

روند معماری سیستم عامل تلفن همراه

جهان مسطح است، چرا که به‌طور فزاینده‌ای متحرک، سریع، متصل و امن است. مردم انتظار دارند به‌راحتی با دستگاه‌های تلفن همراه خود به اطراف حرکت کنند و در عین حال ارتباطات نزدیک خود با شرکا و خانواده‌شان را حفظ کنند و از همه‌ی مدل‌ها و محتویات بی‌نهایت آن بدون هیچ گونه نگرانی در مورد دستگاه و مدیریت داده‌ها بهره ببرند. تمام این نیازها بر روی دستگاه‌های تلفن همراه، که از سیستم عامل به عنوان روح تلفن همراه یاد می‌کنند، قرار داده شده است. براساس تجربه‌ی ما در طراحی سیستم عامل تلفن همراه و بررسی گسترده‌ی وضعیت صنعت حاضر، معتقدیم مشترکاتی در معماری آینده سیستم عامل تلفن همراه، مانند تجربه کاربران، مدیریت انرژی، طراحی امنیتی، پشتیبانی ابر و باز بودن طراحی وجود دارد. بنابراین یک مدل تجزیه و تحلیل برای هدایت تحقیقاتمان ارائه دادیم. در این مقاله، تحقیقات ما در روند معماری سیستم عامل تلفن همراه تا دهه‌ی آینده با تمرکز بر مشترکات عمده توصیف شده است. براساس یافته‌ها، ویژگی‌های سیستم عامل تلفن همراه از دیدگاه روند معماری امروز را نیز بررسی کرده‌ایم.

معرفی

طراحی سیستم عامل موبایل نیاز به تجربه‌ی سه فاز تکامل دارد: از سیستم عامل مبتنی بر PC تا سیستم عامل تعبیه شده در سیستم عامل گوشی‌های هوشمندگرا در دهه گذشته است. در طول این روند، معماری سیستم عامل تلفن همراه از دشوار به ساده یا چیزی مابین این دو تغییر کرده است. روند تکامل به‌طور طبیعی توسط پیشرفت فن‌آوری در سخت افزار، نرم افزار و اینترنت است:

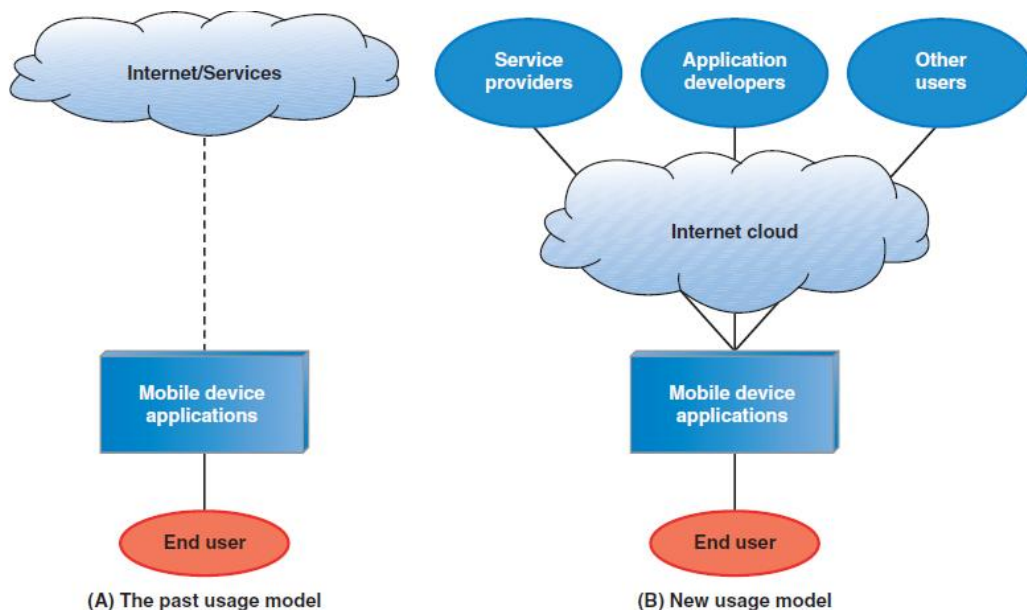
- **سخت افزار:** صنعت موجب کاهش اندازه‌ی ریزپردازنده‌ها و لوازم جانبی برای طراحی دستگاه‌های واقعی همراه است. قبل از کاهش اندازه‌ی دستگاه‌ها، دستگاه تلفن همراه نمی‌توانست همزمان به هر دو قابلیت اندازه‌ی کوچک و قابلیت پردازش برسد. ما تا به حال لپ‌تاپ‌هایی هم اندازه‌ی PC یا یک دستیار بسیار ضعیف اطلاعات شخصی (PDA) در اندازه تلفن داشتیم. سیستم عامل‌های تلفن همراه برای PDA ها معمولاً لازم نیست کاملاً چند وظیفه‌ای باشد و یا

از کارت گرافیک‌های سه بعدی پشتیبانی کند. ویژگی‌هایی مانند حسگرها، شتاب‌سنج و صفحه نمایش لمسی مبتنی بر خازن در سیستم‌عامل تلفن همراه‌های گذشته در دسترس نبود.

- **نرم افزار:** با یک کامپیوتر لپ‌تاپ، نرم‌افزار عمدتاً به بهره‌وری کاربر تمرکز دارد، که برای حمایت از صفحه‌کلید و ماوس برای ورودی‌های دقیق ضروری است. نرم‌افزار برای یک دستیار اطلاعات شخصی، همان‌گونه که از نام آن پیداست، کمک می‌کند تا کاربر اطلاعات شخصی خود مانند اطلاعات تماس، ایمیل و غیره را مدیریت کند. سیستم‌عامل تلفن همراه برای پاسخگویی خوب یا صافی با یک رابط کاربری غنی (UI) از جمله صفحه‌نمایش لمسی و سنسور طراحی شده است.

- **اینترنت:** همراه با توسعه‌ی اینترنت، به‌ویژه پس از وب ۲٫۰، اطلاعات فراوانی در شبکه برای جستجو، سازمان یافتن، استخراج وجود دارد تا به کاربران فرستاده شود. مردم به‌طور فزاینده‌ای به‌جای مرور وب با اینترنت زندگی می‌کنند. بیشتر مردم در توسعه درگیر هستند، از جمله سهم اطلاعات، توسعه‌ی نرم‌افزار و تعاملات اجتماعی. سیستم‌عامل‌های تلفن همراه می‌تواند خود درگیر نشوند، بلکه سیستم‌ها را باز کنند.

مدل استفاده شده در دستگاه‌های تلفن همراه گذشته محدود است. کاربر عمدتاً برنامه‌های کاربردی دستگاه را برای مدیریت داده‌ها و بازی‌های محلی، مرور گاه به گاه صفحات استاتیک وب اینترنت و یا دسترسی به سرویس‌های خاص مانند ایمیل اجرا می‌کند. به‌عبارت دیگر، کاربردهای ممکن دستگاه با برنامه‌های کاربردی از پیش نصب شده زمانی که کاربر آن را خریداری می‌کند از پیش تعریف شده است. این مسئله تا حد زیادی در دستگاه‌های تلفن همراه جدید تغییر کرده است، که در آن دستگاه یک پورتال برای مدل‌های مختلف استفاده است. همه طرف‌های درگیر مانند ارائه‌دهندگان خدمات، توسعه‌دهندگان نرم‌افزار و کاربران دیگر دستگاه‌ها به‌طور مداوم از طریق دستگاه با صاحب آن کمک و تعامل دارند. شکل ۱ تفاوت مدل سطح بالای استفاده شده میان دستگاه‌های گذشته و جدید تلفن همراه را نشان می‌دهد.



شکل ۱: مدل سطح بالای استفاده شده در دستگاه‌های موبایل

- در حال حاضر نمایندگان سیستم‌عامل تلفن همراه شامل اپل در * iOS ۵,۰، گوگل آندروید ۴,۰، مایکروسافت ویندوزفن ۷,۰، و غیره است. در مدل‌های استفاده شده‌ی آنها، آنها شباهت‌ها بیشتری از تفاوت‌ها را به اشتراک می‌گذارند:
- همه آنها یک کیت مستند توسعه نرم‌افزار (SDK) با رابط‌های برنامه کاربردی خوب تعریف شده دارند که توسعه‌دهندگان را قادر به توسعه‌ی برنامه‌های کاربردی رایج برای این سیستم‌ها می‌کند.
 - همه آنها فروشگاه نرم‌افزار آنلاین جهت انتشار برای توسعه‌دهندگان و دانلود برنامه‌های کاربردی برای کاربران، مانند فروشگاه برنامه اپل، گوگل پلی و بازار تلفن ویندوز دارند.
 - همه آنها یک سطح معینی از پشتیبانی گرافیک سه بعدی و چندوظیفه‌ای دارند. تلاش زیادی برای تعامل صاف و پاسخگو به کاربر صرف شده است.
 - مرور تجربه بسیار فراتر از صفحات وب استاتیک می‌باشد. HTML5 در حال تبدیل شد به برنامه‌ای پیش فرض برای اجرای برنامه‌های کاربردی مبتنی بر وب است.
 - همه سیستم‌عامل‌ها از پرداخت مبتنی بر دستگاه پشتیبانی می‌کنند. با توجه به برنامه‌های سازمانی و اطلاعات خصوصی، امنیت سیستم یک نگرانی اصلی برای کاربران دستگاه است.

- با توجه به سیستم‌عامل‌های موبایل، یکی از تفاوت‌های کلیدی طراحی از غیر سیستم‌عامل‌های موبایل، تمرکز بر عمر باتری است. سیستم سعی می‌کند کاهش مصرف انرژی زیادی از اجزای دستگاه داشته باشد و آن‌ها را به زمان‌های بیکاری ممکن نگه دارد.

شباهت سیستم‌عامل‌های تلفن همراه امروزی منعکس‌کننده‌ی روند پیشرفت در سخت‌افزار، نرم‌افزار و اینترنت می‌باشد. پیش‌بینی روند سیستم‌عامل‌های تلفن همراه، به اعتقاد ما از مناطق اصلی تمرکز در طراحی سیستم‌عامل‌های تلفن همراه نسل بعدی، از جمله تجربه کاربر، عمر باتری، آمادگی ابر، امنیت و باز بودن است. درواقع آنها اهداف متضاد زیادی دارند:

- **تجربه‌ی کاربر و عمر باتری:** برای رسیدن به بهترین پاسخ و صافی، سیستم انتظار دارد که تمام منابع سخت‌افزاری در دسترس از بهترین ظرفیت خود بهره‌برداری کنند. در همان زمان، برای حفظ عمر باتری دستگاه تلفن همراه، قطعات سخت‌افزاری باید در هر زمان ممکن آماده به کار باشند.

- **امنیت و باز بودن:** کسی که نمی‌خواهند تمام ویژگی‌های یک سیستم را به اشخاص خارجی افشا کند، به همین دلیل سیستم خود را در تهدید امنیتی قرار می‌دهد. از سوی دیگر، بدون افشای کافی رابط‌های برنامه کاربردی سیستم، ایجاد کاربردهای نوآورانه برای توسعه‌دهندگان غیر ممکن است.

- **آمادگی ابر:** خدمات و برنامه‌های کاربردی بیشتری از طریق ابر ارائه شده است، بنابراین طبیعی است مدل دستگاه کلاینتی که مورد اعتماد ابر باشد و محاسبات آزاد به ابر بسپارد مشخص شود. اما تا به امروز، مدل مشتری هنوز چالش‌های فنی در تجربه کاربر و امنیت دارد.

در این مقاله، ما سعی می‌کنیم تا جنبه‌های مختلف طراحی سیستم‌عامل تلفن همراه و نظرات خود در مورد آینده معماری سیستم‌عامل‌های تلفن همراه را ارائه کنیم.

مقاله به شرح زیر است. بر اساس چارچوب در بخش، ما از بخش‌های جداگانه برای بحث در مورد موضوعات مربوطه در متن زیر استفاده می‌کنیم. آنها در تجربه کاربر مرتب، مدیریت انرژی، امنیت، باز بودن و آمادگی ابر مرتب شده‌اند. در انتها بحث‌ها و خلاصه‌ی مطالب آورده شده است.

تجربه کاربری (UX)

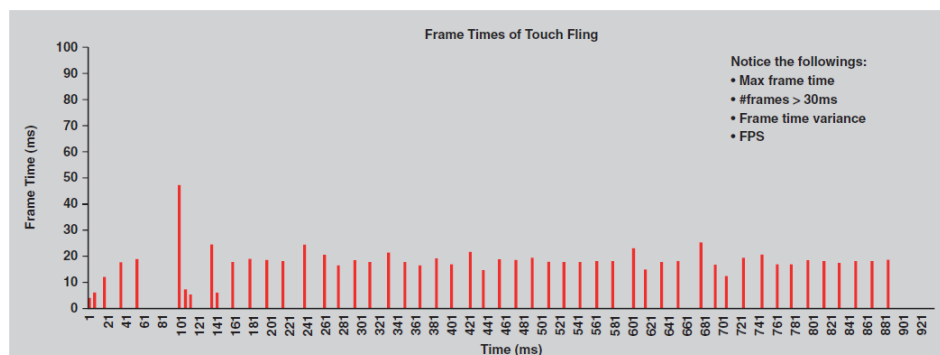
عملکرد سنتی، برای توصیف دستگاه‌های مشتری مدرن ناکافی است. عملکرد بیشتر در مورد وضعیت اجرای مداوم یک پشته نرم‌افزار است و معمولاً با یک نمره‌ی نهایی از توان کل پردازنده و یا زیرسیستم‌های دیگر گزارش می‌شود. تجربه‌ی کاربری بیشتر در مورد انتقال پویا از سیستم توسط ورودی‌های کاربر است. کیفیت تجربه کاربری توسط چنین چیزهایی مانند درک پاسخگویی کاربر، صافی، انسجام و دقت و صحت تعیین می‌شود. عملکرد سنتی می‌تواند هر لینک از زنجیره‌های تعامل با کاربر را اندازه‌گیری کند، درحالی که زنجیره کامل از تعامل با کاربر به‌عنوان یک کل قابل ارزیابی نیست. بنابراین بهینه‌سازی عملکرد سنتی نمی‌تواند به‌سادگی به بهینه‌سازی تجربه کاربری اعمال شود. بنابراین زمان سرمایه‌گذاری در دستگاه بهینه‌سازی تعامل با کاربر برای یک تجربه کاربری لذت‌بخش است.

تعامل کاربر با دستگاه‌های موبایل

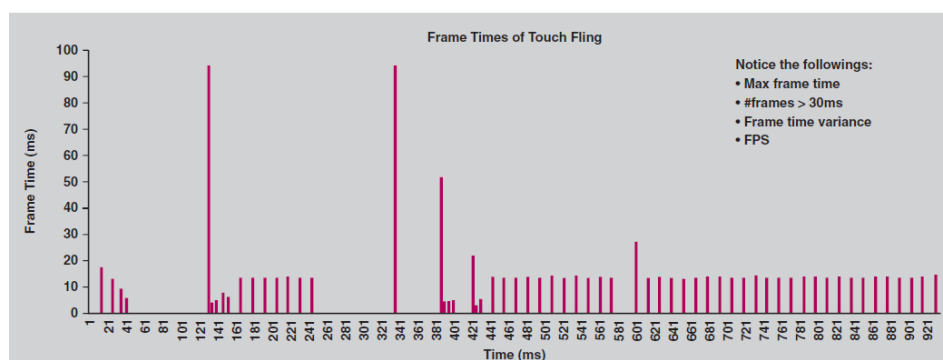
اخیراً در اندازه‌گیری عملکرد با دستگاه‌های آندروید موجود در بازار، یک دستگاه X که به‌طور یکنواخت بدتر از دستگاه Y با معیارهای رایج گرافیک، رسانه‌ها و مرور رفتار می‌کند، یافتیم. اما درک تجربه کاربری با دستگاه X بهتر از دستگاه Y است. دلیل اصلی که مشخص کردیم این است که معیارهای سنتی و یا معیار طراحی در روش‌های سنتی واقعاً تعاملات کاربر را مشخص نمی‌کند، اما قابلیت محاسبات (مانند اجرای دستورات) و یا توان (مانند پردازش دیسک خوانده شده) از سیستم و زیر سیستم را اندازه‌گیری می‌کند.

نگاهی به ارزیابی ویدیویی می‌اندازیم. معیار سنتی تنها عملکرد پخش ویدئو با برخی از معیارها مانند FPS (فریم در ثانیه) و یا میزان افت فریم را اندازه‌گیری می‌کند. این روش دارای حداقل دو مشکل در ارزیابی تجربه کاربری است. مشکل اول این است که پخش ویدئو تنها بخشی از تعامل کاربر در بازی‌های ویدئویی است. چرخه معمولی زندگی تعامل با کاربر معمولاً شامل حداقل لینک‌های زیر است: "راه‌اندازی بازیکن" "شروع به بازی" "دنباله پیشرفت" "پخش ویدئو" "بازگشت به صفحه اصلی". با این حال عملکرد خوب در پخش ویدئو می‌تواند تجربه کاربری واقعی را در بازی‌های ویدئویی مشخص کند. ارزیابی تعامل با کاربر یک مجموعه از سنجش سنتی عملکرد است.

مشکل دیگر این است که، استفاده از FPS به عنوان کلید متریک برای ارزیابی صافی از تعاملات کاربران همیشه نمی تواند منعکس کنندهی تجربه کاربری خوب باشد. به عنوان مثال، زمانی که ما یک تصویر در نرم افزار Gallery3D ارائه می کنیم، دستگاه Y لکنت آشکاری در طول پیمایش تصویر دارد، اما ارزش FPS دستگاه Y بالاتر از دستگاه X بود. به منظور بیان تفاوت دو دستگاه، داده ی هر فریم در طول یک تصویر تحت عملیات نرم افزار Gallery3D در هر دو دستگاه X و Y، به ترتیب در شکل ۲ و شکل ۳ نشان داده شده است. داده های هر قاب در یک نوار عمودی داده شده است، که در آن محور X زمانی است که قاب کشیده شده است، و ارتفاع نوار، زمانی است که طول می کشد سیستم قاب را رسم کند. از آمار و ارقام، می توانیم ببینیم که دستگاه X مقدار FPS پایین تری نسبت به دستگاه Y، اما با کمترین حداکثر زمان قاب، قاب کوچکتری از ۳۰ ms و واریانس زمان قاب کمتری دارد. این به این معنی است که، برای مشخص کردن تجربه کاربران از عملیات پرت کردن تصویر، متریک هایی مانند حداکثر زمان قاب و واریانس زمان قاب نیز باید در نظر گرفته شود.

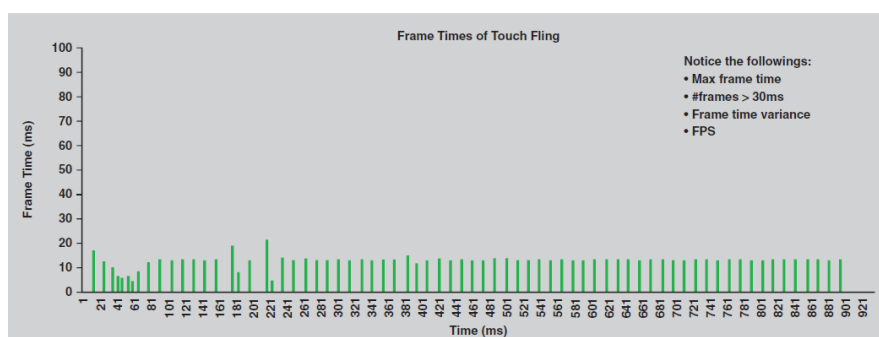


شکل ۲: زمان قاب ها از عملیات پرت کردن در نرم افزار Gallery3D در دستگاه X



شکل ۳: زمان قاب ها از عملیات پرت کردن در نرم افزار Gallery3D در دستگاه Y

به عنوان یک مقایسه، شکل ۴ قاب داده‌ها از یک عملیات پرت کردن بعد از بهینه‌سازی دستگاه Y را نشان می‌دهد. ظاهراً همه معیارها بهبود یافته است و زمان توزیع قاب خیلی بیشتر یکنواخت شده است. تجربه‌ی کاربری در مورد انتقال حالت، پویاتر از سیستم توسط ورودی‌های کاربر است. یک تجربه کاربری خوب با چیزهایی مانند درک پاسخگویی کاربران، صافی، انسجام و دقت به دست می‌آید. عملکرد سنتی می‌تواند هر لینک از زنجیره‌ی تعاملات با کاربر را بدون ارزیابی زنجیره‌ی کامل از تعاملات با کاربر به عنوان یک کل اندازه‌گیری کند.



شکل ۴: زمان قاب‌ها از عملیات پرت کردن در نرم‌افزار Gallery3D در دستگاه Y بعد از بهینه‌سازی

نکته‌ی مهم دیگر این است که تجربه کاربری یک فرآیند ذهنی است؛ فقط تجربه را هنگام تماشای یک فیلم و یا موسیقی قدردانی در نظر می‌گیرد. تحقیقات دانشگاهی در حال حاضر، با استفاده از روش‌های مختلف مانند ردیابی کره چشم، نظارت ضربان قلب و یا فقط رای‌گیری به درک تجربه کاربری می‌رسند. برای هدف مهندسی نرم‌افزار ما، به منظور تجزیه و تحلیل و بهینه‌سازی تعاملات نظام‌مند کاربران، حالات تعامل را به چهار نوع دسته‌بندی می‌کنیم:

- **ورودی دستگاه از کاربران، سنسور، شبکه و غیره:** این دسته ارزیابی می‌کند که آیا ورودی می‌تواند دستگاه را به دقت عمل همانطور که انتظار می‌رود وادارد. برای ورودی صفحه نمایش لمسی، سرعت لمس، فشار، محدوده و غیره را اندازه‌گیری می‌کند.
- **پاسخ دستگاه به ورودی:** این دسته چگونگی پاسخگویی دستگاه به ورودی را ارزیابی می‌کند.
- **انتقال حالت سیستم:** این دسته به خصوص چگونگی انتقال صاف گرافیک بر روی صفحه نمایش را ارزیابی می‌کند. که می‌تواند یک پیگیری از پاسخ دستگاه به برخی از ورودی‌ها باشد.

- **کنترل مداوم دستگاه:** مردم این دستگاه را نه تنها برای یک ورودی، گاهی اوقات برای کنترل اشیاء گرافیکی در صفحه نمایش، مانند کنترل یک جت هواپیما بازی و یا کشیدن یک آیکون نرم افزار به کار می برند. این دسته برای ارزیابی کنترل دستگاه است.

در میان آنها، "ورودی به دستگاه" و "کنترل دستگاه" مربوط جنبه‌ی تجربه کاربری از چگونگی کنترل یک دستگاه توسط کاربر است. "پاسخ دستگاه به ورودی" و "انتقال حالت سیستم" از جنبه‌های مرتبط به چگونگی واکنش دستگاه به کاربر را نشان می دهد. ما می توانیم یک چرخه زندگی تعامل با کاربر به حالات نقشه نگاشت کنیم که در دسته‌های بالا قرار گیرد. سپس برای هر سناریو، می توانیم معیارهای کلیدی در پشته نرم افزار برای اندازه گیری و بهینه سازی را شناسایی کنیم.

بهینه سازی تعاملات با کاربر

همانطور که در آخرین بخش شرح داده شد، هیچ روش روشن و عینی برای اندازه گیری تجربه کاربری وجود ندارد. ما معیارهای زیر را برای اندازه گیری تعاملات کاربر ارائه کردیم:

- **درک:** متریک توسط انسان قابل درک است. در غیر این صورت، با تجربه کاربری بی ربط است.
- **قابل اندازه گیری:** متریک باید توسط تیم های مختلف قابل اندازه گیری باشد. نباید به برخی از زیرساخت های خاص که فقط توسط تیم خاصی قابل اندازه گیری هستند بستگی داشته باشند.
- **تکرار:** نتیجه ی اندازه گیری باید در اندازه گیری های مختلف تکرار شود. انحرافات بزرگ در اندازه گیری به معنی یک متریک بد است.
- **مقایسه:** داده های اندازه گیری شده باید در سیستم های مختلف قابل مقایسه باشند. مهندسين نرم افزار می توانند از متریک ها برای مقایسه ی سیستم های مختلف استفاده کنند.
- **معقول:** متریک باید به دلیل علیت رفتار پشته نرم افزار کمک کند. به عبارت دیگر، متریک باید به رفتار نرم افزار نگاشته شود و ممکن است بر اساس اجرای پشته نرم افزار محاسبه شود.

- **قابل اثبات:** متریک می‌تواند برای بررسی بهینه‌سازی استفاده شود. نتیجه‌ی اندازه‌گیری قبل و بعد از بهینه‌سازی باید تغییر تجربه کاربری را نشان دهد.

- **Automatable:** برای اهداف مهندسی نرم‌افزار، انتظار داریم که متریک تا حد زیادی بی‌مراقبت اندازه‌گیری کند. این امر به‌ویژه در آزمون رگرسیون و یا آزمون قبل از سپردن مفید است. این معیار به دلیل آن که به‌طور مستقیم به تجزیه و تحلیل تجربه کاربری و بهینه‌سازی مربوط نیست، به شدت لازم نیست.
بنا به هدایت با معیارهای اندازه‌گیری، ما به جنبه‌های تجربه کاربری تمرکز می‌کنیم، چگونه یک کاربر یک دستگاه را کنترل می‌کند و چگونه یک دستگاه به یک کاربر واکنش نشان می‌دهد. چگونه یک کاربر یک دستگاه را که به‌طور عمده دو مورد اندازه‌گیری دارد کنترل می‌کند:

- **دقت/فازی:** این که چگونه دقت، ابهام، وضوح تصویر، و محدوده توسط سیستم برای ورودی صفحه نمایش، حسگرها و منابع دیگر لمسی پشتیبانی می‌شود، ارزیابی می‌شود. به عنوان مثال، چه سطح فشاری توسط سیستم پشتیبانی می‌شود، چگونه مختصات حوادث لمسی نمونه، نزدیک به آهنگ حرکت نوک انگشتان بر روی صفحه نمایش هستند، چه تعداد از انگشتان دست را می‌توان در زمان یکسانی به کار برد و غیره.

- **هماهنگی:** فاصله‌ی تاخیر کشیدن بین نوک انگشتان و شی گرافیکی کشیده در صفحه نمایش ارزیابی می‌شود. همچنین انسجام بین عملیات کاربران و اشیاء تحت کنترل سنسور، مانند تفاوت درجه زاویه بین کنترل کج جریان آب و دستگاه زاویه مورب ارزیابی می‌شود.

چگونگی واکنش یک دستگاه به یک کاربر همچنین دارای دو حوزه اندازه‌گیری است:

- **پاسخگویی:** زمان بین یک ورودی که به دستگاه تحویل داده می‌شود و واکنش دستگاه ارزیابی می‌کند. همچنین شامل زمان صرف شده برای به پایان رساندن عمل نیز است.

- **صافی یا همواری:** این حوزه صافی گذار گرافیکی با حداکثر زمان قاب، واریانس قاب زمان، FPS، نرخ رها کردن فریم و غیره را ارزیابی می‌کند. همانطور که بحث شد، FPS به تنهایی نمی‌تواند به دقت منعکس کننده‌ی تجربه کاربران در مورد صافی باشد.

برای این چهار حوزه اندازه‌گیری، یک متریک بتن برای استفاده مشخص می‌کنیم، ما نیاز به درک چگونگی ربط تجربه کاربری "خوب" به این متریک هستیم. از آنجا که تجربه کاربری یک موضوع ذهنی است که به شدت به وضعیت فیزیولوژیکی بودن انسان و سلیقه شخصی آن بستگی دارد، بنابراین هیچ نتیجه‌ی علمی در مورد ارزش یک متریک به‌منزله‌ی یک تجربه کاربری "خوب" وجود ندارد. برای این موارد، ما فقط ارزش تجربه در صنعت را لحاظ می‌کنیم. جدول ۱ چند نمونه از ارزش تجربه در صنعت را نشان می‌دهد.

	Best	Good	Acceptable
Response delay	≤ 100 ms	≤ 200 ms	≤ 500 ms
Graphics animation	≥ 120 fps	≥ 60 fps	≥ 30 fps
Video playback	≥ 60 fps	≥ 30 fps	≥ 20 fps

جدول ۱: مثال ارزش تجربه در صنعت برای تجربه‌ی کاربری

با توجه به ماهیت انسان، دو یادداشت برای مهندسين نرم‌افزار با توجه به بهینه‌سازی تجربه کاربری وجود دارد که به آن بپردازند.

ارزش یک متریک معمولا دارای یک طیف برای تجربه کاربری "خوب" است. ارزش "بهتر" از محدوده لزوما تجربه کاربری "بهتر" را به‌همراه ندارد.

ارزش در اینجا تنها دستورالعمل خشن برای موارد مشترک با انسان معمولی است. به‌عنوان مثال، یک بازیکن بازی ممکن است با 120 فریم انیمیشن در ثانیه راضی نشود. از سوی دیگر، یک کارتون با طراحی خوب یک صافی مناسب با ۲۰ فریم انیمیشن در ثانیه را به‌همراه دارد.

در حال حاضر ما می‌توانیم روشمان را برای بهینه‌سازی تجربه کاربری به‌کار ببریم. که می‌تواند در مراحل زیر خلاصه شود.

مرحله ۱: دریافت رؤیت تجربه کاربری از کاربران و یا شناسایی مسائل تعامل با عملیات دستی. این مسئله می‌توان با کمک دوربین‌های با سرعت بالا و یا دیگر روش‌های موجود مشخص شود.

مرحله ۲: تعریف سناریوهای پشته نرم‌افزار و معیارهایی که موضوع تجربه کاربری را به نشانه نرم‌افزار تبدیل می‌کنند.

مرحله ۳: توسعه‌ی حجم کار نرم‌افزار برای تکثیر دوباره‌ی موضوع به‌روش اندازه‌گیری و تکرار. حجم کار، مقادیر متریک را که منعکس‌کننده تجربه کاربران است گزارش می‌کند.

مرحله ۴: از حجم کار و ابزار مرتبط به تجزیه و تحلیل و بهینه‌سازی پشته نرم‌افزار استفاده می‌کند. حجم کار بهینه‌سازی را نیز تایید می‌کند.

مرحله ۵: گرفتن بازخورد از کاربران و سعی بیشتر برای بهینه‌سازی برنامه‌های کاربردی برای تایید تجربه کاربری بهبود یافته.

بر اساس این روش، یک مجموعه حجم کار آندروید (AWS) [۳۳] توسعه دادیم که تقریباً شامل تمام موارد استفاده معمولی از دستگاه‌های آندروید است. همچنین یک ابزار به نام Uxtune [۳۴] که به تجزیه و تحلیل تعامل با کاربر در پشته نرم‌افزار کمک می‌کند توسعه دادیم. متفاوت از عملکرد سنتی ابزار تنظیم، Uxtune با حوادث قابل مشاهده کاربر و سیستم حوادث کم سطح ارتباط برقرار می‌کند. به عنوان گام بعدی، ما در حال انتقال کار از آندروید به دیگر سیستم‌ها هستیم.

طراحی سیستم عامل تلفن همراه برای تجربه کاربری

بر اساس تجربه ما با آندروید، به بهینه‌سازی UX به نحوی مشابه با بهینه‌سازی موازی نرم‌افزار، تنها با پیچیدگی بیشتر، بنا به چهار دلیل زیر تمرکز می‌کنیم:

- UX شامل قطعات سخت‌افزاری متعدد و چند فرآیند نرم‌افزاری و اثر متقابل آنها است.
- UX در یک دستگاه مشتری به منظور مصرف برق است، بنابراین UX شامل عمر باتری و دستگاه درجه حرارت است.
- UX به زمان‌بندی دقیق نیاز دارد، مانند صافی که در آن کاربر انتظار هیچ واریانس زمان قاب ندارد. نه سریعتر و نه کندتر قابل قبول است. هدف قرار دادن زمان دقیق مورد نیاز است. این بیشتر شبیه به یک زمان واقعی مورد نیاز است.

- UX شامل برخی از عوامل ذهنی است که در طراحی سیستم عامل تلفن همراه مشخص شده است، از جمله اینکه آیا برخی از انیمیشن ها فقط یک اشاره برای UX ضروری است و آیا سیستم می تواند برخی از فریم ها را برای پاسخ بهتر رها کند.

یک درس مهم که از تجربه ما بدست آمد درک مسیر انتقادی از عملیات خاصی است. متفاوت از تنظیم برنامه موازی، طراحی سیستم عامل تلفن همراه همیشه هماهنگ سازی صریح بین اجزای سخت افزاری و موضوعات نرم افزار را ندارد. برای مثال:

- هر نرم افزار از یک حلقه رویداد برای رسیدگی به درخواست ها استفاده می کند. هنگامی که یک نخ A درخواستی به نخ B می فرستد، ممکن است به طور مستقیم به تابع استناد نکند، اما به جای آن یک پیام به نخ B ارسال کند. سپس پیام در صف حلقه ی نخ B برای به کار برده شدن منتظر می ماند. پس از آن از کنترل تماس گیرنده خارج است که رویداد با چه سرعتی می تواند به کار گرفته شود اگر رویدادهای متعددی در صف وجود داشته باشند.

- مثال دیگر زمانی است که نخ A دنباله ای از عملیات را اجرا می کند و پس از آن یک پیام، به نخ B برای اقدامات و پیگیری پاسخ به کاربر ارسال می کند. همه ی دنباله عملیات نباید توسط نخ A به ترتیب انجام شود. برای مثال، می تواند یک پیام به نخ B ارسال کند به طوری که نخ B تا می تواند به کاربر زودتر پاسخ دهند.

نکته مهم این است که، با تجربه کاربری، برخلاف عملکرد بهینه سازی سنتی، سرعت عمل لزوما بهتر نیست. زمانی که سیستم به بهترین درک از پاسخ کاربر می رسد، قدم بعدی بهینه سازی آهسته تر بهتر در محدوده بهترین درک است. به عنوان مثال، اگر یک بازی ۶۰ فریم در ثانیه و بدون مشکل باشد، پس از آن سیستم عامل تلفن همراه باید سعی در بهینه سازی استفاده از CPU و فرکانس پردازنده تا جایی که ممکن است داشته باشد. ما باید همیشه دو مورد را (سریع تر بهتر است و آهسته تر است) به وضوح تشخیص دهیم. تکنیک های بهینه سازی برای دو حالت می تواند کاملاً متفاوت باشد.

هنگامی که چندین هسته و پردازنده معرفی شد، دو مورد فوق بیشتر آشکار شد. هسته های بیشتر به سناریوهای "سریع تر بهتر" کمک کرد، اما به عمر باتری برای سناریوهای "آهسته تر بهتر" صدمه زد. سیستم عامل موبایل مجبور به

روشن و خاموش کردن هسته با یک سیاست هوشمند است زیرا تاخیر روشن/خاموش کردن می تواند بیش از یک عملیات لمسی باشد (معمولا در طیف وسیعی از ۱۰۰ میلی ثانیه به ۵۰۰ میلی ثانیه).

برای تنظیم عملکرد برنامه موازی، مردم "باز پخش" را در اشکال زدایی مفید دانستند. که معمولا یک پاسخ چند نخی، است که، در داخل یک فرآیند است. برای UX، فعل و انفعالات در طول اجرای فرآیندها از طریق IPC، و بین CPU، GPU، و کنترلر صفحه نمایش (DC)، یک طیف گسترده ای از کارهای سیستم است. این روش پخش سنتی نمی تواند کمک کند.

مدیریت انرژی (PM)

مدیریت انرژی همواره یک چالش کلیدی برای طراحان سیستم عامل تلفن همراه با حرکت رو به جلو بوده و خواهد بود. خواسته های انرژی به سرعت و بر روی دستگاه های تلفن همراه در حال افزایش است. با این حال، رشد ظرفیت باتری با توجه به توسعه ای آهسته فن آوری باتری و این واقعیت که مردم دستگاه هایی جمع و جورتر می خواهند هرگز نمی تواند متوقف شود. مدیریت انرژی به یک مشکل پیچیده بر روی دستگاه های تلفن همراه تبدیل شده است و یک رویکرد جامع برای رسیدگی به آن مورد نیاز است.

مدیریت توان پردازنده

سیستم عامل تلفن همراه پیشرفت مداومی در مدیریت توان در دهه گذشته داشته است. در ابتدا کار تمرکز بر مدیریت توان سیستم عامل تلفن همراه در مدیریت قدرت پردازنده بود زیرا پردازنده یک مصرف کننده مهم در پلت فرم توان بوده است. پردازنده های مدرن از فرکانس پویا و پویا پویا و پویا شدن ولتاژ مانند افزایش فناوری SpeedStep® اینتل پشتیبانی می کنند. از جمله قابلیت های پردازنده، فعال کردن سیستم عامل تلفن همراه برای تنظیم فرکانس پردازنده و/یا ولتاژ به صورت پویا در زمان اجرا بر اساس تقاضا از قدرت محاسباتی مورد نیاز حجم کار است که در حال حاضر بر روی پردازنده ها در حال اجرا است. این موجب صرفه جویی مقدار قابل توجهی از قدرت پردازنده فعال می شود، زیرا مصرف قدرت متناسب با مربع ولتاژ فرکانس هسته است. زیر سیستم CpuFreq در کرنل لینوکس یک مثال از مدیریت قدرت پردازنده در وضعیت فعال است. علاوه بر فرکانس پویا و پویا پویا شدن ولتاژ، پردازنده های مدرن معمولا از

پردازنده‌های متعدد بیکار با تفاوت مقدار برق مصرفی پشتیبانی می‌کنند. سیستم‌عامل موبایل می‌تواند پردازنده را به‌طور مستقیم به حالت مناسب غیرفعال براساس پیش‌بینی بیکاری و محدودیت‌های QoS مطرح شده توسط دیگر زیرسیستم‌ها و فضای کاربر وارد کند. زیر سیستم cpuidle لینوکس یک مثال از قدرت مدیریت یک پردازنده در وضعیت غیرفعال است.

دستگاه مدیریت انرژی

تمرکز کار مدیریت انرژی سیستم‌عامل تلفن همراه به دستگاه مدیریت انرژی منتقل می‌شود. به‌طور خاص، یک مکانیسم برای مدیریت قدرت دستگاه‌های I/O در زمان اجرا معرفی شده است. مدیریت انرژی در زمان اجرا برای دستگاه‌های I/O می‌تواند به‌طور خودکار دستگاه‌های I/O را به مناطق با انرژی کم زمانی که دستگاه‌های مربوطه در زمان اجرا بیکار شناسایی می‌شوند هدایت کند. علاوه بر مدیریت انرژی دستگاه‌های I/O درحالی که آن‌ها بیکار هستند، برخی از نوآوری‌های فناوری برای صرفه‌جویی I/O زمانی که فعال هستند وجود دارد. به‌عنوان مثال، GPU های مدرن شروع به حمایت از ولتاژ پویا و فرکانس بر روی پردازنده می‌کنند. GPU ولتاژ پویا و فرکانس می‌تواند مصرف انرژی را به میزان 50 درصد برای گرافیک‌های ۳ بعدی در برخی موارد کاهش دهد. علاوه‌براین، دستگاه‌های O / A درحال تبدیل شدن هستند به این معنا که آن‌ها می‌توانند خودبه‌خود بدون مداخله CPU کار کنند. به‌عنوان مثال، فن‌آوری‌هایی مانند panel self-refresh می‌توانند مقدار قابل توجهی از انرژی را درحالی که تصویر در مواردی ثابت است مانند زمانی که کاربر در حال خواندن یک کتاب در یک دستگاه تلفن همراه است صرفه‌جویی کنند. پانل صفحه‌نمایش می‌تواند در این مورد رندر را از حافظه محلی خود نگه دارد و بسیاری از قطعات سخت‌افزاری به طور سنتی کار کنند درحالی که صفحه نمایش می‌تواند خاموش باشد، از جمله پردازنده، حافظه، موتور صفحه نمایش و پورت‌ها.

موارد مدیریت انرژی سیستم‌عامل‌های موبایل

آندروید قبل از معرفی زیرساخت‌های دستگاه‌های مدیریت انرژی زمان اجرا به هسته لینوکس محبوب شد و با رویکرد دیگری به نام opportunistic suspend به منظور رسیدن به هدف افزایش عمر باتری در دستگاه‌های آندروید ارائه

شد. بدون قابلیت دستگاه‌های مدیریت انرژی زمان اجرا، اندروید تلاش می‌کند هر زمان که هیچ کار جالبی وجود ندارد سیستم را به حالت تعلیق درآورد.

ویندوز ۸ یک سیستم انرژی جدیدی به نام **connected standly** معرفی کرده است. بر خلاف حالت آماده به کار S3 سنتی، که تمام فعالیت‌های سیستم را متوقف می‌کرد، سیستم هنوز در یک منطقه‌ی انرژی بسیار پایینی در حال اجرا است و این به کاربران امکان می‌دهد تا با آخرین اطلاعات مانند ایمیل به روز باشند. ویندوز ۸ آماده به کار بر روی پردازنده مدیریت انرژی و فن‌آوری‌های دستگاه‌های مدیریت انرژی مبتنی بر زمان اجرا است.

بهداشت نرم‌افزار چالش برانگیزترین مشکل برای هر دو رویکرد مدیریت انرژی بر اساس **suspend** و بیکار است و عمر باتری بر بسیاری از رفتارهای نرم‌افزار بر روی سیستم بستگی دارد. یک مطالعه اخیر نشان می‌دهد که برنامه‌های کاربردی رایگان آندروید، ۷۵ درصد از انرژی را بر روی تبلیغات با نگه داشتن یک **wakelock** در پس زمینه تلف می‌کنند، در نتیجه سیستم به حالت تعلیق درمی‌آید. در کل این برای ویندوز ۸ درست است، که در آن نرم‌افزار با هدر دادن انرژی مشغول می‌ماند و بدون هیچ دلیل خوبی کل سیستم از ورود انرژی آماده به کار جلوگیری می‌کند. بعضی از مردم انتظار دارند که مدیریت سیستم انرژی حتی در چنین برنامه‌های کاربردی با معرفی مکانیزم‌هایی قادر به جلوگیری باشد، در حالی که دیگران فکر می‌کنند که قرار دادن چنین مکانیسم‌هایی در محل تنها به منجر گسترش برنامه‌های کاربردی بد رفتار خواهد شد.

باز بودن

یکی روندهای عمده‌ی سیستم‌عامل تلفن همراه باز بودن آن است. در این زمینه، باز بودن سیستم‌عامل تلفن همراه به معنی سطحی از فرصت و آزادی است که مردم مجبور به استفاده، کمک، سفارشی کردن و نوآوری برای سیستم‌عامل تلفن همراه برای اهداف خود هستند. در حال حاضر کاری وجود دارد [۶] که به مسئله‌ی باز بودن از دیدگاه توسعه‌دهندگان پرداخته است. این تحقیق جنبه‌های تصمیم ادراک توسعه‌دهندگان از پلت فرم باز بودن را شناسایی می‌کند. در اینجا ما روند باز بودن از دیدگاه اکوسیستم را مورد مطالعه قرار می‌دهیم، از آنجا که ما معتقدیم مسائل باز بودن برای فعال کردن و پرورش اکوسیستم تلفن همراه ضروری است.

باز بودن به کاربران اکوسیستم‌های تلفن همراه

کاربران اکوسیستم‌های تلفن همراه شامل تولیدکنندگان (OEM ها) که دستگاه‌های تلفن همراه را تولید و به فروش می‌رسانند، ارائه‌دهندگان خدمات (اپراتور) که شبکه اتصال و سایر خدمات ارزش افزوده را ارائه می‌کنند، مصرف‌کنندگان (کاربران نهایی)، ISV ها که برنامه‌های کاربردی تجاری را توسعه می‌دهند و توسعه‌دهندگان که برنامه‌های کاربردی را توسعه می‌دهند و در توسعه سیستم‌عامل تلفن همراه کمک می‌کنند و تکامل می‌دهند اگر سیستم‌عامل تلفن همراه منبع باز باشد.

باز بودن سیستم‌عامل تلفن همراه به معنی چیزهای مختلف به بازیکنان مختلف در اکوسیستم تلفن همراه است. برای اپراتورها، باز بودن سیستم‌عامل تلفن همراه تعیین می‌کند که چگونه خدمات آسان آنها را می‌تواند منتقل شود، مهاجرت کند، مستقر شود، و هموار در سراسر دستگاه‌ها اجرا شود. برای تولیدکنندگان دستگاه‌های تلفن همراه، باز بودن تعیین می‌کند که آنها چقدر می‌توانند سیستم‌عامل تلفن همراه خود را برای اجرا در سراسر سیستم‌عامل و افتراق دستگاه‌های خود از دیگران سفارشی کنند و حتی مهمتر، باز بودن تعیین می‌کند که چگونه می‌توان دستگاه‌های سازگار با تجربه کاربری ساخت. برای ISV و جامعه توسعه‌دهندگان، باز بودن تعیین می‌کند که چگونه می‌توان برنامه‌های جدید آنها را با خلاقیت خود توسط ایده‌ها توسعه داد و چگونه می‌توان سرمایه‌گذاری خود را در توسعه نرم‌افزار از طریق برنامه‌نویسی یک‌بار در حال اجرا در سراسر دستگاه به حداکثر رساند. برای کاربران نهایی و یا مصرف‌کنندگان دستگاه‌های تلفن همراه، باز بودن به معنی این است که چگونه می‌توان برنامه‌های کاربردی بیشتری، مانند برنامه‌های کاربردی دانلود شده از فروشگاه‌ها، بدون نگرانی بیش از حد در مورد تناقض از تجربه کاربری و ناسازگاری برنامه‌های کاربردی در سراسر دستگاه استفاده کرد. باز بودن همچنین ممکن است شانس مردم برای شرکت در توسعه و تکامل سیستم‌عامل تلفن همراه خود را در طول چرخه زندگی سیستم‌عامل تلفن همراه افزایش دهد.

تکامل باز بودن سیستم‌عامل موبایل

چند سال پیش، بیشتر دستگاه‌های تلفن همراه گوشی بودند، و بیشتر افراد از تلفن برای تماس‌های صوتی، به عنوان یک دفترچه تلفن، و پیام استفاده می‌کردند. برای مصرف‌کنندگان، برنامه‌های کاربردی که آنها می‌توانستند بازی کنند

محدود به برنامه‌های کاربردی ساخته شده در دستگاه‌ها بود زمانی که دستگاه از کارخانه خارج می‌شد. برای توسعه‌دهندگان نرم‌افزار و یا ISV های شخص ثالث، دسترسی به هر سطح از کد منبع بدون قرارداد با صاحب سیستم‌عامل وجود نداشت. مانند سیستم‌عامل تلفن همراه که یک سیستم کاملاً بسته بود و معمولاً متعلق به سازنده تلفن همراه بود. بنابراین اپراتورها، مجبور به همکاری نزدیک با توسعه‌دهندگان سیستم‌عامل بودند تا خدمات خود را ارائه کنند، چرا که تنها کسانی که سیستم‌عامل تلفن همراه را توسعه می‌دادند چگونگی توسعه برنامه‌های کاربردی را می‌دانستند.

بعد از آن که، تلفن همراه شروع به انتقال به گوشی‌های تلفن هوشمند کرد، مردم انتظار داشتند تلفن‌های هوشمند قادر به انجام کار بیشتری، مانند مرور وب و پخش موسیقی/ویدئو و نه تنها برقراری تماس، ذخیره‌سازی اطلاعات تماس و ارسال پیام‌های متنی باشند. برای تشویق بیشتر توسعه‌دهندگان به منظور توسعه برنامه‌های کاربردی بیشتر به مردم نیاز بود، سازندگان سیستم‌عامل تلفن همراه به سادگی مجموعه‌ای از رابط‌های برنامه کاربردی و ابزارهای مرتبط مانند SDK ها را ارائه می‌کنند، به طوری که مردم می‌توانند انواع برنامه‌های کاربردی برای دستگاه‌های تلفن همراه را توسعه دهند. با چنین باز بودن، توسعه‌دهندگان نرم‌افزار آزادی توسعه برنامه‌های کاربردی را برای سیستم‌عامل‌های تلفن همراه به دست آوردند، پس از آن خرید و نصب برنامه‌های بیشتر برای مصرف‌کنندگان به جای اینکه به برنامه‌های کاربردی از پیش ساخته شده محدود باشند ممکن شد. از آنجا که توسعه‌دهندگان نرم‌افزار و مصرف‌کنندگان از باز بودن بهره بردند، ارائه مجموعه‌ای از API و SDK برای اکثر سیستم‌عامل تلفن همراه تقریباً تبدیل به یک "باید" شد. iOS اپل یکی از نمونه‌های بزرگ سیستم‌عامل تلفن همراه است که چنین سطحی از باز بودن را ارائه می‌کند. در سال‌های اخیر بیشتر سیستم‌عامل‌های تلفن همراه علاوه بر ارائه رابط‌های برنامه کاربردی و SDK با کد منبع عمومی ساخته می‌شوند. هر کسی می‌تواند فرصتی برای مشاهده همه کد منبع داشته باشد، با کد همکاری کند، سیستم‌عامل تلفن همراه خود را تکامل و سفارشی کند. در مقایسه با سطح باز بودن فقط با ارائه رابط‌های برنامه کاربردی و SDK، در سیستم‌عامل تلفن همراه منبع باز می‌توان از آزادی بیشتری برخوردار بود. برای سازندگان دستگاه تلفن همراه، ممکن است آزادی ساخت سیستم‌عامل تلفن همراه خود بر اساس سیستم‌عامل متن باز بر روی سیستم‌عامل خود را در سراسر

دستگاه‌های اجرا داشته باشند. برای اپراتورها، آنها می‌توانند به راحتی خدمات خود را در سراسر دستگاه‌های در حال اجرا سیستم‌عامل منبع باز و انواع آن ساخته و گسترش دهند. برای توسعه‌دهندگان، سیستم‌عامل تلفن همراه منبع باز همه چیز برای ساخت راحت برنامه‌ها را فراهم می‌کند. در نهایت کاربران نهایی دستگاه تلفن همراه می‌توانند از این سطح از باز بودن بهره‌مند شوند، همان‌گونه که انتخاب‌های بیشتری از برنامه‌ها و دستگاه‌ها را برای اجرای برنامه‌های کاربردی انتخاب می‌کنند. در نهایت، هرکسی آزادی شرکت برای تکامل و شکل دادن به سیستم‌عامل تلفن همراه منبع باز را دارد. سیستم‌عامل آندروید یکی دیگر از نمونه‌های بزرگ از سیستم‌عامل تلفن همراه منبع باز است. موفقیت عالی و رشد چشمگیر آن در بازار تلفن‌های هوشمند در طول چند سال گذشته نشان می‌دهد که صنعت چگونه موفق بوده است و به‌عنوان یک سیستم‌عامل تلفن همراه منبع باز در حال رشد با چه سرعتی در حال رشد است.

به‌طور خلاصه، باز بودن سیستم‌عامل‌های تلفن همراه آینده یکی از عوامل کلیدی برای ارتباط دوستانه‌ی سیستم‌عامل تلفن همراه است، به‌خصوص برای توسعه‌دهندگان نرم‌افزار و مصرف‌کنندگان جذاب است. باز بودن سیستم‌عامل نیاز به محاسبات زنجیره‌ای دارد، که انتظار می‌رود نرم‌افزار برای یک بار ساخته شده و در همه جا اجرا شود با تجربه کاربری به کاربران نهایی سازگار باشد.

آمادگی ابر

ابر به‌طور گسترده‌ای توسط کاربران تلفن همراه استفاده می‌شود و بسیاری از سرویس‌های ابری به‌عنوان وب سایت‌ها معرفی می‌شود و توسط مرورگر ارائه شده در تلفن همراه قابل دسترسی است. بیشتر خدمات ابر از طریق برنامه‌های کاربردی وب ارائه می‌شود، که از یک فروشگاه نرم‌افزار نصب شده و مانند برنامه‌های کاربردی بومی بر روی سرویس‌گیرنده تلفن همراه اجرا می‌شود.

قابلیت HTML5

قابلیت HTML5 برای برنامه‌های کاربردی وب برای ادغام خدمات ابر و برای ارائه یک تجربه کاربری خوب ضروری است.

موتورهای وب استفاده می‌شود: ما موتورهای طرح وب و موتورهای جاوا اسکریپت را که بر روی مرورگرهای تلفن همراه استفاده می‌شوند ذکر می‌کنیم. کروم* می‌تواند مرورگر پیش فرض آندروید شود و موفقیت خود را از کامپیوتر به تلفن همراه ببرد. WebKit در iOS و اندروید استفاده می‌شود.

	Layout engine	JavaScript engine	Score + Bonus
Safari on iOS 5.1	Webkit	Squirrel Fish Extreme (SFX)	324 + 9
Android browser on Android 4.0	Webkit	V8	273 + 3
IE on Windows Phone 8	Trident	Chakra	300 + 6
Tizen 1.0	Webkit	SFX	400 + 15 *
Chrome Beta on Android 4.0	Webkit	V8	364 + 10
Opera Mobile 12	Presto	Carakan	369 + 11
Firefox Mobile 10	Gecko	SpiderMonkey	325 + 9

وب سایت آزمایشی HTML5 [۹] از مرورگرهای مختلف در دستگاه‌های تلفن همراه پشتیبانی می‌کند. می‌توانیم در جدول ۲ ببینیم که سیستم عامل iOS، اندروید و ویندوز فن قابلیت مرورگر خود را برای پشتیبانی از HTML5 بهبود می‌بخشند. گوگل کروم بر روی آندروید ۴,۰ کار می‌کند و جاه طلبی خود را در مرورگر سیستم عامل‌های تلفن همراه نشان می‌دهد. Tizen، از شرکت کنندگان جدید در کمپین سیستم عامل تلفن همراه بالاترین نمره را در اولین کنفرانس دستگاه‌های توسعه در توسعه Tizen از آن خود کرد. به راحتی می‌توان مسابقه شدید از پشتیبانی برای HTML5 را بین سیستم عامل‌های تلفن همراه دید.

مرورگرهای شخص ثالث در شرایط دشوار هستند و نیاز به یک استراتژی برای داشتن میزبان سیستم عامل تلفن همراه خود دارند. فایرفاکس و اپرا در چنین شرایطی هستند. با توجه به این واقعیت است که آنها کنترل کمتر در توسعه سیستم عامل تلفن همراه دارند، آنها قادر نخواهند بود به راحتی برنده شوند در صورتی که مرورگر ساخته شده به اندازه کافی قادر به پشتیبانی HTML5 نباشد. فایرفاکس به دنبال Boot To Gecko برای میزبانی سیستم عامل تلفن همراه خود است. یک ویدئو در یوتیوب [۱۰] سیستم عامل اپرا را در Asus EeePC نشان می‌دهد.

فروشنده‌گان سیستم عامل تلفن همراه، پشتیبانی از HTML5 را مهم می‌دانند و ملاک اصلی معرفی می‌کنند. فروشندگان مرورگر نیز به دنبال امکان قرار دادن آن در سیستم عامل تلفن همراه به طور پیش فرض هستند.

نرم افزار وب

برنامه کاربردی وب، توسعه‌ی برنامه‌های کاربردی سمت سرویس‌گیرنده را با فن‌آوری‌های وب تعریف می‌کند. که ویژگی‌های غنی با ارائه رابط‌های برنامه کاربردی برای توسعه‌ی سمت سرویس‌گیرنده ارائه می‌کند. برنامه‌ی وب می‌تواند حتی در دستگاه‌های آفلاین نصب و اجرا شود. برنامه وب می‌تواند به دستگاه‌ها و منابع محلی به عنوان یک نرم‌افزار بومی دسترسی پیدا کند و می‌تواند در یک فروشگاه نرم‌افزار فروخته شود، که از ارائه‌ی خدمات ابر و صدور صورت‌حساب سودآورتر است.

برنامه وب از یک URL از یک مرورگر قابل دسترسی است. قابلیت‌های مربوط که در چند گروه کاری در W3C تعریف شده است. برنامه‌های کاربردی گروه کاری وب، یک محل مرکزی مربوط به آن آثار است. [۱۱]

برای فعال کردن برنامه‌های کاربردی وب، سیستم‌عامل تلفن همراه نیاز به ارائه یک پلتفرم تحت وب دارد، که شامل زمان اجرا وب، چارچوب وب، و ابزار توسعه است:

- زمان اجرا وب، قابلیت‌های اصلی برای اجرای یک برنامه تحت وب را فراهم می‌کند. این مورد از مرورگر بدست آمده اما با سیستم‌عامل در زمان اجرا یکپارچه است. زمان اجرا وب موتور HTML5 را فراهم می‌کند و APIها برای دسترسی به دستگاه‌های محلی است. این سرویس همچنین قابلیت ادغام با زمان اجرا سیستم‌عامل را دارد، که شامل مدیریت چرخه عمر نرم‌افزار (نصب/به‌روزرسانی/حذف و راه‌اندازی)، یکپارچه‌سازی سیستم‌عامل (یکپارچه‌سازی دسکتاپ، سیاست امنیت، دسترسی به خدمات (OS) است.

- چارچوب وب، کتابخانه غنی برای توسعه نرم‌افزار وب فراهم می‌کند. به عنوان نمونه: jQueryMobile و Sencha. چنین چارچوبی به‌طور گسترده در iOS و اندروید استفاده شده است.

- ابزارهای توسعه باید انعطاف‌پذیر باشند. ابزارهای توسعه بسیار متنوع هستند و می‌توانند با توجه تنظیمات توسعه نرم‌افزار وب انتخاب شوند. ابزارهای توسعه می‌تواند یک مجموعه SDK بسیار متمرکز مانند SDK Tizen ۱.۰، که به‌تازگی منتشر شده، [۱۲] مرورگر مبتنی بر appMobi XDK [۱۳] و یا فقط مجموعه‌ای از ابزار مانند RIB و شبیه‌ساز وب باشد [۱۴] [۱۵].

به عنوان یک روند، سیستم عامل تلفن همراه ملزم به ارائه ی یک وب با عملکرد زمان اجرا بالا و یک چارچوب غنی وب و ابزار توسعه ی قابل انعطاف است.

امنیت همیشه یک موضوع مهم برای محاسبات ابری است. برای ادغام HTML5 در سیستم عامل های تلفن همراه، ویژگی های زیر باید وجود داشته باشد:

- پشتیبانی در زمان اجرا وب. برنامه های کاربردی وب باید در فرآیندهای جدا اجرا شوند و توسط زمان اجرا وب اداره شوند. گودال در مرورگر یا زمان اجرا وب در مدرن ترین سیستم عامل های تلفن همراه پشتیبانی شده است.
- حفاظت از کدهای جاوا اسکریپت. جاوا اسکریپت یک زبان اسکریپت نویسی است، بنابراین هنوز هم بهترین راه برای محافظت از کد، اجرای کد در سمت سرور است.

قابلیت cross-Platform

HTML5 به خوبی برای قابلیت cross-Platform آن شناخته شده است. اما در واقعیت، سیستم عامل های موبایل های مختلف پشتیبانی متفاوتی از HTML5 ارائه می کنند و استاندارد HTML5 همچنان ادامه دارد. PhoneGap برای رسیدگی به پشتیبانی HTML5 cross-Platform با ارائه API های خود دستگاه توسعه داده شده است. iOS، اپل، اندروید و ویندوز فون همگی از PhoneGap پشتیبانی می کنند.

روند این است که یک استاندارد HTML5 داشته باشیم اما آسان نیست. فروشندگان سیستم عامل تلفن همراه ایده های خود را در مسیر خود قبل از استانداردها پیاده سازی می کنند. اپل، گوگل، مایکروسافت همه از شرکت کنندگان فعال در تعریف استاندارد W3C هستند. برای دیگر فروشندگان سیستم عامل تلفن همراه، W3C و یا پیوستن به تعریف، روند زنده ماندن سیستم عامل تلفن همراه است.

کارآیی

کارآیی در موارد بسیاری توسط توسعه دهندگان نرم افزار تلفن همراه شکایت شده است هنگامی که آن ها شروع به ساخت برنامه های کاربردی با HTML5 می کنند. بهینه سازی برای دستگاه های تلفن همراه مهم ترین کار برای انجام موفقیت

HTML5 در حوزه‌ی تلفن همراه است. ما در زیر مهم‌ترین مناطق برای انجام کار بهینه‌سازی برای سیستم‌عامل‌های موبایل را شرح می‌دهیم:

- شتاب سخت‌افزاری. گرافیک و فیلم باید توسط سخت‌افزار شتاب گیرند. WebGL بر روی سیستم‌عامل‌های بیشتری فعال است.
- پشتیبانی multithreading. کارگران وب باید به‌عنوان یک ویژگی اساسی در HTML5 پشتیبانی شوند.
- بهینه‌سازی موتور جاوا اسکریپت. JIT (تنها در زمان) در هر دو SFX و V8 فعال است.
- پشتیبانی نرم‌افزارهای بومی و یا ترکیبی. قابلیت استفاده مجدد از کتابخانه‌های بومی موجود روش دیگری برای برنامه‌های کاربردی وب خواهد بود. آندروید NDK یک چنین توانمندی فراهم می‌کند و به‌طور گسترده‌ای در برنامه‌های کاربردی آندروید استفاده می‌شود. اما در برنامه‌های وب، به‌طور گسترده استفاده نمی‌شود. NaCl با کروم گزینه‌ای برای حمایت از آن است.

ادغام ابر

- علاوه بر قابلیت‌های قدرتمند ارائه شده توسط HTML5، ادغام بدون درز بین ابر و مشتری مهم‌تر است. این نه تنها برای برنامه‌های کاربردی وب است، بلکه برای برنامه‌های کاربردی بومی نیز است. دلایل مهم برای یکپارچه‌سازی شامل:
- ذخیره‌سازی یکپارچه ابر بدون درز. مشتری باید از ذخیره‌سازی ابر فقط به‌عنوان استفاده‌ی آن در ذخیره‌سازی بومی استفاده کند. که نیاز به مشتری ذخیره‌سازی ابر که به‌طور محکم با سیستم‌عامل تلفن همراه یکپارچه شده است دارد. اپل iCloud را که عمیقاً به iOS5 پیوسته است ساخته است. گوگل درایو با خدمات گوگل وب یکپارچه شده است.
 - دسترسی رابط‌های برنامه کاربردی ابر. در سمت سرویس‌گیرنده ابر، سیستم‌عامل تلفن همراه باید قادر به ارائه‌ی کتابخانه‌های آسان برای برنامه‌های کاربردی وب و برای دسترسی به API های بومی مشتری ابر، که به‌طور معمول SOAP، API های آرام، و یا پرس‌وجو هستند باشد.

- مدیریت حساب. با ادغام ابر، مشتریان متعدد با یک حساب، ابر را به اشتراک می‌گذارند و هماهنگ‌سازی بین آن‌ها باید مدیریت شود. حساب باید شدیداً در رابط‌های برنامه کاربردی سیستم‌عامل ادغام شده باشد و حساب تلفن همراه باید برای برنامه‌های کاربردی، دسترسی امن به خدمات ارائه شده یابر و منابع محلی را فراهم کند. هماهنگ‌سازی و اطلاع رسانی از ویژگی‌های کلیدی در سیستم‌عامل‌های موبایل برای استفاده از قابلیت یکپارچه‌سازی ابر در سایز بزرگ است.

بحث و خلاصه

در این بخش، ما ابتدا درباره سیستم‌عامل‌های موبایل عمده ای که امروز در بازار موجود است می‌پردازیم و پس از آن مقاله را به‌طور خلاصه بیان می‌کنیم.

iOS اپل

اپل رهبری طراحی سیستم‌عامل تلفن همراه را دارد. تنها چند سال آیفون و آی‌پد در سراسر جهان حاکم بود. هر دوی این محصولات از ویژگی‌های اپل iOS برخوردار بود.

تجربه‌ی کاربر: iOS عملکرد خوب و به‌طور معمول مجموعه‌ای از معیارها برای دیگر سیستم‌عامل‌های تلفن همراه فراهم می‌کند. اپل به‌طور مداوم عملکرد UX را افزایش می‌دهد. آیفون S۴ عملکرد بسیار بهتری نسبت به نسل‌های قبلی خود، به‌خصوص اینترنت و مرورگر داشت. [۱] با ویژگی‌های جدید اضافه شده، نیازمندی کارآیی بیشتری افزوده می‌شود. مطالعه غیر رسمی [۱] نشان می‌دهد که عملکرد UX بعد از ارتقا از iOS x۴. به آی فون X.۵. در GS3 افزایش یافته است.

مدیریت انرژی: به‌نظر می‌رسد بهینه‌سازی انرژی در iOS قادر به سازگاری با افزایش تقاضا برای انرژی برای ویژگی‌های جدید نیست. Arieso، یک شرکت مدیریت شبکه تلفن همراه، تخمین می‌زند که کاربران آیفون S۴ مصرف داده‌ی بیشتری را نسبت به مدل‌های قبلی آیفون به‌دلیل افزایش استفاده از خدمات آنلاین مانند دستیار شخصی مجازی، که قطعاً مصرف انرژی بیشتری دارد هدر می‌دهند [۳].

باز بودن: در iOS به عنوان یک سیستم عامل تلفن همراه شناخته شده است. کار تحقیقاتی [۶] مفهوم باز بودن پلتفرم (PPO) را تعریف می کند، که در آن درجه‌ی باز بودن یک پلتفرم توسط ادراک توسعه دهندگان آن تصمیم گرفته می شود.

آمادگی ابر: iOS ۵,۰ با پشتیبانی از HTML5 یک مشتری خوب ابر است و از iCloud به طور پیش فرض به عنوان ذخیره سازی یکپارچه استفاده شده است.

گوگل آندروید

در حال حاضر آندروید یک سیستم عامل محبوب برای دستگاه های تلفن همراه است. هدف از پروژه ی آندروید منبع باز ایجاد یک محصول موفق در دنیای واقعی است که تجربه های تلفن همراه برای کاربران نهایی را بهبود می بخشد. [۱۶] تجربه ی کاربر: تیم تجربه ی کاربری آندروید مجموعه ای از اصول طراحی [۱۷] با سه هدف را تعریف می کند: "Enchant me" "Simplify my Life" و "Make me amazing" [۱۸]. دستگاه های آندروید و iOS نتایج مشابهی را در مجموعه ای از آزمون های تست عمر باتری به دست آورد:

مدیریت انرژی: آندروید به شدت دستگاه را برای صرفه جویی در انرژی به حالت تعلیق در می آورد هر زمان که بلوک چیزی را با نگه داشتن wakelock به تعلیق در می آورد [۲۰]. با این حال، آندروید اجازه می دهد تا برنامه های شخص ثالث در پس زمینه اجرا شوند، که ممکن است wakelock را بدون هیچ دلیل خوبی نگه دارد.

امنیت: هر نرم افزار قابل اجرا در یک محیط برای اجرای امنیت در آندروید است و این عمل با اختصاص یک ID منحصر به فرد به هر کاربر نرم افزار و اجرای نرم افزار به عنوان کاربر در یک فرایند جداگانه انجام می شود. [۲۱]

باز بودن: گوگل، کد آندروید را به عنوان منبع باز تحت آپاچی تشخیص می دهد و پروژه منبع باز آندروید جایی است که توسعه ی آندروید و تعمیر و نگهداری در آن اتفاق می افتد. با این حال، گوگل به طور معمول با انتخاب شرکای تولید کننده برای تولید یک دستگاه شاخص برای هر نسخه ی جدید از آندروید همکاری می کند و تنها باعث می شود که پس از انتشار دستگاه کد جدید در دسترس عموم باشد. تکه تکه شدن تبدیل به یک نگرانی بزرگ در اکوسیستم

آندروید شده است. آندروید، برنامه سازگاری آندروید و پیشنهادات تست سازگاری برای تضمین توسعه برنامه‌های کاربردی برای اجرای آندروید بر روی هر دستگاه آندروید را حفظ می‌کند.

مایکروسافت ویندوز فون

مایکروسافت سیستم‌عامل تلفن همراه شدن به نام ویندوز فون را منتشر کرده است. بر اساس تغییر طراحی خود بین ویندوز موبایل ۶،۵ و ویندوز فن ۷، برخی از ویژگی‌های این سیستم‌عامل جدیدتر در معرض دید است.

تجربه‌ی کاربری: با جایگزین تعامل مبتنی بر صفحه نمایش لمسی با کاربران قبلی ورودی قلم، مایکروسافت تصمیم به شکستن سازگاری برنامه بین ویندوز فون و ویندوز موبایل گرفت. [۲۳] مشابه طراحی AppWidget آندروید، ویندوز تلفن مفهوم کاشی‌های زند برای صفحه اصلی صفحه نمایش را اختراع کرد. [۲۸]

مدیریت انرژی: مشابه طراحی امنیت آن، طراحی ویندوز فون برای عمر باتری تا حد زیادی می‌تواند از تجربه ویندوز CE و ویندوز تلفن همراه خود بهره‌مند گردد. یکی از موارد توجه خاص این است که ویندوز فون سیاه و سفید را به عنوان موضوع رنگ اصلی پیش فرض انتخاب می‌کند، چون پیکسل سیاه و سفید هیچ نوری منتشر نمی‌کند، از این رو برای صفحه‌نمایش OLED صرفه‌جویی می‌کند. [۳۲]

امنیت: طراحی ویندوز تلفن از ویندوز اصلی طراحی شرکت-گرا تلفن همراه به یک کاربر نهایی گرا منتقل شده است. تجربه‌ی امنیت انباشته شده برای این محصول شرکت باید هنوز هم مفید باشد. [۳۱]

باز بودن: قبل از این که ویندوز تلفن در بازار موجود باشد، مایکروسافت در دسترس بود و SDK را منتشر کرد که قادر می‌سازد توسعه‌دهندگان برای سیستم‌عامل جدید برنامه‌ریزی کنند. [۲۴] خدمات ویندوز تلفن به ۳۵ کشور/مناطق مختلف ارائه شده است. [۲۵] زبان‌های برنامه‌نویسی در حال حاضر C# و ویژوال اساسی هستند. این یک شگفتی برای توسعه‌دهندگان ویندوز نیست، بنابراین زبان منحنی یادگیری انتظار می‌رود ساده باشد.

آمادگی ابر: ویندوز فون در حال نزدیک شدن سریع به آمادگی ابر می‌باشد. ویندوز فون ۸ اینترنت اکسپلورر ۱۰ را ادغام کرد که ادعا می‌کرد به طور کامل از HTML5 و از بارگذاری موازی صفحه در چند زبانه پشتیبانی می‌کند. [۲۹] علاوه بر این، اسکایپ عمیقاً به سیستم‌عامل یکپارچه تبدیل شده است. [۲۶] یک مفهوم جدید در ویندوز فون هاب‌ها

هستند، که ویژگی‌های مختلف خدمات مشابه به یک مرکز را جمع می‌کنند. این است که تا حد زیادی تجربه کاربری گوشی با خدمات ابر بهبود یافته است [۲۷]. علاوه بر این، طراحی چارچوب نرم‌افزار ویندوزفن شامل دو بخش است: صفحه نمایش و ابر. بخش ابر به خصوص برای "توسعه‌دهنده‌ی پورتال خدمات" و "خدمات ابر" طراحی شده است. در این مقاله، ما جنبه‌های مهم طراحی سیستم‌عامل تلفن همراه بر اساس مدل تجزیه و تحلیل پیشنهادی ما، از جمله تجربه کاربر، عمر باتری، آمادگی ابر، امنیت، و باز بودن مورد بررسی قرار دادیم. این باید زمینه‌ای برای تمرکز نسل بعدی طراحی سیستم‌عامل تلفن همراه باشد.

سیستم‌عامل تلفن همراه آینده نیز به طراحی سخت‌افزاری در دسترس بستگی دارد. ما باور داریم که یک سیستم تلفن همراه موفق در نتیجه شرکت در طراحی نرم‌افزار و سخت‌افزار، همراه با پیشرفت اینترنت می‌باشد.

References

(The paper references the web sites accessed on May 1, 2012.)

- [1] <http://www.youtube.com/watch?v=ng33wXDkyRM>,
<http://www.iphonedownloadbolg.com>.
- [2] <http://www.anandtech.com/show/4951/iphone-4s-preliminary-benchmarks-800mhz-a5-slightly-slower-gpu-than-ipad-2>
- [3] <http://www.arieso.com/news-article.html?id=89>
- [4] http://www.apple.com/iphone/business/docs/iOS_Security_Mar12.pdf
- [5] <http://www.techrepublic.com/blog/security/comparing-android-and-ios-security-how-they-rate/5774>
- [6] <http://www.user.tu-berlin.de/komm/CD/paper/090322.pdf>
- [7] <http://www.apple.com/icloud/>
- [8] <http://www.networkworld.com/community/blog/apples-ios-5-and-cloud>
- [9] <http://html5test.com/results/mobile.html>
- [10] <http://www.youtube.com/watch?v=mWSPNDD0tek>
- [11] <http://www.w3.org/2012/webapps/charter/Overview.html>
- [12] <https://developer.tizen.org/sdk>
- [13] <https://chrome.google.com/webstore/detail/onmkoldigcfmebcinpmineoadckallb>
- [14] <https://01.org/projects/rapid-interface-builder-rib>

- [15] <https://01.org/projects/web-simulator>
- [16] <https://www.mylookout.com/mobile-threat-report>
- [17] <http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2011-mobile-threats-report.pdf>
- [18] <http://webupon.com/security/information-systems-attacker-motivations/>
- [19] <http://source.android.com/tech/security/index.html>
- [20] <http://www.w3.org/2011/webappsec/>
- [21] <http://www.trustedcomputinggroup.org>
- [22] <http://www.globalplatform.org/specifications.asp>
- [23] <http://channel9.msdn.com/Events/TechEd/NorthAmerica/2010/WPH201>
- [24] <http://www.engadget.com/2010/09/16/microsoft-demos-twitter-and-netflix-apps-for-windows-phone-7-ri/>
- [25] http://windowsteamblog.com/windows_phone/b/windowsphone/archive/2011/07/06/windows-phone-around-the-world-language-support-in-mango.aspx
- [26] <http://wmpoweruser.com/microsoft-reps-claiming-windows-phone-8-definitely-coming-to-second-gen-handsets-probably-to-first-gen/>
- [27] <http://www.engadget.com/2010/03/18/windows-phone-7-series-the-complete-guide/>
- [28] <http://www.engadget.com/2010/02/15/windows-phone-7-is-official-and-microsoft-is-playing-to/>
- [29] <http://pocketnow.com/windows-phone/exclusive-windows-phone-7-web-browser-comparison>
- [30] <http://arstechnica.com/microsoft/news/2011/04/windows-phone-7-mango-one-heck-of-an-upgrade.ars>
- [31] <http://arstechnica.com/microsoft/news/2010/03/windows-phone-7-series-in-the-enterprise-not-all-good-news.ars>
- [32] <http://www.neowin.net/news/interview-windows-phone-7-battery-life-copy-paste-multitasking-and-more>
- [33] <http://software.intel.com/en-us/articles/aws-android-workload-suite-for-user-interaction-measurement/>
- [34] <http://software.intel.com/en-us/articles/quantify-and-optimize-the-user-interactions-with-android-devices/>