qasource.com

SST

Scientific
Software
Testing

با کلیک روی این لوگو آخرین شماره از

نظرسنجی را دریافت کنید و به **قطب**

مهندسان تست نرم‌افزار ایران ملحق شوید





یا هک قانونمند است.

دسته‌بندی‌های SECURITY ASSESSMENT

ارزیابی امنیتی به سه بخش تقسیم می‌شود:

۱. Security Audit :

در این بخش با تمرکز لازم روی افراد یا فرآیندها، امنیت را در سطح شبکه پیکربندی می‌کنند. مسئول یک شبکه با وضع معیاری خاص اقدام به بالابردن امنیت آن شبکه کرده و با استفاده از قوانین امنیتی، تکنولوژی‌ها و ارزیابی‌های لازم، به بهبود معیاری‌ها کمک خواهد کرد. در اینجا مدیر شبکه می‌تواند با بررسی موارد امنیتی به صورت دستی یا اتوماتیک استراتژی‌های خود را پیاده‌سازی نماید. مواردی که به صورت دستی می‌توان آن را انجام داد، عبارتند از:

- مصاحبه با کارکنان
 - بررسی دسترسی‌های فیزیکی به سیستم‌ها
 - کنترل دسترسی بر روی اپلیکیشن‌ها و سیستم‌عامل‌ها
- همچنین به صورت اتوماتیک می‌توان با تولید گزارش‌های بازرسی و مانیتورینگ، از تغییرات بوجود آمده در فایل‌ها، موارد امنیتی را اعمال نمود.

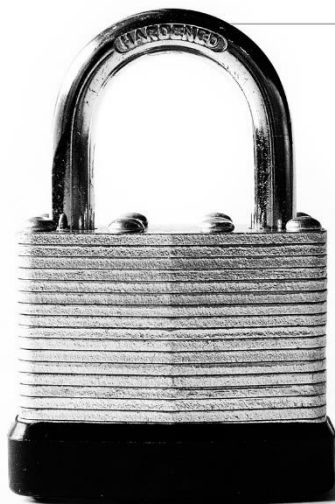
۲. Vulnerability Assessments :

مهمترین بخش از بحث تأمین امنیت، مربوط به بازرسی آسیب‌پذیری‌ها می‌باشد. این بازرسی به شما در یافتن نقاط ضعف امنیتی شناخته شده با روش اسکن شبکه، کمک می‌کند. با کمک ابزارها، شما می‌توانید بخش‌های مختلف شبکه را برای یافتن دویاس‌های دارای IP فعال و شمارش سیستم‌ها، سیستم‌عامل‌ها و اپلیکیشن‌ها جستجو کنید.

در این مقاله قصد دارم شما را با مفاهیم پایه‌ای در دنیای امنیت شبکه و تست نفوذ آشنا کنم تا دید درستی نسبت به این مسئله پیدا کرده و به راحتی و با کمترین اشتباه ممکن بتوانید عملیات تست نفوذ خود را به سرانجام برسانید.

SECURITY TEST

شرکت‌های مختلف برای ارزیابی و رتبه‌بندی شبکه‌های خود به لحاظ امنیتی از روش‌های متفاوتی استفاده می‌کنند. مسؤلان این شرکت‌ها در پیاده‌سازی استراتژی‌های خود دارای محدودیت‌هایی بوده و نمی‌توانند از هر روشی برای تأمین امنیت استفاده نمایند. این افراد برای آماده‌سازی شبکه به لحاظ امنیتی باید مهارت‌های مختلفی را داشته باشند. این مهارت‌ها در دسته‌بندی‌های گوناگونی شامل بررسی تک تک موارد امنیتی، بررسی آسیب‌پذیری و در نهایت تست نفوذ و



به طور معمول تست‌های آسیب‌پذیری که فقط از طریق نرم‌افزارها صورت می‌گیرند نمی‌توانند نتیجه مدنظر ما را داشته باشند. افراد هم می‌توانند متشاء آسیب‌پذیری‌های امنیتی باشند؛ دقیقاً به همان شکل که آسیب‌پذیری‌های موجود در نرم‌افزار و تکنولوژی وجود دارند. تست نفوذ با استفاده از تکنیک‌های مهندسی اجتماعی، قادر به مشخص کردن این موضوع است که چه هنگام کارمندان به افراد بدون احراز هویت آنان مجوز ورود

به بخش‌های شرکت را می‌دهند و یا این که در چه بخش‌هایی می‌توانند دسترسی فیزیکی به کامپیوترها داشته باشند. علاوه بر موارد گفته شده، در یک تست نفوذ می‌توان مشکلات موجود در فرآیندها، مانند ثبیت نشدن به‌روزرسانی‌های امنیتی به مدت سه روز پس از انتشار آنها را نیز تشخیص داده و برطرف ساخت. در چنین مواردی هکر به مدت سه روز یک سیستم عامل آسیب‌پذیر را برای اکتیو بودن و نفوذ در اختیار دارد. بین یک متخصص تست نفوذ با یک هکر تفاوت وجود دارد، چرا که قصد هکر تخریب سرویس‌ها و منابع و یا دزدی از اطلاعات شبکه است؛ در حالی که هدف از انجام تست نفوذ برجسته شدن مشکلات و ضعف‌های موجود در شبکه به منظور برطرف‌سازی آنهاست. این نکته را نیز باید مد نظر قرار داد که تست نفوذی که در شرایط مناسب و بطور دقیق و درست انجام نشود ممکن است منجر به اختلال در فرآیند سرویس‌ها و تخریب و توقف در فعالیت شرکت شود.

مدیریت شرکت باید دستورالعملی را برای تست نفوذ تهیه و اجرایی نماید. این

با استفاده از اسکنرهای آسیب‌پذیری، علاوه بر موارد گفته شده، می‌توانید اشتباهات رایج امنیتی مانند اکانت‌هایی با پسورد ضعیف، فایل و فولدرهایی با Permission ضعیف، سرویس‌ها و اپلیکیشن‌های پیش‌فرضی که باید پاک شوند و اشتباه در پیکربندی امنیتی اپلیکیشن‌های رایج را شناسایی نمایید. این ابزارها می‌توانند کامپیوترهایی را که در معرض آسیب‌پذیری‌های معروف و شناخته شده هستند را شناسایی کنند.

اسکنرهای آسیب‌پذیری می‌توانند سیستم‌ها و دیوایس‌های شبکه که در مقابل حملات رایج، تست و آزمایش نمایند. این تست شامل حملاتی مثل Enumeration اطلاعات مرتبط با امنیت و حملات DoS می‌باشد.

مستولیت تحلیل نتیجه اسکن آسیب‌پذیری‌ها بر عهده مدیران شبکه است و این افراد باید دانش کافی در این زمینه و تجربه کار با سیستم عامل، دیوایس‌های شبکه و اپلیکیشن‌هایی که اسکن شده‌اند را داشته باشند.

۳. Penetration Testing:

تا اینجا کار فقط توانستیم، امنیت فردی سیستم‌ها، دیوایس‌های شبکه یا اپلیکیشن‌ها را تخمین بزنیم. اما تست نفوذ به ما این امکان را می‌دهد تا به مدل امنیتی شبکه، بطور کامل دسترسی پیدا کنیم. تست نفوذ به ما کمک می‌کند تا از عمق اثرات حمله احتمالی به شبکه

آگاه شویم. همچنین تست نفوذ نقاط ضعف امنیتی را که معمولاً در اسکن‌های آسیب‌پذیری نشان داده نشده‌اند را برجسته می‌سازد.

هدف تست نفوذ، فقط نشان دادن نقاط آسیب‌پذیر نیست، بلکه در این تست، چگونگی سوء استفاده (Exploit) از ضعف‌های شبکه و نیز نحوه استفاده هکر از چندین آسیب‌پذیری جزئی برای تهدید کردن شبکه، مستندسازی می‌شود.

تست نفوذ را باید به عنوان فعالیتی که حفره‌های امنیتی را در مدل کلی شبکه نشان می‌دهد، در نظر بگیریم. این تست به سازمان‌ها کمک می‌کند تا نقاط ضعف خود را شناسایی نمایند.

دستورالعمل باید شامل یک محدوده فعالیت مشخص، شرح آن چیزی که باید تست شود و زمان انجام تست، باشد. به علت ذات تست نفوذ، هرگونه اشتباه در مورد این دستورالعمل ممکن است منجر به جراثیم کامپیوتری شود.

اینکه در یک سازمان ارزیابی آسیب‌پذیری به منظور شناسایی تهدیدات احتمالی انجام شود، بسیار مطلوب است. شما می‌توانید در زمینه آسیب‌پذیری‌های امنیتی تست‌های مختلفی را از شبکه و مولفه‌های سیستم به عمل بیاورید؛ نمونه این تست‌ها در زیر آورده شده است:

- خطای ارتباطی
- خطای موجود در تجارت الکترونیک
- درصد از دست رفتن محرمانگی اطلاعات
- میزان دید عموم بر وب سایت‌ها
- Gateway های ایمیل
- پلت فرم‌های دسترس‌پذیر از راه دور (Remote Access)
- ایمیل
- DNS
- فایروال‌ها
- پسوردها
- وب سرورها

انواع تست نفوذ

تست‌های نفوذ به دو دسته تقسیم می‌شوند:

۱. تست نفوذ خارجی

شیوه معمول تست نفوذ به این روش است. در این نوع تست بر روی سرورها، زیرساخت و نرم‌افزارهای مرتبط در هدف تمرکز می‌شود. این نوع تست می‌تواند بدون داشتن هیچگونه اطلاعاتی در رابطه با شبکه مقصد انجام شود و یا بالعکس، می‌تواند با داشتن اطلاعات کامل در



رابطه با توپولوژی و محیط مقصد صورت پذیرد. این نوع تست در میان حجم انبوه اطلاعات و تحلیل سنگین آنها در رابطه با مقصد صورت می‌پذیرد. نفوذگر، تست خارجی را به منظور شبیه‌سازی تهدیدات خارجی بر روی شبکه و یا سیستم انجام می‌دهد. مهاجم می‌تواند این نوع تست را بدون داشتن دسترسی به سیستم و بدون استفاده از اعتبار کاربری و یا مجوزهای لازم انجام دهد. مهمترین هدف موجود در انجام این تست، شناسایی نقاط ضعف بالقوه در محدوده امنیتی هدف است.

۲. تست نفوذ داخلی

تست داخلی از روش‌های مشابه در تست خارجی استفاده کرده و دید بیشتری از امنیت را ارائه خواهد کرد. تست شامل انجام آزمایش نفوذ از چندین اکسس پوینت شبکه و شامل بخش‌های فیزیکی و منطقی می‌شود. این نکته ضروریست که بدانید با وجود این که از تمام اطلاعات امنیتی برای انجام تست استفاده می‌کنیم، با این حال تست داخلی فقط تصویری از موفقیت امنیتی سازمان در آن لحظه خاص را به ما منعکس می‌کند.

هدف از انجام این تست این است که اطمینان حاصل شود که هیچکس قادر نیست از طریق شبکه داخلی و با سوءاستفاده از دسترسی کاربری خود، به سیستمی نفوذ کند. در واقع از این تست برای برجسته کردن ضعف‌های سیستم در یک شبکه خاص استفاده می‌شود. این تست دید روشنی از امنیت شبکه و سایت در اختیار ما می‌گذارد. همانطور که گفته شد، تست داخلی بواسطه چندین اکسس پوینت تحت شبکه انجام خواهد شد و شامل مجموعه تست‌های داخلی زیر می‌باشد:

- تست جعبه سیاه بدون هیچگونه دانشی از مقصد
- تست جعبه خاکستری و با اطلاعات متوسط از هدف
- تست جعبه سفید با دانش کامل از مقصد
- تست با آگاهی قبلی
- تست بدون آگاهی قبلی

اگر به امنیت و مباحث

مرتبط با آن

علاقه‌مندید به کانال

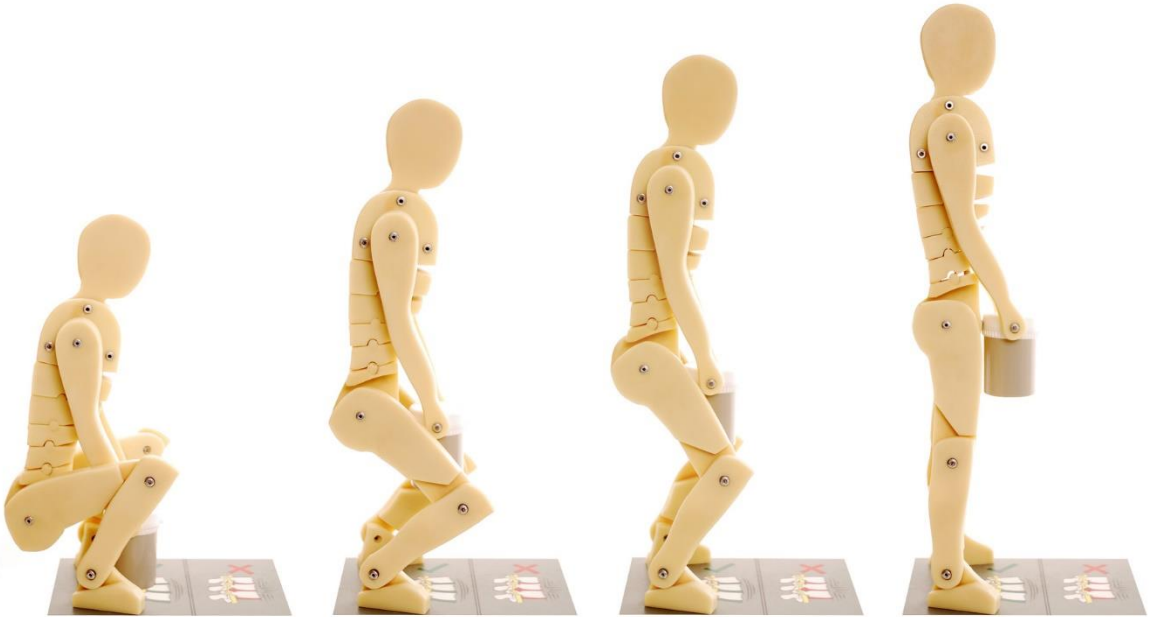
تلگرام ما بپیوندید

<https://telegram.me/GeekTown>



پهروز منصوری

چرا Manual Test؟



با اینکه اتوماسیون تست یک موهبت است، اما هنوز وقت آن نرسیده است که مانند فیلم Terminator به اربابان کامپیوتری و ماشینی خود تعظیم کنیم. هنوز هم وظایف مختلفی وجود دارد، که برای انجام آن باید از تست دستی استفاده نمود.

کاربردپذیری برنامه

اگر شما نیازمند داده‌های صرفاً عددی هستید، یک کامپیوتر برای انجام چنین مأموریت‌هایی به هیچ وجه در رتبه دوم قرار نمی‌گیرد. برای نمونه Load Testing، اغلب به همین دلیل اتومات می‌شود. در این حالت، اتوماسیون قادر است به منظور ارزیابی توانمندی برنامه جهت مدیریت ترافیک، با حداکثر سرعت هزاران

فرآیندهای اتوماتیک در صنایع متعدد، در حال پیشرفت هستند، و صنعت توسعه نرم‌افزار نیز همواره به لطف این طرح‌ها و دگرگونی‌ها، دچار تغییر شده است. اکثریت تیم‌هایی که روی موضوع اتوماسیون تست فعالیت می‌کنند، روی چگونگی خودکارسازی اکثر فریب به اتفاق فعالیت‌های تست، تحقیق و بررسی می‌کنند، تا بدین ترتیب، در حالیکه پرسنل تضمین کیفیت، فرصت کافی برای انجام دیگر کارها دارند، از اجرای تست‌های تکرارپذیر، اطمینان یابند.

اما سوال ما همچنان به قوت خود باقیست، که آیا هنوز هم تست دستی ضرورت دارد؟ پاسخ این است: بله.

در این مقاله قصد داریم تا به موضوع تست دستی بپردازیم. شاید بسیاری از مهندسان تست نرم‌افزار از خود پرسند، چرا هنوز با توجه به پیشرفت چشمگیر، روزافزون و برق‌آسای تکنولوژی‌های تست نرم‌افزار، هنوز هم تست دستی در برخی از حوزه‌های تست نرم‌افزار قدرت‌نمایی می‌کند؟ آیا این موضوع از کمبود توانمندی مهندسان تست در استفاده از تکنولوژی‌های نوین نشأت می‌گیرد، یا تست دستی همچنان از ضروریات معقول صنعت نرم‌افزار محسوب می‌شود. در این مقاله با ما همراه باشید، تا پاسخ این سوال را بگیریم.

آترا مشاهده نماید. تست‌های اتوماتیک زمان بیشتری برای Setup یا راه‌اندازی به خود اختصاص می‌دهند، و همین موضوع به شما اجازه نمی‌دهد تا ایده‌های تستی خود را با سرعت و سادگی، جامه عمل بپوشانید." با اینکه بیشتر سازمان‌ها به دنبال محیط‌های Agile هستند، اما این موضوع بدین معنی نیست که آنها فرآیندهای مرسوم در زمینه مهندسی را رها کرده‌اند. تست دستی قطعاً بهترین انتخاب برای تمام موارد نیست، اما نمونه‌هایی هست، که وجود یک ارزیاب انسانی در آن برای نگهداری از کیفیت برنامه، ضرورت دارد. با درک نقاط قوت تست دستی، تیم‌های QA قادر هستند کار خود را در این زمینه به صورت موثرتر انجام داده و منافع زیادی را در انتهای تلاش‌های خود در حوزه تست (به صورت دستی و اتوماتیک) برداشت کنند.



میلیک وارطانیان

این حالت، معمولاً تست دستی برای اجرا سریعتر است. هنگامیکه یک مجموعه بهینه از تست‌ها راه‌اندازی گردید، تیم‌های QA می‌توانند تعیین کنند که چه نوعی از ابزارهای مدیریت تست برای این تست‌ها مفید خواهند بود، و نیز کدام موارد برای اتوماسیون بهتر هستند.

آزادی برای تغییرات و تحلیل

با راه‌اندازی یک اتوماسیون، جهت اطمینان از این موضوع که اپلیکیشن با تمام مجموعه شرایط مواجه خواهد شد، باید اسکریپت‌ها از قبل آماده شده و گُذ گردند. با این حال، این موارد باید به صورت مداوم بازبینی شوند، آنچنانکه بر اساس نیازمندی‌های جاری به روز بوده و برای هر یک از تغییرات جدید در نظر گرفته شوند.

تست دستی، تیم‌های QA را قادر می‌سازد تا بر مبنای گزینه عمل کرده و فوراً روی عناصر کار کنند. کاربران می‌توانند به سادگی کد را رویت کرده و یک عنصر را بر یک مبنای تجربی در آن بگنجانند. از این جهت تست دستی نه تنها برای اجرا کم‌هزینه است، بلکه می‌تواند بازخورد سریعتری به نسبت تست اتوماتیک ارائه دهد.

Base36 می‌گوید: "در تست دستی هنگامیکه یکی از آن افکار درخشان به سراغان می‌آید، یعنی دقیقاً همان چیزی که می‌تواند مسیر پروژه را تغییر دهد؛ شما قادر هستید تا به سرعت روی آن کار کنید. این کار در تست اتوماتیک بسیار سخت است، چون باید ابتدا Test Case را Setup کرده، سپس آنها را در ابزارهای اتوماتیک تبدیل به برنامه کرده و در آخر تست را اجرا کنید. با تست دستی، شما می‌توانید تست را به سرعت انجام داده و نتیجه

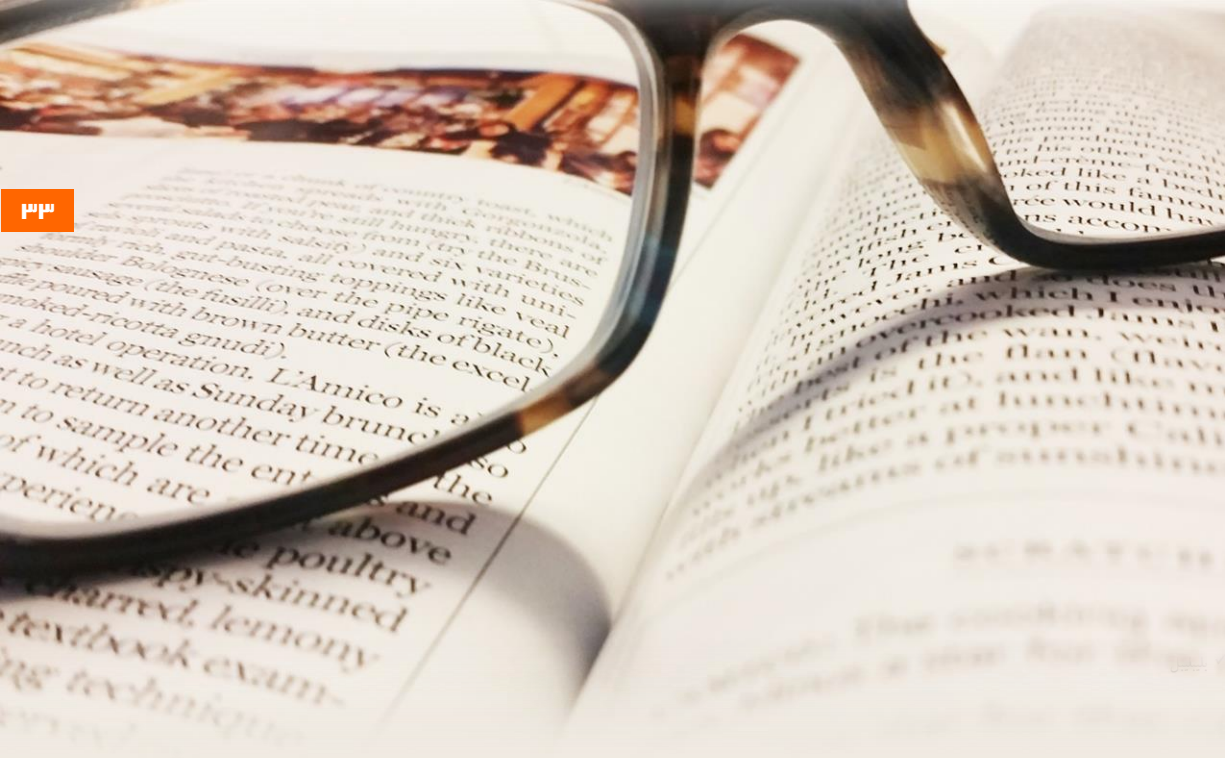
"کاربر در سایه" ایجاد نماید. با این حال زمانیکه، به جنبه‌های دیگر کاربردپذیری (Usability)، یک برنامه می‌رسیم، تست دستی آخرین حکم را صادر می‌کند. تست اتوماتیک قادر نیست یک اپلیکیشن را آنچنانکه کاربر با آن برخورد می‌کند رمزگشایی نماید، برای اینکه اتوماسیون و یک ابزار تست برای شرایط از پیش تعیین شده ساخته شده است. تست‌های انسانی قادر به دیدن یک تصویر کامل، و ارزیابی این موضوع هستند که آیا کاربر با برنامه تجربه مثبت داشته است یا خیر. این موضوع دربرگیرنده نگاهی زیبایی‌شناختی از برنامه، سهولت Navigation و عملکرد ویژگی‌های برنامه است.

با اینکه ممکن است اتوماسیون، نواقص (Defect) را در کد گزینش کند، اما این موضوع بدین معنا نیست که همه چیز مطابق انتظار کار می‌کند. اگر یک ویژگی از سیستم فاقد ابزارهایی مطمئن [برای تست] باشد، این کاربران هستند که باید وظایف آنها را به انجام برسانند. آنها نیز به احتمال زیاد، اپلیکیشن را برای چیزی که به دنبال آن هستند [و قادر به یافتن آن نمی‌باشند] رها می‌کنند. در اینجا تست‌های دستی هستند که می‌توانند همه چیز را از منظر یک کارمند [کاربر سیستم] دیده و به سادگی موارد اختلاف را گزینش کرده، و قبل از اینکه اپلیکیشن منتشر شود، آنها را برطرف نمایند.

حوزه‌های جدید کشف (Exploration)

هنوز مدت زیادی از گوشی‌های هوشمند و اپلیکیشن‌های همه فن حریف آنها نگذشته است، و امروز این دستگاه‌ها به امری عادی در زندگی روزمره مردم تبدیل شده‌اند، نمونه‌های دیگری از سخت‌افزارهای موبایلی در حال ظهور کردن هستند. برای نمونه Wearable‌ها، به عنوان یک ابزار مناسب برای محل کار و افزایش بهره‌وری در تبلیغات عنوان می‌شوند. اما باید گفت این موارد تا حد زیادی یک حوزه ناشناخته برای این اپلیکیشن‌های متنوع هستند. به همین دلیل، احتمالاً راه‌اندازی تست خودکار از ابتدای ماجرا، کاری سخت و دشوار خواهد بود. تست دستی می‌تواند مشخص‌کننده این موضوع باشد که چگونه با بهترین رویکرد این دست از سخت‌افزارها را زیر تست بگیریم. این کار در درازمدت نسبت به فرآیندهای اتوماسیون کم‌هزینه‌تر خواهد بود.

گری اوون (Gerie Owen) به عنوان یکی از مشاورین TechTarget اشاره می‌کند که در مورد روندهای جدید، نیازمند Filed Testing هستیم، تا پایدن ترتیب اطمینان بایم که اپلیکیشن در هر نقطه و همه جا توسط کاربرانی که آترا به کار می‌برند، تست شده است. در



تست و آژره

(بر مبنای ISTQB)

نیز جهت تعیین اینکه نشان دهد آیا آنها برای هدف مناسب هستند یا خیر، و همچنین [فرآیند] برای تشخیص نقص.

- تست دینامیک (Dynamic Test): آزمون‌هایی که کد را اجرا می‌کنند. آزمون‌هایی که عامه مردم آزمون را بدان گونه درک می‌کنند.
- تست استاتیک (Static Test): آزمون یک مولفه یا سیستم در سطح پیاده‌سازی یا مشخصات، بدون اجرای هیچ نرم‌افزاری (به عنوان مثال، با استفاده از بازیابی‌ها و یا تحلیل ایستا).
- تضمین کیفیت (Quality Assurance): مام فعالیت‌های درون مدیریت کیفیت، که روی فرام

را از آن استنباط نمود. مستندات که طراحی و انتخاب Test Case ها را روی آن بنیان می‌شود.

- اسکرپت تست (Test Script): دستورالعملی برای اجرای خودکار یک مورد آزمون یا یک توالی آزمون (با کنترل سطح بالاتری از ابزارهای دیگر آزمون) در یک زبان برنامه نویسی مناسب.
- تست (Test): فرآیندی مشکل از تمام فعالیت‌های ایستا و پویا در چرخه حیات، در رابطه با برنامه-ریزی، تدارک، و ارزیابی محصولات نرم‌افزاری و محصولات کاری مرتبط، جهت تعیین اینکه آیا آنها [نرم‌افزارها و محصولات کاری مرتبط] نیازمندی‌های ویژه را برآورده می‌سازند یا خیر، و-

در این قسمت از دو ماهنامه، تنها سعی می‌کنیم با بخشی از واژگان مهم و اولیه تست نرم‌افزار آشنا شویم. از آنجاییکه، تست نرم‌افزار چند متولی در دنیا دارد، سعی کردیم تا واژگانی را ارائه دهیم، که تدوین آن به عهده مشهورترین متولی تست نرم‌افزار در دنیا، یعنی ISTQB بوده است. به همین دلیل کلیه واژگان مذکور در اینجا بر اساس فرهنگ واژگان ISTQB می‌باشد.

- اجرای آزمون (Test Execution): فرآیند اجرای Test Case ها یا سناریوهای آزمون (یک فعالیت در فرآیند آزمون) که نتیجه (های) واقعی تولید می‌کنند.
- اساس آزمون (Test Basis): تمام اسنادی که می‌توان نیازمندی‌های یک Component یا سیستم

- ندارند). ۲. نتیجه یک عیب که، در طول اجرای آزمون، یک نتیجه اشتباه قابل مشاهده خارجی را نشان می‌دهد. ۳. عدم تطابق رفتار یک شی آزمون با یک کارکرد مشخص شده، که باید برای کاربرد آن [کارکرد] مناسب باشد.
- ناهنجاری (Anomaly): هر حالتی که از انتظارات مبتنی بر مشخصات نیازمندی‌ها، اسناد طراحی، اسناد کاربری، استانداردها، و غیره یا از درک و تجربه فرد منحرف شود [ناهنجاری نامیده می‌شود]. ممکن است ناهنجاری‌ها در طول بازبینی، آزمون، تحلیل، تدوین، یا استفاده از محصولات نرم‌افزاری یا اسناد کاربست پذیر پیدا شوند، که البته محدود به آنها نیستند. [IEEE 1044 نیز ناهنجاری را باگ، نقص، انحراف، خطا، عیب، نارسایی، حادثه و مشکل، نامیده است.
- نتیجه آزمون (Test Result): ۱. تمام اسنادی که در طول یک چرخه آزمون (به طور عمده سیاهه آزمون و ارزیابی آن) نوشته شده است. ۲. انتشار یا توقف شی آزمون (بسته به تعداد و شدت نارسایی‌های کشف شده).
- نتیجه مورد انتظار (Expected Test Result): رفتار یا خروجی پیش‌بینی شده و یا مورد انتظار از یک سیستم یا مولفه‌هایش، آنتنانکه توسط مشخصات یا دیگر منابع تعریف شده است (برای هر مورد آزمون).
- نقص (Defect): یک کاستی در یک مولفه یا سیستم که می‌تواند منجر به عدم موفقیت در انجام کارکرد شود، مثلا، یک عبارت یا تعریف داده نادرست. مواجهه با نقص در طول اجرا، ممکن است ختم به نارسایی مولفه یا سیستم شود.



هدی رضوی

- متد آزمون (Test Method): تکنیک طراحی آزمون را ببینید.
- متریک آزمون (Test Metric): یک ویژگی قابل اندازه‌گیری از یک مورد آزمون، اجرای آزمون، یا چرخه آزمون، مشتمل بر دستورالعمل‌های اندازه‌گیری.
- محیط آزمون (Test Environment): بستر آزمون را ببینید.
- مشخصات (Specification): سندی که به صورت ایده‌آل، نیازمندی‌ها یا دیگر مشخصه‌های مولفه یا سیستم را به شکلی کامل، واقعی، و قابل اثبات، مشخص می‌نماید. این اسناد برای توسعه - دهندگان به عنوان مبنایی برای برنامه‌نویسی است، و برای آزمونگرها به عنوان مبنایی برای توسعه موارد آزمون یا متدهای طراحی آزمون جعبه سیاه، عمل می‌کند. (اغلب، یک اسناد مشخصات، مشتمل بر روال‌هایی برای تعیین این موضوع است که آیا این نیازمندی‌ها برآورده شده‌اند یا خیر).
- مشخصات آزمون (Test Specification): ۱. سندی مشتمل بر مشخصات طراحی آزمون، مشخصات مورد آزمون و/یا مشخصات روال آزمون. ۲. فعالیت تعیین یک آزمون، که به طور معمول بخشی از "تحلیل و طراحی آزمون" در چرخه حیات آزمون است.
- مشکل (Problem): نقص را ببینید.
- معیار خروج (Exit Criteria): مجموعه‌ای از شرایط عام و خاص، که روی آنها، برای ارائه مجوز به یک فرآیند به منظور تکمیل رسمی، با ذینفعان به توافق رسیده‌ایم. هدف معیار خروج، جلوگیری از تکمیل در نظر گرفته شدن وظیفه، تا زمانیست که هنوز قطعات مهمی از وظیفه تمام نشده است. دستیابی به درجه‌ای معلوم از پوشش آزمون برای یک آزمون جعبه سفید، نمونه‌ای از یک معیار خروج است.
- مورد آزمون (Test Case): مجموعه‌ای از مقادیر ورودی، پیش‌شرط‌های اجرایی، نتایج مورد انتظار، و اجرای پس‌شرط‌های توسعه یافته برای یک هدف خاص یا شرط آزمون، از قبیل، به کار گرفتن یک مسیر ویژه برنامه یا ممیزی موافقت با یک نیازمندی ویژه.
- نارسایی (Failure): ۱. انحراف مولفه یا سیستم از تحویل مورد انتظار، سرویس مد نظر، یا نتیجه مورد انتظار. (در این حالت) برای یک نتیجه آزمون، نتیجه مشاهده شده و نتیجه مورد انتظار مطابقت

- آوردن اطمینان از برآورده شدن نیازمندی‌های کیفیت، متمرکز است.
- تکنیک طراحی آزمون (Test Design Technique): یک روال برنامه‌ریزی شده (مبتنی بر مجموعه قوانین) که جهت استخراج و/یا انتخاب موارد آزمون استفاده می‌گردد. در اینجا تکنیک - های طراحی آزمون مبتنی بر مشخصات، مبتنی بر ساختار، و مبتنی بر تجربه وجود دارند. این واژه تکنیک آزمون نیز خوانده می‌شود.
- تکنیک طراحی آزمون White Box (White Box Test Design Technique): هر تکنیکی جهت استخراج و/یا انتخاب موارد آزمون مبتنی بر تحلیلی از ساختار درونی Test Object.
- تکنیک طراحی آزمون Black Box (Black Box Test Design Technique): روال قابل - تکرار جهت استخراج و یا انتخاب موارد آزمون مبتنی بر یک تحلیل مشخصات، به صورت کارکردی یا غیرکارکردی، برای یک مولفه یا سیستم بدون اشاره به ساختار داخلی آن.
- خطا (Error): یک اکشن انسانی که منجر به تولید یک نتیجه نادرست می‌شود. همچنین به طور کلی، واژه‌ای که به طور غیررسمی برای واژگانی مانند اشباه (Mistake)، عیب (Fault)، نقص (Defect)، باگ (Bug)، استفاده می‌شود.
- داده تست (Test Data): ۱. مقادیر وضعیت یا ورودی برای یک شی آزمون و نتایج مورد انتظار پس از اجرای مورد آزمون. ۲. یا داده‌ای که (به عنوان مثال، در یک پایگاه‌داده) قبل از اجرای آزمون وجود دارد و مولفه یا سیستم تحت آزمون را متأثر می‌کند یا [از آن] متأثر می‌شود.
- راهبرد آزمون (Test Strategy): ۱. تعمیم تلاش آزمون فراتر از قطعاتی که تحت آزمون قرار می‌گیرند یا مشخصه‌های کیفی شی آزمون که باید برآورده شوند. انتخاب و تعریف ترتیب (یا تعامل) متدهای آزمون و ترتیب کاربرد آنها در اشیاء آزمون مختلف. تعریف پوشش آزمون که باید توسط هر متد آزمون به دست آید. ۲. تشریح انتزاعی از سطوح آزمون و معیارهای شروع و خروج مربوطه آنها. معمولا، یک راهبرد آزمون را می‌توان برای بیش از یک پروژه استفاده نمود.
- شی آزمون (Test Object): مولفه، سیستم جزئی یکپارچه شده، یا سیستمی (در یک نسخه معلوم) که تحت آزمون قرار خواهد گرفت.
- عیب (Fault): واژه‌ای جایگزین برای نقص.



www.ck12.org

www.ck12.org

Error

در حوزه Dynamic Analysis تکنیک‌های کشف خطا به سه دسته تقسیم می‌شوند:

- White Box:** در این دست از تکنیک‌ها رفتار سیستم با توجه به ساختار درونی برنامه سنجیده می‌شود. اساس این تکنیک روی اجرای حداقل یک بار از هر قطعه کد در **Test Object** و بررسی رفتار آن قطعه کد متمرکز دارد.
- Black Box:** تکنیک‌هایی که در آن صرفاً رفتار خروجی بر مبنای ورودی‌های **Test Object** سنجیده می‌شود. در این روش ساختار درونی **Test Object** کاملاً ناشناخته فرض می‌شود. و تنها روی رفتارهای خروجی متمرکز می‌شویم.
- Intuitive and Experience-Based Testing:** این دست از تست‌ها هم به صورت **Balck Box** و هم به صورت **White Box** انجام می‌شوند. این تست‌ها از اصول مدون تجربی یا از تجربه تستر و آن هم بیشتر در حوزه تست سیستم‌هایی که تستر ابعاد آنها را به خوبی می‌شناسد پیروی می‌کنند.

قبل از اینکه وارد جزئیات مقاله شویم باید سه واژه را خوب بشناسیم. این واژگان در بخش تست‌واژه ذکر شده‌اند. اما یک بار دیگر برای خوانندگان این مقاله به صورت دقیق آنها را بیان می‌کنیم.

- Failure:** تعاریف متفاوتی از **Failure** ارائه شده است، اما مرسوم‌ترین تعریف آن عدم تطابق رفتار **Test Object** (چیزی که تحت تست قرار می‌گیرد) با رفتار مورد انتظار (از سوی ذینفع) است.
- Fault:** عبارتست از کاستی و کمبود در بخشی از سیستم که منجر به عدم کاربرد صحیح سیستم می‌شود. **Fault** عامل ایجاد **Failure** است.
- Error** یا **Mistake:** عبارتست از آکشنی انسانی که منجر به بروز **Fault** می‌شود.

ما در اینجا قصد داریم تا اشاره‌ای داشته باشیم به روش‌هایی که می‌توان با استفاده از آنها اقدام به کشف اشتباهات (**Mistake**) یا همان **Error**‌ها نمود.

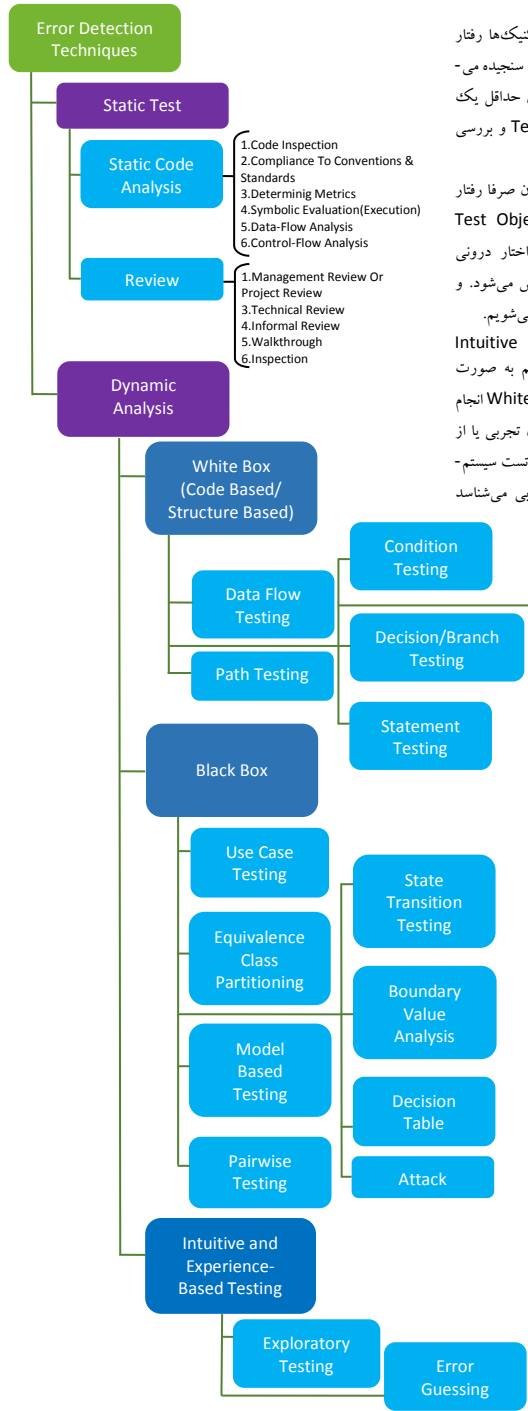
چون تعداد این روش‌ها زیاد است، مشهورترین آنها را توضیح می‌دهیم. قبل از اینکه شروع به توضیح این روش‌ها کنیم، ابتدا پرکاربردترین آنها را بر اساس آمارهای رسمی، تعیین کرده، و سپس در حین توضیح روش‌های مشهور، با حوصله بیشتری این تکنیک‌های پرکاربرد را توضیح خواهیم داد.

به صورت معمول این تکنیک‌ها به دوگفت عمده تقسیم می‌شوند: **Static Test** و **Dynamic Analysis**.

Static Test به تکنیک‌هایی گفته می‌شود که سعی می‌کنند اشتباهات را قبل از اینکه **Object** (یا **Code**) اجرا می‌شود وارد شود یافته و مرتفع کنند. اما **Dynamic Analysis** الزاماً در شرایطی حادث می‌شود که **Object** امکان اجرا روی یک ماشین را یافته است.

حجم بیشتر این تکنیک‌ها به لحاظ تعداد و پیچیدگی در قسمت **Dynamic Analysis** متمرکز شده است، اما باید گفت که **Static Test** هر چند دارای تکنیک‌های نسبتاً ساده‌تریست، اما شاید اعمال آن اگر بیشتر از **Dynamic Analysis** ارزشمند نباشد، دارای ارزش کمتری به نسبت آن نیست. به صورت معمول در ایران از تکنیک‌های مربوط به **Dynamic Analysis** و البته آن هم به صورت دست و پا شکسته استفاده شده، و **Static Test** تقریباً فراموش می‌شود.

در تصویر سمت چپ یک تصویر کلان و نسبتاً دقیق از انواع تکنیک‌های کشف خطا به انضمام دسته‌بندی آنها ارائه شده است که طبق آنچه در ابتدای مقاله وعده دادیم، موارد پر کاربرد آنها را دقیقتر تشریح خواهیم کرد.



• **Review:**

بازبینی یا Review یک واژه عمومیست که برای تمام تکنیک‌های Static Analysis استفاده می‌شود. اما منظور ما در اینجا الزاماً به تکنیک‌های معاینه و Scan یک سند که بعدها پایه‌های تولید سیستم روی آن قرار می‌گیرد، اشاره دارد. Review انواع مختلفی دارد، که بسته به شدت، رسمیت یا جنبه‌ای که بازبینی روی آن انجام می‌شود با یکدیگر متفاوت هستند. یک بازبینی نیازمند یک کار شش مرحله‌ایست: برنامه‌ریزی، شروع (Kick-Off)، آماده‌سازی فردی یا تدارک (Individual Preparation)، جلسه بازبینی (Review Meeting)، دوباره-کاری (Rework) و پیگیری (Follow-Up). بازبینی‌ها دارای فرآیند مفصل و Role‌های ویژه هستند که ذکر جزئیات آنها از حوصله مقاله خارج است. در اینجا صرفاً به انواع مشهور بازبینی اشاره می‌کنیم.

۱. **Project Review:** برخی از بازبینی‌ها، خود

پروژه یا فرآیند توسعه را تحلیل می‌کنند و تمرکزی روی اسناد توسعه ندارند. **Project Management Review** از این دست بازبینی‌ها هستند. برای مثال، چنین بازبینی‌ی تصمیم می‌گیرد که آیا از برنامه‌ها و قوانین، پیروی شده است، آیا وظایف ضروری اجرا شده‌اند، یا اثربخشی تغییرات یا بهبودهای فرآیند حادث شده است یا خیر.

۲. **Technical Review:** در یک بازبینی فنی یا

Technical Review، تمرکز روی تطابق سند با مشخصات، تناسب با اهداف مورد نظر، و تطابق با استانداردهاست. در خلال تدارک، بازبین‌ها، شی مورد بازبینی را با توجه به معیارهای معین شده بازبینی، بررسی می‌نمایند. در این نوع بازبینی بیشترین کار در تدارک انجام می‌شود.

۳. **Informal Review:** بازبینی غیررسمی،

نسخه‌ای سبک از بازبینیست. با این حال، کم و بیش از روال عمومی بازبینی با روشی ساده پیروی می‌کند. در اغلب موارد، نویسنده سند، آغازگر یک بازبینی غیررسمیست. برنامه‌ریزی، محدود به انتخاب بازبین‌ها و درخواست از ایشان جهت تحویل نظرآنها در یک مدت زمان معین است. اغلب، هیچ جلسه یا تبادل روی یافته‌ها، وجود ندارد. در چنین مواردی بازبینی تنها یک

✓ ارجاع شده (r) یا Referenced: مقدار متغیر خوانده شده و/یا استفاده می‌شود.

✓ تعریف نشده (u) یا Undefined: متغیر مقدار تعریف شده‌ای ندارد.

بر اساس تعیین متغیرهای مذکور، سه نوع ناهنجاری امکان ظهور دارد:

✓ ناهنجاری -u: یک مقدار تعریف نشده (u) از یک متغیر، در یک مسیر برنامه خوانده می‌شود (f).

✓ ناهنجاری -du: متغیر با یک مقدار تخصیص یافته پر شده (d) است، که نامعتبر/تعریف نشده بوده (u)، و به همین دلیل بدون استفاده مانده است.

✓ ناهنجاری -dd: متغیر مقداری را برای دومین بار (d) می‌پذیرد و از اولین مقدار استفاده نمی‌شود (d).

تشخیص این موارد می‌تواند منجر به جلوگیری از Failureهای بالقوه سیستم شود.

۶. **Control-Flow Analysis:** در این روش

سعی بر این است تا قطعه کد برنامه که تحت تست قرار گرفته است، تبدیل به یک گراف جهت‌دار شود که به آن **Control-Flow Graph** گفته می‌شود. تبدیل یک قطعه کد به گراف از اصول ساده‌ای پیروی می‌کند.

عبارت پشت سر هم تبدیل به یک **Node** می‌شوند تا زمانی‌که به یک شرط برسیم. در این صورت یک **Node** جدید ایجاد خواهد شد و امکان آنرا دارد که حداقل دو خروجی داشته باشد. حلقه‌ها نیز دارای یک **Node** بوده که با بازگشت به **Node**های قبلی، در گراف تعریف می‌شوند. با توجه به وضوح این گراف، توالی-های موجود در برنامه به سادگی درک شده و به این ترتیب می‌توان ناهنجاری‌های ممکن را تشخیص داد. این ناهنجاری‌ها قادر هستند به بیرون از بدنه حلقه یا ساختار برنامه که دارای چندین خروجیست، بجهند. آنها ضرورتاً به یک **Failure** منجر نمی‌شوند، اما متناسب با اصول برنامه‌نویسی ساختارمند نیستند. با این روش می‌توان ساختار برنامه را به سادگی تحلیل کرده و تمام مسیرها و نیز انتقال مسیرهای غیرمجاز را شناسایی کرد. در اینجا فرض بر این است که گراف بصورت دستی تولید نشده، بلکه بوسیله یک ابزار که یک نگاهت دقیق از متن برنامه به گراف را ضمانت کرده است، ایجاد شده است.

اما در اینجا با واژه دیگری نیز مواجه هستیم؛ **Branch Testing**. **Branch Testing** می‌توان به این شکل تعریف کرد: یک تکنیک طراحی آزمون **White Box** مبتنی بر جریان کنترل، که نیازمند اجرای تمام شاخه‌های گراف جریان کنترل (یا هر خروجی از هر تصمیم‌گیری) در **Test Object** است.

Branch Test و **Decision Test** تقریباً به یک مفهوم اشاره دارند. با این تفاوت که **Decision Test** به تست یک **Decision Test** اشاره داشته و یک **Node** تصمیم‌گیری را هدف گرفته، اما **Branch Test** یک شاخه را در **Control Flow Graph** مد نظر دارد. در ادامه با توضیح این تفاوت، موضوع کمی روشنتر می‌شود. اما قبل از توضیح باید گفت که معمولاً چیزی که به نام **Decision Testing** در دنیای نرم‌افزار شناخته می‌شود همان **Branch Testing** است، چون کسی تمایل به تست کردن یک گره خالی تصمیم‌گیری در برنامه ندارد. و اما ادامه بحث...

اتصال یعنی ارتباط دو گره از یک گراف. در متن برنامه، عبارات **IF** یا **CASE**، حلقه‌ها، و مواردی که از این دست وجود دارند، نیز تصمیم‌گیری نامیده می‌شوند. به همین دلیل این آزمون، آزمون تصمیم‌گیری یا پوشش تصمیم‌گیری (**Decision Coverage**) نیز نامیده می‌شود. البته معمولاً تفاوتی در درجه پوشش میان **Decision Testing** و **Branch Testing** وجود دارد. مثال پیش رو این موضوع را نشان می‌دهد: در اینجا یک عبارت **IF** با یک قسمت خالی **ELSE** مورد بررسی قرار می‌گیرد. در صورتی که شرط به مقدار **true** ارزیابی شود، آزمون تصمیم‌گیری، پوشش ۵۰٪ خواهد داشت. با یک **Test Case** بیشتر که در آن، شرط **FALSE** است، پوشش تصمیم‌گیری ۱۰۰٪ خواهد شد. برای آزمون شاخه، که از گراف جریان کنترل ساخته شده است، مقادیر اندک-متفاوت در پوشش نتیجه می‌شود. قسمت **THEN** شامل دو شاخه و یک گره است، قسمت **ELSE** تنها یک شاخه بدون گره است (در اینجا هیچ عبارتی وجود ندارد). بنابراین، کل عبارت **IF** با قسمت خالی **ELSE** شامل سه شاخه است. اجرای شرط با نتایج درست به دو شاخه از این سه شاخه پوشش می‌دهد، که معادل ۶۶٪،

توالی، ضامن اجرای عبارات آتی در این توالی است. عبارات مشروط (**IF**، **Case**) و حلقه-ها (**WHILE**، **FOR**)، به عنوان گراف‌های جریان کنترلی نمایش داده می‌شوند، که بیش از یک یال به خارج گره دارند. بعد از اجرای **Test Case**‌ها، باید تعیین شود کدام عبارات اجرا شوند. زمانیکه به سطح پوشش (**Statement Coverage**) از پیش تعریف شده دست یافتیم، کفایت آزمون تأیید شده، و بنابراین آزمون خاتمه می‌یابد. به طور معمول، تمام دستورالعمل‌ها باید اجرا شوند، چون تأیید دستورالعمل‌هایی که اجرا نشده‌اند، غیرممکن است. فرمول میزان **Coverage** در این نوع تست به صورت زیر است:

$$\text{Statement coverage} = (\text{number of executed statements} / \text{total number of statements}) \times 100\%$$

Statement Coverage همچنین به عنوان پوشش-**CO** (سی صفر) نیز شناخته می‌شود. این معیار بسیار ضعیف است. با این حال، گاهی اوقات ۱۰۰٪ پوشش عبارت برای دستیابی دشوار است، برای نمونه، زمانیکه، شرایط استثنا در یک برنامه تنها با بروز مشکلی بزرگ راه‌اندازی شود، یا در تمام طول اجرای آزمون این شرایط ظاهر نشوند.

۲. تکنیک پرکاربرد **Decision(Branch)**

Testing: این تکنیک هشتمین تکنیک پرطرفدار دنیاست که مجموعاً ۲۱٫۵٪ شرکت‌ها از آن استفاده می‌کنند. این تکنیک معیاری پیشرفته‌تر بر اساس **Control Flow Graph**، در تست **White Box** می‌باشد.

در **Decision Test**، به جای اجرای عبارت، اجرای تصمیمات در نظر گرفته می‌شود. نتیجه تصمیم‌گیری نیز این است که عبارت (**Statement**) در مرحله بعد اجرایی می‌شود. **Decision Test** را می‌توان بدین صورت تعریف کرد: یک تکنیک طراحی آزمون **White Box** مبتنی بر جریان کنترل، که نیازمند این است که هر خروجی تصمیم‌گیری (**TRUE** و **FALSE**) حداقل یک بار برای هر **Test Object** استفاده شود. (یک عبارت **IF** دو خروجی دارد؛ یک عبارت **CASE** یا **SWITCH** خروجی‌های زیادی دارد).

چرخه نویسنده-خواننده ساده است. بازمینی غیررسمی نوعی مطالعه سطحی بوسیله یک یا چند همکار است.

۴. **Walkthrough**: یک **Walkthrough** یا

مروزه، یک متد بازمینی غیررسمی دستی با هدف پیدا کردن نواقص، ابهامات و مشکلات موجود در سند مکتوب، می‌باشد. نویسنده، سند خود را در یک جلسه بازمینی به بازمین‌ها ارائه می‌کند.

۵. **Inspection**: بازمی، رسمی‌ترین نوع

بازمینیست. این نوع بازمینی از یک فرآیند معین رسمی پیروی می‌کند. همه افراد درگیر، معمولاً افرادی که مستقیماً با نویسنده کار می‌کنند، دارای یک نقش مشخص هستند. قوانین، فرآیند را تعریف می‌کنند. بازمین‌ها از چک‌لیست‌هایی استفاده می‌کنند که شامل معیارهایی برای بررسی جنبه‌های مختلف هستند. در اینجا هدف یافتن آنبم‌های مبهم و نواقص احتمالی، اندازه‌گیری کیفیت شی بازمینی، و بهبود کیفیت فرآیند بازمی و فرآیند توسعه است. اهداف واقعی و اصلی برای هر بازمی در طول برنامه‌ریزی تعیین می‌شود. بازمین‌ها (بازمینی‌ها) تنها برای تعداد مشخصی از جنبه‌های مورد بررسی می‌باشند.

← تکنیک‌های حوزه **Dynamic Analysis**

• **White Box**

۱. تکنیک پرکاربرد **Statement Testing**:

این تکنیک دهمین تکنیک پرطرفدار دنیاست که مجموعاً ۱۸٫۲٪ شرکت‌ها از آن استفاده می‌کنند. این تحلیل روی هر عبارتی در **Test Object** تمرکز می‌کند. **Test Case**‌ها، باید یک قسمت حداقلی از پیش تعریف شده یا تمام عبارات **Test Object** را اجرا کنند. اولین گام، ترجمه سورس کد به گراف جریان کنترل است. وجود گراف، سورس کد را جهت تعیین جزئیات عناصر کنترل که باید پوشش داده شوند، ساده می‌کند. در این گراف، عبارات به صورت گره نمایش داده شده‌اند و جریان کنترل میان عبارات به صورت یال یا **Edge** (اتصال) نمایش داده شده است. اگر توالی عبارات نامشروط [یعنی عباراتی که برای رسیدن به آنها شرط خاصی لازم نیست، و به صورت بی‌در پی اجرا می‌شوند] در قطعه برنامه‌ها ارائه شود، به صورت صرفاً یک گره نمایش داده می‌شود، و علت این امر هم این است که، اجرای اولین عبارت از

یک Test Case را به دلخواه انتخاب می‌کنیم، تا ببینیم که آیا Function‌های تعریف شده در دو زیردامنه مجاور به درستی محاسبات را انجام می‌دهند، یا خیر.

۵. **Path Testing**: این تکنیک نیز مانند تکنیک

Decision Testing و Control Flow

Analysis یک روش مبتنی بر Data Flow

Graph است. این گراف بر اساس یک قطعه

کد تولید می‌شود. در این متد Control Flow

Graph یک برنامه، برای یافتن یک مجموعه

خطی از مسیرهای مستقل اجرایی، تحلیل می‌شود.

این متد به صورت معمول از پیچیدگی

Cyclomatic جهت تعیین تعداد مسیرهای

مستقل خطی استفاده کرده و سپس Test

Case‌ها را برای هر مسیر از نتایج حاصله تولید

می‌کند. در این تکنیک میبھی به نام Basis

Path Testing وجود دارد، که از اهمیت

فوق‌العاده‌ای برخوردار است. Basis Path

در حقیقت مسیریست که با یافتن آن و تست آن

تمام Branch Coverage کامل می‌شود، هر چند

که تمام Branch‌های گراف مربوطه، پیمایش

نشده باشد. موضوع Basis Path Testing

هنوز هم یکی از مسائل پرمطالعه در مباحث علوم

کامپیوتر است. از آنجاییکه این تکنیک نیز مانند

تمام تکنیک‌های مبتنی بر Control Flow

Graph یکی تکنیک مبتنی بر ریاضیات گسسته

است، و جزئیات علمی آن زیاد می‌باشد، امکان

توضیح بیشتر در اینجا وجود ندارد.

توضیح بیشتر در اینجا وجود ندارد.

۶. **Data Flow Testing**: یکی دیگر از

تکنیک‌های White Box که در آن Test

Case‌ها با استفاده از Data-Flow Analysis

طراحی می‌شوند، و تمامیت آزمون در آن با

استفاده از Data-Flow Coverage حاصل

شده، ارزیابی می‌گردد.

• **Black Box**

۱. **تکنیک پراکارد** Equivalence Class

Partitioning: این تکنیک ششمین

تکنیک پرتعداد دنیاست که مجموعاً

۳۴٪ شرکت‌ها از آن استفاده

می‌کنند. دامنه داده‌های

ورودی

است. یک شرط در سورس کد Test Object

می‌تواند شامل شروط اتمی چندگانه باشد.

اما موضوع فقط شروط اتمی نیستند، بلکه گاهی

اوقات با شروط پیچیده‌ای طرف هستیم که

نیازمند نوع دیگری از آزمون شرط هستند؛ یعنی

آزمون چندشرطی. آزمون چند شرطی نیز مانند

آزمون شرط است با این تفاوت که همان

عملیات روی مجموعه شروطی که با عملگرهای

منطقی و سمبول‌های ریاضی به هم متصل شده-

اند(و به این ترتیب یک شرط بزرگتر را فراهم

کرده‌اند) انجام می‌شود. در حقیقت در اینجا،

هدف دیگر یک شرط نیست بلکه یک عبارت

تصمیم‌گیرست.

۴. **Domain Strategy Testing**: توضیح این

موضوع وابستگی زیادی به درک حدودی

مفهوم Decision Testing دارد، که بیشتر

تشریح شد. اما در اینجا توضیح بسیار مختصری

از این تکنیک نیز ذکر می‌کنیم.

کامپوننت‌هایی که در طول Decision Test

اجرا می‌شوند، Branch Predicate

(بارهای شاخه) موجود در Source Code

هستند، که هر Line Segment‌هایی را که از

مرزهای

Subdomain

بوسیله برنامه ایجاد

شده اند را تعریف

می‌کنند. در این

حالت ما از هر

سمت مرز

پوشش است.(آزمون تصمیم‌گیری در این مورد

۵۰٪ پوشش داشت). اجرای Test Case دوم با

شرطی که نادرست است، ۱۰۰٪ پوشش شاخه‌ای

و ۱۰۰٪ پوشش تصمیم‌گیری را به ارمغان خواهد

آورد.

بنابراین، برخلاف Statement Coverage،

برای پوشش شاخه معقول نیست که مثلاً، یک

عبارت IF فاقد هر گونه قسمتی برای ELSE

باشد. این [قسمت(منظور قسمت ELSE)] باید به

هر ترتیب اجرا شود. پوشش شاخه نیازمند

آزمودن تمام خروجی‌های تصمیم‌گیرست: هم

THEN و هم ELSE در عبارت IF؛ تمام

احتمالات برای عبارت CASE و عبور از

CASE؛ برای حلقه‌ها، هم اجرای بنده حلقه، و

هم دور زدن بنده حلقه جهت بازگشت به ابتدای

آن.

۳. **Condition Testing**: Branch

Coverage، منحصرأ مقدار منطقی نتیجه یک

شرط ("true" یا "false") را در نظر می‌گیرد،

اما چگونگی رسیدن به این مقدار منطقی را، رها

می‌کند. در اینجا دو نکته حائز اهمیت است: اول.

آیا هر دو مقدار True و False از شرط مذکور

حاصل می‌شود. دوم. اگر هر دو این حالات رخ

می‌دهند باید از چه مسیری برویم تا به این دو

مقدار منطقی برسیم. با استفاده از همین مقدار

منطقی حاصل شده است که، تعیین می‌شود کدام

شاخه در Control Flow Graph انتخاب

شود و، بر این اساس، کدام عبارت بعداً در برنامه

اجرا گردد. اما چه چیزی این موضوع را مهم

کرده است؟

اگر یک تصمیم‌گیری بر اساس چند

شرط(Partial Condition) متصل شده به هم

توسط عملگرهای منطقی، صورت گیرد، آنگاه

باید پیچیدگی شرط در آزمون، مد نظر قرار

گیرد. هدف آزمون شرط این است که علت هر

شرط اتمی [تجزیه‌ناپذیر] Partial

Condition) در آزمون، هر دو

مقدار true و false را

بپذیرد. یک شرط اتمی،

شرطیست فاقد عملگرهای

منطقی، مانند AND،

OR، و NOT، اما در

بسیاری شامل سمبول‌های

ارتباطی از قبیل > و =



وضعیت‌هایی را طی کرده است، و اکنون در چه وضعیتی قرار دارد)، منجر به رفتاری متفاوت شود، باید یک آزمون انتقال وضعیت انجام گیرد. سطوح متفاوتی از شدت آزمون را می‌توان برای آزمون انتقال وضعیت تعریف نمود. یک نیازمندی حداقلی، تهیه تمام وضعیت‌های ممکن است.

۴. تکنیک پروکاربرد Use Case Testing:

این تکنیک اولین تکنیک پروکاربر دنیا است که مجموعاً ۷۰.۸٪ شرکت‌ها از آن استفاده می‌کنند. ممکن است نیازمندی‌ها به صورت Use Case و یا Business Case تشریح شوند. ممکن است این آیتم‌ها در نمودارها ارائه شوند. این نمودارها در سطحی نسبتاً انتزاعی با توصیف تعاملات معمول کاربر-سیستم، به تعریف نیازمندی‌ها، کمک می‌کنند. ممکن است آزمونگرها Use Case را جهت استخراج Test Case ها بکار گیرند.

هر Use Case دارای یک هدف است و باید به نتیجه خاصی دست یابد. ممکن اتفاقات حادث شده در جریان یک Use Case به جایگزین-های (Alternative) بیشتری منجر شوند. بعد از اجرا، پس‌شرطها وجود خواهند داشت. تمام اطلاعات زیر برای طراحی Test Case ها حیاتیست:

- ✓ وضعیت شروع و پیش‌شرطها
- ✓ دیگر شرایط بالقوه
- ✓ نتایج مورد انتظار
- ✓ پس‌شرطها

با این حال، نمی‌توان داده‌های ورودی واقعی و نتایج را برای هر Test Case مستقیماً از Use Case استخراج نمود. برای این کار، هر ورودی و خروجی داده‌ای، باید انتخاب شوند. علاوه بر این، هر گزینه موجود در نمودار (رابطه "Extend") باید توسط یک Test Case پوشش داده شود. تکنیک طراحی Test Case های مبتنی بر Use Case، ممکن است با سایر تکنیک‌های طراحی آزمون مبتنی بر مشخصات ترکیب شود.

آزمون مبتنی بر Use Case برای آزمون تعاملات معمول کاربر-سیستم بسیار مفید است. بنابراین، بهتر است آنرا در Acceptance Test و System Test، اعمال کنید.

معین شناور، می‌توان "تحمل تعریف شده" داشت. بنابراین، حداقل سه Test Case از هر مرزی نتیجه می‌شود. یکی درون کلاس، یکی برای کران پایین کلاس و یکی برای کران بالای کلاس.

۳. تکنیک پروکاربرد State Transition Testing:

این تکنیک تعیین تکنیک پروکاربر دنیا است که مجموعاً ۲۱.۴٪ شرکت‌ها از آن استفاده می‌کنند. در بسیاری از سیستم‌ها، نه تنها ورودی فعلی بلکه سابقه اجرا (بدین معنی که Test Object تا کنون از چه State‌هایی عبور کرده است) یا رویدادها یا ورودی‌ها می‌توانند، محاسبه خروجی‌ها و چگونگی رفتار آبی سیستم را تحت تاثیر قرار دهند. در اینجا سابقه اجرای سیستم را باید الزاماً در نظر داشت. برای نشان دادن وابستگی موجود روی سابقه، از نمودار وضعیت (State Diagram) استفاده می‌شود. آنها نمودارهای وضعیت، منبئی برای طراحی آزمون هستند (آزمون انتقال وضعیت یا State Transition Test).

سیستم یا Test Object از یک وضعیت اولیه (Initial State) آغاز شده و سپس می‌تواند به وضعیت‌های مختلف برود. رویدادها (Event) می‌توانند، تغییرات یا انتقال وضعیت‌ها را راه‌اندازی کنند. یک رویداد ممکن است فراخوانی یک تابع باشد. انتقال وضعیت می‌تواند اکشن‌ها (Action) را درگیر کند. علاوه بر وضعیت اولیه، یک وضعیت ویژه دیگر، وضعیت پایانی (End State) است. ماشین‌های وضعیت محدود (Finite State Machine یا FSM)، نمودارهای وضعیت، و جداول انتقال وضعیت، این رفتار را مدل‌سازی می‌کنند.

با این توضیحات می‌توان State Transition Testing را بدین گونه تعریف کرد: در این آزمون Test Case ها جهت اجرای انتقال وضعیت معین و غیرمعین در Test Object از وضعیت‌های مختلف، طراحی می‌شوند.

در آزمون انتقال وضعیت، ممکن است Test Object، یک سیستم تکمیل با وضعیت‌های متفاوت سیستمی و نیز یک کلاس در یک سیستم شی‌گرا با وضعیت‌های گوناگون باشد. هر زمان که سابقه ورودی (سابقه ورودی به این معنیست که Test Object تا کنون چه

ممکن، برای هر عنصر داده ورودی به Equivalence Class تقسیم‌بندی می‌شوند (پارتیشن‌بندی Equivalence Class). یک Equivalence Class مجموعه‌ای از مقادیر داده‌ایست که آزمونگر فرض را بر این می‌گذارد که به روشی یکسان توسط Test Object پردازش می‌شوند. به دلیل فرض حاکم، مبنی بر اینکه اگر دیگر مقادیر ورودی، از همان Equivalence Class باشند، Test Object همان واکنش را رفتار را نشان خواهد داد، آزمون نماینده‌ای (Representative) از Equivalence Class کافی در نظر گرفته می‌شود. در کنار Equivalence Class برای داده‌های صحیح، مقادیر ورودی نادرست نیز باید به خوبی تحت آزمون قرار گیرند. با استفاده از این تکنیک، خیال تستر راحت است که سیستم با تمام انواع ورودی‌های مجاز و غیرمجاز مورد تست قرار گرفته است.

این تکنیک از جزئیات زیادی برخوردار است، که مهم‌ترین و بیشترین این جزئیات روی Partitioning صحیح و دقیق متمرکز شده است.

۲. تکنیک پروکاربرد Boundary Value Analysis:

این تکنیک چهارمین تکنیک پروکاربر دنیا است که مجموعاً ۲۸.۲٪ شرکت‌ها از آن استفاده می‌کنند. تحلیل مقدار مرزی، به Test Case‌هایی که بواسطه پارتیشن‌بندی Equivalence Class شناسایی شده‌اند، افزونه‌ای بسیار مناسب ارائه می‌دهد. عیوب، غالباً در مرزهای این کلاس‌ها ظاهر می‌شوند. این موضوع به این دلیل رخ می‌دهد که مرزها اغلب به روشنی تعریف نمی‌شوند و یا نامشخص هستند. معمولاً یک آزمون با مقادیر مرزی، Failure را کشف می‌کند. این تکنیک تنها در صورتی قابل اعمال است که مجموعه‌ای از داده‌ها در یک Equivalence Class به صورت مرتب قرار بگیرند و مرزهای قابل تشخیص داشته باشند. تحلیل مقادیر مرزی، سرحدات Equivalence Class را بررسی می‌کند. روی هر مرز، مقدار دقیق مرزی و مقادیر مجاور و نزدیک (درون و برون Equivalence Class) آزموده می‌شود. برای این کار باید از حداقل افزایش ممکن در هر دو جهت استفاده شود. مثلاً برای داده‌های دارای

نروژی، انگلیسی، اندازه صفحه (کوچک، بزرگ).

اگر تمام ترکیبات برای آزمون انتخاب شوند، $3 \times 2 \times 18$ نمونه Test Case بدست خواهد آمد. با این حال، با انتخاب ترکیبات دو به دو، ما تنها به ۹ عدد Test Case نیاز داریم. جدول ارائه شده در پایین یک راهکار ممکن را نمایش می‌دهد.

راهکار مفروض ما حالتی را نشان می‌دهد که هر سیستم عامل، یا هر زبان ممکن، و با هر اندازه صفحه ممکن، محقق شده است. همچنین نشان می‌دهد که هر زبان با هر اندازه صفحه ممکن و با هر سیستم عامل ممکن تحقق یافته است. در آخر، هر اندازه صفحه ممکن، یا هر زبان، و هر سیستم عاملی محقق شده است. اما تمام ترکیبات سه گانه ممکن (مانند، Mac، انگلیسی، کوچک) در آزمون محقق نمی‌گردند. Test Case شماره ۹ ویژه است: ترکیب Windows و انگلیسی یک ترکیب حیاتیست، اما دیگر ترکیبات با اندازه صفحه در سایر Test Case ها پوشش داده شده است و تنها در Test Case شماره ۹، اندازه صفحه بی‌اهمیت است و موارد مهم روی سیستم عامل و زبان متمرکز شده است. بنابراین در Test Case شماره ۹، اندازه صفحه را می‌توان به صورت آزادانه انتخاب نمود؛ برای مثال، اغلب یکی از حالات تحقق را می‌توان استفاده کرد.

Pairwise Testing هر گونه تعامل مخرب بین پارامترهای ظاهرا مستقل (که مقادیر نماینده انتخابی برای انجام این کار آماده شده است) را پیدا خواهد کرد. سطوح بالاتر تعاملات لزوما مورد اکتشاف قرار نخواهند گرفت. این تکنیک برای اعمال دستی ساده نخواهد بود، اما ابزارهایی برای ساده‌سازی این کار وجود دارند. این تکنیک را می‌توان جهت پوشش به سطوح بالاتر تعاملات گسترش داد.

۸ تکنیک پروکاربرد Attack: این تکنیک دوازدهمین تکنیک پروکاربرد دنیاست که مجموعاً ۱۰٫۴٪ شوکت‌ها از آن

Test case #	OS	Language	Screen
1	Mac	German	small
2	Linux	German	large
3	Windows	German	large
4	Mac	Norwegian	large
5	Linux	Norwegian	small
6	Windows	Norwegian	small
7	Mac	English	large
8	Linux	English	small
9	Windows	English	Choose freely

دو بخش است. در نیمه بالایی، ورودی‌ها (شرایط یا علل) فهرست شده‌اند؛ نیمه پایینی نیز شامل اکشن‌ها (معلول‌ها) است. هر ستون، موقعیت‌ها و شرایط آزمون را تعریف می‌کند، یعنی ترکیبی از شرایط و معلول‌ها (خروجی‌های مورد انتظار). نمونه‌ای از این جدول برای تست یک سیستم خودپرداز در جدول دیدیم.

هر ستون از این جدول، به عنوان یک Test Case تفسیر می‌شود. از جدول، شرایط ورودی - های ضروری و اکشن‌های مورد انتظار را می‌توان مستقیماً یافت. Test Case 5 (TC5) شرط‌های روبرو را نشان می‌دهد: تنها در صورتیکه کارت معتبر باشد، PIN نیز بعد از حداکثر سه بار، درست وارد شود، و وجه مورد نیاز مشتری، هم در حساب وی و هم در دستگاه موجود باشد، آنگاه پول تحویل می‌شود.

۷ تکنیک پروکاربرد Pairwise Testing:

این تکنیک یازدهمین تکنیک پروکاربرد دنیاست که مجموعاً ۱۳٫۲٪ شوکت‌ها از آن استفاده می‌کنند. این تکنیک طراحی آزمون را می‌توان زمانیکه، تعامل میان پارامترهای مختلف، ناشناخته است، مورد استفاده قرار داد. آزمون ترکیبی دو به دو با هدف پیدا کردن تعامل مخرب بین پارامترهای احتمالاً مستقل (با پارامترهایی که در آنها مشخصات شامل وابستگی‌ها نمی‌شوند) انجام می‌پذیرد. این

تکنیک از جدول Equivalence Class شروع می‌شود. برای هر Equivalence Class، مقدار نماینده انتخاب می‌شود. بنابراین، هر نماینده از یک کلاس با هر نماینده از هر کلاس دیگر ترکیب می‌شود (تنها ترکیبات جفتی به حساب می‌آیند، نه ترکیبات سطح بالاتر). برای تشریح بهتر موضوع توجه شما را به مثال زیر جلب می‌کنیم.

یک سیستم چند زبانه که قابلیت راه‌اندازی در سه محیط سیستم‌عاملی و با Resolution های مختلف را دارد، باید تست شود. و اما پارامترهایی که در زمان راه‌اندازی سیستم اهمیت فزاینده دارند عبارتند از: سیستم عامل (Mac، Linux)، یا (Windows)، زبان (آلمانی،

۵ Model Based Testing: این تکنیک

یکی از کاربردهای Model-Based Design برای طراحی و اجرای محصولات به منظور System Testing و Software Testing است. مدل‌ها را می‌توان جهت نمایش رفتار مطلوب سیستم تحت تست (System Under Test/SUT)، یا ارائه استراتژی‌های تست یا به تصویر کشیدن محیط تست استفاده نمود. در اینجا مدلی که یک SUT را تشریح می‌کند معمولاً ارائه بخش انتزاعی (Abstract) رفتار مطلوب از SUT است. Test Case های مشتق شده از چنین مدلی، Functional Test های موجود در همان سطح از Abstract Test Suite هستند. یک Abstract Test Suite را نمی‌توان به صورت مستقیم در مقابل یک SUT اجرا نمود، چون سویت در سطح نادرستی از Abstraction قرار دارد. یک Test Suite قابل اجرا باید از Abstract Test Suite مربوطه مشتق شود. در این حالت Test Suite قابل اجرا می‌تواند مستقیماً با SUT تعامل برقرار کند.

۶ تکنیک پروکاربرد Decision Table: این

تکنیک هفتمین تکنیک پروکاربرد دنیاست که مجموعاً ۲۲٫۷٪ شوکت‌ها از آن استفاده می‌کنند. این تکنیک ارائه دهنده جدول نمایشگر قوانین است. این جداول را

Decision table		TC1	TC2	TC3	TC4	TC5
Conditions	Bank card valid?	N	Y	Y	Y	Y
	PIN correct?	-	N	N	Y	Y
	Third PIN attempt?	-	N	Y	-	-
	Money available?	-	-	-	N	Y
Actions	Reject card	Y	N	N	N	N
	Ask for new PIN	N	Y	N	N	N
	"Eat" card	N	N	Y	N	N
	Ask for new amount	N	N	N	Y	N
	Pay cash	N	N	N	N	Y

می‌توان برای طراحی Test Case استفاده کرد. هدف یک آزمون مبتنی بر جدول تصمیم‌گیری (Decision Table) عبارتست از اجرای ترکیبات ورودی "جالب توجه" (یعنی ترکیباتی که به صورت بالقوه، می‌توان Failure را با استفاده از آنها مشخص نمود). علاوه بر علل و معلول‌ها، ممکن است نتایج میانی نیز با مقادیر منطقی‌شان (Truth-Value) در جدول تصمیم‌گیری گنجانده شوند. جدول تصمیم‌گیری دارای

تجربه تستی که آزمونگر در ابله‌کشن‌های مشابه داشته است استفاده می‌شود تا به این ترتیب دستیابی به نواقصی که با استفاده از تکنیک‌های رسمی ناممکن یا سخت است، امکان‌پذیر شود. از این تکنیک معمولاً بعد از اینکه تکنیک‌های رسمی به طور کامل اعمال شد، استفاده می‌گردد.

دامنه Test Case ها در این روش نیز مانند دیگر روش‌های شهودی تست معمولاً به آزمونگر درگیر در کار بستگی دارد. در این روش تست بر مبنای تجربه قبلی خود تعیین می‌کند که در چه حالات و چه شرایطی سیستم دچار خرابی می‌شود. خطاهای معمولی که غالباً با این روش مشخص می‌شوند، عبارتند از: تقسیم بر صفر، Null Pointer ها، و پارامترهای غیرمجاز.

این متد هیچ قانون مدونی ندارد (هر چند که ممکن است برای آن Best Practice هایی تدوین شده باشد)، Test Case ها نیز بسته به شرایط طراحی می‌شوند.

Test Object ای که تحت آزمون قرار دارد جمع آوری می‌گردد. این موضوع روشن می‌کند که Test Object چه انجام می‌دهد و چگونه کار می‌کند، کدام مشکلات کیفی می‌تواند وجود داشته باشد، و کدام انتظارات باید برای برنامه برآورده شوند. یک نتیجه آزمون اکتشافی ممکن است روشن شدن این مطلب باشد که کدام تکنیک‌های آزمون را می‌توان در زمان باقیمانده پروژه اعمال نمود.

در آزمون اکتشافی با واژه‌ای به نام بلور آزمون یا Test Charter مواجه هستیم.

Test Charter این ذهنیت را متبادر می‌کند که آزمون اکتشافی را، به عناصر معین برنامه (وظایف یا توابع معین) محدود کنید. به این ترتیب این عناصر مورد توجه، بیشتر تجزیه می‌شوند. واژه بلور آزمون نیز برای این قطعات کوچکتر استفاده می‌شود. نباید بیشتر از یکی دو ساعت برای آزمون یک بلور آزمون زمان صرف کرد. زمانیکه بلورهای آزمون را اجرا می‌کنید، سوالات زیر جالب توجه و مهم هستند:

- ✓ هدف از اجرای آزمون چیست؟
- ✓ چه چیزی تحت آزمون قرار می‌گیرد؟
- ✓ کدام منتهای آزمون باید مورد استفاده قرار گیرند؟
- ✓ چه نوع مشکلاتی را باید یافت؟
- ایده کلی آزمون اکتشافی به شرح زیر است:
- ✓ نتایج یک Test Case، طراحی و اجرای Test Case های بیشتر را متأثر می‌کند.
- ✓ در طول آزمون، یک مدل "ذهنی" (Mental Model) از برنامه تحت آزمون ایجاد می‌شود. مدل شما چگونگی "کارکرد" برنامه و چگونگی رفتار آن یا چگونگی رفتار باسته آن می‌باشد.
- ✓ آزمون به نسبت این مدل اجرا می‌شود که تمرکز روی یافتن جنبه‌ها و رفتارهای بیشتری از برنامه است، که هنوز قسمتی از مدل "ذهنی" نیستند، و یا با جنبه‌هایی که قبلاً پیدا شده‌اند متفاوت هستند.

۲. تکنیک پرکاربرد Error Guessing: این تکنیک پنجمین تکنیک پرطرفدار دنیاست که مجموعاً ۳۷٪ شرکت‌ها از آن استفاده می‌کنند. این تکنیک نیز یکی دیگر از روش‌های شهودی برای به دام انداختن خطاهاست. در این روش از مهارت، شهود و

استفاده می‌کنند. تکنیک‌هایی که از آنها به عنوان Attack یاد می‌شود، غالباً اشاره به حمله‌هایی جهت به مخاطره کشیدن پارامتر امنیت در سیستم است. البته گاهی اوقات و البته به ندرت Attack به معنی هجوم Thread ها نیز استفاده می‌شود، مانند همان موضوعی که در Load Test مد نظر است. در این حالت بررسی رفتار سیستم تحت هجوم تعداد زیادی Thread مد نظر است.

اما همانطور که گفته شد، موضوع Attack به معنی، به چالش کشیدن امنیت یک سیستم است، که متأسفانه به دلیل گسترده بودن این میث، توضیحات بیشتر در حوصله مقاله نیست.

• Intuitive and Experience-Based Testing

۱. تکنیک پرکاربرد Exploratory Testing

این تکنیک دومین تکنیک پرطرفدار دنیاست که مجموعاً ۶۶٫۳٪ شرکت‌ها از آن استفاده می‌کنند. اگر اسناد، که پایه و اساس طراحی آزمون را تشکیل می‌دهند، دارای کیفیت پایینی هستند، یا منسوخ شده‌اند، و یا اصلاً وجود ندارند، اصطلاحاً آزمون اکتشافی می‌تواند کمک کننده باشد. در موارد حاد، فقط برنامه وجود دارد. هنگامیکه زمان شدیدا محدود است، به دلیل زمانگیری کم این تکنیک به نسبت دیگر تکنیک‌ها، این روش خوشایند و قابل اجراء است. این رویکرد اصولاً مبتنی بر شهود و تجربه آزمونگر است.

فعالیت‌های آزمون در آزمون اکتشافی تقریباً موازی اجرا می‌شوند. در اینجا فرآیند ساختارمند آزمون اعمال نمی‌شود. یک برنامه‌ریزی روشن از فعالیت‌های آزمون که قبلاً انجام شده است در اینجا اجرا نمی‌شود. در این نوع از آزمون عناصر ممکن در Test Object (وظایف و توابع ویژه آن) "بررسی" می‌شوند. بنابراین تصمیم‌گیری می‌شود که کدام قطعات آزمونده خواهند شد. تنها Test Case های معدودی اجرا می‌شوند و نتایج‌شان تحلیل می‌شود. با اجرای Test Case ها رفتارهای "ناشناخته" Test Object

بیشتر مشخص خواهند شد. در این نوع از آزمون هیچ چیز مورد توجه ویژه قرار ندارد، و تنها هدف پیدا کردن خرابیست. هر اطلاعاتی نیز، جهت تعیین Test Case بعدی مورد استفاده قرار می‌گیرد. در این روش گام به گام، دانش در

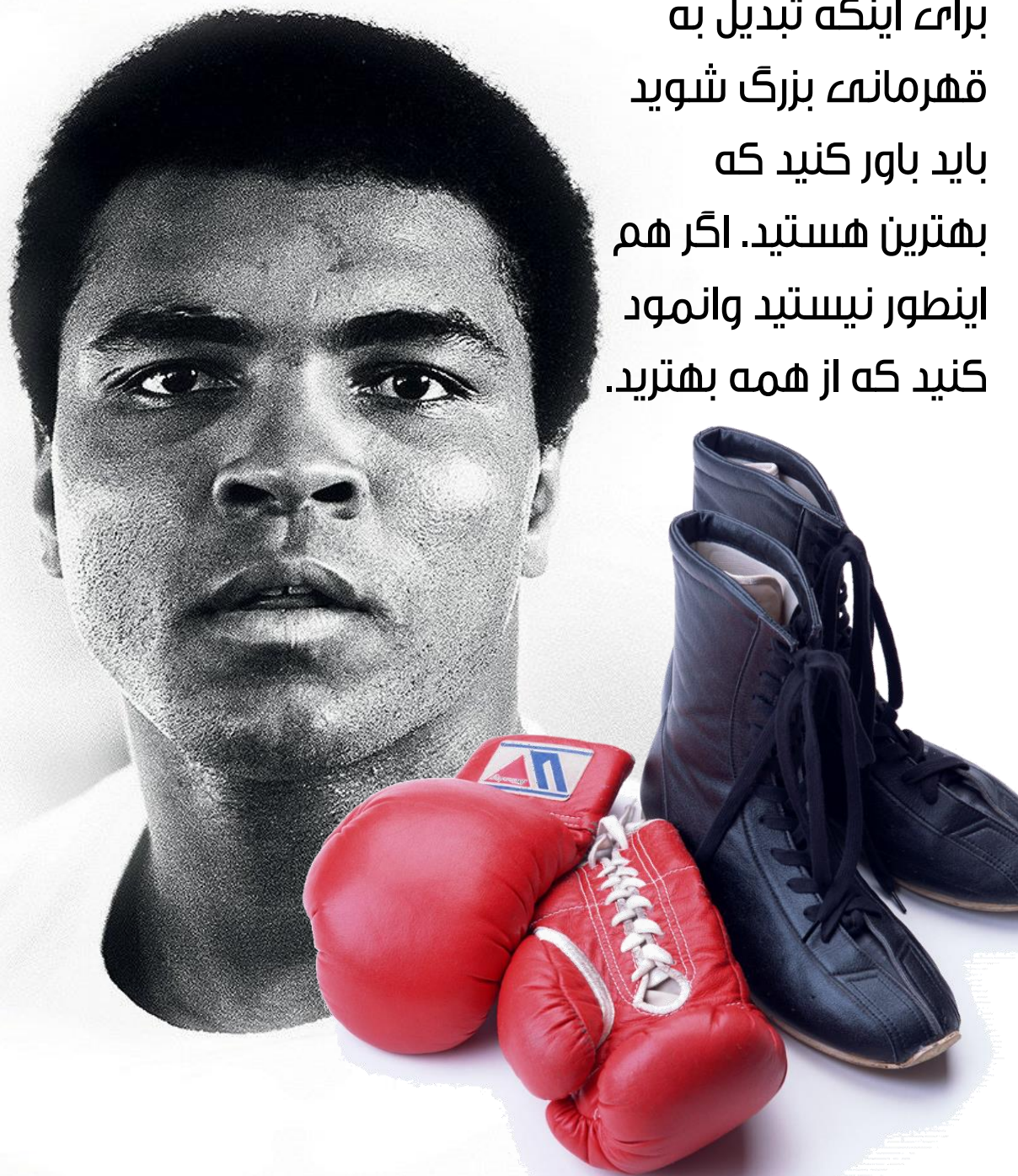


ابوالفضل خواجه دیزجی

Muhammad Ali Clay

(1942-2016)

برای اینکه تبدیل به
قهرمان بزرگ شوید
باید باور کنید که
بهترین هستید. اگر هم
اینطور نیستید وانمود
کنید که از همه بهترید.



منتشر شد

برای تهیه کتاب
لوگوی مقابل را کلیک کنید

SST
Scientific
Software
Testing

ویرایش چهارم

آندریاس اسپیلنر، تیلو لینز، هانس شیبفر

مبانی آزمون نرم افزار

(مبتنی بر ISTQB)

راهنمای مطالعه برای امتحان آزمونگر مجاز

- مطابق با ISTQB
- در مقطع Foundation

ترجمه ابولفضل خواجه دیزجی

اولین مرجع رسمی ISTQB فارسی در ایران جهت اخذ مدرک CTFL و مورد تأیید

ISTQB