



به نام دانای نیا موخته

درس: مهندسی نرم افزار

ترم: نیم سال اول [۰۴-۰۵]

جلسه ۱

دپارتمان: مهندسی کامپیوتر

مدرس: دکتر شیما شفیعی

بارم بندی

۱. پایان ترم : ۶ نمره
۲. میان ترم : ۶ نمره
۳. حل تمرین: ۶ نمره
۴. پروژه گروهی: ۲ نمره

فعالیت مازاد:

مرور

فعالیت کلاسی

حضور در تمام جلسات

توسعه پروژه

نماینده کلاس

حل تمرین کلاس

نرم افزار و مهندسی نرم افزار

- نرم افزار مجموعه ای از برنامه های کامپیوتری، مستندات و داده های پیکربندی است که برای انجام وظایف خاص در یک سیستم رایانه ای طراحی شده اند. نرم افزار نه تنها شامل کدهای اجرایی است بلکه شامل مستنداتی نیز می شود که نحوه استفاده، نگهداری و توسعه نرم افزار را توضیح می دهند. همچنین داده های پیکربندی که برای عملکرد صحیح نرم افزار ضروری هستند نیز بخشی از نرم افزار محسوب می شوند.
- نرم افزار، رابط بین کاربر و سخت افزار است که به رایانه فرمان می دهد که چگونه کار کند. این مجموعه باعث می شود تا کامپیوتر بتواند وظایف متنوعی را به صورت بهینه و مفید انجام دهد.
- مهندسی نرم افزار به معنی استفاده از اصول و روش های مهندسی به صورت منظم و سیستماتیک برای طراحی، توسعه، تست و نگهداری نرم افزارهایی با کیفیت بالا، قابل اعتماد و کارآمد است. این رشته علمی شامل فرایندهای تحلیل نیازمندی ها، طراحی، پیاده سازی، تست و نگهداری نرم افزار است تا محصول نهایی بتواند به طور موثر نیازهای کاربران را پاسخ دهد و با صرفه جویی در هزینه ها و زمان، به تولید برسد.
- به بیان دیگر، مهندسی نرم افزار کاربرد روش ها، تکنیک ها و ابزارهای مهندسی برای ساخت نرم افزارهای قابل اطمینان، با کیفیت و مقیاس پذیر است که روی ماشین های واقعی به طور کارآمد اجرا می شوند.

- مهندسی نرم افزار شامل فرایندهای دقیق و سیستماتیک تحلیل نیازمندی‌ها، طراحی، پیاده‌سازی، تست، ارتقاء و نگهداری نرم افزارها است به گونه‌ای که محصول نهایی بتواند با رعایت استانداردها و کیفیت مورد نظر، نیازهای کاربران را برآورده سازد.
- این تعریف از مهندسی نرم افزار توسط IEEE تأکید می‌کند که توسعه نرم افزار باید بر پایه رویکردهای مهندسی و علمی انجام شود تا اطمینان حاصل شود نرم افزار در محیط عملی قابل استفاده و حمایت است.
- از آنجایی که، نرم افزار به مجموعه‌ای گسترده از کدها، مستندات و داده‌های جانبی گفته می‌شود که نقش کلیدی در عملکرد و نگهداری سیستم‌های کامپیوتری دارند. لذا برنامه‌های اصلی، نرم افزار عبارتند از:

الف) مستندات فنی و کاربری نرم افزار

ب) داده‌های مورد نیاز برای پیکربندی و شخصی سازی

پ) ابزارها و کتابخانه‌های مورد استفاده برنامه

ت) سیستم‌های مدیریت پایگاه داده و ذخیره سازی

تفاوت مهندسی نرم افزار و علم کامپیوتر و مهندسی سیستم

جدول ۱: قیاس مهندسی و علوم نرم افزار و کامپیوتر و سیستم

مهندسی سیستم	علم کامپیوتر	مهندسی نرم افزار	شاخص
طراحی و مدیریت کل سیستم‌ها	مبانی نظری و الگوریتم‌ها	توسعه و نگهداری نرم افزارهای کاربردی	تمرکز اصلی
سیستماتیک، بین رشته‌ای	نظری، تحقیقاتی و علمی	عملی، مهندسی و سازمان‌یافته	نوع فعالیت
هماهنگی و بهینه‌سازی سیستم‌ها	فهم اصول و ساخت الگوریتم‌ها و داده‌ها	تولید نرم افزار قابل اجرا و نگهداری	هدف
طراحی سیستم حمل و نقل، مدیریت پروژه‌های بزرگ	پژوهش در الگوریتم‌ها، نظریه محاسبات	تحلیل نیازمندی‌ها، طراحی نرم افزار، تست	مثال
مهندسی صنایع، مدیریت پروژه، مهندسی برق	هوش مصنوعی، نظریه زبان‌ها، سیستم‌های محاسباتی	مدیریت پروژه‌های نرم افزاری، تضمین کیفیت	شاخه‌های مرتبط

فرآیند نرم افزار

- فرایند نرم افزار مجموعه ای سازمان یافته از فعالیت ها و کارهاست که با هدف توسعه، تکامل، نگهداری و مدیریت نرم افزار انجام می شود. این فرایند، چارچوبی منظم برای انجام مراحل مختلف تولید نرم افزار فراهم می کند تا محصول نهایی با کیفیت بالا، قابل اعتماد و مطابق با نیازهای کاربران ارائه شود.
- همچنین این فرایند تضمین می کند که نرم افزار توسعه یافته از نظر عملکرد، هزینه، زمان بندی و کیفیت در حد مطلوب قرار داشته باشد.
- به طور کلی، فرایند نرم افزار شامل مراحل است. این مراحل عبارتند از:
 - برنامه ریزی و تحلیل نیازمندی ها
 - طراحی نرم افزار
 - پیاده سازی
 - تست و ارزیابی
 - استقرار و نگهداری
- این فعالیت ها در قالب مدل های مختلف فرایند توسعه (مانند مدل آبشاری، تکرار شونده یا مارپیچ) سازماندهی می شوند.

- هدف اصلی فرایند نرم افزار، بهبود کیفیت، سرعت و قابلیت اطمینان تولید نرم افزار با استفاده از روش های منظم، مستندسازی و استانداردهای مهندسی است تا نرم افزار در زمان معین و با هزینه مناسب تولید شود.
- این فرایند به مدیریت چرخه عمر نرم افزار در طول زمان کمک می کند و تضمین می کند که محصول نهایی با نیازهای واقعی کاربران هماهنگ باشد.
- از مزایای این فرایند می توان به افزایش کنترل و پیش بینی پذیری پروژه، کاهش خطاها و دوباره کاری، بهبود هماهنگی بین تیم ها و ذینفعان، افزایش رضایت مشتری، و تسهیل در نگهداری و توسعه آینده نرم افزار اشاره کرد.

انواع مدل های فرایند نرم افزار عبارتند از:

الف) مدل جریان کار (Workflow Model)

- مدل جریان کار به مجموعه ای از فعالیت ها و وظایف مرتبط و متوالی اشاره دارد که برای انجام یک کار نرم افزاری لازم است.
- در این مدل، ترتیب انجام فعالیت ها، وابستگی ها و جریان کار (از شروع تا پایان یک فرایند) نمایش داده می شود. به طور معمول با نمودارهایی مانند فلوچارت یا نمودار جریان کار نشان داده می شود که مراحل اجرایی فرایند را به صورت گام به گام و موازی توصیف می کند.
- هدف مدل جریان کار، تسهیل مدیریت فرایندها، زمان بندی مناسب و بهبود هماهنگی بین فعالیت ها است.
- مزایای مدل جریان کار عبارتند از: افزایش بهره وری با بهبود مدیریت وظایف، شفافیت فرایندها و تسهیل در درک و پیگیری مراحل کار توسط اعضای تیم، کاهش خطاها و دوباره کاری ها با تعریف دقیق وظایف، تسهیل همکاری و هماهنگی بین اعضای تیم، امکان اتوماسیون فرایندهای تکراری برای صرفه جویی در زمان و افزایش کیفیت خدمات و محصولات، بهبود پاسخگویی سازمان به نیازهای مشتریان، و فراهم کردن زمینه ای برای بهبود مستمر و توسعه فرایندهای کسب و کار.
- مثال: یک شرکت نرم افزاری در تهیه و تحویل پروژه های خود از مدل جریان کار استفاده می کند. در این مدل، مراحل مختلفی مثل تحلیل نیازمندی ها، طراحی، برنامه نویسی، تست و استقرار به صورت دقیق تعریف و بین افراد تیم تقسیم شده است. گردش کار به گونه ای طراحی شده که پس از اتمام هر مرحله، گزارش آن برای مرحله بعد ارسال می شود و هر تیم وظیفه و زمان مشخصی دارد. این فرایند باعث می شود پروژه ها به موقع، با کیفیت بهتر و با همکاری تیم ها تحویل داده شود و میزان خطاها و دوباره کاری ها کاهش یابد.

(ب) مدل جریان داده یا فعالیت (Data Flow or Activity Model)

- این مدل روی جریان داده‌ها و اطلاعات در میان فرآیندهای مختلف تمرکز دارد. داده‌هایی که وارد هر مرحله می‌شوند، چگونه پردازش می‌شوند و چه خروجی‌هایی تولید می‌شود، در این مدل تحلیل و نمایش داده می‌شود.
- نمودار جریان داده (DFD) یکی از ابزارهای معروف این مدل است که نشان‌دهنده انتقال داده‌ها و عملیات انجام شده روی آن‌ها بین اجزای مختلف نرم‌افزار است. این مدل به تحلیل و درک بهتر فرآیندهای داده‌محور کمک می‌کند.
- هدف مدل جریان داده یا فعالیت همان نمایش چگونگی حرکت داده‌ها و اطلاعات در یک سیستم یا فرآیند است. این مدل موجب سهولت در تحلیل و بهینه‌سازی جریان داده‌ها برای بهبود عملکرد سیستم و شناسایی نقاط ضعف در پردازش داده‌ها و فعالیت‌ها است.
- مزایای مدل جریان داده یا فعالیت عبارتند از: تصویرسازی واضح و گرافیکی جریان داده‌ها و فعالیت‌ها برای درک بهتر سیستم، شناسایی دقیق فرآیندها و مسیر عبور داده‌ها در سیستم‌های پیچیده، تسهیل در تحلیل نیازمندی‌ها و بهبود فرآیندها، توسعه تدریجی مدل‌ها از دید کلی به جزئیات بیشتر، کاهش خطاها و افزایش هماهنگی بین بخش‌های مختلف، کمک به بهینه‌سازی عملکرد سیستم با شناسایی گلوگاه‌ها و تسهیل ارتباطات بین اعضای تیم فنی و کاربران نهایی.
- مثال: در یک سیستم بانکی، مدل جریان داده فرآیندهای ورود اطلاعات مشتری، پردازش تراکنش‌ها، اعتبارسنجی و ذخیره‌سازی داده‌ها را به صورت گرافیکی نمایش می‌دهد که به توسعه‌دهندگان کمک می‌کند نقاط ضعف و بهبودهای لازم در سیستم را شناسایی کنند.

پ) مدل نقش / فعالیت (Role/Activity Model)

- این مدل فعالیت‌ها را بر اساس نقش‌ها و مسئولیت‌های افراد یا واحدهای مختلف در فرآیند توسعه نرم‌افزار طبقه‌بندی می‌کند. این مدل مشخص می‌کند که هر نقش چه وظایفی دارد و چگونه با دیگر نقش‌ها تعامل می‌کند.
- این مدل به تعیین مسئولیت‌ها، بهبود ارتباطات تیمی و تعریف وظایف کمک می‌کند. به بیان دیگر، مبنای سازماندهی فعالیت‌ها بر اساس نقش افراد در فرآیند نرم‌افزار است.
- هدف این مدل، تعیین دقیق روابط بین فعالیت‌ها و نقش‌ها برای تعریف سازمان‌یافته وظایف و مسئولیت‌ها است.
- مزایای این مدل شامل روشن کردن نقش‌ها و مسئولیت‌های هر فرد یا واحد در فرایندها، افزایش شفافیت در روابط بین فعالیت‌ها و نقش‌ها، تسهیل برنامه‌ریزی و تخصیص منابع انسانی به وظایف مشخص، بهبود همکاری و هماهنگی بین تیم‌ها و افراد مختلف، کمک به شناسایی فرایندهای پیچیده سازمانی و مدیریت بهتر آن‌ها، افزایش بهره‌وری با تمرکز بر نقش‌ها و مسئولیت‌ها، و تسهیل در آموزش و انتقال دانش به اعضای جدید تیم است.
- مثال: در پروژه نرم‌افزاری، این مدل مشخص می‌کند تحلیل‌گر سیستم وظیفه جمع‌آوری نیازمندی‌ها را دارد، برنامه‌نویس کد را پیاده‌سازی می‌کند و تیم تست مسئول آزمون نرم‌افزار است، که روابط همکاری و توالی فعالیت‌ها را به وضوح نشان می‌دهد.

تمرین ۱

۱- یک سناریوی واقعی یا فرضی از پروژه نرم‌افزاری (مانند توسعه یک اپلیکیشن موبایل) را انتخاب کنید و توضیح دهید که چگونه با آن مدل فرایند می‌توان پروژه را مدیریت کرد. این شامل توصیف مراحل یا فعالیت‌ها، نقش‌ها و جریان داده‌ها در پروژه است.

مهلت تحویل: ۲ مهرماه ۱۴۰۴

موفق باشید



دانشگاه شهید باهنر کرمان

به نام دانای نیا موخته

درس: مهندسی نرم افزار
ترم: نیم سال اول (۰۴-۰۵)
جلسه ۲

دپارتمان: مهندسی کامپیوتر
مدرس: دکتر شیما شفیعی

انواع مدل های توسعه نرم افزار با تاکید بر هدف

2

- رهیافت آبشاری (Waterfall)
- توسعه تکاملی (Evolutionary Development)
- تبدیل رسمی (Formal Transformation)
- توسعه مبتنی بر قطعات یا مونتاژ سیستم با قطعات قابل استفاده مجدد (Component-Based Development)

جدول ۱. مقایسه مدل های توسعه نرم افزار

مثال	عیب	مزیت	مدل توسعه نرم افزار
پروژه های با نیازهای ثابت و مشخص مثل پروژه های دولتی	تغییرات و اصلاحات پس از هر مرحله سخت و هزینه بر است	ساختار منظم و مرحله ای	۱- رهیافت آبشاری
نرم افزارهای آزمایشی که به تدریج کامل می شوند	ممکن است زمان پروژه طولانی شود و مدیریت آن پیچیده شود	قابل انعطاف و پذیرش تغییرات	۲- توسعه تکاملی
پروژه های نظامی و فضایی که نیاز به صحت بالا دارند	پیاده سازی پیچیده و زمان بر است	تضمین صحت و دقت بالا	۳- تبدیل رسمی
تولید سیستم هایی با استفاده از کامپوننت های نرم افزاری از قبل توسعه یافته مانند سیستم های اسمبل شده	ممکن است قطعات سازگار به سختی پیدا شوند و کیفیت سیستم کاهش یابد	صرفه جویی در زمان و هزینه با استفاده مجدد قطعات	۴- مونتاژ سیستم با قطعات قابل استفاده مجدد

الف). یک مدل خطی و ترتیبی است که در آن فرایند توسعه نرم افزار به چند فاز متوالی تقسیم می شود و هر فاز پس از کامل شدن، فاز بعدی آغاز می شود. این مدل برای پروژه هایی با نیازمندی های ثابت مناسب است.

ب) در این مدل نرم افزار به صورت تدریجی و افزایشی توسعه می یابد. نسخه های اولیه به کاربران ارائه شده و با دریافت بازخوردها، نسخه های بعدی بهبود می یابند. این روش برای پروژه های پیچیده و با نیازهای متغیر کاربرد دارد.

پ) یک مدلی که در آن توسعه نرم افزار بر پایه روش های رسمی و ریاضی برای تضمین صحت و کیفیت نرم افزار انجام می شود. این روش بیشتر در پروژه های حساس و بحرانی مانند سیستم های نظامی و فضایی به کار می رود.

ت) در این روش سیستم نرم افزاری از مونتاژ قطعات نرم افزاری از پیش ساخته شده و آزموده شده تشکیل می شود. این مدل باعث صرفه جویی در زمان و هزینه توسعه می شود اما نیازمند قطعات سازگار و کیفیت اطمینان بخش است.

هزینه های مهندسی نرم افزار

- هزینه های مهندسی نرم افزار معمولاً به دو بخش اصلی تقسیم می شوند: توسعه نرم افزار و تست آن. تقریباً ۶۰ درصد هزینه ها مربوط به بخش توسعه است و ۴۰ درصد باقی مانده به تست اختصاص دارد.

الف) هزینه های توسعه نرم افزار

بخش توسعه شامل تمامی فعالیت هایی است که در طراحی، برنامه نویسی، مستندسازی، و ایجاد نرم افزار از ابتدا انجام می شود. این فاز عمدتاً به نیروی انسانی متخصص مانند توسعه دهندگان نرم افزار نیاز دارد و شامل هزینه هایی مانند حقوق توسعه دهندگان، ابزارهای توسعه، محیط های برنامه نویسی، و مدیریت پروژه می شود. توسعه نرم افزار معمولاً بیشترین بخش هزینه را دارد زیرا نیازمند متخصصین ماهر و زمان زیادی است و تاثیر زیادی بر کیفیت و عملکرد نهایی محصول دارد.

ب) هزینه های تست نرم افزار

بخش تست نرم افزار شامل فعالیت هایی برای شناسایی عیوب و نواقص نرم افزار است که به صورت سیستماتیک و دقیق انجام می شود. تست نرم افزار وظیفه تضمین کیفیت محصول و اطمینان از عملکرد درست و بدون خطا را دارد. هزینه تست شامل نیروی انسانی برای اجرای تست ها، ابزارهای اتوماسیون تست، محیط های شبیه سازی، و زمان صرف شده برای شناسایی و رفع اشکالات است. هرچند تست کمتر از توسعه هزینه بر است، اما نقش حیاتی در کاهش هزینه های احتمالی بعدی (مانند بعد از تولید) و افزایش رضایت کاربر دارد.

این درصدها نشان دهنده اهمیت هر دو بخش توسعه و تضمین کیفیت در مهندسی نرم افزار هستند و شرکت ها معمولاً باید تعادلی بین این دو برقرار کنند تا محصول نهایی مطمئن و کارآمد باشد، ضمن اینکه هزینه ها هم مدیریت می شوند.

سیستم پشتیبانی

- کیس (CASE)، در مهندسی نرم افزار به معنای سیستم های نرم افزاری است که از فعالیت های مختلف فرآیند توسعه نرم افزار به صورت خودکار پشتیبانی می کنند. به بیانی دیگر، این سیستم ها ابزارهای کامپیوتری هستند که به مهندسان نرم افزار کمک می کنند تا مراحل مختلف چرخه عمر نرم افزار مثل تحلیل، طراحی، برنامه نویسی، آزمون، تست و نگهداری نرم افزار را به شکل خودکار یا نیمه خودکار انجام دهند.
- هدف از استفاده از کیس ها همان افزایش کیفیت، کاهش خطا و افزایش سرعت توسعه نرم افزار است.
- ابزارهای CASE، مانند نرم افزارهای طراحی، مدیریت پروژه، مدل سازی، اشکال زدایی و آزمون نرم افزار هستند که در فرآیند توسعه نرم افزار به صورت ابزار کمکی به کار می روند و به تیم توسعه کمک می کنند تا بتوانند بهتر و سریع تر محصول نهایی را تولید کنند.

خصیصه های نرم افزار کارا

- خصیصه های نرم افزار کارا و مناسب عبارتند از:

الف) قابلیت نگهداری

ب) قابلیت اتکا

پ) کار آمدی

ت) قابلیت استفاده

الف). قابلیت نگهداری (Maintainability)

قابلیت نگهداری به توانایی نرم افزار در پذیرش تغییرات، اصلاح خطاها و ارتقاء اشاره دارد. نرم افزاری که قابلیت نگهداری بالایی دارد، هزینه های اصلاح و به روزرسانی آن کمتر است.

مثال: یک نرم افزار حسابداری شرکتی که به طور مکرر با قوانین مالی جدید تطبیق داده می شود و قابلیت اعمال تغییرات آسان دارد، نمونه ای از نرم افزار با قابلیت نگهداری بالا است.

ب). قابلیت اتکا (Reliability)

این قابلیت نشان دهنده پایداری و صحت عملکرد نرم افزار در طول زمان است. نرم افزاری با قابلیت اتکای بالا، میزان خطا و خرابی کمتری دارد که باعث کاهش هزینه های ناشی از تعمیر و پشتیبانی می شود.

مثال: نرم افزارهای کنترل پرواز که باید بدون خطا و با دقت بالا کار کنند و هزینه خطا در آنها بسیار زیاد است، نمونه ای از نرم افزارهای با قابلیت اتکا بالا هستند.

ادامه

پ). کارآمدی (Efficiency)
کارآمدی به میزان استفاده بهینه از منابع سیستم (مانند حافظه، زمان پردازش) توسط نرم افزار اشاره دارد. نرم افزاری که کارآمد است، هزینه های سخت افزاری و انرژی را کاهش می دهد.
مثال: نرم افزارهای مدیریت داده در سرورهای بانکی که باید با کمترین مصرف منابع بیشترین پردازش را انجام دهند، نمونه های بارز نرم افزارهای کارآمد هستند.

ت). قابلیت استفاده (Usability)
قابلیت استفاده به سهولت استفاده کاربران از نرم افزار اشاره دارد. نرم افزاری که کاربرپسند باشد، نیاز به آموزش کمتر و حمایت فنی کمتری دارد که باعث کاهش هزینه های ضمنی می شود.
مثال: نرم افزارهای سازمانی با رابط کاربری ساده که کاربران بدون نیاز به آموزش طولانی قادر به استفاده از آن باشند، نمونه ای از نرم افزارهایی با قابلیت استفاده بالا هستند.

وظایف تخصصی و اخلاقی

- محرمانگی
- صلاحیت
- حقوق معنوی
- سوء استفاده از کامپیوتر

وظایف تخصصی و اخلاقی در حوزه مهندسی نرم افزار و فناوری اطلاعات شامل موارد زیر است:

الف) محرمانگی (Confidentiality)

حفظ و نگهداری اطلاعات خصوصی و حساس کاربران و سازمان ها از اهمیت بسیار زیادی برخوردار است. متخصصان باید از افشای اطلاعات بدون اجازه جلوگیری کنند و از داده ها به صورت قانونی و اخلاقی محافظت نمایند.

ب) صلاحیت (Competence)

فرد باید دارای تخصص، مهارت و دانش لازم برای انجام کارهای محوله باشد و در زمینه کاری خود به روز باشد تا بتواند خدمات با کیفیت و ایمنی ارائه دهد. عدم صلاحیت می تواند به خسارت های جبران ناپذیری منجر شود.

پ) حقوق معنوی (Intellectual Property Rights) حقوق مالکیت فکری افراد و شرکت‌ها باید رعایت شود. استفاده بدون اجازه از نرم‌افزارها، کدها، الگوریتم‌ها و دیگر آثار فکری دیگران، غیرقانونی و غیراخلاقی است. احترام به قوانین کپی‌رایت و مالکیت معنوی از وظایف حرفه‌ای است.

ت) سوءاستفاده از کامپیوتر (Computer Misuse) استفاده‌های نادرست و غیرقانونی از سیستم‌های کامپیوتری مانند هک، نفوذ غیرمجاز، تخریب داده‌ها، سرقت اطلاعات و استفاده از وسایل کامپیوتری برای اهداف غیرقانونی یا غیر اخلاقی ممنوع است. متخصصین باید در برابر چنین فعالیت‌هایی مقاومت کرده و از آن جلوگیری کنند.

این اصول اخلاقی براساس استانداردهای بین‌المللی مانند کد اخلاق IEEE و ACM تدوین شده‌اند و به حفظ اعتبار حرفه و افزایش اعتماد کاربران و جامعه کمک می‌کنند.

تمرین ۲

۱- پروژه‌ای را تصور کنید که یک مدل توسعه برای آن مناسب باشد و دلیل انتخاب این مدل و رد سایر مدل های توسعه نرم افزار را توضیح دهید.

مهلت تحویل: ۱۱ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیا موخته

درس: مهندسی نرم افزار

ترم: نیم سال اول [۰۴-۰۵]

جلسه ۳

دپارتمان: مهندسی کامپیوتر

مدرس: دکتر شیما شفیعی

مدل های توسعه نرم افزاری با تاکید بر معماری

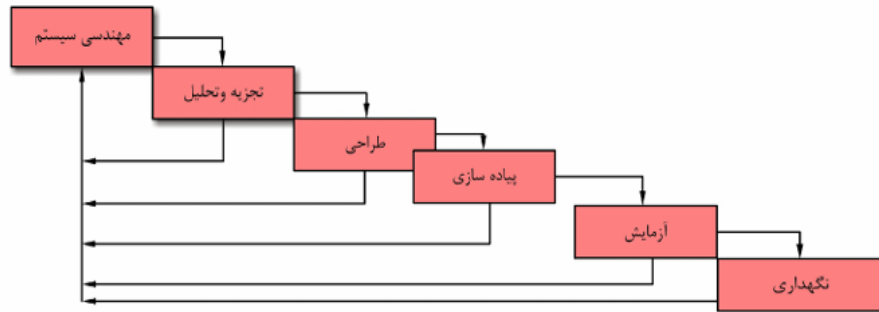
1

جدول ۱. مقایسه مدل های توسعه نرم افزار

ویژگی	Linear Sequential Model (مدل ترتیبی خطی)	Waterfall Model (مدل آبشاری)	Prototyping Model (مدل نمونه سازی)
نوع مدل	ترتیبی، مرحله ای و خطی	ترتیبی، مرحله ای	تکراری، تعاملی و تکاملی
مراحل	جمع آوری نیازها، طراحی، پیاده سازی، تست، نگهداری	جمع آوری نیازها، طراحی (منطقی و فیزیکی)، اجرا، تست، نگهداری	ساخت نمونه اولیه، بازخورد، بهبود و تکرار نمونه
مزایا	سادگی، شفافیت، مستندسازی دقیق، مراحل مشخص	مستندات کامل، ثبات در محدوده پروژه، قابل پیش بینی بودن	شناسایی زودهنگام نیازها، کاهش ریسک، تعامل مشتری
معایب	انعطاف پذیری بسیار کم، دشوار در تغییر نیازها	انعطاف پذیری محدود، دشواری در اصلاحات بعد از فازها	هزینه و زمان بیشتر، پیچیدگی مدیریت نمونه ها
کارایی	پروژه های با نیازمندی های کاملاً مشخص و ثابت	پروژه های با نیازهای واضح و ثابت	پروژه های نیازمند بازخورد مستمر و در حال تغییر
مثال	پروژه توسعه نرم افزاری نظام بانکی با الزامات دقیق	سیستم حسابداری با نیازهای مشخص	طراحی اپلیکیشن موبایل با نیاز به رابط کاربری پویا

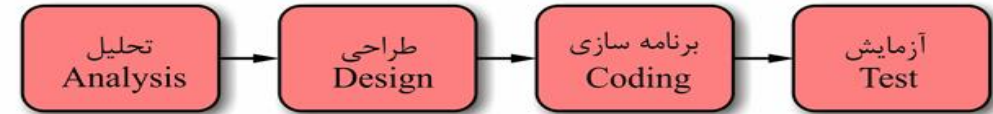
جدول ۱. مقایسه مدل های توسعه نرم افزار

ویژگی	RAD Model (مدل توسعه سریع)
نوع مدل	تکراری، موازی، مبتنی بر بخش بندی پروژه
مراحل	تقسیم پروژه به بخش های کوچک، توسعه موازی، ادغام بخش ها
مزایا	سرعت بالا در توسعه، انعطاف پذیری، کاهش ریسک، مشارکت بیشتر کاربر
معایب	نیاز به تیم های حرفه ای، مناسب نبودن برای پروژه های بزرگ و پیچیده
کارایی	پروژه های کوچک تا متوسط با نیازمندی های شناخته شده و زمان محدود
مثال	توسعه سیستم مدیریت سفارش با بخش بندی ماژول ها و تحویل سریع



مدل آبشاری

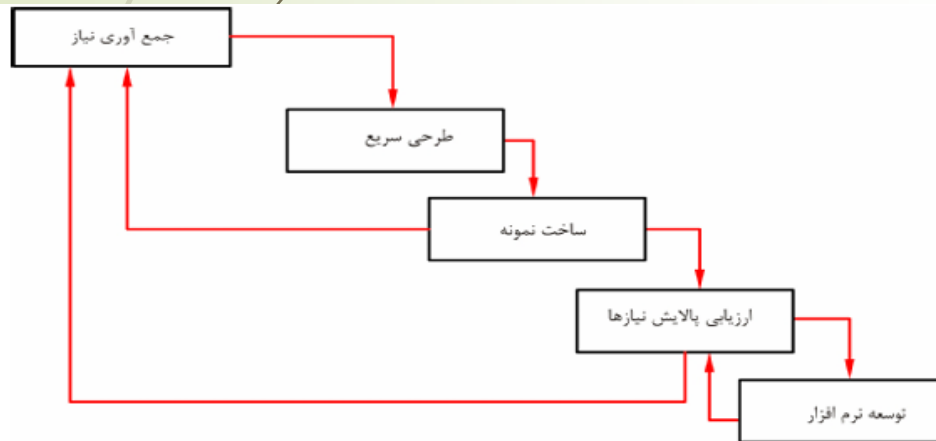
مدل ابشاری



مهندسی سیستم / اطلاعات : شناسایی و تحلیل نیازها

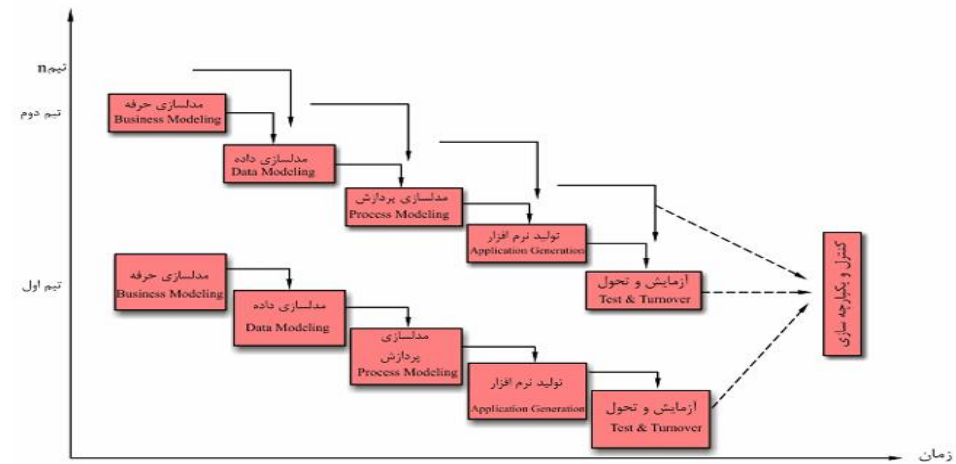
مدل ترتیبی خطی

مد ترتیبی خطی



شکل ۵-۲ نمونه سازی در مدل آبشاری

مدل نمونه سازی



شکل ۶-۲ مدل توسعه سریع کاربرد

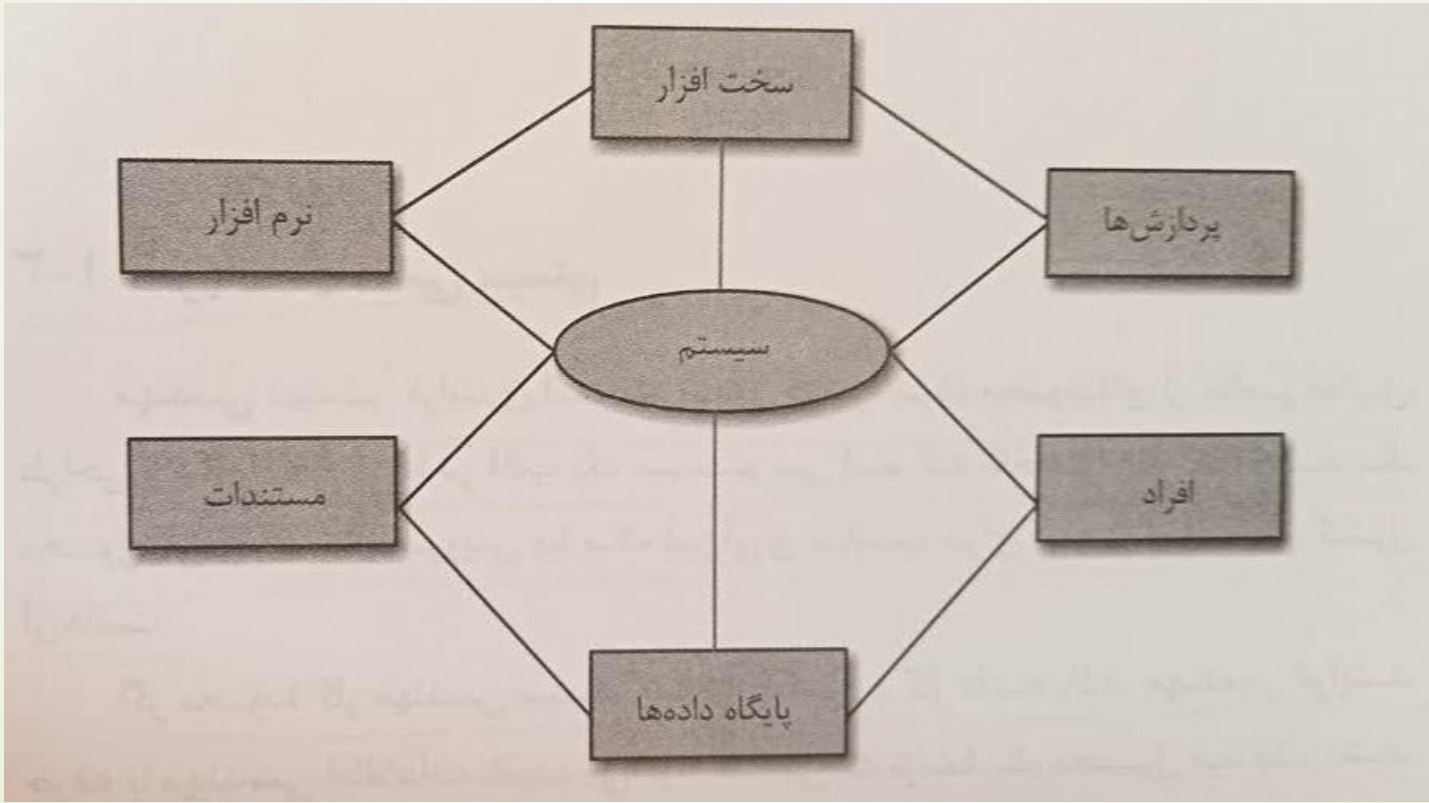
مدل توسعه سریع کاربرد

شکل ۱. انواع مدل های توسعه نرم افزار

سیستم (به مجموعه‌ای منظم و هدفمند از اجزاء (قطعات، موجودیت‌ها یا زیرسیستم‌ها) گفته می‌شود که این اجزا با هم ارتباط متقابل دارند و با همکاری یکدیگر، در جهت تحقق یک هدف مشخص عمل می‌کنند.

جدول ۲. جزئیات سیستم

ویژگی	توضیح
اجزا	سیستم از چند بخش یا مؤلفه تشکیل شده است. هر جزء ممکن است خودش یک سیستم کوچکتر (زیرسیستم) باشد.
ارتباط بین اجزا	اجزا با هم تعامل دارند (از طریق داده، انرژی یا کنترل)، یعنی مستقل عمل نمی‌کنند.
هدف مشخص	همه اجزا برای رسیدن به یک یا چند هدف کلی طراحی شده‌اند. بدون هدف، مجموعه سیستم نیست.
مرز	سیستم یک محدوده مشخص دارد؛ آنچه داخل این مرز است، «درون سیستم» محسوب می‌شود، و بقیه محیط سیستم هستند.
ورودی و خروجی	سیستم معمولاً ورودی (Input) می‌گیرد، آن را پردازش می‌کند، و خروجی (Output) تولید می‌کند.
محیط	سیستم‌ها با محیط بیرونی خود در تعامل‌اند. ممکن است از آن ورودی بگیرند یا بر آن اثر بگذارند.
بازخورد	در سیستم‌های پویا یا کنترل‌شونده، بازخورد برای اصلاح عملکرد اهمیت دارد.



شکل ۱.۲ اجزای سیستم های مبتنی بر رایانه

سیستم

- سیستم یک مجموعه هدفمند از اجزاء یا قطعات مختلف است که به صورت مرتبط و سازمان یافته با یکدیگر تعامل دارند و همکاری می کنند تا یک یا چند هدف مشخص را تحقق بخشند. این اجزاء می توانند فیزیکی، منطقی، مفهومی یا ترکیبی از این ها باشند.
- **ارتباط بین اجزاء:** هر جزء سیستم با سایر اجزاء در ارتباط است و عملکرد هر بخش می تواند روی بخش های دیگر تأثیر بگذارد. ارتباطات باعث هماهنگی و هم افزایی می شوند.
 - **هدف مشترک:** همه اجزاء به منظور تحقق یک هدف کلی در کنار هم کار می کنند؛ بدون هدف، مجموعه ای از اجزاء صرفاً یک دسته پراکنده است، نه سیستم.
 - **مرز مشخص:** سیستم دارای یک مرز یا محدوده تعریف شده است که تعیین می کند چه چیزی داخل سیستم است و چه چیزی خارج از آن (محیط).
 - **ورودی و خروجی:** سیستم معمولاً ورودی هایی از محیط دریافت می کند، آن ها را پردازش می کند و خروجی هایی به محیط می دهد. ورودی و خروجی ممکن است داده، انرژی، مواد یا اطلاعات باشند.
 - **بازخورد:** بسیاری از سیستم ها دارای مکانیسم های بازخورد هستند تا عملکرد خود را ارزیابی و در صورت نیاز اصلاح کنند، به منظور بهبود یا حفظ کارایی.
 - **پویا بودن:** سیستم ها معمولاً پویا هستند و در طول زمان تغییر می کنند و خود را با شرایط محیطی سازگار می کنند.

سیستم های مبتنی بر نرم افزار

الف) سیستم های کامپیوتری تکنیکی

ب) سیستم های اجتماعی- تکنیکی

جدول ۳. مقایسه سیستم ها

ویژگی	سیستم‌های کامپیوتری تکنیکی	سیستم‌های اجتماعی-تکنیکی
تعریف	سیستم‌هایی که شامل سخت‌افزار، نرم‌افزار و اجزای فنی هستند.	سیستم‌هایی که شامل اجزای فنی و انسانی به صورت همزمان می‌باشند
اجزاء اصلی	سخت‌افزار، نرم‌افزار، شبکه، پایگاه داده	انسان‌ها (کاربران، مدیران)، فرآیندها، فناوری
تمرکز اصلی	بهینه‌سازی عملکرد و جریانی پردازش داده، اتوماسیون	تعامل بین انسان و فناوری، بهبود کارایی اجتماعی و فنی
هدف	افزایش کارایی، دقت و سرعت در پردازش اطلاعات	افزایش بهره‌وری، رضایت کاربران و هماهنگی بین انسان و فناوری
ماهیت سیستم	تکنیکی و ماشینی	ترکیبی از تکنولوژی و رفتار انسانی
مثال‌ها	سیستم‌های مدیریت پایگاه داده، سیستم‌های عامل، شبکه‌های کامپیوتری	سیستم‌های سازمانی، تیم‌های کاری، سیستم‌های خدمات مشتری
چالش‌ها	مشکلات سخت‌افزاری، نرم‌افزاری، خطاهای فنی	مسائل انسانی مثل مقاومت در برابر تغییر، فرهنگ سازمانی
طراحی و پیاده‌سازی	بیشتر متمرکز بر مهندسی نرم‌افزار و سخت‌افزار	نیازمند توجه به عوامل انسانی، روانشناسی و مهندسی سیستم‌ها

تمرین ۳

۱- یک سیستم اجتماعی - تکنیکی را در نظر بگیرید و از جنبه های مختلف آن رت آنالیز کنید.

مهلت تحویل: ۱۱ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیا موخته

درس: مهندسی نرم افزار

ترم: نیم سال اول [۰۴-۰۵]

جلسه ۴

دپارتمان: مهندسی کامپیوتر

مدرس: دکتر شیما شفیعی

لایه های مهندسی نرم افزار

- در فرآیند مهندسی نرم افزار، فعالیت ها در چند لایه مفهومی تقسیم می شوند. این لایه ها کمک کرده تا نرم افزار با کیفیت، قابل نگهداری و قابل توسعه ساخته بشوند.

1. Quality Focus (تمرکز بر کیفیت)

- مرکزی ترین لایه، مربوط به تضمین کیفیت نرم افزار، جریان داشتن در تمام مراحل چرخه توسعه، یعنی هر فعالیت باید با در نظر گرفتن کیفیت انجام بشود (مثل قابلیت اطمینان، عملکرد، امنیت و ...).

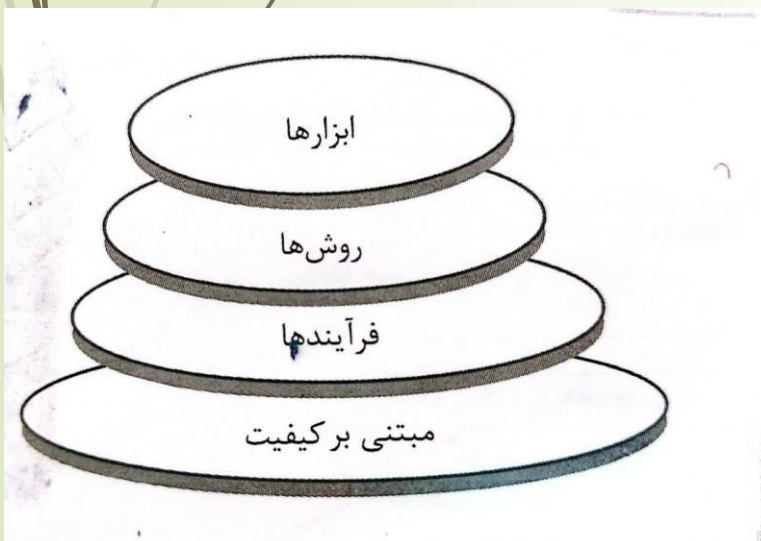
2. Process (فرآیند)

- چارچوبی برای مدیریت پروژه و تولید نرم افزار.
- شامل مدل های فرآیند مانند: Waterfall

3. Methods (روش‌ها)

روش های فنی برای تحلیل، طراحی، پیاده‌سازی و تست نرم‌افزار و چرخه توسعه.
شامل: مدل‌سازی، الگوریتم‌ها، طراحی معماری، طراحی واسط، و غیره.

4. Tools (ابزارها)



نرم‌افزارها و ابزارهایی که برای خودکارسازی و تسهیل فعالیت‌های بالا استفاده می‌شوند.
ابزارهای مدل‌سازی (UML)

جدول ۳. مقایسه مهندسی ها

هدف	تعریف	حوزه
تضمین عملکرد صحیح و کارآمد سیستم‌های پیچیده در کل چرخه عمر آن‌ها، رفع تضادها و افزایش هماهنگی بین اجزا	رویکرد بین‌رشته‌ای برای طراحی، مدیریت، و بهینه‌سازی سیستم‌های پیچیده؛ تمرکز بر عملکرد کل سیستم و هماهنگی اجزا	مهندسی سیستم
بهبود مدیریت و جریان داده‌ها، ایجاد زیرساخت‌های اطلاعاتی کارا، حمایت از اهداف سازمانی در حوزه فناوری اطلاعات	تمرکز بر طراحی، توسعه، پیاده‌سازی و مدیریت سیستم‌های فناوری اطلاعات و داده‌ها	مهندسی فناوری اطلاعات
تولید محصولی متناسب با نیاز بازار، بهبود کیفیت و کارایی محصول، کنترل هزینه‌ها و تضمین سطح عملکرد مورد انتظار	فرآیند طراحی، توسعه، تولید و بهبود محصولات فیزیکی یا دیجیتال، شامل برنامه‌ریزی، مدل‌سازی و کنترل فرآیندها	مهندسی محصول

سلسله مراتب مهندسی سیستم

سلسله مراتب مهندسی سیستم به ترتیب معمول شامل مراحل زیر است:

- تعریف مسئله
- تحلیل نیازمندی‌ها
- طراحی سیستم کلی
- تحلیل زیرسیستم‌ها و اجزا
- طراحی جزئیات هر زیرسیستم
- پیاده‌سازی و ساخت
- تست و ارزیابی عملکرد
- بهره‌برداری و نگهداری
- ارتقاء و بهبود مستمر

این مراحل به صورت چرخه‌ای هستند و در طول عمر سیستم ممکن است بارها تکرار و اصلاح شوند تا سیستم بهترین عملکرد را در شرایط واقعی داشته باشد. این سلسله مراتب رویکردی کل‌نگر برای مدیریت طراحی و توسعه سیستم‌های پیچیده است.

مدل سازی سیستم

- مدل سازی سیستم فرآیندی است که در آن یک سیستم پیچیده به صورت ساده شده، با استفاده از ابزارها و روش های ریاضیاتی، گرافیکی یا کامپیوتری نمایش داده می شود تا رفتار، ویژگی ها و روابط داخلی آن بهتر درک، تحلیل و پیش بینی شود.
- این مدل ها می توانند بر پایه قوانین فیزیکی یا داده های تجربی ساخته شوند و به طراحان یا محققان کمک می کنند تا بدون نیاز به آزمایش های واقعی و پرهزینه، کارایی سیستم ها را بهینه کرده و ریسک ها را کاهش دهند.
- مدل سازی سیستم شامل ایجاد مدل هایی مانند مدل های ریاضی، گرافیکی، مفهومی، ذهنی یا فیزیکی است که از آنها برای تحلیل و بهبود سیستم ها در زمینه های مختلف کاربردی مانند مهندسی، علوم پزشکی، فناوری اطلاعات و علوم اجتماعی استفاده می شود.
- این فرآیند اولین گام در طراحی کنترل کننده ها و مدیریت سیستم های دینامیکی نیز به شمار می آید.

بخش های مهندسی سیستم

- شناخت نیازها: اولین گام در مهندسی سیستم، شناسایی و ثبت دقیق نیازهای کاربران و ذینفعان سیستم است تا پایه‌ای برای طراحی سیستم فراهم شود.
- طراحی مفهومی: ایجاد مدل‌ها و طرح‌های اولیه سیستم بر اساس نیازها و اهداف تعیین شده، که شامل بررسی گزینه‌های مختلف طراحی و تعیین بهترین راهکار است.
- توسعه و پیاده‌سازی: ساخت و تولید اجزاء فیزیکی، نرم‌افزارها و زیرسامانه‌های سیستم براساس طراحی انجام شده.
- تست و ارزیابی: بررسی عملکرد سیستم و ارزیابی تطابق آن با نیازمندی‌ها، به منظور اطمینان از کارایی و کیفیت سیستم.
- مدیریت چرخه عمر سیستم: شامل نگهداری، به‌روزرسانی و بهبود سیستم در طول زمان برای حفظ کارایی و پاسخگویی به تغییرات نیازها.
- هماهنگی تخصص‌ها و تیم‌ها: مهندسی سیستم بین رشته‌ای است و نیازمند همکاری بین تخصص‌های مختلف (مهندسی نرم‌افزار، مکانیک، برق، مدیریت پروژه و...) برای یکپارچه‌سازی سیستم است.
- مدلسازی و شبیه‌سازی: استفاده از مدل‌ها برای تحلیل و پیش‌بینی رفتار سیستم قبل از ساخت واقعی آن.
- این بخش‌ها به همراه ابزارها و روش‌های مدیریتی، به مهندسان سیستم امکان می‌دهد تا سیستم‌های پیچیده را به صورت ساختارمند طراحی و کنترل کنند و از موفقیت پروژه اطمینان حاصل نمایند.

تمرین ۴

۱- بررسی و تحلیل یک سیستم مدیریت کتابخانه را از دیدگاه مهندس سیستم ارائه دهید.

مهلت تحویل: ۱۸ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیا موخته

درس: مهندسی نرم افزار

ترم: نیم سال اول [۰۴-۰۵]

جلسه ۵

دپارتمان: مهندسی کامپیوتر

مدرس: دکتر شیما شفیعی

مهندسی نیازها

- تعریف اول: مهندسی نیازها، روشی منظم برای شناسایی، استخراج، تحلیل، مشخص کردن، اعتبارسنجی و مدیریت نیازها و انتظارات نهادهای مختلف مثل کاربران، مشتریان و ذینفعان برای یک سیستم است؛ این فرایند تکراری است و نیازمندی‌ها را به طور دقیق تعریف، مستندسازی و مدیریت می‌کند.
- تعریف دوم: مهندسی نیازها عبارت است از رویکردی سازمان‌یافته برای تعیین، مستندسازی و مدیریت نیازهای سیستم با هدف دستیابی به اجماع بین نظرات ذینفعان، کاهش ریسک عدم تطابق محصول با نیازها و تسهیل مستندسازی و کنترل تغییرات.
- فرض کنید قرار است یک سیستم فروشگاه اینترنتی طراحی شود. یکی از نیازمندی‌های عملکردی این سیستم می‌تواند این باشد که هر مشتری از نقاط مختلف کشور بتواند با وارد کردن آدرس، فرآیند خرید را کامل کند. یکی از نیازهای آن، می‌تواند سرعت پاسخ‌گویی سرور کمتر از ۲ ثانیه در هر عملیات سفارش باشد.
- این مثال نشان می‌دهد برای پیاده‌سازی چنین سیستمی باید ابتدا نیازها جمع‌آوری، تحلیل و اعتبارسنجی شده و سپس به صورت مستند مدیریت شوند تا محصول مطلوب ذینفعان ایجاد گردد.

مراحل مهندسی نیازها: بیان نیازها

- بیان نیازها یعنی مستندسازی شرایط، اهداف یا انتظاراتی که باید توسط سیستم برآورده شوند تا مسائل اصلی کاربران حل شده یا به اهداف آنان دست پیدا کند؛ این مستندسازی معمولاً در قالب سند نیازمندی‌های نرم‌افزار انجام می‌شود و مبنایی برای توسعه، تست و اعتبارسنجی محصول فراهم می‌آورد.
- اهداف مرحله بیان نیازها عبارت است از:
 - واضح‌سازی انتظارات و اولویت‌های ذینفعان و کاربران.
 - ایجاد معیار ارزیابی محصول نهایی بر اساس نیازها.
 - جلوگیری از سوءتفاهم و تداخل در طول توسعه و اجرا.
 - تسهیل فرآیندهای بعدی مانند طراحی، تست و اعتبارسنجی سیستم.
- در یک پروژه طراحی سامانه ثبت نام دانشگاه، یکی از نیازهای عبارت است از: کاربر باید بتواند با وارد کردن نام کاربری و رمز عبور، وارد سامانه شده و لیست دروس موجود را مشاهده کند. این جمله یک نیاز عملکردی است که به وضوح بیان شده و معیار سنجش و پیاده‌سازی را فراهم می‌کند.

مراحل مهندسی نیازها: تحلیل نیازها

- تحلیل نیازها فرآیندی است که طی آن نیازهای جمع‌آوری شده از ذینفعان بررسی می‌شوند تا شفافیت، کامل بودن، بدون ابهام بودن و هماهنگ بودن نیازها با اهداف پروژه تضمین گردد.
- اهداف تحلیل نیازها عبارت است از:
 - رفع نقص، ابهام و تضاد در نیازها.
 - تضمین ارتباط صحیح و اجماع میان نیازهای ذینفعان.
 - آماده‌سازی نیازها برای تبدیل به مشخصات فنی، طراحی و پیاده‌سازی بدون ریسک خطاهای مفهومی.
- در پروژه‌ای مرتبط با سامانه پرداخت آنلاین قبض، نیاز اولیه این است: کاربر بتواند قبض خود را پرداخت کند. در تحلیل نیازها، مشخص می‌شود باید جزئیات بیان گردد: مثلاً آیا کاربر باید ثبت‌نام کند؟ چه انواع قبض‌هایی قابل پرداخت است؟ آیا پیامک تایید ارسال می‌شود؟ در نتیجه تحلیل، نیاز اصلاح شده چنین خواهد بود: کاربر باید بتواند پس از ثبت‌نام و ورود به سامانه، نوع قبض را انتخاب و مبلغ را وارد کند و پس از پرداخت آنلاین، پیامک تایید به شماره او ارسال شود.
- این مثال نشان می‌دهد تحلیل نیازها چگونه باعث رفع ابهام و تکمیل نیاز می‌شود.

مراحل مهندسی نیازها: تدوین مشخصه نیازها

5

- تدوین مشخصه نیازها فرآیندی است که در آن نیازها، انتظارات و شرایط مورد انتظار از سیستم به زبان ساده، دقیق و بدون ابهام مستند می‌شوند تا برای کلیه افراد پروژه به عنوان یک مرجع معتبر و قابل رجوع مورد استفاده قرار بگیرند. این مشخصات معمولاً به صورت یک سند رسمی (Software Requirements Specification) تهیه می‌شود که پایه طراحی، پیاده‌سازی و آزمون سامانه خواهد بود.
- اهداف تدوین مشخصه نیازها عبارتند از:
 - ایجاد مبنای مشترک برای توسعه، طراحی، و آزمون سیستم
 - تسهیل در پیگیری تغییرات و کنترل نسخه‌های مختلف نیازمندی‌ها در طول عمر پروژه
 - کاهش خطا و سوء تفاهم میان ذینفعان و تیم توسعه با داشتن مرجع واحد و رسمی
- مثال براساس پروژه ساخت سامانه مدیریت کتابخانه دانشگاه عبارتند از: هر کاربر بتواند کتاب مورد نظر را جستجو کند و در صورت موجود بودن، آن را رزرو نماید. سیستم باید امکان جستجوی کتاب بر اساس عنوان، نام نویسنده یا شماره استاندارد ISBN را فراهم کند. سیستم باید وضعیت موجودی هر کتاب را نمایش دهد. سیستم در صورت موجود بودن کتاب، گزینه “رزرو” را فعال کرده و ثبت رزرو را در اکانت کاربر انجام دهد.
- این مثال نشان می‌دهد چگونه نیاز جمع‌آوری شده به صورت مشخصه دقیق، تست‌پذیر و قابل پیاده‌سازی مستند می‌شود.

مراحل مهندسی نیازها: مدل سازی سیستم

- مدل سازی سیستم یکی از مراحل مهندسی نیازها است که در آن، با استفاده از نمودارها، مدل ها و زبان های طراحی، ساختار، رفتار و عملکرد کلی سیستم به شکل انتزاعی و قابل درک ارائه می شود. این مدل ها کمک می کنند تا نیازهای قسمت های مختلف سیستم بهتر فهمیده، تحلیل و ارتباط بین آن ها روشن شود.
- اهداف مدل سازی سیستم عبارتند از:
 - تسهیل فهم دقیق نیازمندی ها و ساختار سیستم توسط همه ذینفعان.
 - برنامه ریزی صحیح برای طراحی و پیاده سازی سیستم.
 - ارتقای ارتباط و تعامل میان اعضای تیم توسعه و مشتری.
- فرض کنید قرار است سیستم مدیریت فروشگاه طراحی شود؛ در مدل سازی سیستم، ابتدا دیاگرام های جریان داده (DFD) ترسیم می شود که نشان می دهد چگونه اطلاعات سفارش مشتریان از ورود به سیستم، پردازش، ذخیره و نهایتاً به انبار و بخش حسابداری منتقل می شود. این مدل به ذینفعان اجازه می دهد جریان داده ها و وظایف سیستم را پیش از کدنویسی بفهمند و تضمین می کند نیازها به درستی منعکس شده اند.

مراحل مهندسی نیازها: اعتبارسنجی نیازها

- فرآیند اعتبارسنجی نیازها شامل بررسی و ارزیابی رسمی نیازهای مشخص شده به منظور اطمینان از اینکه نیازها به درستی تعریف شده‌اند، مورد قبول تمامی ذینفعان هستند و می‌توانند به طور واقعی پیاده‌سازی شوند.
- اهداف اعتبارسنجی نیازها عبارتند از:
 - اطمینان از صحت و قابلیت تحقق نیازها در سیستم نهایی.
 - شناسایی خطاها، نقص‌ها و تناقضات پیش از شروع توسعه عمده.
 - جلب رضایت و تأیید ذینفعان در مورد نیازهای جمع‌آوری شده.
 - کاهش هزینه و زمان ناشی از اصلاحات احتمالی در مراحل بعدی مانند توسعه.
- فرض کنید در پروژه طراحی یک سیستم رزرو بلیت آنلاین، نیاز وجود دارد که کاربر بتواند صندلی خود را در هر زمان پیش از پرواز تغییر دهد. در اعتبارسنجی این نیاز، آزمایش‌هایی انجام می‌شود تا مشخص شود آیا سیستم می‌تواند این قابلیت را به درستی پیاده کند یا خیر و آیا این ویژگی با بقیه عملکردها متناقض نیست. همچنین بازخورد ذینفعان درباره امکان‌پذیری و ضرورت نیاز جمع‌آوری می‌شود و اگر نیاز کامل یا واضح نباشد، اصلاح می‌شود.

مراحل مهندسی نیازها: مدیریت نیازها

- مدیریت نیازها فرایندی است که طی آن، نیازمندی‌های جمع‌آوری شده به صورت منسجم و مداوم پیگیری، کنترل، تغییرات آن‌ها رصد و به روزرسانی می‌شوند تا اطمینان حاصل شود که نیازها در طول پروژه بدون تغییرات بی‌رویه و به درستی تعریف و حفظ شده‌اند و پروژه مطابق با نیازهای حقیقی پیش می‌رود.
- اهداف مدیریت نیازها عبارتند از:
 - انطباق نیازهای مستند شده با خواسته‌های ذینفعان در طول پروژه.
 - کنترل تغییرات نیازمندا به منظور جلوگیری از تغییرات ناخواسته یا زیاد که می‌تواند باعث افزایش هزینه و تأخیر شود.
 - ثبت و پیگیری کامل تغییرات نیاز در طول عمر پروژه.
 - تسهیل ارتباط موثر بین تیم توسعه و ذینفعان درباره وضعیت و تغییرات نیازها.
 - کاهش ریسک خطا و اشتباه در پروژه با اطمینان از سازگاری نیازها در هر مرحله.
- در یک پروژه توسعه نرم‌افزار حسابداری، پس از مرحله آغازین نیازسنجی ممکن است برخی نیازها تغییر کنند، مثلاً افزودن ویژگی گزارش‌گیری جدید پس از بازخورد کاربری. در این مرحله مدیریت نیازها باعث می‌شود این تغییرات به صورت ساخت‌یافته ثبت، ارزیابی اثرات آن روی زمان و هزینه پروژه انجام شده و در نهایت در مستندات نیاز وارد شود و تیم توسعه به روزرسانی گردد تا پروژه هماهنگ با نیازهای جدید پیش رود.

تمرین ۵

با اعمال یک مثال واقعی توضیح دهید، مراحل مهندسی نیازها چگونه می تواند به موفقیت یک پروژه کمک کند.

مهلت تحویل: ۱۸ مهرماه ۱۴۰۴

موفق باشید