

# اهمیت و جایگاه مهندسی نیازمندیها در راستای نگهداری نرم افزار

سید سعید علی اکبر نژاد<sup>1</sup>، سید مهداد شجاعی برهان<sup>2</sup> و علی اکبر ارجمند هاشجین<sup>3</sup>

<sup>1</sup>دانشگاه آزاد اسلامی واحد زنجان ، saeed\_aliakbarnejad@yahoo.com

<sup>2</sup>دانشگاه آزاد اسلامی واحد زنجان ، mahdad0808@yahoo.com

<sup>3</sup>دانشگاه آزاد اسلامی واحد زنجان ، a.a\_arjomand@yahoo.com

چکیده-امروزه برای تولید یک نرم افزار کاربردی و مناسب می بایست به صورت کاملاً تعیین شده و مشخص عمل کرد. در چرخه حیات نرم افزار تمامی فازها به صورت دقیق و علمی انجام می شوند، تا هزینه های فاز نگهداری تا حد بسیار زیادی به کمترین مقدار خود برسد. هدف ما در این مقاله نشان دادن اهمیت مهندسی نیازمندیها در راستای نگهداری نرم افزار می باشد.

کلید واژه-مهندسی نیازمندیها ، کیفیت نرم افزار ، نگهداری نرم افزار ، چرخه حیات نرم افزار

## 1-مقدمه

فاز نیازمندی ها در چرخه حیات نرم افزار ، آنچنان فاز پراهمیتی تلقی میشود که بدان مهندسی نیازمندی ها گفته می شود. یعنی این فاز نیازمند بررسی دقیق ، فنی و مهندسی می باشد و باید بر اساس اصول کاملاً مشخص در یک چهارچوب معین انجام گیرد. به عبارت کلی این فاز مانند پایه و اساس چرخه حیات نرم افزار می باشد. به این صورت که هر قدر در این فاز با دقت و به صورت علمی عمل شود ، در فازهای بعد کمتر دچار مشکل خواهیم شد. مهندسی نیازمندی ها در واقع به معنی شناسایی ، مدل سازی ، مرتبط سازی و مستند سازی نیازمندیها برای یک سیستم خاص می باشد. در مهندسی نیازمندیها به نحوی باید عمل شود که تضمین شود نیازمندیها کامل ، سازگار و مناسب هستند ، تا از دوباره کاریهای پرهزینه و سرسام آور جلوگیری به عمل آید.

## 2- سطوح و انواع نیازمندیها

بیشتر کسانی که در زمینه ی نرم افزار فعالیت میکنند فقط در مورد نیازمندیها صحبت میکنند. در حالی که سطوح مختلف و انواع نیازمندیها وجود دارد که باید شناسایی شوند . در شکل زیر روند انجام نیازمندیها نشان داده شده که در سال 2004 توسط " کارل ویگرز" <sup>1</sup> ایجاد شد. در واقع با استفاده از این شکل افراد توانستند فهم بهتری از اینکه چه اطلاعاتی مورد نیاز است تا استخراج ، تحلیل و آنالیز شود و اینکه چه زمانی این پارامترها در مهندسی نیازمندیها باید تعریف شوند . همه این سلسله مراتب و روابط در شکل مشخص میباشد . لایه "تجاری" روی مسائل کاری و تجاری که قرار است حل شود یا فرصت های تجاری که قرار است به وسیله محصول نرم افزاری آدرس دهی شود ، کار می کند. به طور کلی این لایه مشخص میکند که "چرا نرم افزار باید ایجاد شود؟" معمولاً به نیازمندیهای تجاری ، "هدف مشتری" <sup>2</sup> اطلاق می شود.

"سخت ترین بخش ساختن یک سیستم نرم افزاری ، تصمیم گیری دقیق در مورد این است که چه چیزی می خواهیم بسازیم . هیچ بخشی از کار ادراکی (شامل واسط افراد ، ماشین ها ، مکانیزم ها و سایر سیستم های نرم افزاری) به سختی ثبت کردن جزئیات نیازمندیها نیست. هیچ بخشی از کار اگر درست انجام نشود ، مانند این بخش نتیجه کار را معیوب نمی کند.تصحیح هیچ بخشی مانند این بخش سخت و پرهزینه نیست. "

نیازمندیهای مصرف کننده یک نگاه عاملیتی از منظر مصرف کننده به محصول نرم افزاری خواهد داشت. این قسمت با این مسئله روبرو است که " چه کاری را نرم افزار باید انجام دهد تا مصرف کننده به هدف خود برسد؟" ممکن است چندین سطح از مصرف کننده نیاز باشد تا یک نیازمندی تجاری به صورت کامل انجام شود. بر خلاف نیازمندیهای تجاری ، قوانین تجاری سیاستهای مشخص ، استاندارد و منظم هستند و به اینکه مصرف کننده ها چطور باید کار را انجام دهند کمک می کنند. خصوصیات کیفی در این لایه یک مشخصه غیرتابعی می باشد که کیفیت محصول نرم افزاری را مشخص می کند. مشخصه های کیفی شامل : قابلیت اطمینان ، قابلیت دسترسی ، امنیت ، قابلیت نگهداری ، قابلیت حمل ، قابلیت استفاده و خصوصیات دیگر می باشد.

نقل قولی از "فردریک بروکس" بیان کننده این است که چرا نیازمندیها اینقدر مهم می باشند.

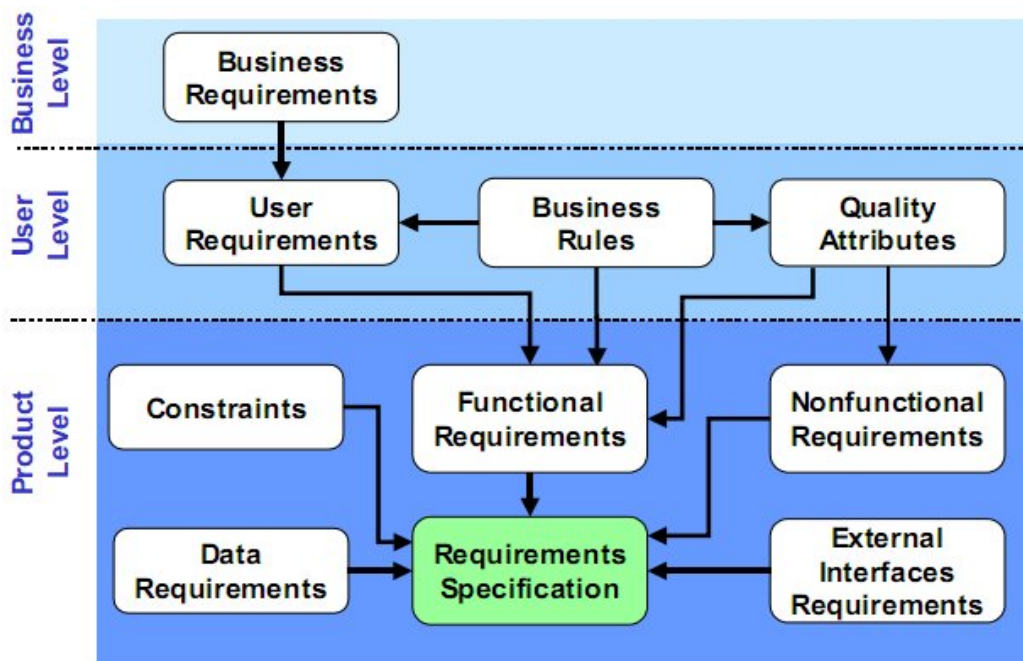


Figure 1 Levels and types of requirements

اگر نیازمندیها ناقص باشد ، نرم افزار تمام نیازهای کاربر و مشتری را برآورده نخواهد کرد.همان طور که در شکل می بینید "نوریتاکی کانو"<sup>3</sup> مدلی را در مورد رابطه بین رضایت مشتری و نیازمندیهای کیفی بیان کرده است.

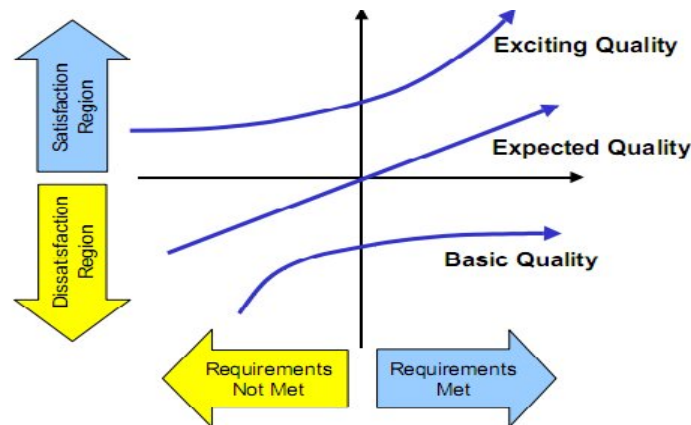


Figure 2 Kano model for quality requirements

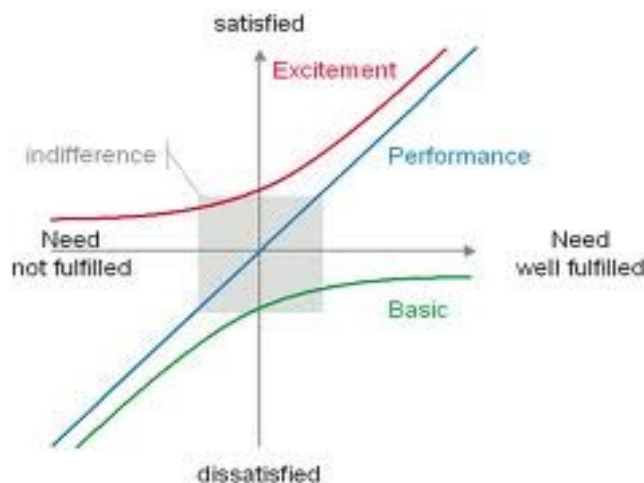
آنالیز اطلاعات و نیازمندیها دقت کنند. خطا در نیازمندیها 70٪ تا 85٪ از هزینه های دوباره کاری در یک پروژه نرم افزاری را شامل می شود . فرض کنید یک نقص یا اشکال در قسمت نیازمندیها تشخیص داده شود و اصلاح نشود. اگر اصلاح این نقص یک واحد هزینه داشته باشد ، بعدها در چرخه حیات نرم افزار بیشتر از 100 واحد هزینه خواهد داشت.

یکی دیگر از مسائلی که در نیازمندیها باید به آن توجه داشت عبارت "چه کسی " می باشد. به این معنی که چه گروهی از افراد با محصول نرم افزاری سر و کار دارند. به طور کلی سه گروه برای این مرحله وجود دارد. گروه اول از نوع دریافت کننده می باشند. که خود شامل دو گروه مشتری ها و کاربران می باشند. مشتری به فردی گفته می شود که برای مثال خریدی انجام میدهد. و کاربر به فردی گفته می شود که با سیستم به صورت مستقیم در ارتباط است و از آن گزارش تهیه می کند. گروه دیگر فراهم کننده ی محصول نرم افزاری هستند. که ممکن است شامل افراد یا تیم هایی باشند که عضوی از گروه توسعه دهندگان نرم افزار بوده اند. گروه سوم " دیگر افراد" <sup>5</sup> می

کیفیت پیش بینی شده نشان دهنده ی نیازمندیهای کیفی است که مشتری به صورت واضح و مشخص آنها را بیان کرده است. برای مثال در خرید یک ماشین، ساخت آن، مدل آن ، مصرف سوخت آن همه نیاز است. اگر مشتری از این ویژگیها ناراضی باشد به فروشگاه دیگری میرود. و این زمانی است که مشتری به درستی نیازمندیهای خود را از محصول ما دریافت نمیکند. یک سطح پایه ای از نیازمندیها وجود دارد که مشتری انتظار دارد که محصول مورد نظر حتماً آن را داشته باشد. این نیازمندیها به وسیله مشتری بیان نمی شوند، بلکه ضمنی هستند. برای مثال وقتی قصد خرید یک ماشین را دارید واضح است که ماشین مورد نظر چهار چرخ دارد ، دارای موتور است و... . بعضی از مشخصه ها و قابلیت ها هستند که حتی خود مشتری نمیداند که به آنها نیاز دارد . مثلاً به یاد بیاورید یخچال ها که اولین بار در ماشین ها کار گذاشته شد . توجه کنید که تمام منحنی کیفیت افروزنده <sup>4</sup> در قسمت رضایت قرار دارد . البته توجه داشته باشید که تمامی نوآوریهای امروزی ، انتظارات آینده خواهند بود . برای تولید کنندگان نرم افزار برآورده کردن دو سطح پایینی نیازمندیها آسان خواهد بود ، اگر در دریافت و

روی محصول نرم افزاری پیاده سازی کنند. اگر نرم افزار بخشی از یک سیستم بزرگتر باشد، طراحان سخت افزار و مهندسين بايد توجه زيادي به نيازمنديها داشته باشند. تست کننده هاي نرم افزار از نيازمنديها به عنوان اصول براي ايجاد تست در شرايط تعيين شده استفاده مي کنند. شكل ديگري از نمودار کانو:

باشند. آناليز نيازمنديها كه آناليز سيستم هم ناميده مي شود، مسئول استخراج نيازمنديها از "مشتري ها"، "كاربران" و "ديگر افراد"، آناليز نيازمندي ها، ثبت ويژگي هاي نيازمنديها و ارتباط بين نيازمنديها براي گسترش نرم افزار را بر عهده دارد. طراحان، مسئول ترجمه و تبديل نيازمنديها به معماری نرم افزار هستند و طراحی های جزئی است که مشخص می کند نرم افزار به چه صورت باید پیاده سازی شود. حال مسئولیت با مهندسين و گسترش دهندگان است كه طرحها را



باشد. همان طور که در شکل زیر مشخص است مهندسی نيازمنديها از دو پروسه ي اصلي ساخته مي شود. 1- ايجاد نيازمنديها و اقتباس آن و 2- مدیریت نيازمنديها. ايجاد نيازمنديها شامل تمامی فعاليت هايي است كه به نحوی در استخراج ، تحليل ، تشخيص و تصديق نيازمنديها درگير هستند.

### 3- ساختار مهندسی نيازمنديها

مهندسی نيازمنديهاي نرم افزار يك پروسه منظم و مشخص با اصول مهندسی به منظور تعريف ، مستندسازی و نگهداری نيازمنديهاي نرم افزار در طول گسترش چرخه حیات نرم افزار می

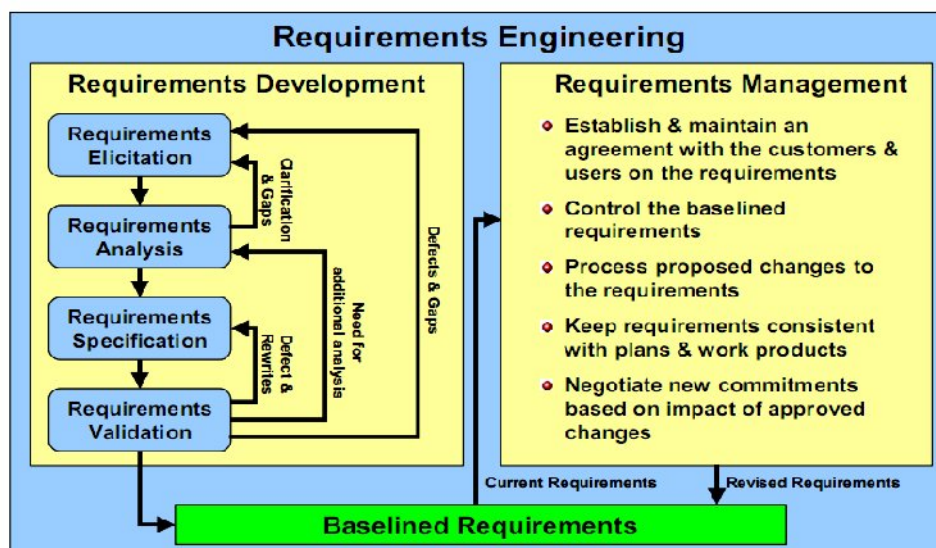


Figure 3 Requirements Engineering Process (based on Wiegiers 2003)

#### 4- نتیجه

محصول را داشته باشند تا در اتمام کار به دلیل قصور درانتخاب روش های مهندسی نیازمندیها ، دچار مشکل نشوند.

مهندسی نیازمندیها یکی از مهم ترین پروسه ها در چرخه حیات یک نرم افزار میباشد.افرادی که در این قسمت فعالیت می کنند باید دید جامعی نسبت به تمامی محیط اطراف نرم افزار ، رابطین و تمام نیازهایی که ممکن است به نحوی با محصول و پروژه در ارتباط باشد را پیشبینی نمایند و آن را در نیازمندیهای خود بگنجانند.اگر خطایی یا نقصی در این قسمت به صورت کامل اصلاح نشود در مراحل بعدی حیات نرم افزار بسیار پرهزینه تر و سخت تر اصلاح خواهد شد . بنابراین افرادی که در طراحی و ایجاد یک محصول نرم افزاری فعالیت می کنند ، باید تفاوت های میان سطوح مختلف نرم افزار را به خوبی درک نمایند تا پروسه مهندسی نیازمندیها را به نحوی کارآمد انجام دهند. این امر مستلزم داشتن درک بالایی از نیازمندیها و استفاده از منابع و تخصیص زمان در طول حیات نرم افزار میباشد. و این نیازمند آن است که متخصصین این امر دید کامل و وسیعی نسبت به تمامی گروه های مرتبط با

پانویس ها

1. Karl Wiegers
  2. customer objective
  3. Noritaki Kano
  4. exciting quality
  5. other skateholders
- [1] Brooks, F. 1995. Mythical man-month: essays on software engineering, 20<sup>th</sup> anniversary edition. Addison-Wesley Professional.
- [2] Gause, D., and G. Weinberg. 1989. Exploring requirements, quality before design. New York: Dorset House Publishing.
- [3] Pyzdek, T. 2000. Quality engineering handbook. New York: Marcel Dekker.
- [4] Wiegers, K. E. 2004. In search of excellent requirements .