پنجمین کنفرانس ملی فناوری‌های نوین در مهندسی برق و کامپیوتر
۳۰ شهریور ماه ۱۴۰۱ - اصفهان
ISC پایگاه استنادی علوم جهان اسلام ۰۱۲۲۰-۹۰۱۳۵
وزارت علوم، تحقیقات و فناوری
موسسه آموزش عالی جهاد دانشگاهی استان اصفهان
جهاد دانشگاهی معاونت پژوهش و فناوری

مروری بر تبیین چالش‌های آزمون برنامه‌های موبایل

علی کریمی، استادیار، a.karimi@ihu.ac.ir[۱]
علی ایرانپور مبارکه، دانشجو کارشناسی ارشد مهندسی نرم‌افزار، Ali.iranpour@ihu.ac.ir[۲]
مهدی قزلسفلو، دانشجو کارشناسی ارشد مهندسی نرم‌افزار، mahdiqezel@gmail.com[۳]

چکیده

اکثر متخصصین آزمون نرم‌افزار، «آزمون برنامه‌های موبایل» را بیشتر از سایر انواع نرم‌فزارها و سکوها، بسیار چالش‌برانگیز می‌دانند. درواقع، دستگاه‌ها و محیط‌های سیار بیش از «نرم‌افزار»یا «برنامه»، چالش را بر توسعه‌دهندگان نرم‌افزارهای موبایل تحمیل می‌کنند. در این مقاله سعی بر ارائه درک بهتری از چالش‌ها و موانع آزمون برنامه‌های موبایل است که با بیان و تشریح چالش‌ها و راه‌حل‌های آن‌ها این مهم به‌دست خواهد آمد. سپس با بیان تعاریف اولیه، دسته‌بندی، رویکردهای آزمون برنامه‌های موبایل، مقایسه حوزه‌های آزمون برنامه‌های موبایل با آزمون برنامه‌های اینترنتی و همچنین مزایا و معایب آزمون با دستگاه‌های حقیقی و آزمون با تقلیدکننده‌ها، افق جدیدی از آزمون برنامه‌های موبایل در پیش‌روی مخطبین ترسیم خواهد شد.

واژه‌های کلیدی: آزمون برنامه‌های موبایل، شبکه حامل، شبکه داده‌ای سلولی، اسکریپت آزمون، چالش‌های آزمون، آزمون مستحکم.

۱- مقدمه

با تغییر سریع فناوری کامپیوتر، میزان تمایل کاربران به دستگاه‌های سیار دستی (موبایل) افزایش یافته است. این افزایش تمایل علاوه بر تغییر سلیقه کاربران، بر روش زندگی، کسب‌وکارها و دولت‌ها نیز اثر گذاشته است. جنبه با اهمیت این تغییرها، تغییر روش‌کار و وظایف توسعه‌دهندگان و آزمون‌گرهای نرم‌افزار می‌باشد. بنابراین آزمون برنامه‌های موبایل، به علت اهمیت بیشتر دستگاه‌ها و محیط‌های سیار در مقایسه با برنامه در این محیط، نسبت به سایر انواع آزمون نرم‌افزار و سکو پرچالش‌ترین آزمون محسوب می‌شود. علاوه بر این چالش‌ها، متغیرها و پیچیدگی‌های زیاد باعث پنهان شدن خطاها شده و طراحی یک آزمون مستحکم را دشوار می‌کنند. به طور خاص در این آزمون نیاز به بررسی کارایی و قابلیت اطمینان شبکه، سازگاری رابط‌های کاربری، تنوع دستگاه‌ها و غیره وجود دارد [1].

در این مقاله در ابتدا با توصیف محیط و برنامه‌های موبایل و در ادامه با تشریح محیط محاسباتی سیّار، برای طراحی یک آزمون موفق، چالش‌های آزمون در این محیط که برگرفته از تنوع دستگاه‌های موبایل، زیرساخت شبکه حامل، متن‌نویسی و قابلیت استفاده می‌باشد، احصاء شده و راه‌های غلبه بر هرکدام از این چالش‌ها به رشته تحریر درآمده است. در ادامه با بیان برخی از رویکردهای آزمون، و نحوه توسعه طرح‌ها و موردهای آزمون، دو طبقه‌بندی آزمون برنامه‌های موبایل، یعنی آزمون با دستگاه‌های حقیقی و آزمون با تقلیدکننده‌ها بیان شده که به تفکیک برای هر طبقه‌بندی، مزایای هرکدام از طبقه‌بندی‌ها از قبیل مقرون‌به‌صرفه و آسان‌ترین نوع آزمون در آزمون با تقلیدکننده‌ها و همچنین امکان درک ظرافت و تفاوت‌های خاص بین دستگاه‌های مختلف برای آزمون با دستگاه حقیقی؛ همچنین چالش‌هایی از قبیل هزینه‌بربودن برای آزمون با دستگاه‌های حقیقی و عدم شناسایی تفاوت‌های ظریف و اشکالات خاص دستگاه‌ها و ناتوانی در تضمین کامل تأمین سازگاری و مشخصات کارایی در آزمون با تقلیدکننده‌ها بیان شده‌است و در پایان ضمن معرفی معروف‌ترین شبیه‌سازهای محیط موبایل، سعی شده است که درک بهتری از چالش‌ها و موانع آزمون برنامه‌های موبایل در خواننده ایجاد شود.

---

[۱] دانشگاه جامع امام حسین علیه السلام
[۲] دانشگاه جامع امام حسین علیه السلام
[۳] دانشگاه جامع امام حسین علیه السلام

www.SID.ir

۲- پیشینه تحقیق

با توجه به گرایش کاربران از برنامه‌های دسکتاپ و وب به سمت برنامه‌های موبایل و همچنین اهمیت آزمون برنامه‌های موبایل و چالش‌های مربوطه، محققین در این‌حوزه فعالیت‌های گسترده‌ای را انجام داده‌اند که برخی از این تحقیقات به شرح زیر ارایه می‌گردد:

آقای سامر و همکاران [2] با اعتقاد به این‌که فنون و روش‌های آزمون خاص آزمون برنامه‌های موبایل، در چندسال گذشته توجه بسیاری از مهندسین نرم‌افزار را به خود جلب کرده‌است، با نگاشت هفتاد و نه مطالعه تجربی به یک طرح طبقه‌بندی، چندین شکاف تحقیقاتی و مسائل کلیدی آزمون برای پزشکان از قبیل الزام به استخراج نیازمندی‌های آزمون در مراحل اولیه توسعه، انجام تحقیقات در محیط‌های توسعه در دنیای واقعی و مطالعات مقایسه‌ای برای آزمون امنیت و قابلیت استفاده را شناسایی کرده‌اند.

آقای ما و همکاران [7] LESDroid را برای تشخیص نشت خدمات صادراتی ارائه کرده‌اند. LESDroid به‌طور خودکار نمونه‌های سرویس و بارهای کاری (شروع/ توقف یا اتصال/ بازکردن سرویس‌های صادراتی) برنامه تحت آزمون را تولید می‌کند و یک اوراکل تعیین‌شده را برای تشخیص نشت سرویس به عکس فوری پشته اعمال می‌کند. LESDroid را با استفاده از ۳۷۵ برنامه تجاری ارزیابی کرده‌اند و ۹۷ سرویس لو رفته و ۹۸ ورودی مجزا در ۷۰ برنامه یافت شده‌است.

آقای سانگ و همکاران [8] چهار الگوی ضد باگ‌های ناکارآمدی استفاده از سرویس را شناسایی کرده‌اند، از جمله ایجاد زودرس، تخریب دیرهنگام، تخریب زودرس و نشت سرویس، یک فن تجزیه و تحلیل استاتیک به نام ServDroid را ارائه کرده تا به‌طور خودکار و مؤثر آن‌ها را بر اساس ضدالگوها تشخیص دهد. ایشان ServDroid را برای مجموعه بزرگی از برنامه های محبوب اندرویدی در دنیای واقعی اعمال کرده‌اند. نتایج نشان می‌دهد که به‌طور شگفت‌انگیزی، ناکارآمدی استفاده از خدمات رایج است و می‌تواند به شدت بر عملکرد برنامه تأثیر بگذارد.

آقای هانگ و همکاران [9] یک روش آزمایشی را بر اساس بسته‌های شبکه فازی برای برنامه‌های اندروید پیشنهاد کرده‌اند. این طرح از فناوری واسطه برای به دست آوردن داده‌های تعامل ارسال شده توسط سرور به برنامه استفاده می‌کند، استراتژی‌های جهش متفاوتی را برای جهش داده‌های اصلی اتخاذ کرده و داده‌های پاسخ جهش‌یافته را به برنامه‌ها برمی‌گرداند، از فناوری نظارت بر گزارش، برای نظارت بر اطلاعات خرابی استفاده می‌کند، در نتیجه امنیت بالقوه را کشف می‌کند. تهدیدها ۱۰ برنامه محبوب بر اساس روش پیشنهادی مورد آزمایش قرار گرفته و ۴ نوع مشکل کشف شد. مشکلات شامل عدم پاسخگویی، خرابی های ناشی از استثنای داده های JSON، جایگزینی محتوای HTML و تغییر مسیر URL است. نتایج نشان داد که روش پیشنهادی در افشای باگ های برنامه‌های موبایل در فرآیند تعامل داده های شبکه موثر بوده است.

آقای اتیلی و همکاران [10] یک مطالعه نقشه‌برداری سیستماتیک با هدف شناسایی، ارزیابی و طبقه‌بندی ویژگی‌ها، روندها و پتانسیل برای پذیرش تحقیقات موجود در تجزیه و تحلیل استاتیک برنامه‌های تلفن همراه ارائه کرده‌اند. با شروع از بیش از ۱۲۰۰۰ مطالعه بالقوه مرتبط، یک روش انتخاب دقیق را اعمال کرده که منجر به ۲۶۱ مطالعه اولیه در یک بازه زمانی ۹ ساله شده‌است. هر مطالعه اولیه با توجه به یک چارچوب طبقه‌بندی کاملاً تعریف شده تجزیه و تحلیل شده‌است. نتایج این مطالعه پایه محکمی برای ارزیابی رویکردهای موجود و آینده برای تجزیه و تحلیل استاتیک برنامه‌های تلفن‌همراه، به ویژه از نظر قابلیت پذیرش آن‌ها ارائه کرده‌است. محققان و متخصصان می‌توانند از نتایج این مطالعه برای شناسایی شکاف‌های تحقیقاتی فنی موجود با هدف، درک نحوه انتقال موفقیت‌آمیز رویکردهای توسعه‌یافته در دانشگاه به صنعت و موقعیت بهتر رویکردهایی برای تجزیه و تحلیل استاتیک برنامه های تلفن همراه، استفاده نمایند.

آقای مارکو و همکاران [11] ابزاری را ارائه کرده‌اند که امکان استخراج پویا، در یک رویکرد جعبه سیاه کامل، فعالیت‌های کاوش‌شده برنامه‌های اندروید را فراهم می‌کند. این یک توسعه ابزار تست iMPacT است که ترکیبی از مهندسی معکوس، اکتشاف پویا و آزمایش است. اطلاعات استخراج شده برای ساخت یک HFSM (ماشین حالت محدود سلسله مراتبی) با سه سطح مجزا از انتزاع استفاده می شود. سطح بالا تعاملات مورد نیاز برای عبور از فعالیت های برنامه تلفن همراه را نشان می‌دهد. سطح میانی صفحه‌هایی را نشان می‌دهد که در حین انجام یک فعالیت خاص عبور کرده‌اند. سطح پایین همه صفحه‌هایی را نشان می‌دهد که در طول کاوش از آن عبور کرده‌اند. این اطلاعات به درک بهتر برنامه کمک می‌کند که تعمیر و نگهداری آن و رفع خطاها را تسهیل می‌کند. همچنین شرح کاملی از ابزار، معماری آن و نتایج برخی از مطالعات موردی انجام شده بر روی برنامه‌های تلفن همراه در دسترس عموم در فروشگاه گوگل ارائه شده‌است.

www.SID.ir

کارهای دیگری نیز در این زمینه انجام شده است که به جهت ارتباط موثر خواننده و همچنین درک آسان، در موقعیت‌های مناسب در ادامه تشریح خواهد شد.

۳- آزمون برنامه‌های موبایل

آزمون برنامه‌های موبایل مشابه سایر آزمون‌ها با چالش‌هایی همراه است. از طرفی با توجه به محبوبیت برنامه‌های موبایل، تنوع کاربر زیادی در این برنامه‌ها وجود داشته که باعث به وجود آمدن تنوع در دستگاه‌های موبایل و سایر مولفه‌های مربوطه آن می‌شود. با این تفاسیر می‌توان از کار آقای اوبلی و همکاران [4] که تکامل اپلیکیشن های موبایل و تاثیر آن بر تکامل موارد تست را از زمان ایجاد تا مرحله انقضا مورد مطالعه قرار داده‌اند، به عنوان بنیادی‌ترین کار در این حوزه یاد کرد.

برای درک عمیق و بهتر چالش‌ها و موانع آزمون برنامه‌های موبایل؛ نیاز است که در ابتدا تصویر درستی از محیط موبایل ساخته شود.

۳-۱- محیط موبایل[۴]

با توسعه گسترده نقاط بی‌سیم و از بین رفتن فاصله بین محاسبات سیار و فعالیت‌های مبتنی بر شبکه‌های بی‌سیم، نیاز به تشریح اصطلاحات دستگاه‌ها و برنامه‌های موبایل مطرح است. با این توصیف دستگاهی که توانایی اجرای برنامه‌های مبتنی بر شبکه[۵] بر روی پیوند داده‌ای ماهواره‌ای یا سلولی را داشته باشد را دستگاه موبایل[۶] گویند. مانند: دستگاه‌های تلفن هوشمند، تبلت‌ها و PDAها. اغلب این دستگاه‌ها بدون هیچ مشکلی قادر به استفاده از نقاط دسترسی بی‌سیم[۷] که دارای سرعت و قابلیت اطمینان بیشتری نسبت به شبکه‌های تلفن همراه[۸] می‌باشند، هستند [1].

باتوجه به تعریفِ دستگاه‌های موبایل، برنامه موبایل[۹]، یک برنامه مبتنی بر شبکه است که روی یک دستگاه موبایل اجرا می‌شود. این برنامه‌ها با فرض استفاده از پیوندهای داده‌ای نسبتاً کند وغیرقابل اطمینان طراحی می‌شوند. همچنین روی یک دستگاه موبایل می‌توان برنامه های مستقلی مانند بازی‌ها[۱۰] را بدون نیاز به استفاده از شبکه حامل – مجموعه‌ای از دستگاه‌ها و زیرساخت‌ها برای انتقال داده از یک مکان به مکان دیگر - توسعه داد. برنامه‌های موردنیاز کاربران از فروشگاه‌های آنلاین برخی فروشندگان، مانند اپل و سامسونگ، قابل دریافت و یا خریداری است همچنین نصب و نگهداری این برنامه‌ها توسط فروشندگان برای سکوی خودشان گواهی و تضمین می‌شود.

۳-۲- محیط محاسباتی سیار[۱۱]

با رشد سریع ارتباطات مبتنی بر دستگاه موبایل، مکان جغرافیایی، در دسترسی به اطلاعات دارای نقش کم‌تر و کم‌رنگ‌تری شده‌است. بنابراین محاسبات مربوط به این اطلاعات نیز در محیط محاسباتی سیار انجام می‌شود که حوزه ای از کاربردهای فناوری اطلاعات است که در حال حاضر توجه بسیاری از صاحب‌نظران این حوزه را به خود جلب کرده است [12]. داشتن درک درست از این محیط محاسباتی سیار، باعث ایجاد طرح‌های آزمون موفق برای برنامه‌های موبایل است.

۳-۳- طراحی آزمون موفق

مطابق اصول روانشناسی آزمون، آزمون نباید اثبات کند که برنامه کار می‌کند بلکه باید اثبات کند که برنامه برای موردهای آزمون[۱۲] موردنظر کار نمی‌کند. به عنوان مثال، درصورت کند یا غیرقابل دسترس بودن شبکه حامل یک برنامه ایمیل، آزمون‌ها باید مشکلات نرم‌افزار را شناسایی کنند. برخی از حوزه‌های مهم حین طراحی طرح‌های آزمون عبارتنداز: اتصال[۱۳]، تنوع دستگاه‌ها[۱۴]، محدودیت‌های دستگاه[۱۵]، وسایل ورودی[۱۶] و نصب و

---

4 Mobile Environment
5 Network-Based Applications
6 Mobile Device
7 Wireless Access Points
8 Cellular Networks
9 Mobile Application
10 Games
11 Mobile Computing Environment
12 Test Cases
13 Connectivity
14 Diversity Devices
15 Device Constraints
16 Input Devices

www.SID.ir

نگهداری[17] که با درنظر گرفتن دستگاه‌ها سه حوزه عمده تنوع، محدودیت و روش‌های ورودی[18] مطرح می‌شود. به عبارت دیگر آزمون‌گرها برای ایجاد طرح‌های آزمون موفق باید دستگاه‌های بی‌شمار با قابلیت‌های مختلف موجود در بازار و همچنین چگونگی تعامل کاربران با آن‌ها را در نظر بگیرند. در پایان نیز باید چگونگی نصب و نگهداری برنامه تشریح شود. وجود یک سیستم تضمین شده، نصب و نگهداری برنامه را تسهیل می‌نماید. برای طرح یک آزمون اقتصادی، تشخیص حداکثر خطا در یک زمان معقول، باید توازنی بین زمان، مسائل مالی و منابع انسانی برقرار کرد.

همچنین هنگام طراحی موارد آزمون علاوه بر درنظر گرفتن سه حوزه عمده تنوع، محدودیت و روش‌های ورودی، چهار طبقه‌بندی چالش‌های آزمون برنامه‌های موبایلی را باید در نظر گرفت. این طبقه‌بندی‌ها عبارتنداز:

۱- تنوع دستگاه؛
۲- زیرساخت شبکه حامل[19]؛
۳- متن‌نویسی[20]؛
۴- قابلیت استفاده[21].

در ادامه این طبقه‌بندی‌ها و چگونگی غلبه بر این چالش‌ها شرح داده می‌شوند.

۳-۴- تشریح چالش‌های آزمون و راه‌کار غلبه بر آن‌ها

در این بخش، تنوع دستگاه‌های موبایل، زیرساخت شبکه حامل، متن‌نویسی و قابلیت استفاده که از طبقه‌بندی چالش‌های آزمون نیز به‌شمار می‌روند، تبیین و تشریح شده تا دید واضح‌تری در اختیار خوانندگان محترم قرار گیرد، در ادامه با طرح برخی مفاهیم فنی مورد بحث در حوزه شبکه‌های کامپیوتری سعی شده است تا چالش‌های موجود در حوزه زیرساخت شبکه حامل، به مخاطب منتقل گردد. در پایان نیز به راهکارهایی جهت غلبه بر این چالش‌ها اشاره شده‌است.

۳-۴-۱- تنوع دستگاه‌های موبایل

با تنوع بی‌وقفه دستگاه‌ها و معرفی دستگاه‌های جدید توسط سازندگان مختلف، سیستم‌های عامل، مرورگرها، محیط‌های زمان اجرای برنامه، دقت تفکیک صفحات، رابط‌های کاربری، ارگونومی‌ها، اندازه صفحه؛ گستردگی آزمون روزبه‌روز بیشتر می‌شود. آگاهی از تمامی این عوامل برای انجام آزمون الزامی است. به عنوان مثال، با ارائه یک روش جدید برای ورود متن برای یک دستگاه موبایل موجود در بازار، آزمون‌گرها باید آزمونی را طراحی کنند که شکست یا عدم شکست برنامه موبایلی در همکاری با روش ورودی جدید را تعیین کند [1].

به دلیل تنوع دستگاه‌های موبایل، آزمون قابلیت استفاده از اهمیت بالایی برخوردار است.

۳-۴-۲- زیرساخت شبکه حامل

آزمون برنامه روی یک شبکه حامل، به دو دلیل ۱. درک و تطبیق با زیرساخت حامل، و ۲. غلبه برموانع مکانی با چالش همراه است. زمانی که پشتیبانی چندین حامل مدنظر باشد، این چالش واقعی‌تر احساس می‌شود.

۳-۴-۲-۱- درک و تطبیق با زیر ساخت حامل[22]

برای توسعه یک طرح آزمون خوب، فهم زیرساخت حامل اساسی است.

معمولا در زیرساخت حامل، پروتکل IP-Based نبوده و RF-Based است. پروتکل‌های RF-Based مانند GSM[23]، TDMA[24] یا CDMA[25]، به‌عنوان حمل‌بار[26] پروتکل IP-Based رفتار کرده و آن‌ها را به دستگاه موبایل تحویل می‌دهند، سپس حمل‌بار را رمزگشایی کرده و آن را به برنامه ارائه می‌کنند.

---

17 Installation and maintenance
18 Input Methods
19 Carrier Network Infrastructure
20 Scripting
21 Usability
22 Understanding and Adapting to the Carrier's Infrastructure
23 Global System for Mobile Communication
24 Time-Division Multiple-Access
25 Code-Division Multiple-Access

www.SID.ir

همچنین اکثر حامل‌ها، بین اینترنت و دستگاه از تبدیل کد یا وب پراکسی استفاده می‌کنند که ممکن است وظایف متنوعی از قبیل تبدیل محتوا به WAP[۲۷] یا HTTP[۲۸]، فشرده‌سازی داده برای توان عملیاتی بهتر، رمزنگاری ترافیک برای حفظ حریم خصوصی و امنیت، مسدود کردن دسترسی به سایت های خاص با پهنای باند بالا و حذف هدرها و سایر متاداده‌های HTML[۲۹] از صفحات وب مورد استفاده برنامه؛ را انجام دهند که خود چالش هایی مانند ناسازگاری رابط گرافیکی بین چند دستگاه پس از تبدیل کد و اختلاف دستگاه‌ها در استفاده از پروتکل‌های WAP و HTTP را رقم می‌زند.

درصورت برخورد با مشکل رابط گرافیکی در بین دستگاه‌ها و حامل‌ها، هرکدام را برای تعیین پروتکل و پروتکل تحویل محتوا مورد استفاده (WAP/WML یا HTTP/HTML) بررسی کنید.

توجه داشته باشید که بر خلاف این تصور که فشرده‌سازی داده، توان عملیاتی را بهبود می‌بخشد، اغلب به دلیل سربار زیاد آن ممکن است توان عملیاتی کاهش یابد و همچنین با افزایش ملاحضات امنیتی، فایروال‌ها و غیره، توان عملیاتی را کند می‌کنند [1].

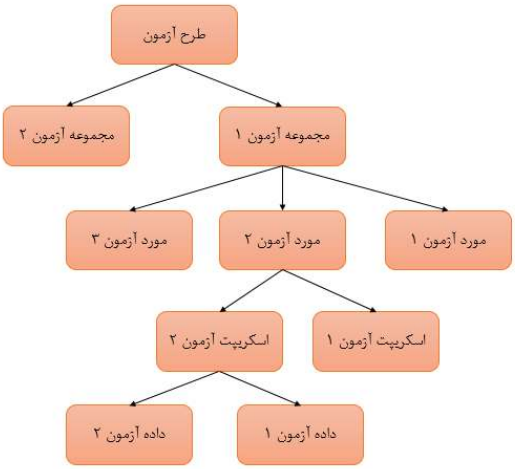
۳-۴-۲-۲- غلبه بر موانع مکانی[۳۰]

برای آزمون در یک شبکه حامل، باید به آن شبکه دسترسی پیدا کرد. برای آزمون شبکه‌های حامل در سایر بخش‌های کشور یا سایر کشورها باید به آن بخش‌ها مسافرت کرد و یا کسی را در آن‌جا هماهنگ نمود تا برای شما آزمایش انجام دهد که هر دو رویکرد، هزینه آزمون را افزایش می‌دهد.

۳-۴-۳- متن‌نویسی

یک افق از آزمون برنامه‌های موبایل ایجاد و اجرای اسکریپت‌های آزمون است [1]. مفهوم اسکریپت (متن) آزمون و مقایسه آن با طرح آزمون، مورد آزمون، مجموعه آزمون و داده آزمون در شکل ۱ – الف به روشنی بیان شده است. در شکل ۱ – ب نیز چند نمونه از اسکریپت‌های آزمون جهت آشنایی کلی خوانندگان محترم آورده شده است.

```
def sample_test_script (self):
type ("TextA")
click (ImageButtonA)
assertExist (ImageResultA)
```



شکل۱- مقایسه test plan، test suite، test case، test script و test data . ب) نمونه اسکریپت آزمون. [13]

در دستگاه‌های موبایل واقعی، بارگذاری خودکار اسکریپت‌های تکرارپذیر مجاز نیست، بنابراین آزمون‌گرها، تمامی اسکریپت‌های آزمون را به صورت دستی اجرا می‌کنند. به عبارت دیگر؛ آزمون‌گر، یک اسکریپت آزمون را که برای کشف خطاها با یک مورد آزمون روی دستگاه هدف در محیط موبایل، طراحی شده است را پیمایش می‌کند. متأسفانه از آنجایی که آزمون دستی معمولاً مستعد خطا می‌باشد، هنگام آزمون برنامه‌های موبایل، خطاها غیرقابل اجتناب هستند.

با قوی‌تر و پیچیده‌تر شدن دستگاه های موبایل و همچنین رقابت در بازار، احتمالاً این مسئله در آینده موضوعیت نخواهد داشت و دستگاه های موبایل واقعی اجازه بارگذاری خودکار اسکریپت‌های تکرار پذیر را خواهند داد.

---

[26] Payload
[27] Wireless Application Protocol
[28] Hyper Text Transfer Protocol
[29] Hyper Text Markup Language
[30] Location-Based Hurdles

www.SID.ir

۳-۴-۴- قابلیت استفاده

آزمون قابلیت استفاده اکثراً یک رویکرد جعبه-سیاه است، زیرا سیستم به عنوان یک جعبه سیاه دیده شده و تنها رابط‌های خارجی سیستم که در تعامل با کاربر نهایی هستند، آزمون می‌شوند. در این آزمون کشف خطاهای رابط‌های کاربری و لایه‌های تعامل کاربری برنامه به صورت دستی مدنظر است، بنابراین چالش‌هایی مشابه اسکریپت‌های آزمون وجود دارند.

دستگاه‌های موبایل شامل بیش از یک سکو برای آزمون می‌باشند، بنابراین نیاز به ملاحظات سازگاری رابط کاربری بین سکوهای مختلف احساس می‌شود [1].

۳-۵- رویکردهای آزمون[31]

در آزمون دستگاه‌های موبایل آزمون لایه‌های کسب‌وکار (لایه ۲) و داده (لایه ۳) تقریباً باید پیکره‌بندی مشابه آزمون یک برنامه استاندارد اینترنتی را داشته باشند. حوزه‌ها در این آزمون مشابه آزمون برنامه‌های اینترنتی که باید مرورگرهای متعددی ارزیابی شوند، به صورت نمایی افزایش پیدا می‌کنند. بنابراین؛ بابه‌کار بستن فنون مشابه آزمون برنامه‌های اینترنتی، آزمون مولفه‌های پشتی، ورودی داده پشتی[32]، زمان‌پاسخ، جامعیت داده، تحمل خطا و قابلیت بازیافت در آزمون لایه داده و آزمون مشخصات کارایی، روال‌های اعتبارسنجی داده و مولفه‌های پردازش تراکنش در آزمون لایه کسب‌وکار؛ آزمایش می‌شوند.

برخلاف آزمون لایه‌های ۲ و ۳، آزمون لایه ۱، یعنی محیط کاربر (معادل آزمون دستگاه)، از آزمون سنتی اینترنت[33]، به دلیل بحث برانگیز بودن مفاهیمی مثل آزمون محتوا و معماری وب‌سایت، متفاوت است [1].

برنامه‌های وبی در مقایسه با شبکه‌های سلولی -شبکه تحت پوشش بیسیم که ناحیه تحت پوشش آن به بخش‌های زیادی که سلول نامیده می‌شود، تقسیم شده‌است. هرکدام از سلول‌ها یک ایستگاه فرستنده-گیرنده در موقعیتی ثابت دارد تا با کاربران درون سلول ارتباط برقرار کند.- به دلیل استفاده از شبکه‌های LANs/WANs پایداری بیشتری دارند و در صورت کار با شبکه‌های سلولی، باید انتظار قطعی شبکه را داشته باشید. هدف از آزمون برنامه در شبکه حامل، یافتن مشکلات مرتبط با پوشش نقص[34] یا قطع ناگهانی[35] اتصال از طریق پاسخگویی به سوالاتی مانند سوالات زیر است:

- هنگام ترمیم پوشش شبکه پس از یک وقفه در طی بارگذاری برنامه، چه اتفاقی می‌افتد؟
- آیا یک خرید دوبار اتفاق می‌افتد؟

اشکالات مرتبط با اداره مقداردهی مجدد نشست[36] و خرابی داده‌ها[37] را نیز در آزمون برنامه مورد بررسی قرار دهید.

برای اطمینان از این‌که مشخصات طراحی با استفاده از آزمون وظیفه‌مندی[38] برآورده می‌شوند، در صورت امکان، مولفه‌های لایه ۲ و ۳ را مستقل از دستگاه آزمون کنید.

۳-۵-۱- توسعه طرح‌ها و موردهای آزمون

اهمیت موردهای کاربری هنگام توسعه طرح‌های آزمون (شکل ۱) برای دستگاه‌ها مهم است. برنامه‌های موبایل بی‌نهایت نقطه شکست دارند، پس، آگاهی از این‌که چه کسی، چگونه و چه وقت برنامه شما را استفاده خواهد کرد، ضروری است [1]. یک مورد آزمون خاصِ آزمون برنامه‌های موبایل، چگونگی مدیریت صوت و پیام‌های متنی وارده توسط برنامه می‌باشد [1]. به‌عنوان مثال، یک کاربر نهایی می‌خواهد در حین پاسخ به تلفن یا خواندن پیام متنی، برنامه را موقتاً تعلیق و یا آن را در پس زمینه اجرا کنند، در چنین مواردی، در نقاطی که صوت و پیام‌های وارده، سبب بروز مشکلات در برنامه می‌شوند، موردهای آزمون را بسازید [1].

۳-۵-۲- دسته‌بندی آزمون برنامه‌های موبایل

در آزمون برنامه‌های موبایل، باید برنامه موبایل از جنبه‌های مختلف آزمون شود؛ بنابراین، اطمینان از این‌که برنامه به‌درستی نصب و یا از حالت نصب خارج می‌شود، بررسی[39] پاسخ مناسب برنامه به مواردی از قبیل قطعی شبکه، ترمیم شبکه و سیگنال‌های ضعیف، آزمایش توانایی کاربر برای

---

[31] Testing Approaches
[32] Backend Data Entry
[33] Traditional Internet Testing
[34] Spotty coverage
[35] Sudden loss
[36] handling session re-initialization
[37] Data Corruption
[38] Functional Testing
[39] Verify

www.SID.ir

قبول صوت و پیام‌های متنی در حین اجرای برنامه، ادامه مجدد برنامه هنگام اتمام فراخوانی‌های صوتی/ پیام‌های متنی، نپذیرفتن فراخوانی‌های صوتی/پیام‌های متنی و آغاز یک فراخوانی صوتی/پیام متنی بدون قطع برنامه، اطمینان از پایدار باقی ماندن برنامه هنگام برخورد دستگاه با وضعیت کمبود حافظه[40] ، آزمایش کارایی تمامی کلیدها (نگاشت کلیدها[41]) مطابق مشخصات، اطمینان از رخداد بازخورد[42] کاربر از فشردن کلید مطابق مشخصات طراحی برنامه، بررسی خروج برنامه با ظرافت و آرامی هنگام آغاز فشردن کلید، بستن کاور و یا استفاده از نوار لغزنده[43] و همچنین تایید[44] برآورد مشخصات طراحی درصورتی که کاربر دستگاه را خاموش کند، اطمینان از مطابق طراحی کار کردن برنامه هنگام ورود، درحین و هنگام خروج از حالت شارژ، آزمایش رفتار برنامه حالت باتری ضعیف، اندازه گیری سرعت تخلیه برنامه توسط باتری و اطمینان از پاسخ مطابق مشخصات برنامه هنگام خروج باتری در حال روشن بودن دستگاه[45]، اطمینان از عدم ایجاد اضافه‌بار[46] روی CPU و عدم مصرف حافظه بیش‌ازحد[47]، مورد توجه و بحث می‌باشد [1].

۳-۵-۳- استفاده از تقلیدکننده‌ها[48] و دستگاه‌های واقعی

در رویکردهای آزمون دستگاه اساسا دو انتخاب وجود دارد [1]:

- آزمون روی دستگاه‌های حقیقی[49]؛
- استفاده از تقلیدکننده‌های دستگاه[50].

استفاده از تقلیدکننده‌ها روش خوبی برای شروع آزمون می‌باشد اما در نهایت نیاز به آزمون دستگاه‌های واقعی روی شبکه‌های حامل واقعی وجود دارد. تقلیدکننده‌ها، دستگاه واقعی نیستند و فقط تقلید می‌کنند، بنابراین ممکن است تفاوت‌هایی بین آزمون با یک تقلیدکننده و یک دستگاه واقعی مشاهده شود. به عنوان مثال ممکن است دقت تفکیک‌پذیری، عمق رنگ صفحه، شکل‌های دکمه‌ها و جعبه‌های ورودی، آزمون‌های قابل قبولی روی یک تقلیدکننده مبتنی بر کامپیوتر داشته باشند اما روی دستگاه‌های واقعی شکست بخورند. بنابراین باید در حین مراحل تحلیل نیازمندی‌ها و نوشتن مشخصات، تصمیم دقیقی برای انتخاب، پشتیبانی و آزمون زیرمجموعه معقولی از دستگاه‌ها اتخاذ کرد.

اما با این وجود از دیدگاه اقتصادی، آزمون روی تقلیدکننده‌ها در مقابل دستگاه‌های واقعی مقرون‌به‌صرفه است. از طرفی هم باید دسترسی بالقوه صدها دستگاه موبایل به برنامه مورد نظر تحقق یابد، در غیر اینصورت، به ازای هر دستگاهی که در زیرمجموعه دستگاه‌های شما نباشد و با برنامه شما کار نکند، ممکن است موجب از دست رفتن نه تنها یک مشتری بلکه یک پایگاه مشتری شود.

برای غلبه بر چالش متن‌نویسی آزمون، برخلاف این‌که تقلیدکننده‌ها وظیفه‌مندی متن‌نویسی غنی‌ای دارند و می‌توانند آزمون‌های بازگشتی و سیستمی را انجام دهند، هنوز نیاز به فردی است که با دستگاه کار کند و بارگذاری اسکریپت‌های تکرارپذیر را به صورت دستی انجام دهد [1].

۳-۵-۳-۱- آزمون با دستگاه‌های حقیقی

آزمون دستی با دستگاه‌های حقیقی اجتناب ناپذیر است. برخلاف هزینه‌بر بودن، این آزمون امتیازاتی از قبیل تجربه تفاوت‌های ظریف بین دستگاه‌ها، امکان آزمون حالت‌های خاص – به عنوان مثال: آزمون قابلیت اطمینان یک شبکه حامل و تعیین اثر یک صوت یا پیام متنی وارده –، ارزیابی چگونگی رفتار یک برنامه – به عنوان مثال: بارگزاری و اجرا با سرعت قابل قبول – و تعیین خطاهای خاص دستگاه را دارا می‌باشد. توجه داشته باشید که چالش اصلی در صورت کشف خطای خاص دستگاه، رفع آن خطا بدون شکست سازگاری با سایر دستگاه‌ها است.

باوجود امتیازات آزمون با دستگاه‌های حقیقی، علاوه بر هزینه بر بودن این نوع آزمون دستگاه موبایلی به دلیل خریداری یا اجاره (کاهش نسبی هزینه) دستگاه، شبکه حامل و دستگاه خاص مانند خانواده Apple iPhone، ضعف‌هایی از قبیل نیاز به تعداد کافی از انواع (iTouch، iPhone و iPad)، حذف اسکریپت‌های آزمون خودکار[51] از جعبه آزمون‌گر نرم‌افزار و الزام به استفاده از اسکریپت‌های دستی و همچنین دشواری در

---

40 Low Memory
41 Key Mappings
42 Feedback
43 Slider
44 Confirm
45 Battery Conditions
46 Overload
47 Device Interaction
48 Emulators
49 Real Devices
50 Device Emulators
51 Automated Test Scripts

www.SID.ir

ایجاد و نگهداری اسکریپت‌های آزمون تفصیلی برای هر دستگاه که ممکن است هنگام به‌روزرسانی دستگاه منسوخ شده باشد، وجود دارند. از طرفی می‌توان با اسکریپت‌های عمومی، مشخصات سیستم در بین چندین دستگاه را آزمایش کرد. به عنوان مثال – iPhoneها، iPad و دستگاه های اندروید-پایه، برای ورودی کاربران، به شدت بر صفحات لمسی تکیه می‌کنند اما سایر دستگاه‌ها مانند BlackBerries (نوعی گوشی همراه) یا گوشی‌های استاندارد برای ورودی کاربران از صفحه کلید استفاده می‌کنند. در جدول ۱ یک اسکریپت نمونه برای بررسی چگونگی مدیریت یک پیام متنی دریافتی در حین خواندن یک کتاب الکترونیکی در یک برنامه الکترونیکی خوان[۵۲] آورده شده‌است که برای انجام هر گام از تسهیلات ورودی کاربر شامل: دکمه‌ها، صفحات لمسی یا دستورات صوتی استفاده می‌کند.

جدول۱- اسکریپت آزمون عمومی دستگاه [1]

ردیف	نوع عمل
۱	برنامه e-reader را شروع کنید.
۲	کتاب الکترونیکی را باز کنید.
۳	یک پیام کوتاه از دستگاه دیگر به دستگاه مورد نظر ارسال کنید.
۴	بررسی کنید (verify) هشدار پیام کوتاه نمایش داده می‌شود.
۵	پیام کوتاه را باز کنید.
۶	عمل reply را برای پیام کوتاه انتخاب کنید.
۷	پیام کوتاه را بنویسید.
۸	پیام کوتاه را ارسال کنید.
۹	اطلاع‌رسانی مربوط به پیام کوتاه را بررسی کنید (verify).
۱۰	به کتاب الکترونیکی برگردید.
۱۱	بررسی کنید (verify) برنامه e-book درحال اجرا است.
۱۲	بررسی کنید (verify) به همان صفحه برمی‌گردید.
۱۳	از برنامه e-reader خارج شوید.

آزمون با دستگاه حقیقی، یک فرایند دستی و جعبه-سفید است. یعنی فردی باید دکمه‌هایی را فشار داده یا به صفحه ضربه بزند و داده‌ها را وارد کند. روشن است که آزمون دستی حتی با بهترین دستورالعمل‌ها و آزمون‌گرهای حرفه‌ای مستعد خطا است.

در مورد هر اسکریپت آزمون[۵۳] و نتایج آن باید، یادداشت‌های دقیقی نگهداری شده سپس اثربخشی اسکریپت‌ها ارزیابی تا آنهایی که ارزش کمی دارند – یعنی برای کشف خطاها شکست می‌خورند – حذف شوند.

۳-۵-۳-۲- آزمون با تقلیدکننده‌ها

ممکن است این رویکرد ارجح نباشد اما معمولا رویکردی کارآمد، عملی‌ترین و مقرون‌به‌صرفه‌ترین روش است. حتی دارای برخی امتیازات ویژه از قبیل انجام ارزان‌تر و سریع‌تر آزمون وظیفه‌مندی، مدیریت آسان، پشتیبانی از دستگاه‌های متعدد و جلوگیری از هزینه گران اعتبار حامل[۵۴]، امکان اجرای شبیه‌ساز با بهره‌گیری از منابع بیشتر (پردازنده سریع‌تر و حافظه بیشتر)، ایجاد آزمون‌ها و اسکریپت‌های خودکار به‌دلیل کاهش خطا در آزمون دستی و عدم وابستگی زبان‌های اسکریپتی به دستگاه (مانند گام ۸ در جدول ۱)؛ می‌باشد. در همین راستا آقای ابراهیم و همکاران [5] یک مطالعه موردی از برنامه تلفن همراه ارائه کرده و در مورد چگونگی ایجاد موارد آزمون به طور خودکار از برنامه با استفاده از ابزارهای خودکار مختلف بحث کرده‌اند. سه ابزار تست نرم‌افزاری را برای تولید خودکار موارد آزمون مورداستفاده قرار داده‌اند. سپس نتایج حاصل از تولید موارد آزمون به طور خودکار از این سه ابزار با نتایج حاصل از تولید موارد آزمون با استفاده از فن آزمون دستی مقایسه شده‌است. همچنین آقای مارینهو و همکاران [3] روشی به نام پلتفرم آزمون برنامه های موبایل (PLATEM) و ابزار پشتیبانی آن به‌نام PLATOOL را برای کمک به توسعه دهندگان در آزمون خودکار سیستم پیشنهاد می‌کنند. PLATEM و PLAtem قادر به مقابله با رویدادهای برنامه‌های کاربردی تلفن همراه در طول آزمون خودکار سیستم هستند. یکی از جنبه‌های مهم PLATEM و PLATOOL، بهره‌گیری از ارتباط آزمون‌های تضمین کیفیت

[52] E-Reader
[53] Well-Document Test Scripts
[54] Carrier Airtime

www.SID.ir

نرم‌افزار و آزمون‌های طراحی آزمون‌محور است. از طرفی نیز آقای کوشارکی و همکاران [6] از یک ابزار استاندارد داده جهش یافته به نام MutantBench برای تولید داده های خود استفاده کرده‌اند. سپس، یک درخت نحو انتزاعی همراه با یک شبکه عصبی کانولوشنال مبتنی بر درخت به عنوان مدل یادگیری عمیق برای خودکارسازی طبقه‌بندی جهش‌یافته‌های معادل برای کاهش هزینه آزمون جهش در تست نرم‌افزاری برنامه‌های اندروید استفاده شده‌است. نتایج نشان می‌دهد که مدل پیشنهادی میزان دقت خوب ۹۴٪ و همچنین سایر معیارهای عملکردی مانند یادآوری (۹۶٪) و دقت (۸۹٪) را تولید می‌کند.

اما استفاده از تقلیدکننده‌ها برای آزمون نیز داری معایبی از قبیل؛ عدم شناسایی تفاوت‌های ظریف و اشکالات خاص دستگاه‌ها و ناتوانی در تضمین کامل تأمین سازگاری و مشخصات کارایی؛ می‌باشد [1].

۳-۵-۳-۳- محبوب‌ترین شبیه‌سازهای موبایل

از محبوب‌ترین شبیه‌سازهای موبایل برای آزمون برنامه‌های موبایلی، می‌توان به سه شبیه‌ساز MeMU ، Nox وBlueStacks اشاره نمود. محیط شبیه‌ساز MeMU، به دلیل پشتیبانی آن از چیپست‌های AMD و Intel باعث پردازش سریع‌تر بازی‌های اندرویدی روی کامپیوترهای شخصی شده و کارآیی آزمون این‌گونه برنامه‌ها را افزایش می‌دهد. شبیه‌ساز Nox، به دلیل رایگان بودن و قابلیت پشتیبانی از اندروید نسخه ۹ برای انواع کاربری‌های مرتبط با بازی و بازی‌سازی و نیز ویژگی‌های بهینه‌سازی‌شده و عملکرد مناسب حتی در ویندوزهای قدیمی‌تر، از محبوبیت خوبی برخوردار است. همچنین، شبیه‌ساز BlueStacks به دلیل قابلیت اجرا بر روی سیستم‌عامل‌های ویندوز و MacOs، و ایجاد چندین همتا[55] به صورت همزمان و داشتن قابلیت‌ها و نمادهای مخصوص بازی، کارآیی مطلوبی برای آزمون برنامه‌های موبایلی از خود نشان می‌دهد. [14] .

۴- نتیجه‌گیری

با بیان پیشینه تحقیقات انجام‌گرفته در حوزه آزمون برنامه‌های موبایل، این نتیجه حاصل شد که آزمون این‌گونه برنامه‌ها کماکان یک مقوله چالش‌برانگیز بوده که بسیاری از متخصصین حوزه مهندسی نرم‌افزار را با خود درگیر کرده است. مطابق آنچه بیان شد، با تعریف محیط موبایل و محیط محاسباتی سیار و پس از ترسیم موانع و چالش‌های آزمون برنامه‌های موبایل و بیان رویکردهای این آزمون، مقایسه‌ای میان آزمون برنامه‌های اینترنتی و برنامه‌های موبایلی انجام گرفت و معیارهایی برای چگونگی انجام این آزمون بیان شد. با تقسیم آزمون برنامه‌های موبایل به دو دسته آزمون با دستگاه‌های حقیقی و آزمون با تقلیدکننده‌ها، این نتیجه به دست آمد که هرکدام از رویکردهای آزمون در محیط‌های واقعی و تقلیدکننده‌ها، دارای مزایا و معایب خاص خود هستند. به‌عنوان مثال، آزمون در محیط‌های واقعی بر خلاف این‌که موجب شناسایی تفاوت‌های خاص هر دستگاه می‌شود اما هزینه‌بر و در مواردی غیرعملی می‌باشد، در مقابل، آزمون در محیط تقلیدکننده، با وجود مقرون‌به‌صرفه بودن، نمی‌تواند سازگاری و مشخصات کارایی را به صورت کامل تضمین نماید. لازم به ذکر است که هیچ‌کدام از این دو رویکرد آزمون با دستگاه‌های واقعی یا تقلیدکننده قابل جایگزین برای دیگری نخواهد بود. لذا بهترین راه‌حل استفاده متعادل از هر دو رویکرد است. به عبارت دیگر هر دو رویکرد مکمل یکدیگر هستند و نمی‌توانند جایگزین یک‌دیگر قرار گیرند.

[55] Clone

www.SID.ir

مراجع

[1] GLENFORD J. MYERS, COREY SANDLER, TOM BADGETT, THE ART OF SOFTWARE TESTING, New Jersey: WILEY, 2012.

[2] Zein, Samer,Salleh, Norsaremah,Grundy, John, "A Systematic Mapping Study of Mobile Application Testing," *Journal of Systems and Software,* vol. 117, 2016.

[3] Marinho, Euler,Resende, Rodolfo, "PLATEM: A method for mobile applications testing," *IET Software,* vol. 11, 2017.

[4] Ait Oubelli, Lynda,Mottu, Jean-Marie,Attiogbé, Christian, Test Cases Evolution of Mobile Applications, 2015.

[5] Rosziati Ibrahim,Nurul Ain Aswini Abdul Jan,Sapiee Jamel and Jahari Abdul Wahab, "Generating Test Cases using Eclipse Environment – A Case Study of Mobile Application," *International Journal of Advanced Computer Science and Applications(IJACSA),* vol. 12, no. 4, 2021.

[6] Kusharki, Muhammad,Misra, Sanjay,Muhammad-Bello, Bilkisu,Anka, Salihu,Suri, Bharti, "Automatic Classification of Equivalent Mutants in Mutation Testing of Android Applications," 2022.

[7] Ma, Jun,Liu, Shaocong,Jiang, Yanyan,Tao, Xianping,Xu, Chang,Lu, Jian, LESdroid: a tool for detecting exported service leaks of Android applications, 2018, pp. 244-254.

[8] Song, Wei,Zhang, Jing,Huang, Jeff, ServDroid: detecting service usage inefficiencies in Android applications, 2019, pp. 362-373.

[9] Huang, Xinyue,Zhou, Anmin,Jia, Peng,Liu, Luping,Liu, Liang, "Fuzzing the Android applications with HTTP/HTTPS network data," *IEEE Access,* vol. PP, p. 1, 2019.

[10] Autili, Marco,Malavolta, Ivano,Perucci, Alexander,Scoccia, Gian,Verdecchia, Roberto, "Software Engineering Techniques for Statically Analyzing Mobile Apps: Research Trends, Characteristics, and Potential for Industrial Adoption," *Journal of Internet Services and Applications,* vol. 12, 2021.

[11] Gonçalves, Marco,Paiva, Ana, "REiMPAcT," in *Reverse Engineering of Android Applications*, 2020, pp. 369-382.

[12] P. Keenan, Geographic Information Systems, Amsterdam: Elsevier, 2003, pp. 421-432.

[13] [Online]. Available: https://softwaretestingfundamentals.com/test-script/.

[14] [Online]. Available: https://vigiato.net/p/280223.

www.SID.ir