مطالعات علوم کاربردی در مهندسی

ISSN: ۲٤۷٦-٤٥۰۷

دوره ۲، شماره ۱، بهار ۱۳۹۵

www.irijournals.com

صفحات ۵۰- ۶۴

دستیابی به مزایای معماری نرم‌افزاربه کمک مدلسازی UML درروش Extreme Programming

کامیارعبدالمحمدی [۱و۲]، منصور اسماعیل پور [۳]، محمدمهدی شیرمحمدی [۴]

[۱] دانشجوی کارشناسی‌ارشد مهندسی کامپیوتر، دانشگاه آزاداسلامی،ساوه، ایران

[۲] مدرس گروه کامپیوتر، دانشکده فنی سما ،دانشگاه آزاداسلامی،اسلام آباد غرب، ایران

[۳و۴]استادیار گروه کامپیوتر، دانشکده فنی مهندسی، دانشگاه آزاداسلامی، همدان، ایران

چکیده

روش‌های چابک نمونه‌ای از روش‌های توسعه نرم‌افزاری هستند که به پیشرفت سریع، سرعت و انعطاف‌پذیری در مقابل تغییرات تأکیددارند. روش‌های چابک دارای روش‌های متنوعی جهت توسعه می‌باشند که روش XP (برنامه‌نویسی بی‌نهایت) با توجه به رویکردش یکی از این روش‌ها می‌باشد. این روش یک سری چالش و ضعف دارد. یکی از چالش‌هامربوط به توجه کم این روش به ویژگی‌های کیفی و فعالیت معماری نرم‌افزار است. معماری نرم افزار به عنوان یکی از مهمترین موضوعات مهندسی نرم افزارشناخته شده و از طرفی نیازهایی مانند قابلیت اطمینان ،کارایی وامنیت نیازهایی هستند که متاثر از کیفیت معماری نرم افزار هستند رویکرد پیشنهادی ما تلاش می‌کند با تغییراتی در فرایند اکس پی و اضافه کردن نقاطی در روش اکس پی، رابطه‌ای بین معماری و روش چابک پیداکندکه هم به مزیت‌های معماری نرم‌افزار دست بیابد و هم باارزش‌ها و اصول چابکی روش اکس پی (برنامه‌نویسی بی‌نهایت) خدشه‌ای وارد نگردد.در این مقاله به ارائه رویکردی برای پاسخگویی به این چالش خواهیم پرداخت.

واژه‌های کلیدی: چابکی، روش برنامه‌نویسی بی‌نهایت، معماری نرم‌افزار، فعالیت‌های معماری، ویژگی‌های کیفی

۱- مقدمه

همان‌طور که می‌دانیم روش‌های توسعه نرم‌افزار به دو روش قدیمی و چابک تقسیم می‌شود و در دهه های اخیر ، پیچیدگی های توسعه نرم‌افزار بیشتر شده وروش های قدیمی با توجه به داشتن طرح و برنامه از ابتدا و نداشتن توانایی در مواجه با تغییرات و جوابگوی نیازهای کنونی که اصلی ترین آن انعطاف در مقابل تغییرات است نیستند.خواسته‌های مشتری از سیستم تغییر می‌کند و همچنین محیطی که نرم‌افزار بر پایه آن توسعه‌یافته تغییر می‌کند و تکنولوژی نیز به‌مرورزمان تغییر می‌کند و همچنین می‌دانیم نرم‌افزار برای مردم و توسط مردم توسعه و گسترش میابد (کراوفورد و دیگران،۲۰زعفرکریمی ودیگران،۲۰۱۲) روش‌های قدیمی، بر اساس طرح و برنامه هستند و نمی‌توانند این تغییرات را کنترل کنند، به همین دلیل داشتن یک روش توسعه که بتواند تغییرات را کنترل کند از اهمیت زیادی برخوردار است. امروزه، روش‌های چابک به‌عنوان نسل جدید روش‌های توسعه تلاش می‌کنند بر مشکلات روش‌های قدیمی غلبه نمایند. به هر صورت روش‌های چابک نیز یک سری ضعف دارند که مهم‌ترین این ضعف‌ها مربوط به معماری نرم‌افزار و عدم توجه به مزایای معماری نرم‌افزار مانند ویژگی‌های کیفی است(جنس و دیگران ،۲۰۰۶،الساموسی و دیگران ،۲۰۰۲، بک و بوهم،۲۰۰۲،تورک و دیگران،۲۰زعفرکریمی ودیگران،۲۰۱۲،شریفلو ، ۱۳۸۷ از میان روش‌های توسعه چابک،برنامه‌نویسی بی‌نهایت یکی از متدولوژی‌های سبک‌وزن است که در سال‌های اخیر موردتوجه جامعه نرم‌افزاری قرارگرفته و بسیاری از محققین بر کارایی آن نسبت به روش‌های سنتی صحه می‌گذارند(ویلیامز و دیگران ،۲۰۰۱).

۲- معماری نرم افزار

معماری نرم‌افزار، نرم‌افزار را در قالب مؤلفه‌ها شناسایی می‌کند اما به اجزاء درونی و ساختمان داده‌های مؤلفه‌ها نمی‌پردازد. معماری نرم‌افزار علاوه بر ساختار، به رفتار سیستم نیز می‌پردازد و به دنبال تحلیل ویژگی‌های کلیدی آن می‌باشد. این ویژگی‌های کلیدی، نیازهای کیفی هستند. معماری یعنی ارائه توصیفی فنی از یک سیستم که نشان‌دهنده ساختار اجزاء آن، ارتباط بین آن‌ها و اصول و قواعد حاکم بر طراحی آن و تکامل آن‌ها درگذر زمان باشد. هر سیستمی دارای معماری است حتی اگر اطلاعات معماری به‌طور مناسبی مستندسازی نشده باشد و حتی اگر معماری بسیار ساده بوده و شامل مؤلفه‌های فراوانی نباشد. به‌طورکلی، محققین سه مزیت اصلی برای انجام فعالیت‌های معماری قائل هستند(باس و دیگران ،۲۰۰۱)

۱) اتخاذ تصمیمات طراحی مناسب و رسیدن به ویژگی‌های کیفی

۲) ایجاد ارتباط مناسب میان ذینفعان: معماری نرم‌افزار باعث می‌شود سهامداران سیستم بتوانند از آن به‌عنوان مبنایی برای درک متقابل، مذاکرات، اجماع ارتباطات دوطرفه استفاده نمایند.

۳) ایجاد مدل انتزاعی از سیستم.

۳- مدلسازی uml

ایجاد یک مدل برای برای سیستم های نرم افزاری به اندازه داشتن نقشه برای ساختمان مهم و ضروری است.به عبارت دیگر،یک زبان ، با ارایه یک فرهنگ لغت و یک مجموعه قواعد امکان می دهد که با ترکیب کلمات این فرهنگ لغت و ساختن جملات ، با یکدیگر ارتباط برقرار کنیم.یک زبان مدلسازی ، زبانی است که فرهنگ لغات و قواعد آن بر نمایش فیزیکی و مفهومی آن سیستم متمرکزند.برای سیستم های نرم افزاری نیاز یه یک زبان مدلسازی داریم که بتواند دیدگاه های مختلف معماری سیستم را در طول چرخه تولیدآن،مدل کند. uml زبانی است برای مدلسازی و ایجاد نقشه نرم افزار.فرهنگ واژگان و قواعد زبانی uml به ما می گوید که چگونه یک مدل را بسازیم و بخوانیم.یکی از مهمترین نقش های uml تسهیل ارتباط بین

اعضای پروژه می باشد.دیگر ویژگی uml مستقل بودن از متدلوژی و فرایند تولید می باشد وما می توانیم طبق متدلوژی خود عمل کرده و به کمک uml عملیات ها را مدل کنیم(سایت پرجمدار)

۴- معماری نرم‌افزار و روش توسعه چابک

اگرچه ارسال‌های اخیر توجه ویژه‌ای به روش‌های توسعه چابک شده است ولی دیدگاه‌های متفاوتی درباره ارتباط بین روش‌های چابک و معماری نرم‌افزار وجود دارد. به گفته آقای ساندمارک[1] و همکارانش، «یکی از تفاوت‌های میان معماری نرم‌افزار و روش‌های چابک، این است که معماری نرم‌افزار باید در رابطه با مسائل کلیدی سیستم و قبل از ساخت آن تصمیم بگیرد و روش چابک باید با ساختار پایه‌ای در طول سیستم سروکار داشته باشد ویژگی‌های پشتیبانی شده را آماده سازد.» .در حقیقت روش‌های چابک برای نیازمندی‌های فعلی طراحی‌شده‌اند درحالی‌که روش‌های مبتنی برمعماری برای شرایط فعلی و قابل پیش‌بینی طراحی‌شده‌اند. بر این اساس آقای بوهم می‌گوید روش ترکیبی که هردو روش چابکی و برنامه محور را ترکیب کند. برای پروژه‌هایی که مخلوطی از ویژگی‌های سرعت و مبتنی بر برنامه را مدنظر قرار می‌دهند، لازم می‌باشد(کراوفورد و دیگران،۲۰زعفرکریمی ودیگران،۲۰۱۲). در مقاله‌ای که توسط آقای ابرامسون همکارانش ارائه‌شده، بیان‌شده است که معماری بر اصل پیش‌بینی استوار است درحالی‌که روش‌های چابک بر تطبیق‌پذیری تأکیدمی کندو می‌گوید: «به نظر می‌رسد تنش میان این دو حوزه، بر محور انطباق در مقابل پیش‌بینی قرار دارد و روش‌های چابک قاطعانه می‌خواهند تطبیق‌پذیر باشند ودروقتی تغییرات رخ می‌دهد، تصمیم‌گیری کنند. در حقیقت روش‌های چابک، معماری نرم‌افزار را به‌عنوان وسیله‌ای برای هل دادن به سمت پیش‌بینی می‌دانند که ابتدا برنامه‌ریزی بیش‌ازحد انجام می‌دهند.» مهم‌ترین دلیل به‌کارگیری معماری نرم‌افزار درروش توسعه چابک، ارتباط مناسب میان ذینفعان است؛ زیرا در توسعه چابک، تعاملات رودررو و چهره به چهره است به همین دلیل بسیاری از اطلاعات به‌خوبی منتقل نمی‌شود بنابراین معماری می‌تواند یک انتزاع مشترک از سیستم ارائه کند که به‌عنوان پایه‌ای برای فهم مشترک، مذاکره و ارتباط میان مشتری و توسعه‌دهنده مورداستفاده قرار گیرد(باس و دیگران ،۲۰۰۱)

۵- بررسی ویژگی های کیفی

نیازمندی‌ها در حالت کلی به دودسته نیازهای وظیفه مندی که به توانایی سیستم برای انجام کاری که برای آن ساخته‌شده و نیازهای غیر وظیفه مندی یا غیر کارکردی که همان مشخصه‌های کیفیتی می باشندمانند کارایی، امنیت، قابلیت استفاده مجدد، قابلیت اطمینان و... تقسیم می شوند.

۵-۱- کارایی[2]

کارایی ریشه در خاصیت منابع استفاده‌شده جهت برآوردن نیازها و همچنین چگونگی به اشتراک‌گذاری منابع در زمان مواجه با درخواست‌های چندگانه که باید روی یک منبع یکسان انجام شود، دارد. این نوع مسائل تحت عنوان مسائل زمان‌بندی مطرح می‌شود. کارایی صفتی است که به پاسخگویی سیستم مربوط می‌شود.

۵-۲-امنیت[3]

[1] -Sandmark

[2] - Performance

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴

امنیت ویژگی از سیستم است که توانایی یک سیستم در مقاومت در برابر دسترسی‌های غیرمجاز و همچنین امتناع از ارائه سرویس درزمانی که در حال ارائه سرویس به کاربران مشروع می‌باشد را نشان می‌دهد. تلاش جهت تجاوز به امنیت، حمله نامیده می‌شود و اشکال مختلفی دارد.

۵-۳-در دسترس بودن[۴]

این صفت با خرابی سیستم و نتایج مرتبط با آن سروکار دارد. خرابی زمانی رخ می‌دهد که سیستم قادر به تحویل سرویسی پایدار بر اساس ویژگی‌ها و خصوصیاتش نباشد. در بحث دسترس‌پذیری سیستم باید به این موارد پاسخ داده شود که چگونه خرابی سیستم تشخیص داده می‌شود؟ در زمان خرابی چه مشکلی پیش می‌آید؟ سیستم می‌تواند حداکثر چه مدت‌زمان خارج از سرویس باشد؟ چگونه می‌توان از خرابی جلوگیری کرد؟ مهم‌ترین موضوع در زمان رخداد خرابی این است که سیستم چه وقت تعمیر می‌شود.

۵-۴-قابلیت استفاده[۵]

قابلیت استفاده به این مطلب موضوع می‌شود که واسط کاربر تا چه حد برای استفاده کاربر ساده می‌باشد و نیازهای او را برآورده می‌سازد. لذا اگر بخواهیم قابلیت استفاده یک محصول نرم‌افزاری را اندازه‌گیری نماییم باید مسائل زیر را بررسی کنیم.

• قابلیت به خاطر سپاری[۶]: زمانی که کاربر پس از یک بازه زمانی وقفه مجدداً به سیستم برمی‌گردد، چقدر راحت می‌تواند عملیات را به یاد آورد؟

• رضایت: آیا کاربر از کار با سیستم احساس رضایت و خوشنودی دارد؟

• قابلیت یادگیری[۷]: سادگی و مدت‌زمانی که طول می‌کشد تا کاربری که به‌تازگی با سیستم روبرو شده است بتواند عملیات اولیه و اساسی سیستم را یاد بگیرد.

۶- معرفی روش xp

روش برنامه‌نویسی بی‌نهایت یک خط‌مشی از توسعه‌ی نرم‌افزار هست که بر مبنای بر مبنای چهار ارزش، ارتباط[۸]، سادگی[۹]، بازخورد[۱۰] وجرأت[۱۱] شکل‌گرفته است(کنت بک ،۲۰۰۰)که ارزش احترام[۱۲] نیز در سال ۲۰۰۰ به آن اضافه‌شده است(کنت بک و

[3] - Security

[4] - Availability

[5] - Usability

[6] - Memorability

[7] - Learnability

[8]- Communication

[9]- Simplicity

[10]- Feedback

[11]- Courage

و آندرس ،۲۰۰۴). ارزش‌ها و اصول و فعالیت‌ها و نقش افراد در کنار یک مدل فرآیندی، پایه و اساس روش برنامه‌نویسی بی‌نهایت را پی‌ریزی می‌نمایند(وست ودیگران،۲۰۰۲،بوهم ودیگران،۲۰۰۴،شریفلو ، ۱۳۸۷) اولین ارزش برنامه‌نویسی بی‌نهایت ارتباط است برنامه‌نویسی بی‌نهایت برای رابطه‌ی کلامی ارزش قائل است. یکی از دلایل شکست پروژه‌های نرم‌افزاری عدم روابط مطلوب بین اعضای تیم توسعه است. دومین ارزش برنامه‌نویسی بی‌نهایت سادگی است و انتظار می‌رود ساده‌ترین روشی که ما را به مقصود می‌رساند استفاده شود. بررسی واکنش مشتری در مواجهه با محصول و اعمال نظرات آن هابسیارحائزاهمیت است. در برنامه‌نویسی بی‌نهایت سیستم همیشه مورد بازخورد قرار می‌گیرد.زمانی که در توسعه سیستم، اشکالی مانند مشکل در طراحی کلان سیستم شناخته شود و به‌تبع آن مشکل در آزمایش سیستم بروز کند، تیم می‌بایست انسجام خود را حفظ کرده و درصدد رفع مشکل مربوطه باشد. ارزش احترام نیز بر حفظ احترام میان اعضای تیم تأکید دارد(کنت بک و آندرس ،۲۰۰۴)در XP توسعه در چندنشر وتکرار انجام می گیرد و در پایان هر نشر مجموعه ای از نیازهای سیستم پیاده می شود در طول هر تکرار زوج برنامه نویس به برنامه نویسی داستان های کاربری که قبلا توسط مشتری پیاده شده می پردازند.هر زوج برنامه نویس یک وظیفه را به عنوان ورودی دریافت و سپس به طراحی و آزمون واحد آن وظیفه می پردازند و در ادامه توسعه انجام گرفته که معمولا بازسازی کد نیز توسط برنامه نویسان انجام می شودزمانی که پیاده سازی وظیفه به پایان میرسد با کد های موجود یکپارچه می گردد واطمینان از پایداری کد نهایی حاصل می شود.به این فرایند در مجموع تکمیل وظیفه می گویند و از آنجایی که این فرایند توسط زوجهای مختلف و به طور موازی و بدون نظارت کیفی صورت می پایرد امکان شکل گرفتن ساختارهای ضعیف معماری بسیار تقویت می گرددکه در نتیجه ویژگی های کیفی را نیز تحت تاثیر قرار داده ومعماری نرم افزار به شدت دچار مشکل میگردد(زعفرکریمی ودیگران،۲۰۱۲)

۷-کارهای انجام شده

در یکپارچه‌سازی QAW با اکس پی QAW کارگاهی است که به ذینفعان، مشتریان و سایر اعضای تیم توسعه کمک می‌کند در اوایل توسعه، ویژگی‌های کیفی سیستم را شناسایی کنند. یکی از امتیازات QAW این است که این کارگاه قبل از عملیات کد نویسی و توسعه برگزار می‌شود.QAW یک‌راه برای شناسایی ویژگی‌های کیفی کلیدی سیستم را قبل از توسعه برای ما ایجاد می‌کند و این‌یک زمان مناسب برای برنامه‌نویسی بی‌نهایت است که قبل از عملیات توسعه ویژگی‌های کیفی شناسایی شود. پس از استخراج نیازمندی‌ها و نوشتن داستان‌های کاربری که در طی مراحل فرایند برنامه‌نویسی بی‌نهایت انجام می شود هماهنگی‌های لازم برای جلسه QAW برگزار می‌شود. در جلسه QAW ویژگی‌های کیفی سیستم استخراج، جمع‌آوری و ساماندهی می‌شود نتایج کارگاه QAW به همراه داستان‌های کاربری تجزیه‌وتحلیل می‌شود و به داستان‌های کاربری ویژگی‌های کیفی اضافه می‌شود. مهم‌ترین فایده این کارگاه، نگاهی با ویژگی‌های کیفی است که خلأ آن در فرآیند برنامه‌نویسی بی‌نهایت قابل‌مشاهده می‌باشد(ویلیامز ودیگران،۲۰۰۴)در یکپارچه‌سازی ATAM/CBAM با اکس پی روش ATAM یک روش ارزیابی معماری فراهم می‌کند و یک ارزش را به‌صورت گام‌به‌گام به XP اضافه می‌کند و در آن رویکرد به ارزیابی معماری پرداخته و خطراتی که در دستیابی به اهداف کیفی تأثیر دارند شناسایی می‌شوند و ارزیابی می‌کند. CBAM اطلاعات بیشتری را در زمینه هزینه تصمیمی که قرار است برای تکرار بعدی بگیریم در اختیارمان می‌گذارد و به ما کمک می‌کند که از میان تصمیم‌هایی که برای معماری می‌خواهیم بگیریم کدام‌یک ازنظر هزینه انتخاب آگاهانه‌تری است. تیم ارزیاب معماری که اعضای آن را پیش‌تر توضیح دادیم در جلسه‌ای با تیم توسعه شرکت کرده به ارزیابی معماری می‌پردازد. در ابتدای جلسه معمار توضیحاتی در رابطه با اهداف سیستم بیان می‌کند و تیم ارزیاب از روی این توضیحات بایستی اهداف و محدودیت

[12]- Respect

هاوالگوهای سیستم را شناسایی کرده و تصمیمات لازم را درباره قطعات تشکیل‌دهنده و ماژول‌ها و ارتباط بین ماژول‌ها و تاکتیک‌های معماری بگیرد(شریفلو ، ۱۳۸۷)

۸-رویکرد پیشنهادی

مهم‌ترین چالش اکس پی عدم توجه به معماری نرم‌افزار و مزایای معماری نرم‌افزار ازجمله دستیابی به ویژگی‌های کیفی می‌باشد. ما برای این چالش رویکردی ارایه داده ایم که البته رویکرد ما نباید باارزش‌ها و اصول روش اکس پی در تضاد باشد. بنابراین ما باید رابطه‌ای بین چابکی و معماری برقرار کنیم که این رابطه بتواند تلاشی در جهت نزدیک‌تر کردن معماری نرم‌افزار و چابکی باشد. همانظور که می دانیم معماری بر اساس طرح و قطعی می باشد درحالیکه چابکی به قطعی نبودن نیازها تأکید می‌کند. ما باید با این حقیقت کنار بیاییم که اکس پی به‌صورت تدریجی و تکاملی ساخته می‌شود و در آن مواجهه سریع و انعطاف‌پذیر با تغییرات پیش‌بینی‌شده است و طرحی از ابتدا ندارد. بنابراین نیاز داریم که شیوه جدیدی از ترکیب این دو بیابیم وویژگی‌های جدیدی را به اکس پی اضافه کنیم.براین کار تغییراتی را در فرایند اکس پی اعمال می کنیم. برای اجرایی کردن فعالیتمان به فرایناکس پی ۳قانون اضافه می کنیم.

۱-در ابتدا تیم مدل کننده ،تحت سرپرستی معمار ارشد تشکیل می‌شود تمام اعضای مشارکت‌کننده اعم از ذینفعان و برنامه‌نویسان و.... باید از هدف و انگیزه‌ی توسعه‌ی سیستم ،آگاه باشند یعنی محیط کل سیستم باید مشخص شود تا بتوان ویژگی‌های کیفی آنرابه طور موثر تحلیل کرد و به عبارتی باید یک شناخت کلی از دامنه به دست آید .

۲-اضافه کردن شخصی به عنوان معمار ارشد.

۳-مدلسازی uml توسط زوج های برنامه نویس.

این سه مرحله را در نقاطی از فرایند اکس پی قرار می دهیم به گونه ای که هم سیستم دارای معماری باشد و هم بتوان به مزایای معماری نرم افزار یعنی ویژگی های کیفی دست یافت.قبل از شروع قرایند اکس پی مرحله اول انجام میشودیعنی راه های دستیابی به ویژگی های کیفی.سپس داستانهای کاربری و ویژگی های کیفی که در مرحله اول بررسی شده بود را باهم ادغام نموده واکنون دارای داستانهای کاربری هستیم که علاوه بر نظر مشتری،ویژگی های کیفی به آن اضافه شده است.حال این داستانهای کاربری را به زوج های برنامه نویس داده و این برنامه نویسان موظف هستند در ابتدا قبل از برنامه نویسی به کمک مدلسازی uml یک مدل مفهومی و ساده از وظیفه محول شده را انجام دهند و آن را برای معمار ارشد بفرستند.معمارارشد با دریافت مدل های ایجاد شده توسط برنامه نویسان و ترکیب آنها ، یک مدل کلی ایجاد میکند.وهمچنین معمارارشد می‌تواند ناسازگاری‌های موجود در معماری را نیز کشف کرده و اصلاح کند.با اضافه شدن ویژگی‌های کیفی به داستانهای کاربری در مرحله‌ی اول باعث اضافه شدن ویژگی‌های کیفی به ساختار سیستم می شود و با انجام فعالیت فوق همواره مدلی مفهومی از ساختار سیستم وجود دارد که قابل‌دسترسی برای تمام اعضای تیم توسعه می‌باشد.

۹-مطالعه موردی

این مطالعه در حقیقت پروژه‌ای نرم‌افزاری است که در پالایشگاه گاز ایلام اجراشده است. در این پروژه هدف و کاری که انجام‌شده است پیاده‌سازی سیستم خریدکالامی باشد و به این معنا که پالایشگاه می‌خواهد در مورد مسئله خرید کالا نظر کارمندانش یا واحد خاصی را جویا شود و برای این کار، یک پیام که حاوی صورت‌مسئله موردبحث در پالایشگاه را به‌صورت سؤال به‌صورت بلی / خیر توسط سیستم مرکزی برای سیستم تمام کارمندان ارسال شود. این سیستم از دو بخش تشکیل‌شده

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴

است و یک بخش آن بر روی سیستم تمام کارمندان نصب می‌شود و بخش دیگر آن‌که سیستم مرکزی نامیده می‌شود و توسط یک فرد در حوزه فناوری اطلاعات مدیریت می‌شود. کارمندان سؤال را از طریق نرم‌افزار نصب‌شده بر روی دستگاهشان دریافت می‌کنند و این نرم‌افزار بخشی از پروژه‌ای است که قرار است طراحی و پیاده‌سازی شود. کارمندان پس از دریافت سؤال، مطالعه و بررسی گزینه‌ها نظر خود را توسط نرم‌افزار طراحی‌شده اعلام می‌کنند و جواب را به سمت نرم‌افزار سیستم مرکز ارسال می‌کنند و سیستم مرکز بایستی بتواند این آراء را دریافت، جمع‌آوری نمود، تحلیل و نتیجه را اعلام دارد. پیاده‌سازی این نرم‌افزاری سؤالی است که توسط مشتری از تیم توسعه پرسیده می‌شود حال می‌خواهیم بر اساس رویکرد پیشنهادی این مسئله را حل کنیم. تا متوجه شویم که آیا قابلیت اضافه کردن ویژگی‌های کیفی و داشتن یک مدل انتزاعی و کلی از سیستم را در هرلحظه دارد و یا خیر. همان‌طور که می‌دانیم در الگوریتم پیشنهادی در مرحله اول جلسه‌ای با حضور همه افراد تأثیرگذار در پروژه مانند مشتری، توسعه‌دهنده، آزمون گر، برنامه‌نویس تحت نظر معمار ارشد برگزار می‌شود و هدف از توسعه سیستم و کار سیستم را مشخص می‌کنیم.

- در مرحله اول الگوریتم

در جلسه با مشتری نیازهای وظیفه مندی سیستم را مشخص می‌کردیم.با بررسی سؤالات مشتری نیازهای وظیفه مندی سیستم را به‌صورت زیر تحلیل می‌کنیم:

۱- هنگامی‌که نیاز به اتخاذ تصمیم‌گیری در مورد مسئله مهمی در سازمان باشد یک نرم‌افزار به نام سیستم مرکزی سؤالی را در مورد مسئله‌ای که سازمان می‌خواهد درباره آن تصمیم بگیرد به‌صورت پیام برای نرم‌افزاری که بر روی سیستم تمام کارمندان قرار دارد ارسال می‌شود و کارمندان می‌فهمند قرار است تصمیم‌گیری شود.

۲- این پیام به‌صورت، دوگزینه‌ای بله یا خیر می‌باشد.

۳- نرم‌افزار نصب‌شده بر روی سیستم کارمندان به‌گونه‌ای طراحی می‌شود که کارمندان نتوانند بیش از یک‌بار در رأی شرکت کنند.

۴- پیام‌های بی‌ربط به سؤال مطرح‌شده و پیام‌های خالی و مخدوش شده بایستی توسط سیستم مرکز شناسایی و حذف گردد.

۵- سیستم مرکزی باید بتواند سؤال را ارسال و جواب‌ها را دریافت نموده، جمع‌آوری کرده، تحلیل و نتیجه را اعلام کند.

تیم توسعه با بررسی این تکرارها ویژگی‌های کیفی، کارایی، امنیت، در دسترس بودن و قابلیت استفاده و قابلیت اطمینان را در نظر می‌گیردبرای طراحی مدل فوق ۳ تکرار را در نظر می‌گیریم. تکرار اول طراحی یک نظرسنجی جدید بر اساس درخواست مدیران و تکرار دوم شامل ارسال، دریافت و ثبت پیام‌ها است و تکرار سوم امور امنیتی را بر عهده دارد.

برای هر تکرار داستان‌های کاربری را مشخص می‌کنیم

- **تکرار اول**

برای تکرار اول طراحی یک نظرسنجی جدید بر اساس درخواست مدیران

همان‌طور از نام‌گذاری این تکرار نشان می‌دهد هدف ایجاد نظرسنجی در اداره است و داستان های کاربری آن موارد زیر هستند.

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴

۱-سیستم درخواست‌های مدیران را برای ایجاد یک نظرسنجی جدید بر اساس نظر مدیران بالادست بررسی می‌کند. (مثلاً مدیر بررسی می‌کند که آیا لازم است برای واحد امور اداری یک نظرسنجی طراحی کنیم یا خیر)

۲-سیستم باید بتواند آیتم‌های یک نظرسنجی را بر اساس درخواست واحد مربوطه ایجاد نماید. (ایجاد سؤالات نظرسنجی)

۳-سیستم باید نفرات یا واحدهای دریافت‌کنندهٔ نظرسنجی را از ابتدا شخص نماید. (مثل ارسال نظرسنجی برای گروه‌های خاص)

- **اضافه کردن ویژگی‌های کیفی به داستان‌های کاربری تکرار اول**

برای داستان کاربری شماره ۲ می‌توان ویژگی‌های کیفی قابلیت استفاده را اضافه کرد و به این معنا که نحوه چینش و استفاده از ادبیات مناسب جهت طراحی سؤالات یک نظرسنجی به نسبت پرسنل پاسخ‌گو طراحی کنیم.

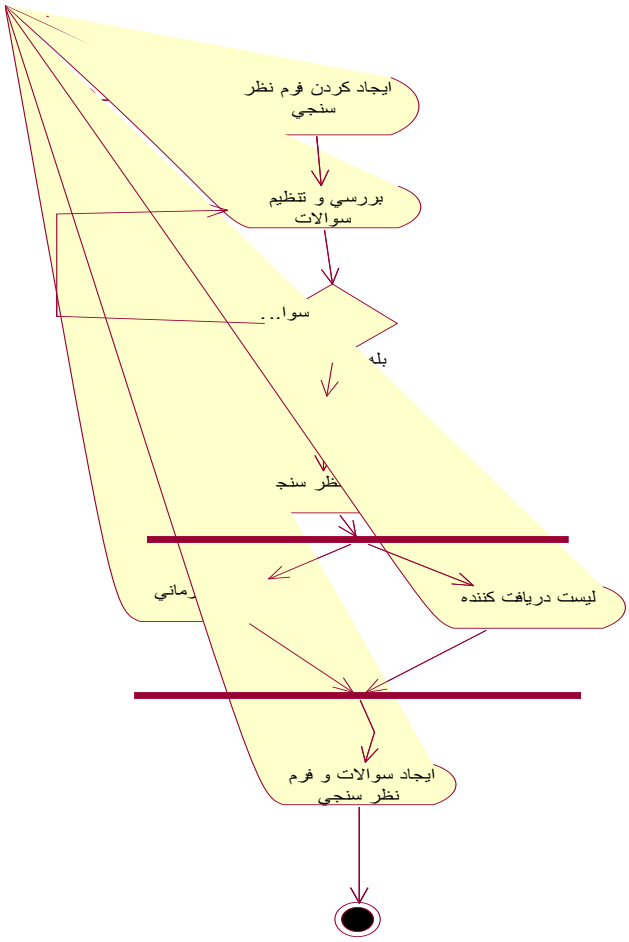
برای داستان کاربری شماره ۳ می‌توان ویژگی‌های کیفی امنیت را اضافه کرد به این معنا که جهت ارسال سؤالات نظرسنجی برای واحدهای خاص یا گروه‌های خاص بکار رود تا درروند تکمیل نظرسنجی توسط افراد یا واحدهای غیر معتبر جلوگیری شود.

- **پیاده‌سازی داستان‌های کاربری**

همان‌طور که مشخص است ما ویژگی کیفی را به داستان‌های کاربری اضافه نموده و اکنون دارای داستان‌های کاربری هستیم که علاوه بر نظر مشتری دارای ویژگی‌های کیفی نیز می‌باشد. حال این را به زوج برنامه‌نویس می‌دهیم و برنامه‌نویس وظیفه خواهد داشت در پایان هر توسعه، یک مدل مفهومی ساده از معماری را برای داستان کاربری ایجاد کرده و برای معمار ارشد بفرستد.

طراحی تکرار اول به کمک uml

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴



شکل شماره ۱. مدلسازی تکرار اول

- **تکرار دوم**

طراحی یک نظرسنجی جدید بر اساس درخواست مدیران

همان‌طور از نام‌گذاری این تکرار نشان می‌دهد هدف ارسال برگه‌های رأی توسط سیستم مرکز برای تمامی دستگاه‌های موجود در اداره است و کارمندان توسط نرم‌افزار نصب‌شده بر روی سیستم خود این برگه‌های رأی را دریافت نموده و پس از بیان نظرات خود بر روی صورت‌مسئله، این برگه‌ها را به سمت سیستم مرکزی ارسال می‌کنند و سیستم مرکزی بایستی بتواند آراء را دریافت نموده و نتایج تحلیل کند بنابراین داستان‌های کاربری طبق نظر مشتری مسائل زیر می‌باشد:

۱-کاربر سیستم مرکزی بایستی بتواند تصمیم را به رأی بگذارد که این به‌صورت بلی/ خیر می‌باشد.

۲- سیستم مرکزی بایستی بتواند پیام را ارسال کند برای تمام کارمندان و واحدهای مجاز از طریق نرم‌افزار سیستم مرکزی.

۳-کارمند بایستی بتواند از طریق نرم‌افزار نصب‌شده بر روی سیستم خود، محتوای پیام نظرسنجی را دریافت نموده و مطالعه نماید و نتیجه را ثبت نماید.

۴-کارمند در یک بازه زمانی خاص بایستی سؤال نظرسنجی را پاسخ دهد در غیراینصورت پاسخ غیر معتبر است.

۵-سیستم مرکزی بایستی بتواند پیام ارسال‌شده توسط کارمند را دریافت نمود.

۶- سیستم مرکزی بایستی بتواند آرای جمع‌آوری‌شده را بر اساس محتوا تحلیل کرده و نتایج آن را در پایگاه داده ذخیره نماید.

- **اضافه کردن ویژگی های کیفی به داستان کاربری دوم**

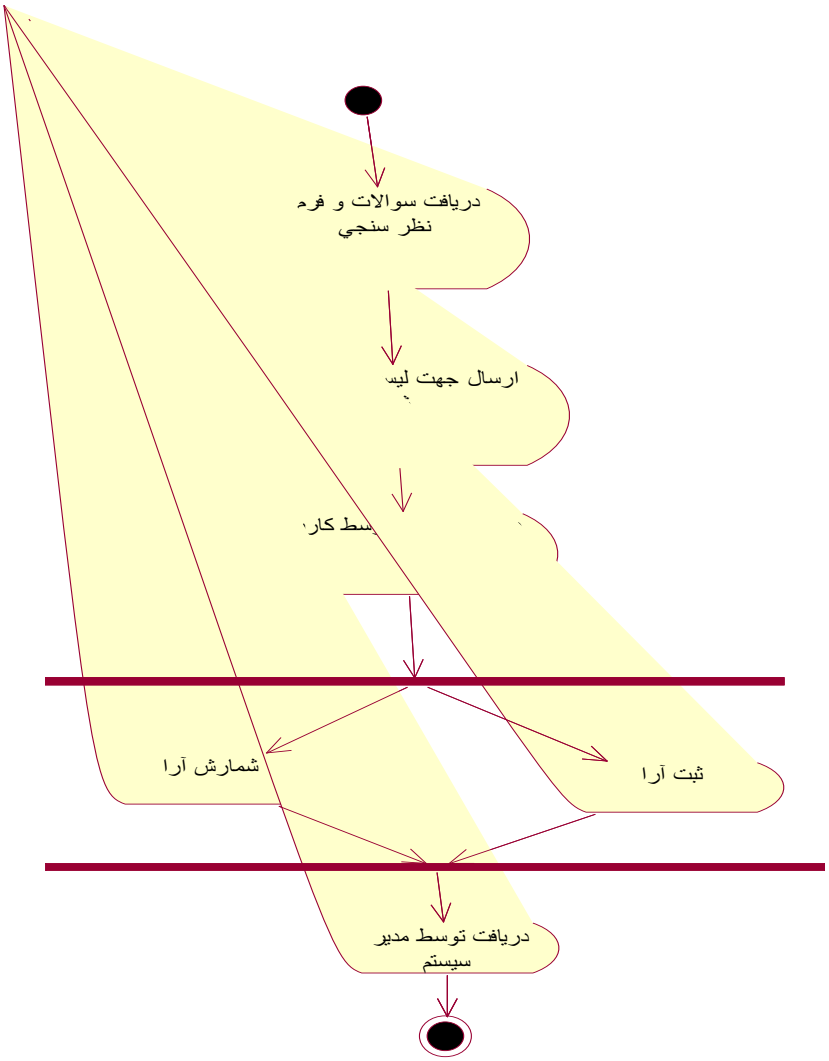
همان‌طور که از داستان‌های کاربری شماره ۲ برمی‌آید سیستم مرکزی باید بتواند همزمان با ۱۰۰ نفر ارتباط داشته باشد همچنین این پیام بایستی فقط در میان افراد اداره به نظرسنجی گذاشته شود نبایستی سیستمی غیر از سیستم اداره این پیام را دریافت کند که ما این قابلیت سیستم را بر اساس ویژگی کیفی امنیت نام‌گذاری می‌کنیم و در داستان‌های کاربری شماره ۵ ما بایستی بتوانیم نتیجه را در هرلحظه موردمطالعه قرار دهیم که این را قابلیت در دسترس بودن می‌نامیم.همچنین سیستم مرکزی بایستی بتواند آراء را به‌درستی بشمارد و بر اساس نظر کارمندان که ما اینجا ویژگی کیفی قابلیت اطمینان را در نظر می‌گیریم. همچنین در داستان‌های کاربری شماره ۳ کارمندان باید پیام را دریافت نموده ما بایستی در نظر داشته باشیم که دانش کامپیوتر بعضی از کارمندان پایین است بایستی ویژگی کیفی قابلیت استفاده را اضافه کنیم.

پیاده‌سازی داستان‌های کاربری

همان‌طور که مشخص است ویژگی کیفی را به داستان‌های کاربری اضافه نموده و اکنون دارای داستان‌های کاربری هستیم که علاوه بر نظر مشتری دارای ویژگی‌های کیفی نیز می‌باشد. حال این را به زوج برنامه‌نویس می‌دهیم و برنامه‌نویس وظیفه خواهد داشت در پایان هر توسعه، یک مدل مفهومی ساده از معماری را برای داستان کاربری ایجاد کرده و برای معمار ارشد بفرستد.

طراحی تکرار دوم به کمک uml

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴



شکل شماره ۲. مدلسازی تکرار دوم

تکرار سوم

یکی از اهداف سیستم این بود که افرادی غیر از افراد اداره توانایی شرکت در نظرسنجی را نداشته باشند و همچنین افراد اداره نتوانند بیش از یک‌بار در نظرسنجی شرکت کنند همان‌طور که مشخص کردیم در تکرار دوم ما قابلیت ارسال و دریافت را به هر دو بخش پروژه اضافه کردیم و همچنین بخش سیستم مرکزی قابلیت شمارش آراء و تحلیل را نیز اضافه نمودیم؛ اما هیچ کنترلی بر روی نظرسنجی نداشته‌ایم.

داستان‌های کاربری تکرار سوم

سیستم مرکزی بایستی بتواند بر اساس سامانه حضوروغیاب پرسنل، افراد غایب را تشخیص داده و از ثبت نظرسنجی توسط یو زر آن‌ها خودداری نماید.هر کارمند در صورتی توسط نرم‌افزار خود می‌تواند رأی بدهد که دارای یک رمز عبور منحصربه‌فرد بوده و این رمز را به‌درستی وارد کرده در غیر این صورت نمی‌تواند رأی دهد.سیستم مرکزی باید بتواند از نظرسنجی بیش از یک‌بار هر سیستم جلوگیری کند.اطلاعات مربوط به کارمندانی که نظرسنجی آن‌ها ازنظر امنیتی دارای اشکال است از سیستم حذف

گردد تا خللی در نتایج کلی ایجاد نگردد.پس از اتمام فرآیند امنیتی، آرا کل افراد شرکت‌کننده در نظرسنجی بررسی می‌شوند و نتایج نظرسنجی در غالب جدول و نمودار برای مدیران ارسال می‌گردد.

اضافه کردن ویژگی‌های کیفی به داستان‌های کاربری تکرار سوم

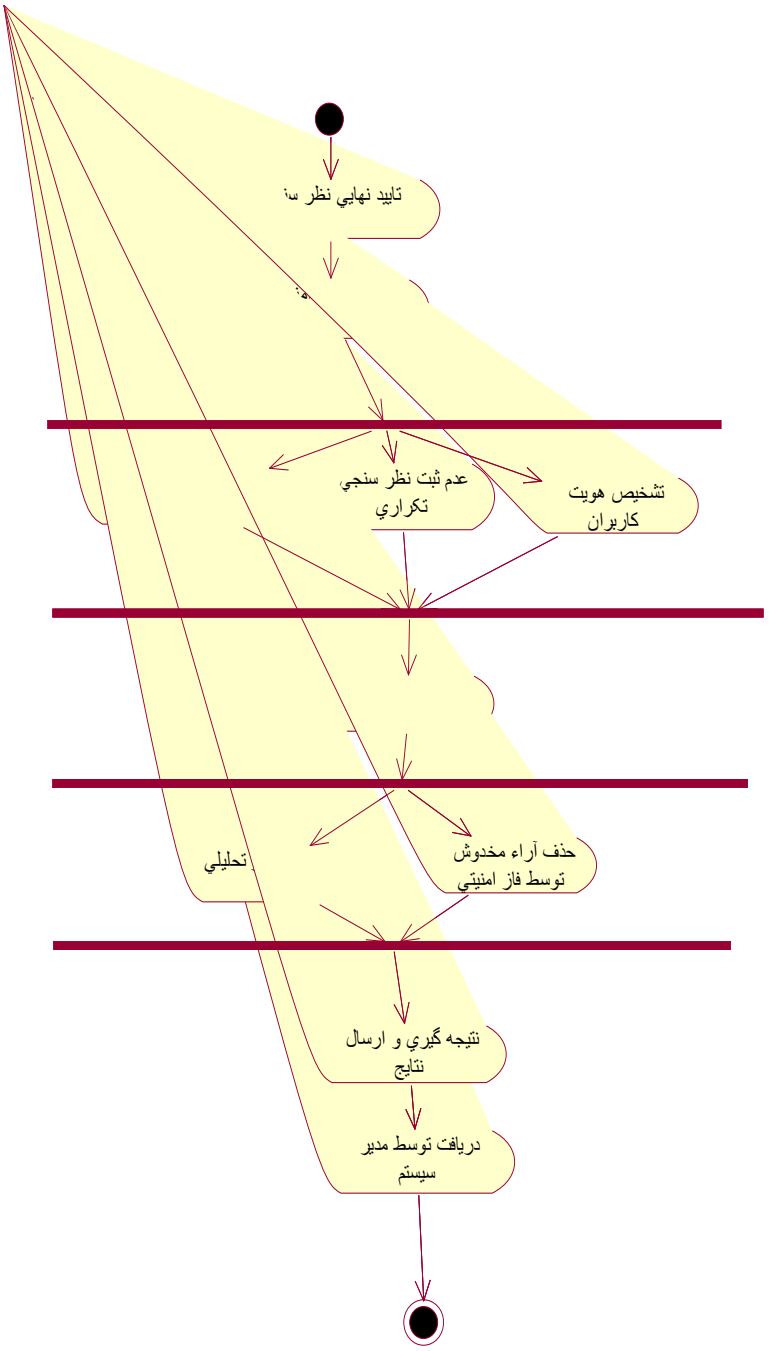
برای پیاده‌سازی داستان‌های کاربری شماره ۱ ما قابلیت همکاری را در نظر می‌گیریم. به این معنا که چگونه دو سیستم بتوانند با یکدیگر ارتباط داشته باشد و اطلاعات ردوبدل کنند و همچنین می‌توان قابلیت استفاده نیز اضافه کرد به این معنا که واسط کاربر ساده‌ای طراحی کنیم که در آن عدم حضور پرسنل مشخص شود و اصلاً پیام نظرسنجی ارسال نشود.برای داستان کاربری شماره دو ویژگی کیفی امنیت را در نظر می‌گیریم که به معنای جلوگیری کردن و مسدود کردن نفوذ گران جهت پیدا کردن رمز عبور.برای داستان کاربری شماره سه قابلیت اطمینان اضافه می‌کنیم.برای داستان کاربری شماره چهار قابلیت صحت و امنیت را در نظر می‌گیریم بدین معنا که فقط اطلاعات معتبر و صحیح را در سیستم باقی می‌گذاریم.برای داستان کاربری شماره پنج قابلیت کارایی و سودمند بودن سیستم را در نظر می‌گیریم بدین‌صورت که نتایج نظرسنجی را به‌صورت متنی و نمودار در اختیار مدیران قرار می‌دهیم.

پیاده‌سازی داستان‌های کاربری تکرار سوم

اکنون داستان‌های کاربری علاوه بر نظر مشتری حاوی ویژگی‌های کیفی می‌باشد هر داستان کاربری را به زوج برنامه‌نویس مخصوص خود می‌دهیم و برنامه نویسان موظف هستند بعد از پایان توسعه یک مدل مفهومی ساده را ایجاد کنند و برای معمار ارشد بفرستند.

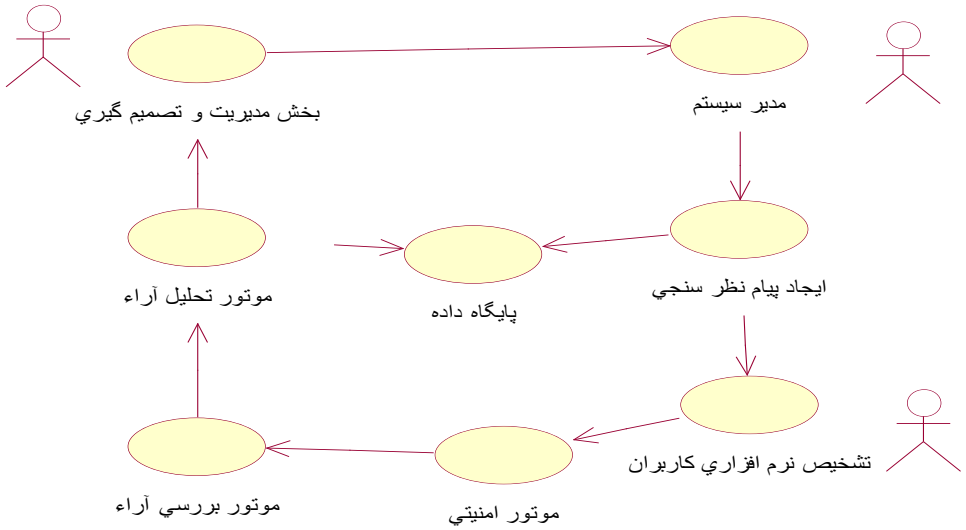
طراحی تکرارسوم به کمک uml

دوره ۲، شماره ۱، بهار ۱۳۹۵، صفحات ۵۰- ۶۴



شکل شماره ۳. مدلسازی تکرارسوم

اکنون معمار ارشد هر سه مدل ایجاد شده برای سه تکرار را دارد وبا ترکیب این سه مدل یک مدل مفهومی را ایجاد می کند .



شکل شماره ۴. مدلسازی معمار ارشد

۹-جمع‌بندی و نتیجه‌گیری

از همان ابتدای پیدایش روش برنامه‌نویسی بی‌نهایت این انتقاد مطرح گردید که عدم توجه به معماری نرم‌افزار باعث مشکلاتی برای پروژه‌ها خواهد گردید. در طول سال‌های اخیر چندین رویکرد برای برطرف نمودن این ضعف ارائه گردیده است ولی از آنجائی که هیچ‌کدام از راه‌حل‌های پیشنهادشده، به‌طور همه‌جانبه به مسئله توجه ننموده‌اند، تاکنون روش مدون و کارایی ارائه نگردیده است تا پاسخگوی جنبه‌های متعدد چابکی و معماری نرم‌افزار باشد. رویکرد پیشنهادی با نگاهی به هر دو حوزه چابکی و معماری نرم‌افزار، راه‌حلی مبتنی بر ارزش‌های چابکی و سازگار با اصول و فعالیت‌های روش برنامه‌نویسی بی‌نهایت ارائه می‌نماید که به‌گونه‌ای مؤثر، دستیابی به ویژگی‌های کیفی را مدنظر دارد

منابع

۱. شریف لو، امیر ملزم، ۱۳۸۷،جاسازی فعالیت‌های معماری نرم‌افزار درروش برنامه‌نویسی بی‌نهایت پایان‌نامه کارشناسی ارشد مهندسی کامپیوتر، دانشگاه شهید بهشتی، تهران.

2. Bass L. Clements P, Kazman R. “Software Architecture in Practice،Addison-Wesley Professional”، 2003.
3. Beck K. Boehm B ،2003 ،”Agility through Discipline A Debate” Computer،vol. 36 ، no.6.
4. Beck K, Andres C2004.. Extreme Programming EXPlained: Embrace Change ،2nded. Addison-Wesley Professional ،
5. Beck K. ،2000.Extreme Programming EXlained: Embrace Change ،1st ed. Addison-Wesle Professional

6. Boehm B. Turner R. ، 2003. Balancing Agility and Descipline A Guide for thePerplexed ،AddisWesley
7. BroderickCrawford,ClaudioLe´ondelaBarra,RicardoSotoandEricMonfroy"AgileSoftware Engineering as CreativeWork" CHASE 2012, Zurich, Switzerland IEEE,PP.20-26
8. Elssamadisy A. Schalliol G ،2002. "Recognizing and responding to "bad smells" inextreme programming" Proceedings of the 24th International Conference onSoftware Engineering ،pp. 617-622
9. Jense R. M ler T. Sonder P, Tjr nehj G، 2006."Programming; Introducing"Developer Stories Proceedings of 7th International Conference on Agile Processes and Extreme Programming in Software Engineering ،pp. 164 – 168
10. parchamdar.com/forum/archive/index.php/t-373.htm
11. Robert L. Nord,William G. Wood,Paul C. Clements" Integrating the Quality Attribute Workshop (QAW) and the Attribute-Driven Design (ADD) Method" Software Architecture Technology Initiative.2004
12. Turk D. France R. Rumpe B ,2012 ،"Limitations of Agile Software Processes"Proceedings of 3rd International Conference on Extreme Programming and Flexible Processes in Software Engineering ،pp. 43-46 ،2002.
13. West D. Metaphor ،2002. ،Architecture ،and XP ،Agile Alliance
14. Williams L, Upchurch R. , 2001. "Extreme programming for software engineering education"31st ASEE/IEEE Frontiers in Education Conference
15. Zafar Karimi,Sajjad Behzady, Ali Broumandnia, 2012, 'Achieving the Benefits of Agility in Software Architecture xp'،(JCSIT) Vol 4, No 5