

به نام خدا

نام درس: آزمایشگاه مهندسی نرم افزار

دانشجو: معصومه فرج

واژه ترم: وردی بهمن 1403

نام مدرس: استاد محمد احمد زاده

فهرست مطالب:

Session2:

3.Requirements Engineering Report(Word)

1: گزارش _مهندسی نیازمندی ها

2: مهندسی نیازمندی ها چیست؟

3: چه روش ها و مدل هایی وجود داره برای مهندسی نیازمندی ها؟

4: چطور می توان فهمید که یک نرم افزار چه نیاز هایی دارد؟

5: و چطور می توان فهمید چه میزان از نیاز ها شناسایی شده است؟

گزارش مهندسی نیازمندی ها

سندی است که در آن نیازمندی های فنی و عملکردی یک پروژه یا محصول به طور دقیق و ساختارمند تعریف و مستند می شود.

این گزارش معمولاً در مراحل اولیه پروژه های مهندسی تهیه می شود و به عنوان مرجع اصلی برای طراحی توسعه و ارزیابی

محصول یا سیستم مورد استفاده قرار می گیرند.

اجزای اصلی گزارش مهندسی نیازمندی ها:

1. عنوان و مقدمه

2. دامنه پروژه

3. نیازمندی های عملکردی

4. نیازمندی های غیر عملکردی

5. نیازمندی های فنی

6. نیازمندی های امنیتی

7. نیازمندی های رابط کاربری

8. نیازمندی های قانونی و استانداردها

9. نیازمندی های نگهداری و پشتیبانی

10. نیازمندی های تست و اعتبار سنجی

11. ریسک ها و محدودیت ها

12. زمینه ها

مهندسی نیازمندی ها چیست؟

مهندسی نیازمندی ها فرآیندی است که در آن نیاز های ذینفعان یک سیستم یا محصول شناسایی، تحلیل، مستند سازی و مدیریت می شود.

این فرآیند نقش کلیدی در توسعه نرم افزار سیستم های مهندسی و پروژه های فنی ایفا می کند و هدف آن اطمینان از این است که محصول نهایی نیاز های کاربران و ذینفعان را به طور کامل و دقیق برآورده کند.

مهندسی نیاز مندی ها یک فرآیند چالش بر انگیز اما ضروری است که موفقیت پروژه های فنی و نرم افزاری را تضمین کند.

مراحل اصلی مهندسی نیاز مندی ها:

1. جمع اوری نیاز مندی ها:

شناسایی و جمع اوری نیاز های ذینفعان از طریق روش هایی

مانند: مصاحبه، نظر سنجی، مشاهده طوفان فکری و کارگاه های آموزشی

2. تحلیل نیازمندی ها:

بررسی نیاز مندی ها جمع اوری شده برای درک بهتر، رفع تناقضات، اولویت بندی و تعیین امکان پذیری.

3. مستند سازی نیازمندی ها:

ثبت نیاز ها به صورت دقیق و ساختارمند.

SRS-Software معمولا در قالب سند نیاز مندی ها

Requirements specification

4. اعتبار سنجی نیاز مندی ها:

بررسی و تایید نیاز ها برای اطمینان از صحت کامل بودن و قابلیت اجرا. این مرحله ممکن است

شامل: بازبینی، تست، و شبیه سازی باشد

5. مدیریت نیاز مندی ها

نظارت بر تغییرات نیاز ها در طول چرخه عمر پروژه و اطمینان از اینکه تغییرات به درستی اعمال و مستند شود.

انواع نیاز مندی ها:

1. نیاز مندی های عملکردی:

توضیح می دهند که سیستم چه کاری باید انجام دهد.

2. نیاز مندی های غیر عملکردی:

مشخصات کیفی سیستم مانند: کارایی، امنیت، قابلیت اطمینان و ...

3. نیاز مندی های کسب کار:

اهداف و نیاز های کلی کسب و کار.

4. نیاز مندی های کاربردی:

نیاز های خاص کاربران نهایی سیستم

اهمیت مهندسی نیاز مندی ها:

1. کاهش ریسک:

شناسایی زود هنگام نیاز ها و رفع تناقضات باعث کاهش مشکلات در مراحل بعدی توسعه می شود.

2. صرفه جویی در هزینه و زمان:

جلو گیری از تغییرات دیر هنگام و باز طراحی های پر هزینه.

3. رضایت ذینفعان:

اطمینان از اینکه محصول نهایی نیاز های کاربران را برآورده می کند.

4. پایه ای برای طراحی و تست:

نیاز مندی های واضح و دقیق پایه برای طراحی سیستم و تست آن فراهم می کنند.

چه روش ها و مدل هایی وجود دارد برای مهندسی نیاز مندی ها؟

1. روش ها و جمع اوری نیاز مندی ها

مصاحبه: گفت گو با ذینفعان برای درک نیاز های آن ها.

پرسش نامه:

استفاده از فرم های پرسش نامه برای جمع اوری اطلاعات از تعداد زیادی از ذینفعان.

مشاهده:

مشاهده مستقیم کاربران در محیط کار برای درک نیاز های آن.

کارگاه های نیاز مندی :

جلسات گروهی با حضور ذینفعان برای بحث و تبادل نظر در مورد نیاز ها.

تحلیل اسناد:

بررسی اسناد موجود .مانند:گزارش ها.دستور العمل ها و مستندات سیستم قبلی

2.روش ها و تحلیل نیاز مندی ها

تحلیل هدف:

تمرکز بر اهداف ذینفعان و نحوه دستیابی به آن ها

تحلیل تاثیر ذینفعان :

شناسایی و تحلیل نیاز ها و انتظارات ذینفعان مختلف .

تحلیل ریسک:

شناسایی و ارزیابی ریسک های مرتبط با نیازمندی ها

تحلیل قابلیت ردیابی:

اطمینان از اینکه نیازمندی ها قابل رد یابی و پیگیری هستند.

3.روش های مستند سازی نیازمندی ها

سند نیازمندی های نرم افزاری:

مستند سازی جامع نیازمندی های عملکردی و غیر عملکردی.

ماتریس ردیابی نیاز مندی ها:

ایجاد ماتریسی که نیازمندی ها را به طراحی .تست و پیاده سازی مرتبط می کند .

داستان های کار بردی :

مستند سازی نیازمندی ها به صورت داستان های کوتاه و ساده.

4.مدل های مدل سازی نیازمندی ها

مدل های سناریو

scenario Models:

User Stories از سناریو ها داستان های کاربری

برای توصیف تعاملات کار بر با سیستم

مدل های مورد کاربری:

توصیف تعاملات بین کاربران سیستم از طریق نمودار های مورد کاربری

مدل های داده ای:

ERD Entity-Relationship Diagrams استفاده از نمودار های

برای مدل داده ها و روابط بین ان ها .

مدل های رفتاری:

State Diagrams استفاده از نمودار حالت

Activity Diagrams نمودار های فعالیت

برای توصیف رفتار سیستم

مدل های ساختاری :

استفاده از نمودار های کلاس برای نمایش ساختار سیستم.

چطور می توان فهمید که یک نرم افزار چه نیاز هایی دارد؟

برای فهمیدن یک نرم افزار باید این مراحل بررسی کنیم

1. مستندات نرم افزار بررسی کنی

مستندات رسمی: اولین و بهترین منبع برای فهم نیاز های نرم افزار مستندات رسمی ان است. این مستندات معمولا شامل بخشی به

نام نیازمندی های سیستم هستند. که مشخص می کند نرم افزار به چه سخت افزار سیستم عامل، کتابخانه ها یا نرم افزار های

جانبی نیاز دارد

راهنما نصب:

در این بخش معمولا نیازمندی های اولیه مانند: نسخه های مورد نیاز سیستم عامل فضای دیسک، رم پردازنده ذکر شده اند.

2. وب سایت رسمی نرم افزار:

System Requirements در وب سایت رسمی نرم افزار بخشی به نام

یا نیازمندی ها دارد که اطلاعات دقیقی درباره نیاز های سخت افزاری و نرم افزاری ارائه می دهد

3. استفاده از ابزار های تحلیل سیستم

SPU-Z یا speccy ابزار های تشخیص سخت افزار: مانند

مشخصات سیستم بررسی کنند و با نیازمندی های نرم افزار مقایسه شوند.

بررسی سازگاری: برخی نرم افزار ها ابزار های داخلی برای بررسی سازگاری سیستم شما دارند .

این ابزار ها قبل نصب سیستم شما را بررسی می کنند و در صورت عدم تطابق هشدار می دهند.

4. تست نصب آزمایشی:

اگر نرم افزار دالود کرده اید می توانید ان را نصب کنید و ببینید آیا سیستم شما قادر به اجرای ان است یا خیر .

اگر سیستم شما ضعیف تر از نیازمندی ها باشد ممکن است نرم افزار به درستی اجرا نشود یا با خطا مواجه شود.

5. بررسی نسخه های مختلف نرم افزار:

برخی نرم افزار ها نسخه های سبک تر قدیمی تری دارند که نیاز مندی های سخت افزاری کمتری دارند .

اگر سیستم شما قدیمی تر است می توانید این نسخه ها را بررسی کنید.

6. نمایش با پشتیبانی:

اگر اطلاعات کافی در مستندات یا وب سایت پیدا نکردید می توانید با پشتیبانی نرم افزار تماس بگیرید و سوالات خود را مطرح کنید.

نیازمندی های رایج نرم افزارها:

سیستم عامل: ویندوز لینوکس و نسخه های خاص آن ها.

حداقل سرعت و تعداد هسته ها.

پردازنده (CPU)

حداقل و توصیه شده.

حافظه (RAM)

فضای ذخیره سازی: مقدار مورد نیاز برای نصب و اجرا

کارت گرافیک:

CPU نیاز به خاص

سایر نرم افزار ها:

نیاز به نصب کتابخانه ها یا فریم ورک های خاص

Java NET Framework مثل

با دنبال کردن این مراحل می توانید نیاز های یک نرم افزار را به طور دقیق شناسایی کرد.

و چطور می توان فهمید چه میزان از نیاز ها شناسایی شده است؟

برای فهمیدن میزان نیاز های شناسایی شده مراحل زیر بررسی می کنم

1. جمع اوری اطلاعات:

ابتدا اطلاعات مربوط به نیاز ها را از منابع مختلف (مصاحبه، نظرسنجی، تحلیل داده ها...) جمع اوری کنید

2. دسته بندی نیاز ها:

نیاز ها بر اساس معیار هایی مانند اولویت حوزه کاری یا نوع نیاز دسته بندی کنید.

3. تحلیل نیاز ها :

با استفاده از ابزار های تحلیلی مانند ماتریس اولویت بندی یا تحلیل نیاز ها را ارزیابی می کند.

4.مقایسه با اهداف:نیاز های شناسایی شده را با اهداف کلی پروژه یا سازمان مقایسه کنید تا مطمئن شوید که همه اهداف پوشش داده شده اند.

5.بازخورد گرفتن:

از ذینفعان و کاربران نهایی بازخورد بگیرید تا مطمئن شوید نیاز هایشان به درستی شناسایی شده است.

6.برسی مستر:

نیاز ها ممکن است با گذشت زمان تغییر کنند بنابراین باید به طور مستر آنها را بروزرسانی کنید.

این مراحل میتوانند میزان نیاز های شناسایی شده را ارزیابی و اطمینان حاصل کنید که همه نیاز ها به درستی در نظر گرفته شده اند.

منابع :

- Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach. McGraw-Hill
- Sommerville, I. (2015). Software Engineering. Pearson
- IEEE Software Engineering Standards