

RUP به عنوان یک متدولوژی شیء، گرا مجموعه گسترده‌ای از راهبردهایی است که جنبه‌های تکنیکی توسعه نرم افزار و سازماندهی آن را در برمی گیرد و توجه بیشتر آن بر روی تحلیل و طراحی نیازمندی‌های سیستم است.

معرفی متدولوژی تحلیل و طراحی RUP

قسمت اول

سمیه وکیلی فرد، فرزانه سیفی

اشاره

تقریباً در اکثر علوم مهندسی مبحثی تحت عنوان «طراحی مهندسی» وجود دارد. در علم نرم افزار روش‌های «تحلیل و طراحی نرم افزار» به آن دسته از فنون مهندسی اطلاق می‌شوند که پیرامون مراحل و روند هر یک از فازهای تحلیل، طراحی، پیاده‌سازی و تست نرم افزار صحبت می‌کنند. برخی از این روش‌ها از آن چنان دقت و ظرافتی برخوردارند که به عنوان استاندارد و یا شبه استاندارد صنعتی شناخته شده‌اند. هدف از به کارگیری این روش‌ها آن است که محصولات نرم افزاری طراحی و پیاده‌سازی شده براساس این استانداردها از درجه بالایی از کارایی و کیفیت برخوردار باشند. مقاله‌ای که پیش رو دارید چکیده‌ای است از یک پایان‌نامه مقطع کارشناسی که درباره متدولوژی RUP (Rational Unified Process) توضیح می‌دهد. نظر به اهمیت موضوع مورد بحث در زمینه کیفیت و کارایی نرم افزار و همچنین مقوله «رعایت استانداردها و کیفیت» در تولید و صادرات نرم افزار، مطالعه این مقاله برای تحلیل‌گران، برنامه‌نویسان نرم افزار، مدیران پروژه‌های نرم افزاری و همه علاقمندان می‌تواند سودمند باشد. علاقمندان به این مبحث جهت تماس با نویسندگان مقاله می‌توانند با پست الکترونیکی editors@shabakeh-mag.com تماس بگیرند.

Boehm Spiral	Incremental	Waterfall	RUP	مزایا
*	*	*	*	نیاز به تخصص نیروی کار
*	*	*	*	قانونمندی و نظم مدیریت پروژه
*	*	*	*	قابلیت گزارش دهی
*	*	*	*	سهولت تخصیص منابع
*	*	*	*	کارایی و وظیفه مندی سریع
*	*	*	*	در ابتدای کار به جمع‌آوری مجموعه کاملی از نیازمندی‌ها نیاز ندارد
*	*	*	*	تعداد و یا مقدار منابع ثابت است
*	*	*	*	کنترل ریسک و هزینه با استفاده از تکنیک Prototyping
Boehm Spiral	Incremental	Waterfall	RUP	معایب
		*	*	در ابتدای کار به جمع‌آوری مجموعه کاملی از نیازمندی‌ها نیاز دارد
		*	*	اجراء رفتارهایی از سیستم که قابل پیاده سازی نیستند مانع ارتباط صحیح تحلیل‌گران و طراحان می‌گردد.
*	*	*	*	در نظر گرفتن بخشی از اهداف و نیازمندی‌های پروژه به جای تمام آن‌ها در ابتدای کار، مشکلاتی را در مدیریت پروژه به وجود می‌آورد
	*	*	*	نیاز به واسطه‌های کامل برای برقراری ارتباط بین ماژول‌ها
*	*	*	*	عدم سازگاری بین بلزینی‌های رسمی و ماژول‌های طراحی شده جهت بررسی قسمت‌های مختلف برنامه
*	*	*	*	موکول کردن حل بعضی از مشکلات اساسی سیستم به آینده تا زمانی که مرحله اول مدل‌سازی و پیاده‌سازی یک سری قسمت‌ها به صورت کامل انجام گیرد.

جدول ۱

مقدمه

به طور کلی مهم‌ترین مرحله در فرآیند تولید یک نرم افزار، تحلیل و طراحی آن است. اهمیت این موضوع از آن جهت است که طی این فاز، سیستم نرم افزاری مورد نظر به شکل سازمان‌یافته‌ای شناسایی شده و تغییرات مورد نیاز در قالب‌های مشخصی آن مستند و بر روی آن اعمال می‌شوند. از طرفی تکنیک‌ها و راهکارهایی جهت پیاده‌سازی نرم افزار به برنامه‌نویسان معرفی می‌گردد که سیستم را به طور قابل توجهی انعطاف‌پذیر می‌سازد. به این ترتیب هرگونه تغییر و توسعه آن به راحتی امکان‌پذیر خواهد بود. در عین حال پشتیبانی از یک سیستم نرم افزاری که فاز تحلیل و طراحی آن به خوبی انجام گرفته باشد بسیار ساده‌تر از پشتیبانی سیستم‌هایی است که در تولید آن‌ها به این مهم توجه کافی نشده است. جهت تحلیل و طراحی سیستم‌ها، متدولوژی‌های متفاوتی وجود دارند. در واقع یک متدولوژی تحلیل و طراحی از یک مدل توسعه نرم افزار به همراه یک یا چند تکنیک به وجود می‌آید. به عبارت دیگر می‌توان گفت:

مدل + تکنیک یا تکنیک‌ها = متدولوژی
 انواع متدولوژی‌ها را می‌توان به سه گروه اصلی تقسیم کرد: ۱- ساخت یافته (Structured) مانند SSADM و Jackson ۲- شیء گرا (Object Oriented) نسل اول مانند Alabiso و COSD و نسل دوم مانند Fosion و Booch ۳- جزءگرا (Component Oriented) مانند ROP و Perspective.

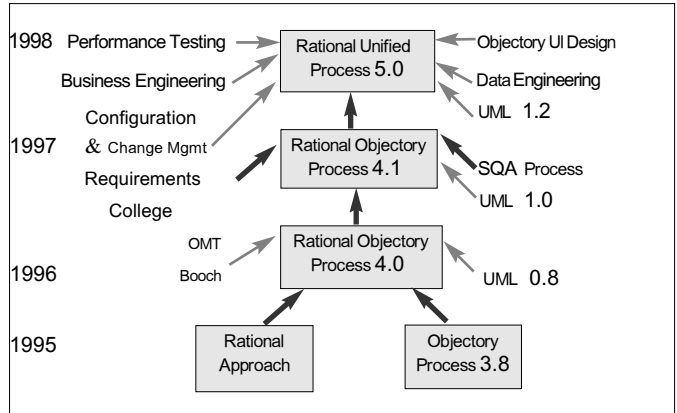
در یک پروژه ممکن است تکنیک‌ها با هم ادغام شده و یا بخشی از یک تکنیک به کار گرفته شود. متدولوژی‌های شیء گرا یکی از انواع روش‌های تحلیل و طراحی سیستم‌های نرم افزاری می‌باشند. این روش در طراحی و پیاده‌سازی آن دسته از نرم افزارهای کامپیوتری که با به کارگیری تکنیک‌ها و ابزارهای برنامه‌نویسی شیء گرا (OOP) تولید و عرضه شده باشند، استفاده می‌شود. RUP به عنوان یک متدولوژی شیء گرا مجموعه گسترده‌ای از راهبردهایی است که جنبه‌های تکنیکی توسعه نرم افزار و سازماندهی آن را در برمی‌گیرد و توجه بیشتر آن بر روی تحلیل و طراحی نیازمندی‌های سیستم است. از ویژگی‌های این روش می‌توان به قانونمندی و نظم مدیریت پروژه، قابلیت گزارش دهی، سهولت تخصیص منابع و کارایی و وظیفه‌مندی سریع اشاره کرد.

اگرچه امروزه در کشور ما به کارگیری روش‌های ساخت یافته تحلیل و طراحی (به عنوان مثال روش Waterfall) متداول‌تر است اما همان‌گونه که زبان‌های برنامه‌نویسی

شیءگرا قابلیت‌های بیشتر و بهتری را در اختیار برنامه‌نویسان قرار می‌دهند، روش‌های تحلیل و طراحی شیءگرا نیز به طور قابل ملاحظه‌ای امکان تحلیل و شناسایی جزئیات بیشتری از سیستم را برای تحلیل‌گر فراهم می‌آورند. جدول ۱ به طور خلاصه مزایا و معایب چند روش معروف و شناخته شده در تحلیل و طراحی نرم‌افزار (RUP, incremental, waterfall, Boehm) را نشان می‌دهد.

تاریخچه

متدولوژی RUP طی سالیان دراز تکامل یافته و حاصل تجربیات افراد و سرمایه‌گذاری شرکت‌های بسیاری می‌باشد. روند تکامل این متدولوژی به طور خلاصه در شکل ۱ نمایش داده شده است.



شکل ۱- روند تکامل متدولوژی RUP

متدولوژی RUP بر اساس مدل Rational Objectory Process (نسخه ۴) ساخته شد. این مدل حاصل ادغام مدل‌های Rational Approach Process و Objectory Process (نسخه ۳) است که پس از یکی شدن شرکت‌های Rational و Objectory AB در سال ۱۹۹۵ بوجود آمد.

بر اساس آن بخش از متدولوژی RUP که به مدل Objectory مربوط می‌باشد، هر فرآیند دارای ساختار خاصی است. این بخش مفاهیم اصلی use case را در بر می‌گیرد. (use caseها نمایانگر پردازش‌های موجود در سیستم می‌باشند). بخش دیگر مرتبط با مدل Rational است که بازبینی‌های مکرر و تکمیل توسعه و ساخت سیستم را امکان‌پذیر می‌سازد.

از طرفی پس از ادغام شرکت‌های Requisite و SQA با شرکت Rational، جهت مدیریت نیازمندی‌ها و انجام عملیات تست، از تکنیک‌های این دو شرکت استفاده شد. این مدل فرآیند، اولین مدلی بود که ابزار UML 0.8 را به کار گرفت. مدل Objectory در سال ۱۹۸۷ در سوئد ارایه شد و نتیجه تحقیقات Ivar Jacobson و همکاری او با شرکت اریکسون بود که در شرکت وی (Objectory AB) تهیه گردید. چون مفاهیم use case و همچنین روش طراحی شیءگرا در متدولوژی Objectory بسیار مورد توجه قرار گرفته بود، این مدل به سرعت در صنعت نرم‌افزار راه یافت و توسط شرکت‌های مختلف در سراسر جهان پذیرفته شد. یک نسخه ساده از این متدولوژی در سال ۱۹۹۲ به صورت کتاب منتشر گردید. RUP به عنوان یک نمونه کامل و خاص از متدولوژی Unified Software Development توسط Ivar Jacobson, James Rumbaugh, Grady Booch در این کتاب شرح داده شده است.

متدولوژی RUP

RUP مجموعه گسترده‌ای از راهبردهایی است که جنبه‌های تکنیکی توسعه نرم‌افزار و سازماندهی آن را در بر می‌گیرد و توجه بیشتر آن بر روی تحلیل و طراحی نیازمندی‌های سیستم است.

موفقیت یک پروژه در این متدولوژی به دو اصل زیر وابسته می‌باشد:

- (۱) زمان - تقسیم چرخه حیات نرم‌افزار به فازها و حلقه‌ها
- (۲) اجزاء پردازش - تولید مجموعه مشخصی از اجزاء و عناصر سیستم که فعالیت‌های آن‌ها به طور کامل تعریف شده است.

بر اساس این متدولوژی یک پروژه از نظر زمانی به چهار فاز تقسیم می‌گردد: ۱. فاز ادراک (Inception): در این فاز شناسایی و درک مناسب از مساله مورد نظر دنبال می‌شود.

خروجی‌های این فاز عبارتند از: نمودارهای سطح یک use case، مدل Business، مدل ارزیابی ریسک (Risk Assessment Model) برنامه جامع مدیریت و کنترل پروژه و سند vision که این سند یک دید عمومی از نیازمندی‌های پروژه است و اصل قراردادی را ایجاد می‌کند که بر اساس آن، تعریف نیازمندی‌ها از قاعده خاصی تبعیت می‌نماید. این سند در اختیار مدیران، actorها (تمام کسانی که با سیستم مربوطه سرو کار دارند)، طراحان و سرمایه‌گذاران پروژه قرار داده می‌شود تا درک مناسب از سیستم فعلی را برای آن‌ها فراهم سازد.

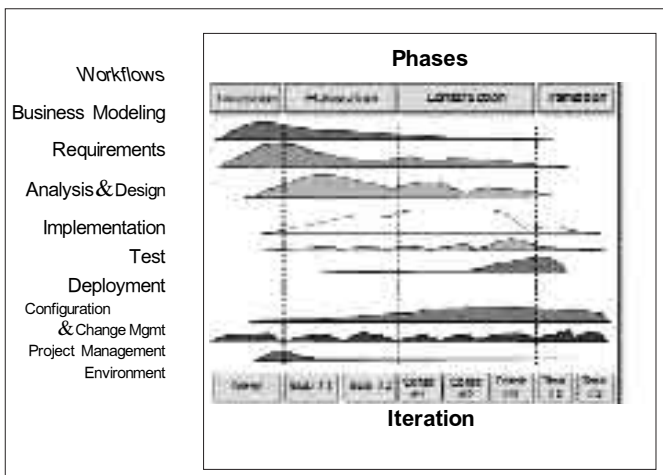
۲. فاز مهارت (Elaboration): تعیین فعالیت‌ها و منابع مورد نیاز، همچنین شناسایی ویژگی‌ها و طراحی ساختار سیستم، اهم اهداف این مرحله هستند. در این فاز از نمودارهای Class، Object، Activity، State، Collaboration و Use Case استفاده می‌شود.

۳. فاز ساخت (Construction): این مرحله به منظور تهیه سیستم طراحی شده در نظر گرفته شده است. در این فاز از نمودارهای Sequence، Collaboration و Component استفاده می‌شود.

۴. فاز انتقال (Transition): انتقال سیستم نهایی که شامل نصب و راه اندازی سیستم و تحویل و آموزش آن به کاربر می‌باشد.

از طرفی با توجه به اصل دوم (اجزای پردازش)، فعالیت‌های زیر در یک پروژه انجام می‌گیرند که این فعالیت‌ها قسمت عمده نمودار اصلی جریان کار را در این متدولوژی تشکیل می‌دهند:

۱. مدل‌سازی Business (Business Modeling): شناسایی قابلیت‌های سیستم و نیازمندی‌های کاربران در این قسمت صورت می‌گیرد.
۲. نیازمندی‌ها (Requirements): تعریف vision همراه با مجموعه‌ای از نیازمندی‌های کارکردی و غیرکارکردی در این بخش مشخص می‌شود.
۳. تحلیل و طراحی سیستم (Analysis & Design): چگونگی تحقق سیستم در فاز پیاده‌سازی شرح داده می‌شود.
۴. پیاده‌سازی (Implementation): تولید کد نرم‌افزار
۵. تست (Test): ارزیابی کل سیستم
۶. استقرار سیستم (Deployment): تحویل سیستم به کاربر و آموزش آن



شکل ۲- چرخه حیات در متدولوژی RUP

معرفی نمودار اصلی جریان کار (Workflow) در متدولوژی RUP

در متدولوژی RUP نمودار اصلی جریان کار، روند فعالیت‌هایی را نشان می‌دهد که برای تحلیل و طراحی یک سیستم نرم افزاری می‌بایست طی شوند. این نمودار دارای دو بخش است: نمودار اصلی جریان کار مربوط به پردازش و نمودار اصلی جریان کار مربوط به امور پشتیبانی.

نمودار اصلی جریان کار مربوط به پردازش

این نمودار به ترتیب دارای قسمت‌های زیر می‌باشد:

۱- مدل‌سازی Business

۲- نیازمندی‌ها

۳- تحلیل و طراحی سیستم

۴- پیاده‌سازی

۵- تست

۶- استقرار سیستم

در ادامه مقاله هریک از شش گام فوق به تفصیل مورد بررسی قرار خواهند گرفت. (در این قسمت تنها دو مرحله اول و دوم را بررسی می‌کنیم و مراحل بعدی را می‌توانید در شماره‌های بعدی ماهنامه شبکه مطالعه کنید).

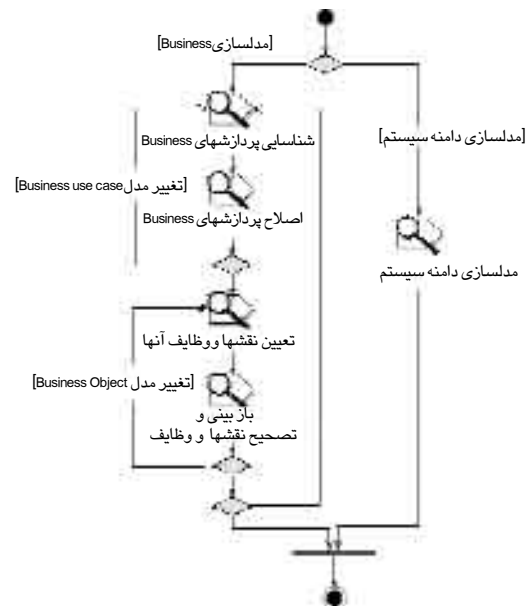
(۱) مدل‌سازی Business

هدف از اجرای این فاز درک ساختار Business و استخراج نیازهای سیستم برای پشتیبانی تشکیلات می‌باشد. همچنین این مدل تضمین می‌کند که کاربران، مشتریان و طراحان سیستم درک کلی از تشکیلات یافته‌اند. نمودار ورود فعالیت‌های این فاز را نشان می‌دهد.

بررسی قسمت‌های مختلف فلوجارت مدل‌سازی Business

الف- شناسایی پردازش‌های Business

در این مرحله لازم است که تیم طراحی نرم افزار به یک درک کلی از سیستم و محدودیت‌های آن برسد و از بین تمامی پردازش‌ها آن‌هایی را که نیاز است با جزئیات بیشتری مشخص گردند، تعیین نماید. اهداف این بخش عبارتند از: تعریف اصطلاحات علمی و فنی مرتبط با سیستم، طراحی مدل Business Use Case و اولویت بندی Business Use Case برای توصیف جزئیات بیشتر.



نمودار ۱- نمودار مدل‌سازی Business

طراحان مدل Business می‌بایست از دامنه Business مربوطه و همچنین چگونگی کارکرد سیستم‌های فعلی جهت اتوماسیون آن آگاهی یافته و اطلاعات دقیقی را درباره تکنیک‌های مدل‌سازی به دست آورند. به این منظور می‌بایست گفتگویی با مشتریان و کاربران سیستم صورت گیرد. نمونه‌هایی از تکنیک‌های شناسایی پردازش‌های Business عبارتند از:

Brain Storming and Idea Reduction .

Use Case Workshops .

Storyboarding .

Role Playing .

ب- اصلاح پردازش‌های Business

در این قسمت تمامی use case ها به طور دقیق و با جزئیات کامل بررسی می‌شوند.

ج- تعیین نقش‌ها و وظایف آن‌ها

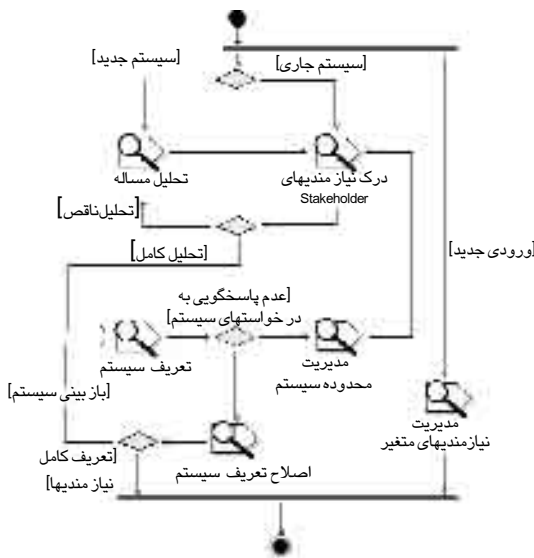
تعریف و تعیین تمامی نقش‌ها موجودیت‌های قابل تحویل و رویدادها در Business مربوطه، شرح شکل‌گیری Business Use Case Realization توسط کارکنان (Business Workers) و موجودیت‌های حرفه (Business Entities) اهداف اصلی این بخش هستند. جهت دستیابی به این اهداف می‌توان از تکنیک‌های ذکر شده در بند (الف) مانند Storyboarding و Role Playing استفاده کرد.

د- بازبینی و تصحیح نقش‌ها و وظایف

در این قسمت جزئیات مربوط به موجودیت‌های Business، وظایف کارکنان و تطبیق نتایج مدل‌سازی این موجودیت‌ها با نیازمندی‌های سیستم مورد بررسی قرار می‌گیرند.

(۲) نیازمندی‌ها (Requirements)

در این بخش مشتریان و کاربران درباره چگونگی عملکرد سیستم به توافق رسیده و طراحان از نیازمندی‌های سیستم درک بهتری پیدا می‌کنند. همچنین محدوده سیستم، تعیین شده و رابط‌های گرافیکی سیستم تعریف می‌گردند. روند فعالیت‌های این فاز در نمودار ۲ مشاهده می‌گردد.



نمودار ۲- نمودار نیازمندی‌ها

بررسی قسمت های مختلف فلوجارت نیازمندی ها

الف - تحلیل مساله

در این قسمت اهدافی چون ایجاد یک vision document برای پروژه و به توافق رسیدن در طرح ها و اهداف سیستم دنبال می شوند.

این نمودار به دفعات در فازهای ادراک و مهارت به کار می رود. فعالیت اصلی در این مرحله توسعه سند vision است. هم زمان با آن تحلیل گر می تواند actorهای سیستم را تعریف نماید.

از جمله تکنیک های مورد استفاده در این بخش روش های Brain Storming, نمودار Fishbone و نمودار Pareto می باشند.

ب- درک نیازمندی های Stakeholderها (ذینفعان)

به طور کلی stakeholder به تمام کسانی گفته می شود که به نوعی با پروژه مربوطه یا نتیجه حاصل از آن در ارتباط هستند. در این بخش هدف، استخراج و جمع آوری اطلاعات از stakeholderها در مورد پروژه مربوطه می باشد. درخواست Stakeholderها به صورت یک لیست درخواست (wish list) تنظیم می گردد که این لیست به عنوان یک پارامتر در تعریف مدل use case و مشخصات تکمیلی موثر می باشد. فعالیت های اصلی این بخش در فازهای ادراک و مهارت صورت می گیرند.

برای استخراج خواسته های ذینفعان، تکنیک های خاصی به کار گرفته شده و نهایتاً اطلاعات به دست آمده اولویت بندی می شوند. لیست درخواست های اولویت بندی شده برای بازبینی و تصحیح سند vision و شناخت بهتر خصوصیات نیازمندی ها بکار می رود، همچنین به طور هم زمان بحث در مورد use case های سیستم و actorهای آن ها آغاز می گردد. برای حصول اطمینان از صحت اطلاعات مربوطه می توان از تکنیک های ذکر شده در بند (الف) بخش مدل سازی Business استفاده نمود.

ج - تعریف سیستم

در این قسمت سه هدف عمده دنبال می شوند که عبارتند از سازماندهی سطح اطلاعاتی اعضای تیم درباره سیستم، تحلیل سطح بالای نتایج به دست آمده از درخواست های stakeholderها و ثبت نتایج توسط مدل ها و مستندات.

به طور کلی فعالیت های این بخش در فازهای ادراک و مهارت صورت می گیرد. با تحلیل مساله و درک نیازهای stakeholderها، امکان بازبینی و تصحیح سند vision، همچنین طراحی اولیه مدل use case و خصوصیات نیازمندی ها ممکن می گردد. در نتیجه اضافه کردن مشخصات تکمیلی و تعریف کامل تر use case ها و actorها از دیگر فعالیت های این قسمت می باشد. در این بخش نیز می توان از تکنیک های Use case workshops و Storyboard استفاده نمود.

د - مدیریت محدوده سیستم

نیازمندی های یک پروژه، محدوده آن را مشخص می سازند. رمز موفقیت در مدیریت خوب یک پروژه، ایجاد هماهنگی بین اهداف پروژه و منابع موجود (زمان، افراد و سرمایه) می باشد. مدیریت طرح نهایی پروژه یک فعالیت دائمی است که در آن، طرح نهایی به بخش های کوچک تری شکسته می شود تا به راحتی قابل مدیریت باشند. فعالیت های این مرحله فقط در صورتی که اطلاعات جمع آوری شده قبلی پاسخگوی تمام درخواست های سیستم نباشند، اجرا می گردند. برای مدیریت و تصحیح طرح نهایی، بررسی خصوصیات نیازمندی ها لازم می باشد و برای بررسی صحیح خصوصیات، از use case ها و actorها استفاده می گردد. در حقیقت



درک بهتر عملکرد سیستم از توصیف use case ها و actorها و مشخصات تکمیلی بیشتر به دست می آید که این درک بهینه به تحلیل گر سیستم امکان اضافه کردن پارامترهایی مانند اولویت، هزینه و ریسک را به خصوصیات نیازمندی ها می دهد و طراح سیستم را قادر می سازد تا use case های اصلی سیستم را توصیف نماید.

هـ- اصلاح تعریف سیستم

پس از آن که طرح نهایی تهیه شد فعالیت در این بخش


آغاز می گردد.

اهداف این قسمت عبارتند از: شرح جزئیات رویدادهای مربوط به هر use case، بررسی جزئیات مشخصات تکمیلی و مدل سازی و طراحی اولیه واسط کاربر. فعالیت این بخش بر پایه use case های طراحی شده، actorهای مشخص شده و درک محدوده پروژه که در vision بیان شده بود آغاز می شود. در این مرحله، طراحان use case ها و واسط های کاربر می بایست برای توصیف جزئیات نیازمندی های سیستم با یکدیگر همکاری نزدیک داشته باشند.

و - مدیریت نیازمندی های متغیر

اهداف زیر در مرحله مدیریت نیازمندی های متغیر دنبال می شوند: ارزیابی تغییرات لازم در درخواستها و تعیین تاثیر آن ها بر روی مجموعه نیازمندی های موجود، سازماندهی مدل use case، تنظیم مناسب خصوصیات نیازمندی ها و قابلیت پی گیری (traceability) و حصول اطمینان از این که نتایج به دست آمده از بررسی بخش نیازمندی ها با دید مشتری از سیستم تطابق داشته باشد.

اعمال تغییرات در مجموعه نیازمندی ها طبیعتاً بر روی مدل های تولید شده در بخش تحلیل و طراحی تاثیر می گذارد. از طرفی، پی گیری نیازمندی ها یکی از مسائل مهم در این قسمت می باشد که مرورگران سیستم، اطلاعات لازم برای اعمال تغییرات مناسب در نیازمندی ها را با توجه به ماهیت این تغییرات بدست می آورند.

تیم توسعه (نمایندگان مشتری، stakeholder، خبره دامنه و دیگران) می بایست درک مناسبی از دامنه مساله و مشکلات تکنیکی پروژه داشته، همچنین مهارت های لازم در زمینه مدیریت نیازمندی ها و مدل سازی use case را کسب کرده باشند. 

منابع:

- 1- A classic that should be read and re-read by everyone involved in software development. I recommend the 20-year anniversary edition rather than the original 1975 edition. DAV95 Alan Davis 1995. 201 Principles of Software Development, New York: McGraw-Hill.
- 2- JAC98 Ivar Jacobson, Grady Booch, and James Rumbaugh 1998. The Unified Software Development Process, Addison Wesley Longman.
- 3- AND96 Stephen J. Andriole 1996. Managing Systems Requirements: Methods, Tools, and Cases. McGraw Hill
- 4- LEF99 Dean Leffingwell Don Widrig 1999. Effective Requirements Management. Addison-Wesley Longman, Inc.
- 5- IEEE93 IEEE Std 830-1993, Recommended Practice for Software Requirements Specifications, Software Engineering Standards Committee of the IEEE Computer Society, New York, 1993,....