

مهندسی نیازمندی‌ها خانم دکتر سپیده آدابی

علیرضا سلطانی نشان

۱۰ اسفند ۱۴۰۲

فهرست مطالب

۲	۱ مجوز
۲	۲ مهندسی نیازمندی
۲	۱.۲ تعریف
۳	۳ نکته تجرید
۳	۴ متدولوژی
۳	۵ دلیل متدولوژی‌های مختلف
۳	۶ ماهیت مدل
۴	۷ الگو
۴	۸ استاندارد
۴	۹ مهندسی نیازمندی
۵	۱.۹ دلیل استفاده از زبان UML
۵	۱۰ بررسی شروع کار مهندسی نیازمندی
۵	۱.۱۰ بررسی UML to goal
۵	۱.۱.۱۰ نمودار هدف
۵	۲.۱.۱۰ نمودار ریسک
۵	۳.۱.۱۰ نمودار Agent
۵	۲.۱۰ مهندسی نرم‌افزار و مهندسی نیازمندی
۶	۳.۱۰ مهندسی نیازمندی و مدیریت نیازمندی
۶	۱۱ فصل اول
۶	۱.۱۱ اصطلاحات
۶	۱.۱.۱۱ World problem یا Environment
۶	۲.۱.۱۱ Machine
۶	۳.۱.۱۱ Context

۷	Statement ۴.۱.۱۱ یا جمله
۷	Phenomena ۵.۱.۱۱ یا پدیده‌ها
۷	System as is ۶.۱.۱۱
۷	System to be ۷.۱.۱۱
۷	Assumption یا مفروضات ۸.۱.۱۱
۷	Domain property یا ویژگی دامنه ۹.۱.۱۱

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 کاملاً پر در درسر هستند زیرا خیلی از دلایل شکست پروژه‌ها عدم بررسی نیازمندی‌ها 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 چگونه^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.

زمانی که می‌گوییم نرم‌افزار ثبت نام درس، دقیقاً بالاترین سطح تجرید^۴ 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

^۱ Stable

^۲ What

^۳ How

^۴ Abstract

- همیشه باید بین مسائلی که در مهندسی نرم افزار پیش می آید یک سبک سنگینی^۵ صورت گیرد. معمولاً Benchmarks ها به ما این امکان را می دهند. یعنی نرم افزار می تواند به چند شکل مختلف توسعه پیدا کند اما با گرفتن Benchmark ها می توانیم بررسی کنیم که کدام یک از آنها در قسمت عملیاتی و عملکرد صحیح بهتر بوده اند. به عبارت دیگر، روش ها را نمی توان بدون بررسی و با میل شخصی انتخاب کرد، بلکه باید روش ها بررسی و سبک سنگین شوند.

- فرایندها در مهندسی نیازمندی را process گویند

- توضیح و بازنویسی نیازمندی ها، کار پایه مهندس نیازمندی است.

- تمام مراحل در فرایند به یکدیگر وابسته می باشند، فرایند اساساً در مورد جزئیات صحبت نمی کند بلکه به ماهیت کلی و تجرید می پردازد. برای مثال فرایند جمع آوری داده و تحلیل و دیگر مراحل کاملاً به صورت مرحله ای و بازگشت پذیر می باشد. خروجی فرایند بعد از طی کردن تمام مراحل، نیازمندی را مشخص می کند.

- هیچ وقت فرایند با نیازمندی ها هم ارز نیست، بلکه نیازمندی خروجی فرایند می باشد. در حقیقت به خروجی فرایند، سند نیازمندی یا Requirement Document (RD) می گویند.

- در فرایند تکنیک ها و استانداردها دیده می شود.

۳ نکته تجرید

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

۴ متدولوژی

متدولوژی^۶ یک جهان بینی کلی، در تولید نرم افزار است (دید از بالا برای انجام کارها و وظایف). تمام متدولوژی ها را برای تولید استفاده می کنند و تمام راهنمایی ها توضیحات دارند. در حقیقت تمام متدولوژی ها از خواستگاه تولید نرم افزار ایجاد شده اند و حتی می شوند. نکته مهم آن است که فرایندها درون متدولوژی ها هستند. متدولوژی یک نقشه است که آن را معمار نرم افزار با دیدگاه کاملاً جامع انتخاب می کند.

۵ دلیل متدولوژی های مختلف

ماهیت و ذات پروژه ها متفاوت و پیچیده است، پس در این جهت متدولوژی های مختلفی برای مهار آنها ارائه شده است که نوع تولید را متفاوت می کند. متدولوژی بایستی کاملاً منعطف باشد. مراحل و فرایندها در متدولوژی ها متغیر می باشد.

۶ ماهیت مدل

انسان همیشه با خواندن مشکل دارد. خواندن دائماً با مشکلات محاوره ای همراه است. محاوره با ابهام همراه است. در پروژه مهندسی نرم افزار، وقتی افراد بخواهند با یکدیگر در مورد پروژه صحبت کنند، زبان میان آنها مدل های بصری و گرافیکی می باشد. افراد بعد از جمع آوری 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 جمله های ترکیبی مشکل دارد:

Trade off^۵
Methodology^۶

$$(A \& B)v(c) \rightarrow x \quad (1)$$

یا

$$A \& (BvC) \rightarrow x \quad (2)$$

راهکار: استفاده از Decision table که بتوان منطقی به نتیجه رسید.
زبان مدل سازی: ریاضی و گرافیک (بصری)

عملیات به دو دسته تقسیم می شوند

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

۷ الگو

الگو، راهنمایی برای حل مسائل مشابه می باشد. مشابه بودن مسائل به دلیل تکرار بودن آنها در پروژه های مختلف است.

۸ استاندارد

مجموعه ای از قواعد^۷ یا دستورات است. اجرای دستور ما را به خواسته می رساند. مانند تمام Rule هایی که روی فایروال شبکه اعمال می شوند. یا اینکه یکسری قواعد محیطی را بیان می کند.

۹ مهندسی نیازمندی

مهندسی نیازمندی یعنی مدلی که همه روی آن توافق دارند. یکسری حساب و کتاب، استاندارد، مدل ها و غیره که خوش تعریف هستند بدون هیچ گونه ابهام، مطرح می شوند.

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^۸ است. اهداف در نهایت به نیازمندی‌هایی می‌رسد که قرار است در سیستم محقق شود. بیان نیازمندی یعنی بیان اهداف.

۲.۱.۱۰ نمودار ریسک

ریسک‌ها 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^۹ 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.

^۸Goal diagram

^۹Discipline

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.

۱۱ فصل اول

۱.۱۱ اصطلاحات

۱.۱.۱۱ Environment یا World problem

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^{۱۰}. 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، بیوتک و زنتیک علوم 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.

^{۱۰} Software to be

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 نرم‌آل‌سازی 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

۷.۱.۱۱ System to be

۸.۱.۱۱ مفروضات یا Assumption

۹.۱.۱۱ ویژگی دامنه یا Domain property

is as system به محیط مربوط است.

به عکس گرفته شده مراجعه شود. مهندسی نیازمندی بیشتر به قرمز و حاشور حساسیت انجام می‌دهد.

Prescriptive

Descriptive

Phenomena

نرم‌افزار به اجزای زیر نیاز دارد:

People: عوامل Device: دستگاه‌های واسط برای دریافت داده و انتقال به بخش software: exists software تمام توابع اجرایی مثل

پروتکل‌ها و غیره

قرار است گزارش‌هایی را تهیه کنند که به آنها سیستم تحویل داده شود.

world Env/Problem

تولزها واسط بین انسان و انجام کار هستند.

دیوایس‌ها مثل سنسورها

Prescriptive:

تجوزی است که نیاز سیستم مشخص می‌کند که چیکار باید کرد.

– requirement: system وظایف اعلان‌های موجود در سیستم را مشخص می‌کند. درهای قطار موقع حرکت قفل شود. استیت منت

میشه و فعل‌ها بسته شدن و حرکت کردن.

سنسورهای قطار و اون نرم‌افزار باهم میشه سافتور تو بی، و در کنار هم میشه سیستم تو بی.

یک requirement سیستم مجموعه از assumption ها و requirement sw هاست.

be to system مجموعه از المان‌های محیطی و be to sw

روی دکمه تایید زدن assumption هستش که کاربر باید دخالت کنه.

– requirement: SW Assumption: مفروضات. تک کارهای کوچکی که به محیط میدیم میشه مفروضات

System:

– is: as system چیزی که هست. المان‌ها و ارتباطاتی که الان وجود دارد. مانند افراد و دستگاه‌ها

– be: to system چیزی که باید رخ بدهد.

Descriptive:

property: Domain – یک عبارت توصیفی است. یک حقیقت از فیزیک است. قابل مذاکره نیست. همیشه بگیریم بعدا. اصلا کم و زیادش همیشه کرد.

برای مثال نمی‌توان دو تا درس در دو زمان یکسان برداشت. یعنی از نظر فیزیکی نمی‌توان همزمان در دو کلاس حاضر شد. و این پیام را نیازمندی نرم‌افزار برنامه نویس مشخص کرده است که این پیام را نشان بدهد.

Requirement: SW.

requirement functional non – requirement Functional –

نکته:

یک assumption یک پدیدست که statement را مشخص می‌کند.