پنجمین کنفرانس سیستمهای فازی ایران
هفدهم الی نوزدهم شهریورماه ۱۳۸۳

روش جدید سنجش کارایی فرایندهای نرم افزاری بر اساس معیارهای چند وضعیتی فازی

میرعلی سیدی[۱]، فریدون شمس[۱]، محمد تشنه لب[۲]
دانشکده فنی واحد علوم و تحقیقات تهران ـ گروه کامپیوتر ـ نرم‌افزار
Maseyyedi2002@yahoo.com

چکیده

در این مقاله به ارائه روشی نوین و کاربردی جهت اندازه گیری میزان رشد یافتگی و کیفیت فرآیندهای نرم افزاری پرداخته شده است. این روش بر اساس مدل قابلیت رشد نرم افزار (SW-CMM)[۳] و یک مدل چند سطحی استنتاج فازی طراحی و به صورت یک ابزار سنجش و تحلیل پیاده سازی شده است. از مهمترین ویژگیهای این روش می توان به سادگی بکارگیری، دقت اندازه گیری، کمی بودن ارزیابی ، قابلیت مقایسه پذیری و تحلیل و قابلیت پیاده سازی مدل مربوطه بصورت یک ابزار اتوماتیک سنجش اشاره کرد.

واژه های کلیدی:مدل قابلیت رشد نرم افزار-روش هدف/پرسش/اندازه-حوزه های فرایندی کلیدی-سیستم فازی

۱- مقدمه

بهره وری و کیفیت تولیدنرم افزار بر سه محور نیروی انسانی،فن آوری و فرآیندهای نرم افزاری استوار است [۱] . مدل ها و روش های مدیریت و رشد نیروی انسانی از سابقه طولانی برخوردار بوده و روش های تعریف شده ای نیز در این زمینه موجود است. از سوی دیگر محور فن آوری و ابزارهای تولید نرم افزار نیز با سرعت مناسب در حال رشد بوده و مسیر معینی را طی می کند. در این میان فرآیندهای نرم افزاری اساسی ترین زمینه بهبود کیفیت و بهره وری بشمار می روند و بر این اساس مدل های استاندارد فرآیندهای نرم افزاری بطور مداوم در حال رشد هستند. این مقاله بر روش اندازه گیری میزان رشد یافتگی و کیفیت فرآیندهای نرم افزاری در یک سازمان نرم افزاری تمرکز دارد. روش مذکور که به عنوان نمونه برای حوزه فرآیندی مدیریت نیازمندیها از مدل SW-CMM پیاده سازی شده است،به گونه ای تنظیم گردیده که برای تمام حوزه های فرآیندی این مدل قابل تعمیم باشد.

تا به حال مطالعاتی روی سنجش نرم‌افزار بر اساس مدل SW-CMM انجام شده است که از مهمترین آنها می‌توان به گزارش تهیه شده توسط گروه Baumart & Mcwhinnel [۲] شامل مجموعهٔ شاخص‌هایی (معیارهای مرکب) که سازگار با فعالیتهای سنجش‌و تحلیل در SW-CMM نیز است، اشاره نمود [۳]. این شاخص‌ها بر مبنای خصوصیات کیفی محصول

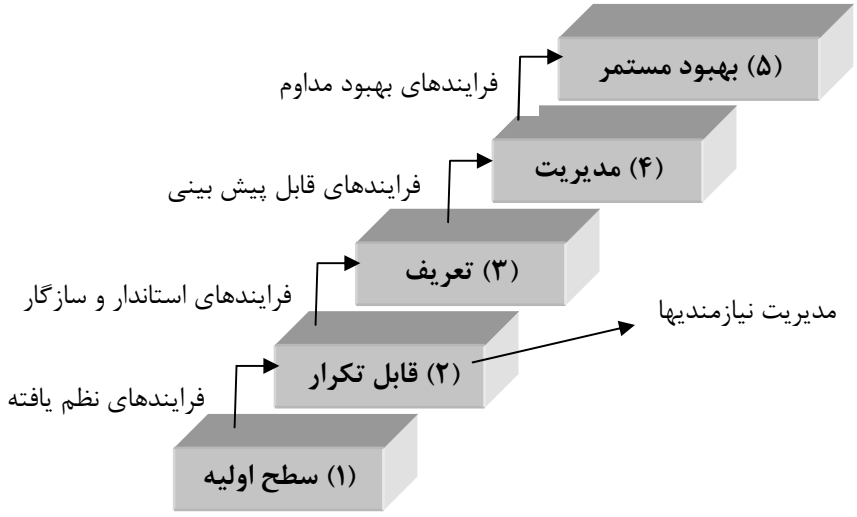
۱- دانشکده فنی واحد علوم و تحقیقات تهران ـ گروه کامپیوتر ـ نرم‌افزار
۲- دانشگاه خواجه نصیرالدین طوسی ـ دانشکده برق

3- Software Capability Maturity Model

طبقه‌بندی شده اند، نه بر مبنای حوزه های فرایندی کلیدی(KPA) [1]. گزارش Raynus اشاره به رابطه بین مقادیر کمی فرایندها و SW-CMM دارد و تکیه بر این نکته دارد که سنجش فرایند می‌تواند در بهبود رفتار یک سازمان تولید کننده نرم افزار تاثیر گذار باشد[۴]. روشهای مذکور به اهداف معین و تعریف شده موجود در حوزه های فرایندی مدل SW-CMM بطور خاص نپرداخته اند وروشهای نا دقیق و پیچیده برای بکار گیری هستند . حال آنکه هدف این مقاله ارائه سنجش‌های معین برای هر حوزه فرایندی بطورخاص است که به راحتی قابل فهم و بکارگیری باشد.

۲- حوزه فرایندی مدیریت نیازمندیها در مدل SW-CMM

مدل قابلیت رشد نرم افزار (SW-CMM) یک مدل سطح بندی شده فرایندهای نرم افزاری است. این مدل راهنمایی ها و دستورالعملهای مورد نیاز سازمانهای نرم افزاری را جهت بهبود مستمر فرایندهای نرم افزاری ارائه می کند. علاوه بر این، مدل مذکور بعنوان چارچوبی مناسب جهت سنجش میزان رشد یافتگی فرایندهای نرم افزاری مستقر در یک سازمان نرم افزاری بوده و مبنای قابل قبولی برای ارزیابی پیمانکاران نرم افزاری است.این مدل از پنج سطح شامل سطح آغازین[2]، قابل تکرار[3]، تعریف شده[4]، مدیریت شده[5] و بهبود یافته[6] تشکیل شده که در شکل (۱) نشان داده شده است. هر یک از سطوح فوق شامل چند حوزه فرآیندی کلیدی است و برای هر حوزه فرایندی در این مدل اهداف معین و تعریف شده ای وجود دارد. لذا سنجش میزان حصول این اهداف می تواند معیار مناسبی برای سنجش رشد یافتگی حوزه های فرایندی در یک سازمان نرم افزاری باشد. در این مقاله فرایند مدیریت نیازمندیها که یکی از حوزه های فرایندی مدل SW-CMM است به عنوان یک فرآیند نمونه جهت پیاده سازی کامل روش اندازه گیری مورد نظر انتخاب شده است. هدف از مدیریت نیازمندیها ایجاد درک مشترک از نیازمندیها بین مشتری و سازمان یک پروژه نرم‌افزاری است. این بدین معنی است که نیازمندیها باید کامل، مستند و کنترل شده باشد تا محصول نرم‌افزاری طراحی شده نیازهای مشتری را ارضا کند. اغلب اوقات نیازمندیها در چرخه حیات نرم‌افزار تغییر می‌کنند اما کنترل درخواستهای مربوط به تغییرات بخوبی انجام نمی‌شود. فرایند مدیریت نیازمندیها شامل فعالیتهای کنترل، جمع‌آوری، تدوین، توافق با مشتری، آزمون، مرور و مدیریت تغییرات روی نیازمندیها است. به عبارت دیگر این فرایند شامل فعالیتهای حصول اطمینان از مطابقت محصول نرم‌افزاری ساخته شده بر اساس نیازمندیها است [۳].



شکل ۱- حوزه‌های فرایندی مدل قابلیت رشد نرم‌افزار

1- Key Process Areas
2- Initial
3- Repeated
4- Defined
5- Managed
6- Optimized

۳- روش هدف / پرسش/ اندازه (GQM)[1]

روش GQM روشی است که به تمرکز سازمان روی سنجش اهداف خود کمک می‌کند. طی این روش اهدافی برای سازمان قبل ازجمع‌آوری داده درنظرگرفته می‌شود البته این اهداف، اهداف قطعی وغیرقابل تغییر نیستند.براساس اهداف تعریف شده پرسشهایی با پاسخ‌های قابل اندازه‌گیری طراحی می‌شود. این پرسشها، واحدهای اندازه‌گیری و پاسخ به آنها داده‌های مورد نیاز برای سنجش میزان حصول اهداف را فراهم می‌آورد.[5]

روش GQM شامل مراحل اصلی به شرح ذیل است:

- توصیف یک مجموعه از اهداف مبتنی بر نیازهای سازمان یا فرآیندها و تعریف آنچه که باید بهبود یابد و یا ارزیابی شود.
- تولید مجموعه ای از سؤالهای قابل کمی سازی بر اساس دستورالعملهای مختلفی که برای طبقه‌بندی سوالات براساس محصول یا فرایند موجود است. این سوالها قابل کاربرد برای تحلیل داده‌های مربوط به اهداف مختلف تلقی می شوند.
- تعریف یک مجموعهٔ از واحدهای اندازه گیری که اطلاعات قابل کمی سازی مورد نیاز برای پاسخ به سؤالات به تفکیک هر یک از آنها را فراهم آورد. در این مرحله واحدهای اندازه گیری که اطلاعات موردنیاز برای پاسخ به سؤالات تعریف شده در مرحله قبل را فراهم می‌آورد، تعریف و به هر یک از سوالات نسبت داده می‌شود. البته یک واحد اندازه گیری تعریف شده ممکن است در پاسخ به چند سوال بکار رود و لذا پاسخ هر سوال بصورت ترکیبی از چند واحد تعریف شده است. با یکبار تعریف قدمهای فوق، داده مورد نیاز برای پاسخ به سؤالات قابل کمی سازی تعریف شده براساس اهداف سازمان، جمع‌آوری و مورد تحلیل قرار می‌گیرد.

۴- کاربرد روش هدف/پرسش/اندازه در مدل قابلیت رشد نرم افزار

به نظر می‌آید مدل GQM , SW-CMM براحتی قابل ادغام هستندکه این مطلب در شکل ۲ نشان داده شده است. SW-CMM برای هر حوزه فرایندی یک یا چند هدف تعریف می‌کند که کلیه این اهداف قابل استفاده برای اولین قدم از GQM هستند[6]، اگرحوزه فرایندی مدیریت نیازمندیها را برای بررسی موضوع انتخاب کنیم،SW-CMM دو هدف مجزا برای فرایند مدیریت نیازمندیها تعریف می‌کند. *اولین هدف به شرح ذیل می‌باشد*[7].

" نیازمندیهای سیستم که به نرم افزار تخصیص داده شده به منظور تدوین یک محور مهندسی نرم‌افزار و مدیریت فرایند تولیدکنترل می‌شوند. "

این هدف بر ایجاد یک محور کنترل نیازمندیها تمرکز دارد. در صورت عدم کنترل نیازمندیها امکان ارائه تصویر شفاف ازمحصول نهایی ساخته شده بر پایه نیازمندیها وجود نخواهد داشت.

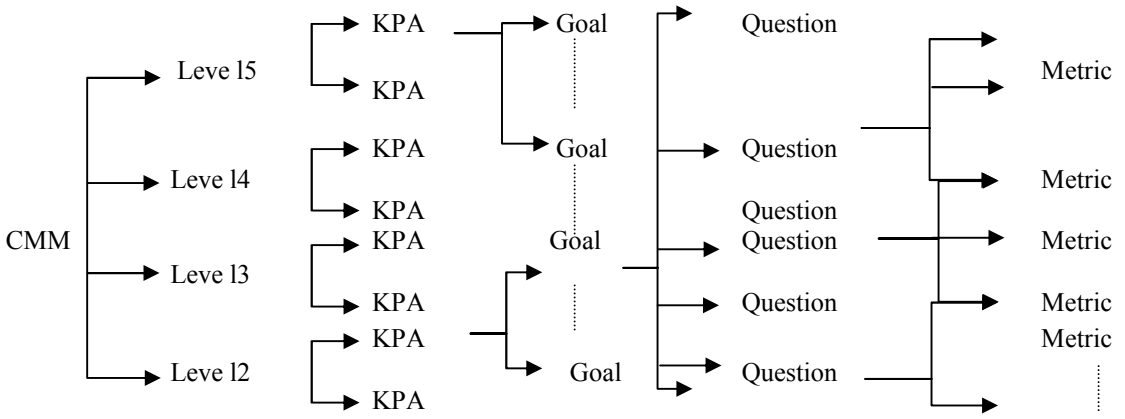
هدف دوم حوزه فرایندی مدیریت نیازمندیها به شرح ذیل است:

." سازگاری برنامه ریزیهای نرم‌افزار، محصولات و فعالیتها با نیازمندیهای سیستمی تخصیصی داده شده به نرم‌افزار بایستی کنترل شود "

تمرکز اصلی این هدف کنترل سازگاری بین نیازمندیها و محصول نرم‌افزاری ایجاد شده براساس این نیازمندیها است. این سازگاری در طراحی محصول مورد پذیرش کاربر نتیجه خواهد داد.

دومین قدم در روش GQM ایجاد یک مجموعهٔ سوالات قابل کمی سازی می‌باشد.[17]

[1] - Goal / Question / Metric



شکل ۲- رابطه بین GQM , CMM

۱-۴- سوالات اولین هدف حوزه فرایندی مدیریت نیازمندیها

جدول ذیل شامل سوالات مرتبط با اولین هدف حوزه فرآیندی مدیریت نیازمندیها است. این سوالات بر اساس حصول اطمینان از وجود کنترل تغییرات نیازمندیها طراحی شده است. لذا پایداری نیازمندیها ،هزینه پائین تغییرات، مدیریت مناسب تغییرات مطرح شده توسط مشتری ، کاهش میزان تغییرات، کاهش ناسازگاری ونقص نیازمندیها در یک پروژه نرم افزاری نشان دهنده وجود مدیریت و کنترل مناسب نیازمندیها است. در پاسخ به سوالات فوق باید به این نکته توجه داشت که پیچیدگی مدیریت و کنترل نیازمندیها رابطه مستقیم با حجم پروژه نرم افزاری دارد. لذا در معیارهای پاسخ به این سوالات می بایست حجم پروژه مورد ارزیابی را نیز لحاظ نمود.

جدول ۱ - سوالات و معیارهای اندازه گیری مربوط به هدف اول مدیریت نیازمندیها

ردیف	سوالات	دامنه عضویت فازی	معیارها
۱	سطح پایداری نیازمندیها چیست؟ (RS)[1]	U: [0…1]	RS_1= تعداد نیازهای اولیه ـ تعداد نیازهای نهایی / تعداد نیازهای نهایی
		U: [0…10]	RS_2 = تعداد تغییرات هر نیاز
		U: [3,5,7]	RS_3 = سطح تغییرات
۲	منشا تغییر نیازمندیها چیست؟(RCR)[2]	U: [0…1]	$RCR_1 = RS_1$
		U: [0…10]	$RCR_2 = RS_2$
		U: [0…15]	RCR_3 = متوسط تعداد آزمونهای هر نیازمندی
		U: [3,5,7]	RCR_4 = RS3
		U: [3,5,7]	RCR_5= مرحله ای از فرایند تولید که تغییرات تقاضا شده است
۳	هزینه تغییرات نیازمندیها چیست؟(CC)[3]	U: [0…1]	CC= هزینه تغییر / هزینه هر نیازمندی
۴	آیا تغییرات مورد نیاز قابل مدیریت هستند؟ (CM)[4]	U: [0…1]	CM_1= تغییرات رد شده ـ تغییرات پیشنهادی / تعداد تغییرات پیشنهادی
		U: [0…1]	CM_2= تغییرات انجام شده ـ تغییرات مطرح شده / تغییرات مطرح شده
		U: [0…1]	CM_3= تغییرات اساسی ـ تغییرات انجام شده / تغییرات انجام شده
		U: [0…1]	CM_4= اقلام پیکربندی متأثر از تغییرات / اقلام پیکربندی
		U: [0…1]	CM_5= تعداد نیازهای متاثر از یک تغییر / تعداد نیازمندیها
۵	آیا تعداد تغییرات با زمان کاهش می‌یابد؟ (CR)[5]	U: [0…1]	CR= تعداد تغییرات نهایی ـ تعداد تغییرات اولیه / تعداد تغییرات اولیه
۶	چه تعداد نیازمندی ناقص، ناسازگار ودر نظر گرفته نشده شناسائی می‌شود؟ (RIIM)[6]	U: [0…1]	$RIIM_1$= تعداد نیازمندیهای ناقص / تعداد کل نیازمندیها
		U: [0…1]	RIIM2= تعداد نیازمندیهای سازگار / تعداد کل نیازمندیها
		U: [0…0.5]	$RIIM_3$= تعداد نیازمندیهای فراموش شده / تعداد کل نیازمندیها

1- Requirement Stability
2- Requirement Change Reason
3- Change Cost
4- Change Mangementability
5- Charge Rate
6- Requirement Inconstancy, Incompleteness, Missed

۲-۴- سوالات برای دومین هدف از حوزه فرایندی مدیریت نیازمندیها

دومین هدف مربوط به کنترل سازگاری نیازمندیها با پروژه نرم‌افزاری است . این امر از طریق قابلیت ردیابی طی مستندات نرم‌افزار قابل حصول است. با پیاده سازی تغییر در نیازمندیها، سایر مستندات نیز متاثر از این تغییر خواهند بود. بنابراین در هر تغییر وضعیت نیازمندیها برنامه‌ریزی‌ها، محصولات کاری و فعالیتهای تعریف شده، فعالیتهای مستند شده و برنامه‌ریزی شده بایستی مورد بررسی قرار گیرند. سوالات مطرح شده در جدول شماره ۲ این موضوع را نشان می دهد.

جدول ۲ ـ سوالات و معیارهای اندازه گیری مربوط به هدف دوم فرایند مدیریت نیازمندیها

ردیف	سوالات	تابع فازی سازی	معیارها
۱	آیا محصول نرم‌افزاری نیازمندیها راتامین می‌کند؟(RS)[1]	U: [0…1]	$RS_1 =$ (تعداد نیازهای اولیه ـ تعداد نیازهای نهایی) / تعداد نیازهای نهایی
		U: [0…10]	$RS_2 =$ تعداد تغییرات هر نیاز
		U: [3,5,7]	$RS_1 =$ میزان تغییرات
۲	تاثیر تغییرات نیازمندیها روی پروژه نرم‌افزار چیست؟ (PEF)[2]	U: [0…0.3]	$PEF_1 =$ میزان نفر ماه صرف شده برای مدیریت نیازمندیها / میزان نفر ماه صرف شده برای پروژه
		U: [0…۱]	$PEF_2 =$ تعداد اقلام پیکربندی متأثر از یک تغییر در نیازمندیها / اقلام پیکربندی پروژه
		U: [0…0.7]	$PEF_3 =$ میزان نفر ماه صرف شده برای اعمال تغییرات / میزان نفر ماه صرف شده برای پروژه
۳	آیا تمام محصولات کاری با نیازمندیها سازگار هستند؟(RCO)[3]	U: [0…0.7]	$RCO =$ تعداد عدم تطابق / تعداد محصولات کاری

۵- هوش مصنوعی توزیع شده

مدلسازی و دانش هوش مصنوعی توزیع شده در سال۱۹۹۱ توسط گرسرشامل دوحوزه کاری زیر ارائه شده است[8].

الف- حل مسائل توزیع شده:

در این حوزه یک مسئله با دانش پیرامون آن به زیر مسئله های قابل حل تقسیم می شود. هر یک از این زیر مسئله ها بر اساس دانش پیرامون خود حل شده و با تلفیق نتایج آنها مسئله اصلی حل می گردد. نکته حائز اهمیت در این روش چگونگی تقسیم مسئله به زیر مسئله ها به طوری که با توجه دانش پیرامونی قابل حل بوده و در هنگام تلفیق شرایط جدیدی به مسئله وارد نسازد و حل نهایی کارایی لازم را داشته باشد، است

1- Requirement Stability
2- Requirement Effectiveness
3- Requirement Consistency

ب- سیستم های چند عاملی[1]

این سیستم ها شامل رفتار مجموعه ای از عاملهای خودکار با هدف حل یک مسئله است. خروجی هر عامل ورودی مناسب عامل بعدی است. عناصری چون کلاس عامل، ساختار عامل، دانش، قابلیت استنتاج و نتیجه گیری، تطبیق پذیری و یادگیری در داخل عامل تعریف می شوند. البته هر عامل می تواند بصورت ایستا و یا پویا طراحی و بکارگیری شود.

گروهی از عامل ها نیز هنگامی که در کنار همدیگر قرار می گیرند شامل عناصری چون سازمان گروه، هماهنگی ، همکاری، راهنمایی، رفتار همگون، طرح و هدف هماهنگ و ارتباطات عوامل داخل گروهی هستند. مدل مورد استفاده در این مقاله سیستم فازی چند معیاره است که نمونه ای از سیستم های چند عاملی بشمار می آید.

۱-۵- طراحی سیستم های فازی

سیستم های فازی به منظور مدلسازی دانش بشری در مورد یک مسئله خاص استفاده می شوند. بصورت کلی دانش بشری را به دو دسته خودآگاه و ناخودآگاه می توان تقسیم کرد. دانش خودآگاه بصورت صریح و روشن در قالب کلمات قابل توصیف است. دانش ناخودآگاه شامل وضعیتهائی است که انسانهای خبره می دانند چه کاری انجام می دهند. ولی نمی توانند آنرا بطور دقیق در قالب کلمات بیان کنند[۹]. در توصیف دانش ناخودآگاه خبره رفتار خود را در وضعیتهای مختلف نشان می دهد. در این حالت فرد خبره به عنوان یک جعبه سیاه در نظر گرفته می شود و ورودیها و خروجیهای او سنجیده می شوند. بدین ترتیب رفتار خبره به مجموعه ای ازورودی – خروجی تبدیل می شود. مسئله اساسی، ساخت سیستم های فازی بر اساس زوجهای ورودی ـ خروجی است. روشهای رایج جهت ایجاد سیستم های فازی عبارتند از:

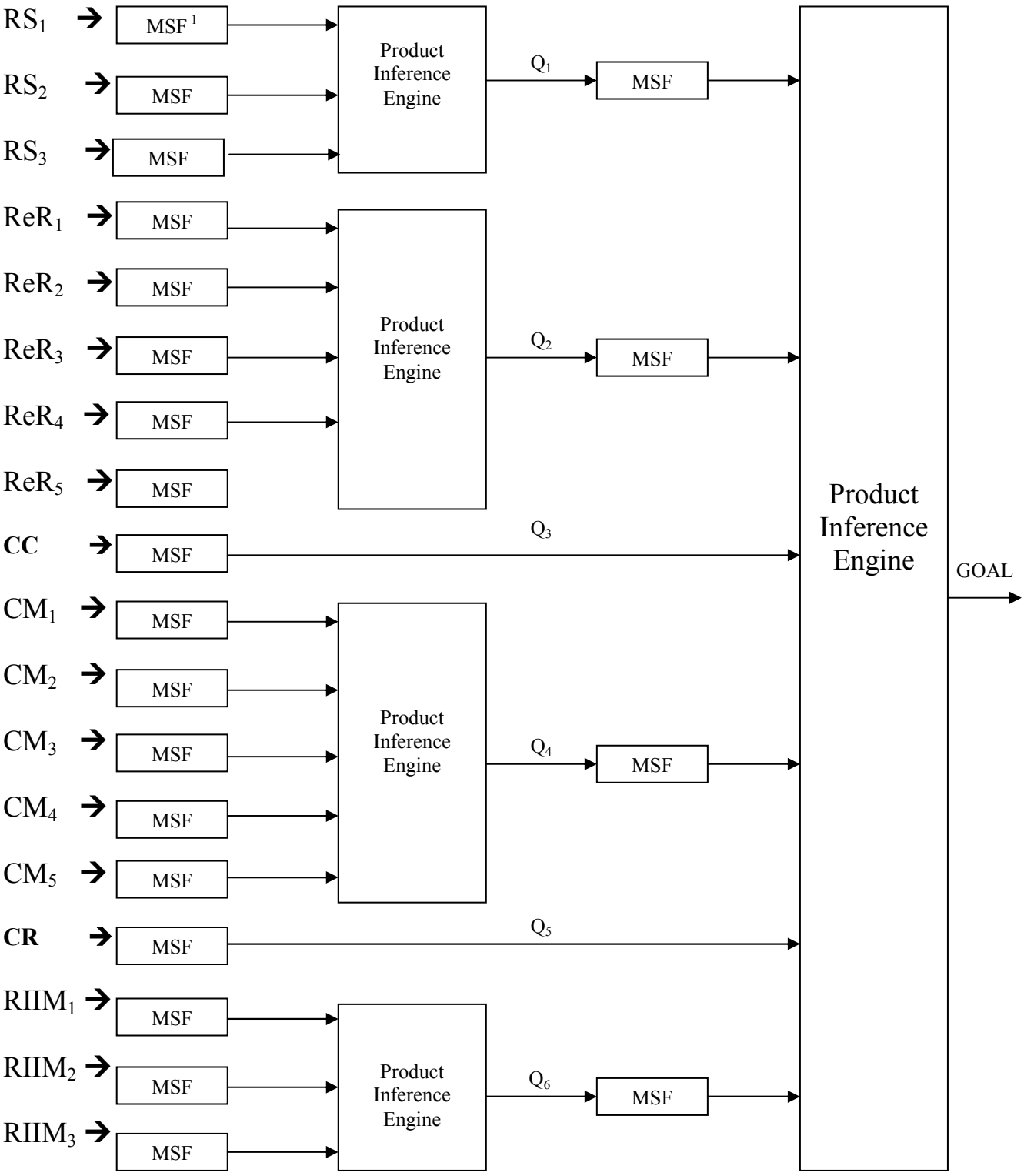
- روش شهودی ساده[2]
- تنظیم پارامترها بوسیله آموزش گرادیان نزولی
- نظیم پارامترها بوسیله آموزش کمترین مربعات بازگشتی

در طراحی سیستم های فازی انتخاب تعداد قواعد مناسب بسیار مهم است. انتخاب تعداد قواعد زیاد باعث پیچیده تر شدن سیستم فازی می شود که ممکن است برای حل مسئله ضروری نباشد. از طرفی دیگر انتخاب تعداد قواعد کم موجب ایجاد یک سیستم فازی ضعیف می شود.

۶- سیستم فازی چند عاملی سنجش فرایندهای نرم افزاری

حاصل اعمال روشGQM بر اهداف حوزه های فرایندی مدل SW-CMM ، مجموعه ای از سوالات و معیارهای پاسخ به آنها است. همانطور که در جداول شماره ۱ و ۲ نشان داده شده است برای هر یک از این معیار ها یک دامنه عضویت فازی قابل تعریف است. بدین ترتیب هر یک از معیارهای پاسخ ورودی های عامل اول سیستم فازی مورد بحث را تشکیل می دهند که در شکل ۳ نشان داده شده است. خروجی عامل اول فازی (سطح اول در نمودار بلوکی شکل ۳) مقادیر نهایی پاسخ به سوالات را فراهم می آورد. این مقادیر ورودی عامل دوم سیسنم فازی مورد بحث را تشکیل می دهد. خروجی این عامل (سطح دوم در بلوک دیاگرام شکل ۳) میزان حصول اهداف حوزه های فرایندی را تعیین می کندو خروجی عامل سوم این سیستم، میزان رشد یافتگی حوزه فرایندی مورد نظر مبتنی بر اهداف آن حوزه خواهد بود. شکل ۳ نمودار بلوکی سیستم چند عاملی پیاده سازی شده برای هدف اول حوزه مدیریت نیازمندیها را نشان می دهد. عامل اول در این سیستم دارای ۱۸ ورودی و ۶ خروجی است و عامل دوم داری ۶ ورودی و ۱ خروجی است که نشان دهنده میزان حصول هدف اول فرایند مدیریت نیازمندیها است . سیستم مذکور دارای فازی ساز منفرد، موتور استنتاج ضرب و غیر فازی ساز میانگین مراکز است.

1- Multi Agent Systems
2- Heuristics



شکل ۳ـ سیستم فازی چند معیاره حصول هدف اول فرآیند مدیریت نیازمندیها

۷- ابزار سنجش و تحلیل

سیستم فازی چند عاملی تشریح شده در بخش قبل به صورت یک ابزار سنجش و تحلیل فرایندهای نرم افزاری پیاده سازی شده است . از مهترین ویژگیهای این ابزار می توان به واسط کاربر بسیار ساده و قابل فهم آن ، اتوماتیک سازی محاسبات فازی ،ایجاد یک مخزن ذخیره سازی نتایج سنجشها ، قابلیت مقایسه و تحلیل نتایج سنجش های مختلف به منظور بررسی نحوه رشد فرایند های نرم افزاری سازمان نرم افزاری مرد نظر اشاره نمود. لذا ابزار مذکور امکان بسیار مناسبی را جهت تحلیل و پیش بینی اثرات و پیامدهای تغییرات در فرایندهای مستقر در یک سازمان نرم افزاری را فراهم می آورد. همچنین با استفاده از این ابزار میتوان یک پایگاه داده از میزان رشد یافتگی فرایندهای نرم افزاری پیمانکاران نیز ایجاد نمود.

۸- مثال

مدل اندازه گیری فرآیندهای نرم افزاری ارائه شده در این مقاله تاکید بر سنجش میزان حصول اهداف مستقیم حوزه های فرآیندی بر اساس یک مدل فرمال دارد. همانطور که در بخش های قبل ارائه شده هیچ یک از روشهای موجود به سنجش مستقیم کارایی فرآیندهای نرم افزاری نپرداخته اند و عمدتاً از طریق پارامترهایی مانند هزینه، زمان و نیروی انسانی صرف شده برای ساخت محصول نهایی به تحلیل فرآیندهای نرم افزاری اشاره کرده اند. لذا نتایج روش مورد بحث در این مقاله قابل مقایسه با سایر روشهای موجود نیست. ادامه این بخش شامل نتایج اجرای روش اندازه گیری مورد بحث برای حوزه فرآیندی مدیریت نیازمندیها در پروژه تولید نرم افزار امور متقاضیان و مشترکین تلفن همراه که در دانشگاه صنعتی خواجه نصیرالدین طوسی انجام شده است می باشد:

جدول نتایج اندازه گیری هدف اول فرآیند مدیریت نیازمندیها سیستم امور متقاضیان و مشترکین تلفن همراه

هدف	سوال	معیار	پاسخ سوال	میزان حصول هدف
اول	سطح پایداری نیازمندیها چیست؟	RS_1=.38 RS_2=4 RS_3=5	.۴۸	.۴۶
	تغییر نیازمندیها چیست؟	RCR_1=0.38 RCR_2=4 RCR_3=6 RCR_4=5 RCR_5=7	.۱۴	
	هزینه تغییرات نیازمندیها چیست؟ (CC)	CC=0.6	.۶	
	آیا تغییرات قابل مدیریت هستند؟(CM)	CM_1=1 CM_2=0.2 CM_3=0.8 CM_4=0.4 CM_5=0.2	.۵۵	
	آیا تعداد تغییرات کاهشی می باشد. (CR)	CR=0.8	.۸	
	چه تعداد نیازمندی ناقص، ناسازگار و در نظر گرفته نشده شناسائی می شود؟ (RIIM)	RIM_1=0.1 $RIIM_2$=0.05 $R.IIM_3$=0.02	.۸۱	

۹- نتیجه‌گیری

سیستم فازی سنجش و تحلیل فرایندهای نرم افزاری ارائه شده در این مقاله حاصل ادغام مدل SW-CMM و روش GQM و اصول طراحی سیستمهای فازی چندعاملی است از مهمترین ویژگیهای این سیستم می توان به تاکید آن بر اهداف فرایندهای نرم افزاری بصورت مستقیم ، بکارگیری معیارهای مستقیم سنجش ، قابلیت تعمیم آن به تمام حوزه های فرایندی

مدل SW-CMM ، سادگی بکار گیری ، قابلیت اتوماتیک سازی محاسبات و ایجاد یک محیط فرمال بر اساس منطق فازی در یک محیط غیر فرمال اشاره نمود.

در صورتی که کنترلهای فازی مورد استفاده در این مدل قابلیت یادگیری و توسعه پایگاه قواعد خودشان را داشته باشند.محدودیت زمان نیز از نظر تغییر نگرش توابع عضویت و دانش موجود در پایگاه قواعد مرتفع می شود. چرا که با تغییر زمان دانش پایگاه قواعد نیز رشد می کند و این بدین معنی است استفاده از متدولوژی شبکه عصبی در سیستم فازی اشاره شده در این مقاله می تواند باعث تنظیم و بهینه نمودن قواعد و توابع عضویت و در نتیجه ارتقاء دقت مدل گردد.

۱۰- منابع و مآخذ

[1] Paulk M.C.,Curtis B,Chrissis M.B. and Weber C.V., 1993,"Capability Maturity Model for Software Version 1.1", Software Engineering Institute Technical Report, CMU/SEI-93-TR-24, ESC-TR-93-177, Pittsburgh, PA, 1993.

[2] Baumert J.H and McWhinney M.S.,1992, "Software Measures and the capability Maturity Model". Software Engineering Institute Technical Report, CMU/SEI-92-TR-25, ESC-TR-92-0, 1992.

[3] Paulk M.C., Weber C.V., Garcia S., Chrissis M.B and Bush M., 1993 ,"key Practices of the Capability Maturity Model Version 1.1",1993, Software Engineering Institute Technical Report, CMU/SEI-93-TR-25 ESC-TR-93-178, Pittsburgh, PA, 1993.

[4] Raynus J., 1999, "Software Process Improvement With CMM", Artech House Publishers, Boston.

[5] Van Solingen R and Berghout E , 1999, "the Goal/Question/Metric Method- A Practical Guide for Quality Improvement of Software development", McGRAW-Hill Companiew, London.

[6] Bergstrom C, 2000, "process Metrics for Ericsson Erisoft AB-a proposal", Umea University Report Umnad 292/2000, Umea, Jan .

[7] Goodbrand A.D. and Wang Q, 2000, "Software Measurement Plan for the Requirements Management Key Process Area of the Capability Maturity Model For SENG623 Inc." March 1997, http://www.cpsc.ucalgary.ca/~alang/SENG623/, (12Feb.2000)

[8] Alexander, 1996, "Distributed fuzzy Control of multivariable systems" Klawer academic Publishers.

[9] Gregory E. Kersten, Stan Szpakowiz,1994, "Negotration in distributed artificial intelligence, IEEE.

[10] Basili V.R. and Rombach H.D,1988, "The TAME Project: Towards improvement – oriented software environments", in IEEE Transactions on Software Engineering 14(6), 1988, pp.758-773.

[11] Basili V.R., Caldiera C., and Rombach H.D,1994, "Experience Factory ", Encyclopedia of Software Engineering Volume 1, (Marciniak, J.J., editor),Jogn Wiley & Sons,1994, pp.469-476.

[12] Cline A,1997, "Overcoming Cultural barriers to the adoption of object technology", in OOPSLA 97 conference proceeding, Atlanta, Oct.

[13] El Eman K, Drouin J.N, and Melo W,1997, "SPICE The Theory and Practice of Software Process Improvement and Capability Determination", IEEE Computer Sociery Press, Los Alamitos,California.

[14] Fenton N.E. and Pfleeger S.L , 1997, "Software Metrics-A Rigorous & Practical Approach", 2nd Edition, International Thomson Publishing, Boston MA.

[15] Hammer T.F., Huffman L.L. and Rosenberg L.H.,1998, "Doing Requirements Right the First Time", CrossTald, Dec 1998, pp. 20-25.

[16] Kruchten P., 1999, "The Rational Unified Process- An Introduction", Addison Wesley Longman Ltd, Reading Massachusetts, May.

[17] Kuvaja P. and Bicego A., 1993, "BOOTSTRAP-Europe's Assessment Method", IEEE Software, May 1993, pp.83-85.

[18] Pulford, K., Kunntzmann-Combelles, A., and Shirlaw, s., "the Ami Handbook: A Quantitative Approach to Software Management", Addison Wesley Longman Ltd. UK, 1996.

[19] Reifer D.J.,2000, "Requirements Management: The search for Nirvana", IEEE Software, May / June 2000, PP. 45-47.

[20] Wohlin C, Runeson P, Host M., Ohlsson M.C., Regnell B. and Wesslen A,2000, "Experimentation in Software Engineering An Introduction", Kluwer Academic Publishers, Bost on/Dordrecht/ London, 2000.