فصلنامه علمی- پژوهشی
فناوری اطلاعات و ارتباطات ایران

سال نهم، شماره‌های ۳۱ و ۳۲، بهار و تابستان ۱۳۹۶
صص: ۶۶- ۵۷

ارائه یک روش اولیه جهت تخمین پروژه‌های نرم‌افزاری مبتنی بر تراکنش منطقی

* مهرداد شهسواری ** علی مهجور
* دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه آزاد اسلامی، واحد آشتیان، ایران
** دکتری نرم افزار ، دانشگاه صنعتی مالک اشتر، تهران، ایران
تاریخ دریافت: ۱۳۹۲/۱۱/۰۹ تاریخ پذیرش: ۱۳۹۳/۰۶/۲۴

چکیده

در این مطالعه به منظور تعیین میزان توجّه بصری کاربران به منوی راست و به منوی چپ صفحات وب شاخص تعداد خیرگی‌ها روی منوی راست و منوی چپ با استفاده از دستگاه ردیاب چشم سنجیده شد تا مشخص گردد که کدام منو به لحاظ توجّه بصری برای کاربران در ارجحیت قرار دارد. روش: ۱۱۶صفحهٔ حاوی دو منوی راست و چپ در قالب ۳ مجموعهٔ فارسی- انگلیسی و انگلیسی- فارسی برای ۳۰ آزمودنی نمایش داده شد و آزمودنی‌ها ملزم به یافتن یک لغت در منوها بوده‌اند. داده‌های مربوط به تعداد خیرگی‌های کاربران روی هر منو که نشانی از میزان توجّه بصری آن‌ها به آن منو بود با استفاده از دستگاه ردیاب نگاه ثبت و جمع آوری شد. یافته‌ها: تعداد کل خیرگی‌های کاربر بر منوهای مجموعهٔ انگلیسی با توجّه به جهت چینش منوها (راست یا چپ) متفاوت نبود. اما تعداد کل خیرگی‌های کاربر بر منوهای مجموعهٔ فارسی، مجموعهٔ فارسی- انگلیسی و به طور کلی در مجموع همهٔ صفحات سه مجموعه با توجّه به جهش چینش منوها (راست یا چپ) متفاوت و روی منوی راست بیش از منوی چپ بود. نتیجه‌گیری: با توجّه به بیش‌تر بودن تعداد خیرگی‌ها روی منوی راست در این مطالعه و مزیت‌های اثبات شدهٔ منوی راست و همین طور به دلیل آن که تا به حال مزیت قابل پیش‌بینی بودن منوی چپ چین در فرهنگ‌هایی بوده است که زبان مادریشان از چپ به راست خوانده می‌شود و این‌که نتایج مطالعات پیشین پیرامون سرعت بالای انجام تکالیف و تعامل با منوی چپ چین، متناقض است، می‌توان به طراحان وب سایت‌ها توصیه کرد که در طراحی وب سایت‌های بومی از منوی راست چین استفاده کنند.

واژه‌های کلیدی: توجّه بصری، دستگاه ردیاب چشم، صفحات وب، منو.

۱. مقدمه

اندازه‌گیری از ملزومات اولیه پروژه‌های نرم‌افزاری می‌باشد. آنچه که قابل اندازه‌گیری نباشد قابل کنترل و در نتیجه قابل مدیریت علمی و دقیق نیز نمی‌باشد. مدیریت پروژه بدون اندازه‌گیری تبدیل به مدیریت حسی، غیرعلم ی و غیر دقیق می‌شود. حتی در صورتیکه پروژه موفق باشد، موفقیت آن در بعد مدیریتی بصورت آگاهانه قابل تکرار نمی‌باشد. مدیر پروژه باید بتواند اهداف کیفی و پیشرفت پروژه را به سنجه‌ها یا پارامترهای قابل اندازه‌گیری تبدیل نماید و آن‌ها را در طول پروژه اندازه‌گیری نماید.

نویسندهٔ عهده‌دار مکاتبات: مهرداد شهسواری msh89102003@yahoo.com

تخمین اندازه محصول، کلیدی‌ترین ورودی فرایند تخمین پروژه می‌باشد. در صورتی که تخمین اندازه محصول درست نباشد، زمانبندی انجام شده دور از واقعیت خواهد بود. [۱].

روش‌های تخمین پروژه‌های نرم‌افزاری اغلب براساس اطلاعات پروژه‌های قبلی و پایان یافته می‌باشند. پژوهشگران و مهندسین، اطلاعات پروژه‌ها را مطالعه کرده و معادلات و رابطه‌هایی را ارایه نموده‌اند. برخی از رابطه ها خیلی ساده و برخی دیگر پیچیده می باشند. در مجموع هیچ‌کدام با همه پروژه‌های پایان یافته قبلی کاملاً مطابق نیستند. آن‌ها بیشتر به طور احتمالی و براساس رابطه‌هایشان حجم کار درخواست شده برای پروژه را پیش‌گویی می‌کنند[۵].

صدها روش تخمین نرم‌افزار مستند وجود دارد که برخی از آن‌ها هزینه نرم‌افزار را تخمین می‌زنند. در بیشتر موارد تخمین دارای تنوع وسیعی از ورودی‌ها بوده و خروجی آن نیروی انسانی است. بسیاری از روش‌ها نیز از یک رابطه تحلیلی براساس کنترل‌کننده‌های هزینه (که به نوعی خصوصیات و مشخصات پروژه مانند اندازه سامانه، حوزه سامانه، پیچیدگی و روش توسعه هستند) استفاده می‌کنند [۵].

۲. بیان مساله

امروزه تعداد زیادی از پروژه‌های نرم‌افزاری در کشور ما از نظر زمان، هزینه و رفع تمام نیازهای مشتری، با مشکل روبرو شده و در نهایت با لغو شدن و شکست پروژه مواجه می‌شوند. درصد بالایی از پروژه‌های نرم‌افزاری در همان مراحل ابتدایی کار و پس از تحلیل نیازمندی‌ها به دلایل گوناگون با مشکل مواجه می‌گردند.

به همین دلیل توجه به پروژه‌هایی که به موقع به پایان رسیده‌اند می‌تواند کمک بزرگی در جلوگیری از شکست پروژه در میان راه باشد.

البته حتی پروژه‌های موفق نیز در نحوهٔ رسیدن به این مرحله یعنی از شکست تا موفقیت، با هم متفاوت می‌باشند. یکی از مهمترین تفاوت‌ها نحوهٔ رسیدن آن‌ها به زمانبندی، هزینه و منابع پیش‌بینی شده وکیفیتی است که در ابتدای کار تخمین زده می‌شود.

متاسفانه در کشور ما با وجود جایگاه بالای تخمین و میزان تاثیرگذاری آن بر میزان موفقیت پروژه‌های نرم‌افزاری، بدلیل پیچیدگی بالا، سرعت کم و پائین بودن دقت ابزارهای خودکار تخمین پروژه نرم‌افزاری، تخمین اندازه، تلاش، زمان و هزینه بصورت تجربی و قیاسی انجام می‌گردد.

هدف و سوال اصلی این تحقیق این است که چگونه می‌توان روشی اولیه جهت تخمین پروژه‌های نرم‌افزاری تراکنشی ارایه نمود که در یکی از زمینه‌های سرعت، دقت و یا سادگی نسبت به سایر روش‌های موجود بهتر باشد؟

۳. ادبیات پژوهش

پاول کومبز[1] در رابطه با تخمین پروژه‌های نرم‌افزاری می‌گوید: "هیچ کس تخمین‌زدن را دوست ندارد. تخمین‌زدن در شغل ما بی اجرترین کاری است که می‌توان انجام داد_ قبول مسوولیت سنگینی است برای کاری دشوار و دقیق. اگر دنبال درخشیدن هستید از تخمین‌زدن پرهیز کنید." [۳].

با وجودی که تخمین از نظر مفهوم چیز ساده‌ای است اما در حقیقت بسیار پیچیده است. میزان موفقیت تخمین نرم‌افزار در صورت استفاده از ابزارهای تخمین بسیار بیشتر از روش‌های تخمین دستی می‌باشد. از طرفی ابزارهای تخمین نرم‌افزار تجاری نیز بدون نقص نیستند و حتی می‌توانند نادرست هم باشند. اما ابزارهای خودکار اغلب از نظر دقت و همیشه از نظر سرعت بی‌نقص‌تر از تخمین‌های انسانی می‌باشند.

در مجموع هیچ یک از روش‌های تخمین عاری از خطا نیستند. بهترین روش رایج برای تخمین هزینه نرم‌افزار استفاده ترکیبی از ابزارهای تخمین هزینه نرم‌افزار به همراه ابزارهای مدیریت پروژه نرم‌افزاری زیر نظر مدیران با تجربه پروژه‌های نرم‌افزاری و متخصصین تخمین می‌باشد. [۴].

سه روش بنیادی و مطرح در تخمین پروژه‌های نرم‌افزاری روش تحلیل درجه عملکردی[2]، روش کوکومو به نام "مدل هزینه ساخت" و روش تخمین درجه موردکاربرد[3] می‌باشند

---

1. Paul Coombs
2. Function Point Analysis (FPA)
3. Usecase Point (UCP)

که در ادامه شرح مختصری از هرکدام به طور جداگانه بیان می‌گردد.

۴. روش تحلیل درجه عملکردی

در این روش اندازه‌گیری نرم‌افزار مستقل از زیرساخت[۴] توسعه استفاده شده برای آن می‌باشد. در روش تحلیل درجه عملکردی اندازه نرم‌افزار بر مبنای یک سنجه اختصاصی با نام "FunctionPoint" و با توجه به جداول منطقی[۵] و تراکنش[۶]های مورد نیاز در نرم‌افزار می‌باشند.

در این روش، اندازه نرم‌افزار با تعیین تعداد عملکرد[۷] تهیه شده برای کاربر به دست می‌آید. اندازه‌گیری در این روش مستقل از پایگاه[۸] توسعه استفاده شده است و نیازمندی‌های عملکردی کاربر[۹] اساس اندازه‌گیری هستند [۴].

۵. روش کوکومو به نام "مدل هزینه ساخت"

روش کوکومو وابستگی زیاد به توانائی مدیر برای تخمین اندازه نرم‌افزار دارد. کوکومو برای تخمین اندازه نرم‌افزار از روش تخمین درجه عملکردی استفاده می‌کند.

در این روش ابتدا درجه عملکردی یا تعداد FP پروژه محاسبه می‌گردد و پس از محاسبه FP زبان برنامه‌نویسی پروژه مشخص و بر اساس آن LOC پروژه نیز بدست می‌آید [۹].

۶. روش تخمین درجه موردکاربرد

روش تخمین درجه موردکاربرد به عنوان روش تخمین هزینه و زمان در برخی از ابزارهای مهندسی نرم‌افزار که از UML برای مدل‌سازی استفاده می‌کنند، پیش‌بینی شده است. از آن‌جایی‌که در دیدگاه خدمت‌گرا[۱۰] که امروزه مقبولیت و عمومیت بسیاری در مدل‌سازی سامانه‌ها پیدا کرده است، شناخت سامانه‌ها و تعیین نیازمندی‌ها مبتنی بر تعیین موردهای‌کاربرد است. لذا مستقیما اندازه پروژه‌ها را براساس درجه موردکاربرد و بدون در نظر گرفتن درجه عملکردی مشخص می‌کنند. در این راستا میزان پیچیدگی عملگرها[۱۱]، میزان بزرگی سناریوی موردکاربرد، فاکتورهای محیطی و فاکتورهای تکنیکی مطرح می‌شود [۷].

۷. سابقه تحقیقات و مطالعات انجام گرفته

در سطح داخلی (کشور ایران) تحقیقات زیادی روی تخمین پروژه‌های نرم‌افزاری انجام نشده است و با وجود جایگاه بالای آن، در کتب مهندسی نرم‌افزار خیلی مختصر به آن اشاره شده است و در بین پایان‌نامه‌ها و مقاله‌های موجود در سطح دانشگاه‌ها نیز مطالب زیادی در رابطه با تخمین پروژه‌های نرم‌افزاری یافت نمی‌شود.

در سطح خارجی هم تا سال ۱۹۹۷ روش‌هایی مانند تحلیل درجه عملکردی، مدل هزینه ساخت، درجه عملکرد موردکاربرد و غیره ارایه شده‌اند ولی از سال ۱۹۹۷ به بعد تحقیقات محققین بیشتر بر اصلاح روش‌های ارایه شده و بهبود روش‌ها متمرکز بوده است و می‌توان گفت در این دوره روش مطرح جدیدی ارایه نشده است. طی این دوره محققین همواره بدنبال آن بوده‌اند که بتوانند در چرخه حیات پروژه‌های نرم‌افزاری روشی برای پاسخ به نیازمندی‌های مشتری‌ها در زمانی کمتر ارایه دهند و سرعت تخمین را افزایش دهند.

طی این سال‌ها یک تیم نرم‌افزاری بنام "تیم راه‌حل‌های چابک و سریع" شروع به جمع‌آوری سنجه‌های جدید تخمین پروژه‌های نرم‌افزاری نموده است [۲].

۸. روش پژوهش

از آنجا که هدف این تحقیق ارائه یک روش جدید و افزایش دانش جدید در حوزه تخمین پروژه‌های نرم‌افزاری است این تحقیق یک تحقیق بنیادی است و از طرفی چون به دنبال ارایه یک الگوی مناسب و عملی برای تخمین پروژه‌های نرم‌افزاری هستیم و الگوی ما در قالب یک نرم‌افزار کاربردی پیاده‌سازی خواهد شد از نظر نوع هدف، تحقیق ما کاربردی هم می‌باشد.

این تحقیق از نظر روش جمع‌آوری داده‌ها از نوع "توصیفی" می‌باشد. تحقیق توصیفی را می‌توان به پنج دسته پیمایشی،

4. Platform
5. Logical file
6. Transaction
7. Functionality
8. Platform
9. Functional User Requirement (FUR)
10. Service Oriented
11. Actor

همبستگی، اقدام پژوهی، بررسی موردی و پس رویدادی تقسیم کرد که تحقیق حاضر از نوع تحقیق اقدام پژوهی و بررسی موردی می‌باشد.

۹. یافته‌های پژوهش

ما در این تحقیق مختص تخمین نرم‌افزارهای تراکنشی و مبتنی بر داده به روشی دست یافته‌ایم که ورودی مورد نیاز آن نیازمندی‌های عملکردی سامانه می‌باشد. این روش مناسب جهت تخمین در مراحل اولیه پروژه است. هدف از ارایه این روش رسیدن به یک سنجه مناسب برای اندازه‌گیری نرم‌افزار است که با تعداد خطوط کد تناسب زیادی داشته باشد و در یکی از زمینه‌های سادگی، دقت و سرعت از سنجه‌های دیگر بهتر باشد.

روش پیشنهادی ما پس از شناسایی نقاط ضعف و قوت روش‌های موجود، در راستای تقویت نقاط قوت و رفع نواقص ارایه گردیده است. این روش کلیه مولفه‌های موثر در روش‌های دیگر تخمین پروژه‌های نرم‌افزاری را مد نظر قرار می‌دهد.

۱۰. روش تخمین درجه تراکنش منطقی[۱۲]

در این روش هر مورد کاربرد به تعدادی تراکنش شکسته می‌شود. اندازه هر تراکنش بصورت جداگانه بر اساس فاکتورهای اندازه تراکنش (TSF) به دست می‌آید. سپس با جمع کردن اندازه تراکنش‌های هر مورد کابرد، اندازه هر مورد کاربرد و در نهایت با جمع کردن اندازه موارد کاربرد، اندازه کل پروژه به دست می‌آید. اندازه‌های بدست آمده برای تراکنش‌ها، موارد کاربرد و کل پروژه بر مبنای سنجه ابداعی با نام درجه تراکنش منطقی (LTP) می‌باشد. در این مقاله اثبات خواهیم کرد که اندازه‌های بدست آمده بر اساس این سنجه (نسبت به سنجه UCP) همبستگی بیشتری با اندازه واقعی نرم‌افزار (بر اساس سنجه KLOC) دارد.

۱۱. انواع تراکنش‌های منطقی

نیازمندی‌ها کارهایی هستند که برنامه باید انجام دهد و چگونگی آن مهم نیست نیازمندی‌ها قابلیت‌ها و شرایطی هستند که سیستم و پروژه باید از آن‌ها پیروی نماید. نیازمندی‌ها به دو دسته تقسیم می‌شوند:

۱- عملکردی[۱۳]: چه کارهایی باید انجام شود (رفتار سیستم).

سیستم).

۲- غیرعملکردی[۱۴]: کارها چگونه باید انجام شوند (صفات سیستم) [۱].

در نرم‌افزارهای تراکنشی نیازمندی‌های عملکردی کاربران در قالب دو بخش ورود اطلاعات به سامانه و خروج اطلاعات از سامانه قرار می‌گیرند.

ورود اطلاعات به سامانه معادل با تراکنش‌های منطقی ثبت اطلاعات و خروج اطلاعات از سامانه متناظر با تراکنش‌های منطقی مشاهده اطلاعات می‌باشد. از طرفی کدهایی که برآورده‌کننده نیازمندی عملکردی سامانه نرم‌افزاری می‌باشند، در نهایت منجر به یکی از تراکنش‌های منطقی ثبت، ویرایش، حذف و مشاهده اطلاعات می‌شوند. پس جهت اندازه‌گیری بخش عملکردی نرم‌افزار می‌توان تعداد و ویژگی‌های تراکنش‌های منطقی ثبت، ویرایش، حذف و مشاهده را طی یک مرحله تحقق[۱۵] از موارد کاربرد استخراج کرد.

۱۲. فاکتورهای موثر بر اندازه تراکنش‌ها

برای تخمین اندازه هر تراکنش شش فاکتور مطابق ۰ پیشنهاد می‌گردد. این فاکتورها بر اساس تحلیل و جمع‌بندی فاکتورهای استفاده شده برای تخمین اندازه نرم‌افزار در روش‌های دیگر و همچنین تجربه بدست آمده‌اند. در ادامه مقاله نگاشت فاکتورهای پیشنهادی به فاکتورهای روش‌های مطرح دیگر آورده شده است. با توجه به یکسان نبودن میزان تاثیر این فاکتورها در اندازه تراکنش، وزن پیشنهادی برای هر یک از این فاکتورها نیز در این جدول آورده شده است. وزن‌های پیشنهادی بصورت تجربی و سعی و خطا بدست آمده‌اند.

12. Logical Transaction Point (LTP)

13. Functional
14. Non-Functional
15. Realization

جدول ۱. فاکتورهای موثر بر اندازه تراکنش‌ها

فاکتور	توضیح	وزن
TSF1	تعداد صفات موجودیت اصلی	۵
TSF2	تعداد موجودیت مرتبط با تراکنش	۵
TSF3	تعداد ورودی جهت مشاهده اطلاعات	۵
TSF4	تعداد خروجی در مشاهده اطلاعات	۱
TSF5	میزان محاسبات	۱۰
TSF6	نوع محیط نمایش اطلاعات	۲

۱۳. نحوه تاثیر فاکتورهای موثر بر اندازه انواع تراکنش

همه فاکتورهای ذکر شده در جدول ۱ بـر روی انـدازه همـه انواع تراکنش‌ها تاثیر ندارند. به عنوان مثـال فـاکتور TSF1 که همان تعداد صفات موجودیت می‌باشـد، تـاثیری بـر روی اندازه تراکنش حذف ندارد. فاکتورهای مـوثر بـر هـر نـوع از تراکنش‌ها در ۰ آورده شده است.

جدول ۲. نحوه تاثیر فاکتورهای موثر بر اندازه انواع تراکنش

نوع تراکنش	فاکتور موثر	نحوه تاثیر
ثبت و ویرایش	TSF1	موثر در ساخت واسط کاربری
	TSF2	موثر در ساخت واسط کاربری و موثر در کدنویسی
حذف	TSF2	موثر در میزان کنترل های لازم حذف
مشاهده	TSF2	موثر در ساخت واسط کاربری و موثر از نظر کنترل روابط بین موجودیت‌ها
	TSF3	موثر در ساخت واسط کاربری
	TSF4	موثر در ساخت محیط نمایش
	TSF5	موثر در کدنویسی
	TSF6	موثر در ساخت محیط نمایش

۱۴. افزایش دقت وزن‌های تعیین شده جهت عوامل موثر در اندازه تراکنش‌های منطقی

از بین ۲۵ پروژه فضای نمونه آماری، تعداد ده پروژه را جهت ورزیـدگی[16] مـدل پیشـنهادی و در راسـتای افـزایش دقـت وزن‌هـای تعیـین شـده جهـت عوامـل مـوثر در انـدازه تراکنش‌های منطقی (جدول۱) مورد استفاده قرار دادیم کـه وزن عوامل موثر در انـدازه تـراکنش‌هـای منطقـی بـه روش آزمون و خطا به شرح جدول۳ به دست آمد.

جدول ۳. تعیین وزن عوامل موثر در تراکنش‌های منطقی

فاکتور	ویژگی اندازه‌گیری شده	شرایط	وزن
TSF1	تعداد صفات موجودیت اصلی	کمتر از ۳	۵
		بین ۳ تا ۵	۱۰
		بیشتر از ۵	۱۵
TSF2	تعداد موجودیت مرتبط با تراکنش	کمتر از ۳	۵
		بین ۳ تا ۵	۱۰
		بیشتر از ۵	۱۵
TSF3	تعداد ورودی جهت مشاهده اطلاعات	کمتر از ۳	۵
		بین ۳ تا ۵	۱۰
		بیشتر از ۵	۱۵
TSF4	تعداد خروجی در مشاهده اطلاعات	کمتر از ۵	۱
		بین ۵ تا ۱۰	۲
		بیشتر از ۱۰	۳
TSF5	میزان محاسبات	بدون محاسبه	۰
		محاسبات معمولی	۱۰
		محاسبات پیچیده	۲۰
TSF6	نوع محیط نمایش	بدون محیط نمایش و جهت کنترل	۰
		نمایش در فرم	۲
		نمایش در قالب گزارش	۴

۱۵. محاسبه اندازه تراکنش‌های منطقی، اندازه موارد کاربرد و اندازه نرم‌افزار

بـا اسـتفاده ازجـدول۳ انـدازه هـر تـراکنش منطقـی را محاسبه نمودیم سپس با جمع اندازه تـراکنش‌هـای هـر مورد کاربرد، اندازه مورد کاربرد را بدسـت آوردیـم و در نهایت با جمع وزن کلیه تراکنش‌های منطقی نـرم‌افـزار، اندازه نـرم‌افـزار را بـر حسـب درجـه تـراکنش منطقـی (LTP) بدسـت آوردیـم. البتـه حاصـل جمـع وزن

16. Training

تراکنش‌ها در واقع ULTP[17] یعنی اندازه تعدیل‌نشده نرم‌افزار است که با اعمال فاکتورهای فنی و محیطی تعدیل خواهد شد. رابطه محاسبه ULTP برای هر تراکنش به صورت زیر است:

$$ULTP = \sum 1..6\ TSFi \quad \text{رابطه (۱)}$$

$$ULTP = \sum 1..n\ (\sum 1..6\ TSFi) \quad \text{رابطه (۲)}$$

در اینجا n تعداد تراکنش‌های منطقی پروژه و TSFi وزن فاکتور موثر بر هر تراکنش می‌باشد. لازم به ذکر است که TSFi ها بر حسب نوع تراکنش و با استفاده از جدول۲ و۳ تعیین می‌گردند. به عنوان مثال برای یک تراکنش حذف طبق جدول۲ فقط فاکتور موثر بر آن یعنی TSF2 مقداردهی و مابقی فاکتورها بصورت پیش‌فرض صفر در نظر گرفته می‌شوند.

۱۶. مطالعه موردی و اثبات روش

ما جهت اثبات دقیق‌تر بودن این روش نسبت به روش UCP از روش مطالعه موردی[18] و تحلیل‌های آمار استنباطی استفاده نمودیم. در این راستا تعداد ۱۵ پروژه نرم‌افزاری انجام شده را مورد بررسی قرار دادیم که با توجه به اینکه تعداد موارد کاربرد در این پروژه‌ها کمتر از ۲۰ موردکاربرد بود این پروژه‌ها جزء پروژه‌های کوچک محسوب می‌شدند. فرایند تولید این پروژه‌ها مبتنی بر RUP ، محیط برنامه نویسی آن‌ها C#.Net و پایگاه داده آن‌ها SQL Server 2000 بود.

در ادامه با رعایت کلیه موارد زیر، تعداد خطوط کد پروژه‌ها (LOC) را محاسبه نمودیم.

۱. کدهایی که در راستای آزمون و پشتیبانی سامانه نباشند، محاسبه می‌شوند.

۲. خطوطی از کد که به وسیله کارکنان تیم پروژه تولید می‌شود، مد نظر است.

۳. کدهایی که به وسیله سازنده‌های خودکار کد[19] تولید می‌شوند محاسبه نمی‌گردند.

۴. تعریف متغیرها[20] جزء کد هستند ولی توضیحات[21] شمارش نمی‌شوند[۸].

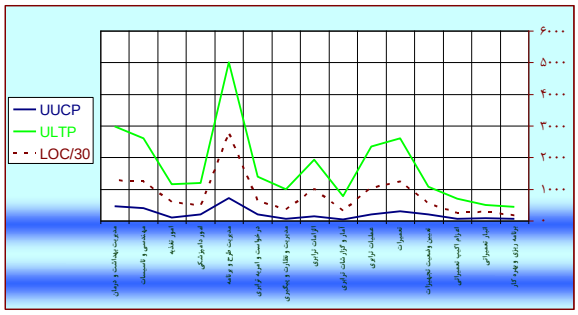
سپس اندازه تعدیل‌نشده پروژه‌ها را توسط روش UCP و روش پیشنهادی (LTP) محاسبه نمودیم که در نهایت نتایج به شرح ۰ بدست آمد.

جدول ۴. اندازه‌های بدست آمده از پروژه‌های نرم‌افزاری نمونه

R	پروژه نرم‌افزاری	UUCP	ULTP	LOC
۱	برنامه‌ریزی و بهره‌کار	۵۸	۴۲۸	۵۳۳۷
۲	انبار تعمیراتی	۷۰	۴۹۴	۸۸۳۶
۳	اعزام اکیپ تعمیراتی	۶۵	۶۹۰	۷۷۳۶
۴	تعیین وضعیت تجهیزات	۲۰۸	۱۰۶۴	۱۶۹۲۳
۵	تعمیرات	۳۰۶	۲۶۰۶	۳۷۵۲۷
۶	عملیات ترابری	۱۹۶	۲۳۳۹	۳۱۵۷۴
۷	آمار و گزارشات ترابری	۴۸	۷۶۵	۹۶۴۱
۸	الزامات ترابری	۱۴۰	۱۹۲۹	۳۰۶۷۷
۹	مدیریت و نظارت و پیگیری	۵۱	۱۰۰۲	۱۰۵۲۶
۱۰	درخواست و امریه ترابری	۱۹۷	۱۳۹۸	۱۹۷۰۵
۱۱	مدیریت طرح و برنامه	۷۰۶	۵۰۱۳	۸۴۲۲۳
۱۲	امور دامپزشکی	۱۹۱	۱۱۹۸	۱۴۷۳۶
۱۳	امور تغذیه	۱۰۴	۱۱۴۹	۱۷۹۲۰
۱۴	مهندسی و تاسیسات	۳۹۵	۲۶۰۶	۳۷۵۲۵
۱۵	مدیریت بهداشت و درمان	۴۶۳	۲۹۸۷	۳۸۵۲۶

17. Unadjusted Logical Transaction Point (ULTP)
18. Case Study
19. Code Generator
20. Variable Declatation
21. Comment

نمودار داده‌های بدست آمده در جدول۴ در نمودار ۱ نشان داده شده است.



نمودار۱. اندازه پروژه‌های نرم‌افزاری نمونه

ما به منظور نشان دادن کارایی روش پیشنهادی و کامل بودن آن، از روش‌های آماری و نگاشت مولفه‌ها جهت مقایسه روش پیشنهادی با دیگر روش‌های مطرح تخمین استفاده نمودیم.

۱۷. کارایی روش

پس از ثبت داده‌های جدول۴ در نرم‌افزار SPSS و انجام تحلیل همبستگی دو متغیری و رگرسیون خطی روی آن‌ها نتایج زیر به دست آمد که به تشریح آن می‌پردازیم.

الف- تحلیل همبستگی دو متغیری

در این تحلیل یک‌بار متغیر LOC را با متغیر UUCP و یک‌بار با متغیر ULTP مقایسه نمودیم که خروجی تحلیل در قالب دو جدول۵ و جدول۶ بدست آمد.

جدول ۵. همبستگی دو متغیری LOC با UUCP

		UUCP	LOC
UUCP	همبستگی پیرسون	۱	۰/۹۳۸ (**)
	Sig. (2-tailed)		۰/۰۰
	N	۱۵	۱۵
LOC	همبستگی پیرسون	۰/۹۳۸ (**)	۱
	Sig. (2-tailed)	۰/۰۰	
	N	۱۵	۱۵

جدول ۶. همبستگی دو متغیری LOC با ULTP

		ULTP	LOC
ULTP	همبستگی پیرسون	۱	۰/۹۸۷ (**)
	Sig. (2-tailed)		۰/۰۰
	N	۱۵	۱۵
LOC	همبستگی پیرسون	۰/۹۸۷ (**)	۱
	Sig. (2-tailed)	۰/۰۰	
	N	۱۵	۱۵

همان‌طور که مشاهده می‌شود در هر دو مورد همبستگی در سطح معنادار ۰/۰۱ (علامت **) وجود دارد. ولی ضریب همبستگی پیرسون[22] در متغیر ULTP نسبت به ضریب همبستگی پیرسون در متغیر UUCP به اندازه ۰/۰۴۹ به عدد یک نزدیک‌تر است و این مساله نشان می‌دهد که همبستگی ULTP به LOC به اندازه ۰/۰۴۹ از همبستگی UUCP به LOC بیشتر است.

ب- تحلیل رگرسیون خطی

در این تحلیل LOC را متغیر وابسته و UUCP و ULTP را متغیر مستقل در نظر گرفتیم که پس از انجام تحلیل، خروجی آن در قالب جدول۷ به دست آمد.

جدول ۷. ضرایب رگرسیون

مدل	ضریب استاندارد نشده		ضریب استاندارد شده	T	Sig.
	B	استاندارد خطا	بتا		
(ثابت)	-۲۸۷۴/۸۷۹	۱۶۴۹/۴۶۰		-۱/۷۴۳	۰۰/۱۰۷
UUCP	۲/۶۵۹	۱۵/۶۰۱	۰/۲۵	۰/۱۷۰	۰۰/۸۶۷
ULTP	۱۵/۸۱۹	۲/۳۵۹	۰/۹۶۴	۶/۷۰۷	۰۰/۰۰۰

با توجه به جدول۷ رابطه اول کمترین مجذورهای متداول[23] به صورت زیر نتیجه می‌شود:

LOC = -2874/88 + 2/659(UUCP) + 15/819(ULTP) رابطه (۳)

22. Pearson Correlation
23. Ordinary Least Squares(OLS)

با دقت در رابطه۳ مشخص می‌شود که میزان تاثیر غیرمستقیم ULTP روی LOC نسبت به تاثیر غیرمستقیم UUCP روی LOC به اندازه ۱۳/۱۶ بیشتر است.

۱۸. کامل بودن فاکتورها

جهت اثبات روش پیشنهادی(LTP)، فاکتورهای تعدادی از روش‌های مطرح تخمین را استخراج کرده و نشان دادیم که هر یک از این فاکتورها چگونه و توسط چه فاکتوری از روش پیشنهادی ما (LTP) پوشش داده می‌شود.

جدول ۸. نگاشت مولفه‌ها

R	روش		مولفه		مولفه LTP
۱	FPA	تحلیل درجه عملکردی	ILF	فایل منطقی داخلی	TSF1,TSF2
			EIF	فایل واسط خارجی	TSF1, TSF2
			EI	ورودی خارجی	TSF3
			EO	خروجی خارجی	TSF4
			EQ	پرس و جوی خارجی	TSF6
۲	Mark II	تحلیل درجه عملکردی نمره ۲	Wi	وزن عناصر داده‌ای ورودی	TSF3
			We	وزن عناصر داده‌ای ارجاعی	TSF1, TSF2
			Wo	وزن عناصر داده‌ای خروجی	TSF4
۳	Feature Point	تخمین مبتنی بر خصیصه	ILF	فایل منطقی داخلی	TSF1, TSF2
			EIF	فایل واسط خارجی	TSF1, TSF2
			EI	ورودی خارجی	TSF3
			EO	خروجی خارجی	TSF4
			EQ	پرس و جوی خارجی	TSF6
			Complexity	پیچیدگی الگوریتم	TSF5
۴	Object Point	تخمین مبتنی بر شیء	Screen	صفحه نمایش	TSF1-TSF3
			Report	گزارش	TSF4, TSF6
			3GL	ماژول های سه زبان تولید	TSF5
۵	UCP	درجه عملکرد مورد کاربرد	UAW	وزن تعدیل نشده عملگر	TSF1-TSF3
			UUCW	وزن تعدیل نشده موردکاربرد	TSF1-TSF6

۱۹. پیاده‌سازی

ما بر اساس روش پیشنهادی، نرم‌افزاری به نام ابزار تخمین درجه تراکنش منطقی یا LTP-SE[۲۴] پیاده‌سازی نمودیم. همان‌طور که در شکل۱ نشان داده شده است ورودی این ابزار، اطلاعات تراکنش‌های منطقی پروژه نرم‌افزاری به تفکیک موردکاربرد و خروجی آن، اندازه هر موردکاربرد یا اندازه کل پروژه می‌باشد.

شکل۱. شمای کلی نرم‌افزار

ابزار LTP-SE در محیط Net2005. تحت وب و با زبان برنامه نویسی C# پیاده‌سازی و از SQL 2005 به عنوان پایگاه داده استفاده می‌کند. نرم‌افزار LTP-SE دارای سه لایه واسط کاربری[۲۵]، لایه دسترسی به داده[۲۶] و پایگاه داده[۲۷] می‌باشد.

۲۰. نتیجه گیری و پیشنهادات

بر اساس تحلیل‌های آماری انجام شده نتیجه می‌گیریم روش LTP نسبت به روش UCP دقیق‌تر می‌باشد و بر خلاف

24. Logical Transaction Point -Software Estimation (LTP-SE)
25. User Interface Layer
26. Data Access Layer
27. DataBase Layer

روشUCP به ترسیم نمودار فعالیت (سناریو گام به گام) موردکاربرد نیاز ندارد. همچنین با توجه به جدول شماره۵ این روش کلیه عوامل موجود در روش‌های COCOMO، UCP و FP را پوشش داده و در مقایسه با روش FP برای الگوریتم‌ها (در قالب پیچیدگی محاسبات) ضریبی لحاظ نموده است.

این روش جهت تخمین در فازهای اولیه پروژه مناسب است و با روش‌های متداول امروزی مانند RUP سازگار است. قابلیت انعطاف و تطبیق زیادی دارد و بدلیل فرمال بودن روش، ابهام آن کم است.

خروجی و محصولات فرعی این روش در چارچوب فرایند توسعه نرم‌افزار (مثلا در ترسیم نمودار اشیا تجاری[۲۸]) قابل استفاده است.

درک سنجه درجه تراکنش منطقی یا LTP، ساده و آسان است. زیرا این سنجه مبتنی بر یک نظریه قابل اطمینان می‌باشد (اساس کار این روش بر چهار عملیات , Select Delete , Update , Add استوار است). از طرفی این سنجه به دلیل این که یک سنجه عملکردی است، برای برآورد مناسب است.

بنابراین پیشنهاد می‌شود پژوهشگران علاقه‌مند در رابطه با موارد زیر تحقیق نمایند..

الف- اندازه تعدیل‌نشده حاصل از روش تخمین درجه تراکنش منطقی (ULTP)، باید با اعمال عوامل فنی (TCF) و محیطی (EF) تعدیل شود.

ب- با آزمون این روش روی تعداد زیادی از پروژه‌های نرم‌افزاری انجام شده، میزان دقت روش از طریق تنظیم وزن عوامل موثر بر اندازه تراکنش‌های منطقی قابل افزایش است.

ج- این روش می‌بایست علاوه بر نرم‌افزارهای تراکنشی و داده محور، روی انواع دیگر نرم‌افزار مورد آزمون و ارزیابی قرار گیرد.

د- عامل بهره‌وری در روش‌های موجود تخمین، بیشتر بصورت تجربی و از طریق مقایسه با پروژه‌های انجام شده قبلی بدست می‌آید. ولی پیشنهاد می‌شود در کار آینده مختص روش LTP روشی جهت محاسبه عامل بهره‌وری ارایه شود.

در صورت انجام تحقیقات بالا می‌توان تلاش پروژه را با داشتن اندازه تعدیل شده پروژه و عامل بهره‌وری بدست آورد. هزینه و زمان پروژه نیز با استفاده از تلاش پروژه قابل محاسبه است.

منابع

۱. جلالی، ح. (۱۳۸۵). مدیریت پروژه‌های نرم‌افزاری. تهران: دانشگاه صنعتی مالک اشتر.

2.Adams, Scott. (2007). Requirements analysis and specification. CSE432. 3-4.

3.Carroll, Edward. (2002).Estimating Software Based on Use Case Points. Agilis Solutions, A Business Unit Of Hepieric, Inc, 257-258.

4.Coombs, Paul. (2003). IT Project Estimation- A Practical Guide of the Costing of Software. Cambridge University Press.

5.Jones, Capers. (2005). Software Estimating Methods for Large Projects. Crosstalk, 10.

11. Laird, Linda, M. (2006). The Limitations of Estimation. Published by the IEEE Computer Society. 42-44.

6.Laird, Linda, M. (2006). Software Measurement and Estimation : A Practical Approach. IEEE Computer Society / Wiley Partnership, 2.

7.Lakshmanan, D. (2007). Potentiality Of Usecase Point Method for Estimating the Size and Effort of Software Development Projects. Medwell

28. Business Object Model

Journals, 1565-1568.
8.Molokken, K & Jorgensen, M. (2004). A Survey on Software Estimation in the Norwegian Industry. IEEE Computer Society, 2.
9.Peters, k. (2009). Software Project Estimation",Simon Farser University.