



به نام دانای نیآموخته

درس: مهندسی نرم افزار
ترم: نیم سال اول (۱۴۰۵)

جلسه ۱

دپارتمان: مهندسی کامپیووتر
مدرس: دکتر شیما شفیعی

بارم بندی

2

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 مجموعه‌ای از برنامه‌های کامپیوتری، مستندات و داده‌های پیکربندی است که برای انجام وظایف خاص در یک سیستم رایانه‌ای طراحی شده‌اند. نرم افزار نه تنها شامل کدهای اجرایی است بلکه شامل مستنداتی نیز می‌شود که نحوه استفاده، نگهداری و توسعه نرم افزار را توضیح می‌دهند. همچنین داده‌های پیکربندی که برای عملکرد صحیح نرم افزار ضروری هستند نیز بخشی از نرم افزار محسوب می‌شوند.
- نرم افزار، رابط بین کاربر و سخت افزار است که به رایانه فرمان می‌دهد که چگونه کار کند. این مجموعه باعث می‌شود تا کامپیوتر بتواند وظایف متنوعی را به صورت بهینه و مفید انجام دهد.
- مهندسی نرم افزار به معنی استفاده از اصول و روش‌های 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 نیازمندی‌ها، طراحی، پیاده‌سازی، تست و نگهداری نرم افزار است تا محصول نهایی بتواند به طور موثر نیازهای کاربران را پاسخ دهد و با صرفه‌جویی در هزینه‌ها و زمان، به تولید برسد.
- به بیان دیگر، مهندسی نرم افزار کاربرد روش‌ها، تکنیک‌ها و ابزارهای مهندسی برای ساخت نرم افزارهای قابل اطمینان، با کیفیت و مقیاس پذیر است که روی ماشین‌های واقعی به طور کارآمد اجرا می‌شوند.

ادامه

- مهندسی نرم افزار شامل فرایندهای دقیق و سیستماتیک تحلیل نیازمندی‌ها، طراحی، پیاده‌سازی، تست، ارتقاء و نگهداری نرم‌افزارها 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 شخصی‌سازی
 - پ) ابزارها و کتابخانه‌های مورد استفاده برنامه
 - ت) سیستم‌های مدیریت پایگاه داده و ذخیره‌سازی

تفاوت مهندسی نرم افزار و علم کامپیوتر و مهندسی سیستم

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

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

فرآیند نرم افزار

- فرایند نرم افزار مجموعه‌ای سازمان یافته از فعالیت‌ها و کارهاست که با هدف توسعه، تکامل، نگهداری و مدیریت نرم افزار انجام می‌شود. این فرایند، چارچوبی منظم برای انجام مراحل مختلف تولید نرم افزار فراهم می‌کند تا محصول نهایی با کیفیت بالا، قابل اعتماد و مطابق با نیازهای کاربران ارائه شود.
- همچنین این فرایند تضمین می‌کند که نرم افزار توسعه یافته از نظر عملکرد، هزینه، زمان‌بندی و کیفیت در حد مطلوب قرار داشته باشد.
- به طور کلی، فرایند نرم افزار شامل مراحلی است. این مراحل عبارتند از:
 - برنامه‌ریزی و تحلیل نیازمندی‌ها
 - طراحی نرم افزار
 - پیاده‌سازی
 - تست و ارزیابی
 - استقرار و نگهداری
- این فعالیت‌ها در قالب مدل‌های مختلف فرایند توسعه (مانند مدل آبشاری، تکرارشونده یا مارپیچ) سازماندهی می‌شوند.

ادامه

- 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 پر ژوه، کاهش خطاهای دوباره کاری، بهبود هماهنگی بین تیمها و ذینفعان، افزایش رضایت مشتری، و تسهیل در نگهداری و توسعه آینده نرم افزار اشاره کرد.

مدل های فرآیند نرم افزار

انواع مدل های فرآیند 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 فرآیند) نمایش داده می شود. به طور معمول با نمودارهایی مانند فلوچارت یا نمودار جریان کار نشان داده می شود که مراحل اجرایی فرآیند را به صورت گام به گام و موازی توصیف می کند.
- هدف مدل جریان کار، تسهیل مدیریت فرآیندها، زمان بندی مناسب و بهبود هماهنگی بین فعالیت ها است.
- مزایای مدل جریان کار عبارتند از: افزایش بهره وری با بهبود مدیریت وظایف، شفافیت فرآیندها 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 فرآیندهای 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 فرآیندهای کسب و کار.
- مثال: یک شرکت نرم افزاری در تهیه و تحويل 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 فرآیند باعث می شود پروژه ها به موقع، با کیفیت بهتر و با همکاری بهینه تیم ها تحویل داده شود و میزان خطاها و دوباره کاری ها کاهش یابد.

ادامه

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 داده محور 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 فرایندها، توسعه تدریجی مدل‌ها از دید کلی به جزئیات بیشتر، کاهش خطاهای 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 توسعه دهنده‌گان کمک می‌کند نقاط ضعف و بهبودهای لازم در سیستم را شناسایی کنند.

ادامه

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 فرآیندهای 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

11

- ۱- یک سناریوی واقعی یا فرضی از پروژه نرم افزاری (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 تحويل: ۲ مهرماه ۱۴۰۴

موفق باشد



دانشگاه شهرداری شهر کرمان

به نام دانای نیآموخته

درس: مهندسی نرم افزار
ترم: نیم سال اول [۱۴۰۵]
جلسه ۲

دپارتمان: مهندسی کامپیووتر

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

مدل توسعه نرم افزار	مزیت	عيوب	مثال
۱-رهیافت آبشاری	ساختار منظم و مرحله‌ای	تغییرات و اصلاحات پس از هر مرحله سخت و هزینه‌بر است	پروژه‌های با نیازهای ثابت و مشخص مثل پروژه‌های دولتی
۲-توسعه تکاملی	قابل انعطاف و پذیرش تغییرات	ممکن است زمان پروژه طولانی شود و مدیریت آن پیچیده شود	نرم افزارهای آزمایشی که به تدریج کامل می‌شوند
۳-تبدیل رسمی	تضمين صحت و دقت بالا	پیاده‌سازی پیچیده و زمان‌بر است	پروژه‌های نظامی و فضایی که نیاز به صحت بالا دارند
۴-مونتاژ سیستم با قطعات قابل استفاده مجدد	صرفه‌جویی در زمان و هزینه با استفاده مجدد قطعات	ممکن است قطعات سازگار به سختی پیدا شوند و کیفیت سیستم کاهش یابد	تولید سیستم‌هایی با استفاده از کامپوننت‌های نرم افزاری از قبل توسعه یافته مانند سیستم‌های اسمبل شده

ادامه

3

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 پژوههایی با نیازمندی‌های ثابت مناسب است.

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

پ) یک مدلی که در آن توسعه نرم افزار بر پایه روش‌های رسمی و ریاضی برای تضمین صحت و کیفیت نرم افزار انجام می شود. این روش بیشتر در پژوههای حساس و بحرانی مانند سیستم‌های 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، مستند سازی، و ایجاد نرم افزار از ابتدا انجام می شود. این فاز عمدتاً به نیروی انسانی متخصص مانند توسعه دهنده گان نرم افزار نیاز دارد و شامل هزینه هایی مانند حقوق توسعه دهنده گان، 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. هر چند 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 درصد ها 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 سیستم‌های نرم افزاری است که از فعالیت‌های مختلف فرآیند توسعه نرم افزار به صورت خودکار پشتیبانی می‌کنند. به بیانی دیگر، این سیستم‌ها ابزارهای کامپیوتراًی هستند که به مهندسان نرم افزار کمک می‌کنند تا مراحل مختلف چرخه عمر نرم افزار مثل تحلیل، طراحی، برنامه‌نویسی، آزمون، تست و نگهداری نرم افزار را به شکل خودکار یا نیمه خودکار انجام دهند.
- هدف از استفاده از کیس‌ها همان افزایش کیفیت، کاهش خطأ و افزایش سرعت توسعه نرم افزار است.
- ابزارهای CASE، مانند نرم افزارهای طراحی، مدیریت پروژه، مدل‌سازی، اشکال‌زدایی و آزمون نرم افزار هستند که در فرآیند توسعه نرم افزار به صورت ابزار کمکی به کار می‌روند و به تیم توسعه کمک می‌کنند تا بتوانند بهتر و سریع‌تر 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 خطاهای 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 دقیقیت 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 نرمافزار اشاره دارد. نرمافزاری که کارآمد است، هزینه‌های سختافزاری و انرژی را کاهش می‌دهد. مثال: نرمافزارهای مدیریت داده در سرورهای بانکی که باید با کمترین مصرف منابع بیشترین پردازش را انجام دهند، نمونه‌های بارز نرمافزارهای کارآمد هستند.

ت). قابلیت استفاده (Usability) قابلیت استفاده به سهولت استفاده کاربران از نرمافزار اشاره دارد. نرمافزاری که کاربرپسند باشد، نیاز به آموزش کمتر و حمایت فنی کمتری دارد که باعث کاهش هزینه‌های ضمنی می‌شود. مثال: نرمافزارهای سازمانی با رابط کاربری ساده که کاربران بدون نیاز به آموزش طولانی قادر به استفاده از آن باشند، نمونه‌ای از نرمافزارهایی با قابلیت استفاده بالا هستند.

وظایف تخصصی و اخلاقی

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

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

الف) محترمانگی (Confidentiality) حفظ و نگهداری اطلاعات خصوصی و حساس کاربران و سازمان‌ها از اهمیت بسیار زیادی برخوردار است. متخصصان باید از افشای اطلاعات بدون اجازه جلوگیری کنند و از داده‌ها 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 می‌تواند به خسارات‌های جبران‌ناپذیری منجر شود.

ادامه

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 وسائل 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

10

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

مهلت تحولی: ۱۱ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیآموخته

درس: مهندسی نرم افزار
ترم: نیم سال اول (۱۴۰۵)

جلسه ۳

دپارتعان: مهندسی کامپیووتر
مدرس: دکتر شیما شفیعی

مدل های توسعه نرم افزاری با تاکید بر معماری

1

جدول ۱. مقایسه مدل های توسعه نرم افزار

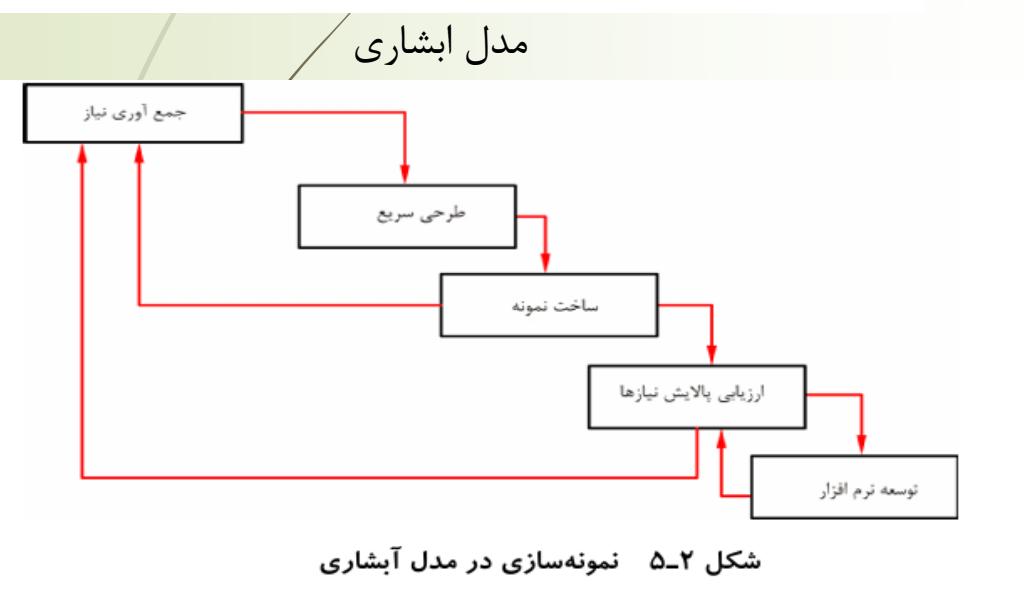
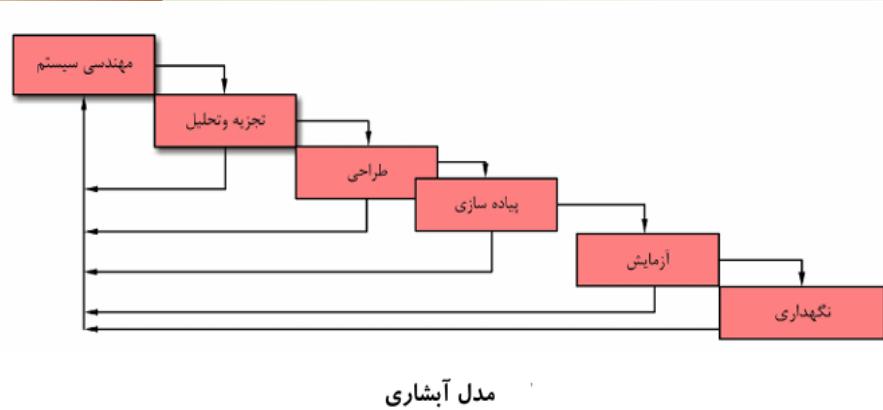
ویژگی	Linear Sequential Model (مدل ترتیبی خطی)	Waterfall Model (مدل آبشاری)	Prototyping Model (مدل نمونه‌سازی)
نوع مدل	ترتیبی، مرحله‌ای و خطی	ترتیبی، مرحله‌ای	تکراری، تعاملی و تکاملی
مراحل	جمع‌آوری نیازها، طراحی (منطقی و فیزیکی)، اجرا، تست، نگهداری	ساخت نمونه اولیه، بازخورد، بهبود و تکرار نمونه	
مزایا	سادگی، شفافیت، مستندسازی دقیق، مراحل مشخص	مستندات کامل، ثبات در محدوده پروژه، قابل پیش‌بینی بودن	شناسایی زودهنگام نیازها، کاهش ریسک، تعامل مشتری
معایب	انعطاف‌پذیری محدود، دشوار در تغییر نیازها	انعطاف‌پذیری محدود، دشوار در اصلاحات بعد از فازها	هزینه و زمان بیشتر، پیچیدگی مدیریت نمونه‌ها
کارایی	پروژه‌های با نیازمندی‌های کاملاً مشخص و ثابت	پروژه‌های با نیازهای واضح و ثابت	پروژه‌های نیازمند بازخورد مستمر و در حال تغییر
مثال	پروژه توسعه نرم‌افزاری نظام بانکی با الزامات دقیق	سیستم حسابداری با نیازهای مشخص	طراحی اپلیکیشن موبایل با نیاز به رابط کاربری پویا

2

جدول ۱. مقایسه مدل های توسعه نرم افزار

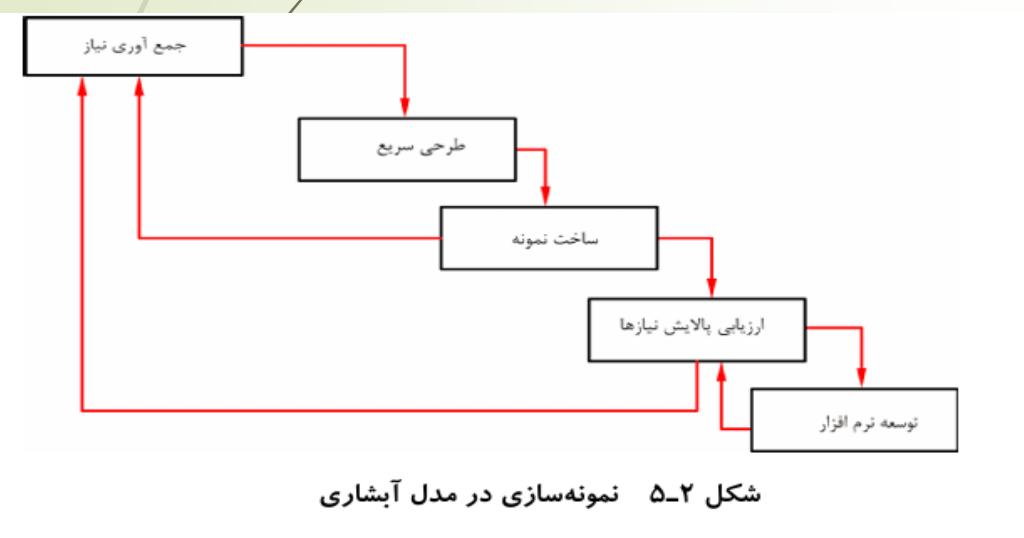
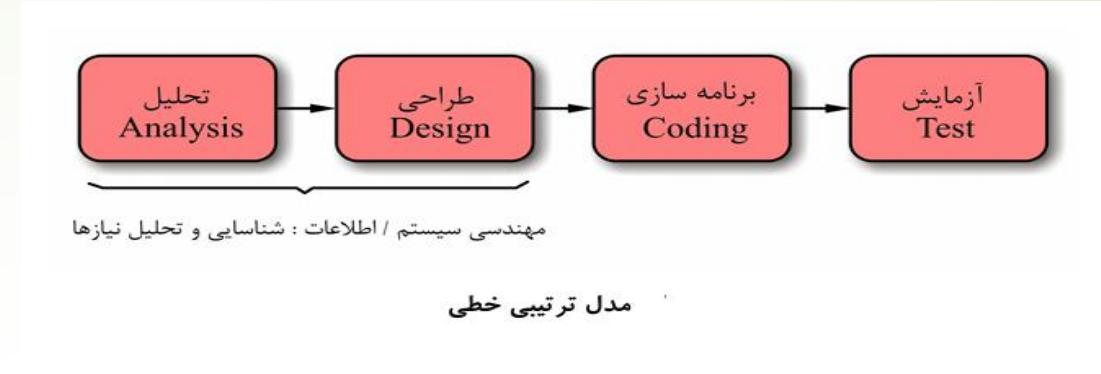
ویژگی	RAD Model (مدل توسعه سریع)
نوع مدل	تکراری، موازی، مبتنی بر بخش بندی پروژه
مراحل	تقسیم پروژه به بخش های کوچک، توسعه موازی، ادغام بخش ها
مزایا	سرعت بالا در توسعه، انعطاف‌پذیری، کاهش ریسک، مشارکت بیشتر کاربر
معایب	نیاز به تیم های حرفه ای، مناسب نبودن برای پروژه های بزرگ و پیچیده
کارایی	پروژه های کوچک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

ادامه



مدل نمونه سازی

شکل ۱. انواع مدل های توسعه نرم افزار

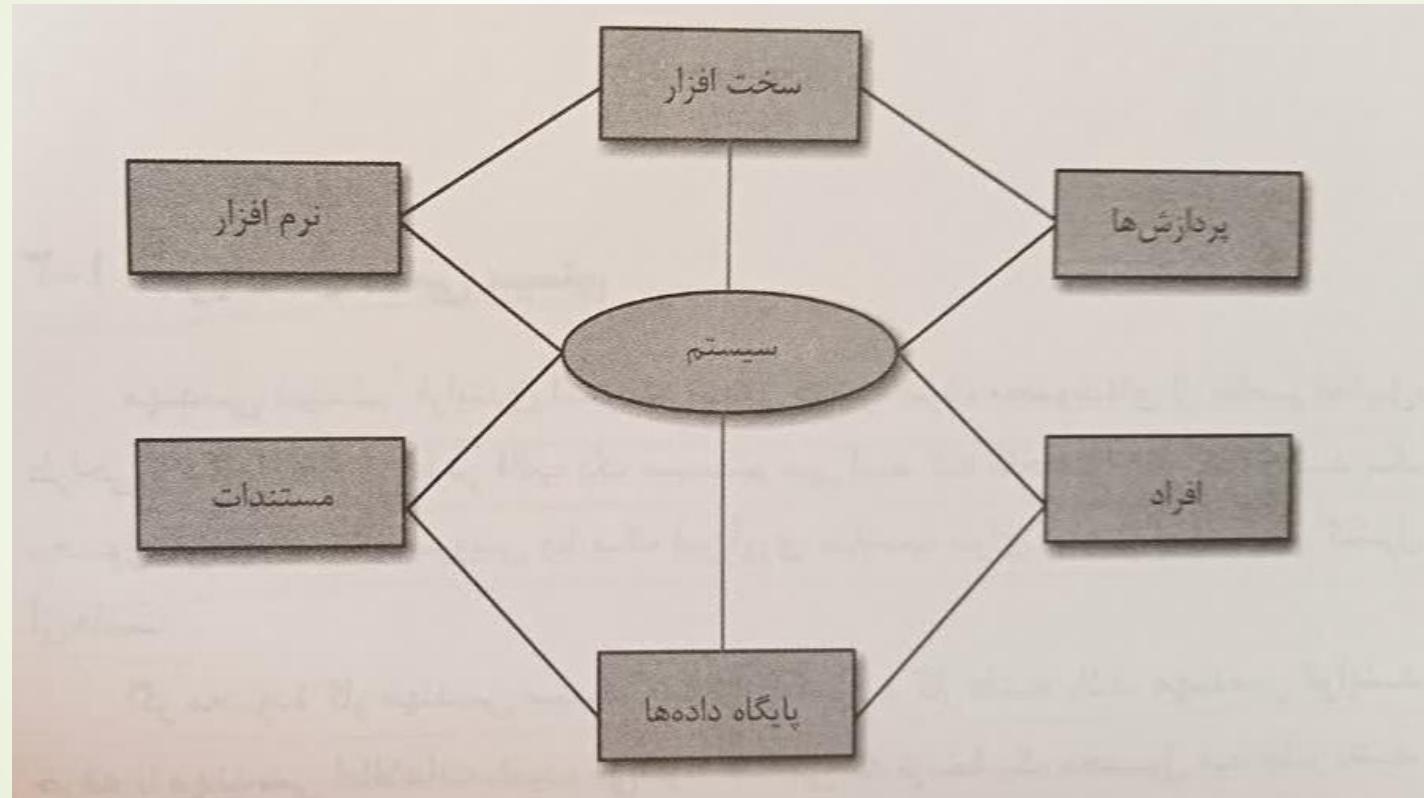


مدل توسعه سریع کاربرد

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 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 می‌کنند تا یک یا چند هدف مشخص را تحقق بخشنند. این اجزاء می‌توانند فیزیکی، منطقی، مفهومی یا ترکیبی از این‌ها باشند.
- **ارتباط بین اجزاء**: هر جزء سیستم با سایر اجزاء در ارتباط است و عملکرد هر بخش می‌تواند روی بخش‌های دیگر تأثیر بگذارد. ارتباطات باعث هماهنگی و هم‌افزایی می‌شوند.
 - **هدف مشترک**: همه اجزاء به منظور تحقق یک هدف کلی در کنار هم کار می‌کنند؛ بدون هدف، مجموعه‌ای از اجزاء صرفاً یک دسته‌پراکنده است، نه سیستم.
 - **مرز مشخص**: سیستم دارای یک مرز یا محدوده تعریف شده است که تعیین می‌کند چه چیزی داخل سیستم است و چه چیزی خارج از آن (محیط).
 - **ورودی و خروجی**: سیستم معمولاً ورودی‌هایی از محیط دریافت می‌کند، آن‌ها را پردازش می‌کند و خروجی‌هایی به محیط می‌دهد. ورودی و خروجی ممکن است داده، انرژی، مواد یا اطلاعات باشند.
 - **بازخورد**: بسیاری از سیستم‌ها دارای مکانیسم‌های بازخورد هستند تا عملکرد خود را ارزیابی و در صورت نیاز اصلاح کنند، به منظور بهبود یا حفظ کارایی.
 - **پویا بودن**: سیستم‌ها معمولاً پویا هستند و در طول زمان تغییر می‌کنند و خود را با شرایط محیطی سازگار می‌کنند.

سیستم های مبتنی بر نرم افزار

- الف) سیستم های کامپیوتری تکنیکی
- ب) سیستم های اجتماعی - تکنیکی

جدول ۳. مقایسه سیستم ها

ویژگی	سیستم های کامپیوتری تکنیکی	سیستم های اجتماعی-تکنیکی
تعریف	سیستم هایی که شامل سخت افزار، نرم افزار و اجزای فنی هستند.	سیستم هایی که شامل اجزای فنی و انسانی به صورت همزمان می باشند
اجزاء اصلی	سخت افزار، نرم افزار، شبکه، پایگاه داده	انسان ها (کاربران، مدیران)، فرآیندها، فناوری
تمرکز اصلی	بهینه سازی و عملکرد جویی پذیری داده، اتوماسیون	تعامل بین انسان و فناوری، بهبود کارایی اجتماعی و فنی
هدف	افزایش کارایی، دقت و سرعت در پردازش اطلاعات	افزایش بهره وری، رضایت کاربران و هماهنگی بین انسان و فناوری
ماهیت سیستم	تکنیکی و ماشینی	ترکیبی از تکنولوژی و رفتار انسانی
مثال ها	سیستم های مدیریت پایگاه داده، سیستم های عامل، شبکه های کامپیوتری	سیستم های سازمانی، تیم های کاری، سیستم های خدمات مشتری
چالش ها	مشکلات سخت افزاری، نرم افزاری، خطاهای فنی	مسائل انسانی مثل مقاومت در برابر تغییر، فرهنگ سازمانی
طراحی و پیاده سازی	بیشتر مت مرکز بر مهندسی نرم افزار و سخت افزار	نیازمند توجه به عوامل انسانی، روانشناسی و مهندسی سیستم ها

تمرین ۳

۹

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

مهلت تحويل: ۱۱ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیاموخته

درس: مهندسی نرم افزار
ترم: نیم سال اول (۱۴۰۵)

جلسه ۳

دپارتمان: مهندسی کامپیوتر
مدرس: دکتر شیما شفیعی

لایه های مهندسی نرم افزار

- در فرآیند مهندسی نرم افزار، فعالیت‌ها در چند لایه مفهومی تقسیم می‌شوند. این لایه‌ها کمک کرده تا نرم افزار با کیفیت، قابل نگهداری و قابل توسعه ساخته بشوند.

1. Quality Focus (تمرکز بر کیفیت)

- مرکزی‌ترین 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 مدل‌های فرآیند مانند: Waterfall

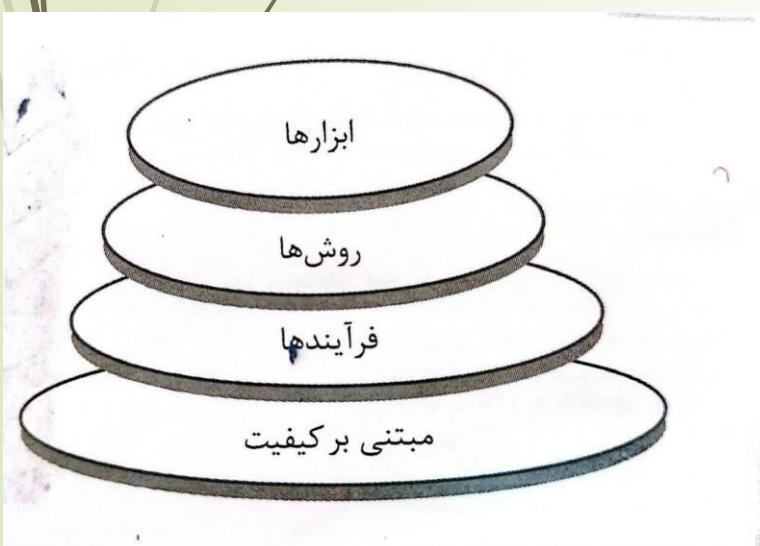
3

ادامه

3. Methods (روش‌ها)

روش‌های فنی برای تحلیل، طراحی، پیاده‌سازی و تست نرم‌افزار و چرخه توسعه.
شامل: مدل‌سازی، الگوریتم‌ها، طراحی معماری، طراحی واسط، وغیره.

4. Tools (ابزارها)



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

انواع مهندسی

4

جدول ۳. مقایسه مهندسی ها

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

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 روش‌های ریاضیاتی، گرافیکی یا کامپیوتری نمایش داده می‌شود تا رفتار، ویژگی‌ها و روابط داخلی آن بهتر درک، تحلیل و پیش‌بینی شود.
- این مدل‌ها می‌توانند بر پایه قوانین فیزیکی یا داده‌های تجربی ساخته شوند و به طراحان یا محققان کمک می‌کنند تا بدون نیاز به آزمایش‌های واقعی و پرهزینه، کارایی سیستم‌ها را بهینه کرده و ریسک‌ها را کاهش دهند.
- مدل سازی سیستم شامل ایجاد مدل‌های ریاضی، گرافیکی، مفهومی، ذهنی یا فیزیکی است که از آنها برای تحلیل و بهبود سیستم‌ها در زمینه‌های مختلف کاربردی مانند مهندسی، علوم پزشکی، فناوری اطلاعات و علوم اجتماعی استفاده می‌شود.
- این فرآیند اولین گام در طراحی کنترل‌کننده‌ها و مدیریت سیستم‌های دینامیکی نیز به شمار می‌آید.

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 زیرسamanه‌های 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

8

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

مهلت تحويل: ۱۸ مهرماه ۱۴۰۴

موفق باشید



به نام دانای نیاموخته

درس: مهندسی نرم افزار

ترم: نیم سال اول (۱۴۰۵-۰۵)

جلسه ۸

دپارتمان: مهندسی کامپیووتر

مدرس: دکتر شیما شفیعی

مهندسی نیازها

- تعريف 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.
- تعريف 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

3

- 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

4

- 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 پرتو زده 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 خطأ و سوءتفاهم 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

6

- مدل سازی سیستم یکی از مراحل مهندسی نیازها است که در آن، با استفاده از نمودارها، مدل‌ها و زبان‌های طراحی، ساختار، رفتار و عملکرد کلی سیستم به شکل انتزاعی و قابل درک ارائه می‌شود. این مدل‌ها کمک می‌کنند تا نیازهای مختلف سیستم بهتر فهمیده، تحلیل و ارتباط بین آن‌ها روشن شود.
- اهداف مدل سازی سیستم عبارتند از:
 - تسهیل فهم دقیق نیازمندی‌ها و ساختار سیستم توسط همه ذینفعان.
 - برنامه‌ریزی صحیح برای طراحی و پیاده‌سازی سیستم.
 - ارتقای ارتباط و تعامل میان اعضای تیم توسعه و مشتری.
- فرض کنید قرار است سیستم مدیریت فروشگاهی طراحی شود؛ در مدل‌سازی سیستم، ابتدا دیاگرام‌های جریان داده (DFD) (ترسیم می‌شود که نشان می‌دهد چگونه اطلاعات سفارش مشتریان از ورود به سیستم، پردازش، ذخیره و نهایتاً به انبار و بخش حسابداری منتقل می‌شود. این مدل به ذینفعان اجازه می‌دهد جریان داده‌ها و وظایف سیستم را پیش از کدنویسی بفهمند و تضمین می‌کند نیازها به درستی منعکس شده‌اند.

مراحل مهندسی نیازها: اعتبارسنجی نیازها

7

- 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 تائید 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 رصد⁸ 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 نیازمندها 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 خطأ و اشتباہ 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

۹

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

مهلت تحويل: ۱۸ مهرماه ۱۴۰۴

موفق باشید