



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پروژه کارشناسی
گرایش فناوری اطلاعات

مروری بر پروژه کارشناسی - پیاده سازی ابزار مبتنی وب
به منظور سجنش استفاده پذیری رابط کاربری سامانه های
مبتنی بر وب به روش جمع سپاری

نگارش
امیر حقیقتی ملکی

استاد راهنما
استاد احمد عبدالله زاده بارفروش

تیر ۱۳۹۷

چکیده

با تقریب خوبی می‌توان گفت تمامی مدل‌های کیفی نرم‌افزار، استفاده‌پذیری را جزو مشخصه‌های اصلی کیفیت یک نرم‌افزار مطرح می‌کنند. وجه مشترک تعاریف متعددی که برای استفاده‌پذیری مطرح می‌شود، در سه بعد کاربر، انجام یک فعالیت مشخص و تعامل با یک واسط برای انجام آن فعالیت، قابل بیان است. به عنوان یک مهندس نرم‌افزار، افزایش کیفیت در محصولات و کاهش هزینه‌های ناشی از خرابی‌ها و یا درخواست‌های تغییر، چالشی تامل برانگیز است. وب‌اپلیکیشن‌ها به عنوان نوعی محصول نرم‌افزاری که در آن‌ها زیبایی، واسط کاربری و نحوه تعامل کاربران مهم است، به دلیل استفاده گسترده‌شان، می‌توانند تاثیر شگرفی در موفقیت یک پروژه صنعتی، کسب‌وکارهای نوپا و یا تسهیل زندگی روزمره با استفاده از نرم‌افزارها داشته باشند. از جمله نقاط ضعف بیشتر وب‌اپلیکیشن‌ها، طراحی نه‌چندان کاربرپسندانه واسط کاربری آن‌هاست که موجب شده تا در بسیاری از موارد، کاربران، علاقه‌مندی استفاده از محصول مبتنی وب یک سازمان را در عین سرمایه‌گذاری‌های زیاد آن سازمان برای جذب کاربر، از دست بدهند و در نتیجه متضرر شوند. گرچه، به صورت ایده‌آل، تمامی تصمیم‌گیری‌های مدیریتی و کلان (از قبیل اتخاذ مدل‌های فرایندی مناسب برای تولید نرم‌افزار با هزینه کم) با نهایت دقت و تجربه انجام می‌شوند، ولی در بسیاری از موارد همچون پروژه تقویم شرکت گوگل، مواردی ملاحظه می‌شود که واسط کاربری ناکارآمد، به ناچار، هزینه‌های گاه‌آزادی به تیم مهندسی نرم‌افزار تحمیل کرده است. با مروری بر منابع مختلف، ارزیابی و تست روی نمونه‌های اولیه رابط کاربری وب‌اپلیکیشن‌ها به منظور رفع نواقص آن‌ها، امری واضح به نظر می‌رسد. اما پاسخ دادن به این سوال که «چه واسط کاربری‌ای خوب است؟» همیشه آسان نبوده و با تغییر فناوری و گذشت زمان شاهد تغییر سریع در نیازمندی‌ها هستیم که شاید چک‌لیست‌ها و توصیه‌ها نیز پاسخگوی دقیقی برای آن‌ها نباشند. بنابراین می‌بایست در طراحی واسط کاربری، به یک روش کمی و قابل استناد، نیازمندی‌ها را با استفاده از نمونه‌های اولیه بسنجیم (که به دلیل هنری انگاشتن اکثر کارها، این امر نادیده گرفته می‌شود). اما سنجش دقیق، نیازمند جمع‌آوری داده از ارزیابی و تست واسط کاربری توسط کاربران نهایی است تا بتوان تحلیل دقیق انجام داد و مشکلات طراحی واسط را به درستی تشخیص داد. یکی از روش‌های جمع‌آوری داده، استفاده از جمع‌سپاری است. باید توجه داشت که استفاده از جمع‌سپاری چالش‌هایی را فرارویمان خواهد گذاشت که از جمله آن‌ها می‌توان به عدم وجود صحت در داده‌ها اشاره کرد. در این پروژه وب‌اپلیکیشنی به منظور ارائه داشبورد مدیریتی برای صاحبان طراحی و افراد متمایل به انجام تست‌های مختلف با معیارهای متفاوت و دلخواه، پیاده خواهد شد. همچنین دادگان و پاسخ‌ها و تحلیل‌های تست اپلیکیشن در مواجهه کاربران واقعی با آن‌ها، به اطلاع کاربر خواهد رسید؛ علاوه بر موارد فوق، قسمت اصلی این پروژه در پاسخ به چالش صحت داده در روش جمع‌سپاری، ابتدا رفتار کاربران پاسخ‌دهنده (کارگران) توسط ماتریسی مدل می‌شود که برای مدل‌سازی و به دست آوردن مقادیر مدل‌ها، از روش تزریق سوالات طلایی استفاده خواهد شد. سپس در صورت پایین بودن کیفیت کار کارگران از حد مشخصی که در هنگام مدل‌سازی مشخص می‌شود، نتیجه کار آن‌ها به عنوان داده نامربوط شناخته شده و حذف می‌گردد. امکان تعریف تست‌های دلخواه و محدود نبودن به تست‌های از پیش تعریف شده تفاوت عمده ابزار استفاده‌پذیر با سایر ابزارهای مشابه است؛ از جمله ابزارهای مطرح موفق در این حوزه، می‌توان به Optimizely UsabilityHub و CrazyEgg اشاره کرد که همانطور

که ذکر شد، در طی این پروژه، سعی بر برطرف سازی برخی از نواقص آنهاست.

واژه های کلیدی:

کلیدواژه اول، ...، کلیدواژه پنجم (نوشتن سه تا پنج واژه کلیدی ضروری است)

فهرست مطالب

صفحه

عنوان

۱	۱ مقدمه
۲	۱-۱ کیفیت در نرم افزار
۳	۱-۱-۱ تضمین و کنترل کیفیت
۴	۲-۱-۱ کیفیت در سامانه های نرم افزاری مبتنی بر وب
۴	۲-۱ مشخصه های کیفی نرم افزار
۴	۱-۲-۱ کارآیی
۶	۳-۱ چرخه طراحی سامانه های مبتنی بر وب
۷	۱-۳-۱ تلفیق نگاه مهندسی و هنری
۷	۴-۱ جمع سپاری
۸	۱-۴-۱ جمع سپاری برای جمع آوری داده
۹	۲ شیوه پژوهش
۱۲	۳ مروری بر کارهای گذشته، روش ها و ابزارهای موجود
۱۳	۱-۳ مدل های کیفیتی
۱۳	۱-۱-۳ مدل های سلسله مراتبی
۱۵	۲-۱-۳ مدل های مبتنی بر متامدل
۱۵	۳-۱-۳ مدل های آماری و ضمنی
۱۵	۲-۳ تمرکز بر استفاده پذیری روی مدل های کیفیتی
۱۶	۳-۳ مطالعه استفاده پذیری و ارزیابی تجربه کاربری
۱۷	۱-۳-۳ ارزیابی خرد
۱۷	۲-۳-۳ ارزیابی کلان
۱۸	۳-۳-۳ اهداف کاربری
۲۰	۴-۳ خصیصه های استفاده پذیری و اندازه گیری آن
۲۰	۵-۳ ابزارهای سنجش استفاده پذیری
۲۱	منابع و مراجع
۲۴	پیوست
۲۵	واژه نامه ی فارسی به انگلیسی

فهرست اشکال

شکل	صفحه
۱-۱ نرم‌افزار به عنوان پلی بین دامنه مسئله و دامنه راه‌حل [۴۴]	۲
۲-۱ تابلویی که به خاطر نصب در جای نامناسب و استفاده‌پذیر نبودن آن، منجر به تصادف و مرگ ۱۰ نفر شد [۷]	۵
۳-۱ هرم طراحی وب‌اپلیکیشن‌ها [۳۶]	۶
۴-۱ مدل تشدید خرابی در نرم‌افزار [۳۶]	۶
۱-۲ متدلوژی پژوهش آزمایشگاه سیستم‌های هوشمند و گام‌های آن	۱۰
۱-۳ خط زمانی ارائه برخی از مدل‌های کیفی [۳۵]	۱۳
۲-۳ بررسی انواع مدل/رویه‌های کیفیتی استفاده شده در سازمان‌ها [۴۵]	۱۴
۳-۳ ساختار مدل کیفی مک‌کال به عنوان یک مدل کیفی سلسله‌مراتبی [۱۲]	۱۴
۴-۳ مثالی از دو نوع ارزیابی مختلف خرد و کلان [۳]	۱۷
۵-۳ مدل کانوکه نشان‌دهنده ارتباط رضایت کاربر و ویژگی‌های محصول است [۸]	۱۹

فهرست جداول

صفحه

جدول

۱-۲	کلیدواژه‌های به دست آمده از دانش زمینه	۱۱
۱-۳	مقایسه تطبیقی مدل‌های کیفیتی ارائه شده با تمرکز بر استفاده‌پذیری	۱۶

فهرست نمادها

نماد	مفهوم
\mathbb{R}^n	فضای اقلیدسی با بعد n
\mathbb{S}^n	کره n یکه بعدی
M^m	خمینه m -بعدی M
$\mathfrak{X}(M)$	جبر میدان‌های برداری هموار روی M
$\mathfrak{X}^1(M)$	مجموعه میدان‌های برداری هموار n یکه روی (M, g)
$\Omega^p(M)$	مجموعه p -فرمی‌های روی خمینه M
Q	اپراتور ریچی
\mathcal{R}	تانسور انحنای ریمان
ric	تانسور ریچی
L	مشتق لی
Φ	۲-فرم اساسی خمینه تماسی
∇	التصاق لوی-چویتای
Δ	لاپلاسین ناهموار
∇^*	عملگر خودالحاق صوری القا شده از التصاق لوی-چویتای
g_s	متر ساساکی
∇	التصاق لوی-چویتای وابسته به متر ساساکی
Δ	عملگر لاپلاس-بلترامی روی p -فرم‌ها

فصل اول

مقدمه

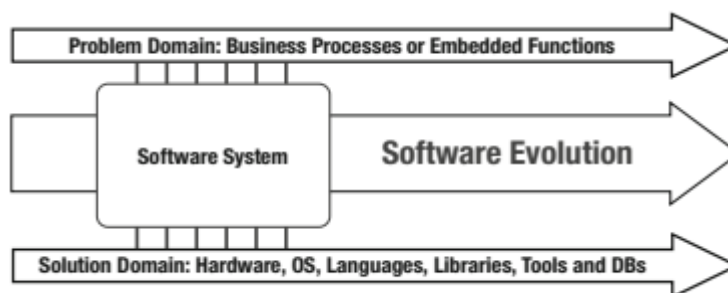
خریداری یا استفاده از یک محصول با این پیش‌زمینه و تفکر که محصول مورد نظر نیاز خاصی را برطرف خواهد کرد، خود به خود انتظار برطرف کردن نیازمندی‌های ذهن مصرف‌کننده را در وی می‌انگیزد [۴۶]. در ابتدا شاید صرفاً رفع نیاز مصرف‌کنندگان، به هر روش ممکن، دغدغه اصلی تولیدکننده باشد اما به مرور و با گذشت زمان که نیازمندی‌ها پخته‌تر می‌شوند و ارتقا می‌یابند، کیفیت نیز در آن‌ها دخیل می‌شود. از طرفی، وجود نام‌ونشان‌های متعدد و متنوع در بسیاری از صنایع نیز، منجر به ایجاد رقابت میان فعالان هر عرصه شده است؛ رقابتی که کیفیت تعیین‌کننده‌ترین عامل برد و باخت در آن است [۳۶]. صنعت نرم‌افزار نیز، به عنوان یکی از صنایع نوین که محصول‌اتش امروزه سهم قابل توجهی از بازار را در مصارف روزمره اداری و شخصی به خود اختصاص داده است، از این قاعده مستثنی نیست. بنابراین در تولید و توسعه یک محصول نرم‌افزاری نیز به منظور موفقیت هرچه بیشتر، می‌بایست به کیفیت، نگاه جدی داشته باشیم.

به طور خاص، در سامانه‌های کاربردی مبتنی بر وب (وب‌اپلیکیشن‌ها) و موبایل که جامعه کاربریشان هر روز بیشتر و بیشتر می‌شود، نیازمندی‌های مختلفی در طول چرخه عمر نرم‌افزار بروز پیدا می‌کنند. از طرفی در دنیای نرم‌افزار، گسترده‌تر شدن دامنه دسترسی به یک محصول نرم‌افزاری، الزاماتی برای آن فراهم می‌آورد که برای مثال، می‌توان گفت محصول نرم‌افزاری می‌بایست توسط یک فرد عادی از جامعه هدف مشتریان، قابل استفاده باشد. قابل استفاده بودن را نه در دانش فنی کاربران سیستم، بلکه در قابل فهم بودن رابط میان سیستم و کاربران تعریف می‌کنیم [۱۳].

البته ناگفته نماند دانش فنی و مهارت استفاده از ابزارهای فناوری‌محور، بخش غیرقابل اغمازی از توانایی استفاده از یک محصول نرم‌افزاری را ممکن می‌سازد؛ ولی امروزه، در مورد محصولات نرم‌افزاری تحت وب که به طور معمول با تعداد کاربران زیادی مواجه هستند، قابل استفاده بودن و استفاده‌پذیری آن‌ها در هنگام کار یک کاربر عادی، یکی از معیارهای مهم کیفیت به شمار می‌رود.

۱-۱ کیفیت در نرم‌افزار

کیفیت یک نرم‌افزار، یک خصیصه ثابت و مشخص کلی نیست. بلکه به انتظارات و نیازمندی‌های ذی‌نفعان بستگی زیادی دارد؛ برای قرار دادن کیفیت در اولویت‌های تولید نرم‌افزار، می‌بایست، در همان ابتدای کار و قبل از شروع هرچیز دیگری، یک تعریف مدون و کاملاً مشخص از کیفیت داشته باشیم.



شکل ۱-۱: نرم‌افزار به عنوان پلی بین دامنه مسئله و دامنه راه‌حل [۴۴]

بسیاری از تحقیقات در سال‌های گذشته، صرف به دست آوردن فرآیندهای نرم‌افزاری با کیفیت شده است؛ البته که فرآیندهای

باکیفیت در نهایت منجر به تولید محصولی با کیفیت می‌شود، ولی برای بروز کیفیت در فرایندها نیز خصیصه‌های کیفی محصول نرم‌افزاری هدف، باید به طور مشخص قید شوند [۴۳]. هرچند که داشتن یک تعریف مدون و مشخص از کیفیت، لازمه کار هر فرآیند مهندسی نرم‌افزار است، نکته حائز اهمیتی که بسیاری از محققین و پژوهشگران در آثار خود از جمله آقایان پرسمن [۳۶]، سامرویل [۴۳] و واگنر [۴۴] به آن اشاره کرده‌اند، بیانگر این موضوع است که داشتن یک توصیف کیفیتی کامل و دقیق از سیستم هدف نیز، به تنهایی، کافی نیست؛ چرا که همین توصیف کیفیتی نیز با گذر زمان، دچار تغییر و تحول خواهد شد و دیگر نیازمندی‌های کیفیتی، معتبر نخواهند بود.

همانطور که در شکل ۱-۱ ملاحظه می‌شود، نرم‌افزار میان دو دامنه مسئله و راه‌حل ارتباط برقرار می‌کند و می‌بایست پلی بین فرایندهای کسب‌وکاری و پلتفرم‌های فناوری (سیستم‌های عامل، سخت‌افزارها و نرم‌افزارهای مختلف) ایجاد کند. اما توجه به این نکته حائز اهمیت است که هم فرایندهای کسب‌وکار و هم پلتفرم‌های فناوری، در طول زمان دچار تغییر می‌شوند؛ به خصوص که سرعت تغییرات در عصر حاضر به شدت زیاد است. سخت‌افزارها منسوخ می‌شوند، سیستم‌های عامل به نسخه‌های جدیدتری ارتقا پیدا می‌کنند، زبان‌های برنامه‌نویسی پیشرفته‌تر می‌شوند، ابزارهای جدیدی تولید می‌شوند و کسب‌وکارها در نتیجه این تغییرات، خود را به روز می‌کنند و فرایندهای کسب‌وکاری نیز می‌بایست بتوانند این تغییرات را پشتیبانی کنند و در نتیجه تغییر می‌کنند [۴۴].

در نتیجه ویژگی‌ها و نیازمندی‌های کیفی نرم‌افزار نیز تغییر پیدا می‌کند و اگر خود نرم‌افزار مطابق این تغییرات به روز رسانی نشود، سیستم کم‌کیفیتی خواهیم داشت.

۱-۱-۱ تضمین و کنترل کیفیت

همانطور که پرسمن در کتابش [۳۶] مطرح می‌کند، رسیدن به یک محصول با کیفیت در مهندسی نرم‌افزار، به صورت ضمنی و خود به خود ممکن نیست؛ بلکه نتیجه بازنگری در چهار بعد کلی در فرآیند مهندسی نرم‌افزار و اعمال مجموعه آن‌ها است:

- روش‌های مهندسی نرم‌افزار
- تکنیک‌های مدیریت پروژه
- فعالیت‌های کنترل کیفیت
- فعالیت‌های تضمین کیفیت

طبق این اظهار نظر، با فرض اعمال شدن روش‌های درست و بهره‌ور مهندسی نرم‌افزار و تکنیک‌های موثر در مدیریت پروژه تولید نرم‌افزار - که با تقریب خوبی هر دو را می‌توان جزو روش‌های مدیریتی و در حوزه تصمیم‌گیری‌های کلان سیستم دانست - بدیهی است که همچنان کنترل کیفیت و تضمین آن، دو بعد فنی و جزئی‌تر رسیدن به نرم‌افزار با کیفیت را تشکیل می‌دهند. بنابراین می‌بایست روش‌های موثر به منظور انجام فرایندهای کنترل کیفیت و تضمین رسیدن به آن، توسط تیم مهندسی نرم‌افزار اتخاذ شود.

اما، مشابه هر فرایند و فعالیت دیگری، رسیدن به کیفیت نیز هزینه‌های خاص خود را دارد. هزینه کیفیت در نرم‌افزار، مطابق اظهار نظر پرسمن، به سه دسته هزینه‌های پیش‌گیری، هزینه‌های ارزیابی و هزینه‌های خرابی تقسیم می‌شود. هرکدام از این هزینه‌ها، در صورت پیش‌بینی و رفع نواقص محتمل/پیش‌آمده در هر مرحله از طراحی و پیاده‌سازی، بدون اینکه وارد مرحله بعدی شویم، می‌تواند با نرخ بسیار زیادی کاهش یابد [۳۶].

۲-۱-۱ کیفیت در سامانه‌های نرم‌افزاری مبتنی بر وب

یکی از علل عدم رضایت کاربران و مشتریان از وب‌اپلیکیشن‌ها - که در نتیجه این نارضایتی، آمار کاربران وب‌اپلیکیشن‌های کسب‌وکارها دستخوش تغییرات نامطلوب شده و حتی هزینه‌های گزافی به تیم مهندسی نرم‌افزار به خاطر اعمال تغییر پس از تحویل، وارد می‌شود- طراحی نه‌چندان کاربرپسندانه واسط کاربری و زیبایی آن‌هاست [۱۱]؛ بدیهی است که استفاده از مدل‌های فرایندی چابک و تکراری می‌تواند در کاهش هزینه‌های طراحی مجدد پس از تحویل و یا اعمال تغییر در رابط‌های موجود، موثر باشد [۳۶]، اما هنوز یک سوال بدون پاسخ خواهد ماند: «چه رابطی برای کاربران وب‌اپلیکیشن (محصول) من مناسب است و طبق نیازمندی‌های فعلی حداکثر کیفیت را تامین خواهد کرد؟» برای پاسخ به این سوال، چک‌لیست‌ها و توصیه‌های فراوانی ارائه شده است [۳۶، ۴۳] که هرکدام به نحوی در افزایش کیفیت رابط‌های کاربری تاثیرگذار بوده‌اند، اما برای تست یک رابط کاربری به صورت کمی، تحلیل و یافتن نقاط ضعف در زیبایی و همچنین ریزبینی در مورد استفاده‌پذیری یک واسط کاربری، به نظر می‌رسد که بررسی بیشتری مورد نیاز است [۱۳].

۲-۱ مشخصه‌های کیفی نرم‌افزار

۱-۲-۱ کارایی

به تعبیر نویسندگان مرجع [۱۳] هر کاربر می‌تواند برای خودش تعریفی از کارایی ارائه نماید. در ادامه بررسی مفصل و مقایسه تطبیقی از مدل‌های کیفی مختلف و کارایی در هرکدام انجام شده است ولی در اینجا به طور مختصر به ارائه و مقایسه سه نوع دیدگاه از تعریف کارایی می‌پردازیم:

۱. سازمان بین‌المللی استانداردها (ایزو ۱۱-۹۲۴۱) کارایی را در سه حوزه تعریف می‌کند: «میزان سودی که استفاده از یک محصول در رسیدن به اهداف مورد نظر کاربران در رابطه با کاربردی مشخص، که همراه با تاثیرگذاری، بهره‌وری و رضایت باشد، کارایی آن محصول نامیده می‌شود.»

۲. جامعه متخصصین کارایی^۱ بیشتر روی فرایند تولید و توسعه محصول تمرکز می‌کنند و با بیان استفاده‌پذیری به عنوان «یک روش برای کاستن هزینه‌ها و تولید ابزارهایی که مختص کاربرانشان باشد»، از ویژگی مرتبط بودن همواره استفاده‌پذیری با کاربران، استفاده می‌کند.

۳. استیو کورگ در کتاب خود، کاری نکن که من به فکر کردن بیفتم تعریف عامیانه‌تری را ارائه می‌دهد؛ وی معتقد است که استفاده‌پذیری به معنی اطمینان حاصل کردن از کار کردن خوب محصول نهایی است. با این توضیح که یک فرد با دانش، توانمندی و تجربه کم نیز بایستی بتواند از محصول به راحتی استفاده کند و نیازهای خود را برطرف سازد.

با بررسی‌های مرجع [۱۳] تمامی تعاریف مطرح برای استفاده‌پذیری، شامل سه زمینه کلیدی و مهم هستند:

۱. کاربری وجود دارد.

۲. این کاربر مشغول انجام کاری است.

۳. کاربر در حین کار خود، با یک سیستم یا محصول نرم‌افزاری در تعامل است.

^۱ Usability Professionals Association

یکی از عوامل بسیار تاثیرگذار در استفاده‌پذیری هر محصولی، رابط کاربری آن است؛ همچنین، بی‌کیفیتی و استفاده‌پذیری نبودن یک محصولی که کاربر با آن در ارتباط است و کاربرد حساسی دارد، مانند تابلویی که در شکل ۲-۱ قابل مشاهده است و در مرجع [۱۳] از این حادثه یاد شده است، می‌تواند به مرگ و زندگی افراد ختم شود.



شکل ۲-۱: تابلویی که به خاطر نصب در جای نامناسب و استفاده‌پذیر نبودن آن، منجر به تصادف و مرگ ۱۰ نفر شد [۷]

اینکه کاربر در طول دوره کاری‌اش با سیستم به طور دقیق به چه موارد منفی یا مثبت یا حتی خنثی برخوردده، نقش مهمی در تجربه کاربری وی خواهد داشت.

استفاده‌پذیری به طور کلی به توانایی کاربر در انجام موفق یک کار مشخص دلالت دارد، در حالی که تجربه کاربری به جنبه وسیع‌تری پرداخته و شامل احساسات، عواطف و ادراکات کاربر در حین کار با سیستم می‌شود [۱۳]. در بخش‌های بعدی و با بررسی مدل‌های کیفی مختلف که به منظور سنجش کمی کیفیت نرم‌افزار ارائه شده‌اند، خواهیم دید که استفاده‌پذیری نرم‌افزار، به عنوان یکی از مشخصه‌های اصلی در اغلب این مدل‌ها و به صورت صریح بیان شده است. با بررسی پژوهش‌ها و کارهای گذشته و همچنین نکاتی از مرجع [۳۶]، می‌توان گفت از سال ۱۹۷۰ تا به اکنون، تقریباً در هر مدل کیفی ارائه شده برای نرم‌افزار و به طور خاص برای سامانه‌های کاربری تحت وب، استفاده‌پذیری به صورت صریح به عنوان یک مشخصه اصلی بیان شده است؛ بنابراین می‌توان ادعا کرد استفاده‌پذیری یک نرم‌افزار، از جمله ویژگی‌های مهم کیفی در دستیابی و کنترل کیفیت نرم‌افزار است.

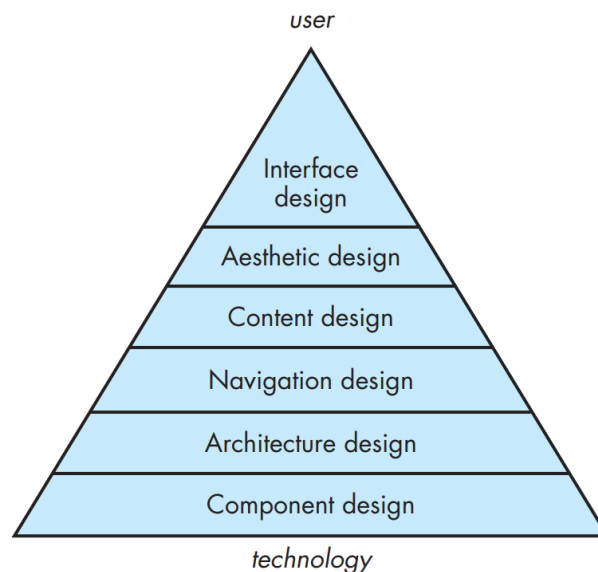
کارآیی و لایه‌های طراحی سامانه‌های مبتنی بر وب

استفاده‌پذیری در وب‌اپلیکیشن‌ها - که امروزه نقش مهمی در ارائه محتوا و سرویس به کاربران دارند - به عنوان یکی از ابعاد و مشخصه‌های اصلی و مهم در کیفیت مطرح است [۳۶]. رسیدن به کیفیت بالا نیازمند صرف هزینه (تلاش و زمان) است؛ صرفاً با در نظر گرفتن بعد کارآیی، پرواضح است که هرچه مشکلات و نواقص رابط‌های کاربری زودتر پیدا شده و مرتفع گردند،

با پرداخت هزینه (تلاش و زمان) کمتر به کیفیت بیشتری رسیده‌ایم؛ لایه‌های طراحی سامانه‌های مبتنی بر وب، هر کدام تمرکز جدایی دارند که در ادامه به طور مختصر قید شده‌اند. هر کدام از این لایه‌ها، به نوعی کارآیی نهایی محصول را تامین می‌کنند و در تضمین کیفیت باید به هر لایه به طور جداگانه توجه ویژه‌ای را معطوف نمود.

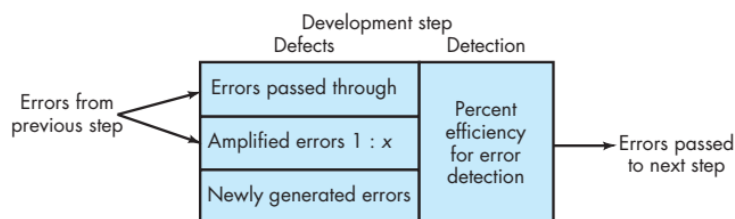
۳-۱ چرخه طراحی سامانه‌های مبتنی بر وب

از جمله مراحل هرم طراحی وب‌اپلیکیشن [۳۶]، طراحی واسط کاربری است. همانطور که در شکل ۳-۱ مشاهده می‌شود، طراحی زیبایی، محتوا، پیمایش، معماری و همچنین مولفه نیز در فرایند طراحی می‌بایست انجام شوند که هر کدام نکات خاص خود را دارند و می‌توانند در استفاده‌پذیری وب‌اپلیکیشن تاثیرگذار باشند.



شکل ۳-۱: هرم طراحی وب‌اپلیکیشن‌ها [۳۶]

همچنین شایان ذکر است که لایه‌های مختلف این هرم، هر کدام توجه جداگانه‌ای دارند و می‌بایست در تامین کیفیت، به در هر لایه سیاست‌های به خصوصی اتخاذ شود. قبل از تولید کد وب‌اپلیکیشن، واسط کاربری، به صورت یک نمونه اولیه و در قالب طرح‌های ابتدایی، ماکت‌های مفهومی و یا چارچوب‌های کلی توصیف و طراحی می‌شوند. پس از رسیدن به توافق با مشتری (در صورت نیاز) و یا اعمال تغییرات متعدد تا رسیدن به توافق، این طراحی به کد قابل اجرا و پیاده‌سازی روی وب‌اپلیکیشن تبدیل می‌شود و نهایتاً به تولید واسط کاربری آن می‌انجامد [۴۳].



شکل ۴-۱: مدل تشدید خرابی در نرم‌افزار [۳۶]

مطابق شکل ۱-۴ خرابی‌ها و خطاها در صورتی که برطرف نشوند و وارد مرحله بعد شوند، می‌توانند در تولید وب‌اپلیکیشن مشکلات جدی‌ای ایجاد کنند؛ چرا که این خطاها تشدید می‌شوند و دچار خرابی کار سایر لایه‌ها نیز می‌گردند و در نهایت منجر به افت کیفیت محصولات نهایی می‌گردند. از جمله خطاها و خرابی‌های مطرح در حوزه طراحی رابط کاربری، ناکارآمد بودن ایده‌های اولیه و چکش‌نخورده است. مطابق آنچه در قسمت تضمین و کنترل کیفیت گفته شد، در صورت ارزیابی، تحلیل و رفع ایرادات مربوط به استفاده‌پذیری رابط کاربری، در همان مراحل ابتدایی و پس از تولید نمونه اولیه، می‌توان هزینه‌های بعدی را به طور قابل ملاحظه‌ای کمتر کرد.

مانند هر روش کیفی دیگری در تضمین کیفیت نرم‌افزار، به منظور دستیابی به استفاده‌پذیری قابل قبول (مطابق نیازهای مشتری) در واسط کاربری وب‌اپلیکیشن‌ها (همچون هر مشخصه اصلی دیگری) می‌بایست فاکتورها، معیارها و مولفه‌های مختلفی به منظور خرد و قابل اندازه‌گیری کردن این مفهوم کلان مطرح شود به طوری که بتوان در قالب مقادیر کمی، نیازمندی‌ها را با داده‌های به دست آمده از ارزیابی رابط کاربری وب‌اپلیکیشن مقایسه و تحلیل کرد. اما در بسیاری از موارد، همانطور که [۱۱، ۳۵، ۱۳] ذکر می‌کنند، حقیقت محض و یا هیوریستیک تضمین‌کننده‌ای برای رسیدن به یک رابط کاربری «خوب» وجود ندارد و طراحی‌های استفاده‌پذیری و موثر موفقیت خود را اغلب یا به روش‌های تجربی، که الزاماً با روش‌های علمی به اثبات نرسیده‌اند، و یا به ذوق هنری طراح مدیون‌اند.

۱-۳-۱ تلفیق نگاه مهندسی و هنری

دور از ذهن نیست که بگوییم یکی از فاکتورهای محبوبیت یک اثر هنری، جذابیت اثر در دید مخاطبان‌اش است. بنابراین پرواضح است که در مورد رابط‌های کاربری، که در ابتدا به صورت یک طرح مفهومی بوده و اثر یک طراح -که الزاماً شاید سرشته‌ای از مهندسی نداشته باشد- هستند، نظر کاربران و استفاده‌کنندگان آن طرح مفهومی و نحوه تعاملشان با طرح مفهومی، یکی از مشخصه‌های تعیین‌کننده برای موفقیت رابط کاربری هدف و تضمین کیفیت آن است. در نتیجه به نظر می‌رسد اندازه‌گیری نظرات کاربران و داشتن یک دید مهندسی در نقطه نظرات کاربران و واکنش‌های آن‌ها هنگام کار با یک طرح مفهومی که به منظور استفاده در یک رابط کاربری ساخته شده است، امری لازم و مثبت خواهد بود و درکل منجر به افزایش اطلاعات طراح و تیم توسعه از نیازهای کاربران خواهد شد.

۴-۱ جمع‌سپاری

تا سال ۲۰۱۲، با بررسی‌های مرجع [۲۳]، حدود ۴۰ تعریف مختلف در مقالات و پژوهش‌های علمی، حتی گاهی تعاریف متناقض با هم، برای جمع‌سپاری ارائه شده است. نویسندگان آن اثر، با درنظر گرفتن ابعاد مطرح در تعاریف مختلف، در نهایت تعریف نسبتاً مفصلی از این مفهوم ارائه می‌دهند که ترجمه آزاد آن در ادامه ذکر شده است:

جمع‌سپاری که ترجمه شده عبارت Crowdsourcing است، نوعی فعالیت برخط^۱ مشارکتی است که طی آن یک فرد، یا یک سازمان با ابزارهای کافی به گروهی از افراد با سطح دانش متغیر و گونه‌های متفاوت و با تعداد نامعلومی به انجام فعالیت‌هایی می‌پردازند. در این کار دو سر برد، کارگران انجام دهنده کار^۲ به دلیل داوطلبانه بودن مشارکتشان، از انجام کار خود احساس

^۱Online

^۲Crowd Workers

رضایت می‌کنند؛ چه به خاطر پولی که در ازای انجام کار دریافت می‌کنند و چه به خاطر توسعه مهارت‌های شخصی و یا سایر انگیزه‌ها؛ افراد جمع‌سپارنده هم از مشارکت افراد در حل مسائل پیچیده کمک بسته و سودآوری خود را خواهند داشت. یکی از انگیزه‌های استفاده از جمع‌سپاری، جمع‌آوری داده^۳ است. در این استفاده، از کارگران جمع‌سپاری شده بهره گرفته می‌شود تا بتوان به مجموعه عظیمی از دیتاست‌ها و یا داده‌های جدید دست پیدا کرد.

۱-۴-۱ جمع‌سپاری برای جمع‌آوری داده

انگیزه اصلی استفاده از جمع‌سپاری در این پروژه، جمع‌آوری داده است. ابزار هدف، قادر خواهد بود تا با استفاده از جمع‌سپاری، بتواند نتایج تست‌های تعریف‌شده توسط مشتریان را از کارگران جمع‌آوری کرده و روی آن‌ها تحلیل و پردازش انجام دهد. عدم وجود یک حقیقت محض قابل اتکا^۱ در رابطه با خوب بودن و یا بد بودن یک طراحی رابط کاربری و سلیقه‌ای بودن آن، مهم‌ترین انگیزه استفاده از جمع‌سپاری است؛ همچنین مبتنی بودن تصمیمات و داده‌ها بر داده‌های کاربران مخاطب، می‌تواند منجر به موفقیت حداکثری یک محصول در سازمان شود.

همچنین به عنوان یک مهندس، همواره بر آنیم که روش‌های مهندسی و رویکردهای قابل تکرار داشته باشیم. بنابراین نتیجه تلاش در استفاده از یک روش مهندسی برای مدیریت نظرات، استفاده از جمع‌سپاری خواهد بود.

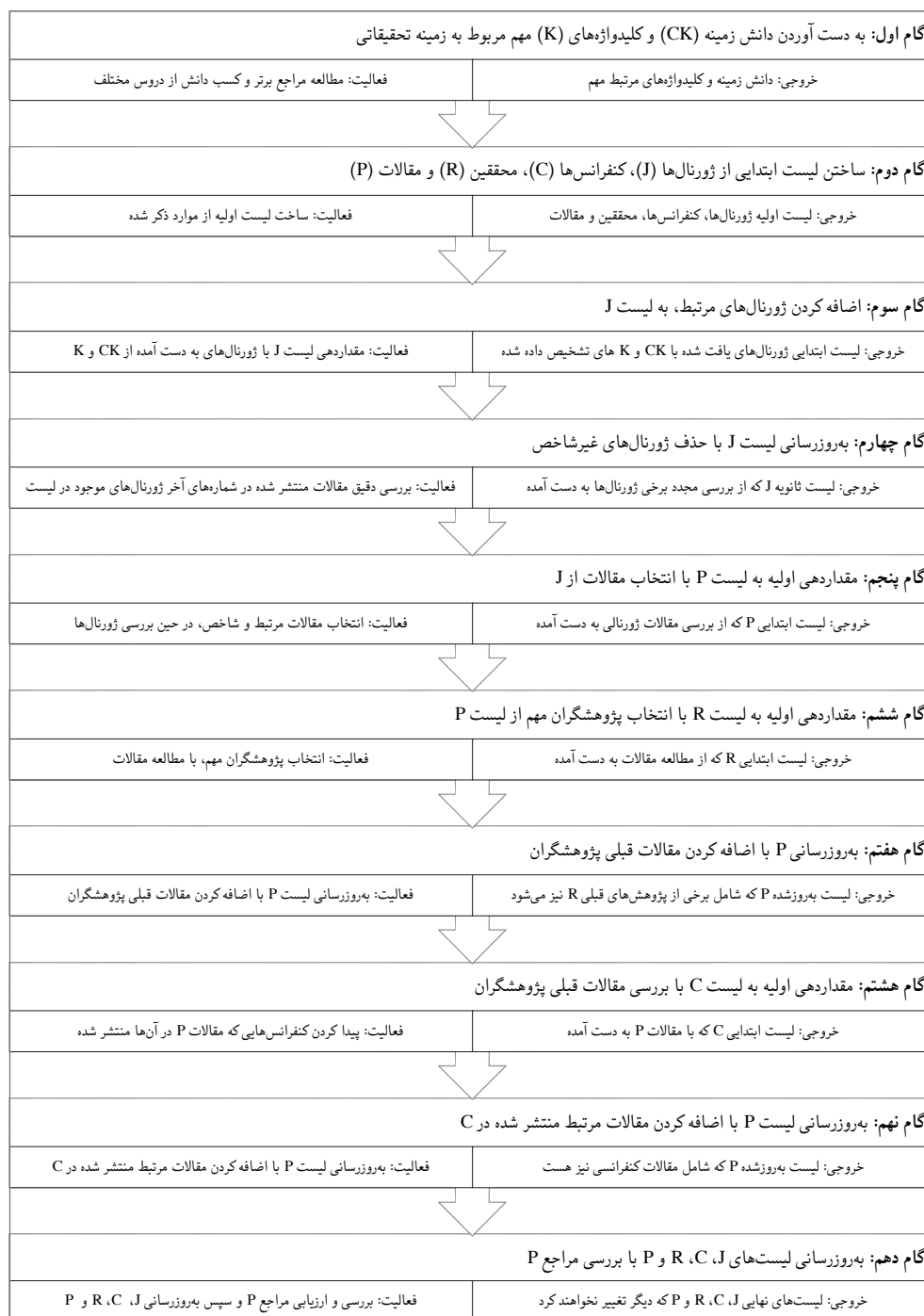
^۳Data Collection

^۱Ground Truth

فصل دوم

شیوه پژوهش

در ابتدای پژوهش و با کسب دانش زمینه مورد نیاز، به جستجوی پراکنده پرداختیم و برخی تعاریف و مفاهیم را از منابع مختلفی که در بخش مراجع ذکر شده‌اند استخراج کردیم اما به منظور انجام این پژوهش، نیازمند یک روش ساختارمند و مشخص در تحقیق بودیم که بتوانیم استدلال قوی‌تر و نتیجه‌گیری موثق‌تری داشته باشیم.



شکل ۱-۲: مدل‌لوی پژوهش آزمایشگاه سیستم‌های هوشمند و گام‌های آن

به عبارتی دیگر برای انجام این پژوهش، می‌بایست ژورنال‌ها (J)، کنفرانس‌ها (C)، محققین (R) و مقاله‌های (P) مرتبط

و موثر را شناسایی می‌کردیم.

با رجوع به شیوه پژوهش آزمایشگاه سیستم‌های هوشمند^۱ که یکی از شیوه‌های پژوهش موثر در مطالعات و پژوهش‌های موردی، در حال استفاده توسط پژوهشگران این آزمایشگاه است [نیازمند مرجع - شیوه ارائه شده توسط آقای نظریانی]، درمی‌یابیم که می‌بایست در چند تکرار و به صورت تکاملی منابع مورد نیاز برای پژوهش خود را آماده سازیم. بدین منظور، مطابق شکل ۱-۲ ابتدا به جمع‌آوری دانش موضوعی^۲ (CK) در زمینه مورد نظر پرداختیم و سپس شروع به یافتن کلیدواژه‌های مطرح (K) در این زمینه تحقیقاتی کردیم؛ از جمله این فعالیت‌ها می‌توان به مطالعه منابعی چون [۱۳]، [۳۶]، [۳۵] و [۴۳] و همچنین اخذ درس‌هایی مانند مهندسی نرم‌افزار نام برد (گام اول).

جدول ۱-۲: کلیدواژه‌های به دست آمده از دانش زمینه

ردیف	کلیدواژه
۱	Business Data Processing
۲	Crowd
۳	Crowdsourcing
۴	Crowdsourced Data Cleaning
۵	Interaction
۶	Online Experiment
۷	Software Quality
۸	Usability
۹	Usability Evaluation
۱۰	Usability Quality Metrics
۱۱	Usability Study
۱۲	Usability Quality Model
۱۳	User Experience
۱۴	User Interface
۱۵	Web Usability

سپس، چند لیست تهی برای ذخیره اطلاعات ژورنال‌ها، کنفرانس‌ها، پژوهشگران و همچنین مقالات مختلف مرتبط و موثر در این حوزه تحقیقاتی ساختیم، که به‌طور خلاصه با حروف J، C، R و P از آن‌ها یاد می‌کنیم (گام دوم). با استفاده از دانش زمینه به دست آمده و همچنین کلیدواژه‌های شناخته شده که گزیده‌ای از آن‌ها در جدول ۱-۲ قابل مشاهده است، لیست‌های J، C، R و P را به طور مرتب و به با تکرارهای متعدد، به‌روزرسانی کردیم (گام‌های سوم تا دهم) تا اینکه به منابع ذکر شده در این سند رسیدیم و منابع مورد نیاز پژوهش را، تا جایی که بتوان به نتیجه قابل اتکایی رسید، با این متدولوژی پژوهشی و با استفاده از ابزارهایی چون [۱] و [۲] و نیز [۶] پیدا کردیم. پس از شناخت دامنه و به دست آوردن منابع لازم، یک تعریف نیازمندی برای ابزار تست مورد نظر ارائه دادیم که بر اساس آن سیستم هدف و ابزار مطرح در این پژوهش ساخته شد.

^۱ISLAB Research Methodology

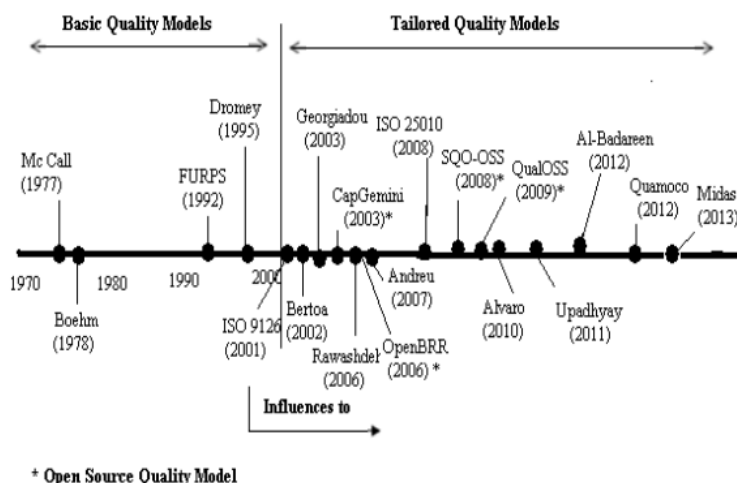
^۲Context Knowledge

فصل سوم

مروری بر کارهای گذشته، روش‌ها و ابزارهای موجود

۱-۳ مدل‌های کیفیتی

با یک نگاه اجمالی بر منابعی همچون [۴۴]، [۴۰]، [۳۶]، [۳۵] و همچنین [۴۳] که به بررسی و مقایسه تطبیقی مدل‌های کیفی پرداخته‌اند، به این نکته پی می‌بریم که صحبت از کیفیت و پژوهش در مورد مدل‌های کیفی از همان ابتدا و به صورت همزمان با پژوهش‌های مربوط به توسعه نرم‌افزار و متدولوژی‌ها مورد توجه بوده است. در شکل ۱-۳ ملاحظه می‌شود که از سال ۲۰۰۱، کم‌کم مدل‌های عام‌منظوره‌ای همچون مدل‌های مک‌کال و درومی^۱ کم‌رنگ‌تر شدند و شاهد معرفی شدن مدل‌های خاص‌منظوره بودیم.



شکل ۱-۳: خط زمانی ارائه برخی از مدل‌های کیفی [۳۵]

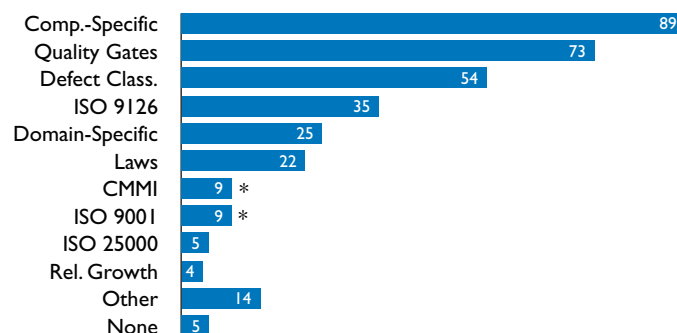
مدل‌های عام‌منظوره که در شکل با نام Basic شناخته می‌شوند، ابعاد کلی کیفیت نرم‌افزار را هدف قرار داده‌اند و تقریباً می‌توانند در هر نرم‌افزاری مورد استفاده قرار بگیرند؛ مدل‌های بعدی که ارائه شدند، روی ابعاد خاصی از سازمان یا محصول نرم‌افزاری تمرکز داشته‌اند. این مدل‌ها در نتیجه افزایش پیچیدگی محصولات نرم‌افزاری و فرایندهای سازمانی، برای استفاده در کاربردهای خاص و برای سازمان‌های خاص توسعه داده شدند [۳۵].

در بررسی مدل‌های کیفی، مرجع [۴۴] دسته‌بندی‌ای را ارائه داده است که بر اساس آن، مدل‌های کیفی را می‌توان به سه دسته سلسله‌مراتبی، مبتنی بر متامدل و همچنین مدل‌های ضمنی تقسیم‌بندی کرد که توضیحات هرکدام در ادامه به صورت مختصر قید شده است.

۱-۱-۳ مدل‌های سلسله‌مراتبی

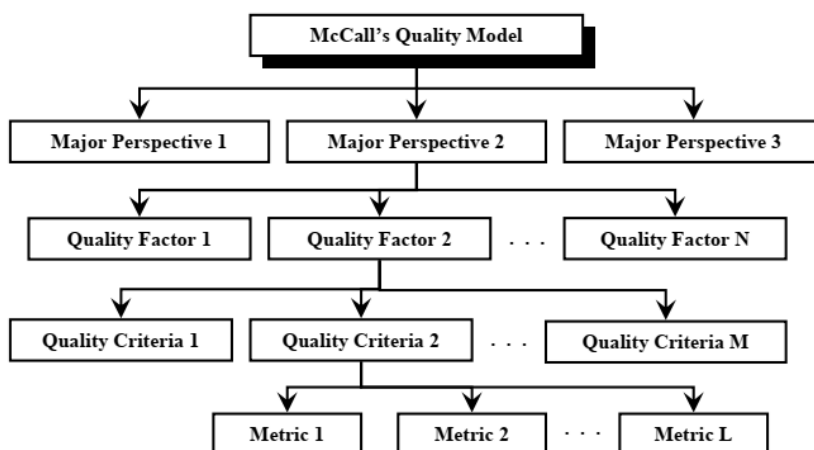
روش‌های Boehm [۲۰] و مک‌کال [۳۱] در ارائه مدل کیفی، تشابه زیادی باهم دارند؛ هر دو در خرد کردن مفهوم کیفیت، از یک روش سلسله‌مراتبی استفاده کردند و مطابق شکل ۳-۳ کیفیت را به خصله‌های مشخصی (که از آن‌ها با نام فاکتورهای کیفیت یاد می‌شود) تقسیم کرده‌اند. این‌گونه مدل‌ها در طول زمان دچار تغییراتی شدند و تفاوت نحوه تقسیم‌بندی آن‌ها، تفاوت مدل‌ها را پدید آورده است.

^۱Dromey



شکل ۳-۲: بررسی انواع مدل/رویه‌های کیفیتی استفاده شده در سازمان‌ها [۴۵]

به تعبیر واگنر [۴۴] رویکرد این نوع مدل‌های کیفی، خرد کردن کیفیت به معیارهای قابل اندازه‌گیری و در نهایت اندازه‌گیری و مقایسه آن‌هاست. همچنین واگنر در بررسی خود، از نقدهایی همچون «مبهم بودن برخی از این تقسیم‌بندی‌ها و شفاف نبودن آن‌ها به طور کامل» یاد می‌کند که از عوامل مهم ناکارآمدی برخی از آن‌هاست؛ همچنین وی در سال ۲۰۱۲ این نکته را متذکر شد که تنها کمتر از ۲۸٪ سازمان‌های فعال در حوزه نرم‌افزار، از مدل‌های استاندارد در تضمین کیفیت فعالیت‌ها و محصولاتشان استفاده می‌کنند و ۷۱٪ این سازمان‌ها، مدل‌های کیفی خود را از روی این مدل‌های کیفی، گلچین کرده و شخصی‌سازی می‌کنند. [۴۵]



شکل ۳-۳: ساختار مدل کیفی مک‌کال به عنوان یک مدل کیفی سلسله‌مراتبی [۱۲]

همانطور که در شکل ۳-۲ مشاهده می‌شود، طی این بررسی، ۷۹ سازمان مدل‌های کیفیتی شخصی‌سازی شده توسط سازمان خود را در اولویت قرار داده و از آن‌ها استفاده می‌کنند. طبق اظهارات این بررسی و همچنین بسیاری از منابع دیگر همچون [۳۶] و [۴۳]، نیاز برای شخصی‌سازی مدل‌های کیفیتی استاندارد وجود دارد. چرا که این مدل‌های سلسله‌مراتبی، به صورت تجریدی بیان شده‌اند و نیازمند دقیق شدن روی متریک‌ها و روش‌های اندازه‌گیری هر متریک هستند.

۳-۱-۲ مدل‌های مبتنی بر متامدل

با آشکار شدن این نیاز که می‌بایست مدل‌های پایه‌ای را بیشتر شفاف‌سازی کرد و آن‌ها را بر نیازمندی‌ها تطبیق بیشتری داد، ایده ارائه متامدل‌ها مطرح شد. متامدل در اصل مدلی از یک مدل کیفی است؛ قواعد و ساختارهایی که برای توصیف دقیق یک مدل کیفی نیاز داریم (همچون متریک‌ها و نحوه اندازه‌گیری آن‌ها)، توسط متامدل تعیین می‌گردند [۲۱]. به عبارت دیگر، توصیف اینکه چگونه یک مدل کیفی می‌تواند بر نیازمندی‌ها منطبق شود، به عهده متامدل است [۴۴]. درومی به عنوان مثال، در سال ۱۹۹۵، متامدل نسبتاً مفصلی ارائه داد که ذیل آن، میان مولفه‌های محصول نرم‌افزاری (که باید حامل کیفیت باشند - مانند کد منبع نرم‌افزار) و ویژگی‌های عملیاتی نرم‌افزار تفاوت و تمایز قائل شد [۲۲].

۳-۱-۳ مدل‌های آماری و ضمنی

این مدل‌ها سعی در ترجمه مفهوم اطمینان‌پذیری سخت‌افزار و استفاده آن در حوزه نرم‌افزار را دارند. ایده اصلی استفاده از این مدل‌ها، مشاهده خرابی‌ها در طول زمان و پیش‌بینی روند رخ دادن خرابی‌ها در آینده است. به منظور دستیابی به خصیصه‌های کیفیتی مشخصی که برای نیازهای توسعه محور بیان شده، در برخی از مدل‌ها سعی شده از داده‌های آماری برای به دست آوردن برخی از ویژگی‌ها و متریک‌ها استفاده شود.

به عنوان مثالی برای این نوع مدل‌ها می‌توان به مدل‌های رشد اعتمادپذیری^۱ [۳۲] اشاره کرد. مدل‌هایی که از الگوریتم‌ها و روش‌های یادگیری ماشین برای تخمین موارد مختلف، از قبیل مولفه‌های آسیب‌پذیر و یا غیر کارا و مولفه‌هایی که دارای برخی از ویژگی‌های کیفی خاص نیستند نیز از این نوع‌اند که نمونه‌ای از این مدل‌ها در مرجع [۳۳] تحت عنوان Vulture یاد شده؛ این مدل، از روش‌های یادگیری ماشین و از یک پایگاه دانش آسیب‌پذیری استفاده می‌کند تا در طول زمان و با گسترش نرم‌افزار، بتواند مولفه‌های آسیب‌پذیر نرم‌افزار را پیش‌بینی کند.

همچنین به تعبیر مرجع‌های [۴۳] و [۴۴]، ابزارها و روش‌های مرور، داشبوردهای مدیریتی و مصورسازی داده، ابزارهای شناخت الگوی رخداد خطا در کد منبع نرم‌افزار و چک‌لیست‌ها، که شاید در ظاهر به طور مستقیم ارتباطی با مدل‌های کیفی نداشته باشند، اما در نهایت به یک یا چند متریک کیفی در ذیل یک مدل کیفی ختم می‌شوند؛ این اشاره به مدل‌های کیفی به صورت ضمنی و غیرصریح بوده و اغلب به طور دقیق ارتباط خود با مدل‌ها را مشخص نکرده‌اند. در نتیجه‌ی تمام موارد ذکر شده، تضمین و کنترل کیفیت به واسطه این سازوکارهای غیرصریح و مدل‌هایی که به طور ضمنی مطرح هستند، منجر به پیچیدگی بیشتر و سختی کار خواهد شد.

۳-۲ تمرکز بر استفاده‌پذیری روی مدل‌های کیفیتی

در جدول ۳-۱ مقایسه‌ای تطبیقی میان مدل‌های مطرح از سال ۱۹۷۰ تا ۲۰۱۱ انجام شده است که در انجام این مقایسه، به طور خاص، روی خصیصه استفاده‌پذیری این مدل‌ها تمرکز داشتیم. مدل‌های عام‌منظوره در کنار سایر مدل‌ها به مقایسه درآمده‌اند تا خصیصه‌های استفاده‌پذیری در هر کدام از آن‌ها بررسی شود؛ مدل‌های خاص‌منظوره به خاطر نیاز سازمان خاصی به وجود آمده‌اند که مشتریان مخصوص به خود را داشتند که در صورت تعویض محصول و مشتری و استفاده مدل مفروض در یک سازمان دیگر، الزاماً به جواب بهینه منتهی نخواهد شد. در حقیقت ویژگی اصلی مدل‌های کیفیتی مبتنی بر متامدل و دلیل گستردگی آن‌ها، متفاوت بودن نیازهای مشتریان و شرایط سازمان‌هاست [۴۳].

^۱Reliability Growth Models

جدول ۳-۱: مقایسه تطبیقی مدل‌های کیفیتی ارائه شده با تمرکز بر استفاده‌پذیری

ردیف	مدل کیفیتی ارائه شده	سال ارائه	متریک‌های استفاده‌پذیری	مرجع
۱	McCall	۱۹۷۰	Communicativeness Training, Operability,	[۳۱]
۲	Boehm	۱۹۷۶	Maintainability Portability,	[۲۰]
۳	IEEE 1061	۱۹۹۰	Communicativeness Learning, of Ease Comprehensibility,	[۳۷]
۴	Shackel	۱۹۹۱	Pleasing Subjectively Flexibility, Learnability, Effectiveness,	[۴۱]
۵	Bevan	۱۹۹۱	Acceptability Use, of Ease User, of Type Product, of Type	[۱۹]
۶	FURPS	۱۹۹۲	Training of Material user, the of Documentation Aesthetic, Factors, Human	[۲۵]
۷	Nielsen	۱۹۹۴	Satisfaction Errors, Memorability, Efficiency, Learnability,	[۳۴]
۸	ISO 9126	۲۰۰۱	Compliance Usability Attractiveness, Operability, Learnability, Understandability,	[۲۷]
۱۰	Bertoa	۲۰۰۲	Operability Learnability, Understandability,	[۱۸]
۱۱	Georgiadou	۲۰۰۳	Consistency Help, Online Update, Documentation Learnability, Support,	[۲۴]
۱۳	Abrian	۲۰۰۳	Security Learnability, Satisfaction, Effectiveness, Efficiency,	[۱۰]
۱۴	Bass	۲۰۰۳	Security Performance, Reusability, Scalability, Modifiability,	[۱۷]
۱۵	Schneiderman	۲۰۰۵	Subjective time, over Retention users, by Errors of Rate Performance, of Speed learn, to Time Satisfaction	[۴۲]
۱۶	Rawashdeh	۲۰۰۶	Complexity Operability, Learnability, Understandability,	[۲۸]
۱۷	ISO 25010	۲۰۰۸	Accessibility Aesthetics, Interface User Protection, Error User Operability, Learnability, Recognizability, Appropriateness,	[۴]
۱۸	Alvaro	۲۰۱۰	Operability Learnability, Understandability,	[۱۶، ۱۵]
۱۹	Alonso-Rios و بقیه	۲۰۱۰	Satisfaction Subjective Safety, Robustness, Efficiency, Operability, Knowability,	[۲۴]
۲۰	Dubey و بقیه	۲۰۱۱	Learnability Satisfaction, Efficiency, Effectiveness,	[۳۰]

شایان ذکر است که در همه مدل‌های ذکر شده، الزاما به استفاده‌پذیری به عنوان یک خصیصه اصلی در محصول اشاره نشده است؛ در بعضی از مدل‌ها همچون ISO 25010 استفاده‌پذیری یکی از سطوح اصلی بوده، در اولین سطح سلسه‌مراتبی مدل قرار داشته و جزئی از محصول نهایی است و در برخی دیگر همچون Boehm، به طور صریح و مشخص به استفاده‌پذیری اشاره‌ای نشده است اما در فرآیندهای توسعه محصول روی آن توجه زیادی وجود دارد.

همچنین از بررسی مدل‌های کیفیتی مختلف که صرفا برای توسعه سامانه‌های مبتنی بر وب این نتیجه برمی‌آید که هر متامدل می‌بایست در زمینه مربوط به خود مورد استفاده قرار گیرد و نه جای دیگر [۹]. نتیجه پیشین به این معنی است که در توسعه سامانه‌های مبتنی بر وب، محدوده کاربران، دانش قبلی آن‌ها، تخصص هرکدام، سن و سایر متغیرهای غیرقابل کنترل توسط توسعه‌دهنده نیز در استفاده‌پذیر بودن این سامانه تحت وب تاثیرگذار است؛ بنابراین در طراحی رابط کاربری هر سامانه مبتنی بر وب، می‌بایست به این نکات نیز توجه داشت و از آخرین توصیه‌های مربوط به توسعه این نوع سامانه‌ها استفاده کرد [۱۳]. از جمله این توصیه‌ها و پیشنهادهای طراحی، توصیه‌های گوگل برای ساخت اپلیکیشن‌های پیشرو^۱ [۵] است که جدیداً مطرح شده و طبق بررسی‌های انجام شده آینده‌ای روشن در انتظار این نوع از اپلیکیشن‌هاست.

۳-۳ مطالعه استفاده‌پذیری و ارزیابی تجربه کاربری

متریک‌هایی که در مطالعه استفاده‌پذیری و به طور خاص هنگام بررسی تجربه کاربری، اندازه‌گیری می‌شوند و مورد سنجش قرار می‌گیرند داده‌هایی را به دست ما می‌دهند که در بررسی و استفاده از این داده‌ها می‌توان دو رویکرد کلی داشت [۱۳]: ارزیابی خرد و ارزیابی کلان^۲.

همانطور که در شکل ۳-۴ دیده می‌شود، می‌توان این دو نوع ارزیابی را به غذایی بدیل کرد که توسط آشپز و مشتری چشیده می‌شوند؛ آشپز در فرآیند پختن غذا به طور مرتب ممکن است غذا را بچشد تا در نهایت خروجی مطلوبی به دست مشتری برسد و غذا از کیفیت لازم برخوردار باشد. در حالی که مشتری در نهایت، محصول نهایی را مشاهده می‌کند و صرفا نظر خود در مورد آن غذا و یا کیفیت رستوران را اعلام می‌کند. ذکر این نکته شایان ذکر است که به وضوح می‌توان دریافت که هزینه اعمال تغییرات در صورت درخواست مشتری از آشپز زیاد خواهد بود؛ به طور مشابهی، در صورت عرضه محصول نرم‌افزاری، هزینه تعمیر یک خرابی به مراتب بیشتر از مرور در حین تولید است.

^۱Progressive Web Applications

^۲ در سال ۱۹۸۶ و طی مقاله‌ای با عنوان «نقش ارزیابی مستمر و چرخشی در طراحی سیستم‌ها برای کیفیت» [۲۶]، دو اصطلاح برای مطالعه تجربه کاربری و استفاده‌پذیری Formative و Summative مورد استفاده قرار گرفتند که هر دو از مفاهیم کلاس درسی برداشت شده‌اند؛ یک ارزیابی مستمر (Formative) به معنی پرسیدن سوال در سر کلاس درس توسط معلم بوده و به صورت تدریجی و خرد خرد است؛ در حالی که یک ارزیابی کلان (Summative) مطالعه‌ای است که در انتهای هر بازه (مثلا هنگام امتحانات پایان‌ترم) و با برگزاری آزمونی خاص، ارزیابی‌ها انجام می‌شوند. در مقاله ذکر شده همچنین این مورد مطرح می‌شود که که ارزیابی مستمر و خرد برای رسیدن به دقت بالا در برآورد نیازهای مشتری، بهتر است.



شکل ۳-۴: مثالی از دو نوع ارزیابی مختلف خرد و کلان [۳]

۱-۳-۳ ارزیابی خرد

در یک مطالعه استفاده‌پذیری با رویکرد ارزیابی خرد، محقق به طور مستمر و به صورت دوره‌ای، محصول نهایی را مورد بررسی قرار می‌دهد و در تمامی مراحل تولید نواقص آن را سنجیده و کشف می‌کند و پیشنهاداتی برای رفع آن نواقص ارائه می‌دهد؛ این روند تا آن جا ادامه پیدا می‌کند که نهایتاً یک محصول تقریباً ایده‌آل و یا یک محصول خوب به اندازه کافی^۳ به دست آید. درواقع هدف در این نوع مطالعه هدف بهبود مستمر و رفع ایرادات محصول قبل از عرضه نهایی آن است؛ در نتیجه با بررسی فرایندهای نرم‌افزاری و همچنین مطالعاتی از قبیل [۴۳]، [۲۹] و [۱۳] به نظر می‌رسد که هرچه ارزیابی خرد زودتر رخ دهد، تاثیر بیشتری روی محصول نهایی و افزایش کیفیت آن خواهد داشت.

با اتخاذ این رویکرد، برخی از سوالاتی که می‌توان در فرایند طراحی پرسید عبارتند از:

- مهم‌ترین مواردی که کاربران را از رسیدن به اهدافشان منع می‌کند و یا به عدم کارایی آن‌ها می‌شود چیست؟
 - نقاط قوت و ضعف محصول از نقطه نظر کاربران چیست؟
 - اشتباهات متداول کاربران هنگام کار با محصول حول چه مواردی است؟
 - آیا بهبودهای مطرح شده توسط محققین تجربه کاربری، در هر نسخه از طراحی رابط کاربری، مورد استفاده و توجه قرار می‌گیرند؟
 - پس از عرضه نهایی محصول، چه مواردی در رابطه با استفاده‌پذیری به نظر می‌رسد که هنوز جای کار خواهد داشت؟
- شایان ذکر است که در صورتی که فرصت اصلاح طراحی واسط کاربری وجود نداشته باشد، استفاده از این روش ارزیابی به نظر می‌رسد که کارایی چندانی نداشته باشد و بیشتر باعث هدررفت منابع شود.

۲-۳-۳ ارزیابی کلان

در این روش، محقق همچون یک منتقد، محصول نهایی را از زوایای مختلف مورد بررسی قرار می‌دهد و حتی با محصول‌های دیگر مقایسه می‌کند تا نقدی بر آن وارد سازد. نکته حائز اهمیت این است که در اینجا محصول ارائه شده است و دیگر در فاز توسعه و تولید نیست. هدف از انجام این نوع ارزیابی، پی بردن به این نکته است که این محصول خاص چه قدر خوب می‌تواند

^۳ Good Enough

به نیازمندی‌های کاربران پاسخ دهد و به چه میزان با آن‌ها هم‌جهت است. بر خلاف ارزیابی خرد، این روش، مبتنی بر اصول و قواعد و چک‌لیست‌های مشخصی است که در نهایت محصول با آن‌ها بررسی می‌شود. در مقایسه محصولات مختلف نیز مجدداً این اصول و قواعد مینا قرار می‌گیرند. با بررسی منابع مختلفی از قبیل [۲۳] و [۳۶] می‌توان به این نکته پی برد که سولاتی از قبیل سولات زیر بیشتر مناسب انجام این نوع ارزیابی هستند:

- آیا اهداف استفاده‌پذیری پروژه (مطرح شده در نیازمندی‌های پروژه) رعایت شده‌اند؟
- استفاده‌پذیری کلی سیستم در چه سطحی است؟
- نقاط ضعف و قوت محصول مورد نظر در مقایسه با سایر رقبا چیست و چگونه می‌توان در صورت داشتن ضعف، آن را ارتقا داد؟
- آیا بهبودهای مطرح برای هر نسخه از نرم‌افزار، پس از عرضه نسخه جدید، اعمال می‌شوند؟

در نهایت فراموش نکنیم که همواره تغییر نیازمند صرف هزینه و زمان است؛ بنابراین در صورت استفاده از این روش ارزیابی می‌بایست در نظر داشت که برخی فعالیت‌های پسا ارزیابی نیز باید در پس ذهن مدیر پروژه باشد؛ چرا که ممکن است حتی در صورت نیاز پروژه‌ای برای برطرف کردن مشکلات استفاده‌پذیری یک سیستم تعریف شود که خود این پروژه هزینه‌بر باشد.

۳-۳-۳ اهداف کاربری

سولاتی همچون «آیا محصول مورد نظر نیاز روزانه کاربران را برآورده خواهد کرد و کاربران به طور متداول با این محصول نرم‌افزاری در ارتباط خواهند بود؟» و نیز «آیا کارایی کاربران و بهره‌وری آن‌ها در طول انجام یک وظیفه مشخص در هنگام کار با این نرم‌افزار مهم است؟ چگونه می‌توان آن را بهبود داد؟» به قسمتی از محصول توجه دارند که با نیازمندی‌های کاربر درگیر است. با بررسی مراجعی همچون [۱۳]، [۹]، [۲۶] و [۱۰] می‌توان به این نکته پی برد که همه این قبیل سولات که به نیازهای ضمنی و نه الزاماً صریح کاربر، در تعامل با رابط کاربری می‌پردازند، به دو متریک اساسی و قابل اندازه‌گیری از نیاز کاربران اشاره می‌کنند:^۱ کارایی و رضایت کاربر^۲.

کارایی به عنوان یک متریک کیفیتی به طور خاص در رابطه با تجربه کاربری، به اندازه‌گیری توانایی کاربران در انجام وظایف مشخصی می‌پردازد؛ در این راستا، اندازه‌گیری‌های جنبی نیز اهمیت زیادی پیدا می‌کنند. از جمله این اندازه‌گیری‌ها می‌توان به موارد زیر اشاره کرد که به طور غیر مستقیم در کارایی تاثیرگذار هستند:

- زمان سپری شده برای انجام وظیفه
- میزان تلاش برای انجام وظیفه (برای مثال تعداد کلیک‌ها و یا توان ذهنی مصرف شده)
- زمانی که طول می‌کشد تا کاربر با وظیفه آشنا شود و بدون صرف تلاش خاصی آن را انجام دهد (یادگیری)

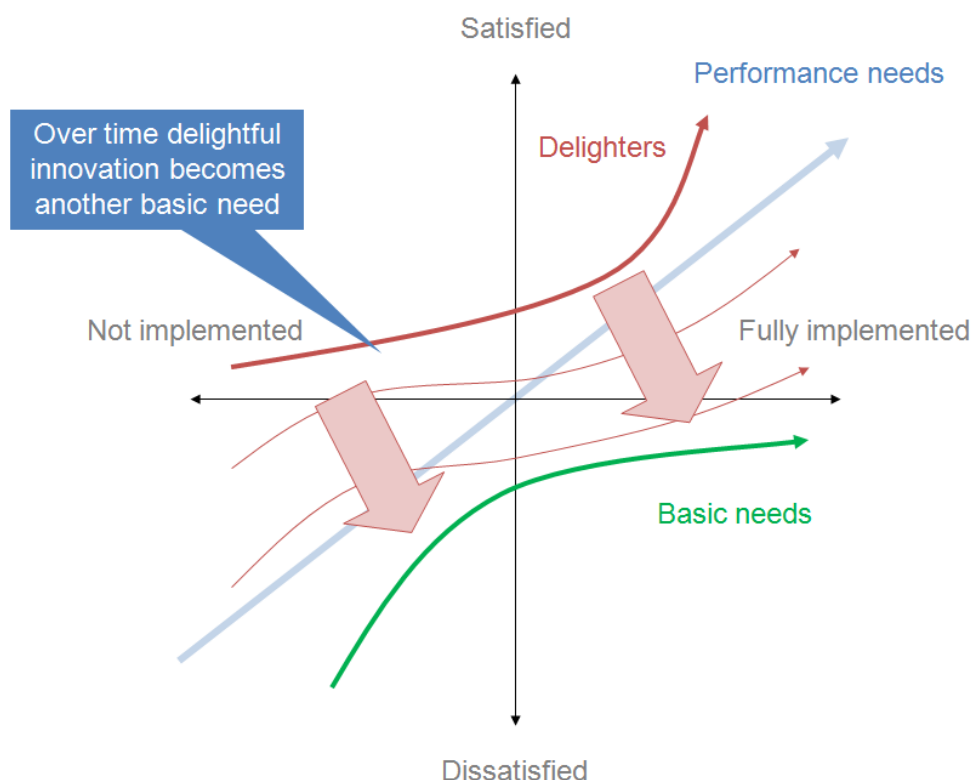
اندازه‌گیری‌های مربوط به متریک کارایی یک رابط کاربری، اهمیت زیادی دارند چرا که اگر کاربران نتوانند وظایف اصلی در رابطه با تعامل با سیستم را به درستی و با موفقیت به انجام برسانند، در عمل محصول نرم‌افزاری به شکست منتهی شده است و یا حداقل رابط کاربری خوبی ندارد و امکان تعامل موفق کاربر وجود نخواهد داشت.

^۱ ذکر این نکته خالی از لطف نیست که کارایی کاربر و رضایت کاربر در اینجا اشاره به دید کاربر به سامانه هدف دارند و نه اینکه متریک‌های اصلی مدل کیفیتی باشند. در اینجا کارایی و رضایت از دید کاربر و به عنوان نظر وی در مورد سامانه مورد نظر است.

^۲Performance

^۳Satisfaction

رضایت در واقع نظر نهایی کاربر در مورد تعاملش با سیستم است؛ قضاوتی که کاربر در مورد سیستم و نحوه تعاملش با آن می‌کند می‌تواند با جملات مختلفی مانند «استفاده از آن سخت/آسان بود»، «گیج‌کننده/ساده بود» و ... بیان شود. البته که این تعبیرات غیردقیق هستند اما می‌توان با اعطای درجه‌های آزادی خاصی به کاربران، در حین تعامل با سیستم برخی از متریک‌ها را از آن‌ها به طور خوداعلامی^۴ از کاربران گزارش عددی گرفت؛ چه بسا که به گفته مراجعی همچون [۱۳]، [۱۴] و [۴۰] این متریک‌ها در سامانه‌های کاربردی مبتنی بر وب - که هدف اصلی این پروژه هستند - بسیار مهم و تاثیرگذارند. اما باید به این نکته توجه کرد که در محدوده سامانه‌های مبتنی بر وب، رضایت کاربر الزاما همیشه همراه با کارایی حداکثری وی در تعامل با سامانه نیست؛ فاکتورهای بسیاری از قبیل زیبایی و وجود تکنولوژی‌های مختلف، بر این رضایت تاثیر مستقیم دارند و چه بسا که کاربری با رضایت حداکثری از یک سامانه استفاده کند ولی کارایی عملیات وی بسیار پایین باشد.



شکل ۳-۵: مدل کانو که نشان‌دهنده ارتباط رضایت کاربر و ویژگی‌های محصول است [۸]

مدلی توسط آقای کانو در سال ۱۹۸۴ [۲۸] برای تمییز دادن ویژگی‌های اشتیاق‌برانگیز و صریح و همچنین ویژگی‌های ضمنی و بایندی یک محصول، ارائه شد. طی این مدل، کیفیت محصول نهایی در گرو پنج دسته از نیازمندی‌های زیر است که رسیدن به هر دسته از این‌ها نیازمند اتخاذ سیاست‌های مختلف در طول ساخت محصول است:

- نیازمندی‌های بایندی: که مشتریان به طور ضمنی خواستار آن‌ها هستند و ممکن است صریحا بیان نشوند. به عنوان مثال اینکه یک سامانه کاربردی تحت وب همیشه با یک آدرس اینترنتی خاص URL شناخته شود.
- نیازمندی‌های تک‌بعدی: که در صورت وجودشان کاربر احساس رضایتمندی و در صورت عدم وجودشان در محصول نهایی، کاربر احساس عدم رضایت از محصول را خواهد داشت. برای نمونه می‌توان به واکنش‌گرا بودن یک سامانه تحت وب روی پلتفرم موبایل اشاره کرد؛ که گفتنی است این روزها به یکی از ویژگی‌های اصلی موفقیت بسیاری از کسب‌وکارهای فعال در ایران تبدیل شده است.

^۴Self-reported Metrics

- نیازمندی‌های اشتیاق‌برانگیز: که در صورتی که به طور کامل پیاده‌سازی شوند، منجر به رضایتمندی کاربران خواهند شد ولی در صورت عدم پیاده‌سازی، رضایت کاربران از بین نخواهد رفت. به عنوان مثال اینکه یک سامانه پست الکترونیکی تحت وب^۱ در کنار لیست ایمیل‌های دریافتی، وضعیت آب‌وهوا و زمان فعلی و گزیده‌ای از اخبار را نشان دهد می‌تواند یک ویژگی اشتیاق‌برانگیز باشد.

- نیازمندی‌های بی‌تفاوت: بودن و نبودنشان تفاوتی در رضایت مشتری نخواهد کرد. به عنوان مثال در بسیاری از پروژه‌های منتهی به یک سامانه کاربردی تحت وب، پلتفرم و زبان مورد استفاده برای توسعه سامانه، تفاوتی در رضایت مشتریان ایجاد نخواهد کرد.

- نیازمندی‌های معکوس: به دسته‌ای از نیازمندی‌ها اشاره دارد که پرداختن بیش‌ازحد به آن‌ها باعث کاهش رضایت کاربران می‌شود. به عنوان مثال برخی از کاربران ممکن است از ابزارهایی که امکانات زیادی به آن‌ها در داشبورد مدیریتی می‌دهند خوششان بیاید و در مقابل برخی از کاربران از پیچیدگی بیش از حد ابزار گلایه کنند.

در شکل ۳-۵ ملاحظه می‌شود که بسیاری از ویژگی‌های جذاب محصول که هنوز به عنوان نیازمندی مطرح هستند و هنوز پیاده‌سازی نشده‌اند و در نتیجه کاربر امکان انجام عملیات مورد نظر خود را ندارد، انگیزه‌ای برای ساخت سامانه هستند و پس از اینکه این ویژگی‌های عملیاتی (منحی قرمز رنگ) در محصول پدیدار می‌شوند، رضایت کاربران از محصول افزایش پیدا می‌کند؛ گرچه الزاما شاید این ویژگی‌ها، کارایی بالایی از دید کاربران نداشته باشند. به مرور زمان که فناوری پیشرفت می‌کند، نیازمندی‌های فعلی آهسته آهسته به بایدهای سامانه تبدیل می‌شوند (منحنی سبز رنگ).

همچنین در مرجع [۳۹] از خط آبی قابل مشاهده در شکل ۳-۵، به عنوان نیازمندی‌های تک‌بعدی یاد شده است که مشتری فقط به طور صریح و مشخص، این دسته از نیازمندی‌ها را مطرح می‌کند و بقیه نیازمندی‌ها معمولا به طور ضمنی مطرح می‌شوند. نتیجه نهایی از دو بحث پیشین در مورد رضایت کاربر از سامانه و کارایی کاربر در تعامل با سامانه، اینکه این دو خصیصه الزاما دارای همبستگی خاصی نیستند؛ ولی همواره باید در اندازه‌گیری استفاده‌پذیری مدنظر قرار بگیرند چرا که طبق تعریف استفاده‌پذیری، در یک سیستم استفاده‌پذیر، کاربر می‌بایست در نهایت از سیستم راضی بوده باشد و تجربه کاربری خوبی داشته باشد.

۳-۴ خصیصه‌های استفاده‌پذیری و اندازه‌گیری آن

با بررسی مدل‌های کیفی

۳-۵ ابزارهای سنجش استفاده‌پذیری

از جمله

^۱Webmail

منابع و مراجع

- [1] Computing Research & Education.
- [2] dblp: computer science bibliography.
- [3] Formative and Summative Evaluation.
- [4] ISO 25010.
- [5] Progressive Web Apps | Web | Google Developers.
- [6] Scimago Journal & Country Rank.
- [7] Bluffton University bus crash, May 2018. Page Version ID: 843580777.
- [8] Kano model, June 2018. Page Version ID: 847738497.
- [9] MeasuringU: The User Experience of University Websites, July 2018.
- [10] Abran, Alain, Khelifi, Adel, Suryn, Witold, and Seffah, Ahmed. Usability meanings and interpretations in ISO standards. *Software quality journal*, 11(4):325–338, 2003.
- [11] Agarwal, Ritu and Venkatesh, Viswanath. Assessing a Firm’s Web Presence: A Heuristic Evaluation Procedure for the Measurement of Usability. *Information Systems Research*, 13(2):168–186, June 2002.
- [12] Al-Qutaish, Rafa E. Quality Models in Software Engineering Literature: An Analytical and Comparative Study. page 10, 2010.
- [13] Albert, William and Tullis, Thomas. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier: Morgan Kaufmann, 2 edition, July 2013.
- [14] Alonso-Ríos, David, Vázquez-García, Ana, Mosqueira-Rey, Eduardo, and Moret-Bonillo, Vicente. Usability: a critical analysis and a taxonomy. *International Journal of Human-Computer Interaction*, 26(1):53–74, 2009.
- [15] Alvaro, Alexandre. Quality Attributes for a Component Quality Model. page 8, 2005.
- [16] Alvaro, Alexandre, Santana de Almeida, Eduardo, and Romero de Lemos Meira, Silvio. A Software Component Quality Framework. *SIGSOFT Softw. Eng. Notes*, 35(1):1–18, January 2010.

- [17] Bass, Len and John, Bonnie E. Linking usability to software architecture patterns through general scenarios. *Journal of Systems and Software*, 66(3):187–197, 2003.
- [18] Bertoa, Manuel F and Vallecillo, Antonio. Quality attributes for COTS components. 2002.
- [19] Bevan, N, Kirakowski, J, and Maissel, J. What is Usability? In *Proceedings of the 4th international Conference on HCI*, 1991.
- [20] Boehm, Barry W, Brown, John R, and Lipow, Mlity. Quantitative evaluation of software quality. In *Proceedings of the 2nd international conference on Software engineering*, pages 592–605. IEEE Computer Society Press, 1976.
- [21] Deissenboeck, Florian, Juergens, Elmar, Lochmann, Klaus, and Wagner, Stefan. Software quality models: Purposes, usage scenarios and requirements. In *Software Quality, 2009. WOSQ'09. ICSE Workshop on*, pages 9–14. IEEE, 2009.
- [22] Dromey, R. Geoff. A model for software product quality. *IEEE Transactions on software engineering*, 21(2):146–162, 1995.
- [23] Estellés-Arolas, Enrique and González-Ladrón-De-Guevara, Fernando. Towards an Integrated Crowdsourcing Definition. *J. Inf. Sci.*, 38(2):189–200, April 2012.
- [24] Georgiadou, Elli. GEQUAMO—a generic, multilayered, customisable, software quality model. *Software Quality Journal*, 11(4):313–323, 2003.
- [25] Grady, Robert B. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [26] Hewett, T T. The Role of Iterative Evaluation in Designing Systems for Usability. In *Proceedings of the Second Conference of the British Computer Society, Human Computer Interaction Specialist Group on People and Computers: Designing for Usability*, pages 196–214, New York, NY, USA, 1986. Cambridge University Press.
- [27] ISO/IEC. ISO/IEC 9126: Information Technology—Software Product Evaluation—Quality Characteristics and Guidelines for their Use. 1991.
- [28] KANO, N. Attractive quality and must-be quality. *Hinshitsu (Quality, the Journal of Japanese Society for Quality Control)*, 14:39–48, 1984.
- [29] Krug, Steve. *Don't make me think!: a common sense approach to Web usability*. Pearson Education India, 1 edition, 2000.
- [30] KumarDubey, Sanjay, Rana, Ajay, and Sharma, Arun. Usability Evaluation of Object Oriented Software System using Fuzzy Logic Approach. *International Journal of Computer Applications*, 43(19):1–6, April 2012.
- [31] McCall, Jim A, Richards, Paul K, and Walters, Gene F. *Factors in software quality*, volume 1,2,3. General Electric, 1977.
- [32] Musa, John D. *Software reliability engineering: more reliable software, faster and cheaper*. Tata McGraw-Hill Education, 2004.

- [33] Neuhaus, Stephan, Zimmermann, Thomas, Holler, Christian, and Zeller, Andreas. Predicting vulnerable software components. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 529–540. ACM, 2007.
- [34] Nielsen, Jakob. *Usability engineering*. Elsevier, 1994.
- [35] P. Miguel, José, Mauricio, David, and Rodríguez, Glen. A Review of Software Quality Models for the Evaluation of Software Products. *International Journal of Software Engineering & Applications*, 5(6):31–53, November 2014.
- [36] Pressman, Roger S. *Software engineering: a practitioner's approach*. McGraw-Hill Education, New York, NY, eighth edition edition, 2015.
- [37] Radatz, Jane, Geraci, Anne, and Katki, Freny. IEEE standard glossary of software engineering terminology. *IEEE Std*, 610121990(121990):3, 1990.
- [38] Rawashdeh, Adnan and Matalkah, Bassem. A New Software Quality Model for Evaluating COTS Components. *Journal of Computer Science*, 2(4):373–381, April 2006.
- [39] Sauerwein, Elmar, Bailom, Franz, Matzler, Kurt, and Hinterhuber, Hans H. The Kano model: How to delight your customers. In *International Working Seminar on Production Economics*, volume 1, pages 313–327, 1996.
- [40] Seffah, Ahmed, Donyaee, Mohammad, Kline, Rex B., and Padda, Harkirat K. Usability measurement and metrics: A consolidated model. *Software Quality Journal*, 14(2):159–178, June 2006.
- [41] Shackel, Brian. Usability-context, framework, definition, design and evaluation. *Human factors for informatics usability*, pages 21–37, 1991.
- [42] Shneiderman, Ben and Plaisant, Catherine. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Addison Wesley, 4 edition, 2004.
- [43] Sommerville, Ian. *Software engineering*. Always learning. Pearson Education Limited, Boston, Mass. Amsterdam Cape Town, tenth edition, global edition edition, 2016. OCLC: 934508916.
- [44] Wagner, Stefan. *Software Product Quality Control*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [45] Wagner, Stefan, Lochmann, Klaus, Winter, Sebastian, Goeb, Andreas, Klaes, Michael, and Nunnenmacher, Sabine. Software Quality Models in Practice Survey Results. *Technical Report TUM-I128*, page 24, 2012.

[۴۶] عبدالله‌زاده بارفروش، احمد. کلیات متدولوژی تامین کیفیت. نشر آدینه، تهران، ویرایش اول، ۱۳۸۹.

پیوست

موضوعات مرتبط با متن گزارش پایان نامه که در یکی از گروه‌های زیر قرار می‌گیرد، در بخش پیوست‌ها آورده شوند:

۱. اثبات‌های ریاضی یا عملیات ریاضی طولانی.
۲. داده و اطلاعات نمونه (های) مورد مطالعه (Case Study) چنانچه طولانی باشد.
۳. نتایج کارهای دیگران چنانچه نیاز به تفصیل باشد.
۴. مجموعه تعاریف متغیرها و پارامترها، چنانچه طولانی بوده و در متن به انجام نرسیده باشد.

کد میپل

```
with(DifferentialGeometry):  
with(Tensor):  
DGsetup([x, y, z], M)  
frame name: M  
a := evalDG(D_x)  
D_x  
b := evalDG(-2 y z D_x+2 x D_y/z^3-D_z/z^2)
```

واژه‌نامه‌ی فارسی به انگلیسی

Degree	درجه	آ	
	ر		
Satisfaction	رضایت	استفاده‌پذیری	Usability
	ز	ارزشیابی خرد	Formative Evaluation
Submodule	زیرمدول	ب	
	س	بالا بر	Lift
Character	سرشت	پ	
	ص	پایا	Invariant
Faithful	صادقانه	ت	
	ض	تناظر	Correspondence
Inner product	ضرب داخلی	ث	
	ط	ثابت‌ساز	Stabilizer
Loop	طوقه	ج	
	ظ	جایگشت	Permutation
Valency	ظرفیت	چ	
	ع	چند جمله‌ای	Polynomial
Nonadjacency	عدم مجاورت	ح	
	ف	حاصل ضرب دکارتی	Cartesian product
Vector space	فضای برداری	خ	
	ک	خودریختی	Automorphism
Performance	کارایی	د	
	گ		

گراف.....Graph

م

ماتریس جایگشتی.....Permutation matrix

ن

ناهمبند.....Disconnected

و

وظیفه.....Task

ه

همبند.....Connected

ی

یال.....Edge

Abstract

This page is accurate translation from Persian abstract into English.

Key Words:

Write a 3 to 5 KeyWords is essential. Example: AUT, M.Sc., Ph. D, ..