

تاریخ دریافت: ۹۷/۰۵/۱۶

تاریخ پذیرش: ۹۷/۰۹/۲۵

مروری بر آزمون‌های غیر کارکردی در سیستم‌عامل اندروید و ابزارهای موجود

علیرضا بیطرفان*

دانشجوی کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران
پست الکترونیکی: bitarafana@email.kntu.ac.ir

مرتضی معمر

دانشجوی کارشناسی ارشد، دانشکده فناوری اطلاعات، دانشگاه ایوانکی، تهران، ایران
پست الکترونیکی: moammer.morteza@gmail.com

مهدی شعبانی

دانشجوی کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد علوم تحقیقات، تهران، ایران
پست الکترونیکی: mahdishabani@outlook.com

چکیده

سیستم‌عامل اندروید از جنبه آزمون‌های غیرکارکردی^۱ بررسی خواهیم کرد، که با وجود تأثیرگذاری زیاد در کیفیت محصول، مقالات کمی در گذشته به آن پرداخته‌اند. در انتهای مقاله نیز به معرفی برخی از پرستفاده‌ترین ابزارهای موجود برای هر آزمون می‌پردازیم و مزایا و معایب آن‌ها را بیان می‌کنیم.

واژه‌های کلیدی: آزمون نرم‌افزار، آزمون موبایل، آزمون رفتاری، ابزار آزمون اندروید

۱- مقدمه

در دنیای امروز با پیشرفت فناوری و همچنین با افزایش سرعت زندگی افراد جامعه، هرروز کاربران بیشتری به سمت برنامه‌های موبایل جلب می‌شوند و انتظار دارند کارهای روزمره خود مانند خرید لباس، کرایه اتومبیل، شستشوی لباس و حتی رزرو هتل را از طریق دستگاه‌های موبایل هوشمند خود انجام دهند. بر اساس

با توجه به افزایش روزافزون استفاده از گوشی‌های هوشمند، هرروز شرکت‌ها و افراد بیشتری به سمت تولید برنامه‌های موبایلی حرکت می‌کنند تا محصولات آن‌ها از کاربران بیشتری برخوردار باشد. اما توسعه یک برنامه علاوه بر مزایای غیر قابل انکار، چالش‌هایی را نیز به همراه دارد که یکی از آن‌ها انجام صحیح و کامل آزمون‌های نرم‌افزاری خواهد بود. اگرچه این آزمون‌ها هزینه‌های ماهیانه یا سالیانه را به تولیدکننده تحمیل می‌کنند، اما باعث تسهیل فرآیند تولید و نگهداری محصول و کاهش هزینه‌های مرتبط با آن خواهد شد. علاوه بر این، عدم اجرای صحیح آزمون نرم‌افزار به‌طور مستقیم بر روی کیفیت محصول موردنظر تأثیر می‌گذارد که نتیجه آن ممکن است دریافت بازخورد منفی از سوی کاربران و آسیب دیدن اعتبار تولیدکننده یا شرکت موردنظر باشد. بنابراین در این مطالعه به‌طور اختصاصی، آزمون‌های موبایل را برای

1- Non-Functional

* نویسنده مسئول

آمار منتشر شده توسط بخش ارتباطات در ایران، حدود ۳۳ میلیون ایرانی تا انتهای سال ۸۵ از اینترنت موبایل خود استفاده می‌کرده‌اند، یا به بیانی دیگر از برنامه‌هایی که نیازمند اینترنت بوده است. این رقم در انتهای سال ۹۶ به ۵۳ میلیون نفر رسیده و هنوز در حال رشد است [۱]. همچنین وبگاه استاتیستا در آمار، روند صعودی استفاده از گوشی‌های تلفن هوشمند در دنیا را نشان داده است. در سال ۲۰۱۴ تعداد کاربران این دستگاه‌ها، ۱ میلیارد و ۵۷۰ میلیون نفر بوده است که این رقم به بیش از ۲٫۵ میلیارد در سال ۲۰۱۸ رسیده است، یعنی حدود بیش از یک سوم جمعیت جهان. همچنین پیش‌بینی شده است که در سال ۲۰۲۰ نزدیک به ۲٫۹ میلیارد نفر از این دستگاه‌ها استفاده کنند که رقمی دور از انتظار نیست [۲].

در این میان صاحبان صنعت و فناوری نیز برای به دست آوردن کاربران بیشتر به سمت تولید برنامه‌های موبایل حرکت کرده‌اند. اما تنها محصولاتی توانایی جلب نظر کاربران را خواهند داشت که کیفیت بالاتر و خدمات بهتری ارائه دهند. برای مثال در یک مطالعه تحقیقاتی در سال ۲۰۱۵، درباره تاثیر کیفیت برنامه موبایل بر روی کاربران، این نتیجه به دست آمده است که ۴۸٪ از کاربران بعد از مشاهده کندی مداوم در برنامه، اقدام به حذف یا عدم استفاده مجدد از برنامه می‌کنند. همچنین ۵۳٪ از کاربران در صورت مواجهه با خطاهای مداوم در برنامه و یا عدم پاسخگویی بموقع، دیگر از آن برنامه استفاده نخواهند کرد [۳].

از طرف دیگر انجام آزمون‌های نرم‌افزاری در حین تولید نرم‌افزار، می‌تواند از بروز مشکلات بزرگتر در انتهای کار و همچنین از زیان مالی تولیدکنندگان جلوگیری کند، اگرچه انجام دقیق آزمون نیز بدون چالش نخواهد بود. وجود افراد متخصص در زمینه آزمون و پیدا کردن بهترین ابزارها که ویژگی‌های مدنظر تولیدکننده را داشته باشد و کارایی و هزینه آن نیز مطلوب باشد از مواردی است که باید به آن توجه شود. در این رابطه، اینستا‌باگ که

یک ابزار گزارش‌خرابی برنامه‌های موبایل است، در وبگاه خود گزارش جالبی ارائه داده است [۴]. در این گزارش تحلیلی بر روی ۱۰۰ میلیون خطای موجود در برنامه‌های موبایلی در سال ۲۰۱۸ انجام شده است و نتایج مهمی مانند آمار رشد خطاهای پیدا شده در برنامه‌های موبایلی بین سال‌های ۲۰۱۲ تا ۲۰۱۸ یا میانگین زمان صرف شده برای اصلاح خطاهای موجود در برنامه، ارائه شده است که همگی نشان‌دهنده افزایش اهمیت انجام آزمون‌های موبایل پیش از انتشار محصول است. به دلیل مفصل بودن این گزارش، تنها به معرفی آن بسنده کرده‌ایم تا خواننده مقاله در صورت علاقمندی به آن رجوع کند.

در سال‌های اخیر در مورد آزمون نرم‌افزار کارهای زیادی انجام شده است که اصول آزمون نرم‌افزار و مفاهیم پایه را بیان کرده‌اند اما از سیستم‌عامل یا بن‌سازه^۲ خاصی صحبت نشده است [۱۱-۱۶، ۱۶]. همچنین به طور تخصصی در زمینه آزمون موبایل نیز مطالعاتی انجام شده است [۷-۱۰، ۱۷]. این کارها سیستم‌عامل‌های اندروید، iOS و ویندوز موبایل را مورد بررسی قرار داده‌اند و در محدود مواردی به برنامه‌های چندبن‌سازه‌ای نیز پرداخته‌اند [۶]. شایان ذکر است که کارهای نام‌برده شده، بیشتر توجه خود را به آزمون‌های کارکردی^۳ معطوف کرده و از اهمیت آزمون‌های غیرکارکردی صرف‌نظر کرده‌اند. دلیل این امر می‌تواند اهمیت بیشتر آزمون‌های کارکردی در نزد کارشناسان آزمون باشد، در صورتی که آزمون‌های غیرکارکردی نیز می‌توانند عامل تعیین‌کننده‌ای در کیفیت نرم‌افزار باشند.

بنابراین با توجه به اهمیت آزمون موبایل و با الهام گرفتن از کارهای انجام شده در این زمینه، در این مطالعه سعی کرده‌ایم تا کاری جدید در زمینه آزمون‌های موبایل ارائه کنیم. ما به‌طور تخصصی، آزمون‌های موبایل را برای سیستم‌عامل اندروید از منظر آزمون‌های غیرکارکردی بررسی خواهیم کرد، موضوعی که در کارهای پیشین

2- Platform
3- Functional

آن‌طور که شایسته است، به آن پرداخته نشده است. این مطالعه از بخش‌های مختلفی تشکیل شده است: ابتدا در بخش ۲ کارهای پیشین و مرتبط با موضوع آزمون موبایل، بخصوص سیستم‌عامل اندروید را به صورت خلاصه مرور می‌کنیم تا کمبودهای موجود در نوشتگان این حوزه را شناسایی کنیم. در بخش ۳ به جزئیات آزمون موبایل نگاهی می‌اندازیم و پس از تشریح هر کدام از آزمون‌های غیرکارکردی در بخش ۴، ابزارهای موجود و پرستفاده برای این آزمون‌ها را در بخش ۵ به طور مختصر معرفی می‌کنیم و نکات مثبت و منفی هر کدام را می‌شماریم.

۲- کارهای مرتبط

در این بخش کارهای گذشته در حوزه آزمون نرم‌افزار و به طور جزئی‌تر آزمون موبایل را بررسی می‌کنیم. در کتابی که توسط نات[۵] نوشته شده است، مقدمات آزمون موبایل و تقسیم‌بندی‌ها از منظر کارکردی و غیرکارکردی بررسی شده است و پس از بررسی مختصر هر آزمون که به صورت شخصی‌سازی شده برای موبایل و بخصوص برای سیستم‌عامل اندروید در نظر گرفته شده است، برخی ابزارها به صورت سطحی بررسی شده‌اند و پیشنهادهای مفیدی در مورد رفع مشکلات ارائه شده است که بیشتر جنبه عملی دارند تا نظری.

اما این دسته‌بندی کارکردی و غیرکارکردی آزمون‌های نرم‌افزار فارغ از محیطی است که نرم‌افزار در آن اجرا می‌شود. برای مثال امروزه سیستم‌عامل‌های رایج که در دستگاه‌های موبایل استفاده می‌شوند، شامل سیستم‌عامل iOS که توسط شرکت اپل ارائه شده است، سیستم‌عامل اندروید توسط شرکت گوگل و سیستم‌عامل بلک‌بری است. برای آزمون هر کدام از این سیستم‌عامل‌ها روش‌ها و ابزارهای متنوعی ارائه شده است. در پژوهشی که توسط کولسووس[۶] انجام شده است، آزمون‌ها و ابزارهای خودکارسازی آزمون واسط کاربری که در دسته آزمون‌های کارکردی قرار دارد، بر روی سیستم‌عامل iOS

بررسی شده است و نتایج آزمون به همراه پیشنهادهایی برای بهبود مشکلات به خواننده ارائه شده است. قابل ذکر است که در مورد آزمون‌های کارکردی کارهای زیادی انجام شده است. از سایر این کارها کتابی است که توسط میلانو[۲۶] ارائه شده است و در آن آزمون‌های اندروید از منظر کارکردی بررسی شده است مانند آزمون‌های واحدی و سیستمی و پس از بررسی ابزارهای مرتبط با آن‌ها، یک نمونه آزمون انجام شده است و راهکارهای مفیدی برای بهینه‌سازی نیز ارائه شده است. نمونه دیگر نیز تحقیقی است که بانگریوس و همکاران[۷] در پژوهش خود انجام داده‌اند. در این پژوهش درباره مقدمات آزمون نرم‌افزار و موبایل به طور مختصر بحث شده است و پس از بررسی ابزارهای آزمون کارکردی در سیستم‌عامل اندروید، با رویکرد جدید خود، آزمون‌های کارکردی معرفی شده را بر روی یک برنامه اندرویدی، که توسط خود آن‌ها تولید شده است، بررسی کرده‌اند.

اما با این‌که امروزه آزمون‌های غیرکارکردی، به دلیل اثرگذاری مستقیم بر روی نظرات کاربران نهایی، از اهمیت بالایی برخوردار هستند، هنوز در این زمینه پژوهش‌های زیادی انجام نشده است. یک نمونه از آن‌ها کاری است که تحت عنوان آموزش آزمون برنامه‌های موبایلی توسط گائو و همکاران[۸] ارائه شده است. در این مقاله نیز آزمون‌های کارکردی و غیرکارکردی در کنار هم قرار گرفته‌اند اما تمرکز اصلی بر روی آزمون‌های غیرکارکردی بوده است. معرفی مختصر آزمون‌ها و بررسی ابزارهای بیشتر و بروزتر برای آزمون موبایل برتری این مقاله نسبت به کارهای قبلی محسوب می‌شود. اما یک کار ویژه در زمینه آزمون اندروید، توسط گوث[۹] انجام شده است. او در پژوهش خود ابتدا به مفاهیم آزمون نرم‌افزار و موبایل پرداخته است و در ادامه سعی کرده است تا بررسی دقیق‌تری بر روی آزمون‌های مهم موبایل که ضامن کیفیت محصول هستند و هر دو دسته کارکردی و غیرکارکردی را شامل می‌شود، داشته باشد. در پایان برخی ابزارهای آزمون با

توجه به معیارهای مدنظر نویسنده معرفی شده‌اند و انجام آزمایش بر روی یک برنامه موبایلی با توجه به این ابزارها انجام شده است.

یکی از مقالات مهم در زمینه آزمون غیرکارکردی برنامه‌های اندرویدی توسط شهریار و همکاران [۱۰] ارائه شده است که به‌طور خاص آزمون نشت حافظه را مورد بررسی قرار داده است و یک روش جدید برای آزمون حافظه پیشنهاد شده است. برای آزمایش این روش پیشنهادی ابتدا وضعیت نشت حافظه شبیه‌سازی شده است که خود این شبیه‌سازی نیز قابل توجه است.

در کتاب‌های [۱۱، ۱۲، ۱۴، ۱۶] به‌طور گسترده و مفصل در مورد مفاهیم مهندسی و آزمون نرم‌افزار و همچنین تضمین کیفیت صحبت شده است که هم جنبه عملی و هم جنبه نظری موضوع را در نظر گرفته‌اند. در واقع این کتاب ۳ می‌تواند منابع مفیدی برای انجام آزمون اصولی باشند و فارغ از هر چارچوب و سخت‌افزار خاصی به مفهوم آزمون پرداخته‌اند.

۳- آزمون موبایل

در این بخش به بحث آزمون در برنامه‌های موبایلی می‌پردازیم و برای پرهیز از اطناب، مفاهیم پایه در مورد مهندسی نرم‌افزار و آزمون‌ها را که در منابع دیگر [۵، ۱۱، ۱۲، ۱۴، ۱۶] آمده‌اند بازگو نمی‌کنیم. آمان و افوت [۱۴] در کتاب خود تحت عنوان مقدمه‌ای بر آزمون نرم‌افزار، در یک فصل مجزا به این سوال پاسخ می‌دهند که «چرا نرم‌افزار را تحت آزمون قرار می‌دهیم؟». آن‌ها معتقد هستند که نرم‌افزارها رفتار زیرساخت‌ها و فناوری‌هایی را مشخص می‌کنند که تمدن بشر در قرن حاضر بر پایه آن‌ها بنا شده است. برای مثال فضای جهانی وب (اینترنت)، سامانه‌های مدیریت حمل‌ونقل و سیستم‌های محاسباتی پیچیده که تنها گوشه‌ای از اهمیت وجود نرم‌افزار را بیان می‌کند. در ۲ دهه اخیر صنعت نرم‌افزار بزرگ‌تر و رقابتی‌تر از قبل شده

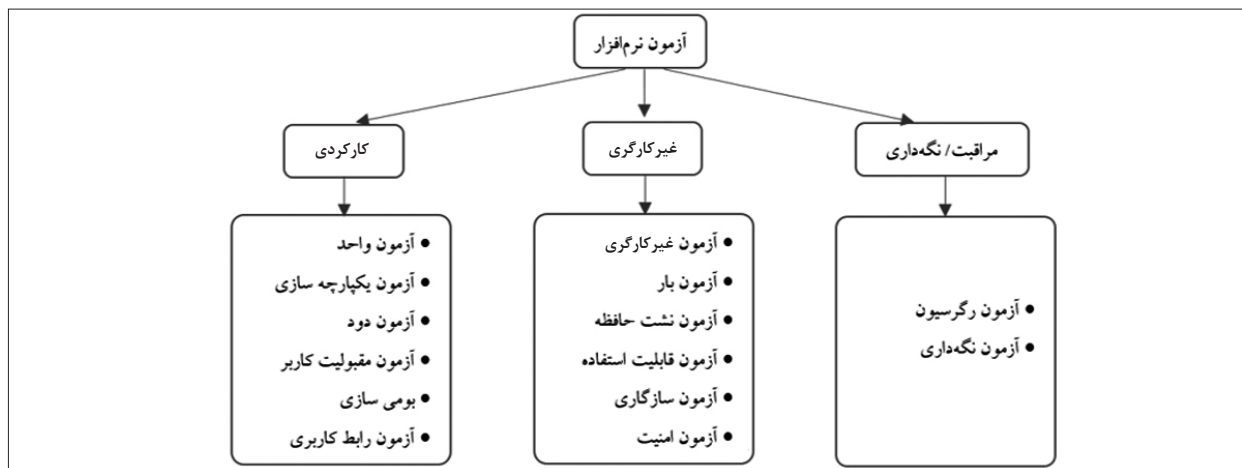
است و کاربردهای مهم‌تری مانند کنترل ترافیک هوایی نیز به آن محول شده است. نویسندگان کتاب بر این عقیده هستند که اگرچه عوامل مختلفی بر کیفیت یک نرم‌افزار تولیدشده اثرگذار خواهند بود، مانند سنجش مناسب نیاز کاربران پیش از طراحی محصول یا میزان خطای انسانی در هنگام نوشتن کد، اما این موارد به هیچ وجه جایگزین آزمون نرم‌افزار نیستند و بخش بزرگی از کیفیت نرم‌افزار مدیون یک آزمون مناسب و جامع خواهد بود [۱۴].

مطالب بالا به‌طورکلی در مورد اهمیت نرم‌افزار گفته شد، حال آن‌که با توجه به آمارهای ارائه شده در مقدمه، برنامه‌های کاربردی موبایل بخش عمده‌ای از محصولات نرم‌افزاری در سال‌های اخیر را به خود اختصاص داده‌اند. امروزه به علت سهولت استفاده از گوشی‌های موبایل و یا لوحه (تبلت)، شاهد آن هستیم که علاوه بر برنامه‌های استفاده شده در زندگی روزمره مانند فروشگاه‌اینترنتی یا درخواست تاکسی، نرم‌افزارهای حساسی مانند مدیریت فروش محصولات در کارخانه‌ها و یا سیستم‌هایی با کاربرد نظامی نیز به‌طور خاص برای این دستگاه‌ها تولید می‌شوند. بنابراین می‌توان مطالب گفته‌شده در مورد اهمیت آزمون نرم‌افزار را برای برنامه‌های کاربردی موبایل نیز در نظر گرفت. همچنین مبنای طراحی موبایل‌های هوشمند نیازهایی بوده است که رایانه‌های قابل‌حمل یا خانگی از عهده آن بر نمی‌آمدند. پس باید در نظر گرفت که کیفیت و سرعت در این محصولات برای کاربران بسیار مهم خواهد بود و برای مثال برنامه‌ای که با کندی مواجه است یا در مدیریت تعداد زیاد کاربران دچار مشکل است، هیچ‌گاه جزء انتخاب‌های اول کاربران نخواهد بود.

۳-۱- دسته‌بندی آزمون‌ها

در منابع مختلف دسته‌بندی‌های گوناگونی برای آزمون‌های موبایل و به‌طورکلی آزمون‌های نرم‌افزار ارائه شده است [۱۱، ۱۲، ۱۴، ۱۶]. اما در یک تقسیم‌بندی، که به این مطالعه مرتبط است، می‌توان مانند شکل (۱) آزمون‌ها را به ۳ بخش کارکردی، غیرکارکردی و نگه‌داری تقسیم کرد.

4- Why Do We Test Software?



شکل ۱: تقسیم‌بندی آزمون‌های نرم‌افزاری

جدول ۱: معرفی انواع مختلف برنامه‌های کاربردی موبایل

نوع برنامه	توضیح مختصر	مزایا	معایب
بومی ^۱	برای یک دستگاه یا پلتفرم خاص (مثلاً اندروید) توسعه داده می‌شود و باید بر روی آن دستگاه نصب شود.	<ul style="list-style-type: none"> قابلیت‌های متعدد سریع و متناسب با ابعاد صفحه دارای رابط کاربری خوب موجود در فروشگاه‌های برنامه 	<ul style="list-style-type: none"> نیازمند زمان بیشتر برای به‌روزرسانی نوشتن کد برای سیستم‌عامل‌های مختلف دارای هزینه‌های اضافی
تحت وب	این برنامه‌ها معمولاً توسط HTML, JS, CSS توسعه داده می‌شوند و می‌توان از طریق یک مرورگر اینترنت مانند کروم به آن‌ها دسترسی داشت.	<ul style="list-style-type: none"> نگهداری و به‌روزرسانی آسان عدم نیاز به نصب سازگار با نسخه‌های قدیمی دستگاه کد مشترک برای همه دستگاه‌ها 	<ul style="list-style-type: none"> ممکن است هزینه‌بر باشد سختی جستجوی برنامه تحت وب نگرانی‌های امنیتی محدودیت کارایی
ترکیبی ^۲	ترکیبی از ۲ مورد قبل، با استفاده از فناوری تحت وب (چند-پلتفرمی) نوشته شده‌اند ولی بر روی دستگاه نصب می‌شوند.	<ul style="list-style-type: none"> قابلیت نصب بر روی دستگاه‌های مختلف قابلیت استفاده از ویژگی‌های بومی قابلیت استفاده از بسته‌های محلی 	<ul style="list-style-type: none"> رابط کاربری نه‌چندان مطلوب رفع مشکلات به وجود آمده دشوار است به کتابخانه‌های ثالث وابسته است.

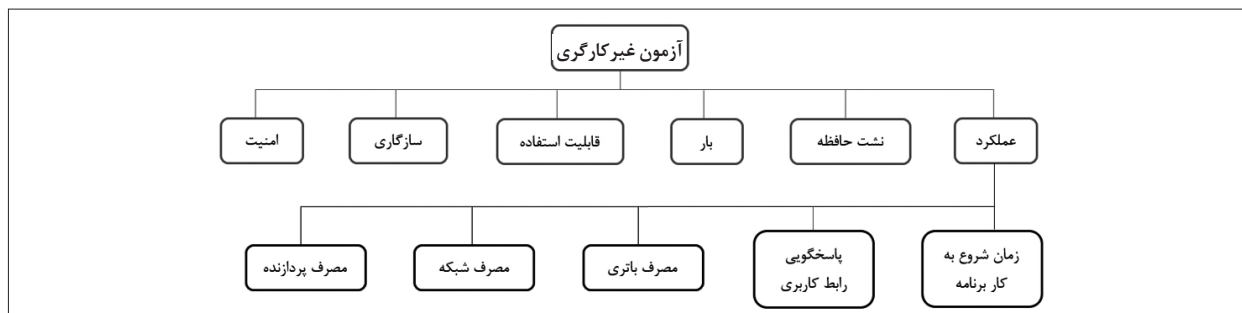
به‌درستی نشان داده نمی‌شوند) خواهد بود [۱۳، ۱۵]. در بخش ۴ آزمون‌های غیرکارکردی بررسی خواهند شد.

۳-۲- انواع برنامه‌های موبایل

از آنجایی که توسعه برنامه موبایل، بر اساس نیازمندی‌های کاربر و هدف تولیدکننده از انتشار آن، می‌تواند به روش‌های مختلف انجام شود، پیش از شروع فرآیند آزمون باید در نظر داشته باشیم که نرم‌افزار موردنظر ما چه ویژگی‌هایی دارد تا با توجه به آن بتوانیم ابزارهای مناسب آزمون را انتخاب کنیم. انواع برنامه‌های کاربردی موبایل به‌طور خلاصه در جدول (۱) ارائه شده است [۵]:

آزمون‌های نگره‌داری مانند آزمون پس‌نمایی^۵ اکثراً زمانی مورد استفاده قرار می‌گیرند که ویژگی جدیدی به سیستم اضافه شده و یا یک ویژگی بدون کاربرد از آن حذف شده است، بنابراین باید دوباره بررسی کرد که آزمون‌های قبلی به‌درستی اجرا می‌شوند یا خیر [۱۶]. این دسته‌بندی بر اساس خطاهایی که در دستگاه به وجود می‌آید به دو صورت خطاهای غیرکارکردی (مانند نشت حافظه که ممکن است باعث خطای رایج عدم پاسخگویی برنامه^۶ شود) یا خطاهای کارکردی (که با وجود آن‌ها، برنامه اجرا می‌شود اما نه مطابق انتظار، برای مثال، عناصر گرافیکی برنامه

5- regression
6- App Not Responding



شکل ۲: دسته‌بندی آزمون‌های غیر کارگری معرفی شده در این مطالعه

۴- آزمون‌های غیر کارگری

با توجه به دسته‌بندی نشان داده شده در شکل (۱)، در این بخش به‌طور مفصل هرکدام از آزمون‌های غیر کارگری مرتبط با سیستم عامل اندروید مورد بررسی قرار می‌گیرد که پس از بررسی‌های مختلف، در این مقاله از نمودار شکل (۲) برای ادامه کار استفاده شده است. باید توجه داشت که با در نظر گرفتن ماهیت مطالعه، یعنی نرم‌افزارهای موبایلی، آزمون‌ها تغییراتی داشته‌اند. ناگفته نماند که در طول انجام پژوهش ابزارهایی که شرایط بررسی را داشته‌اند (مانند رایگان بودن یا دارای هزینه کم) بر روی برنامه کاربردی Waze^v بررسی عملی شده‌اند. این برنامه پرتعداد جهت مسیریابی در شهرها و کشورهای مختلف به‌صورت برخط قابل استفاده است.

برای جداسازی بهتر، در این بخش تنها خود آزمون‌ها بررسی می‌شوند و در انتهای مقاله اکثر ابزارهای پر استفاده برای هرکدام از آزمون‌ها به همراه معایب و مزایا، به‌صورت جدولی نشان داده می‌شود تا دید خوبی به کاربر بدهد.

۴-۱- آزمون عملکرد

در منابع مختلف از این آزمون به‌عنوان خانواده آزمون عملکرد یاد می‌شود زیرا همان‌طور که در شکل (۲) قابل مشاهده است، خود شامل چندین آزمون است و همچنین تعاریف مختلفی برای آن ارائه شده است. باید توجه داشت که آزمون بار نیز بعضاً در آزمایشگاه‌های مختلف و یا ابزارهای مختلف، زیرمجموعه آزمون

7- https://play.google.com/store/apps/details?id=com.waze&hl=en_US

عملکرد در نظر گرفته می‌شود اما در این مطالعه به دلیل اهمیت آزمون بار، به‌صورت جداگانه به آن پرداخته شده است. در یک گردهمایی مربوط به کارشناسان آزمون موبایل [۲۱]، برخی مشکلات رایج در زمینه عملکرد برنامه‌های موبایل نام برده می‌شود که شامل این موارد می‌شود: (۱) خطای عدم پاسخگویی برنامه^۸، (۲) خطای مصرف بیش از حد حافظه^۹، (۳) جابجایی آهسته صفحات، لرزیدن صفحه و از دست رفتن قاب‌های موجود در صفحه. (۴) مصرف بیش از حد باتری دستگاه.

درباره مورد ۴ از جدول (۲) باید توجه شود که هنگام اتصال به اینترنت دو موردی که باعث محدودیت عملکردی می‌شود، پهنای باند^{۱۰} (اندازه یا اندازه مسیر^{۱۱}) و تأخیر^{۱۲} (طول مسیر یا میزان شلوغ بودن آن) هستند. اتصال بیسیم یک اتصال با بازده بالا^{۱۳} و تأخیر کم است و هزینه اضافی برای استفاده از این شبکه‌ها وجود ندارد. در مقابل اینترنت داده، در حالت فعال، مصرف انرژی بیشتری نسبت به بیسیم خواهد داشت. همچنین زمانی که طول می‌کشد تا یک فایل بارگیری شود در این دو اتصال متفاوت خواهد بود. برای مثال سرعت بالاتر اینترنت نسل ۴ باعث می‌شود دستگاه زمان کمتری در حالت متصل باقی بماند پس انرژی کمتری مصرف خواهد شد [۲۰].

8- App Not Responding
9- Out Of Memory
10- Bandwidth
11- Pipe
12- Latency
13- High Throughput

جدول ۲: خلاصه مجموعه آزمون‌های عملکرد

ردیف	نام آزمون	شرح	اهمیت
۱	زمان شروع به کار برنامه ^۲	از زمان انتخاب برنامه مورد نظر توسط کاربر، چه مقدار زمانی طول می‌کشد تا برنامه به کاربر نشان داده شود. ۲ نوع سرد و گرم دارد از این نظر که برنامه قبلاً اجرا شده است یا خیر [۶].	بر اساس مطالعه نظر کاربران [۱۸] این زمان نباید به طور میانگین بیش از ۲ ثانیه طول بکشد و ۸۰ درصد افراد انتظار دارند این زمان ۳ ثانیه یا کمتر باشد. در غیر این صورت خط پیشرفت ^۳ نیاز است.
۲	پاسخگویی رابط کاربری ^۵	زمان لازم برای نشان دادن صفحات مختلف به کاربر است که می‌تواند در کد، توسط ابزارهای خودکار سازی رابط کاربری و یا توسط ابزارهای مخصوص این کار اندازه‌گیری شود [۵].	از آنجایی که کاربران همواره در حال عجله هستند و نمی‌خواهند برای برنامه‌های کند، زمان تلف کنند، برنامه باید بدون تأخیر، به حرکت کاربر واکنش نشان دهد.
۳	مصرف باتری	به عوامل مختلفی از جمله مصرف پردازنده و شبکه دستگاه بستگی دارد. برای مثال اگر برنامه به اینترنت نیاز داشته باشد، میزان استفاده از شبکه و داده‌های ردوبدل شده مهم خواهد بود [۵]. البته استفاده از دوربین و فشار بیش از حد به صفحه نمایش نیز مهم است [۱۹].	بررسی عملکرد صحیح برنامه در زمانی که باتری دستگاه کمتر از ۲۰ درصد است و یا دستگاه در وضعیت مصرف بهینه باتری ۶ قرار دارد. مصرف بهینه باتری باعث بهبود تجربه کاربری می‌شود.
۴	استفاده بهینه از شبکه دستگاه	در این آزمون بررسی می‌شود که آیا مصرف باتری دستگاه به صورت بهینه انجام می‌شود یا خیر. برای مثال استفاده از فشرده‌سازی در هنگام ارسال درخواست و دریافت پاسخ یا ذخیره محلی داده‌های پرکاربرد.	استفاده بهینه از شبکه دستگاه، علاوه بر کاهش مصرف باتری و افزایش سرعت اجرای برنامه، از هزینه‌های اینترنت مصرفی کاربر نیز می‌کاهد.
۵	مصرف پردازنده	عامل اصلی مصرف باتری در دستگاه موبایل عملیات انجام شده توسط پردازنده است و بستگی دارد به این که برای مثال روال‌های مختلف برنامه مانند رابط کاربری یا روال اصلی چگونه مدیریت شده‌اند و چه باری را تحمل می‌کنند [۲۰].	مزایای مصرف بهینه پردازنده، فراهم کردن تجربه کاربری سریع تر و روان تر و ذخیره باتری است. همچنین می‌تواند به عملکرد بهتر برنامه در دستگاه‌های با نسخه سیستم عامل قدیمی تر و یا جدیدتر نیز کمک کند.

۴-۲-۴ آزمون نشت حافظه^{۱۴}

این آزمون همچنین با نام آزمون مدیریت حافظه یا عملکرد حافظه نیز شناخته می‌شود. در ادامه بررسی کلی در مورد حافظه انجام می‌دهیم تا این بخش بهتر شناخته شود و اهمیت آن مورد توجه واقع شود. در ادامه منظور از حافظه، حافظه اصلی دستگاه است.

انواع حافظه مصرفی توسط سیستم عامل اندروید در شکل (۳) معرفی شده است [۱۱]. در حال حاضر، به این دلیل که بیشتر دستگاه‌های موبایل هنوز از نحوه اجرای دلویک^{۱۵} استفاده می‌کنند، رایج‌ترین نوع حافظه استفاده شده، حافظه اختصاصی و شلوغ است (حافظه فقط توسط یک برنامه استفاده می‌شود و داده‌های برنامه تنها در حافظه اصلی ذخیره می‌شود).

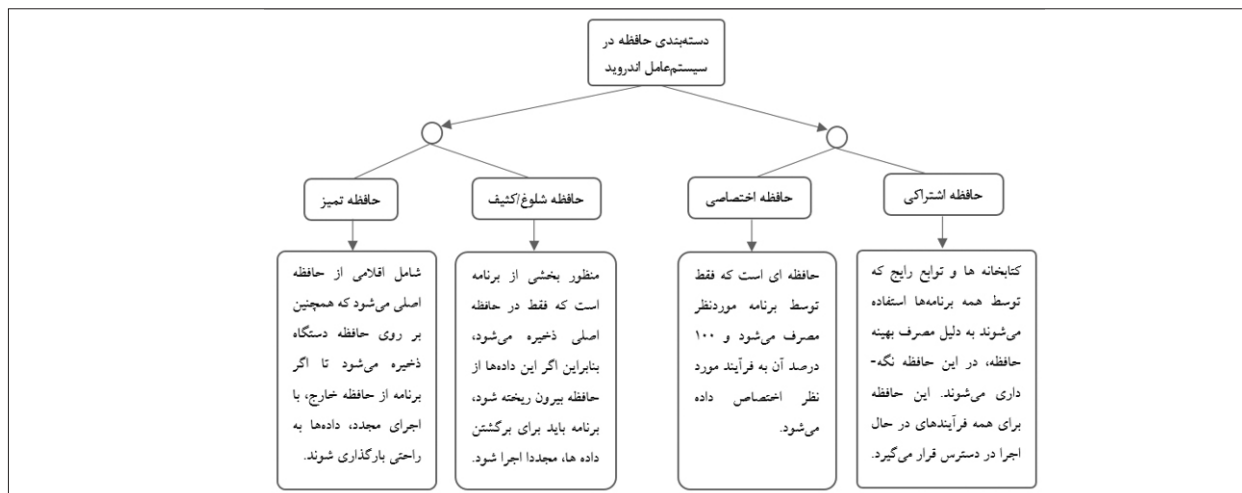
وقتی به مسئله مدیریت حافظه می‌رسیم، اولین قانون این است که باید سعی کنیم مقدار اطلاعات ذخیره شده

در حافظه اصلی کمینه شود، در واقع باید کد برنامه برای دستگاه‌هایی با مقدار حافظه کم نیز بهینه‌سازی شود (در صورتی که این دستگاه‌ها نیز هدف برنامه شما هستند). با کاهش استفاده برنامه از حافظه اصلی و مدیریت صحیح اشیاء ساخته شده توسط برنامه، احتمال رخ دادن خطاهای مرتبط کاهش می‌یابد و همچنین تعداد اشیاء کمتری باید از حافظه حذف شوند که باعث سریع‌تر شدن عملیات جمع‌آوری زباله^{۱۶} می‌شود [۲۰].

اگرچه سیستم عامل اندروید، مانند هر سیستم عامل دیگری، در زمان اجرای برنامه‌ها حافظه را مدیریت می‌کند، اما برنامه‌نویسان باید حافظه را به صورت بهینه و مدیریت شده مصرف کنند. نشت حافظه در برنامه‌های اندرویدی زمانی محتمل است که لازم است اشیاء در حافظه باقی بمانند. این اتفاق می‌تواند به خاطر اشاره‌های تصادفی یا پیوندهای دیگر بین فعالیت‌هایی باشد که از جمع‌آوری زباله جلوگیری می‌کنند. این اشاره‌های تصادفی می‌تواند به

16- Garbage Collection

14- Memory Leakage Testing
15- Dalvik



شکل ۳: بررسی حافظه سیستم عامل اندروید به طور مختصر

را حتی زمانی که در پس زمینه در حال اجراست، در اختیار داشته باشد و آزاد نکند. این رفتار می تواند جمع آوری زباله غیرضروری را به همراه داشته باشد که عملکرد حافظه سایر قسمت ها را نیز پایین می آورد. در نهایت سیستم مجبور می شود برای بازپس گیری حافظه، فرآیند برنامه شما را از بین ببرد. سپس وقتی که کاربر به برنامه شما باز می گردد باید دوباره از ابتدا شروع به کار کند که ناخوشایند خواهد بود [۲۰].

۴-۳- آزمون بار

یکی از مهم ترین آزمون های رفتاری است و در صورتی که برنامه موردنظر برای سرویس دهی به اینترنت نیاز داشته باشد (با یک کارساز ارتباط داشته باشد)، اطمینان حاصل می کند سیستم تحت بار تحمیل شده به آن، مدت زیادی (بسته به هدف نرم افزار و تولیدکننده) بدون خرابی باقی بماند و به کار خود ادامه دهد. باید توجه داشت که ممکن است زمانی که تعداد کمی کارشناس آزمون به صورت محتاطانه در حال انجام آزمون بار هستند، سیستم بدون نقص کار کند. با این حال اگر تعداد بسیار زیادی کاربر به سیستم متصل شوند که برای مثال هر کدام در زمان کار با سیستم از دستگاه های موبایل مختلف یا از تنظیمات متفاوتی استفاده می کنند، رخ دادن برخی از مشکلات زیر محتمل است: (۱) سیستم کند شود.

مشکلات کمبود حافظه در دستگاه هایی با حافظه کم منجر شود، بنابراین ضروری است که پیش از انتشار نرم افزار حل شوند. اگر شما مشکلات مرتبط با حافظه را در برنامه خود مشاهده می کنید (یا نتایج دور از انتظاری را توسط ابزارهای رصد حافظه مشاهده می کنید)، ممکن است مشکل نشت حافظه وجود داشته باشد [۱۰، ۲۰].

توجه: اندروید بخشی به نام «فضای حافظه مدیریت شده» فراهم می کند که وقتی تشخیص می دهد برنامه موردنظر دیگر از برخی اشیاء ساخته شده استفاده نمی کند، جمع کننده زباله این بخش از حافظه اشغال شده را به هر ۱۷ باز می گرداند. این که اندروید چگونه حافظه بدون استفاده را پیدا می کند، همواره در حال بهبود است اما در یک نقطه ای، در همه نسخه های اندروید، سیستم عامل باید به طور موقت کد شما را متوقف کند. در اکثر اوقات این توقف ها از سوی کاربر فهمیده نمی شوند (نامحسوس هستند). اما در صورتی که اختصاص حافظه^{۱۸} توسط برنامه شما سریع تر از جمع آوری حافظه توسط سیستم عامل باشد، ممکن است برنامه شما با تأخیر مواجه شود تا سیستم عامل به اندازه کافی حافظه را خالی کند، بنابراین باید مشکل پیدا و برطرف شود.

حتی اگر کندی برنامه شما محسوس نباشد، در حالتی که برنامه نشت حافظه دارد، ممکن است مقداری از حافظه

17- Heap
18- Memory allocation

(۲) سیستم با مشکلات عملکردی مواجه شود (در بخش ۳-۱ توضیح داده شد). (۳) عدم پاسخگویی به برخی از درخواستها. (۴) از دسترس خارج شدن سیستم (عدم سرویس دهی) [۵، ۱۱].

آزمون بار عمدتاً برای برنامه‌های نوع کارخواه/ کارساز^{۱۹} یا برنامه‌های تحت وب (کاربر از طریق مرورگر برنامه را اجرا می‌کند) مانند یک فروشگاه اینترنتی انجام می‌شود و برای هر دو مدل درون‌نت^{۲۰} یا جهانی قابلیت اجرا دارد. در واقع منظور از بار تحمیل شده به سیستم، استفاده همزمان تعداد زیادی کاربر از آن است [۲۲].

برخلاف سایر آزمون‌هایی که تا این بخش بررسی شد، آزمون بار بدون استفاده از ابزار خودکار آزمون قابل انجام نیست. معمولاً برای انجام این آزمون از کاربران مجازی استفاده می‌شود، به این معنی که با استفاده از ابزارهای موجود (مراجعه شود به بخش ۵) کاربران شبیه‌سازی می‌شوند، تا اعتبارسنجی شود که آیا سیستم می‌تواند بار پیش‌بینی شده را پشتیبانی کند یا خیر. اما باید توجه داشت که بار همیشه توسط تعداد زیادی کاربر شبیه‌سازی نمی‌شود و بسته به کاربرد برنامه موردنظر، ممکن است منظور از بار تحمیل شده به سیستم، بارگذاری یا بارگیری یک فایل با حجم بالا باشد.

هدف از انجام آزمون بار بررسی موارد زیر است تا مشخص شود که برای بهبود عملکرد، آیا نیاز است که تنها پیکربندی سیستم تغییر کند و یا باید تغییری در نرم‌افزار یا سخت‌افزار نیز به وجود بیاید [۵، ۲۲]:

- زمان پاسخگویی به هر تراکنش (درخواست و پاسخ)
- عملکرد اجزای سیستم زیر مقدار فشارهای مختلف (معادل شرایط واقعی)
- عملکرد پایگاه داده
- تأخیر شبکه و نرخ بازده^{۲۱} (توان عملیاتی)
- موارد مرتبط با طراحی نرم‌افزار (مانند بهینه بودن کد یا عدم وجود خطاهای منطقی)

• محدودیت‌های سخت‌افزاری مانند پردازنده، حافظه و گلوگاه‌های شبکه^{۲۲}

توجه) آزمون بار و فشار^{۲۳} مفاهیم تقریباً مشابهی را القاء می‌کنند با این تفاوت که در آزمون بار می‌خواهیم اطمینان پیدا کنیم که سیستم موردنظر می‌تواند برای چندین ماه در مقیاس بزرگ (با تعداد کاربران زیاد) به درستی کار کند. در حالی که هدف از آزمون فشار تحمیل بار بیش از حد به سیستم است تا خرابی رخ دهد و در این نقطه، مشاهده کنیم که در چه قسمت‌هایی خرابی رخ داده و علت آن چه بوده است.

به‌عنوان یک نمونه، در سال‌های اخیر با رشد چشم‌گیر کاربران اینترنتی شاهد قطع سرویس‌دهی از سوی وبگاه‌های خرید اینترنتی داخلی و خارجی بوده‌ایم که در مقطعی از سال بعد از تبلیغات فراوان با هجوم کاربران روبرو بوده‌اند. دلیل این امر می‌تواند عدم اجرای کامل آزمون بار یا عدم پیش‌بینی درست تعداد کاربران احتمالی باشد. از طرفی برخی از وبگاه‌های اینترنتی در زمان افزایش ترافیک، با تأخیر سرویس‌دهی مواجه می‌شوند و این نکته زمانی حائز اهمیت است که می‌فهمیم کاربران بیشتر از ۸ ثانیه منتظر نخواهند بود.

۴-۴- آزمون قابلیت استفاده^{۲۴}

این آزمون با نام سهولت استفاده نیز شناخته می‌شود. به‌طور خاص آزمون قابلیت استفاده در دستگاه‌های موبایل یک موضوع پیچیده و بسیار مهم است. در گروه تولید نرم‌افزار همواره باید یک متخصص این آزمون وجود داشته باشد تا در فرآیند تولید نرم‌افزار، این موضوع را پوشش دهد. این مسئله بسیار مهمی است که برنامه نوشته شده، سهولت استفاده داشته باشد، در غیر این صورت ممکن است با نظرات منفی از سوی کاربران روبرو شده که باعث از دست رفتن کاربران می‌شود. یکی از اصطلاحاتی که به‌صورت رایج در مبحث تولید نرم‌افزار

22- Network Bottlenecks

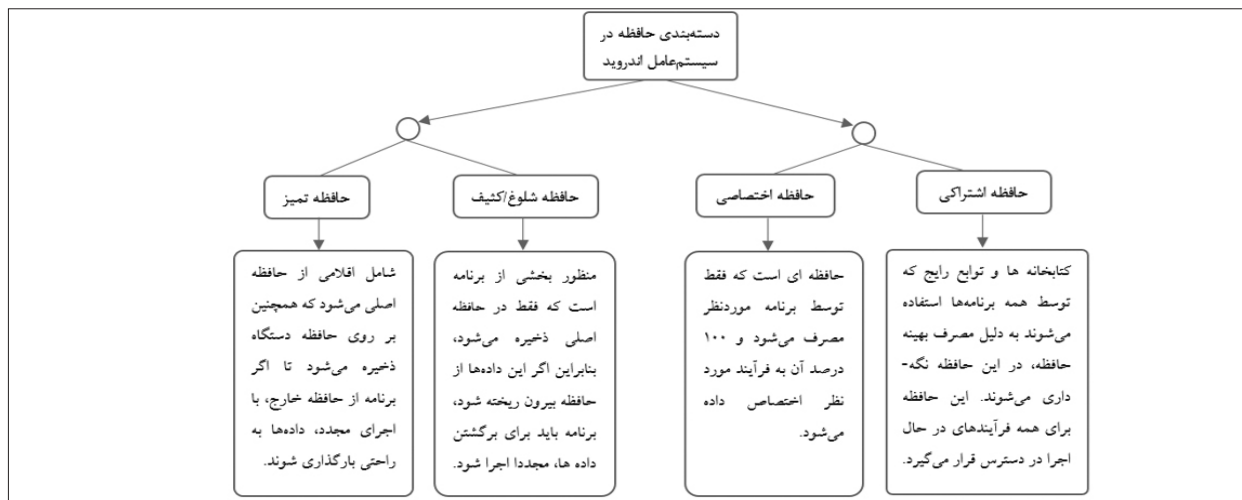
23- Stress Testing

24- Usability Testing

19- Client/Server

20- Intranet

21- Throughput rate



شکل ۴: جنبه‌های آزمون قابلیت استفاده بر اساس [۵]

وال‌مارت^{۲۷} با کاهش زمان بالا آمدن صفحه اصلی وب‌گاه خود از ۷ ثانیه به ۳ ثانیه، به ازای هر ثانیه کاهش زمان، ۲ درصد افزایش نرخ خرید توسط کاربران را تجربه کرده است که نشان از اهمیت بالای این موضوع دارد [۲۳]. در شکل (۴) جنبه‌های مختلف آزمون قابلیت استفاده به‌طور خلاصه بیان شده است. همچنین موارد گفته‌شده در جدول (۳) باید در حین آزمون قابلیت استفاده بررسی شوند تا بتوان گفت برنامه موردنظر از لحاظ تجربه کاربری کیفیت موردنظر را داراست [۱۱، ۲۴].

شایان‌ذکر است که این آزمون به دور روش آزمایشگاهی (با استفاده از ابزار خاص و در حضور مراقبان) و روش از راه دور (دسترسی به آزمونگر از طریق وب‌کم و ضبط صدا و تصویر او) انجام می‌شود. یک نمونه از این ابزارها نیز، Silver Back3^{۲۸} است.

۴-۵- آزمون سازگاری^{۲۹}

آزمون سازگاری یکی از آزمون‌های حیاتی در تضمین کیفیت نرم‌افزار است که تضمین می‌کند نرم‌افزار موردنظر با مجموعه گسترده‌ای از چارچوب‌ها، تنظیمات و سخت‌افزارها سازگار است و می‌تواند به‌صورت موازی در کنار آن‌ها کار کند [۵]. در واقع گسترده‌گی مرورگرهای وب، سیستم‌عامل‌ها و نسخه‌های آن‌ها شرکت‌ها را مجاب

27- Walmart
28- <https://silverbackapp.com/>
29- Compatibility Testing

استفاده می‌شود همواره بیان می‌کند که برنامه باید همیشه با سرعت مناسب، ساده و با قابلیت استفاده بالا به کاربران عرضه شود که مخفف آن در زبان انگلیسی به‌صورت KIFSU^{۳۰} بیان می‌شود [۵].

شرکت گوگل در یکی از مقالات خود در سال ۲۰۱۵ به‌وضوح بیان می‌کند که از مهم‌ترین عوامل بهبود تجربه کاربری^{۳۱} در برنامه‌های موبایل و برنامه‌های تحت وب، افزایش سرعت و بهینه‌سازی برنامه است [۲۳]: «صبر و حوصله افراد برای یک برنامه یا وب‌گاه کند و دارای مکث در حال از بین رفتن است. آن‌ها انتظار دارند تا داخل یک برنامه موبایل با سرعت نور حرکت کنند.»

سه راه‌حل نام برده شده در این مقاله عبارت‌اند از [۲۳]:
۱. حذف مراحل: برای مثال ذخیره کردن یک شماره تلفن جدید نیازمند چندین مرحله نباشد و کار با یک کلیک انجام شود.

۲. پیش‌بینی نیازهای کاربر: برای مثال عملیاتی که کاربر در هنگام باز کردن برنامه نیاز خواهد داشت در فهرست اصلی قرار گیرد.

۳. زمان بالا آمدن برنامه: ۴۰ درصد از خریداران اینترنتی بیش از ۳ ثانیه برای مشاهده کالای موردنظر خود صبر نمی‌کنند.

همچنین در مطالعه انجام‌شده در سال ۲۰۱۴ شرکت

25- Keep It Fast, Simple and Usable
26- UX(User Experience)

جدول ۳: موارد آزمون قابلیت استفاده

ردیف	مورد	شرح	توضیحات
۱	حذف موارد غیر ضروری	<ul style="list-style-type: none"> • نباید عناصر بدون استفاده در برنامه وجود داشته باشند. • باید بدانیم هدف از طراحی برنامه رفع چه مشکلی بوده است. • باید بر روی متن‌ها و کلمات نیز انجام شود، در واقع باید تمرکز بر روی محتوای اصلی باشد نه موارد فرعی و بیهوده. 	موارد اضافی در طراحی (چه متن و چه اشکال و دکمه‌ها) باعث پرت شدن حواس کاربر می‌شود و باید از بین بروند تا موضوع و هدف اصلی جلب توجه کند.
۲	عدم نیاز به توضیحات	<ul style="list-style-type: none"> • آیا برنامه نیاز به توضیحات دارد؟ • برای مثال آیا کاربر می‌تواند اجزای مورد نیاز خود را ببیند؟ • آیا هر عنصر موجود بر روی صفحه واضح و قابل فهم است؟ 	بررسی این مورد در طول زمان سخت‌تر می‌شود زیرا در طول زمان، کار کردن با اجزای غیر قابل فهم برای شما عادی‌تر می‌شود. بنابراین باید فردی که در فرآیند توسعه برنامه حضور نداشته آزمون را انجام دهند.
۳	توجه به الگوها	<ul style="list-style-type: none"> • عناصری که از یک نوع هستند مانند دکمه‌ها یا برجسب‌های متنی باید میزان فاصله، اندازه و رنگ یکسان داشته باشند. • آیا همه اجزاء قابل دسترسی هستند؟ برای مثال یک دکمه خاص فقط می‌تواند توسط فردی با انگشتان چاق زده شود. 	برنامه باید از یک الگوی طراحی و قابلیت استفاده پیروی کند و باید بررسی شود که الگوهای پیش فرض که توسط بن‌سازه مورد نظر (در اینجا اندروید) فراهم شده است، توسط برنامه رعایت شده باشد.
۴	اجازه انجام اشتباه	<ul style="list-style-type: none"> • در برنامه با قابلیت استفاده بالا، کاربران اجازه اشتباه کردن دارند. • برای مثال، اگر کاربر یکی از ورودی‌های فرم را خالی بگذارد، به صورت دوستانه و مفید یک پیغام خطا ببیند. • همچنین خوب است در زمان تایپ کردن، پیشنهادهایی به کاربر داده شود. برای مثال زمانی که کاربر «ای» را تایپ می‌کند، کلمه «ایران» به او پیشنهاد داده شود. 	برای نمایش پیغام‌ها یا پیشنهادهای باید گروه کاربران هدف را در نظر داشت. برای مثال اگر کاربران هدف مدیران سازمان‌ها هستند نحوه خطاب کردن با دانشجویان متفاوت است.
۵	بررسی جریان کار و جزئیات	<ul style="list-style-type: none"> • اگر برنامه از یک جریان کار یا فرآیند خاص پیروی می‌کند، بررسی کنید که در هر بخش از همان جریان کار مشابه پیروی شود. • جزئیات باید بررسی شوند برای مثال دکمه‌ها یا برجسب‌ها به اندازه کافی بزرگ یا کوچک هستند که استفاده شوند. • آیا برخی از این گام‌ها قابل ادغام شدن هستند؟ • آیا حرکت در برنامه توسط یک دست فراهم است؟ 	تقسیم صفحه مسئله مهمی است، بنابراین برنامه را بر روی صفحه‌های و دستگاه‌های مختلف بررسی کنید. جزئیات مهم نیز باید بررسی شوند که آیا کاربر می‌تواند آن‌ها را پیدا کند یا خیر. این‌ها جزئیات مهمی هستند که به‌طور مستقیم بر روی نظرات کاربران تأثیر می‌گذارد.
۶	بررسی جمله‌بندی‌ها	<ul style="list-style-type: none"> • متن‌های استفاده شده در برنامه باید واضح و قابل فهم باشد. • در متن نباید غلط‌های املائی وجود داشته باشد زیرا این امر می‌تواند تأثیر زیادی بر روی نظرات در مورد برنامه داشته باشد. 	برای مثال از افراد مختلف پرسید که آیا متوجه جمله‌بندی‌ها و ویژگی پشت آن می‌شوند؟

توجه) آزمون سازگاری می‌تواند به‌صورت آزمون روبه‌جلو^{۳۰} (که بررسی می‌کند محصول مورد نظر با بن‌سازه‌های به‌روزتر و جدیدتر به‌خوبی کار کند) و یا آزمون برعکس^{۳۱} (که بررسی می‌کند محصولی که با نسخه حال حاضر یک محیط یا بن‌سازه خاص طراحی شده است، با نسخه‌های قدیمی‌تر آن نیز به‌خوبی کار کند) انجام شود [۱۲].

با توجه به مفاهیم آزمون نرم‌افزار، می‌توان آزمون سازگاری را به‌عنوان یک مفهوم عمومی‌تر نسبت به آزمون‌های زیر در نظر گرفت:

کرده است تا بررسی کنند که آیا محصولات آن‌ها بر روی همه دستگاه‌ها و سیستم‌عامل‌های موجود به‌صورت برنامه‌ریزی شده عمل می‌کنند یا خیر.

اگر در مورد آزمون ایده‌آل صحبت کنیم، یک برنامه باید بر روی همه بن‌سازه‌های موجود به‌صورت دستی بررسی شود. اما در این راه محدودیت‌هایی وجود دارد، مانند زمان، هزینه و نیاز به زیرساخت. بنابراین اغلب ابزارها و دستگاه‌های مختلف برای آزمون برنامه یا وبگاه مورد استفاده قرار می‌گیرند و در این مورد ابزارهای مبتنی بر فضای ابری بهترین کارایی را خواهند داشت (با توجه به محدودیت‌های گفته شده).

30- Forward Compatibility Testing
31- Reverse Compatibility Testing

جدول ۴: ۱۰ مورد مهم آسیب پذیری برنامه های موبایل

ردیف	مورد	شرح
۱	استفاده نامناسب از بن سازه	شامل استفاده نادرست از ویژگی ها یا اشتباه در استفاده از موارد امنیتی موجود در بن سازه مورد استفاده.
۲	ذخیره نامن داده ها	ترکیبی از ۲ مورد در لیست سال ۲۰۱۴ است: ذخیره نامن داده ها و نشت داده به صورت ناخواسته.
۳	ارتباطات نامن	مواردی مانند دسته دهی ۹ ضعیف، نسخه های نادرست SSL، ارتباطات حساس با متن قابل خواندن.
۴	احراز هویت نامن	<ul style="list-style-type: none"> اصلاً توانایی تشخیص کاربر را نداشته باشد، زمانی که احتیاج است. خطا در نگه داری از اطلاعات برای تشخیص کاربر، زمانی که نیاز است. ضعف در مدیریت نشست کاربر.
۵	رمزنگاری نامناسب	زمانی که رمزنگاری انجام شده اما نه به طور صحیح و یا همه موارد پوشش داده نشده است. توجه شود که هر چیزی مرتبط با TLS, SSL, (ارتباطات) به مورد ۳ و عدم استفاده از رمزنگاری نیز به مورد ۲ ارتباط دارد.
۶	صدور مجوز به صورت نامن	<ul style="list-style-type: none"> بررسی همه مشکلات صدور مجوز (برای مثال تصمیمات مجوز در سمت کاربر). اگر برنامه به طور کلی احراز هویت کاربر را انجام ندهد (در موقعیتی که باید این کار انجام شود) مانند دادن دسترسی ناشناس به برخی از منابع یا سرویس ها زمانی که نباید انجام شود، مربوط به مورد ۴ خواهد بود.
۷	کیفیت پایین کد سمت کاربر	<ul style="list-style-type: none"> این مورد قبلاً با نام «تصمیمات امنیتی از طریق ورودی های نامطمئن» شناخته می شد، یکی از بخش هایی که کمتر استفاده می شده است. یک مورد فراگیر برای مشکلات پیاده سازی در سطح کد سمت کاربر است و با کد سمت سرور متفاوت است. این مورد شامل سرریز شدن میانگیرها، آسیب پذیری رشته ها ۱۰ و سایر اشتباهات سطح کد می شود که راه حل آن باز نویسی برخی از کدهای در حال اجرا است.
۸	تغییر دادن کد	<ul style="list-style-type: none"> این مورد شامل رمزنگاری دودویی، تغییر دادن منابع محلی، تغییر قسمت های مهم کد و تغییر حافظه پویا می شود. هنگامی که برنامه نصب می شود، کد و منابع داده در آنجا مستقر می شوند و رخنه گر می تواند به صورت مستقیم کد را تغییر دهد، محتوای حافظه را به صورت پویا تغییر دهد، API های سیستم را که برنامه از آن ها استفاده می کند، تغییر دهد یا جایگزین کند و یا داده ها و منابع برنامه را تغییر دهد. این قضیه یک روش مستقیم برای مهاجم فراهم می کند تا به صورت شخصی یا غیر شخصی سوء استفاده کند.
۹	مهندسی معکوس	<ul style="list-style-type: none"> شامل تحلیل هسته دودویی نهایی برای تشخیص کد آن، کتابخانه ها، الگوریتم ها و سایر محتویات است. برخی ابزار مانند Hopper، Otool و سایر ابزارهای رصد دودویی به مهاجم یک دید به محتویات برنامه می دهد. ممکن است برای پیدا کردن سایر آسیب پذیری های برنامه استفاده شود، همچنین برای افشای اطلاعات در مورد کارساز و موارد ثابت در رمزنگاری.
۱۰	کارایی های فرعی ۱۱	<ul style="list-style-type: none"> برنامه نویسان اغلب از مواردی به نام در پستی^{۱۱} (مخفی) یا سایر کنترل کننده های امنیتی داخلی استفاده می کنند که نباید به همراه محصول نهایی منتشر شوند. برای مثال برنامه نویسی در زمان آزمون، احراز هویت دومرحله ای را غیرفعال کرده یا به طور تصادفی گذرواژه را به صورت توضیح داخل کد قرار داده است.

شود که داده های ذخیره شده را از حافظه پاک کند، این کار را به درستی انجام دهد).

۴-۶- آزمون امنیت

آزمون امنیت در برنامه های موبایل شامل احراز هویت، امنیت داده، آسیب پذیری منجر به نفوذ و مدیریت نشست کاربران می شود [۱۲]. دلایل مختلفی برای اثبات اهمیت آزمون امنیت وجود دارد، برای مثال جلوگیری از مورد حمله واقع شدن برنامه یا جلوگیری از وارد شدن ویروس ها و بدافزارها به برنامه. بنابراین از منظر تجاری،

آزمون قابلیت حمل^{۳۲}: بررسی می کند که آیا برنامه موجود، بر روی محیط های متفاوت قابلیت اجرا دارد یا خیر و باید ۴ مورد قابلیت نصب، قابلیت همکاری با نرم افزارهای موجود در محیط جدید، قابلیت تطبیق با محیط جدید بدون تغییرات و قابلیت جایگزینی یک نرم افزار دیگر با همان هدف در محیط مشابه را داشته باشد.

آزمون قابلیت نصب^{۳۳}: این آزمون تنها بررسی می کند که برنامه مورد نظر قابلیت نصب و سپس حذف شدن را به درستی داشته باشد (برای مثال اگر از کاربر پرسیده

32- Portability Testing
33- Installation Testing

انجام آزمون امنیت بسیار حیاتی است و ارتباطی با بزرگ یا کوچک بودن نرم‌افزار و شرکت ندارد و باید برای دستگاه‌ها و نسخه‌های مختلف سیستم‌عامل انجام شود. یکی از منابع ارزشمند برای آزمون امنیت نرم‌افزارها، به نام پروژه OWASP^{۲۴}، در سال‌های اخیر بررسی برنامه‌های موبایل را نیز در دستور کار خود قرار داده است. این پروژه در سال ۲۰۱۶ تحقیقات ۹۰ روزه‌ای را بر روی برنامه‌های موبایل انجام داده است و ۱۰ مورد مهم در امنیت برنامه‌های موبایل را مشابه جدول (۴) به دست آورده است [۲۵].

۵- ابزارها

در بخش قبل هرکدام از آزمون‌های رفتاری به تفکیک معرفی شدند و اهمیت آن‌ها بیان شد. برای این‌که خواننده مقاله درک بهتری نسبت به موضوع پیدا کند و این پژوهش جنبه عملی نیز داشته باشد، در این بخش و در جدول (۵)، سعی شده است پراستفاده‌ترین ابزارهای موجود برای آزمون‌های معرفی‌شده در بخش ۴ به همراه نقاط ضعف و قوت آن‌ها بررسی شوند. شش ستون آخر به این معنا هستند که ابزار موردنظر قابلیت انجام آن آزمون را داراست.

34- Open Web Application Security Project

جدول ۵: مقایسه ابزارهای معرفی‌شده

ردیف	ابزار	مزایا	معایب	آزمون عملکرد	نشت حافظه	آزمون بار	قابلیت استفاده	سازگاری	آزمون امنیت
۱	ADB	رایگان، به دست آوردن اطلاعات مفید در مورد دستگاه، رایگان بودن	نیاز به وجود دستگاه حقیقی برای آزمون، نداشتن رابط گرافیکی	✓	✓				
۲	SeeTest	امکان گزارش‌گیری، آزمون خودکار (JUnit) و دستی، فضای ابری (عدم نیاز به نصب)، دستگاه‌های حقیقی و مجازی	هزینه‌دار بودن برای استفاده تجاری، ابزار رصد جزئیات را در اختیار کاربر نمی‌گذارد، رایگان آزمایشی	✓			✓		
۳	Battery Historian	رایگان، رابط کاربری گرافیکی، گزارش‌گیری منحصر به فرد	سختی اجرا (ساخت برنامه از روی کد)، عدم نمایش بیدرنگ منابع	✓					
۴	Video Optimizer	اختصاصی برای آزمون شبکه موبایل، توصیه‌شده توسط انجمن تضمین کیفیت، اینترنت داده و بیسیم، رایگان	پیش‌نیازهای زیاد برای نصب، نیاز به دستگاه حقیقی برای آزمون	✓					
۵	Wireshark	رایگان، رصد بیدرنگ، ابزار تحلیل قوی، رابط کاربری گرافیکی قوی، ابزار فیلتر قوی، سهولت استفاده	عدم رمزگشایی HTTPS	✓					
۶	Fiddler18	در نقش پراکسی برای داده‌های ردوبدل شده، رمزگشایی HTTPS	هزینه‌بر (بیش از ۱,۳۰۰ دلار)، امکانات کمتر نسبت به Wireshark	✓				✓	
۷	Android Studio Profiler19	رایگان، رابط کاربری گرافیکی، امکان رصد بیدرنگ منابع، تحلیل قوی، امکان رجوع به کد، دستگاه حقیقی و شبیه‌ساز	عدم امکان گزارش‌گیری برای استفاده بعدی، نیاز به سخت‌افزار قوی، نیاز به وجود کد	✓	✓				
۸	AWS Device Farm20	امکان آزمون خودکار یا از راه دور، وجود آزمون فازی پیش‌فرض، اجرا بر روی دستگاه مسیریابی نشده	تنها ۱۰۰۰ دقیقه آزمون رایگان، تحریم بودن ایران	✓				✓	

ردیف	ابزار	مزایا	معایب	آزمون عملکرد	نشت حافظه	آزمون بار	قابلیت استفاده	سازگاری	آزمون امنیت
۹	MAT (Eclipse)	رایگان، استفاده به عنوان افزونه یا مستقل، رابط کاربری گرافیکی، امکان آزمون بر روی دستگاه حقیقی و شبیه ساز	عدم امکان رصد بیدرنگ، عدم سهولت استفاده، توصیه بیشتر نسبت به استفاده از Profiler	✓	✓				
۱۰	pCloudy	آزمون خودکار و دستی، گزارش گیری، چند آزمون همزمان، فضای ابری، آزمون رایگان به مدت ۳۰ دقیقه	هزینه بر (تقریباً ۱۰۰ دلار در ماه)، جزئیات کم در مورد مصرف منابع، سرعت پایین تر نسبت به رقبا	✓			✓	✓	
۱۱	Leak Canary	رایگان، تشخیص خودکار نشت حافظه، رابط کاربری مناسب	استفاده به عنوان کتابخانه در کد، نیازمند تغییرات در کد، عدم امکان رصد بلادرنگ حافظه	✓					
۱۲	AndroBug	پیدا کردن آسیب پذیری‌ها در ۴ سطح امنیتی، سهولت استفاده، گزارش خوب، سرعت و دقت بالا، رایگان	نداشتن رابط کاربری گرافیکی، تولید گزارش به صورت فایل متنی ساده					✓	
۱۳	Drozer	خودکار سازی فعالیت‌های پیچیده، پشتیبانی از شبیه سازها، دقت، رایگان	به روز نبودن، عدم اجرا بر روی ویندوز (فقط لینوکس و مک)					✓	
۱۴	Firebase	آزمون از راه دور، دستگاه‌های حقیقی و مجازی، آزمون خودکار قوی، امکانات منحصر به فرد، ۱۵ آزمون رایگان در روز	محدودیت تعداد آزمون رایگان، استفاده از پراکسی برای دسترسی	✓				✓	
۱۵	MobSF	اجرای انواع آزمون امنیت، آزمون یویا و ثابت، تحلیل کد، به روز، راهنمای کاربری خوب، سهولت استفاده، گزارش گیری خوب، رایگان	نیاز به ساخت برنامه از روی کد، پیش نیازهای زیاد، پیدا نکردن همه آسیب پذیری‌ها					✓	
۱۶	CTS	رایگان، پشتیبانی توسط گوگل، پشتیبانی آزمون خودکار	اجرای نمونه آزمون فقط به زبان جاوا، نیاز به دستگاه حقیقی، آزمون سازگاری از منظر عملکردی	✓				✓	
۱۷	UserLytics	هزینه پایین تر نسبت به سایرین، تنها آزمون معتدل نشده	هزینه بر (بیش از ۴۹ دلار به ازای هر کاربر)، تعداد سوالات محدود				✓		
۱۸	Validity	آزمون معتدل شده و غیر معتدل، آزمون نمونه اولیه، تعیین وظایف و سوالات، ارتباط بیدرنگ	تعداد آزمون کم نسبت به هزینه (۴۰-۲۰۰ دلار در ماه)				✓		
۱۹	jMeter	رایگان، رابط کاربری گرافیکی، سهولت استفاده، گزارش گیری، امکانات مناسب، آزمون خودکار	نیاز به وجود دستگاه حقیقی برای آزمون، نیاز به نصب جاوا برای اجرا		✓				
۲۰	Blaze-Meter	فضای ابری، هزینه بر اساس نیاز، امکان آزمون از مناطق مختلف، تجمیع با ابزارها، گزارش گیری	هزینه بر (۱۰۰-۵۰۰ دلار در ماه)، تعداد آزمون محدود	✓					
۲۱	Locust	متن باز، رابط کاربری گرافیکی، آزمون خودکار، توزیع شده، رایگان	نیاز به دانستن زبان پایتون برای آزمون، نیاز به ساخت برنامه از روی کد						✓

جمع بندی

است. شرکت‌های بزرگ و کوچک نیز برای جلب مشتریان بیشتر، مایل هستند تا برای این دستگاه‌ها، بخصوص سیستم عامل اندروید به دلیل متن باز بودن و وجود برخی ویژگی‌های رایگان، نرم افزار تولید کنند. در سال‌های اخیر

در دنیای امروز، به دلیل افزایش سرعت زندگی، استفاده کاربران از گوشی‌های هوشمند و برنامه‌های تولید شده برای آن‌ها افزایش یافته است که در آمار و ارقام کاملاً مشهود

9-Göth, B.R., "Testing techniques for mobile device applications", Master's Thesis, Masaryk University, Czech Republic, 2015.

10-H. Shahriar, S. North, E. Mawangi, "Testing of Memory Leak in Android Applications", IEEE 15th International Symposium on High-Assurance Systems Engineering, 2014.

11-Naik, K and Tripathy, P. "Software Testing and Quality Assurance: Theory and Practice", Wiley, New Jersey, United States, 2008.

12-Tian, J. "software quality engineering: Testing, Quality Assurance, and Quantifiable Improvement", IEEE Computer Society, California, United States, 2005.

13-Earl T. Barr et al. "The Oracle Problem in Software Testing: A Survey". In: IEEE Transactions on Software Engineering 41.5, 2015.

14-Ammann, P., & Offutt, J., "Introduction to software testing", 2nd Edition, Cambridge University Press, United States, 2016.

15-International Software Testing Qualifications Board, "ISTQB: Foundation Level Syllabus", 2011, <http://www.istqb.org/downloads/finish/16/15.html>.

16-Craig, R. D., Jaskiel, S. P., "Systematic Software Testing", Artech House Publishing, Massachusetts, United States, 2002.

17-Macharla, Pradeep, "Android Continuous Integration: Build-Deploy-Test Automation for Android", Apress, United States, 2017.

18-Klick, Mobile Apps: What Consumers Really Need and Want, March 2013, <https://www.klick.com/health/news/blog/mhealth/mobile-apps-what-consumers-really-want/>.

19-Perfecoto, Mobile Apps, Battery Life and Why Testing Is Critical, [Accessed: June 2018], <http://blog.perfectomobile.com/mobile-application-testing/mobile-apps-battery-life-and-why-testing-is-critical/>.

20-Sillars, D., "High Performance Android Apps, Improve Ratings with Speed, Optimizations and Testing", O'reilly, California, United States, 2015.

21-DevConf, Performance Testing Tools in Android, October 2016, <https://www.youtube.com/watch?v=gk2D7A8MjDo>.

22-Zhang, P., "Automated Test Case Generation to Validate Non-functional Software Requirements", Ph.D. Thesis, University of Nebraska, Nebraska, United States, 2013.

23-Google, Speed is key, September 2015, <https://www.thinkwithgoogle.com/marketing-resources/experience-design/speed-is-key-optimize-your-mobile-experience/>

24-Software Quality Assurance, ISO 9126 Software Characteristics, <http://www.sqa.net/iso9126.html>.

25-OWASP, Top 10 List, February 2017, https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10.

26-Milano, D. T., "Android Application Testing Guide", Packt, Birmingham, United Kingdom, 2011.

مقالات و پژوهش‌های زیادی به موضوع آزمون نرم‌افزار و آزمون‌های موبایلی پرداخته‌اند. اما اگرچه آزمون‌های رفتاری در کنار آزمون‌های عملکردی تعیین‌کننده کیفیت محصول نرم‌افزاری و مخصوصاً برنامه‌های موبایل است، مقالات زیادی به این حوزه پرداخته‌اند و اکثراً در مورد آزمون‌های عملکردی بحث کرده‌اند. به همین دلیل در این مقاله به چالش آزمون‌های رفتاری در برنامه‌های مبتنی بر سیستم‌عامل اندروید پرداخته و سعی در ارائه کاری جدید کرده‌ایم تا بتوانیم به ارتقاء کیفیت محصولات تولیدشده کمک کنیم. از طرفی کارهای گذشته بیشتر به صورت نظری به آزمون‌ها نگاه کرده‌اند، در صورتی‌که این مقاله سعی در پر کردن فاصله بین پژوهش و صنعت داشته است و به همین دلیل در انتهای کار ابزارهای پرستفاده در آزمون‌های غیرکارکردی موبایل (اندروید) معرفی و مزیت و عیب هر کدام نام برده شد تا خواننده بتواند بسته به نیازها و معیارهای خود ابزارها را مقایسه کرده و مناسب‌ترین ابزار را پیدا کند. در کارهای آینده نیز در نظر داریم تا همین کار را برای آزمون‌های کارکردی و ابزارهای مربوطه انجام دهیم.

مراجع

۱- تسنیم، ضربیب نفوذ تلفن همراه در کشور، مهر ۱۳۹۷، <https://www.tasnimnews.com/fa/news/1397/07/22/1852391>.

2- Statista, Number of smartphone users worldwide, <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.

3-STL Partners, Lag kills! how app latency wrecks customer experience, November 2015, <https://stlpartners.com/research/lag-kills-how-app-latency-wrecks-customer-experience/>.

4-InStabug, State of mobile app quality 2018, <https://instabug.com/state-of-mobile-app-quality-2018>.

5-Knott, Daniel. "Hands-On Mobile App Testing: A Guide for Mobile Testers and Anyone Involved in the Mobile App Business", Addison Wesley, 2015

6- Kulesovs, I., "Mobile Application Testing", Ph.D. Thesis, University of Latvia, Latvia, 2017.

7-Bangerius, S. Fröberg, F., "Functional Testing of an android application", B.Sc. Thesis, Linköping University, Latvia, 2016.

8-J. Gao, X. Bai, W. Tsai and T. Uehara, "Mobile Application Testing: A Tutorial", in IEEE Computer Society, vol. 47, no. 2, pp. 46-55, 2014.