تمرین دوم درس مهندسی نرم‌افزار

گروه پنجم

اعضای گروه:

امیرحسین فراهانی 97106154
سارا آذرنوش 98170668
پیمان حاجی محمد 98170776
محمدعلی حسین نژاد 98170787
رویا قوامی 98171031

.1

طوفان فکری (Brain Storming)

طوفان فکری یک فعالیت گروهی است که شامل تولید ایده ها و راه حال هایی برای نیاز های سیستم نرم افزاری می باشد. به طور کلی، برای شناسایی راه حل های ممکن برای مشکلات و روشن کردن جزئیات فرصت هایی که ممکن است وجود داشته باشد استفاده می شود. افراد گروه تشویق می شوند که ایده های خود را به اشتراک بگذارند و بر اساس پیشنهادات یکدیگر نظرهای بیشتری ارائه دهند. معمولا این جلسات توسط فردی که بتواند بحث را هدایت کند و ایده ها را روی تخته یا چارتی بنویسد مدیریت می شود.

نمونه سازی اولیه (Prototyping)

نمونه سازی اولیه، شامل ایجاد یک مدلی که کار می کند از سیستم نرم افزاری است تا از ذینفعان و کاربران بتوان بازخورد جمع آوری کرد. نمونه اولیه را می توان برای آزمایش عملکرد، قابلیت استفاده و عملکرد سیستم استفاده کرد. بازخورد از نمونه اولیه می تواند برای اصلاح نیازهای سیستم استفاده شود.

مشاهده (Observation)

این تکنیک شامل مشاهده کاربران و ذینفعان در هنگام تعامل با سیستم موجود یا انجام وظایف خود است. مشاهده را می توان به صورت حضوری یا از راه دور با استفاده از نرم افزار فیلمبرداری یا ضبط صفحه انجام داد. مشاهدات می تواند بینش ارزشمندی در مورد رفتار و نیازهای کاربر ارائه دهد.

نظرسنجی (Survey)

نظرسنجی روشی است برای جمع آوری نیازمندی ها از طریق پرسشنامه هایی که بین ذینفعان، کاربران و کارشناسان موضوع توزیع می شود. نظرسنجی ها برای جمع آوری اطلاعات در مورد الزامات، اولویت ها و انتظارات سیستم طراحی شده اند.

مصاحبه (Interview)

این تکنیک شامل انجام مصاحبه های انفرادی یا گروهی با ذینفعان، کاربران و کارشناسان موضوع برای جمع آوری الزامات است. مصاحبه کننده برای به دست آوردن اطلاعاتی در مورد ویژگی ها، عملکرد و عملکرد سیستم، سؤالاتی باز خواهد پرسید.

.2

- روش MoSCoW

این روش برگرفته از چارچوب DSDM است و باعث به وجود آمدن یک درک مشترک بین تمام ذی‌نفعان می‌شود. این روش به طور کلی کارها را به چهار دسته must have، Should have، could have، won't have this time تقسیم میکند.

must have: این دسته شامل کارهایی است که نیاز ضروری پروژه هستند و باید حتما انجام شوند

should have: شامل کارهایی است که برای پروژه مهم هستند اما ضرورت خیلی بالایی برای رسیدن آن‌ها به یک ددلاین مشخص وجود ندارد.

could have: با انجام این دسته از کارها شما منافعی به دست خواهید آورد، اما با انجام ندادن آن‌ها ضرری متوجه شما نخواهد بود

won’t have: شما در حال حاضر نیازی به انجام این کارها ندارید و انجام آن‌ها را می‌توانید به آینده موکول کنید

- RICE

این کلمه سرآیند واژه‌های Reach,Impact,Confidence,Effort است. برای هر کار تیم باید روی یک عدد برای هر کدام از این موارد به توافق برسد که در زیر به تفصیل توضیح می‌دهم.

Reach: باید مشخص کنیم که این کار تا چه میزان به دست افراد استفاده کننده خواهد رسید. اعدادی مثل mau یا همان کاربران فعال ماهانه می تواند نشانگر خوبی برای این عدد باشد.

Impact: این شاخص نشانگر میزان اهمیت این پروژه است. به ترتیب عدد ۳ در حالتی که کار تاثیر بسیار زیادی داشته باشد. عدد ۲ برای کارهایی که تاثیر زیادی دارند. ۱ برای کارهایی که تاثیر متوسط دارند. ۰.۵ برای کارهایی که تاثیر کمی دارند و ۰.۲۵ برای کارهایی که تاثیر بسیاری کمی دارند.

Confidence: این عدد نشانگر این است که ما چقدر از میزان تاثیر اطمینان داریم. ۱۰۰ برای اطمینان کامل، ۸۰ برای اطمینان متوسط و ۵۰ برای اطمینان کم.

effort: این عدد به صورت فلان قدر نفر ماه یا فلان قدر نفر ساعت یا به همین شکل تعریف می شود. مثلا ۱۰ نفر/روز یعنی یک نفر به مدت ده روز روی یک پروژه کار کند یا دو نفر به مدت ۵ روز.

بعد از اینکه برای هر کار اعداد بالا محاسبه شدند، امتیاز نهایی هر کار از طریق فرمول زیر محاسبه می‌شود. امتیاز بالاتر به معنای اولویت بالاتر است:

Rice = (Reach * Impact * confidence)/ effort

از مزایای این روش می توان به جامعیت بالای اون، مبتنی بودن بر اعداد مشخص و اهمیت زیادی برای مشتری نام برد. این روش معایبی مانند زمانبر بودن، مبتنی بودن بر داده‌هایی که ممکن است فراهم کردن آن‌ها راحت نباشد و عدم اذعان نسبت به مسئولیت شخص در این کار اشاره کرد.

- ABCDE

در این روش هر کار در یکی از حالت‌های زیر دسته بندی می‌کنیم

A: کارهای بسیار مهمی که باید حتما به آن‌ها توجه کنیم و در غیر این صورت از پیامدهای آن در امان نخواهیم بود

B: کارهایی که مهم هستند و باید به آن‌ها توجه کنیم اما ضروری ضروری نیستند

C: کارهایی که بهتر است انجام شوند

D: کارهایی که می‌توانیم مسئولیت آن‌ها را به افراد دیگر واگذار کنیم و برای ما سرباری نخواهند داشت.

E: کارهایی هستند که شما می‌توانید آن‌ها را حذف کنید

در این روش شما به ترتیب از بالا به پایین باید بر روی کارها وقت بگذارید و با اهمیت ترین آن‌ها را ابتدا انجام دهید.

- ICE

در این روش شما بر اساس سه پارامتر Impact،confidence و Ease تصمیم گیری را برای اولویت هر کار انجام می‌دهید. شما باید به هر پارامتر از یک تا ده نمره بدهید. من در ادامه هر پارامتر را بیشتر توضیح می‌دهم

Impact: این کار چقدر تاثیرگذار خواهد بود و چقدر می‌تواند در رسیدن به هدف نهایی پروژه تاثیرگذار باشد

Confidence: این کار با چه قطعیتی موفق خواهد بود؟ اگر قطعا موفق می‌شود عدد ده و در غیر این صورت عدد یک

Ease: این کار چقدر آسان خواهد بود؟ چه میزان تلاش برای تکمیل آن لازم است؟ آسانترین ۱۰ و سخت ترین یک را خواهند گرفت.

حال شما با جمع زدن این اعداد می‌توانید امتیاز هر کار را برای اولویت بندی بفهمید. مشکل اصلی این روش در این است که بسیار انتزاعی است و در بسیاری از موارد ما امکان تخصیص یک عدد دقیق را نخواهیم داشت و در نتیجه امتیاز نهایی می تواند خطای زیادی داشته باشد

- WSJF (Weighted Shortest Job First)

در این روش چابک که به بهترین وجه در شرکت‌های سایز متوسط تا بزرگ استفاده می‌شود، ما امتیاز هر کار را با تقسیم هزینه تأخیر بر طول مدت کار تخمین می‌زنیم.

cost of delay: هزینه تاخیر بیانگر این است که چه میزان سود را با انجام ندادن این کار از دست خواهیم داد

job duration: مدت زمان انجام کار را بر اساس دسته‌بندی زیر مشخص می‌کند.

- کار ساده با ارزش افزوده بالا
- کار ساده با ارزش افزوده پایین
- کار پیچیده با ارزش افزوده بالا
- کار پیچیده با ارزش افزوده پایین.

طیبعتا در این دسته بندی ما ترجیح می‌دهیم کار با ارزش افزوده بالاتر و زمان کمتر را انتخاب کنیم. ما می‌تواین نموداری رسم کنیم که میزان ارزش خلق شده هر کار محور عمودی باشد و میزان زمان لازم محور افقی آن، سپس به وضوح می‌توانیم تصمیم بگیریم که کدام کار برای انجام دادن بهتر است.

.3

- قابلیت ردیابی از دو کلمه یعنی ردیابی و توانایی تشکیل شده است. ردیابی به معنای یافتن کسی یا چیزی و قابلیت به معنای مهارت یا توانایی یا استعداد برای انجام کاری است. بنابراین، قابلیت ردیابی صرفاً به معنای توانایی ردیابی نیاز، ارائه کیفیت بهتر، یافتن هرگونه خطر، نگهداری و تأیید سوابق تاریخچه و تولید یک کالا یا محصول با استفاده از شناسایی مستند است.

به همین دلیل، برای تامین کنندگان آسان است که در صورت یافتن هرگونه خطر یا مشکلی را کاهش دهند و کیفیت کالا یا محصول را بهبود بخشند. بنابراین، داشتن قابلیت ردیابی به جای عدم قابلیت ردیابی مهم است. استفاده از قابلیت ردیابی، یافتن الزامات و هرگونه خطر برای بهبود کیفیت محصول بسیار آسان می شود.

چند نوع آن شامل:

- ☐ Source traceability
- ☐ Requirements traceability
- ☐ Testing traceability
- ☐ Risk traceability
- ☐ Quality traceability

-

☐ درک نیاز

قبل از اینکه بتوانید الزامات طراحی نرم افزار را درک کنید، ابتدا باید دانش معقولی در مورد کاربران و اصول اساسی برای الزامات کسب کنید. تیمی که الزامات یک پروژه را مدیریت می کند باید بتواند یک نیاز را به نیازی که جزء ضروری پروژه پیشنهادی است ردیابی کند. با بررسی هر نیاز، می توانید نیازهای از دست رفته را در مراحل اولیه طراحی یا اجرا شناسایی کنید. قابلیت ردیابی نیازمندی‌ها همچنین به شما امکان می‌دهد نیازهای اضافی را که واقعاً مورد نیاز نیستند، شناسایی کنید.

- ☐ پیش بینی تغییرات

در مرحله طراحی، قابلیت ردیابی نیازمندی‌ها به شما امکان می‌دهد قبل از طراحی مجدد سیستم، اتفاقاتی را که هنگام اعمال تغییرات انجام می‌شود، پیگیری کنید. مدیران پروژه باید بتوانند پیش بینی کنند که چه اتفاقی باید بیفتد تا یک شرکت با موفقیت خود را با این تغییر وفق دهد. وجود اقدامات موثر ردیابی به شما درک بهتری از انواع تغییرات مورد نیاز می دهد. علاوه بر این، ردیابی نیازمندی ها در مراحل مختلف نشان می دهد که آیا یک نیاز با موفقیت مورد توجه قرار گرفته است یا اینکه یک نیاز نیاز به آزمایش مجدد دارد.

- ☐ ساده سازی مرحله آزمایش

قابلیت ردیابی به تیم پروژه در تعیین اینکه در چه زمینه هایی الزامات باید آزمایش شوند، کمک می کند. آزمایش هر نیاز فرآیندی زمان بر و پرهزینه خواهد بود و این رویکرد خاص را غیرعملی می کند. آزمایش معمولاً بر اساس خطر بروز مشکل و همچنین تأثیر آن بر سازمان در صورت بروز مشکل خاص است. در بیشتر موارد، الزامات دارای اولویت بالا، مواردی هستند که در موارد آزمایشی یا آزمایشات ردیابی می شوند.

- ☐ تضمین موفقیت پروژه

ردیابی الزامات به جلوگیری از اجرای الزامات غیر ضروری کمک می کند. کمک به تضمین تکمیل پروژه؛ به کنترل هزینه ها کمک می کند؛ و از تاخیر پروژه جلوگیری می کند. به طور کلی، قابلیت ردیابی برای جلوگیری از نتیجه ضعیف پروژه کار می کند. در طول فرآیند توسعه یک پروژه، قابلیت ردیابی تضمین می کند که منابع کافی از زمان، نیروی انسانی و پول برای کدنویسی، آزمایش و تأیید الزامات پروژه در دسترس خواهد بود.

- ☐ اسناد انطباق با مقررات

اگر پروژه مشمول مقررات یا خط مشی شرکت است، ردیابی الزامات می تواند کمک کند تا در جهت رعایت همه مقررات تلاش کنید. اگر فهرستی از الزامات هر مقررات دارید، می توانید به یاد داشته باشید که همه آنها را بگنجانید. در برخی از صنایع، الزامات قانونی برای ثبت قابلیت ردیابی الزامات در طول تولید و گزارش نتایج وجود دارد تا اطمینان حاصل شود که محصولاتی مانند اتومبیل، تجهیزات

صنعتی، هواپیما و سایر ماشین‌آلات قدرتمند دارای نرم‌افزار تست‌شده و ایمن هستند.

- ماتریس ردیابی سندی است که هر سند دو پایه ای را که برای بررسی کامل بودن رابطه نیاز به یک رابطه چند به چند دارند، به هم مرتبط می کند.

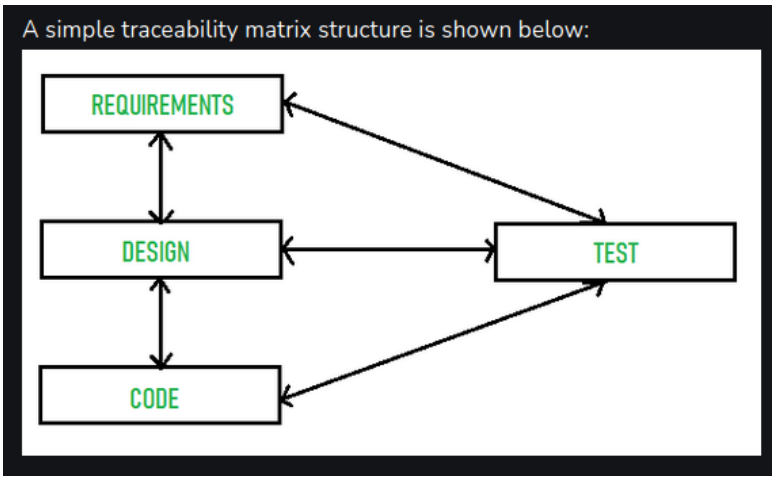
برای پیگیری الزامات و بررسی برآورده شدن الزامات فعلی پروژه استفاده می شود.

موارد مورد نیاز:

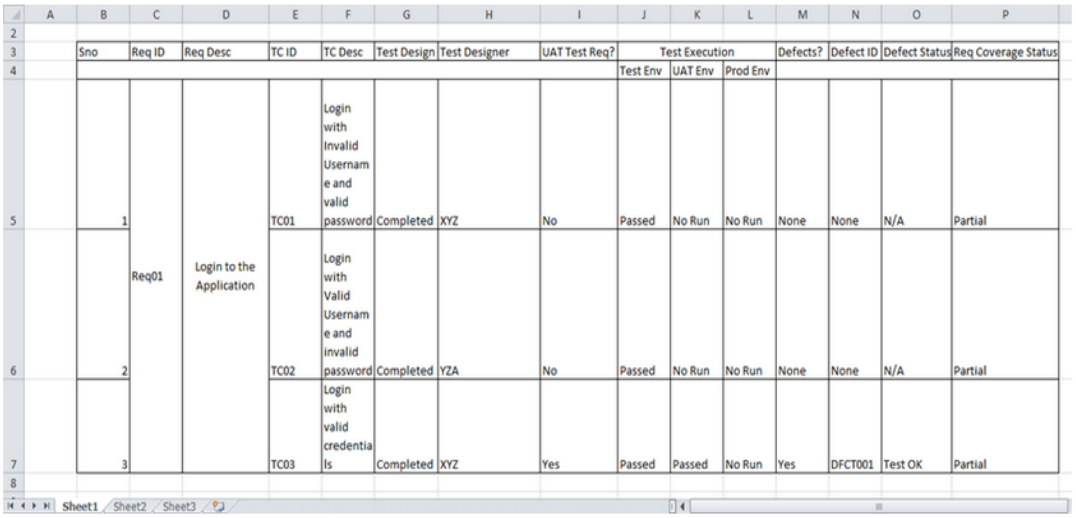
- ☐ شناسه مورد نیاز
- ☐ نوع و شرح نیاز
- ☐ موارد تست با وضعیت

معمولا شامل موارد بیشتری مانند موارد زیر است:

- ☐ پوشش مورد نیاز را در تعداد موارد آزمایش نشان دهد
- ☐ وضعیت طراحی و همچنین وضعیت اجرا برای مورد آزمایشی خاص اگر آزمایش پذیرش کاربر توسط کاربران انجام شود، وضعیت UAT نیز می‌تواند در همان ماتریس ثبت شود.
- ☐ عیوب مربوط به آن و وضعیت فعلی را نیز می توان در همین ماتریس ذکر کرد.



Req No	Req Desc	Testcase ID	Status
123	Login to the application	TC01,TC02,TC03	TC01-Pass TC02-Pass
345	Ticket Creation	TC04,TC05,TC06, TC07,TC08,TC09 TC010	TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run
456	Search Ticket	TC011,TC012, TC013,TC014	TC011-Pass TC012-Fail TC013-Pass TC014-No Run



Sno	Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution			Defects?	Defect ID	Defect Status	Req Coverage Status
								Test Env	UAT Env	Prod Env				
1	Req01	Login to the Application	TC01	Login with Invalid Username and valid password	Completed	XYZ	No	Passed	No Run	No Run	None	None	N/A	Partial
2			TC02	Login with Valid Username and invalid password	Completed	YZA	No	Passed	No Run	No Run	None	None	N/A	Partial
3			TC03	Login with valid credentials	Completed	XYZ	Yes	Passed	Passed	No Run	Yes	DFCT001	Test OK	Partial

.4

a. تحلیل دامنه از جمله فعالیت های چتری در مهندسی نرم افزار می باشد که در آن ویژگی ها و مفاهیم و محدودیت های یک حوزه فعالیتی مورد بررسی قرار گرفته و شناسایی و تعریف می شوند. در تحلیل دامنه اصلی ترین فعالیت ها، identification, analysis, specification می باشند نقش تحلیل دامنه کسب و تجمیع اطلاعات در مورد یک اپلیکیشن در یک دامنه مشخص می باشد. در واقع تمامی قابلیت های مورد نیاز سیستم نرم افزاری را به طور دقیق شناسایی کرده و فهرست بندی کرده و از آن برای فرآیند طراحی و توسعه نرم افزار استفاده می کنیم.

b. هدف از انجام این نوع تحلیل، افزایش reusability در سیستم و درک عمیقی از نیازمندی های آن می باشد. بدین معنا که برای پروژه های جدید تر نیازی به انجام کار های مشابه از قبل انجام شده در سایر پروژه ها نیست و به کمک همان اطلاعات میتوان کار را به اندازه

قابل توجهی جلو انداخت. همچنین تحلیل دامنه، در ارتباط با الگوی نیازمندی ها بدین صورت می باشد که در تحلیل دامنه استخراج ویژگی های یک حوزه انجام می پذیرد و سپس به کمک الگو های نیازمندی و این ویژگی های تعریف شده میتوان به اطلاعات مورد نیاز اولیه برای شروع طراحی سیستم دست پیدا کرد.

5. الف) در مهندسی نرم افزار، دو نوع اصلی از مدل ها وجود دارد: مدل های تحلیلی و مدل های طراحی. مدل های تحلیلی برای درک مسئله ای که نرم افزار قرار است حل کند استفاده می شوند، در حالی که مدل های طراحی برای ایجاد یک راه حل برای آن مسئله استفاده می شوند.

معمولاً مدل های تحلیلی در ابتدای پروژه نرم افزاری ایجاد می شوند و برای درک نیازهای سیستم استفاده می شوند. این مدل ها شامل نمودار های مورد استفاده، نمودار های فعالیت و نمودار های جریان داده و غیره می باشند. هدف این مدل ها، اطمینان حاصل کردن از درک روشن تمام عوامل درباره آنچه که نرم افزار قرار است انجام دهد است.

از سوی دیگر، مدل های طراحی پس از مرحله تحلیل ایجاد می شوند و برای طراحی راه حل نرم افزاری استفاده می شوند. این مدل ها شامل نمودار های کلاس، نمودار های توالی و نمودار های ماشین حالت و غیره هستند. هدف این مدل ها، ایجاد یک برنامه دقیق برای پیاده سازی نرم افزار است.

ب) interaction: مدل interaction از چهار المان اصلی (use-cases, sequence-diagrams, state diagrams, user interface prototype) تشکیل شده است. در این مدل اینترکشن در مدل ها میتوان مشخص نمود. اساسا هنگامی که شناخت کافی از سیستم وجود ندارد protorype میتواند راهشگا باشد. از طرفی در UI و content به sequence diagram نیاز خواهیم داشت. تا بدانیم کاربر با چه بخش‌هایی حرکت میکند تا یک کار رو انجام بدهد. مهم ترین نگرانی ما usabiity هست.

navigation: خیلی از اپلیکیشن‌های فعلی مشکل عدم به توجه به navigation را دارند. بنابراین در این مدل به موارد ذیل توجه میشود: ۱) یکی از اهداف این هست که طراحی المان ها به صورتی باشند که با استپ کمتر و راحت‌تر به هدف مورد نظر برسید.۲) هدف بعدی این هست که برای navigate بهتر باید بعضی از المان‌ها باید بولد بشوند.۳) قدم بعدی این هست که چطوری ارورهای navigatio رو هندل بکنیم؟.۴) باید ترتیب رسیدن به المان‌ها اولویت بندی شوند. ۵) باید نحوه navigation مشخص شود که آیا با لینک هست یا ... ۶) باید navigation هارو سیو کنیم تا راحت‌تر مشتری را بین صفحات navigate بکنیم. ۷) تصمیم گیری درمورد اینکه باتوجه به منابع navigation را برای هرگروه از کاربران تا چه حدی آسان تر کنیم.

ج) علاوه بر اینکه این دو نمودار در المان‌هایی که دارند تفاوت‌هایی دارند، همینطور هدفی که از وجود آن‌ها وجود دارد متفاوت هست. در interaction هدف بررسی تعامل اجزای مختلف سیستم و کاربر میباشد. در حالی که navigation برای بررسی و بهبود مسیر طی شده کاربر و بهبود navigation بین المان‌های مختلف میباشد. درواقع navigation بیشتر در فاز UX میباشد در حالی که interaction کل اجزای سیستم را در بر میگیرد.

6. از جمله نکات مثبت این امر می توان به کاهش هزینه‌های تغییر و زمان مورد نیاز برای ایجاد تغییرات در روند توسعه برنامه، مطابقت بیشتر وب اپ با خواسته کارفرما و کاهش احتمال خطا در زمان تحویل برنامه به او اشاره کرد. همچنین این کار دارای نکاتی منفی ای می باشد که شامل: در شمای کلی اختصاص وقت و منابع به مدل کاربردی می‌تواند باعث افزایش هزینه‌ها و زمان مورد نیاز برای توسعه و تحویل کار بشود و این کار برای یک شرکت کوچک که از حجم کار نسبتا کمی برخوردار است، مقرون به صرفه نیست زیرا هزینه‌های آن برای شرکت مبتدی بسیار بالا خواهد بود.