مهندسی نیازمندی‌ها خانم دکتر سپیده آدابی

علیرضا سلطانی نشان

۱۵ اسفند ۱۴۰۲

فهرست مطالب

۱ مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLv۳ منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می‌دهد.

۲ مهندسی نیازمندی

۱.۲ تعریف

طبق تعریف کتاب پرسمن، نیازمندی‌ها تنها ثابت در حال تغییر می‌باشند. مهندسی نیازمندی مهم‌ترین فاز انجام هر کاری در مهندسی نرم‌افزار می‌باشد. زیرا مشتری دائماً در حال تغییر درخواست‌های خودش است به همین خاطر نیازمندی‌های برآورد شده ملزوم به بروز شدن هستند. هر تغییری که صورت می‌گیرد به دلیل ماهیت پیچیده نرم‌افزار بایستی پایدار [۱] باشد. پایداری به منظور بررسی تغییرات از جوانب مختلف مانند امنیت و آزمون عملکرد صحیح می‌باشد. نیازمندی‌ها کاملا پر دردسر هستند زیرا خیلی از دلایل شکست پروژه‌ها عدم بررسی

[۱] Stable

نیازمندها بوده است. درست است که با آزمون و خطا تجربه به دست می‌آید ولی این تجربه‌ها در پروژه‌های مقیاس بزرگ می‌تواند خطر آفرین باشد چرا که خود تجربه‌ها نیز نیازمند بررسی و آزمون هستند که بتوانیم از آنها در پروژه‌های بعدی یا فعلی خود استفاده کنیم.
دو کلمه اصلی در مهندسی نیازمندی‌ها وجود دارد:

۱. کلمه چه چیزی [۲]: دقیقاً آن چیزی است که سیستم بایستی قادر به انجام آن باشد. مثلاً کاربر باید بتواند در نرم‌افزار لاگین کند.

۲. کلمه چطور [۳]: همانطور که از نامش پیداست چطور انجام شدن کار را تعریف می‌کند. برای مثال بالا می‌توان گفت سیستم لاگین باید کاملاً امن باشد. در این سیستم لاگین کاربران مختلف اعم از استاد، دانشجو و رئیس دانشگاه باید بتوانند زیر پنج ثانیه احراز هویت انجام دهند.

۳. کلمه چه کسی [۴]: عوامل محیطی (افراد، دستگاه‌ها، نرم‌افزارهای آماده) دخیل در برنامه

زمانی که می‌گوییم نرم‌افزار ثبت نام درس، دقیقا بالاترین سطح تجرید [۵] را در نیازمندی بیان کرده‌ایم.

نکات

- مفاهیم کیفی به اندازه مفاهیم اجرایی مهم هستند. درست است نرم‌افزار باید اجرا شود اما این اجرا شدن باید صحیح باشد. امنیت نرم‌افزار خود خواسته می‌تواند تخریب شود، یعنی نرم‌افزاری نوشته می‌شود که می‌تواند ورودی‌های اشتباه و نادرست را بپذیرد، پس در این صورت امنیت و کارایی درست را زیر سوال می‌برد.
- سوال چه چیزی به صورت عملیاتی است و سوال چگونه به صورت غیر عملیاتی
- همیشه باید بین مسائلی که در مهندسی نرم‌افزار پیش می‌آید یک سبک سنگینی [۶] صورت گیرد. معمولاً Benchmarks ها به ما این امکان را می‌دهند. یعنی نرم‌افزار می‌تواند به چند شکل مختلف توسعه پیدا کند اما با گرفتن Benchmark ها می‌توانیم بررسی کنیم که کدام یک از آنها در قسمت عملیاتی و عملکرد صحیح بهتر بوده‌اند. به عبارت دیگر، روش‌ها را نمی‌توان بدون بررسی و با میل شخصی انتخاب کرد، بلکه باید روش‌ها بررسی و سبک سنگین شوند.
- فرایندها در مهندسی نیازمندی را process گویند
- توضیح و بازنویسی نیازمندی‌ها، کار پایه مهندس نیازمندی است.
- تمام مراحل در فرایند به یکدیگر وابسته می‌باشند، فرایند اساساً در مورد جزئیات صحبت نمی‌کند بلکه به ماهیت کلی و تجرید می‌پردازد. برای مثال فرایند جمع‌آوری داده و تحلیل و دیگر مراحل کاملاً به صورت مرحله‌ای و بازگشت پذیر می‌باشد. خروجی فرایند بعد از طی کردن تمام مراحل، نیازمندی را مشخص می‌کند.
- هیچ وقت فرایند با نیازمندی‌ها هم ارز نیست، بلکه نیازمندی خروجی فرایند می‌باشد. در حقیقت به خروجی فرایند، سند نیازمندی یا Requirement Document (RD) می‌گویند.
- در فرایند تکنیک‌ها و استانداردها دیده می‌شود.

۳ نکته تجرید

هر موقع در مورد تجرید صحبت شد، در واقعیت امر میزان سطح پرداختن به جزئیات را توضیح می‌دهد.

---

[۲] What
[۳] How
[۴] Who
[۵] Abstract
[۶] Trade off

۴ متدولوژی

متدولوزی [۷] یک جهان‌بینی کلی، در تولید نرم‌افزار است (دید از بالا برای انجام کارها و وظایف). تمام متدولوزی‌ها را برای تولید استفاده می‌کنند و تمام راهنمایی‌ها توضیحات دارند. در حقیقت تمام متدولوژی‌ها از خواستگاه تولید نرم‌افزار ایجاده شده‌اند و حتی می‌شوند. نکته مهم آن است که فرایندها درون متدولوژی‌ها هستند. متدولوژی یک نقشه است که آن را معمار نرم‌افزار با دیدگاه کاملاً جامع انتخاب می‌کند.

۵ دلیل متدولوژی‌های مختلف

ماهیت و ذات پروژه‌ها متفاوت و پیچده است، پس در این جهت متدولوزی‌های مختلفی برای مهار آنها ارائه شده است که توع تولید را متفاوت می‌کند. متدولوژی بایستی کاملا منعطف باشد. مراحل و فرایندها در متدولوژی‌ها متغیر می‌باشد.

۶ ماهیت مدل

انسان همیشه با خواندن مشکل دارد. خواندن دائماً با مشکلات محاوره‌ای همراه است. محاوره با ابهام همراه است. در پروژه مهندسی نرم‌افزار، وقتی افراد بخواهند با یکدیگر در مورد پروژه صحبت کنند، زبان میان آنها مدل‌های بصری و گرافیکی می‌باشد. افراد بعد از جمع‌آوری اطلاعات و تحلیل آنها، بایستی با آنها به مفهوم بصری برسند تا به کارشناسان دیگر آن را انتقال دهند. به بیانی دیگر، مدل زبان مشترک برای انجام فرایندها، بیان گرافیکی با حفظ سطح تجرید است.

انسان روی جمله‌های ترکیبی مشکل دارد:

$$(A \wedge B) \vee (C) \rightarrow x \quad (۱)$$

یا

$$A \wedge (B \vee C) \rightarrow x \quad (۲)$$

راهکار: استفاده از Decision table که بتوان منطقی به نتیجه رسید.

زبان مدلسازی: ریاضی و گرافیک (بصری)

عملیات به دو دسته تقسیم می‌شوند

۱. عملیات ریاضی: $y = x$

۲. عملیات بصری: نمودارها و مختصات

نکات

- تجرید میزان پرداختن به جزئیات است
- سطح تجرید نسبت به هر کلاس و مدل‌های مختلف* متفاوت است

[۷] Methodology

- خروجی هر فاز فرایند در متدولوژی مدل می‌شود. در حقیقت در متدولوژی مشخص می‌شود که مدل بخش مورد نظر به چه شکلی باشد.
- از آنجایی که زبان بین انسان و ماشین زبان برنامه نویسی (کامپایلر و گرامر) می‌باشد، زبان بین افراد برای نمایش بصیری نتیجه فرایندها مدل می‌باشد.
- عملیات ریاضی صرفاً محاسباتی نیستند، بلکه می‌توانند در قسمت آنالیز هم بررسی و انجام شوند

۷ الگو

الگو، راهنمایی برای حل مسائل مشابه می‌باشد. مشابه بودن مسائل به دلیل پر تکرار بودن آنها در پروژه‌های مختلف است.

۸ استاندارد

مجموعه‌ای از قواعد [۸] یا دستورات است. اجرای دستور ما را به خواسته می‌رساند. مانند تمام Rule هایی که روی فایروال شبکه اعمال می‌شوند. یا اینکه یکسری قواعد محیطی را بیان می‌کند.

۹ مهندسی نیازمندی

مهندسی نیازمندی یعنی مدلی که همه روی آن توافق دارند. یکسری حساب و کتاب، استاندارد .مدل‌ها و غیره که خوش تعریف هستند بدون هیچ‌گونه ابهام، مطرح می‌شوند.

۱.۹ دلیل استفاده از زبان UML

در مهندسی نیازمندی زبان مشترک بین تیم توسعه و طراحی با مشتری (کسی که درخواست دارد) زبان UML است. زبان درخواست کننده محاوره‌ای است و می‌تواند از آن هر برداشتی داشت.

۱۰ بررسی شروع کار مهندسی نیازمندی

۱.۱۰ بررسی UML to goal

قبل از انجام هر کاری بایستی اقدامات مهمی در شروع مهندسی صورت گیرد. تهیه نمودارهایی که با یکدیگر ارتباط مهمی دارند و لازمه ورود به بخش طراحی معماری نرم‌افزار است.

۱.۱.۱۰ نمودار هدف

اولین نموداری که در مهندسی باید کشیده شود نمودار هدف [۹] است. اهداف در نهایت به نیازمندی‌هایی می‌رسد که قرار است در سیستم محقق شود. بیان نیازمندی یعنی بیان اهداف.

[۸] Rules
[۹] Goal diagram

۲.۱.۱۰ نمودار ریسک

رسیک‌ها اتفاقات محیطی هستند که باید اقداماتی نسبت به آن‌ها در سیستم پیاده شود. مانند برقرار امنیت یا مشکلات کند بودن سرویس‌دهی مربوط به لود بالانسینگ. آن مواردی که به عنوان ریسک در اهداف پیدا می‌شود هم نیازمند کشیدن نمودار ریسک است.

۳.۱.۱۰ نمودار Agent

برخی از اقدامات توسط نرم‌افزار انجام می‌شود و برخی دیگر توسط کاربر (عامل). برخی از اهداف ممکن است به یکسری قابلیت‌های محیطی مربوط شوند. یعنی نرم‌افزار هیچ قوه تحلیلی برای مشتری ندارد بلکه مشتری است که با دخالت خود می‌تواند به هدف مورد نظر برسد. عامل کسی است که تعیین میکند قرار است چه عملیاتی رخ دهد.

۲.۱۰ مهندسی نرم‌افزار و مهندسی نیازمندی

در مهندسی نرم‌افزار مجموعه‌ای از ترتیب‌های [۱۰] مخصوص به آن وجود دارد مانند:

۱. مدیر پروژه Project manager
۲. مالک پروژه Product owner
۳. بخش‌های زیرساختی مانند زیرساخت شبکه و پشتیبانی و سرویس
۴. بخش پیاده‌سازی Implementation
۵. بخش بررسی استانداردها و متدولوژی‌ها
۶. بخش مستندات Documentation
۷. بخش آزمون Test

مهندسی نیازمندی یکی از زیر بخش‌های مهم مهندسی نرم‌افزار است.

۳.۱۰ مهندسی نیازمندی و مدیریت نیازمندی

مهندسی کلمه‌ای است که داشتن یک فرایند مرحله به مرحله را الزام‌آور می‌کند. یعنی برای مهندسی یک پروژه نرم‌افزاری باید تمام جنبه‌های نرم‌افزاری به همراه ابزارها را بشناسیم که با صحیح و خطا و آزمایش موجب تولید یک محصول نهایی نشویم.
برای مثال فرایند مهندسی نیازمندی چهار مرحله‌ای زیر:

۱. جمع‌آوری نیازمندی‌ها
۲. تمیز کردن داده‌ها و معنادار کردن آنها
۳. بیان زبان برای مطرح کردن داده‌ها
۴. صحت‌سنجی و اعتبارسنجی کارها

مدیریت یعنی توزیع منابع. این منابع می‌تواند زمان، نیروی انسانی و ارزش‌های مالی مانند پول و غیره باشد. مدیریت نیازمندی شامل مجموعه‌ای از ترتیب‌ها و توضیحات است که بیشتر به مدیریت پروژه مربوط می‌شود. مدیر پروژه سهم بین هر بخش از توسعه را تقسیم می‌کند. وظیفه مدیر نیازمندی، تقسیم وظایف به زیر عوامل است، اینکه بتواند منابع اصلی را بین افراد و زیر بخش‌های خود (مفهوم چتری) تقسیم کند.
فعالیت اصلی زیر بخش مدیریت نیازمندی، مهندسی نیازمندی‌ها می‌باشد.

[۱۰] Discipline

۱۱ فصل اول

۱.۱۱ اصطلاحات

۱.۱.۱۱ World problem یا Environment

دنیای مسئله جایی است که مشکلی در آن رخ داده است و کسی وجود دارد که این مشکل را در ابتدا بررسی و بعد از آن حل می‌کند. در حقیقت دنیا، محیط عملیاتی ما در مهندسی نیازمندی است. این دنیا می‌تواند سینما باشد یا دانشگاه. جنس این مسائل می‌تواند مشکل باشد که بایستی برطرف شود یا قابلیتی که می‌خواهیم در آینده اتفاق بیوفتد.

۲.۱.۱۱ Machine

ماشین راه‌حلی برای حل مسئله‌ای می‌باشد که پیش آمده است. ماشین می‌تواند به صورت آماده خریداری شود یا توسط تیم توسعه از صفر توسعه داده شود. ما باید در سند نیازمندی این نوع از نیازمندی را مشخص کنیم. ماشین در حقیقت نرم‌افزاری است که قرار است داشته باشیم [۱۱]. مدیر نیازمندی با توجه به هزینه می‌تواند برای مهندس نیازمندی تعیین کند که آیا داشتن نرم‌افزار آماده هزینه کمتری برایش دارد یا توسعه آن نرم‌افزار از صفر توسط تیم توسعه خود.

۳.۱.۱۱ Context

کلمه Context به معنای زمینه می‌باشد. تمام رفتارها و شکل‌های انجام کار را نشان می‌دهد. مشخص می‌کند که چه نیازمندی‌های علمی را باید بدانیم تا بتوانیم در نرم‌افزار آن را پیاده‌سازی کنیم. زمینه‌های مرتبطی برای توسعه که باید به علوم آنها واقف شویم. برای مثال هنگام توسعه یک نرم‌افزار تشخیص پیوند مولکولی و طراحی پروتئین نیازمند آن هستیم که در مورد شاخه‌های علمی بایولوژی، بایوتک و ژنتیک علومی را کسب کنیم. این علوم می‌تواند توسط تحقیقات و پژوهش‌های فردی بدست آید یا اینکه در راستای تحصیل در یک رشته می‌توانیم در رشته دیگر به تحصیلات آکادمیک بپردازیم و به نوعی مدرک کارشناسی آن حوزه را بدست آوریم که بتوانیم به صورت کامل روی موضوع عملیاتی خود واقف و مسلط شویم.

۴.۱.۱۱ Statement یا جمله

Statement یک جملست که ترکیبی از پدیده‌ها می‌باشد. برای مثال گفته می‌شود، وقتی ترمز خودرو فشرده شد، درها قفل شود و کاربر بتواند وضعیت دنده خود را تغییر دهد. بعضی از این پدیده‌ها در دنیای مسئله یا محیط اتفاق می‌افتد. فعل‌های محیطی را به هم متصل می‌کند و به فعل‌های نرم‌افزاری دخالتی ندارد.

۵.۱.۱۱ Phenomena یا پدیده‌ها

تمام اتفاقاتی که در مسئله (یا جمله) رخ می‌دهد را پدیده یا Phenomena گویند. برخی پدیده‌ها دقیقاً داخل نرم‌افزار رخ می‌دهد، مانند خطای TLS یا خطای پیدا نشدن صفحه. برخی پدیده‌ها بین ارتباطات رخ می‌دهد مانند نرمال‌سازی دیتابیس. پدیده خرید کردن یک پدیده محیطی است. وقتی برای کاربر اعلانی ارسال می‌شود در واقع این اعلانات پدیده بین محیط و نرم‌افزار است.

۶.۱.۱۱ System as is

سیستمی که در حال حاضر وجود دارد سیستم جاری یا System as is گویند. سیستم جاری بیشتر به محیط مربوط است. به عبارتی دیگر، المان‌ها و ارتباطاتی است که الان وجود دارد مانند افراد و دستگاه‌ها.

---

[۱۱] Software to be

۷.۱.۱۱ System to be

System to be دقیقاً سیستمی است که در آینده خواهیم داشت. تمام فرایند مهندسی که منجر به تولید سیستمی جدید می‌شود. چیزی که باید رخ دهد. مجموعه‌ای از المان‌های محیطی و Software to be.

۸.۱.۱۱ عوامل Prescriptive

عواملی که تجویزی هستند که نیاز سیستم را مشخص می‌کنند که چه کاری باید انجام شود:

۱. System requirement: یک System requirement مجموعه‌ای از Assumptionها و Software requirementهاست. تمام تک کارهای کوچکی که به محیط اختصاص می‌دهیم.

۲. Software requirement: تمام نیازمندی‌های نرم‌افزاری که می‌تواند به دو دسته Functional و Non-functional تقسیم شود. تمام تسک‌های کوچکی که به نرم‌افزار اختصاص می‌دهیم.

۳. Assumption: تمام عوامل محیطی که در پایین توضیح داده شده است.

مثال‌هایی از انواع System requirement:

- تمام درهای قطار بایستی در هنگام حرکت بسته باشند.
- مشتریان هیچ وقت نمی‌توانند بیشتر از سه کتاب را در یک زمان قرض بگیرند.
- تمام محدودیت‌های دعوت یک شرکت کننده به یک میتنینگ آنلاین بایستی به زودی برطرف شود.

۹.۱.۱۱ مفروضات یا Assumption

تمام عواملی که محیطی هستند و مستقیماً با نرم‌افزار ارتباطی ندارند. در واقعیت امر همان محیط و یا World problem هستند. ابزارهایی واسط بین انسان و انجام کار.

۱. People: مردم و کاربران

۲. Device: دستگاه‌ها مانند سنسورها، جمع‌آور داده و ارسال کننده به موتور تحلیل (نرم‌افزار)

۳. Exists softwares نرم‌افزارهای موجود: نرم‌افزارهایی که خودشان عملیات متعددی انجام می‌دهند و داده‌ها را برای تحلیل به نرم‌افزار اصلی سیستم ما ارسال می‌کنند.

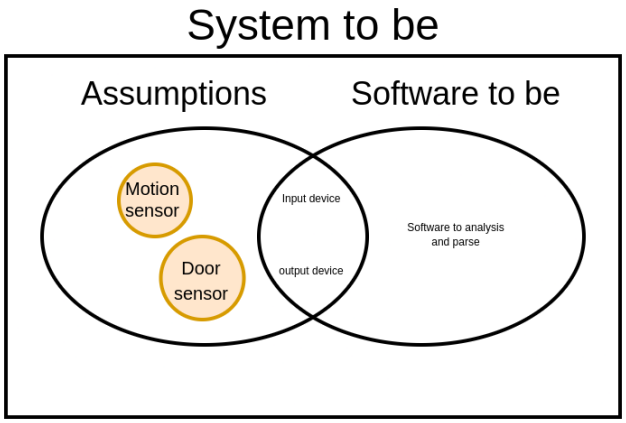
عوامل محیطی گسترده هستند. برای مثال وقتی که کاربر در اپلیکیشن سبد خرید خود را می‌خواهد حساب کند، زدن روی دکمه "پرداخت آنلاین" کاملا یک عامل محیطی است یعنی Assumption. زیرا با دخالت کاربر می‌توان سبد خرید را پرداخت کرد، در غیر این صورت نرم‌افزار خودش نمی‌تواند تصمیم بگیرد که پرداخت نهایی را کی باید انجام دهد (دیدگاه یک سیستم ساده).

۱۰.۱.۱۱ مثال

سناریو: درهای قطار موقع حرکت قفل شود. در این سناریو Statement، پدیده‌ها (Phenomena) و نیازمندی سیستم و پدیده‌های محیطی را مشخص کنید.

- جمله: درهای قطار موقع حرکت قفل شود.
- پدیده‌ها در این جمله دو نمونه هستند. حرکت کردن قطار و بسته شدن درها

- عوامل محیطی یا Assumptionها سنسور تشخیص حرکت قطار و محرک بازوی درهای قطار هستند که دائماً در حال مانیتور و کنترل در و حرکت قطار هستند.
- Assumptionها یعنی سنسورهای قطار و نرم‌افزاری که قوه تحلیل دارد یا Software requirement می‌شود نرم‌افزاری که قرار است در آینده داشته باشیم یا Software to be.
- کل این مجموعه را System to be گویند.



شکل ۱: مهندسی نیازمندی بیشتر به Assumption و قسمت اشتراکی شامل می‌شود.

۱۱.۱.۱۱ مفهوم Definition

یک معنای دقیق از چیزایی است که می‌نویسم به عبارت دیگر تمام اصطلاحاتی که در سیستم می‌تواند وجود داشته باشد را بیان می‌کند.

۱۲.۱.۱۱ مفهوم مانیتور کردن

مانیتور کردن یعنی بررسی داده‌های ورود و انجام تحلیل روی آنها.

۱۳.۱.۱۱ مفهوم کنترل کردن

کنترل کردن یعنی فرایند بعد از تحلیل، یعنی اعمال کردن نتایج بدست آمده.

۱۴.۱.۱۱ عوامل Descriptive

عوامل توصیفی، قوانین طبیعی و قید و شرط‌های فیزیک که غیر

۱۵.۱.۱۱ ویژگی دامنه یا Domain property

یک عبارت توصیفی است که یک حقیقت از فیزیک را بیان می‌کند. این عبارت قابل مذاکره نیست که برای مثال بگوییم بعداً می‌توان آن را تغییر داد. به هیچ وجه نمی‌توان آن را کم یا زیاد کرد.
برای مثال:

۱. برای مثال دانشجو نمی‌تواند دو درس مختلف در زمان یکسان اخذ کند. یعنی از نظر فیزیک نمی‌توان همزمان در دو کلاس در زمان یکسان حاضر شد. و این پیام را نیازمندی نرم‌افزار در حقیقت برنامه نویس مشخص می‌کند.

۲. هنگامی که درهای قطار بسته باشند، یعنی دیگر باز نیستند.

۳. اگر شتاب قطار مثبت باشد، بدان معانست که سرعت قطار =! صفر می‌باشد.

۱۶.۱.۱۱ دامنه‌ها

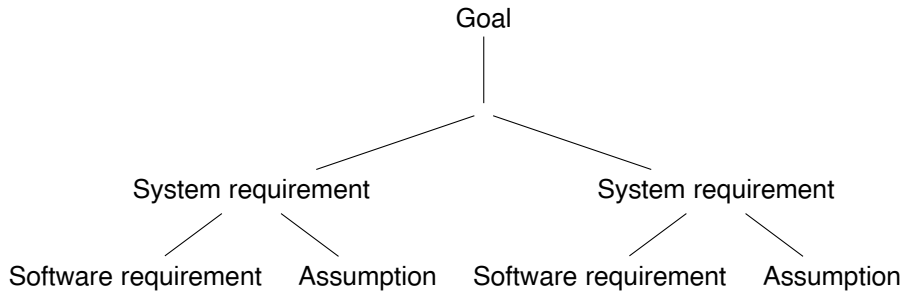
دامنه‌های در دل سازمان‌ها هستند، مانند دامنه پژوهشی، دامنه‌های مالی و ارتباط بین آدم‌ها در دامنه وجود دارد.

۱۷.۱.۱۱ اسکوپ‌ها

مجموعه‌هایی از System requirement هستند که نرم‌افزار می‌تواند در آنها ورود داشته باشد. مثلا فعالیت‌های مربوط به ثبت‌نام دانشجو، که اصطلاحاً به آنها System scope می‌گویند.

نکات

- مهندس نیازمندی باید در کنترل و مدیریت اسکوپ‌ها حساسیت داشته باشد که نرم‌افزار از دست خارج نشود و باعث پیچیده‌تر شدنش نگردد.
- دامنه‌ها درست است که ثابت و غیرقابل مذاکره هستند، اما از یک دامنه به دامنه دیگر می‌تواند ویژگی‌ها تغییر کنند در حالی که ساختار این دامنه حفظ شود. برای مثال زمانی که دامنه مورد نظر یک کتابخانه فیزیکی است، همزمان دو نفر نمی‌توانند یک کتاب مشترک را تقاضا کنند. اما در کتابخانه دیجیتال که به صورت اپلیکیشن می‌باشد، درست است که ساختار دامنه همانند موجودیت‌ها و شکل کتابخانه فیزیکی است اما نحوه استفاده آن کاملاً تغییر کرده و چندین کاربر می‌توانند همزمان یک کتاب را به صورت دیجیتال مطالعه کنند.
- در مهندسی نیازمندی تنها یک نمودار استفاده نمی‌شود. برای مثال زمانی که یک نمودار Sequence برای نمایش ارتباطات دستگاه ها کشیده می‌شود نیازمند آن است که نمودار هدف نیز داشته باشد. بعد از آن بایستی تمام ریسک‌های مربوط به آن نیز به صورت نمودار اعلام شود. چرا که باعث تولید یک سند مهندسی نیازمندی کامل می‌شود که در زمان‌های مختلف می‌توان به آن مراجعه کرد و متوجه تمام موضوعات بدون فراموشی تنها یک بخش شد.
- بُعد Why در نمودار معمولاً نشان‌دهنده اهداف است. مثلا پیاده‌سازی این قابلیت هدف‌اش رضایت مشتری است.
- همیشه از اهداف شروع می‌کنیم و به نیازمندی‌های سیستمی می‌رسیم و نیازمندی سیستمی را در نیازمندی‌های نرم‌افزاری و محیطی بررسی می‌کنیم.



۱۸.۱.۱۱ تفاوت‌های بین Descriptive و Prescriptive

- جملات تجویزی را می‌توان برای آنها مذاکره کرد، آنها را کم و زیاد کرد یا حتی برای آنها جایگزینی معرفی نمود.
- جملات توصیفی اصلاً قابل تغییر نیستند.

۲.۱۱ مولفه‌های مربوط به نیازمندی نرم‌افزار در نیازمندی سیستم

۱. مانیتورینگ: تمام مقادیر محیطی که نرم‌افزار توسط دستگاه‌های ورودی مانند سنسورها، داده‌های آن را دریافت می‌کند.

۲. کنترل: مقادیر محیطی که نرم‌افزار آنهارا می‌تواند از طریق دستگاه‌های خروجی (Actuators) آن‌ها را کنترل (اعمال) کند.

۳. مقادیر دستگاه‌های ورودی [۱۲]: تمام داده‌هایی که به عنوان ورودی در نرم‌افزار استفاده می‌شود.

۴. متغیرهای خروجی [۱۳]: مقادیری که نرم‌افزار آنها را در دستگاّه‌ای خروجی اعمال می‌کند.

نکته

بیشتر سازمان‌ها به دو دسته زیر فعالیت‌های خودشان را انجام می‌دهند:

۱. سازمان‌هایی که هدفگرا هستند و تنها برای رسیدن به محصول آخرین تلاش و فعالیت خود را می‌کنند.

۲. سازمان‌هایی که تعداد Agent و کاربرانشان زیاد است و ارزش‌های زیادی برای آنها قائل می‌شوند به صورت گراAgent یا عاملگرا هستند.

سطح System requirement بالا می‌باشد، چرا که مشتری تنها درخواست می‌کند که می‌خواهد چنین قابلیت‌هایی وجود داشته باشد، به ماهیت و نیازمندی و حتی پیچیدگی آنها کاری ندارد.

۳.۱۱ توافق بر لغات

۱. SOFTREQ: منظور Software requirement

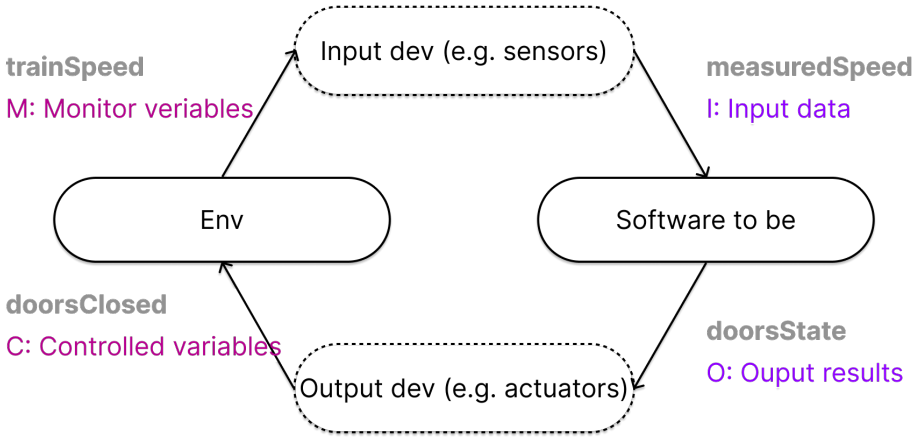
۲. ASM: منظور مفروضات یا Assumption

۳. DOM: منظور دامنه یا Domain

$$SOFTREQ + ASM + DOM \rightarrow SYSTEMREQ \quad (۳)$$

اگر نیازمندی نرم‌افزار، مفروضات و دامنه‌ها همگی مقید و راضی باشند نیازمندی سیستم نیز بدست می‌آید. با استفاده از پارامترهای بالا می‌توان به سیستم نهایی رسید.

[۱۲] Input
[۱۳] Output



شکل ۲: ارتباط نیازمندی سیستم در نرم‌افزار به همراه استدلال‌ها

- SOFTREQ: Input ‘ Ouput
- ASM1: Monitor ‘ Input
- ASM2: Ouput ‘ Control
- SYSREQ: Monitor ‘ Control

استدلال سناریو

$$SOFTREQ : measuredSpeed \neq 0 \rightarrow doorsState = "closed" \quad (۴)$$

$$ASM1 : measuredSpeed \neq 0 iff trainSpeed \neq 0 \quad (۵)$$

$$ASM2 : doorsState = "closed" iff doorsClosed \quad (۶)$$

$$DOM : trainMoving iff trainSpeed \neq 0 \quad (۷)$$

$$SYSREQ : trainMoving \rightarrow doorsClosed \quad (۸)$$

۴.۱۱ دسته‌بندی نیازمندی‌ها

۱.۴.۱۱ Functional requirement

تعیین می‌کند که چه سرویسی قرار است در Software to be ارائه شود. برای مثال:

- نرم‌افزار کنترل قطار باید بتواند سرعت تمام بخش‌های سیستم قطار را کنترل کند.
- سیستم آنلاین فهرست کتب باید براساس موضوع کتاب نام تمام کتابخانه را نمایش دهد.
- کاربران در سیستم پارکینگ آنلاین باید بتوانند رزرو لحظه و رزرو روزانه را به انتخاب خودشان استفاده کنند.
- دانشجویان زمانی که وارد کلاس آنلاین می‌شوند باید قابلیت به اشتراک گذاری صفحه نمایش خود را داشته باشند.

همچنین می‌توانند براساس شرایط محیطی باشند که تحت آن چه عملیاتی باید انجام شود:

- درهای قطار تنها در زمانی می‌توانند باز شوند که قطار به طور کامل ایستاده باشد.

دسته‌بندی توابع

۱. Information: اطلاع رسانی، اعلانات هر چیزی که قابلیت ارسال و دریافت را داشته باشد.

۲. Satisfaction: تعیین State یک کار است که در جریان معنا دارد.

۳. Stim-response: محرک پاسخ، وقتی دکمه در UI زده شد آلارم را صدا کند.

۲.۴.۱۱ Non-functional requirement

تعیین می‌کنند که چگونه یک سرویس می‌تواند ارائه شود. برای این دسته باید مجموعه‌ای از اقدامات که بار اجرایی دارند را استفاده کرد:

- معیارها و نیازمندی‌های کیفی:
  - معیارهای ایمنی
  - معیارهای امنیتی
  - سرعت و دقت
  - عملکرد زمانی و حافظه‌ای
  - قابلیت استفاده
- بقیه موارد
  - هنجارها
  - معماری
  - نیازمندی‌های توسعه

برای مثال:

- دانشجویان هنگام به اشتراک گذاری صفحه خود کیفیت صوت را به خوبی قبل از اشتراک گذاری داشته باشند.
- قطار هنگام حرکت امکان باز کردن در را نداشته باشد.
- دستورات شتاب قطار هر ۳ ثانیه یکبار می‌تواند ارسال شود.

۵.۱۱ کیفیت سرویس‌دهی یا QoS (محصول)

پارامتری را نشان می‌دهد که می‌خواهیم آن را از نظر کیفی تامین کنیم. برای مثال برقراری اهداف امنیتی.

۶.۱۱ Service Level Agreement

یک توافق بین معمار نرم‌افزار و کارفرما برای تعیین سطح سرویس از نظر کیفی می‌باشد. در قراردادهای SLA مقدار قابل قبولی از QoSهایی که دنبالش هستیم را بیان می‌کنیم.

۷.۱۱ تفاوت بین Constraint و Limitation

Constraint به معنای قید و شرط است، مقید شدن به چیزی. برای مثال نرم‌افزاری توسعه داده شود که قابلیت نصب روی دستگاه‌های موبایل را داشته باشد.

Limitation به معنای محدودیت است که بار منفی دارد. در این حالت نرم‌افزار باید با آن کنار بیاید.

۸.۱۱ مفهوم هنجارها یا Compliance (محصول)

منظور از Compliance قواعد و هنجارهایی است که الزاما ثابت نیستند. نرم‌افزار باید تابع این هنجارها باشد. قواعدی که در نرم‌افزار قید می‌شود برای مثال فاصله بین دو ماشین در سال ۲۰۲۰ با تصمیم‌گیری شهرداری برای ماشین‌های خودران ۴ متر توافق شد. اما بعد از پیشرفت تکنولوزی و علوم مربوطه این فاصله به یک متر کاهش یافت.

۹.۱۱ قیدهای معماری Architectural constraint (محصول)

بعضی از قیدهای معماری مربوط به نصب و راه‌اندازی هستند و برخی دیگر مربوط به توزیع می‌باشند.

۱. نصب

(آ) نرم‌افزار باید روی پلتفرم موبایل یا عینک گوگل قابل نصب باشد

(ب) مشخصات لازم برای نصب موفقیت‌آمیز نرم‌افزار و بازی

(ج) این نیاز می‌تواند پایین‌تر از سطح سکو نیز باشد، مثلاً نصب تنها در یک سیستم عامل مخصوص

(د) قابلیت نصب تنها در سخت‌افزارهای X۸۶

۲. قید توزیع: ورودی و خروجی از دو درب مختلف در دانشگاه، به دلیل آنکه داده‌های محیطی ورودی و خروجی در دو محل متفاوت است برای رسیدن به توافق در این توزیع باید این داده‌ها را در یک جا با هم سینک کنیم تا اطلاعات ورودی و خروجی مناسب یکدگیر پدید آید.

۱۰.۱۱ قیدهای توسعه Development constraint (مدیر پروژه)

یکی از مهم‌ترین عوامل نگرانی مدیر پروژه است، کاری به ماهیت محصول ندارد بلکه برای او مهم‌ترین عوامل انتخاب مناسب متدولوژی و تصمیم درست می‌باشد. تعیین هزینه زمانی و مالی نیز از دیگر نگرانی‌های مدیر پروژه می‌باشد تا در نهایت طراح معماری بتواند با دید کامل و بدون تحت فشار قرار گرفتن، معماری مناسب را طراحی کند.

جلسه چهارم:

کیفیت جمله‌ها میتواند صرفا کیفیت درستی نداشته .

مجموعه ای از قابلیت‌ها میشه اسکوپ که در دامین‌ها تعریف می‌شوند.

برای ساخت سبد باید فرایندهای مهندسی نیازمندی را برویم:

چهار قدم مهندسی نیازمندی:
تمام مراحل تکرار پذیر هستند.
ساعت گرد. داده‌ها جمع‌آوری میشود ولی همه داده‌ها انتخاب نمی‌شود. یکی از آنها انتخاب می‌شود.
دامنه و استخراج نیازمندی‌ها
ارزیابی و توافق
ممکنه چیزایی جمع‌اوری کرده باشیم که کاملا نامربوط به اسکوپ باشد. هر آنچه به ما می‌گویند ممکنه ریسک‌هایی باشد کگه می‌تواند قابلیت بخش به نرم‌افزار باشد.
مثلا ریسک این وجود دارد که ممکنه اینترنت قطع بشود.
نیازمندی‌هایی که به ما وارد میکند الزاما همراستا نیست
میخواهد کارنامه ببیند دانشجو، استاد میگه ثبت نمره میخوام کارنامه رو ببینم. اگه یکی رو برآورده کنیم ممکنه کانفلیکت به وجد آید.
اولویت نیازمندی ها را آیا تعین کرده‌ایم؟ لیست دروس رو نشون ندیم ولی دانشجو بتواند درس انتخاب کند. (امکان ندارد.)
همه این تصمیمات رو باید ارزیابی کنیم.
دسته آخر: یک سبدی که خیلی آشفته بود تبدیل به سبدی میشود که همه روی آن توافق دارند و به طراح داده‌میشود.
نیازمندی‌هایی که روی آنها توافق شده است.
سبد بعد از آن به طراح تحویل داده می‌شود که زبان مشترک بین طراح و نیازمندی میشه اشکال بصری که مطالب صریح و سریع انتقال شود.
سند یک قالب می‌خواد که استاندارد است. با چی بنویسیم با نماد گرافیکال نمایش بده.
بخش آخر:
ادغام، اعتبارسنجی: سبد دستخوش تغییرات است. تا برسد به ۸۰ درصد ثابت و ۲۰ درصد تغییر کننده. این تغییر ۲۰ درصدی می‌تواند ساید افکت ایجاد کند. پس برای رفع ساید افکت باید اعتبارسنجی حتما انجام شود.
می‌تواند چندین دور بزد.
اسلاید ۵۲:
آیا هر سیستمی نیازمند مهندسی نیازمندی است؟ خیر.
سیستم‌های لگیسی حتما سند نیازمندی می‌خواهند.
کلا اونایی که ورکفلوهای اصلی رو می‌چرخونن سند نیازمندی می‌خواد
پروژه‌های استارتآپی که قراره خدمات به مردم بده که جنس خدمت یکیه و نحوه انجام آن متفاوت است. این سیستم‌ها هم سند نیازمندی براشون اهمیت داره.
سند نیازمندی قابلیت reuse را به پروژه‌های مشابه می‌دهد.
یک مبنعی برای پروژه‌های مشابه میشه نه یه الگو.
اسلاید ۵۲
سند نیازمندی: اول خواسته در آورده می‌شه بعد قرارداد پروژه در میاد.
سازمان‌ها براساس proposal for Request کار میکنند. که مهندس نیازمندی و متخصصین اوجنا می‌نویسن که معمولا واحدهای it مسئول آنها هستند.
پروتوتایپ در خصوص بخری نیازمندی‌ها که مبهم است که واضح نیست یک پروتوتایپ درست می‌کنیم که بفهمیم اون نیاز چیست. می‌تواند روی کاغذ باشد باشد یا یه ui اشد. که منظور اولیه رو برساند. می‌تواند تو سطح فاکنشن باشد هم می‌تواند نان فانکشن باشد.
چرا این سند دو طرفه‌است. توی محیط یه چیزی به ما گفتن. یعنی یه نیازی به ما منقل شده که ما پروتوتایپ درست کردیم. بعد اون پروتوتایپ رو بهش نشون میدیم که ببینیم درست فهمیدیم یا نه. ممکن نیازمندی اضافه بشه یا حذف بشه تا سبد اسکوپ ما کامل شود.
تخمین پروژه: باید سند نیازمندی‌ها دیده شود که بفهیمیم چقدر خواسته داریم تا زمان مشخص کنیم، تا هزینه و اندازه تخمین بزنیم. ورژن بندی اینجا انجام می‌شود. یکی از نیازمندی‌ها غیرعملیاتی مربوط به توسعه بوده. روی سبد تاثیر گذار است. که بایستی براساس زمان معقول باشد.

test acceptance باید با نیازمندی‌های مشتری مطابقت داشته باشد. باید سناریو تست وجود داشته باشد. سناریوهای تست از سند نیازمندی می‌آید.

معماری:

چه قیدهایی، چه ترتیبی تا بتواند نان فانکشن‌ها رو چک کنه که نان فانکشن‌ها در سند نیازمندی نوشته شده است. مانند ،availability ،useability و دیگر خواسته‌ها.

معماری یکسری الزامات را ایجاد می‌کند. اون الزامات از جنس فانکشن‌هاستند. که روی سند نیازمندی‌ها تاثیر می‌ذاره.

ریپورت صفحه ۵۳ نمودار رو نگاه اینجا بنویس.

expectation منظور همان .هاستAssumption فصل دوم، معادل فاز یک استخراج داده.

بستنی فروشی میشه سازمان

هر سطح پایینی میشه دامنه (ظرف‌ها) و هر هر دامنه شامل اسکوپ

مجموعه مشکلات در اسکوپ‌ها. is as system

دو دسته برای جمع‌آوری داده داریم:

تکنیک‌های فراورده گرا یا :artifact

هر آنچیزی که در پروژه تولید یا استفاده می‌شود. از آرتفیکت‌هایی استفاده میکنمی که نیاز بکشیم بیرون. قواعد آموزشی میشه ازش نیاز کشید بیرون. پروتوتایپ‌ها هم از آن نیاز کشیده میشود بیرون.

آرتیفکت منبع نیازمندی است.

دسته دوم ذینفع گراkl

آدم‌ها منبع نیازمندی‌ها هستند. مثل جلسات.

مستندات موجود را مطاله می‌کنیم. سندهای موجود در سازمان را ارتیفکت می‌گویند.

مشکلات آرتیفکت:

background

حجم مستندات زیاد است. جزییات نامرتبط مثلا بخش بایگانی اسنادی را نگهداری میکند که ممکن است نامرتبنط باشد. اسناد ممکنه outdate باشد. یعنی یه جورایی با ذینفعان هم درگیر میشن.

هرس کردن مستندات:

بررسی بخش‌هایی که معتبر است و حذف بخش‌هایی که نامعتبر است. مثل خواندن درس در محدوده فصل‌های مشخص شده تا تمام فصل‌ها.

روش دوم:

دیتا کالکشن: آرتیفکت اینجا دیتا هستش. این دیتا می‌تواند متادیتا باشد مانند فرم ثبت نام. جمله ندارد بلکه براساس داده‌ها به جمله می‌رسند. مثلا در جمع آوری داده در مورد مستندات نرم‌افزار مثلا ۵ تا نرم‌افزار نوشته مثلا تو طراحی ui جست و جوشون براساس درختی بوده. یعنی سوابق طراحیشون رو دیده. براساس داده‌هایی که جمع کردیم به استیت منت نان_فانکشن می‌نویسیم. نوشتن جمله و تفسیرش توسط مهندس نیازمندی براساس داده‌های جمع‌اوری شده است.

یکی از مشکلاتش اینه که ممکنه درست نباشه این تفسیر مهندس نیازمندی. یعنی لزوماً حرف آخر را ممکن است با تفسیر نا درست بیان کند.

یه زمانی دیتا جمع‌کردن خودش زمان بره و کار مشکلیه.

یک نکته مثبت: در روش قبلی که سند قبلی میخوندیم سندهای علمیاتی بودن ولی در این روش (دیتا کالکشن) غیرعملیاتی است.

اینا تماماً روش‌های پایه است.

mining Text – elicitation: requirement

پرسشنامه آرتیفکت:

هم می‌سازیمش هم ازش استفاده و بهره برداری می‌شود.

انتها باز

انتها بسته مانند انتخاب‌های چک مارکی یا رادیو باکس.

بعضی از سوالات کیفی است یعضی کمی. کیفی: (خوب، خیلی خوب) کمی: درصدی

دلیل: تنوع زیاد کاربران و عدم تمرکز لوکیشن‌ها فرهنگ فناوری در کاربران متفاوته. پس برای پوشش دایورسیتی از پرسش نامه استفاده میکنیم.

سریع، ارزان و می‌تواند از راه دور نیازمندی بسیاری از افراد را جمع‌آوری کنیم.

پرسشنامه باید روایی و کارایی داشته باشد.

روایی یعنی چی در پرسشنامه چیست؟

پرسشنامه اینترویو: از آدمی که میداند بپرس. سوالاتی که جواب نداریم را در اینترویو می‌پرسیم. مثلا بعد از ثبت نام نوتیف بدیم؟ مثلا میگن نه فقط در تغییر کلاس نوتیف بده.

تا آخر کوئسشنری. صفحه ۱۰.

موضوعات گروه‌های سه نفری و یک مقاله چدید از Springer. engineering requirement یا حتی کنفرانسش.