

ارائه روشی برای رتبه‌بندی نیازمندی‌های کارکردی و غیرکارکردی با در نظر گرفتن تاثیر متقابل نیازها

امید زارع^۱، همایون موتمنی^۲

^۱ دانشجوی کارشناسی‌ارشد مهندسی فناوری اطلاعات، دانشگاه علوم و فنون مازندران، بابل، ایران
^۲ دانشیار گروه کامپیوتر، دانشکده کامپیوتر، دانشگاه آزاد اسلامی واحد ساری، ساری، ایران

چکیده

با توجه به محدودیت‌های زمان و بودجه در نظر گرفته شده، رتبه‌بندی نیازمندی‌های نرم‌افزار، امری حیاتی می‌باشد. خروجی مرحله رتبه‌بندی نیازمندی‌ها، یک لیست مرتب شده از نیازمندی‌ها می‌باشد که نشان دهنده این مطلب است که کدام نیازمندی‌ها می‌بایست در ابتدای فرایند ساخت نرم‌افزار، در نظر گرفته شوند. برای رسیدن به یک سیستم با کیفیت مطلوب، هر دو نوع نیازمندی کارکردی و غیرکارکردی، باید در خلال فرایند ساخت سیستم، در نظر گرفته شوند. اگرچه تاکنون تکنیک‌های زیادی برای رتبه‌بندی نیازمندی‌های سیستم‌های نرم‌افزاری ارائه شده است، اما هیچ‌کدام از آنها قادر نیستند که تأثیرات نیازمندی‌های غیرکارکردی و همچنین فاکتورهای اصلی تجاری، مانند سود، جریمه، هزینه و ریسک و زمان را بر روی رتبه‌بندی نیازمندی‌های کارکردی در نظر بگیرند. در این مقاله، تکنیکی ارائه می‌شود که قادر باشد هر دو نوع نیازمندی را به صورت همزمان رتبه‌بندی کرده و نقایص معرفی شده را نیز پوشش دهد. با ارزیابی و مقایسه نتایج بدست آمده در رتبه‌بندی نیازمندی‌ها توسط روش پیشنهادی با روش‌های قبلی مشخص شد که در نظر گرفتن نیازمندی‌های غیرعملیاتی در کنار فاکتورهای مهم تجاری در خلال عملیات رتبه‌بندی سبب افزایش دقت در رتبه‌بندی و متعاقباً نتایج بهتری خواهد شد.

واژه‌های کلیدی: نیازمندی‌های کارکردی، نیازمندی‌های غیرکارکردی، رتبه‌بندی نیازمندی‌ها، فاکتورهای تجاری.

۱- مقدمه

در پروژه‌های نرم افزاری به دلیل محدودیت منابع، نیازمند برنامه ریزی و مدیریت دقیق در استفاده از منابع هستیم. در همین راستا کارشناسان تولید نرم افزار با تحقیق بر روی بخش‌های مختلف مراحل ساخت نرم افزار در صدد اند تا بهترین تکنیک را برای مدیریت یک پروژه نرم افزاری انتخاب کنند. اولین و مهمترین گام در ساخت یک نرم افزار، نیازسنجی است که در فازهای اولیه ساخت یک نرم افزار صورت می‌گیرد. در این مرحله لازم است تا با تحصیل و تحلیل مناسب نیازها و رتبه‌بندی مناسب نیازها و تعیین ارجحیت ترتیب اجرای آن‌ها، تکنیک توسعه و میزان استفاده از منابع را طوری تنظیم کنیم تا در بهینه ترین حالت و در زمان خواسته شده پروژه را به نحو احسن به پایان برسانیم تا از دوباره کاری‌ها تا حد ممکن پرهیز کرده باشیم. در فرآیند توسعه نرم افزار از فاز شناسایی و استخراج نیازمندی‌ها به عنوان مهمترین و شاخص ترین گام توسعه نام برده می‌شود، چرا که عدم برخورداری از درک کامل و دقیق از نیازمندی‌های مشتری ضمن افزایش هزینه‌های توسعه نرم افزار تیم توسعه را در رسیدن به محصولی مطابق با نیازهای کاربر غافل خواهد ساخت. تاکنون تلاش‌های مهم و تأثیرگذاری برای رتبه‌بندی نیازمندی‌ها انجام شده است. به غیر از تکنیک^۱ IPA معرفی شده توسط (دباغ و لی، ۲۰۱۴) غالب تکنیک‌های ارائه شده، تنها قادر هستند که نیازمندی‌های کاربردی و غیر کاربردی را به صورت مجزا رتبه‌بندی کنند که طبیعتاً در این‌گونه تکنیک‌ها، اهمیت نیازمندی‌های غیر کاربردی بر روی رتبه‌بندی نیازمندی‌های کاربردی، بی‌تأثیر خواهد بود. همچنین در تکنیک IPA نیز معیار رتبه‌بندی نیازمندی‌های کاربردی، تنها رتبه و تأثیر نیازمندی‌های غیر کاربردی بر روی نیازمندی‌های کاربردی است، در صورتی که عوامل تجاری که از عوامل بیرونی تأثیرگذار بر روی رتبه‌بندی نیازمندی‌های ساخت یک نرم‌افزار است نادیده گرفته شده است. از جمله عوامل تجاری تأثیرگذار در مشخص سازی میزان اهمیت نیازها می‌توان به ریسک، هزینه، جریمه، سود، زمان و غیره اشاره کرد. با در نظر گرفتن تمامی عوامل اثرگذار بر تصمیم می‌توان تکنیکی جامع طراحی کرد که کلیه موارد ذکر شده را در نظر بگیرد تا رتبه‌بندی نیازمندی‌ها در مطلوب‌ترین حالت صورت بگیرد و نیازهای ذینفعان به طور کامل در زمان مشخص شده ارضا شود.

هدف اصلی این پژوهش، ارائه تکنیکی است که بتواند هر دو نوع نیازمندی را همزمان و با در نظر گرفتن تأثیرات نیازمندی‌های غیر کاربردی بر روی نیازمندی‌های کاربردی، رتبه‌بندی کرده و تأثیر فاکتورهای اصلی تجاری مانند ریسک، هزینه پیاده‌سازی، جریمه، سود و زمان را در خلال فرایند رتبه‌بندی، در نظر بگیرد.

۲- ادبیات تحقیق

هدف نهایی از ساخت هر سیستم نرم‌افزاری، برآورده کردن نیازهای ذینفعان می‌باشد. یکی از مهمترین مراحل در پروژه‌های نرم‌افزاری و نقطه شروع هر پروژه نرم‌افزاری، مرحله مهندسی نیازمندی‌هاست. در این مرحله، تمام جوانب و نیازمندی‌های سیستم مورد نظر مشتری، مورد تجزیه و تحلیل قرار گرفته و مشخصات آن، تعیین می‌شوند. مهندسی نیازمندی‌ها، یک مرحله بسیار مهم و ویژه، در فرایند ساخت نرم‌افزار به‌شمار می‌رود که وجود خطا در این مرحله، باعث ایجاد مشکلاتی فراوان در سیستم‌های نرم‌افزاری خواهد شد. خروجی این مرحله در تمامی مراحل ساخت نرم‌افزار مورد استفاده قرار خواهد گرفت و در طی مراحل طراحی و پیاده‌سازی سیستم، به‌طور مستمر به مستندات این مرحله مراجعه خواهد شد. استخراج ضعیف نیازمندی‌ها می‌تواند هزینه زیادی را به پروژه نرم‌افزاری تحمیل کرده و یا منجر به شکست پروژه نرم‌افزاری شود. موفقیت در این مرحله به انتخاب تکنیک مناسب برای استخراج نیازمندی‌ها وابسته است. انتخاب نامناسب تکنیک تحصیل نیازمندی‌ها، منجر به پنهان ماندن نیازمندی‌ها، اولویت‌ها و مسائلی خواهد شد که تحلیل گران از آن اطلاع نداشته و ذینفعان هم نمی‌توانند به آسانی آنها را بیان کنند؛ بنابراین، انتخاب ذینفعان مناسب و مرتبط با موضوع، یکی از عوامل موفقیت در مهندسی نیازمندی‌هاست. از این رو در ادامه به شرح تکنیک‌های مختلف استخراج نیازمندی و مفاهیم اولیه در این راستا می‌پردازیم.

¹ Integrated Prioritization Approach

۱-۲- تعریف نیازمندی‌ها در مهندسی نرم‌افزار

استاندارد IEEE 610، «نیازمندی» را این‌گونه تعریف می‌کند: ۱- یک شرط یا قابلیت مورد نیاز توسط کاربر، برای حل یک مسأله یا دستیابی به یک هدف. ۲- یک شرط یا قابلیت که باید در سیستم وجود داشته باشد تا یک قرارداد، استاندارد، توصیف یا سایر مستندات رسمی برآورده گردد.

در ^۱SWEBOK نیازمندی به این صورت تعریف شده است (بورک و همکاران، ۲۰۱۲): یک ویژگی که باید توسط یک سیستم ساخته شده، برای حل یک مسئله خاص ارائه شود. تعریف ^۲RUP از نیازمندی‌ها بدین شرح است (کروتچن، ۲۰۰۴): یک «نیازمندی» به‌عنوان یک شرط یا قابلیت تعریف می‌شود که یک سیستم باید مطابق آن کار کند و بالاخره لبریح ^۳ نیازمندی را یک عبارت دربارهٔ سیستم پیشنهادی که توافق کلیه ذینفعان باید برای آن اخذ شود و در جهت رفع مشکل کاربر باشد تعریف می‌کند (سومروایل، ۲۰۱۱). در کل دو نوع اصلی نیازمندی در پروژه‌های نرم‌افزاری وجود دارد.

نیازمندی‌های مربوط به توانایی یک سیستم^۴: امکانات اصلی و اساسی که توسط سیستم برای پاسخ به خواسته‌های ذینفعان و مشتری فراهم می‌شود که «نیازمندی‌های کارکردی»^۵ نیز خوانده می‌شوند. در حقیقت، نرم‌افزار، برای برآورده کردن این نیازمندی‌ها به وجود آمده است و امکانات خاصی را تعریف می‌کنند که باید توسط سیستم نرم‌افزاری ارائه شوند.

نیازمندی‌های کیفی^۶: صفات نشان‌دهندهٔ سیستم نرم‌افزاری که مستقیماً با سیستم رابطه داشته و از طریق رتبه‌بندی صفات کیفی سیستم، قابل اندازه‌گیری هستند که به آنها «نیازمندی‌های غیرکارکردی»^۷ نیز اطلاق می‌شود. همان‌طور که اشاره شد، نیازمندی‌های کیفی به وسیلهٔ مشخصه‌های کیفی که دارند اندازه‌گیری می‌شوند، لذا برای مقایسهٔ این نیازمندی‌ها، روی ویژگی‌های کیفی آنها متمرکز می‌شوند. در جدول زیر مقایسه‌ای میان نیازمندی‌های کارکردی و غیرکارکردی صورت گرفته است.

جدول شماره ۱. مقایسه نیازمندی‌های کارکردی و غیرکارکردی

نیازمندی‌های کارکردی	نیازمندی‌های غیرکارکردی	ردیف
با رابطه بین محرک‌ها، عکس‌العمل‌ها مشخص می‌شوند	با خصوصیات کیفی خود مشخص می‌شوند	۱
به‌عنوان معیار «تکمیل سیستم» به کار می‌روند	به‌عنوان معیار «موازنه» به کار می‌روند	۲
تنها روی یک واحد مسئولیت دارند	روی همهٔ واحدها مسئولیت دارند	۳
قبل از تحویل، تأیید می‌شوند	بعد از تأیید، مورد سنجش قرار می‌گیرند	۴
تأیید شدن آنها با یک سلسله تست‌های مجزا انجام می‌گیرد	در یک دورهٔ زمانی مورد سنجش قرار می‌گیرند	۵
مثال در یک سیستم نرم‌افزاری: قابلیت خرید آنلاین	مثال: سرعت اجرای سیستم	۶

۲-۲- ذینفعان پروژه

افرادی که در فرایند استخراج و تحلیل نیازمندی‌ها درگیر می‌شوند «ذینفعان» پروژه شناخته می‌شوند. یکی از مهمترین وظایف مهندسی نیازمندی‌ها، شناسایی ذینفعان از طریق تعیین افرادی است که پروژه را تحت تأثیر قرار می‌دهند و یا پروژه بر روی کار آنها تأثیر می‌گذارد. ذینفعان یک پروژه نرم‌افزاری را می‌توان در یکی از این چهار گروه دسته‌بندی نمود: مالکین-

^۱Software Engineering Body Of Knowledge

^۲Rational Unified Process

^۳Lethbridge

^۴Capability Requirements

^۵Functional Requirements

^۶Quality Requirements

^۷Non-Functional Requirements

مدیران- مشتریان و کاربران. در فرایند مهندسی نیازمندی‌ها، ذینفعان نقش حیاتی را ایفا می‌کنند و موفقیت در این فرایند، به مشارکت فعال ذینفعان وابسته است. شکل ۱ مثالی از ذینفعان یک پروژه را به تصویر کشیده است.

به دلایلی که در ادامه بیان خواهد شد، مشارکت ذینفعان نقش مهمی در فرایند جمع‌آوری و تحلیل نیازمندی‌ها خواهند داشت.

- ذینفعان، نیازمندی‌های خود را با روش و ادبیات گوناگونی بیان می‌کنند و یا به درستی از نیازمندی‌های خود با خبر نیستند.
- عوامل گوناگونی مانند عوامل فرهنگی و سیاسی، ممکن است، نیازمندی‌ها را تحت تأثیر قرار دهند.
- نیازمندی‌ها ممکن است در طول فرایند مهندسی نیازمندی‌ها دچار تغییر شوند.



شکل ۱. مثالی از ذینفعان یک پروژه

۲-۳- مهندسی نیازمندی‌ها

مهندسی نیازمندی‌ها، مکانیسم مناسبی را جهت درک نیازمندی‌های مشتریان، تحلیل نیازمندی‌ها، امکان‌سنجی اجرای پروژه، مذاکره در مورد راه‌حل‌ها، اعتبارسنجی مشخصه نیازمندی‌ها و مدیریت نیازمندی‌ها، فراهم می‌کند. فعالیت‌های مهندسی نیازمندی‌ها، منجر به دستیابی به مشخصه کارکردی نرم‌افزار شده و محدودیت‌هایی را مشخص می‌کند که نرم‌افزار باید داشته باشد. دونالد ریفر^۱ فرایند مهندسی نیازمندی‌ها را به این صورت توصیف می‌کند (ریفر، ۲۰۰۶): «مهندسی نیازمندی‌ها، استفاده سیستماتیک از اصول اثبات شده، تکنیک‌ها، زبان‌ها و ابزارهایی برای تحلیل، مستندسازی و تکامل نیازمندی‌های کاربر و مشخصه رفتار خارجی سیستم، به منظور برآورده نمودن آن نیازمندی است.» مهندسی نیازمندی‌ها شامل هفت وظیفه می‌باشد (پرسمن، ۲۰۰۵) که عبارتند از: ۱- آغاز^۲ ۲- تحصیل نیازمندی‌ها^۳ ۳- طبقه‌بندی نیازمندی‌ها^۴ ۴- مذاکره با کاربر^۵ ۵-

¹ Donald Reifer

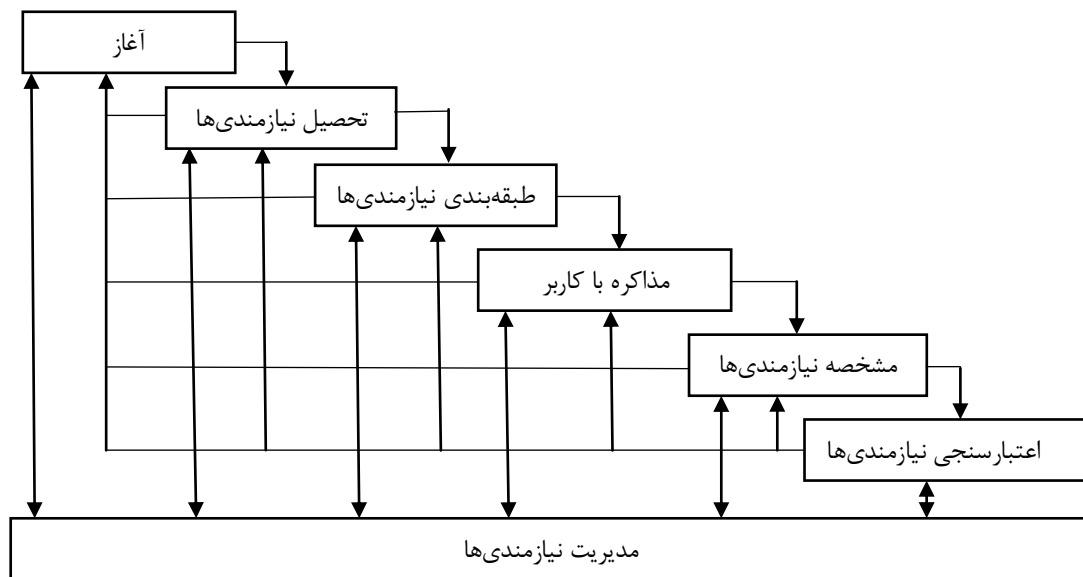
² Inception

³ Elicitation

⁴ Elaboration

⁵ Negotiation

مشخصه نیازمندی‌ها^۱ ۶- اعتبارسنجی نیازمندی‌ها^۲ ۷- مدیریت نیازمندی‌ها^۳. با توجه به نوع پروژه ممکن است برخی از فعالیت‌ها پی‌درپی و برخی به صورت موازی انجام شوند. همه این مراحل تلاش دارند که نیازمندی‌ها به درستی شناسایی شده و سیستم بر مبنای نیازمندی‌های درست پیاده‌سازی شود. با استفاده از این مراحل، زیربنای محکمی برای مراحل طراحی و پیاده‌سازی نرم‌افزار، به وجود خواهد آمد و این زیربنا در مشخصه نیازمندی‌ها مستند شده تا مبنای مراحل بعدی نرم‌افزار قرار گیرد؛ بنابراین، همه این هفت مرحله برای رسیدن به این دو هدف اصلی است: ۱- فهم درست نیازمندی‌های کاربر ۲- ایجاد یک مبنای محکم به منظور استفاده در مراحل طراحی و پیاده‌سازی نرم‌افزار. در شکل زیر وظایف مهندسی نیازمندی‌ها و ارتباط این مراحل با یکدیگر نشان داده شده است.



شکل ۲. نقش مهندسی نیازمندی‌ها

۲-۴- رتبه‌بندی نیازمندی‌ها

رتبه‌بندی، مهارتی ضروری برای استفاده از تلاش و قابلیت‌های فردی و تیمی است. این موضوع، به‌ویژه هنگامی اهمیت پیدا می‌کند که زمان محدود و تقاضاها نامحدود هستند. در این حالت، رتبه‌بندی می‌تواند در تخصیص زمان به نیازمندی‌های مهم، آزاد کردن زمان تخصیص یافته به نیازمندی‌های دارای اهمیت کمتر و حتی از قلم انداختن برخی نیازمندی‌ها، کمک کند. با توجه به محدودیت زمان، هزینه و منابع در دسترس، برای ساخت یک سیستم مطلوب، ذینفعان می‌بایست درک درستی از نیازمندی‌های مهم سیستم مورد نظر داشته باشند (فنکلشتاین، ۲۰۰۹). در این بخش، فاکتورهای مهمی که سازندگان سیستم‌های نرم‌افزاری، برای رتبه‌بندی نیازمندی‌ها، مد نظر قرار می‌دهند، معرفی می‌گردد.

اهمیت^۴ (سود): ذینفعان باید نیازمندی‌ها را بر اساس اهمیتی که برای سیستم دارند، رتبه‌بندی کنند. بر این اساس، مبنای رتبه‌بندی «اهمیت» تعیین می‌شود. از طرف دیگر، مبنای «اهمیت» می‌تواند یک مفهومی چند بعدی باشد که شدیداً به

¹ Specification

² Validation

³ Requirement Management

⁴ Importance

دیدگاه ذینفعان بستگی دارد. برای مثال، فاکتور «اهمیت» می‌تواند ضرورت انجام و پیاده‌سازی یک نیازمندی برای یک طراح محصول، یا اهمیت راهبردی یک نیازمندی برای یک شرکت باشد. لذا بسیار مهم است که مشخص شود چه نوع اهمیتی را افراد ذینفع برای رتبه‌بندی در مورد خود به کار ببرند.

جریمه^۱ - ناشی از عدم توجه به برخی از نیازمندی‌ها: ممکن است اگر یک نیازمندی برآورده نشود، جریمه‌ای برای آن در نظر گرفته شود. جریمه نقطه مقابل اهمیت نیست. برای مثال، کوتاهی در رعایت کردن یک استاندارد ممکن است جریمه بسیار بالایی را تحمیل کند؛ حتی اگر آن نیازمندی، اهمیت کمی برای مشتری داشته باشد.

هزینه و ریسک^۲: هزینه و ریسک پیاده‌سازی یک نیازمندی، از جمله عواملی است که باید در خلال فرایند رتبه‌بندی نیازمندی‌ها در نظر گرفته شود که این فاکتورها، توسط مهندسیین سازنده نرم‌افزار، مشخص می‌شوند.

زمان^۳: مدت زمان لازم برای پیاده‌سازی یک نیازمندی و این فاکتور نیز بدلیل نیاز به تسلط بر مهارت‌های مهندسی نرم‌افزار، توسط مهندسیین سازنده نرم‌افزار مشخص خواهد شد.

۳- تحقیقات پیشین

تمامی تکنیک‌های در نظر گرفته شده برای رتبه‌بندی نیازمندی‌ها، نیازمندی‌های کارکردی و غیرکارکردی را به‌صورت جداگانه رتبه‌بندی می‌کنند. تنها تکنیک IPA قادر است که همزمان هر دو نوع نیازمندی را رتبه‌بندی کند. مزیت تکنیک IPA نسبت به دیگر تکنیک‌ها در این است که چارچوبی برای رتبه‌بندی نیازمندی‌ها فراهم می‌کند که هر دو نوع نیازمندی کارکردی و غیرکارکردی با یکدیگر در طول فرایند رتبه‌بندی مرتبط می‌شوند و همچنین نیازمندی‌های غیرکارکردی رتبه‌بندی شده بر روی رتبه‌بندی نیازمندی‌های کارکردی تأثیرگذار خواهند بود، درحالی‌که در سایر تکنیک‌ها، اولویت نیازمندی‌های غیرکارکردی بر روی رتبه‌بندی نیازمندی‌های کارکردی در نظر گرفته نمی‌شد.

در تکنیک IPA، معیار رتبه‌بندی نیازمندی‌های کارکردی، تنها اولویت و تأثیر نیازمندی‌های غیرکارکردی بر روی نیازمندی‌های کارکردی است، در حالی که در برخی شرایط این امکان وجود دارد که پیاده‌سازی یک نیازمندی کارکردی برای ذینفع از اولویت بسیار بالایی برخوردار باشد (سود زیادی برای ذینفع داشته باشد) و یا عدم پیاده‌سازی آن مشکلات زیادی برای سیستم به وجود آورد (جریمه نسبی آن بالا باشد) ولی اهمیت هر نیازمندی غیرکارکردی در رابطه با آن نیازمندی کارکردی از اهمیت بالایی برای کاربر برخوردار نباشد، در این صورت، نیازمندی مورد نظر در این تکنیک، پس از انجام محاسبات، از اولویت پایینی برخوردار خواهد بود. همچنین ممکن است هزینه و ریسک پیاده‌سازی آن نیازمندی که از اولویت بالایی در این تکنیک برخوردار شده است بسیار بالا باشد، در این صورت این امکان وجود دارد که ذینفع از پیاده‌سازی آن نیازمندی منصرف شود. در حقیقت مزایای تکنیک VOP و Wigers Method را می‌توان به‌عنوان نقصی برای تکنیک IPA دانست. در جدول ۲ مقایسه کلی درباره ویژگی‌های تکنیک‌های رتبه‌بندی معرفی شده، انجام شده است.

¹ Penalty

² Risk and Cost

³ Time

جدول ۲. مقایسه تکنیک‌های مختلف رتبه‌بندی نیازمندی‌ها.

ردیف	نویسندگان و سال انتشار مقاله	نام تکنیک	رتبه‌بندی نیازمندی‌های کارکردی	رتبه‌بندی نیازمندی‌های غیرکارکردی	در نظر گرفتن تاثیر نیازمندی‌های غیرکارکردی بر روی نیازمندی‌های کارکردی	در نظر گرفتن فاکتورهای اصلی تجاری برای رتبه‌بندی	همکاری تیم سازنده و ذینفع
۱	اسمیت و همکاران ^۱ ۱۹۹۷	BST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۲	ویگرو همکاران ^۲ ۱۹۹۹	Wigers method	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۳	کارلسون و همکاران ^۳ ۲۰۰۱	AHP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۴	لفینول و ویدریج ^۴ ۲۰۰۳	CV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۵	آزار و همکاران ^۵ ۲۰۰۷	VOP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۶	دباغ و لی ^۶ ۲۰۱۳	CAPSQA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
۷	دباغ و لی ۲۰۱۴	IPA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

۴- روش پیشنهادی

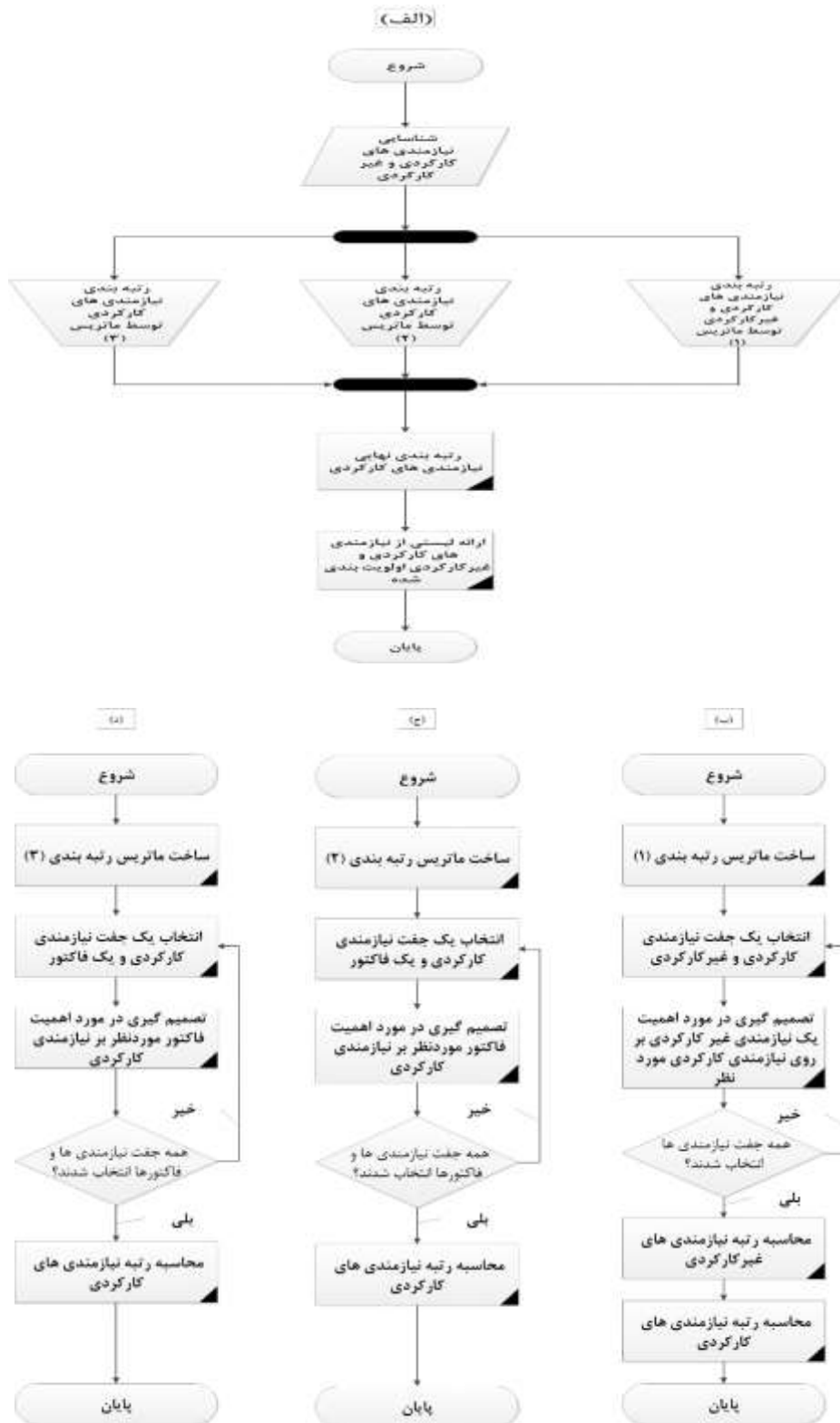
در فرایند رتبه‌بندی نیازمندی‌های یک سیستم، همکاری تیم سازنده و ذینفع، امری ضروری است. در روش ارائه شده در این مقاله، رتبه‌بندی در سه بعد مورد بررسی قرار می‌گیرد که دو بعد آن را ذینفع مشخص کرده و یک بعد دیگر را مهندسین سازنده نرم‌افزار مشخص می‌کنند. روش مورد نظر برای رتبه‌بندی نیازمندی‌های کارکردی و غیرکارکردی در این مقاله، شامل دوازده مرحله می‌باشد که در جدول ۳ مراحل گوناگون این روش نشان داده شده است.

¹ Smith et al.² Wiger et al.³ Karlsson et al.⁴ Leffingwell and Widrig⁵ Azar et al.⁶ Dabbagh and Lee.

جدول ۳. مراحل روش پیشنهادی برای رتبه‌بندی نیازمندی‌های کارکردی و غیر کارکردی.

مرحله	توضیح
۱	تعیین نیازمندی‌های کارکردی و غیر کارکردی
۲	ایجاد ماتریس‌های سه‌گانه تصمیم‌گیری
۳	تصمیم‌گیری در مورد درجه اهمیت هر نیازمندی غیر کارکردی در رابطه با نیازمندی‌های کارکردی، (توسط ذینفع)
۴	تصمیم‌گیری در مورد تأثیر فاکتورهای سود و جریمه بر روی نیازمندی‌های کارکردی، (توسط ذینفع)
۵	تصمیم‌گیری در مورد تأثیر فاکتورهای هزینه و ریسک و زمان بر روی نیازمندی‌های کارکردی، (توسط سازنده)
۶	محاسبه میانگین وزنی و رتبه‌بندی نهایی نیازمندی‌های غیر کارکردی در ماتریس یک با استفاده از تکنیک‌های Alpha Cut و Triangular Fuzzy Numbers
۷	محاسبه میانگین وزنی فاکتورهای سود و جریمه در ماتریس دو، با استفاده از تکنیک‌های Alpha Cut و Triangular Fuzzy Numbers
۸	محاسبه میانگین وزنی فاکتورهای هزینه و ریسک و زمان در ماتریس سه، با استفاده از تکنیک‌های Alpha Cut و Triangular Fuzzy Numbers
۹	محاسبه رتبه نیازمندی‌های کارکردی در ماتریس D با استفاده از Weighted Average Decision Matrix و وزنی که در مرحله ششم محاسبه شد.
۱۰	محاسبه رتبه نیازمندی‌های کارکردی در ماتریس D' با استفاده از Weighted Average Decision Matrix و وزنی که در مرحله هفتم محاسبه شد.
۱۱	محاسبه رتبه نیازمندی‌های کارکردی در ماتریس D'' با استفاده از Weighted Average Decision Matrix و وزنی که در مرحله هشتم محاسبه شد.
۱۲	محاسبه رتبه نهایی نیازمندی‌های کارکردی

همچنین برا شفافیت بیشتر روش پیشنهادی برای رتبه بندی نیازمندی‌های کارکردی در شکل ۳ به ترتیب الف، ب، ج، د شرح داده شده است.



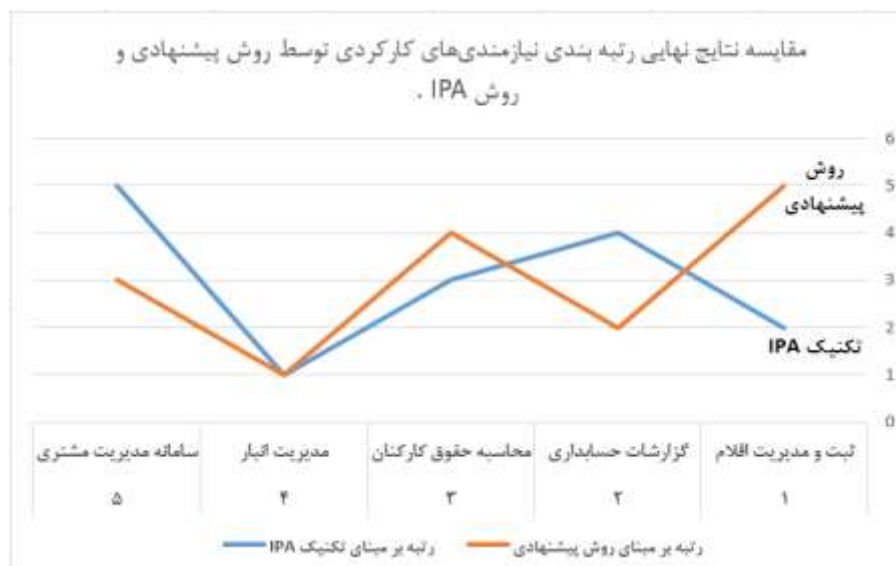
شکل ۳. دیاگرام مراحل روش پیشنهادی برای رتبه بندی نیازمندی‌های کاربردی و غیر کاربردی

۵- یافته‌های تحقیق

پس از پیاده سازی روش پیشنهادی توسط نرم افزار طراحی شده و ذکر یک مطالعه موردی نرم افزار حسابداری مشخص شد که تفاوت در رتبه بندی نهایی نیازمندی های کارکردی بیانگر این مطلب است که فاکتورهای اصلی تجاری مانند سود، جریمه، هزینه و ریسک و زمان پیاده سازی، در فرایند رتبه بندی نیازمندی ها، نقش مهمی را ایفا می کنند و عدم در نظر گرفتن این فاکتورها در خلال فرایند رتبه بندی، باعث ایجاد نتایجی متفاوت خواهد شد. جدول ۴ و شکل ۴ نتایج حاصل از تکنیک IPA و همچنین روش پیشنهادی را نشان می دهد. همانطور که مشاهده می شود، رتبه نیازمندی های کارکردی نرم افزار حسابداری، در روش پیشنهادی، متفاوت با تکنیک IPA (که تنها تاثیر نیازمندی های غیر کارکردی بر روی نیازمندی های کارکردی را در نظر می گیرد) خواهد بود.

جدول ۴. مقایسه رتبه نیازمندی های کارکردی نرم افزار حسابداری با توجه به روش ارائه شده و تکنیک IPA.

رتبه بر مبنای روش پیشنهادی	رتبه بر مبنای تکنیک IPA	نیازمندی کارکردی	ردیف
۵	۲	ثبت و مدیریت اقلام	۱
۲	۴	گزارشات حسابداری	۲
۴	۳	محاسبه حقوق کارکنان	۳
۱	۱	مدیریت انبار	۴
۳	۵	سامانه مدیریت مشتری	۵



شکل ۴. مقایسه نتایج نهایی رتبه بندی نیازمندی های کارکردی توسط روش پیشنهادی و روش IPA

۶- بحث و نتیجه‌گیری

با مقایسه نتایج ستون «رتبه بر مبنای تکنیک IPA» در جدول شماره ۴ (که رتبه نیازمندی‌های کارکردی را بر مبنای تکنیک IPA نشان می‌دهد) و همچنین نتایج ستون «رتبه بر مبنای روش پیشنهادی» (که رتبه نیازمندی‌های کارکردی را بر اساس روش پیشنهادی در این مقاله نشان می‌دهد) تفاوت در رتبه‌بندی نهایی نیازمندی‌های کارکردی نشان دهنده این مطلب است که علاوه بر تاثیر نیازمندی‌های غیرکارکردی بر روی نیازمندی‌های کارکردی، فاکتورهای اصلی تجاری مانند سود، جریمه، هزینه و ریسک و زمان پیاده‌سازی، در فرایند رتبه‌بندی نیازمندی‌ها، نقش مهمی را ایفا می‌کنند و عدم در نظر گرفتن این فاکتورها در خلال فرایند رتبه‌بندی، باعث ایجاد نتایجی متفاوت خواهد شد. همچنین در روش پیشنهادی، همکاری ذینفع و سازنده نرم‌افزار در بدست آمدن نتایج نهایی قابل اعتمادتر و دقیق‌تر کمک بسزایی کرده است. از روش پیشنهادی این مقاله می‌توانیم برای رتبه بندی نیازمندی‌های نرم‌افزارهای بزرگ و حساس استفاده کنیم (آنجایی که نیاز به حساسیت بالا در رتبه بندی داریم یا هزینه‌های ساخت نرم‌افزار بسیار زیاد است. در ادامه و در جدول شماره ۵ مقایسه کلی درباره روش معرفی شده در این مقاله، با سایر تکنیک‌های موجود برای رتبه‌بندی نیازمندی‌ها، انجام شده است.

جدول ۵. مقایسه‌ای کلی بین روش ارائه شده و سایر روش‌ها.

ردیف	نام روش	رتبه‌بندی نیازمندی‌های کارکردی	رتبه‌بندی نیازمندی‌های غیرکارکردی	در نظر گرفتن تاثیر نیازمندی‌های غیرکارکردی بر روی نیازمندی‌های کارکردی	در نظر گرفتن فاکتورهای اصلی تجاری برای رتبه‌بندی	همکاری تیم سازنده و ذینفع
۱	BST	✓	×	×	×	×
۲	Wiger's Method	✓	✓	×	✓	✓
۳	AHP	✓	✓	×	×	×
۴	CV	✓	✓	×	×	×
۵	VOP	✓	×	×	✓	✓
۶	CAPSQA	×	✓	×	×	✓
۷	IPA	✓	✓	✓	×	×
۸	روش پیشنهادی	✓	✓	✓	✓	✓

منابع

۱. راد ع، طالبی د و جلالی ح. (۱۳۸۷). ارائه مدلی برای رتبه‌بندی نیازمندی‌ها در سیستم‌های نرم‌افزاری و اطلاعاتی با استفاده از تکنیک AHP. چشم‌انداز مدیریت بازرگانی، شماره ۲۸، ص ۱۵-۱۸۳.
2. Bourque F, Robert J and Lavoie A (2012). "Guide to the software engineering body of knowledge and the software engineering education knowledge" in software technology and engineering practice.
3. Dabbagh M. and Lee S.P (2014). "An Approach for Integrating the Prioritization of Functional and Nonfunctional Requirements" *The Scientific World Journal*.
4. Finkelstein A, Harman M, Mansouri S, Ren J and Zhang Y. (2009) "A Search Based Approach to Fairness Analysis in Requirement Assignments to Aid Negotiation, Mediation and Decision Making" *Requir. Eng.* 14, pp. 231-245.
5. K. Rinkevics, R. Torkar (2012) "Equality in cumulative voting: A systematic review with an improvement proposal", *Information and Software Technology*, Elsevier.
6. Kruchten P,(2004). the rational unified process: An introduction: addison-Wesley.
7. P. Berander, P. Jonsson, (2006) A goal question metric based approach for efficient measurement framework definition, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ACM, New York, NY, USA, pp. 316-325.
8. Presman R. (2005) "Software Engineering: a Practitioner's Approach" McGraw-Hill International Edition.
9. R. Feldt, R. Torkar, E. Ahmad, B. Raza,(2010). Challenges with software verification and validation activities in the space industry, in: Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation, IEEE Computer Society, Washington, DC, USA, , pp. 225-234.
10. Reifer D. (2006) "Software Management" 7th Edition ed., Wiley.
11. Saaty (1995).T "Decision Making for Leaders" *University of Pittsburgh*.
12. Saaty T . (1994)"Fundamentals of the Analytical Hierarchy Process" *RWS Publications*.
13. Sommerville,(2011) .software engineering, 9th edition ed. USA: Pearson.
14. View (1990). "IEEE-Std '610'-IEEE standard glossary of software engineering terminology" in IEEE std,ed.

A Method for Prioritization of Functional & Non-functional Requirements Considering Requirements interaction

Omid Zare¹, Dr. Homayun Motameni²

¹Student, Department of Computer, Oloomfonoon University of Mazandaran, Iran

²Associate professor, Department of Computer, Islamic Azad University, Sari branch, Iran

Abstract

Due to the budgetary deadlines and time to market constraints, it is essential to prioritize software requirements. The outcome of requirements prioritization is an ordering of requirements, which need to be considered first during the software development process. To achieve a high quality software system, both functional and nonfunctional requirements must be taken into consideration during the prioritization process. Although several methods have already been proposed to prioritize the requirement of software systems, but none of them are able to consider the influences of non-functional requirement as well as the main factors of business, such as benefit, penalties, costs, risk and time on the prioritization of functional requirement. In this project, a method is provided that can prioritize both the functional and non-functional requirements at the same time and have covered implicit weaknesses. By evaluating and comparing the results obtained from proposed approach with the former approaches, it has proved that considering the influence of non-functional requirements as well as main business factors through the prioritization process of functional requirements, has led to further accuracy and better outcomes.

Keywords: Functional Requirements, non-Functional Requirements, Requirements Prioritization, Main Business factors
