مطالعات علوم کاربردی در مهندسی

دوره ۱۰، شماره ۱، بهار ۱۴۰۳

صفحات ۱۱۹-۱۰۲

Online ISSN: ۲۴۷۶-۴۵۰۷

Print ISSN: ۲۵۳۸-۲۰۵۵

www.irijournals.com

بکارگیری اصول وروشهای علم رایانه، مهندسی وتجزیه وتحلیلهای ریاضی برای طراحی، توسعه، آزمایش وارزیابی نرم افزار ها وسیستم های رایانه ای

شه بخش حوت

کارشناسی ارشد مدیریت فناوری اطلاعات

چکیده

مهندسی نرم‌افزار یعنی استفاده از اصول مهندسی بجا و مناسب برای تولید و ارائه محصول نرم‌افزاری با کیفیت که قابل اطمینان و با صرفه بوده و بر روی ماشین‌های واقعی به‌طور کارآمدی عمل کند. مهندسی نرم‌افزار یک روش سیستماتیک، منظم و دقیق برای ساخت و ارائه محصولی نرم‌افزاری با کیفیت است. مهندسی نرم‌افزار اغلب شامل فرایند خطی تحلیل، طراحی، پیاده‌سازی و آزمون است؛ که با به‌کارگیری روش‌های فنی و علمی از علوم مهندسی موجب تولید نرم‌افزاری با کیفیت مطلوب در طول یک فرایند انتخابی مناسب پروژه می‌شود. کاربردهای مهندسی نرم‌افزار دارای ارزش‌های اجتماعی و اقتصادی هستند، زیرا بهره‌وری مردم را بالا برده، چند و چون زندگی آنان را بهتر می‌کنند. مردم با بهره‌گیری از نرم‌افزار، توانایی انجام کارهایی را دارند که قبل از آن برای‌شان شدنی نبود. نمونه‌هایی از این دست نرم‌افزارها عبارت‌اند از: سامانه‌های توکار، نرم‌افزار اداری، بازی‌های رایانه‌ای و اینترنت. فناوری‌ها و خدمات مهندسی نرم‌افزار به کاربران برای بهبود بهره‌وری و کیفیت یاری می‌رساند. نمونه‌هایی از زمینه‌های بهبود: پایگاه داده‌ها، زبان‌ها، کتابخانه‌ها، الگوها، فرایندها و ابزار. برای چینش اجزاء مختلف سیستم نرم افزاری و نمایش روابط بین آنها و سایر موجودیتهای سیستم نرم افزاری. برای اینکه طراحی مدل برای سیستمهای نرم افزاری قالبی یکدست و یکپارچه و جهان شمول داشته باشد و تبادل اطلاعات بین مدلهای طراحی شده توسط افراد مختلف امکان پذیر باشد تلاشهای متعددی صورت گرفته است که UML یکی از آنهاست، که در حال حاضر متداولترین استاندارد تولید مدل برای سیستمهای نرم افزاری در سراسر دنیاست. UML مخفف Unified Modeling Language است. UML برای مدل سازی سیستمهای نرم افزاری و تسهیل طراحی شیء گرای سیستم ۹ دیاگرام (و استانداردهای مرتبط با هرکدام) را ارائه مینماید. قبل از توضیح بیشتر و ارائه تعاریف مقدماتی به نکته ذیل توجه کنید اغلب سوال میکنند که چرا UMLمهم است و این روزها مانور زیادی روی آن میشود؟ آیا لزومی دارد که به UML مسلط شویم؟ آیا اصولا" این جانور به درد ما میخورد در جواب باید گفت: تا حالا دیده اید که کسی یک ساختمان بزرگ با پیچیدگیهای مختلف را "بدون نقشه" و

مطالعات علوم کاربردی در مهندسی

دوره ۱۰، شماره ۱، بهار ۱۴۰۳

صفحات ۱۱۹-۱۰۲

Online ISSN: ۲۴۷۶-۴۵۰۷

Print ISSN: ۲۵۳۸-۲۰۵۵

www.irijournals.com

الگوی از پیش معین شده بسازد و این مقاله موفقیت آمیز باشد؟ آیا تا کنون شنیده اید که هیچ کدام از کارخانه های تلوزیون سازی بودن هیچ نقشه و پیش بینی فنی موفق به ساخت تلوزیونی شوند که کار کند؟ یا اصلا" ساخته شود؟ آیا تا کنون دیده اید کشوری بدون سیاستهای کلان و بدون سنجش جوانب امر، موفق به مدیریت امور داخلی خود شود؟ و ده ها سوال از این دست ! خواندن این سوالها بدون اینکه حتی ثانیه ای به جواب انها فکر کنید، خود، جواب به سوالات است. UMLبه عنوان استانداردی برای طراحی و پیش بینی جزئیات فنی سیستم نرم افزاری، نحوه ارتباط اجزاء، نوع و نحوه کارکرد قسمتهای مختلف و ... یکی از ملزومات تولید کنندگان نرم افزار در دنیای امروز است. حتی اگر مستقل کار میکنید و نرم افزارهای کوچک تولید میکنید با استفاده از UML در "اغلب" موارد به بالاترین حد بهینگی مراحل طراحی و تولید نرم افزارتون خواهید رسید و نکته آخر این که UML و استانداردهای آن و ابزارهای آن ها که آنقدر ساده و سهل هستند که صرف هزینه و وقت برای یادگیری و تسلط بر آنها نسبت به مزایائی که در قبال آن کسب خواهید کرد تقریبا غیر قابل توجه است. تولید نرم‌افزارهای کاربردی روزبه‌روز گسترش می‌یابد و لزوم به کارگیری روش‌ها و اصول مهندسی نرم‌افزار، در مراحل توسعه، مدیریت و پشتیبانی آن‌ها بیشتر نمود پیدا می‌کند. کیفیت نرم‌افزار (Software Quality)، شاخص حیاتی برای تولید نرم‌افزارهای با کیفیت بالا است که ضمن بالا بردن بهره‌وری در تولید نرم‌افزارها، به ایجاد نرم‌افزارهای قدرتمند و شکست‌ناپذیر منجر می‌گردد. مدل‌سازی نرم‌افزار، به کارگیری فنون پیشرفته آزمایش نرم‌افزار، مدیریت ریسک نرم‌افزار، تضمین کیفیت نرم‌افزار، مهندسی محصول و.... ، تنها عناوینی از فهرست گسترده زیرساخت‌های مرتبط با توسعه نرم‌افزارهای قوی و مهندسی ساز است. در این‌جا به طور خاص به بررسی علمی و فنی یکی از این زیر ساخت‌ها با عنوان کیفیت نرم‌افزار و روش‌های تعیین شاخص پرداخته شده است.

واژه‌های کلیدی: اصول، علم رایانه، مهندسی، تجزیه وتحلیل، ریاضی طراحی، توسعه، آزمایش، ارزیابی، نرم افزار، سیستم های رایانه ایی

دوره ۱۰، شماره ۱، بهار ۱۴۰۳، صفحات ۱۱۹-۱۰۲

مقدمه:

مهندسان نرم‌افزار طرفدار تکنولوژی‌ها و روش‌های عملی بسیار متفاوت و مختلفی هستند، که با هم ناسازگار هستند. این بحث در سال‌های دهه ۶۰ میلادی شروع شد و ممکن است برای همیشه ادامه پیدا کند. مهندسان نرم‌افزار از تکنولوژی‌ها و روش‌های عملی بسیار متنوعی استفاده می‌کنند. کسانی که کار عملی می‌کنند از تکنولوژی‌های متنوعی استفاده می‌کنند: کامپایلرها، منابع کد، پردازشگرهای متن. کسانی که کار عملی می‌کنند از روش‌های عملی بسیار متنوعی استفاده می‌کنند تا تلاش‌هایشان را اجرا و هماهنگ کنند: برنامه‌نویسی در دسته‌های دونفری، بازبینی کد، و جلسات روزانه. هدف هر مهندس نرم‌افزار بایستی رسیدن به ایده‌های جدید خارج از الگوهای طراحی شده قبلی باشد، که باید شفاف بوده و به‌خوبی مستند شده باشد. با وجود رشد فزاینده اقتصادی و قابلیت تولید فزاینده‌ای که توسط نرم‌افزار ایجاد شده، هنوز هم بحث و جدل‌های ماندگار درباره کیفیت نرم‌افزار ادامه دارند.

طراحی و پیاده سازی پایگاه داده‌ها: با توجه به آنچه که یک کارشناس مهندسی نرم افزار در طول تحصیل فرا می‌گیرد، یک کارشناس ارشد قدرت طراحی و پیاده سازی پایگاه های داده پیچیده تر و دقیق تر با حداقل خطا و همرنگ‌تر با سیستم های روز دنیا را دارا خواهد شد.

مهندسی نرم افزار: مهندسی نرم افزار موضوع بسیار پیچیده ای در مورد روش‌های مدیریت، طراحی، پیاده سازی و نگهداری برنامه‌ها و پروژه‌های نرم افزاری است و کارشناس ارشد، قدرت کنترل تمام موارد بالا را در برنامه های عظیم نرم افزاری خواهد آموخت.

سیستم های عامل: دانشجویان کارشناسی در طول تحصیل خود به طور تئوری و خلاصه با سیستم‌های عامل و زیر و بم های آنها آشنا می‌شوند. حال آنکه کارشناس ارشد مهندسی نرم افزار تا جایی پیش می‌رود که بتواند یک سیستم عامل مستقل و جدید و در عین حال شایسته را پیاده سازی کند.

اتوماسیون اداری: اتوماتیک کردن روند فعالیت‌های یک اداره موسسه یا سازمان به وسیله کامپیوتر را شامل می شود. برنامه نویسی همروند به وجود آوردن همزمان چند برنامه در کامپیوتر و مدیریت آنها، برنامه نویسی همروند است.

الگوریتم‌های موازی: شکستن الگوریتم‌ها به قسمت هایی که می توانند به صورت همزمان توسط چند پردازنده انجام شوند به منظور افزایش سرعت اجرای الگوریتم در بحث الگوریتم های موازی می گنجد.

طراحی و ساخت کامپایلرها: بر خلاف کارشناسان نرم افزار که نسبت به کامپایلرها شناخت پیدا می‌کنند، کارشناسان ارشد، قدرت طراحی و ساخت این قسمت مهم از دنیای تولید نرم افزار را خواهند آموخت.

شبیه سازی و گرافیک کامپیوتری: کارشناس ارشد مهندسی نرم افزار از آموزش‌های قدرتمندی جهت انجام شبیه سازی و گرافیک کامپیوتری بهره خواهد برد.

مهندسی نرم‌افزار، یک روش علمی، ریاضی و اقتصادی برای تولید نرم‌افزارها است که بر اساس آن، نرم‌افزار در طی یک فرایند علمی، تجزیه و تحلیل، طراحی، پیاده‌سازی، آزمایش و پشتیبانی می‌شود. به کارگیری مهندسی نرم‌افزار برای پیاده‌سازی نرم‌افزارهایی که اهداف مهم و حیاتی دارند، یک ضرورت است.

در مهندسی نرم‌افزار برای ساخت یک سیستم نرم‌افزاری، سه فرآیند مهم تأثیرگذار است:

۱. فرآیند توسعه :(Development Process)سازماندهی فعالیت‌ها برای ساخت یک سیستم است.

۲. فرآیند مدیریت :(Management Process) انتخاب افراد، تجهیزات و فرآیندها برای توسعه یک سیستم و کنترل و نظارت بر روند اجرای پروژه است.

۳. فرآیند پشتیبانی :(Maintenance Process) کنترل و پشتیبانی نرم‌افزار پس از تولید آن.

در فرآیند توسعه، هدف آن است که یک سیستم با مشخصاتِ خواسته‌شده تولید شود. فرآیند توسعه، از مرحله طرح یک راه حل مفهومی برای مسأله خواسته‌شده (مطالعه امکان‌سنجی) آغاز شده، پس از دریافت خواسته‌ها و بررسی سیستم، طراحی صورت گرفته و در نهایت، این طراحی با کمک ابزارهای پیاده‌سازی، به یک سیستم واقعی تبدیل می‌شود. هدف این فرآیند، آن است که از یک سو، برآورده ساختن نیازهای کاربران، و از سوی دیگر، کیفیت مناسب عملکرد سیستم تضمین گردد. بنابراین، باید مشتمل بر مکانیسم‌هایی برای اعتبارسنجی نرم‌افزار (خروجی مطابق با خواسته‌ها (Validation)) و وارسی‌پذیری نرم‌افزار (صحت عملکرد خروجی (Verification)) باشد.

بیان مسئله:

اساس ریاضی علم رایانه مدرن توسط کرت گدل با قضیه ناقصش در سال ۱۹۳۱ آغاز شده بود. در این قضیه او نشان داده که، جایی محدود بوده که داخل یک سیستم قراردادی چیزی را نتواند رد یا اثبات کند. این قضیه به کاری توسط Godel و بقیه راهنمایی کرده تا این سیستم‌های رسمی یا قراردادی را تشریح یا تعریف کنند. که این شامل مفاهیمی مانند توابع بازگشتی mu(μ) و توابع تعریف پذیر Lambda(λ) می‌باشد. سال ۱۹۳۶ یک سال کلیدی برای علم رایانه بوده‌است. آلن تیورینگ و آلونزو جرج، مستقلاً و همچنین با هم الگوریتم را به‌طور رسمی، با محدوده‌هایی که می‌توانست محاسبه کند و مدلی برای محاسبات مکانیکی خالص بود، معرفی کردند.

این عناوین توسط چیزی که اکنون قضیه جرج – تیورینگ نامیده می‌شود، تحت پوشش قرار می‌گیرد. که یک فرضیه در مورد طبیعت وسایل محاسباتی مکانیکی مانند رایانه‌های الکترونیکی، می‌باشد. این قضیه ادعا کرده که هر محاسباتی که ممکن است (حل شدنی است) می‌تواند توسط اجرای الگوریتم روی رایانه انجام شود. که با زمان کافی و فضای ذخیره‌سازی در دسترس فراهم می‌شود. همچنین تیورینگ، قضیه توصیف ماشین تیورینگ را شامل می‌شود. یک ماشین تیورینگ یک نوار طولانی نا محدود و یک هد یا سوزن نوشتن یا خواندن (R/W) دارد که همراه با نوار می‌تواند حرکت کند و تنها مقادیر مسیر را می‌تواند حرکت دهد. به‌طور شفاف همانند ماشینی است که هرگز نمی‌تواند بسازد، اما با این وجود مدلی است که می‌تواند محاسبات هر الگوریتمی را که روی هر رایانه مدرنی قابل اجراست، تقلید کند.

همچنین تیورینگ برای علم رایانه مهم است چونکه نام او نمایانگر جایزه تیورینگ و آزمایش تیورینگ است. او در جنگ جهانی دوم همکاری موفقیت‌آمیز و بزرگی با کد شکن‌های بریتانیایی داشته‌است و طراحی رایانه‌ها و نرم‌افزارها را در دهه ۱۹۴۰ کاملاً ادامه داده‌است.

در یک نشست خیلی بزرگ ماشین‌های دیجیتال در کمبریج، تورینگ گفت: «ما در تلاشیم تا ماشینی بسازیم تا همه نوع چیزهای متفاوت را به سادگی توسط برنامه نویسی سریع تر از راه اضافه کردن دستگاه‌های اضافی انجام دهیم. «اولین رایانه‌ای که قابل کنترل و برنامه‌نویسی با زبان‌های برنامه نویسی بود را کنراد زوسه (konrad Zuse) در سال ۱۹۴۱ در آلمان به نام Z3تکمیل و ارائه کرد. وی اولین رایانه خود را به نام Z1 در سال ۱۹۳۸ ساخته بود که البته قابلیت برنامه‌نویسی را نداشت.

در سال ۱۹۴۸ توسط بابی منچستر، رایانه دیگری که می‌توانست برنامه‌های ذخیره شده را اجرا کند، بر مبنای مدل ماشین تیورینگ ساخته شد.

بی گمان، نرم افزار یکی پیچیده ترین و در عین حال قابل انعطاف ترین دستاوردهای بشر می باشد. با وجودی که بیش از چند دهه از پیدایش نرم افزار نمی گذرد. این پدیده ی شگفت آور قرن بیستم، به عنوان یکی از مؤلفه های کلیدی فناوری های نوین اطلاعات و ارتباطات، تاثیر شگرفی بر کلیه ی جوانب زندگی بشر داشته است امروزه نرم افزار، سوخت لازم برای راه اندازی و به حرکت درآوردن موتورهای اقتصاد نوین تلقی می شود. هیچ سازمان و کسب وکار نوینی، نمی تواند بدون نرم افزار به حرکت و تکامل خود ادامه دهد. در طول چند دهه ی اخیر، با کمک رایانه ها و نرم افزارهای مختلف، حجم دانش بشری چندین برابر شده است. در آینده ی بسیار نزدیک، هر یک از ما شاهد بکارگیری نرم افزار در منزل، خودرو، تلویزیون، ساعت مچی، کتاب، و حتی لباس های خود خواهیم بود.

اما به واسطه ی تغییرات بسیار سریع و غافل گیرکننده ی فناوری های نوین اطلاعاتی و ارتباطی و به طور خاص نرم افزار، و به موازات آن، تغییر نیازها، خواسته ها، و انتظارات استفاده کنندگان از نرم افزار و قابلیت های آن، طراحی و تولید نرم افزار، بسیار پیچیده می باشد. عوامل دیگری مانند رقابت شدید، کمبود نیروی متخصص و حرفه ای، عدمِ دسترسی به دانش و تجربه ی موفق دیگران، لزوم تولید سریع، لزوم تولید مقرون به صرفه، نیاز روز افزون به همکاری میان رشته های مختلف، و مهم تر از همه، عدم استفاده ی مناسب از اصول و مبانی مهندسی در طراحی تولید نرم افزار، این صنعت را با چالش های بسیاری روبرو نموده است. حدود ۵۰ سال پیش، یعنی در اوایل پیدایش نرم افزار استفاده کنند گان این فراورده ی نوین همان طراحان و تولید کنندگان بودند. دران زمان نرم افزار عمدتا برای محاسبات و حل مسائل ریاضی استفاده می شد. وجود زبان های سطح پایین ۳ و محدودیت های سخت افزاری (کمبود حافظه و سرعت پردازش کم) از دیگر مشخصه های دوران اولیه ی پیدایش نرم افزار است در آن روزهای اولیه، نرم افزار چیزی جدا از سخت افزار نبود و حتی برای فروش سخت افزار، بطور رایگان در آن تعبیه می شد اما با گسترش دامنه ی کاربرد رایانه و به تبِ ع آن نرم افزار در زمینه های مختلف، به مرور شرایطی به وجود آمد که استفاده کنندگان و کاربران نرم افز صرفاً تولید نرم افزار بود. حالا دیگر نرم افزار قیمت داشت و اتفاقا برخلاف روند کاهش قیمت در سخت افزارها از طراحان و تولید کنندگان آن جدا شدند؛ سازمان ها و شرکت هایی به وجود آمدند که کارشان روز به روز بر قیمت نرم افزار افزوده می شد. نیازهای جدید استفاده کنندگان فراتر از محاسبات(رایانش) بود آنها به مدیریت اطلاعات نیاز داشتند. پیدایش زبان های سطح بالا و رفع محدودیت های سخت افزاری، از دیگر مشخصه های عصر جدید نرم افزار می باشد. درست در همین زمان است که اولین شکست ها و مشکلات نیز خود را نشان دادند. مشکلات و چالش ها به قدری جدی و پر هزینه بود که از آن به بحران نرم افزار یاد می شد.

سرانجام برای اولین بار، در سال ۱۹۶۸ و در یک کنفرانس که توسط ناتودر کشور آلمان برگزار شده بود، بر لزوم مهندسی این دستاورد جدید بشر، یعنی نرم افزار، تأکید شد. از آن زمان به بعد، تکنیک های مهندسی، ابزارها، و دانش و تجربه، صنعت نرم افزار به یکی از صنایع برتر جهانی تبدیل شده است.

اهمیت وضرورت:

سیستم‌های پشتیبان تصمیم گیری که در سال‌های اخیر مطرح شده است، بخشی از نظام اطلاعات مدیریتی را شکل می دهد. این سیستم‌ها با ماهیت روش های تحلیل، تعریف و حل مسئله در ارتباط است و بیشتر از روش ها و روال پژوهش عملیاتی

دوره ۱۰، شماره ۱، بهار ۱۴۰۳، صفحات ۱۱۹-۱۰۲

بهره می گیرد. شبیه سازی، برنامه نویسی ریاضی، روش‌های پیش بینی آینده و... بخشی از گنجینه‌ی سیستم‌های پشتیبان تصمیم گیری را شکل می دهد. سیستم پشتیبان ایده پردازی تقریباً مفهوم جدیدی است که یونگ به تفصیل آن را تشریح کرده است. سیستم های پشتیبان ایده پردازی با ترکیب ایده‌ها در "موقعیت‌های مسئله دار" و تعریف مسئله در گیر است. در این موقعیت ها رایانه به مثابه ابزار سازمان دهنده ی فکر و دست کاری کننده معانی و مفاهیم به کار گرفته می شود. ایده پردازی، فرایندهای ذهنی را در بر می گیرد و در تعریف و حل مسایل تصمیم گیری که به الگوریتم های ریاضی و انواع دیگری از روش های گام به گام نیاز ندارد، کاربرد پذیر است. از آن جا که از منطق در شکل گیری این نوع تفکر استفاده ی زیادی نمی شود، می توان آن را با شهود، برابر فرض کرد. آیزنبرگ گزارش داده است که اظهاریه‌های نمونه گیری از تفکر" چگونگی تفکر مدیران ارشد را نشان می دهد. اظهاریه ها تصریح می کنند که چگونه مدیران ارشد، مسایل مجزا از هم را حل می‌کنند و درشبکه ای از مسایل مرتبط به هم وارد می‌شوند. همچنین، اظهاریه های مزبور، چگونگی استفاده از آن چه که مدیران آن را "شهود سطح بالا" می‌خوانند و در آن الگوهای باز شناسی به سرعت شکل می گیرد و اطلاعات جدید به دست می آید، نشان می دهد، ایده پردازی از مبانی تصمیم گیری در مدیریت است. اما معمولاً اهمیت آن مورد مسامحه قرا رمی گیرد. الگوی مورد اشاره ی آیزنبرگ – ایده پردازی – موضوع پژوهش زیادی در قرن حاضر بوده است.

تاثیر شدید رایانه ها و فن آوری اطلاعات بر زندگی روزمره، نیاز به طراحی و توسعه سیستمهای نرم افزاری رایانه ای جدید و به کاربردن فن آوریهای جدید را در طیف وسیع و در حال رشدی از کاربردها الزامی ساخته است. وظایفی که به وسیله افرادی که با عنوان مهندس نرم افزار شناخته می شوند انجام میشود به سرعت در حال رشد ونمو هستند که منعکس کننده بخشهای تخصصی جدید و تغیرات فنآوری ونیز اولویتها و عملکرد کارفرمایان میباشند. مهندسان نرم افزار با بکارگیری اصول و روشهای علم رایانه، مهندسی و تجزیه و تحلیلهای ریاضی به طراحی، توسعه، آزمایش و ارزیابی نرم افزارها و سیستمهای رایانه یی می پردازند که به کمک آنها رایانه ها قادر به انجام بسیاری از عملکردهایشان می شوند. مهندسین نرم افزار شاغل در فرایند تولید سیستمها و برنامه های کاربردی نیازها ی کاربرها را تجزیه وتحلیل کرده و سیستمها و نرم افزارهای کاربردی را برای رایانه ها طراحی و تولید کرده ویا تغییر میدهند. آنها در طراحی و توسعه انواع نرم افزارها مشارکت دارند از جمله سیستمهای عامل، توزیع شبکه و کامپایلرها که برنامه ها را برای پردازش سریعتر تغییر می دهند. مهندسان نرم افزار با برنامه نویسی یا کد گذاری به رایانه برای انجام کاری خاص خط به خط دستورالعمل میدهند.

در ضمن مشکلات فنی ایجاد شده را نیز بر طرف می کنند. مهندسان نرم افزار باید در زمینه برنامه نویسی از مهارتهای بالایی برخوردار باشند اما اکثرا به جای نوشتن کد، با طراحی الگوریتمها، تجزیه و تحلیل و حل مشکلات برنامه نویسی سرو کار دارند. مهندسان نرم افزارهای کاربردی به تجزیه و تحلیل نیازهای کاربرها پرداخته و نرم افزارهای کار بردی عمومی ویا برنامه های ابزاری تخصصی را طراحی و بوجود می آورند ویا تغییر میدهند. با توجه به هدف برنامه، از زبانها ی برنامه نویسی متفاوتی استفاده می شود.

زبانهای بر نامه نویسی اصلی مورد استفاده شامل سی، سی پلاس پلاس و جاوا و برنامه هایی که کمتر کاربرد دارند شامل فرترن و کوبول می باشند. بعضی مهندسین نرم افزار هم به توسعه بسته های نرم افزاری سیستم و هم برنامه های کاربردی سفارشی میپردازند. مهندسین نرم افزارهای سیستم وظیفه هماهنگ کردن نصب ونگهداری سیستمهای رایانه ای یک شرکت را برعهده داشته و رشد و گسترش این سیستمها را در آینده برنامه ریزی می کنند. مهندسین شاغل در یک شرکت به هماهنگ کردن نیازها ی رایانه ای هربخش ازجمله دادن سفارشات، صورت برداری کالاها، صورت حساب ها و ثبت صورت پرداخت ها

می پردازند و از لحاظ فنی پیشنهاداتی ارائه می دهند. آنها در ضمن باید اینترانتهای شرکت، بعنی شبکه هایی که رایانه ها را درون یک شرکت به هم متصل کرده و ارتباطات درون شرکتی را راحت میکنند راه اندازی کنند.

مهندسین نرم افزارهای سیستم برای شرکتهایی که سیستمهای کامل رایانه ای را طراحی، اجرا و نصب میکنند نیز کار میکنند. احتمال دارد که انان در بخش بازاریابی و فروش کار کنند که در این بخشها به عنوان مرجع فنی اولیه برای فروشندگان و مشتریان به فعالیت می پردازنند. در ضمن فعالیت هایی چون فروش تولیدات و حمایت های فنی پس از فروش برای مشتریان نیز در حیطه کار آنان می باشد. مهندسان نرم افزاراغلب به عنوان بخشی از یک تیم که طراحی سخت افزار و نرم افزار و سیستمها را بر عهده دارد به فعالیت می پردازند. یک تیم هسته ای کارهای مهندسی، فروش، ساخت و فرماندهی کارکنان را تا ارائه محصول نهایی به عهده میگیرد.

تاریخچه:

سریع‌ترین ابزار شناخته شده در دوران باستان، برای محاسبه' چرتکه بوده‌است. چرتکه در حدود ۲۴۰۰ سال قبل در بابل اختراع شد. چرتکه‌های بابلی در اصل خطوطی روی خاک بودند و دانه‌های چرتکه، دانه‌های شن. چرتکه اولین رایانهٔ شناخته شده و پیشرفته‌ترین سیستم محاسباتی شناخته شده تا ۲۰۰۰ سال پیش است.

در ۱۱۱۵ سال پیش از دوران فعلی، در چین باستان ارّابهٔ بندکشی جنوبی اختراع شده بود. آن اختراع اولین سازوکار چرخ دنده شناخته شده بود که در چرخ‌های گوناگون مورد استفاده قرار می‌گرفته‌است و بعداً در رایانه‌های قیاسی یا آنالوگ استفاده شدند. به‌علاوه، در حدود دو قرن پیش از دوران فعلی چینی‌ها چرتکه ماهرتری را اختراع کردند. در سدهٔ پنجم پیش از دوران فعلی در هند باستان، پانینی متخصص گرامر، دستور زبان سانسکریت را در ۳۹۵۹ قانون شناخته شده قاعده دار و تدوین کرده‌است. مثل آشتادهایی (Ashtadhyayi)که سیار فنی اسلوب داده شده بود. پانینی از قوانین تغییر شکل و بازگشتی‌ها استفاده کرده بود و با این قبیل مهارتی که گرامرش داشت، معادل قدرت محاسبه ماشین تیورینگ بود. بین ۲۰۰ تا ۴۰۰ سال پیش از این، جاینا ریاضی دان هندی، لگاریتم را اختراع کرد. از قرن سیزدهم جدول لگاریتم توسط ریاضی دانان مسلمان ابداع شد.

باور بر این است که ماشین آنتیکیترا قدیمی‌ترین رایانه قیاسی شناخته شده‌است. آن ماشین برای اینکه موقعیت‌های نجومی را محاسبه کند طراحی شده بود. آن ماشین در سال ۱۹۰۱ در لاشه کشتی آنتی کیترا در جزیره یونانی آنتی کیترا بین کیترا و کرت حدود ۱۰۰ سال قبل کشف شده بود. مکانکی رایانه آنالوگ دو باره در یک هزار سال اخیر در جهان اسلام و در قرون وسطی ظاهر شده و توسط ستاره شناسان مسلمان توسعه داده شده بود. مانند خط استوایی توسط آرزاشل، چرخ دنده‌های مکانیکی توسط ابوریحان بیرونی و نیروی گشتاور توسط جابر ابن افلح. اولین ماشین‌های قابل برنامه‌ریزی هم توسط مهندسان مسلمان اختراع شده بود. مانند نوازنده فلوت اتوماتیک توسط برادران بنو موسی (Banu Musa) و روباتهای شبه انسان توسط الجازاری. همچنین ریاضی دانان مسلمان مهم‌ترین پیشرفت‌ها را در رمز نگاری کرده بودند مانند پیشرفت در کشف نوشته رمزی و آنالیز فرکانس توسط آلکیندوس.

زمانی که جان نپیر در اوایل قرن هفدهم لگاریتم را برای اهداف محاسباتی کشف کرده بود به دنبال آن یک دوره پیشرفت قابل توجه توسط مخترعان و دانشمندان در ساخت ابزار محاسباتی شروع شد. حدود سال ۱۶۴۰، بلیز پاسکال یک ریاضی‌دان اهل فرانسه، اولین وسیله جمع مکانیکی را مبنی بر طراحی توصیف شده توسط قهرمان آلکساندر، ریاضی دان یونان،

ساخته‌است.

در ابتدا هیچ‌کدام از وسایل محاسباتی واقعاً رایانه در مفهوم پیشرفته نبوده‌اند. آن پیشرفت قابل توجه در ریاضیات و تئوری قبل از اولین رایانه مدرن طراحی شده، گرفته شده‌است.

در قرن هفتم، ریاضی‌دان هندی، براهماگوپتا اولین بار سیستم اعداد هندی-عربی را توضیح داد و از عدد صفر استفاده کرد. در حدود سال ۸۲۵ میلادی ریاضی‌دان ایرانی، خوارزمی کتابی تحت عنوان «محاسبات با اعداد هندی» نوشت. کتابی که بعدها باعث انتشار سیستم اعداد هندی در خاورمیانه و اروپا گردید. کتاب او در حدود قرن دوازدهم میلادی تحت عنوان(Algoritmi De (Numero Indorum به لاتین ترجمه شد که البته علاوه بر مفهوم اعداد هندی شامل کاربردها و استفاده‌های محاسباتی قاعده دار در عمل جمع نیز می‌شد. نام الگوریتم در زبان‌های اروپایی از نام خوارزمی، دانشمند ایرانی مشتق شده‌است.

قبل از دهه ۱۹۲۰ رایانه‌ها کارمندان از نوع انسان بوده‌اند که محاسبات را انجام می‌دادند. آن‌ها معمولاً تحت هدایت یک فیزیکدان بوده‌اند. هزاران رایانه در تجارت، دولت و تشکیل یک تحقیق کار می‌کردند یا استخدام شده بودند. بیشتر این رایانه‌های خانم‌ها بوده‌اند و آن‌ها مدرکی در حسابان داشته‌اند. تعدای از آن‌ها برای سالنامه‌ها محاسبات نجومی را انجام می‌دادند. بعد از دههٔ ۱۹۲۰، بیان ماشین محاسبات، به ماشینی که کار یک رایانه انسانی را انجام می‌داده اطلاق می‌شده‌است. مخصوصاً آن‌ها با روش‌های مؤثر قضیه جرج – تیورینگ مطابقت داشته‌اند. این قضیه که روشی است برای یک ریاضی دان، اگر توانایی تنظیم یک لیست دستورالعمل‌ها را به وسیلهٔ یک کارمند انسانی با مداد و کاغذ داشته باشد، مؤثر است. به دلیل ضرورت خیلی زیاد و بدون قوه ابتکار و نبوغ. ماشینهایی که مقادیر متوالی را محاسبه می‌کردند به عنوان نوع آنالوگ یا قیاسی شناخته شده‌اند. آن‌ها از دستگاهی که کمیت عددی متوالی را نشان بدهد استفاده می‌کردند. مانند زاویه چرخش یک گلوله یا اختلاف در پتانسیل الکتریکی.

ماشین‌های دیجیتال در تقابل با آنالوگ 'توانایی تحویل وضعیت یک مقدار عددی و ذخیره هر تک رقم را داشتند. ماشین دیجیتال قبل از اختراع قطعات سریعتر حافظه، در ماشین‌های مختلف یا دستگاه‌های تقویت نیرو استفاده می‌شده‌است.

عبارت «Computing Machine» (ماشین محاسبه) به تدریج بعد از دهه ۱۹۴۰ از بین رفته‌است. درست بودن عبارت رایانه از زمانی که ماشینهای دیجیتال الکترونیک رایج شده‌اند شروع شده‌است. این رایانه‌های توانایی انجام محاسبات را که قبلاً توسط کارمندان انسان انجام می‌شده‌است را داشته‌اند. چون مقادیر ذخیره شده توسط ماشینهای دیجیتال مانند وسایل آنالوگ به خصوصیات فیزیکی محدود نبود، یک رایانه منطقی مبتنی بر تجهیزات دیجیتال، توانایی انجام هر چیزی را که می‌توانست ماشین خالص را تولید کند، داشته‌است. آلن تیورینگ که به عنوان پدر علم رایانه شناخته می‌شده‌است، این قبیل رایانه‌های منطقی را که به عنوان ماشین تیورینگ شناخته می‌شود را اختراع کرده بود. که بعد از آن داخل رایانه‌های مدرن و پیشرفته رشد پیدا کرد. به علاوه این رایانه‌های جدید توانایی انجام محاسبات غیر عددی را مثل موسیقی داشته‌اند. از زمانی که پردازش‌های محاسباتی توسط کارمندان انسانی انجام می‌شده‌است، این مطالعهٔ توانایی انجام محاسبات، به وسیله آشکار ساختن چیزی که در مفهوم متداول آشکار نبوده‌است، یک علم را آغاز کرده‌است.

بحث وتحلیل:

فرصت های شغلی برای مهندسان نرم افزاری از رشد سریعی بر خوردار است. احتمال می رود که رشد سریع در صنایع رایانه ای و خدمات پردازش اطلاعات که بالا ترین شمار استخدام مهندسین نرم افزار را داراست منجربه ایجاد فرصت های شغلی مناسبی برای فارغ التحصیلان این رشته باشد. کار فرمایان به دنبال استخدام مهندسان زبده ای هستند که در زمینه برنامه نویسی، تجزیه و تحلیل سیستمها روابط عمومی و تجارت دارای مهارت های بالایی باشند. از آنجایی که مشاغل تجاری و دیگر سازمان ها روز به روزبا فنآوری جدید وفق میابند و به دنبال افزایش کارایی سیستمهای رایانه ای خود می باشند، انتظار می رود که استخدام مهندسان نرم افزار در همه زمینه ها روند صعودی یابد. رقابت میان مشاغل تجاری عاملی در جهت ادامه نوآوری های پیشرفته فنآوری میشود و سازمانها برای اجرای تغییرات لازم در این مسیر به همکاری مهندسان نرم افزار بیشتری نیاز پیدا خواهند کرد. نیاز به مهندس نرم افزار با رشد شبکه های رایانه ای افزایش می یابد به عنوان مثال نتیجه گسترش فنآوری اینترنت و رشد سریع تجارت الکترونیکی و داد و ستد اینترنتی استخدام بیشتر مهندسان نرم افزار است که توانایی راه اندازی اینترنت، و سایر امور شبکه را داشته باشند. همچنین گسترش سیستمهای پردازش اطلاعات در مشاغل تجاری، مخابرات، مراکز دولتی و سایر موارد روز به روز شکل پیچیده تری می یابد وبه منظور راه اندازی، حفظ امنیت و ارتقای سیستمها و حل مشکلات به گروهی از مهندسان سیستمهای نرم افزاری نیاز می باشد. همزمان با این تحولات فرصت های مشاوره نرم افزاری نیز برای مهندسان فراهم شود.

کیفیت در مفهوم عام آن، به معنای خصوصیت یا صفتی از یک شیء است. در مورد یک شیء، کیفیت، به خصوصیاتی از قبیل: رنگ، شکل، اندازه و... اشاره دارد و در مورد یک نرم‌افزار شامل: درجه پیچیدگی درونی الگوریتم‌های آن، تعداد خطوط برنامه نرم‌افزاری، ارتباطات داخلی زیر برنامه‌ها و... می‌شود.

کیفیت بالای محصول نرم‌افزاری، به صرفه‌جویی در هزینه و ارتقای همیشگی سطح نرم‌افزار می‌انجامد. تمامی توسعه‌دهندگان نرم‌افزاری توافق دارند که دستیابی به نرم‌افزارهای با کیفیت، بالاترین هدف در ایجاد و ساخت سیستم‌های نرم‌افزاری است؛ اما کیفیت نرم‌افزار چگونه تعریف می‌شود؟

کیفیت نرم‌افزار مطابق با نیازهای عملیاتی و استانداردهای توسعه نرم‌افزار تعریف و تدوین می‌گردد و در این میان، توجه به سه اصل زیر اهمیت دارد:

۱. استانداردها، مجموعه‌ای از معیارهای توسعه را تعریف می‌کنند و چنانچه این معیارها به درستی دنبال نشوند، نتیجه آن فقدان کیفیت خواهد بود.

۲. چنانچه یک نرم‌افزار بر نیازهای اصلی خود منطبق باشد، اما نیازهای جانبی خود مانند سهولت کاربری و پشتیبانی مناسب را برآورده نسازد، کیفینت نرم‌افزار حاصل نگردیده است.

۳. نیازمندی‌های نرم‌افزار و آنچه نرم‌افزار برای آن طراحی و پیاده‌سازی گردیده، مبنای اندازه‌گیری کیفیت است. عدم تطبق نرم‌افزار با نیازمندی‌های آن، موجب عدم کیفیت نرم‌افزار خواهد شد.

۴. ISO 9001، یک استاندارد تضمین کیفیت است که در مهندسی نرم‌افزار نیز کاربرد دارد. این استاندارد، دربردارنده نیازمندی‌های لازم برای تضمین کیفیت یک سیستم نرم‌افزاری است. کنترل طراحی، کنترل مستندات، شناسایی محصول نرم‌افزاری، کیفیت، امکان‌سنجی و نیازمندی‌های وظیفه‌ای و غیر وظیفه‌ای، بازبینی و آزمایش نرم‌افزار و بررسی کیفیت‌های داخلی، از عناوینی است که در این استاندارد مورد توجه قرار می‌گیرد.

این بخش، کلیات چرخه حیات نرم‌افزار را تشریح می‌کند. در این استانداردها غالباً از عبارت «پروژه نرم‌افزاری» استفاده می‌شود. واضح است که تولید نرم‌افزار، جنبه‌های سخت‌افزاری رایانه را نیز در بر می‌گیرد. محصولات نرم‌افزاری لازم است با روش معینی طراحی و اجرا گردند. یک مدل چرخه حیات، فعالیت‌های پروژه را در قالب مراحل مشخص سازماندهی می‌کند و تعیین می‌نماید کدام یک از فعالیت‌ها باید در کدام مرحله انجام گیرد. بر اساس این استاندارد، باید شش مرحله در چرخه حیات یک نرم‌افزار طی شود که عبارت‌اند از:

مرحله:UR تعیین نیازهای کاربر(User Requirements) ؛

مرحله :SR تعیین نیازهای نرم‌افزار(Software Requirements) ؛

مرحله :AD طراحی معماری(Architectural Design) ؛

مرحله:DD طراحی تفصیلی و تولید برنامه(Detailed Design) ؛

مرحله :TR انتقال و واگذاری نرم‌افزار برای بهره‌برداری(Transfer of the Software) ؛

مرحله:OM بهره‌برداری و نگهداری(Operation & Maintenance) ؛

اینک مدل‌های کیفیت نرم‌افزاری که تاکنون ارائه شده است را مورد بررسی قرار می‌دهیم و نقاط قوت و ضعف آن‌ها را با یکدیگر مقایسه می‌کنیم. ارتباط بین اجزای سطوح یک مدل کیفیت می‌تواند یک به چند یا چند به چند باشد. در ارتباط یک به چند، هر فاکتور یا خصوصیت کیفی، فقط با ویژگی‌های فرعی خود در سطح پایین‌تر مرتبط است؛ ولی در ارتباط چند به چند، هر فاکتور می‌تواند با ویژگی‌های فرعی دیگر نیز مرتبط باشد. کیفیت نرم‌افزار، از ارزیابی این ویژگی‌های فرعی توسط معیارهای اندازه‌گیری حاصل می‌شود.

به مجموعه مفاهیم، ایده ها و نمادها که به شکل های گوناگون نیاز کاربر را در موضوعات مختلف برآورده می کند نرم افزار گفته میشود که قابل لمس نیز نمی باشد. امروزه برای رفع نیاز مخاطبان و سرعت عمل بخشیدن به فعالیت های گوناگون که بزرگترین آن تبادل اطلاعات میان کاربران کل کره زمین است از نرم افزارها استفاده میشود. این تکنولوژی که به مرور زمان بسیار پیشرفت کرده و کار همگان را نیز آسان تر کرده بسیار تخصصی است و نیاز به علم و دانش در این زمینه دارد. مهندسان نرم افزار با نوشتن برنامه های مختلف برای مخاطبان مختلف خود کمک قابل توجهی کرده اند، حالا در اینجا لغت سخت افزار مطرح میشود که وابسته به اجزای تشکیل دهنده رایانه و قطعات سخت افزاری است.

نتیجه گیری:

ارزیابی، فرایندی است که به جمع‌آوری اطلاعات درباره ارزش یک برنامه با هدف تصمیم‌گیری درباره اثربخشی یا بهبود آن برنامه صورت می‌گیرد. به بیانی دیگر، ارزیابی، «اندازه‌گیری عملکرد یا خدمات یک نظام و بررسی اثر آن در پیشبرد اهداف تعیین‌شده» است. ارزیابی می‌تواند به عنوان وظیفه‌ای در نظر گرفته شود که به یک یا چند خروجی می‌انجامد و در عین حال، هدفی برای برنامه‌ریزی به شمار می‌آید. افزون بر این، ویژگی دیگر ارزیابی، «هدف‌محور» بودنِ آن است. ارزیابی در زمینه گسترش نرم‌افزارها، از دهه‌های اخیر که نقش ارزیابی در تعامل انسان و رایانه تغییر یافته، به عنوان یک هدف کاربرد داشته است. تولید روزافزون محصولات نرم‌افزاری با هدف برطرف کردن نیازهای جدید کاربران، و در نتیجه نیاز به ارتقا و تکمیل محصولات نرم‌افزاری، اهمیت ارزیابی نرم‌افزارها را دوچندان نموده است. نرم‌افزارها را می‌توان از دیدگاه‌های متفاوتی مانند:

کارکرد، قابلیت اطمینان، کاربردپذیری، بازدهی، قابلیت نگه‌داری و یا حمل‌پذیری بررسی کرد. فرایند ارزیابی نرم‌افزار، در عمل می‌تواند دست کم یکی از این سه هدف را دنبال کند: مقایسه نرم‌افزارها، تعیین میزان کیفیت و یا تعیین نقاط ضعف آن‌ها.

به دنبال تولید روزافزون انواع محصولات نرم‌افزاری، در زمینه نرم‌افزارهای اسلامی نیز در سال‌های اخیر تلاش‌هایی در شهرها و سازمان‌های مختلف کشور ما ایران، صورت گرفته است و در مدت نه‌چندان زیادی که از تولید این نرم‌افزارها گذشته است، شاهد پیشرفت‌های قابل توجهی بوده‌ایم. تولید نرم‌افزارهای اسلامی در زمینه: علوم قرآنی، علوم حدیث، فقه، رجال، سیره معصومین(ع)، احکام و... ، و کاربرد آن در بین کاربران رسانه‌ها و محصولات دیجیتال، مسلماً نقش به‌سزایی در اشاعه و گسترش علوم اسلامی در میان کاربران دنیای دیجیتال خواهد داشت. به همین منوال، تلاش برای بهبود و ارتقای این نرم‌فزارها، می‌تواند در جذب مخاطب و رضایت آن‌ها از این محصولات بسیار مؤثر باشد. در سال‌های اخیر، پژوهش‌هایی در زمینه ارزیابی نرم‌افزارهای اسلامی انجام شده است. مقاله حاضر، علاوه بر گردآوری متون دانشگاهی و مقاله‌های منتشرشده در زمینه ارزیابی نرم‌افزارهای مختلف در کشور و همچنین مرور تعدادی از مقاله‌های مهم انگلیسی‌زبان در زمینه ارزیابی نرم‌افزار، در تلاش است پشینه‌های انجام ارزیابی در زمینه نرم‌افزارهای اسلامی را به صورت منسجم جمع‌آوری نماید.

نرم‌افزارها را از دیدگاه‌های گوناگون می‌توان مورد ارزیابی قرار داد. نوع، کارکرد و هدف خود نرم‌افزار هم می‌تواند تعیین‌کننده نوع ارزیابی باشد. به اعتقاد عده‌ای از صاحب‌نظران، ارزیابی نرم‌افزار، فرایندی هدف‌محور است. مراحل ارزیابی، زمان، هزینه و تمامی فرایند ارزیابی، بستگی به هدفی دارد که به واسطه آن انجام ارزیابی صورت می‌گیرد. ارزیابی نرم‌افزارها می‌تواند با اهداف متفاوتی صورت بپذیرد؛ برای مثال، نرم‌افزاری را در نظر بگیرید که با هدف نیازسنجی کاربران برای گنجاندن یک سرویس اضافی در سرویس‌های خود، بررسی عملی این امکان را به همراه ارزیابی نظرات کاربران در این مورد آغاز می‌کند. در مقابل، نرم‌افزاری برای ارزیابی نظرات کاربران درباره دسته‌ای از خدمات خود به کار گرفته می‌شود تا در صورت نیاز، ارتقای خدمات خود را در تولید نرم‌افزار لحاظ نماید. مطمئناً اهداف ارزیابی در این دو نرم‌افزار، یکسان نیست و مراحل، زمان و هزینه‌های متفاوتی خواهند داشت. در مورد هدف ارزیابی نرم‌افزارها حتی می‌توان مواردی را بیان کرد که ارزیابی نرم‌افزار به منظور مقایسه و انتخاب نرم‌افزارها برای انطباق با محیط و عملکردهای خاصی که مد نظر یک فرد یا یک سازمان است، انجام می‌گیرد. با وجود گزینه‌های مختلف نرم‌افزارهایی که برای رفع نیازهای فرد یا سازمان وجود دارد، ارزیابی نرم‌افزارها قبل از انتخاب و خرید و بررسی مطابقت آن‌ها با نیازهای سازمان، اهمیت قابل توجهی دارد و موجب صرفه‌جویی در زمان و هزینه‌های فردی و سازمانی می‌شود. رولی (۱۹۹۲) معتقد است باید به چنین فرایندهایی که با هدف بررسی و مقایسه نرم‌افزارها به منظور انتخاب و خرید آن‌ها انجام می‌شود، از منظر یک پروژه نگاه کرد. برای این کار، باید بعد از مشخص ساختن اهداف، روش‌شناسی مناسب را اتخاذ نمود و در مرحله بعد، معیارهای متناسب را استخراج نمود و فرایند بررسی را انجام داد. در زمینه مقایسه نرم‌افزارهای جایگزین، فر، مودیگوندا و الامی (۲۰۰۹) نیز روشی آماری برای مقایسه ارائه داده‌اند.

همچنین مهندسان نرم‌افزار در جریان تولید و طراحی برنامه‌های رایانه‌ای، ممکن است با مشکلات و مسائلی اخلاقی روبه‌رو شوند و برای عبور از این مسائل، نیازمند رهنمودهایی اخلاقی و معین باشند. با توجه به اصول اخلاقی برگرفته از قرآن و حدیث، حمید، خطیب و موتاز (۲۰۱۰) سیاهه‌ای اخلاقی تهیه نموده‌اند و پیشنهادی ارائه کرده‌اند که مهندسان نرم‌افزار در روند طراحی نرم‌افزار، مسائل اخلاقی را با استفاده از آموزه‌های دین اسلام برطرف سازند.

از سوی دیگر، سازمان‌هایی که به تولید و طراحی نرم‌افزارهای مختلف می‌پردازند، نیاز دارند که مطالعه‌ای روی فناوری‌های مورد استفاده خود در تولید نرم‌افزارها انجام دهند؛ برای مثال، نرم‌افزارهای اسلامی که در کشور ما تولید می‌شوند، همگی با در

دوره ۱۰، شماره ۱، بهار ۱۴۰۳، صفحات ۱۱۹-۱۰۲

نظر گرفتن استفاده حداکثری کاربران ایرانی از محیط ویندوز، تأکید بیشتری بر محصولات مبتنی‌بر ویندوز دارند؛ تا سیستم‌های عامل دیگر. در مورد ابزارها و فناوری‌های داخلی هر کدام از این نرم‌افزارها هم، مطالعه‌ای تجربی نیاز است؛ برای مثال، استفاده از نرم‌افزار فلش برای اجرای بعضی پایگاه‌ها و نرم‌افزارها ضروری است که مسئولان آن پایگاه اینترنتی یا نرم‌افزار باید امکان استفاده و دسترسی کاربران به نرم‌افزار فلش را در هنگام انتشار محصول خود در نظر داشته باشند. در نظر نگرفتن ویژگی‌هایی این‌چنینی، باعث بعضی خسارت‌ها و عدم موفقیت‌ها است؛ برای مثال، نسخه جستجوی منابع کتابخانه ملی روی گوشی‌های تلفن همراه، در شرایطی مبتنی بر سیستم عامل سیمبین طراحی شده است که کاربران سیستم سیمبین در این روزها به‌شدت رو به کاهش هستند و روزبه‌روز به تعداد کاربران سیستم اندروید افزوده می‌شود. به این ترتیب، نوع دیگری از ارزیابی نرم‌افزارها که می‌تواند به عنوان پایه‌ای‌ترین و راهبردی‌ترین نوع ارزیابی‌ها در نظر گرفته شود، «ارزیابی فناوری نرم‌افزار » است. در این نوع ارزیابی، فناوری خاصی که نرم‌افزار با استفاده از آن تهیه و استفاده می‌شوند، مورد تحلیل و ارزیابی قرار می‌گیرد. به این ترتیب، اگر نیازمندی و تطابق خاصی مورد نظر باشد، در طراحی نرم‌افزار اعمال می‌شود. لانوبیل (۱۹۹۷) در مقاله‌ای خلاصه مباحث مطرح‌شده در کارگاه بین‌المللی مطالعات غربی نگهداری نرم‌افزار (WESS96) را در مورد ارزیابی فناوری برای تولید و نگهداری نرم‌افزارها ارائه می‌دهد. ارزیابی فناوری، به بررسی قابلیت‌ها و ظرفیت‌های فناوری‌های جایگزین برای تولید و طراحی نرم‌افزار اشاره دارد.

از لحاظ استمرار فعالیت‌های ارزیابی نرم‌افزار، به طور کلی، می‌توان دو دسته ارزیابی را تشخیص داد:

۱. ارزیابی‌های مستمر؛
۲. ارزیابی مقطعی.

با توجه به محیط متغیر، فناوری‌های رو به گسترش و تجربه‌های دهه‌های اخیر، بسیاری از پروژه‌های نرم‌افزاری به طور خودکار، نوعی از ارزیابی مستمر را در فرایند تولید نرم‌افزارهای خود جای داده‌اند. هدف این پروژه‌ها، جلب رضایت مشتری و حفظ آن‌ها در طول زمان است. در این پروژه‌ها، دریافت بازخورد از سوی کاربران و مشتریان در کنار پژوهش و بررسی برای به‌کارگیری فناوری‌های جدید و منطبق‌ساختن نرم‌افزار با برترین فناوری‌های روز، به صورت جزئی از روند تولید و گسترش نرم‌افزار در نظر گرفته شده است. در مقابل، تعدادی از نرم‌افزارها، روش‌های ارزیابی مقطعی را در مورد: سرویسی خاص، امکان به‌کارگیری نوع خاصی از فناوری و... انتخاب می‌کنند. انتخاب روش‌های ارزیابی مقطعی می‌تواند نتیجه: به‌کارگیری سیاستی خاص، کمبود بودجه برای انجام ارزیابی مستمر و یا دلایل دیگر باشد. شاید بتوان گفت نوع کامل ارزیابی نرم‌افزارها، به‌کارگیری یک فرایند ارزیابی مستمر و در کنار آن، بهره‌گیری از ارزیابی‌های مقطعی است. آکونا، دی‌آنتونیو، فره، لوپز و ماته (۲۰۰۰) به روش‌های ارزیابی فرایند نرم‌افزاری اشاره و خاطرنشان می‌کنند که عبارت «فرایند نرم‌افزاری» در مقابل عبارت «چرخه حیات)» به‌کار برده شده است. تفاوت عمده این دو عبارت را می‌توان این‌گونه بیان کرد که چرخه حیات، حالت‌های متفاوتی را که از ابتدای تولید محصول تا پایان عمر آن طی می‌شود، در نظر دارد و متمرکز بر محصول است؛ اما فرایند نرم‌افزاری، بر ساختار فعالیت‌هایی دلالت دارد که در طراحی نرم‌افزار، ارزیابی و بهبود آن دنبال می‌شوند. با این توصیف، شاید بتوان گفت ارزیابی مستمر را بیشتر در حیطه فرایند نرم‌افزاری و ارزیابی مقطعی بیشتر در حوزه چرخه حیات مطرح می‌شوند.

اما در متون ارزیابی نرم‌افزار، آثاری مشاهده می‌شود که کوشیده‌اند دسته‌بندی از ارزیابی نرم‌افزارها و همچنین روش‌هایی برای انجام آن ارائه دهند. کیچنهام و همکاران (۱۹۹۶) در پروژه دسمت (DESMET) گزارشی را منتشر کردند و هدف پروژه دسمت را «گسترش و تأیید اعتبار روشی برای ارزیابی روش‌ها و ابزارهای مهندسی نرم‌افزار» بیان نمودند. در این متن،

دوره ۱۰، شماره ۱، بهار ۱۴۰۳، صفحات ۱۱۹-۱۰۲

راهبردهای پروژه دسمت برای: ۱. انتخاب یک روش ارزیابی مناسب؛ ۲. انجام تحلیل ویژگی‌نما و ۳. انجام مطالعه موردی کمّی شرح داده شده است. به این ترتیب، پروژه دسمت، چارچوبی را ترسیم می‌کند که بر اساس آن، برای تعیین روش ارزیابی روند توسعه یک نرم‌افزار یا ابزار، مراحلی مشخص باید طی شود تا بهترین روش ارزیابی انتخاب شود. در این چارچوب، مؤلفه‌های مختلفی مانند بافت، هدف ارزیابی (روش کلی، روش اختصاصی توسعه یک نرم‌افزار، ابزار)، انواع ارزیابی (کمّی یا کیفی) و شیوه ارزیابی (تجربیات رسمی، مطالعه موردی، پیمایش در نظر گرفته می‌شود تا در ابتدا برای هر ارزیابی روش و راهبرد متناسب تعیین شود. بر این اساس، مطالعات ارزیابانه، تجربه‌ای رسمی از ارزیابی نرم‌افزارها هستند، یا مطالعه‌ای موردی بر روی یک نرم‌افزار خاص، و یا پیمایش نظرات متخصصان یا کاربران در مورد نرم‌افزار با هدف عیب‌یابی یا بهبود و ارتقا. به اعتقاد کیچنهام، پیکارد و فلیجر (۱۹۹۵) اگر کار ارزیابی در حد یک پروژه توسط گروهی محدود باشد، می‌توان از مطالعه موردی استفاده کرد. اگر چندین پروژه مشابه یا یک پروژه تکرارپذیر توسط یک گروه مورد ارزیابی قرار بگیرد، می‌توانیم از تجربه رسمی استفاده کنیم و در صورتی که گستردگی کار از لحاظ تعداد و تیم‌های پژوهشی زیاد باشد، می‌توانیم با پیمایش نظرات کاربران و کارشناسان به ارزیابی روش‌ها یا ابزارها اقدام کنیم. از آن‌جا که بر خلاف تجربیات رسمی و پیمایش‌ها، مطالعات موردی از پایه نظری مستحکمی برخوردار نیستند و قابلیت تعمیم آن‌ها کمتر است، کیچنهام و همکاران (۱۹۹۵) در اثر خود، به‌تفصیل در مورد شیوه انجام مطالعه موردی بحث می‌کنند؛ به طوری که بتوان به عنوان روشی استاندارد برای انجام مطالعات موردی و موثق‌تر ساختن این نوع پژوهش‌ها، از آن بهره برد.

آنترکالمشتاین و چنگ (۲۰۱۲) در مقاله‌ای مروری، راهبردهای بهبود فرایندی نرم‌افزاری (Software Process Improvement = SPI)را در متون بین سال‌های ۱۹۹۱ تا ۲۰۰۸ م بررسی نموده‌اند و متوجه شده‌اند که متون از لحاظ ابزار مطالعه، به سه دسته تقسیم می‌شوند: دسته اول، با نام چهارچوب (Framework) مربوط به راهبردهایی مانند CMM یا شش سیگما می‌شود که استراتژی‌های چهارچوب‌دار و مدل‌دار را شامل می‌شود. دسته دوم، تجربه‌های عملی است که قابلیت کاربرد برای نرم‌افزارهای دیگر را نیز دارا هستند و دسته سوم، ابزارهایی هستند که به مهندسی نرم‌افزار کمک می‌کنند. همچنین چندین راهبرد مطالعاتی متمایز را تشخیص داده‌اند:

۱. مقایسه پیشین و پسین؛
۲. تحلیل آماری؛
۳. مقایسه پیشین و پسین به همراه پیمایش؛
۴. کنترل فرایند آماری؛
۵. تحلیل هزینه ـ سودمندی؛
۶. تحلیل آماری به همراه پیمایش؛
۷. روش فیلیپ کراسبی؛
۸. مقایسه پیشین و پسین به همراه تحلیل هزینه ـ سودمندی؛
۹. پیمایش؛
۱۰. روش تحلیل بهره‌وری نرم‌افزار؛
۱۱. تحلیل هزینه ـ سودمندی به همراه پیمایش.

شاخصه‌های موفقیت نرم‌افزار دسته‌بندی و معرفی شده‌اند: کیفیت پیشرفت، دقت تخمین، بهره‌وری، کیفیت محصول، تلاش تیمی، رفع کاستی‌ها، هزینه، زمان رسیدن به بازار، بازگشت سرمایه، رضایت کاربر (کیفی)، رضایت کاربر (کمی) و دیگر معیارهای کمی و کیفی.

علاوه بر ملاک‌هایی که درباره انواع ارزیابی نرم‌افزارها در بالا ذکر شد، توجه به نظرات کاربران در مقابل نظرات متخصصان دسته‌بندی دیگری در حوزه ارزیابی نرم‌افزارها پدید می‌آورد. از آن‌جا که بهره‌مند شدنِ کاربران نهایی، دلیل و هدف اصلی تولید بسیاری از نرم‌افزارها به‌شمار می‌رود، می‌توان «ارزیابی کاربرمحور» را یکی از زمینه‌های اصلی ارزیابی نرم‌افزارها عنوان نمود. از سویی دیگر، تعدادی از نرم‌افزارها را با استفاده از روش «ارزیابی مکاشفه‌ای (۲۷)» به ملاحظه تعداد مشخصی کارشناس (بین ۳ تا ۵ نفر) در زمینه کاری نرم‌افزار می‌رسانند و در این نوع ارزیابی‌ها، مشخصات فنی و کارآیی نرم‌افزار از دیدگاه متخصصان مورد بررسی قرار می‌گیرد. در واقع، ارزیابی نرم‌افزارها با استفاده از شیوه پیمایش می‌تواند از دو دیدگاه کاربرمحور یا فنی (متخصص‌محور) انجام شود.

گذشته از این ابزارها، شیوه‌ها و روش‌ها، این‌که کدام قسمت یا کارآیی نرم‌افزارها باید مورد بررسی قرار گیرد، می‌تواند حوزه‌های متفاوتی از ارزیابی را تعیین کند؛ به این معنا که هر کدام از خدمات نرم‌افزارها می‌توانند به صورت جداگانه مورد ارزیابی قرار گیرند. در ادامه، زمینه‌ها و کارکردهای متفاوتی که در متون درباره ارزیابی نرم‌افزار به آن‌ها اشاره شده است، مرور خواهد شد.

رابط کاربریِ نرم‌افزارها، نقطه ارتباط رایانه با کاربران به شمار می‌رود. به واسطه رابط کاربر است که تعامل بین کاربر و نرم‌افزارها شکل می‌گیرد. در نتیجه، به هر میزان، کاربران بتوانند تعامل آسان، سریع و قابل فهمی با نرم‌افزارها داشته باشند، کارایی نرم‌افزار بالاتر می‌رود و رضایت‌مندی کاربران از آن در سطح بالاتری قرار می‌گیرد. خالقی (۱۳۸۵) در مقاله خود، ضمن تأکید بر تأثیر رابط کاربر در جذب کاربران و مطلوبیت نرم‌افزارها، برای آن‌ها معیارهایی عمومی را برمی‌شمرد که رعایت آن‌ها در رابط کاربریِ تمامی نرم‌افزارها ـ یا شاید تمامی محیط‌های رایانه‌ای که قرار است با انسان تعامل داشته باشند ـ می‌تواند به بهبود عملکرد نرم‌افزار و دستیابی به هدف‌اش در تعامل با کاربر کمک کند. این معیارهای عمومی عبارت‌اند از:

۱. ثبات :هر چه صفحات و علمکردهای نرم‌افزار از ثبات بیشتری برخوردار باشند، زمان کمتری برای یادگیری و انس کاربر با محیط نرم‌افزار نیاز است. نمونه ثبات در طراحی رابط کاربری را در محصولات شرکت مایکروسافت، مثلاً در مجموعه نرم‌افزارهای آفیس می‌بینیم.

۲. سادگی و وضوح :سادگی رابط کاربر، علاوه بر آن‌که به یادگیری و استفاده کمک می‌کند، یاری‌گر ثبات طراحی نیز هست. با دوری از طراحی پیچیده در رابط کاربر نرم‌افزارها می‌توان وضوح و شفافیت را در مسیرهایی که کاربر باید برای انجام کارکرد مورد نظرش پی‌گیر شود، افزایش داد.

۳. رهنمونی :ویژگی رهنمونی، به هشدارها و جملاتی اشاره دارد که وظیفه راهنمایی کاربر را در هنگام عملکردهای گوناگون به عهده دارد. پیشنهادهای جستجویی که گوگل بعد از جستجو با املای غلط به ما ارائه می‌دهد، از این دسته‌اند.

۴. زیبایی :بدون شک، چشم‌نواز بودن رابط کاربری در عین سادگی و آسانی عملکرد، می‌تواند راهبرد مؤثری جهت جذب و تأمین رضایت کاربران باشد.

۵. گرافیک و ترکیب رنگ‌ها :لحاظ کردن مفهوم رنگ‌ها با در نظر گرفتن تفاوت‌های فرهنگی، در معناداری و تناسب رابط کاربری تأثیر به‌سزایی دارد.

۶. استفاده از پیام‌ها و علائم اخطار دهنده :هشدارها و اخطارهای به‌موقع و به‌جا، از ارکان مهم طراحی رابط کاربر نرم‌افزارها است که عدم توجه به آن، می‌تواند نارضایتی کاربران را به همراه داشته باشد.

۷. انعطاف‌پذیری :برطرف کردن نیازمندی‌های متفاوت کاربرانی با سلیقه‌های متفاوت را می‌توان به عنوان یکی از معیارهای مهم رابط کاربری‌های قوی برشمرد. تنوع کارکردها و سازگاری نرم‌افزارها با برنامه‌های دیگر، یکی از اساسی‌ترین ملاک‌های انتخاب کاربران از بین نرم‌افزارهای مختلف در یک حوزه است.

۸. توجه به اصول روان‌شناسی :آسانی و راحتی کاربری نرم‌افزارها، یکی از تبعات توجه به اصول روان‌شناسی است و ملاک مهمی برای موفقیت نرم‌افزارها است.

۹. سایر معیارها :تسلط به محیط، صراحت، کاهش حجم کار کاربران و مساعدت، از دیگر ملاک‌هایی است که خالقی به عنوان معیارهای عمومی طراحی رابط کاربر برمی‌شمرد.

البته طراحی و زیباسازی، توجه به کاربرپسندی و تعامل‌پذیرتر کردن رابط‌های کاربری، همیشه به طور مستقیم با استفاده از برنامه‌نویسی در خود نرم‌افزارها انجام نمی‌شود. یکی از راهکارهای ارتقای ویژگی‌های رابط کاربری، استفاده از نرم‌افزارهای سیستمی و ابزارهای دیگری است که منحصراً برای کمک به طراحی رابط‌های کاربری تولید شده‌اند. مایر (۱۹۹۴) در مقاله‌اش با تأکید بر اهمیت رابط کاربری نرم‌افزارها، ابزارهایی را معرفی می‌کند که طراحان می‌توانند از آن‌ها استفاده کنند تا با صرفه‌جویی در زمان، به مطلوبیت بالاتری از نظر نمایش و رابط کاربری دست پیدا کنند. مقاله مایر اگرچه مقاله‌ای قدیمی است، اما تأکید او و هم‌نسلانش بر نمایش اطلاعات، به نوعی جهت‌گیری صاحب‌نظران و دست‌اندرکاران را از حدود بیست سال پیش برای گرافیکی‌تر نمودن نرم‌افزارها نشان می‌دهد. پیگیری این نکته و پافشاری بر روی آن، تا جایی پیش رفته است که امروزه، همین ابزارهای گرافیکی، علاوه بر طراحان نرم‌افزار، در اختیار کاربران در محیط‌های مختلف قرار می‌گیرد و امکان شخصی‌سازی نمایش در بسیاری از محیط‌ها را فراهم آورده است.

طراحی رابط کاربری برای پایگاه‌های اینترنتی نیز از اهمیت بالایی برخوردار است. یمین‌فیروز (۱۳۸۲) به بررسی عناصر مهم در طراحی رابط کاربری پایگاه‌ها می‌پردازد و معیارها را در سه دسته کلی ساختاری (شامل: محتوا و مندرجات، انسجام، سازماندهی، استفاده از تکنیک فرارسانه‌ای پیوندها)، ظاهری (شامل: ششمایل، تصاویر و صفحه‌آرایی) و کاربری (برای انواع کاربر: مبتدی، تصادفی و ماهر) دسته‌بندی می‌کند.

در نرم‌افزارهای تخصصی حوزه‌های متفاوت نیز، ارزیابی رابط کاربری یکی از زمینه‌های مهم برای ارزیابی به شمار می‌رود؛ برای مثال، در حوزه نرم‌افزارهای کتابخانه‌ای و همچنین پایگاه‌های اطلاعاتی وبی پژوهش‌هایی چند با موضوع رابط کاربری به چشم می‌خورد. زره‌ساز، فتاحی و داورپناه (۱۳۸۵)، بعد از تحلیل ویژگی‌های رابط کاربر نرم‌افزار کتابخانه‌ایِ«سیمرغ»، میزان رضایت کاربران را نیز از تعامل با این نرم‌افزار بررسی نموده‌اند. از نظر کاربران (نمونه‌ای متشکل از دانشجویان دانشکده روان‌شناسی و علوم تربیتی دانشگاه فردوسی مشهد)، ویژگی‌هایی از نرم‌افزار مذکور، نظیر: صفحه‌های نمایش اطلاعات نرم‌افزار، واژگان و پیام‌های نرم‌افزار، یادگیری چگونگی کار با نرم‌افزار، راهنمای نرم‌افزار و قابلیت‌ها و امکانات نرم‌افزار سیمرغ در حد متوسطی قرار دارد و سطح رضایت آنان از تعامل با نرم‌افزار، در حد متوسط است. عوامل متعددی ممکن است بر میزان رضایت کاربران از تمایل با نظام‌های رایانه‌ای تأثیر بگذارند. با وجود این، در این پژوهش، تنها تأثیر پنج عامل: رشته تحصیلی، مقطع تحصیلی، سابقه و میزان آشنایی با نرم‌افزار، سواد رایانه‌ای و سواد کتابخانه‌ای کاربران، بر میزان رضایت آن‌ها از تعامل با نرم‌افزار سیمرغ سنجیده شد. نتایج این پژوهش نشان داد که مقطع تحصیلی کاربران و همچنین میزان سواد رایانه‌ای آن‌ها، به طور معناداری بر میزان رضایت آن‌ها از تعامل با نرم‌افزار سیمرغ مؤثر است. در پایان، آن‌ها پیشنهادهایی کاربردی در زمینه‌های ذیل ارائه داده‌اند:

۱. صفحه نمایش اطلاعات، (مانند: استفاده از فونت‌های متنوع فارسی و عدم استفاده از فونت‌های عربی، استفاده بهینه از فضاهای خالی صفحه‌های نمایش اطلاعات، استفاده معقول از ابزارهای جلب توجه‌کننده کاربران در صفحه‌های نمایش و...؛

۲. واژگان و پیام‌های نرم‌افزار، نظیر: استفاده از واژگان ساده‌تر و قابل فهم‌تر برای تمام کاربران، بازنگری در بعضی واژگان، استفاده معقولانه از واژگان تخصصی در این نرم‌افزار و...؛

۳. بهبود یادگیری چگونگی کار با نرم‌افزار سیمرغ، همچون: تهیه و تدوین برنامه‌های آموزشی پیوسته، برگزاری جلسه‌های منظم آموزشی، انتخاب قالبی مشخص و یکدست برای عملیات جستجوی پیشرفته و...؛

۴. راهنمای نرم‌افزار، مانند: ترغیب کاربران به استفاده از راهنمای نرم‌افزار، استفاده از واژگان ساده‌تر و همه فهم‌تر و...؛

۵. قابلیت‌ها و امکانات نرم‌افزار، از قبیل: توسعه و گسترش امکانات و قابلیت‌های جدید نرم‌افزار، افزایش سرعت پاسخگویی نرم‌افزار، پیش‌بینی امکان تصحیح عبارت جستجوی مجدد و....

در همین زمینه، مهاجری و محمد صالحی (۱۳۸۸)، در پژوهشی پیمایشی و از نوع کاربردی، ویژگی‌های رابط کاربر نرم‌افزار کتابخانه‌ای رسا را با استفاده از روش مکاشفه‌ای مورد مطالعه قرار دادند. ده مؤلفه اصلی که در سیاهه وارسی بیان شده‌اند، عبارت‌اند از: وضعیت رؤیت سیستم، تطبیق بین سیستم و جهان واقعی، کنترل و آزادی کاربر، ثبات و استانداردها، کمک به کاربر در شناسایی و تشخیص و جبران خطاها، پیشگیری از خطا، تشخیص به جای یادآوری، انعطاف‌پذیری و کارایی سیستم، زیبایی‌شناختی و راهنمایی، و مستندسازی.

تجزیه و تحلیل داده‌های پژوهش آنان نشان داد که در طراحی رابط کاربر نرم‌افزار رسا، ۶۴ درصد از معیارهای مورد نظران تعدادی از کارشناسان متخصص در حوزه کتابداری لحاظ شده است و بیشترین تطابق (۹۰ درصد) در زمینه زیبایی‌شناختی بوده است. همچنین از بین ویژگی‌ها و معیارهای مورد ارزیابی، ویژگی «پیشگیری از خطا»، بالاترین میانگین شدت، و ویژگی «ثبات و استانداردها» و «تشخیص به جای یادآوری»، پایین‌ترین میانگین شدت را دارند.

منابع:

۱. اسکندری، فائزه (۱۳۸۸). «ارزیابی نرم افزارهای علوم اسلامی (قرآن، حدیث، فقه و احکام) شهر قم از سه دیدگاه رابط کاربر، قابلیتهای جستجو و تجربه کاربری». پایان‌نامه کارشناسی ارشد، دانشگاه تربیت مدرس، دانشکده مدیریت و اقتصاد.

۲. انتظاریان، ناهید، رحمت‌الله فتاحی (۱۳۸۸). «بررسی درک کاربران از محیط رابط پایگاه‌های اطلاعاتی بر اساس مدل نیلسن (مقایسه پایگاه مقاله‌های الکترونیکی مرکز منطقه‌ای اطلاع‌رسانی علوم و فناوری با پایگاه مقاله‌های پژوهشگاه اطلاعات و مدارک علمی ایران)». فصلنامه کتابداری و اطلاع‌رسانی، ۱۲ (۳)، ۴۳-۶۴.

۳. حاجی زین‌العابدینی، محسن، داودزاده سالستانی، سیروس و پازوکی، فاطمه (۱۳۹۰). نرم‌افزارهای کتابخانه‌ای در ایران. تهران، نشر کتابدار.

۴. خالقی، نرگس (۱۳۸۵). مهم‌ترین معیارهای عمومی رابط کاربر مطلوب. کتابداری و اطلاع‌رسانی، ۹ (۳)، ۸۶-۹۶.

۵. ضایی، محمدجواد (۱۳۷۷). «ارزشیابی قابلیت‌های ذخیره و بازیابی در برنامه‌های نرم‌افزاری مراکز اطلاع‌رسانی علوم قرآنی و معارف اسلامی (در شهرهای تهران، قم و اصفهان)». پایان‌نامه کارشناسی ارشد، دانشگاه تربیت مدرس، دانشکده روان‌شناسی و علوم تربیتی.

۶. زره‌ساز، محمد، فتاحی، رحمت‌الله و داورپناه، محمدرضا (۱۳۸۵). «بررسی و تحلیل عناصر و ویژگی‌های مطرح در رابط کاربر نرم‌افزار سیمرغ و تعیین میزان رضایت دانشجویان دانشکده علوم تربیتی وروان‌شناسی دانشگاه فردوسی مشهد از تعامل با این نرم‌افزار». کتابداری و اطلاع‌رسانی، دوره ۹، (۴)، ۱۲۷-۱۵۹.

۷. عباس‌پور، جواد، رضایی شریف‌آبادی، سعید (۱۳۸۵). «ارزیابی رابط کاربر پایگاه اطلاعات پایان‌نامه‌های مرکز اطلاعات و مدارک علمی ایران از دید کتابداری و اطلاع‌رسانی». فصلنامه پژوهش و برنامه‌ریزی در آموزش عالی، (۴۱)، ۷۵-۹۲.

۸. غائبی، امیر (۱۳۷۴). «ارزیابی نرم‌افزارهای کتابخانه‌ای: معیارهای سنجش پایگاه داده کتاب‌شناسی». فصلنامه کتاب، سال ششم (۳، ۴)، ۳۵-۶۸.

۹. مهاجری، فاطمه، محمد صالحی، راحله (۱۳۸۸). «ارزیابی ویژگی‌ها و عناصر تشکیل‌دهنده رسا: نرم‌افزار جامع کتابخانه، مرکز اسناد و آرشیو بر اساس عامل رابط کاربر». فصلنامه دانش‌شناسی (علوم کتابداری و اطلاع‌رسانی)، (۶)، ۷۵-۸۷.

۱۰. نوروزی، یعقوب، نعمتی، سحر (۱۳۸۹). «ارزیابی نرم‌افزارهای جامع کتابداری تحت وب پارس آذرخش، نوسا و نمایه در بازیابی اطلاعات». تحقیقات اطلاع‌رسانی و کتابخانه‌های عمومی، سال شانزدهم (۶۰)، ۲۳ - ۴۳.

۱۱. یمین‌فیروز، موسی (۱۳۸۲). «ویژگی‌ها و عناصر تشکیل‌دهنده رابط کاربر در وب‌سایت‌ها». فصلنامه کتاب، (۵۶)، ۱۵۹-۱۶۸.

12. Acuna, Silvia T. , DE Antonio, A. , Ferre, X. , Lopez, M. , & Mate, L. (2000). The Software Process: Modeling, Evaluation and Improvement. In S. K. Chang (Ed.), Handbook of Software Engineering and Knowledge Engineering (pp. 193-237). River Edge, NJ: World Scientific.

13. Boehm, B. W. , Brown, J. R. , & Lipow, M. (1976). Quantitative Evaluation of Software Quality. Paper presented at the Proceedings of the 2nd international conference on Software engineering. iEEE Computer Society.

14. Gediga, G. , (2002) Hamburg, K. , and Duntsch, I.: Evaluation of SoftwareSystems. Encyclopedia of Library and Information Science,72:166-192.

15. Hameed, Shihab A. , Khateeb and Mutaz, Zubayda (2010). Software Engineer Islamic Ethics An Interactive Web-Based Model. Paper present at International Conference on Computer and Communication Engineering (pp. 1-7).
16. Kitchenham, B. , linkman, S. & law, D. (1996). DESMET: A method for evaluating Software Engineering methods and tools. Computing & Control Engineering Journal, 8 (3), 120-126.
17. Kitchenham, B. , Pickard, l. &Lawrence Pfleeger, S. (1995). Case Studies for Method and Tool Evaluation. IEEE Software, 12 (4), 52-65.
18. komi Sirvio, S. , (2004). Development and evaluation of software process improvement methods. Espoo: VTT publication.
19. Lanubile, F. (1997). Empirical Evaluation of Software Maintenance Technologies. Empirical Software Engineering, 2, 97-108.
20. Mayer, B. (1994). User Interface Software Tools. Advances in Human- Computer Interaction, 4, 110-150.
21. Miyoshi, T. , Togashi, Y. , & Azuma, M. (1989). Evaluating software development environment quality. Paper present at Proceedings of the 13th Annual International Computer Software and Applications Conference, COMPSAC. (pp. 501-508).
22. Nielsen, J. (1994). Chapter 5 – Usability Heuristics. In Usability Engineering (2nd ed. , pp. 115-163). San Diego, California: Academic Press.
23. Owston, R. (2008). Models and Methods for Evaluation. Handbook of Research on Educational Communications and Technology (3rd ed. , pp. 605-617). New York, NY: Routledge.
24. Rowley, J. E. , (1992). Selection and evaluation of software. Aslib Proceedings, 45(3), 77-81.