

دانشگاه شاهرود
پایه

تجزیه و تحلیل سیستم‌ها

دکتر مسرت آیت سیده لیلا عمران‌ی خراسانی

امروزه کتاب‌خوانی و علم‌آموزی، نه تنها یک وظیفه ملی، که یک واجب دینی است.^۱

مقام معظم رهبری

در عصر حاضر یکی از شاخصه‌های ارزیابی رشد، توسعه و پیشرفت فرهنگی هر کشوری میزان تولید کتاب، مطالعه و کتاب‌خوانی مردم آن مرز و بوم است. ایران اسلامی نیز از دیرباز تاکنون با داشتن تمدنی چندهزارساله و مراکز متعدد علمی، فرهنگی، کتابخانه‌های معتبر، علما و دانشمندان بزرگ با آثار ارزشمند تاریخی، سرآمد دولت‌ها و ملت‌های دیگر بوده و در عرصه فرهنگ و تمدن جهانی به‌سان خورشیدی تابناک همچنان می‌درخشد و با فرزندان نیک‌نهاد خویش هنرنمایی می‌کند. چه کسی است که در دنیا با دانشمندان فرزانه و نام‌آور ایرانی همچون ابوعلی سینا، ابوریحان بیرونی، فارابی، خوارزمی و ... همچنین شاعران برجسته‌ای نظیر فردوسی، سعدی، مولوی، حافظ و ... آشنا نباشد و در مقابل عظمت آنها سر تعظیم فرود نیاورد. تمامی این افتخارات ارزشمند، برگرفته از میزان عشق و علاقه فراوان ملت ما به فراگیری علم و دانش از طریق خواندن و مطالعه منابع و کتاب‌های گوناگون است. به شکرانه الهی، تاریخ و گذشته ما، همیشه درخشان و پر بار است. ولی اکنون در این زمینه در چه جایگاهی قرار داریم؟ آمار و ارقام ارائه‌شده از سوی مجامع و سازمان‌های فرهنگی در مورد سرانه مطالعه هر ایرانی، برایمان چندان امیدوارکننده نمی‌باشد.

کتاب، دروازه‌ای به سوی گستره دانش و معرفت است و کتاب خوب، یکی از بهترین ابزارهای کمال بشری است. همه دستاوردهای بشر در سراسر عمر جهان، تا آنجا که قابل کتابت بوده است، در میان دست‌نوشته‌هایی است که انسان‌ها پدید آورده و می‌آورند. در این مجموعه بی‌نظیر، تعالیم الهی، درس‌های پیامبران به بشر، و همچنین علوم مختلفی است که سعادت بشر بدون آگاهی از آنها امکان‌پذیر نیست. کسی که با دنیای زیبا و زندگی‌بخش کتاب ارتباط ندارد بی‌شک از مهم‌ترین دستاورد انسانی و نیز از بیشترین معارف الهی و بشری محروم است. با این دیدگاه، به‌روشنی می‌توان ارزش و مفهوم رمزی عمیق در این حقیقت تاریخی را دریافت که اولین خطاب خداوند متعال به پیامبر گرامی اسلام (ص) این است که «بخوان!» و در اولین

۱. پیام مقام معظم رهبری به مناسبت آغاز هفته کتاب ۷۲/۱۰/۴

سوره‌ای که بر آن فرستاده عظیم‌الشان خداوند، فرود آمده، نام «قلم» به تجلیل یاد شده‌است: «إِقْرَأْ وَ رَبُّكَ الْأَكْرَمُ. الَّذِي عَلَّمَ بِالْقَلَمِ» در اهمیت عنصر کتاب برای تکامل جامعه انسانی، همین بس که تمامی ادیان آسمانی و رجال بزرگ تاریخ بشری، از طریق کتاب جاودانه مانده‌اند.

دانشگاه پیام‌نور با گستره جغرافیایی ایران‌شمول خود با هدف آموزش برای همه، همه‌جا و همه‌وقت، به‌عنوان دانشگاهی کتاب‌محور در نظام آموزش عالی کشورمان، افتخار دارد جایگاه اندیشه‌سازی و خردورزی بخش عظیمی از جوانان جویای علم این مرز و بوم باشد. تلاش فراوانی در ایام طولانی فعالیت این دانشگاه انجام پذیرفته تا با بهره‌گیری از تجربه‌های گرانقدر استادان و صاحب‌نظران برجسته کشورمان، کتاب‌ها و منابع آموزشی درسی شاخص و خودآموز تولید شود. در آینده هم، این مهم با هدف ارتقای سطح علمی، روزآمدی و توجه بیشتر به نیازهای مخاطبان دانشگاه پیام‌نور با جدیت ادامه خواهد داشت. به‌طور قطع استفاده از نظرات استادان، صاحب‌نظران و دانشجویان محترم، ما را در انجام این وظیفه مهم و خطیر یاری‌رسان خواهد بود. پیشاپیش از تمامی عزیزانی که با نقد، تصحیح و پیشنهادهای خود ما را در انجام این وظیفه خطیر یاری می‌رسانند، سپاسگزاری می‌نماییم. لازم است از تمامی اندیشمندانی که تاکنون دانشگاه پیام‌نور را منزلگاه اندیشه‌سازی خود دانسته و ما را در تولید کتاب و محتوای آموزشی درسی یاری نموده‌اند، صمیمانه قدردانی گردد. موفقیت و بهروزی تمامی دانشجویان و دانش‌پژوهان عزیز آرزوی همیشگی ما است.

دانشگاه پیام‌نور

فهرست مطالب

یازده

پیشگفتار

۱	فصل اول: مفاهیم پایه
۱	هدف کلی
۱	هدف‌های یادگیری
۱	مقدمه
۳	۱-۱ تاریخچه مهندسی سیستم
۴	۲-۱ مهندس سیستم
۷	۳-۱ فرایندهای مهندسی سیستم
۹	۴-۱ مدیریت ریسک
۱۰	۵-۱ وظایف مهندس سیستم
۱۲	۶-۱ مقایسه مهندسی سیستم با مهندسی نرم‌افزار
۱۳	۷-۱ زبان مدل‌سازی همگون
۱۴	۸-۱ فازهای تولید نرم‌افزار
۱۵	۹-۱ پایداری در سیستم تولید نرم‌افزار
۱۶	۱۰-۱ چالش‌ها و راهکارهای تولید نرم‌افزار
۱۷	۱۱-۱ روش‌های شناخت نیازمندی‌های سیستم
۱۸	۱-۱۱-۱ روش مصاحبه
۱۸	۲-۱۱-۱ روش توفان فکری
۱۸	۳-۱۱-۱ روش تحلیل مستندات
۱۹	۴-۱۱-۱ روش پرسش‌نامه
۱۹	۵-۱۱-۱ روش کارگاه نیازمندی‌ها
۱۹	۱۲-۱ روش‌های تحلیل (آنالیز) نرم‌افزاری

۲۱	۱-۱۳ مهندسی نرم افزار
۲۲	خلاصه فصل اول
۲۲	خودآزمایی چهارگزینه‌ای فصل اول
۲۴	خودآزمایی تشریحی فصل اول
۲۵	فصل دوم: مدل‌های توسعه نرم افزار
۲۵	هدف کلی
۲۵	هدف‌های یادگیری
۲۵	مقدمه
۲۶	۱-۲ چرخه توسعه نرم افزار
۳۰	۲-۲ متدولوژی‌های تولید نرم افزار
۳۳	۲-۲-۱ متدولوژی‌های طراحی ساخت یافته
۳۸	۲-۲-۲ متدولوژی‌های توسعه سریع
۴۷	۲-۲-۳ متدولوژی‌های شیء گرا (Object Oriented Development)
۵۰	۲-۲-۴ متدولوژی‌های چابک (Agile Development)
۵۹	۳-۲ انتخاب متدولوژی مناسب برای انجام پروژه‌های نرم‌افزاری
۶۲	۲-۴ نقش‌ها و مهارت‌های تیم پروژه
۶۶	خلاصه فصل دوم
۶۷	خودآزمایی چهارگزینه‌ای فصل دوم
۶۸	خودآزمایی تشریحی فصل دوم
۶۹	فصل سوم: تحلیل نیازمندی‌ها
۶۹	هدف کلی
۶۹	هدف‌های یادگیری
۶۹	مقدمه
۷۱	۳-۱ فهم مسئله
۷۴	۳-۱-۱ مدل سازی در UML
۷۶	۳-۱-۲ نمودارهای ساختاری
۸۳	۳-۱-۳ نمودارهای رفتاری
۹۰	۳-۲ تحلیل امکان سنجی
۹۰	۳-۲-۱ امکان سنجی فنی
۹۲	۳-۲-۲ امکان سنجی مالی
۹۳	۳-۲-۳ امکان سنجی زمانی
۹۴	۳-۲-۴ امکان سنجی سازمانی
۹۶	۳-۳ تحلیل شیء گرا
۹۸	۳-۳-۱ شیء و دیدگاه شیء گرا

۹۹	۳-۲ مفاهیم و اصول شیء‌گرایی
۱۰۴	۳-۴ تحلیل سیستم شیء‌گرا
۱۰۷	۳-۵ تحلیل عناصر در سیستم شیء‌گرا
۱۰۸	۳-۵-۱ شناسایی کلاس‌ها و اشیا
۱۱۱	۳-۵-۲ مسئولیت‌ها
۱۱۴	۳-۵-۳ همکاری‌ها
۱۱۶	۳-۵-۴ نهایی کردن تعریف هر شیء
۱۱۷	۳-۵-۵ تعیین ساختارها و سلسله‌مراتب
۱۱۹	۳-۵-۶ ترسیم مدل ارتباط بین اشیا
۱۲۲	۳-۵-۷ مدل رفتار شیء
۱۲۴	خلاصه فصل سوم
۱۲۵	خودآزمایی چهارگزینه‌ای فصل سوم
۱۲۷	خودآزمایی تشریحی فصل سوم
۱۲۹	فصل چهارم: طراحی سیستم
۱۲۹	هدف کلی
۱۲۹	هدف‌های یادگیری
۱۲۹	مقدمه
۱۳۰	۴-۱ رویکرد طراحی سیستم
۱۳۱	۴-۱-۱ رویکرد مشتق شده از مدل
۱۳۱	۴-۱-۲ طراحی ساخت یافته مدرن
۱۳۲	۴-۲ مهندسی اطلاعات
۱۳۲	۴-۳ نمونه اولیه
۱۳۵	۴-۴ طراحی شیء‌گرا (OOD)
۱۳۵	۴-۵ توسعه کاربردی سریع
۱۳۶	۴-۶ استراتژی‌های طراحی سیستم سریع (FAST)
۱۳۷	۴-۷ طراحی سیستم برای ساخت نرم‌افزار
۱۴۲	۴-۸ مدل‌سازی و معماری برنامه‌های کاربردی
۱۴۳	۴-۸-۱ دیاگرام یا نمودار جریان داده فیزیکی
۱۴۴	۴-۸-۲ معماری داده - پایگاه داده‌های رابطه‌ای توزیع شده
۱۴۶	۴-۸-۳ معماری واسط کاربر، ورودی‌ها و خروجی‌ها و میان‌افزار
۱۴۶	۴-۸-۴ معماری فرایند جهت توسعه نرم‌افزار
۱۴۷	۴-۸-۵ استراتژی‌های معماری برنامه کاربردی برای طراحی سیستم
۱۴۸	۴-۸-۶ استراتژی معماری برنامه کاربردی با تدبیر
۱۴۸	۴-۸-۷ مدل‌سازی معماری برنامه کاربردی از یک سیستم اطلاعاتی
۱۴۸	۴-۸-۸ ترسیم نمودار جریان داده فیزیکی

۱۴۹	۹-۴ طراحی شیء گرا و مدل سازی با استفاده از UML
۱۵۰	۱-۹-۴ طراحی یک سیستم شیء گرا
۱۵۲	۲-۹-۴ طراحی روابط
۱۵۳	۱۰-۴ ویژگی و مدل های قابل رؤیت
۱۵۴	۱-۱۰-۴ مسئولیت پذیری اشیا
۱۵۵	۱۱-۴ فرایند طراحی شیء گرا
۱۵۶	۱-۱۱-۴ پالایش مدل موارد استفاده
۱۵۷	۱۲-۴ مدل سازی
۱۶۸	۱۳-۴ قابلیت استفاده مجدد از اشیا و طراحی الگو
۱۶۹	۱۴-۴ الگوهای طراحی
۱۷۰	۱-۱۴-۴ چارچوب های اشیا و مؤلفه ها
۱۷۱	۱۵-۴ نمودارهای پیاده سازی و طراحی UML اضافی
۱۷۳	خلاصه فصل چهارم
۱۷۴	خودآزمایی چهارگزینه ای فصل چهارم
۱۷۷	خودآزمایی تشریحی فصل چهارم
۱۷۹	فصل پنجم: ساخت، تست و پیاده سازی سیستم
۱۷۹	هدف کلی
۱۷۹	هدف های یادگیری
۱۷۹	مقدمه
۱۸۰	۱-۵ مفهوم ساختن و پیاده سازی یک سیستم
۱۸۰	۱-۱-۵ فاز ساختن
۱۸۵	۲-۱-۵ فاز پیاده سازی
۱۹۱	خلاصه فصل پنجم
۱۹۱	خودآزمایی چهارگزینه ای فصل پنجم
۱۹۲	خودآزمایی تشریحی فصل پنجم
۱۹۳	فصل ششم: مستند سازی و پشتیبانی سیستم
۱۹۳	هدف کلی
۱۹۳	هدف های یادگیری
۱۹۳	مقدمه
۱۹۴	۱-۶ اهمیت مستند سازی
۱۹۵	۲-۶ انواع مستندات نرم افزاری
۱۹۶	۳-۶ چگونگی مستند سازی
۲۰۰	۴-۶ نوع مستند
۲۰۱	۵-۶ پشتیبانی نرم افزار

۲۰۲	۱-۵-۶ روش های پشتیبانی نرم افزار
۲۰۳	۲-۵-۶ کارشناس پشتیبانی نرم افزار
۲۰۴	خلاصه فصل ششم
۲۰۴	خودآزمایی چهارگزینه ای فصل ششم
۲۰۵	خودآزمایی تشریحی فصل ششم
۲۰۶	پیوست
۲۱۱	پاسخنامه
۲۱۳	منابع

پیشگفتار

امروزه کشور ایران علی‌رغم داشتن تعداد مهندسين نرم‌افزار بسیار بالا نسبت به ساير کشورهاى دنيا، سهم بسیار ناچيزى در عوايد حاصل از توسعه، فروش و پشتیبانى از نرم‌افزارها دارد. اين غفلت موجب شده است که از یک سو فارغ‌التحصیلان این رشته به‌طور عمده بی‌کار بوده و از سوى ديگر فرصت‌هاى توسعه نرم‌افزارى در داخل و خارج کشور را از دست بدهند. عوامل متعدد و مهمى در پس چالش‌هاى پیش‌آمده در این زمینه وجود دارد که هر کدام نیازمند تحليل، ریشه‌یابى و ارائه راه‌حل‌هاى بنیادى و کاربردى هستند تا موجب جذب نخبگان این رشته در ساختارهاى کسب‌وکار داخلى شده و بیش از این بازارها و فرصت‌هاى داخلى و خارجى از دست نرود. يکى از این چالش‌ها عدم شناخت کاربردى فارغ‌التحصیلان از روش‌هاى اصولى توسعه نرم‌افزارها است. مراجع دانشگاهى موجود ابتدا بیشتر برپايه مباحث قدیمى در حوزه توسعه نرم‌افزار هستند که گاهى حتى روش‌هاى توسعه جدید را نیز منعکس نمى‌کنند و دوم اینکه از مثال‌هاى کاربردى مورد نیاز برای دانشجو جهت درک چگونگى توسعه یک نرم‌افزار در مقیاس متوسط به بالا رنج مى‌برند.

کتاب حاضر با درک این خلأها و نیازهاى موجود در این حوزه توسط مؤلفان به رشته تحرير درآمده است. محتوای کتاب با تکیه بر مباحث جدیدتر حوزه مهندسى نرم‌افزار و با در نظر گرفتن سادگى و روانى مطالب برای درک بهتر دانشجویان، به‌خصوص برای علاقمندانى که مایل به توسعه علمى نرم‌افزارها برای عرضه به بازار و مشتریان هستند، نوشته شده است.

در فصل اول به طور ساده مفاهیم پایه مربوطه از تعریف سیستم و مهندسی گرفته تا مهندسی نرم افزار تشریح شده‌اند.

در فصل دوم فازها و مدل‌های توسعه نرم افزار مورد توجه قرار گرفته و از هر نسل به چند مورد اشاره شده است. با این حال با تمرکز بر روی جدیدترین متدولوژی‌های توسعه نرم افزار مهم‌ترین متدولوژی‌های مطرح که در بازار کار نیز به شدت مورد علاقه و استفاده هستند مورد بررسی و تشریح قرار گرفته‌اند.

در فصل سوم به تشریح فازهای برنامه‌ریزی و تحلیل سیستم پرداخته شده است. همچنین برای تفهیم بهتر مباحث مربوط به شیء‌گرایی که در فصل‌های بعدی مورد توجه زیادی قرار گرفته‌اند، زبان مدل‌سازی یکپارچه (UML) بیان شده است.

در فصل چهارم فاز طراحی سیستم نرم افزاری بیان شده است و طراحی یک سیستم نرم افزاری به همراه جزئیات طراحی بخش‌های مختلف آن مورد بررسی قرار گرفته است. همچنین طراحی شیء‌گرا به شکل کامل مطرح شده است.

در فصل پنجم فاز نحوه پیاده‌سازی یک سیستم نرم افزاری بیان شده است. طی این فصل ضمن بیان روش‌های پیاده‌سازی یک نرم افزار، انواع آزمون (تست‌های نرم افزار نیز تشریح شده‌اند.

در فصل ششم به مستندسازی نرم افزار پرداخته شده و اهمیت مستندسازی یک نرم افزار و چگونگی انجام آن بیان شده است. همچنین چگونگی تحویل و پشتیبانی نرم افزار آماده با ارائه راهکارهایی تشریح شده است.

در پایان هر فصل برای تست و آزمون بیشتر و سنجش میزان فراگیری مطالب سؤالات چهارگزینه‌ای و سؤالات تشریحی مربوط به آن فصل به صورت مجزا بیان شده است و تا دانشجویان ضمن مرور مطالب اصلی هر فصل کیفیت یادگیری خود را به همان شیوه که در آزمون‌های پایان‌ترم دانشگاه پیام‌نور انجام می‌شود، ارزیابی کنند. در انتهای کتاب نیز پاسخ سؤالات چهارگزینه‌ای آورده شده است.

در خاتمه از زحمات کلیه افرادی که به هر شکل ما را در تدوین این اثر یاری کردند تشکر می‌کنیم. همچنین از همکاران و دانشجویان گرامی تقاضا می‌شود در صورت مشاهده هرگونه لغزش چاپی و غیرچاپی ما را در تصحیح چاپ‌های بعدی یاری نمایند.

دکتر مسرت آیت

سیده لیلا عمرانی خراسانی

فصل اول

مفاهیم پایه

هدف کلی

در این با مفاهیم اولیه سیستم و مهندسی نرم افزار آشنا می شوید.

هدف های یادگیری

پس از مطالعه این فصل از شما انتظار می رود:

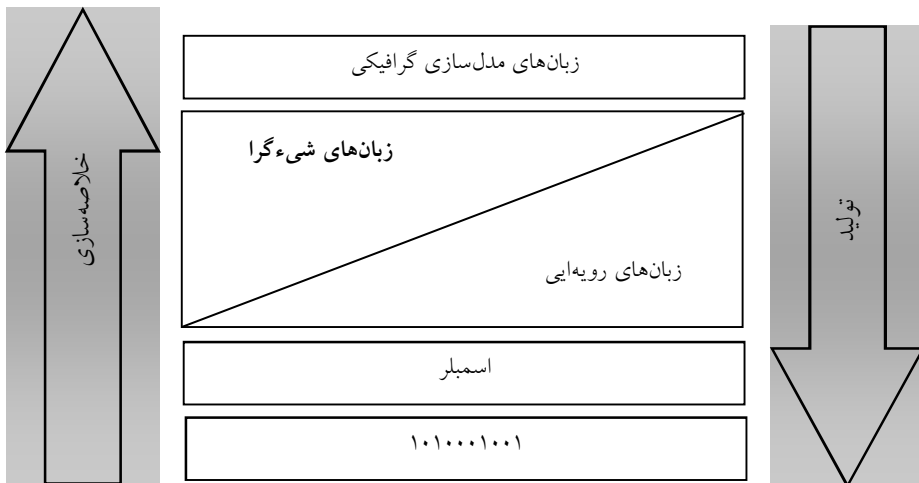
۱. مهندسی سیستم را تعریف و تحلیل کنید.
۲. مدیریت ریسک توضیح دهید.
۳. مفاهیم اولیه مهندسی نرم افزار را شرح دهید.
۴. فازهای تولید نرم افزار را بیان کنید.
۵. چالش های تولید نرم افزار را توضیح دهید.
۶. چگونگی تحلیل نرم افزار را بیان کنید.

مقدمه

امروزه روش ها و ابزارها برای انجام یک پروژه مهندسی پیچیده تر و دارای ابعاد گسترده تری شده اند. برای اجرای یک پروژه نیاز به افراد زیادی با مهارت بالا است؛ با این وجود حتی نمی توان با استفاده از تمامی این افراد متخصص و در نظر گرفتن تمام جنبه های کار تولید کامل یک سیستم کاملاً مطلوب را تضمین کرد. در حال حاضر به جایی رسیده ایم که نیاز به

سیستم‌های پیچیده و توزیع شده بیشتر احساس می‌شود. اما روش‌های توسعه یافته مطلوب امروزی هنوز در آن اندازه نیستند که بتوانند این نوع سیستم‌ها را در زمان کوتاه‌تر، قیمت مناسب‌تر و کاملاً منطبق بر نیازمندی‌های متصور ارائه دهند. طبیعی است که نمی‌توان انتظار داشت که یک سیستم پیشرفته و توسعه یافته را با استفاده از همان ابزارها و روش‌های قدیمی ایجاد کرد. بنابراین استفاده از زبان‌های مدل‌سازی و محیط‌های پیشرفته جدید بخشی از این توسعه و پیشرفت محسوب می‌شوند.

سیر تکاملی و سمت و سوی افزایش پیچیدگی توسعه نرم‌افزار نکات واضحی را در ذهن مخاطب ایجاد می‌سازد. ابزارها به سمت مؤلفه‌های بزرگ‌تر پیش رفته‌اند. برای مثال استفاده از بیت‌های ۰ و ۱ تا استفاده از کلاس و شی. برنامه‌ها ابتدا با کارت‌های سوراخ‌دار نوشته می‌شدند پس از آن استفاده از اسمبلر و زبان‌های برنامه‌نویسی رویه‌ای رواج یافت و در انتها زبان‌های شیء‌گرا برای ایجاد سیستم‌های پیچیده‌تر مورد استفاده قرار گرفتند. تکامل نسل بعد با استفاده از زبان‌های مدل‌سازی همگون (UML) برای تولید سیستم‌های نرم‌افزاری توسعه یافته رواج یافتند که به نسبت زبان‌های برنامه‌نویسی متداول توانستند وظایف و جنبه‌های بیشتری از مسائل را آشکار کنند (شکل ۱-۱).



شکل ۱-۱. مقایسه زبان‌ها منبع: (ولکینز، تیم^۱ ۲۰۱۴)

در مهندسی سیستم‌ها برای سیستم‌های توسعه‌یافته و پیچیده، کم‌کم حد و مرزهای بین طراحی سیستم توسعه‌یافته و نرم‌افزار توسعه‌یافته از بین رفت و نرم‌افزارهای توسعه‌یافته در بخش‌های مختلف مورد استفاده قرار گرفتند. بنابراین یک سیستم ترکیبی جدید به وجود آمد که چالش‌هایی در پیشرفت و توسعه سیستم محسوب می‌شد. سیستم‌های پیچیده و توسعه‌یافته خود، دارای زیر سیستم‌ها و مؤلفه‌هایی هستند که هریک به‌طور مستقل با استفاده از یک روش استاندارد قابل ایجاد و تولید هستند. اما درکل یک سیستم ترکیب همه این مؤلفه‌هاست و کنترل ارتباط و تبادل بین این مؤلفه‌ها در عملکرد یکپارچه یک سیستم پیچیده و توسعه‌یافته کار دشواری است. به‌منظور کنترل و ساخت یک سیستم پیچیده و توسعه‌یافته یکپارچه نیاز به استفاده از روش‌های مدل‌سازی و نمادگذاری ویژه‌ای است، تا هزینه‌های ایجاد و کنترل این سیستم‌ها به‌طور فزاینده افزایش نیابد. در راستای تحقق این هدف مهندسی سیستم نیازمند یک ارتباط تنگاتنگ و متقابل با توسعه نرم‌افزار است.

در این فصل ابتدا تاریخچه مهندسی سیستم و دراصل تعریف مهندسی سیستم ارائه شده است. همچنین نقش مهندس سیستم‌ها در تولید سیستم‌ها مشخص شده و در نهایت با تمرکز بر توسعه سیستم‌های نرم‌افزاری به‌عنوان زیرمجموعه‌ای از انواع مختلف سیستم موجود به چالش‌های موجود در این حوزه پرداخته می‌شود. در انتهای این فصل نیز مهندسی نرم‌افزار تعریف شده و مورد بررسی قرار می‌گیرد.

۱-۱ تاریخچه مهندسی سیستم

سیستم‌های توسعه بشری هزاران سال پیش شاید در زمان ایجاد اهرام ثلاثه مصر شکل گرفتند. در آن زمان هیچ‌کس حرفی از مهندسی سیستم نمی‌زد. می‌توان اذعان داشت که اگرچه شروع نظم و قاعده مهندسی سیستم به آن زمان برمی‌گردد، اما شروع رسمیت یافتن و بیان آن به شکل کنونی به ابتدای قرن بیستم میلادی باز می‌گردد. انقلاب صنعتی سیستم‌های بسیاری را ایجاد کرد که با مبحث مهندسی سیستم منطبق بودند. برای مثال اتومبیل‌ها، هواپیماها و موتورهای هرکدام نوعی سیستم محسوب می‌شوند. با این وجود هنوز مهندسین با مهندسی سیستم با مفهوم کنونی آشنا نبودند. مهندس ناظر معمولاً وظیفه نظارت بر کل سیستم را به‌عهده داشت و زیر بخش‌های مختلف پروژه را کمتر و یا بیشتر توسعه می‌داد. در آن زمان‌ها واسط بین زیرسیستم‌های مختلف نیز بسیار ساده بود.

آنچه که امروزه به عنوان مهندسی سیستم می‌شناسیم به اواخر دهه ۱۹۵۰ میلادی باز می‌گردد. در آن زمان بشر با سیستم‌های پیچیده و پیاده‌سازی پروژه‌هایی درگیر شد که بر تمام دانسته‌های او غالب بودند. سرعت رشد صنعت فضا و یا صنایع نظامی پروژه‌هایی را ایجاد کرد که مجبور به توسعه گسترده سیستم‌های پیچیده بودند. به عبارت دیگر سیستم‌ها مجبور بودند تا دسترس‌پذیرتر بوده و در کمترین زمان ممکن، به طور موفقیت‌آمیز پاسخ‌گو باشند. این شرایط منجر به توسعه روش‌ها در مهندسی سیستم شد.

در مباحث اقتصادی نیز سیستم‌ها پیچیده‌تر شدند و نیاز به دید کلی‌تری از سیستم به چشم می‌خورد. تأثیر مهندسی سیستم، رشد آرام ولی مطمئن‌تری را دنبال می‌کرد بنابراین روش‌ها و تکنیک‌های این قاعده و نظم، نه تنها در سیستم‌های بزرگ بلکه در پروژه‌های معمولی و کوچک نیز به کار گرفته شدند. در نهایت در سال ۲۰۰۲ میلادی یک چارچوب فرایندی تحت عنوان ISO/IEC 15288 به وجود آمد که فرایندها را در طول چرخه حیات (تولید سیستم) معرفی می‌کرد و مهندسی سیستم را به صورت یک قاعده رسمی و قابل قبول در توسعه سیستم‌ها جای داد.

۲-۱ مهندس سیستم

هریک از واژه‌های مهندسی و سیستم به تنهایی دارای معنای مشخص و شناخته شده‌ای هستند. اما وقتی لازم باشد که این دو را در کنار هم به کار ببریم یک اصطلاح به دست می‌آید که در ذهن هر شنونده‌ای معنی متفاوتی را القا می‌کند. برای آشنایی با تعریف دقیق‌تری از مهندسی سیستم لازم است تا هریک از بخش‌های آن به طور مجزا تفسیر شود.

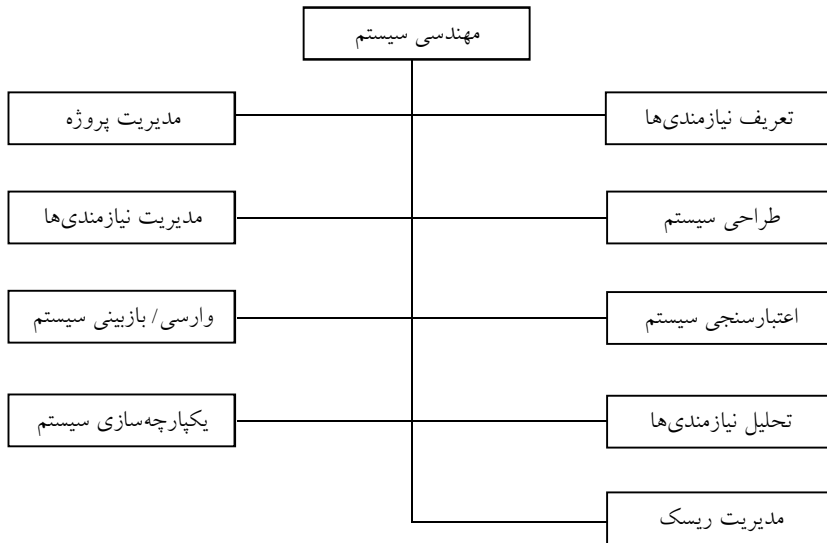
وقتی سخن از سیستم به میان می‌آید در ذهن شنونده بسیاری از موارد پیرامون خود را در ذهن تداعی می‌کند مانند یک هواپیما و یا یک ماشین یا سیستم مدیریت انبار یا سیستم ناوبری و هوانوردی یا حتی لپ‌تاپ و یا برنامه واژه‌پرداز و یا بسیاری از موارد مشابه که همه آن‌ها برای مخاطبان خود به شکل یک سیستم شناخته شده‌اند. در واقع سیستم مجموعه‌ای از مؤلفه‌ها و اجزای متفاوت است که برای رسیدن به یک هدف مشترک و معین در یک محیط مشخص باهم همکاری می‌کنند و یک کلیت جدید را به وجود می‌آورند. در یک سیستم هیچ‌یک از مؤلفه‌ها به طور مجزا نمی‌توانند به هدف نهایی سیستم دست یابند. هر مؤلفه می‌تواند شامل

نرم افزار، سخت افزار، شخص و یا هر بخش دیگری باشد. (سامرویل^۱، یان. ترجمه جعفر نژاد قمی ۱۳۹۳)

سیستم‌ها دارای اندازه‌های متفاوتی هستند و حتی یک سیستم می‌تواند از مجموعه‌ای از سیستم‌ها تشکیل شده باشد که زیر سیستم نامیده می‌شود. برای مثال سیستم هوانوردی یک کشور و سیستم اجزا یک ماشین هر دو سیستم‌هایی کامل هستند که از نظر پیچیدگی و ابعاد قابل مقایسه با یکدیگر نیستند. با این وجود وقتی صحبت از مدل کردن سیستم به میان می‌آید توجه به زیر سیستم‌ها اهمیت پیدا می‌کند. در یک ماشین کوچک‌ترین قطعات به کاررفته یک زیر سیستم محسوب می‌شوند و در سیستم هوانوردی نیز هواپیما و فرودگاه زیر سیستم‌های کوچک‌تر از کل سیستم هستند.

از سوی دیگر واژه مهندسی در معنی عام، استفاده از روش‌ها و ابزارها به شکل ساخت یافته در تولید یک محصول است. اما قرارداد دو واژه مهندسی و سیستم کنار یکدیگر و ساخت اصطلاح مهندسی سیستم به معنی روش‌های به کاررفته در توسعه یک سیستم به طور موفقیت آمیز است. مهندسی سیستم به تعاریف و مستندسازی از نیازمندی‌های سیستم در مرحله اولیه تولید، آمادگی برای طراحی سیستم و اعتبارسنجی سیستم در برآورده کردن نیازمندی‌های مورد نظر تمرکز دارد.

نقش مهندسی سیستم در مراحل مختلف ساخت نرم افزار به طور مشخصی برجسته است و مهندسی سیستم دربرگیرنده تمامی قواعد و تعاریفی است که در یک فرایند توسعه نرم افزار ساخت یافته مطرح می‌شود. این موارد می‌تواند از مفاهیم مربوط به تولید محصول در مرحله عملیاتی تا خارج شدن یک زیر سیستم از یک مرحله و ورود به مرحله دیگر را دربر گیرد. به طور کلی مهندسی سیستم کلیه نیازمندی‌های کاربران، جنبه‌های تکنیکی و اقتصادی را در نظر می‌گیرد. شکل ۱-۲ مجموعه‌ای از وظایف مهندسی سیستم را نشان می‌دهد که مفاهیم خاص در مهندسی سیستم در آن مشاهده می‌شود. در این شکل همان گونه که مشاهده می‌شود چگونگی برخورد با مسائل در مهندسی سیستم و یافتن راه حل‌های مناسب برای آن‌ها به شکل موازی مطرح شده است. به طور مثال در یک سطر تعریف نیازمندی‌ها و مدیریت پروژه وظایفی هستند که یکدیگر را پوشش می‌دهند.



شکل ۲-۱. مجموعه وظایف مهندسی سیستم (ولکینز، تیم ۲۰۱۴)

چرخه حیات یک سیستم مراحل مختلفی از موجودیت یک سیستم را توصیف می‌کند که شامل: توسعه، تشخیص، به کارگیری و پایان کار سیستم است. مشکل توصیف سیستم به این روش این است که باید دنباله‌ای از تصمیمات در مرحله توسعه اتخاذ شود تا بتواند در مراحل بعدی مؤثر واقع شوند. درواقع هیچ راهی برای افزایش اندازه سیستم در مراحل بعدی وجود ندارد به‌همین دلیل باید در مرحله توسعه به سؤالات زیر پاسخ داده شود:

- این سیستم چه مسائلی را ایجاد می‌کند؟
- این سیستم چه مسائلی را حل می‌کند؟
- این سیستم در چه محیطی قابل استفاده است؟
- این سیستم چه مدتی قابل استفاده است؟
- چگونه این سیستم قابل جایگزینی است؟

برای فهم بهتر توصیف سیستم به روش فوق، می‌توان با ذکر یک مثال آن را بهتر تفهیم کرد. با فرض وجود یک شرکت که قصد جابه‌جایی مکان استقرار خود را دارد، چه مواردی در اینجا به‌جایی بایستی مدنظر قرار گیرند؟ سؤالاتی مطرح می‌شود که هریک در نحوه چیدمان محل جدید و ارتباط بخش‌ها به‌طوری‌که مشکلی ایجاد نشود، تأثیرگذار

است. مثلاً میزها به چه شکلی قرار گیرند؟ دیوارهای جداکننده بخش‌ها چگونه چیدمان شوند تا ارتباط بین افراد و زیر مجموعه‌های شرکت درست برقرار باشد؟ اتصالات برق و تلفن چگونه انجام شود؟ آیا چیدمان شرکت مطابق با آنچه خواسته و نیاز اعضا و مسئولین شرکت است انجام شده است؟ به‌طور قطع همه این مسائل در عملکرد شرکت در مرحله استفاده تأثیرگذارند و تغییرات در مراحل بعدی را دشوار و حتی غیرممکن می‌کنند.

۳-۱ فرایندهای مهندسی سیستم

برای داشتن یک دیدگاه کلی و خوب در مورد فرایندهای مهندسی سیستم موارد ذیل را باید مدنظر قرار داد:

(الف) تعریف مسئله

(ب) ارزیابی راه‌حل‌های مختلف برای حل مسئله

(ج) مدل‌کردن سیستم

(د) یکپارچه‌سازی

(ه) آغاز به کار سیستم

(و) ارزیابی کارایی سیستم

(ز) بازبینی ارزیابی مجدد و ترمیم سیستم در صورت نیاز

در ادامه هریک از موارد فوق به اختصار شرح داده می‌شود:

(الف) تعریف مسئله: برای شروع به کار یک پروژه توسعه سیستم، ابتدا لازم است کلیه وظایف سیستم تعریف شود. راه‌حل مناسب زمانی به درستی انتخاب می‌شود که همه اولویت‌ها، وظایف و اهداف سیستم در این مرحله به درستی شناسایی و مشخص شود. خطاها در مرحله تعریف مسئله، بسیار هزینه‌بردار هستند. در واقع هر وظیفه، آن کاری است که سیستم انجام می‌دهد و یا آن نیازی است که در سیستم مشاهده می‌شود. مهندسین سیستم نیز با نیازمندی‌هایی که توسط کاربران آن سیستم بیان می‌شوند، سروکار دارند. مشخص کردن نیازمندی‌ها لزوماً یافتن راه‌حل نیست، اما پیدا کردن درست آن‌ها می‌تواند مانع از سردرگمی مهندس سیستم در انتخاب راه‌حل درست شود.

(ب) ارزیابی راه‌حل‌های مختلف برای حل مسئله: یکی از مهم‌ترین کارهایی که یک مهندس سیستم باید انجام دهد ارزیابی و وزن‌دادن به راهکارهای مختلف است.

متأسفانه اغلب در این مسئولیت اهمال و غفلت صورت می‌گیرد. یک راه‌حل تنها برای حل یک مسئله مشخص به ارزیابی و بررسی گزینه‌های جایگزین دیگر ترجیح داده می‌شود، در صورتی که ممکن است راه‌حل دیگری وجود داشته باشد که بسیار مناسب‌تر بوده و بهتر عمل کند. بنابراین یک مهندس سیستم در مدل‌کردن نیازمندی‌ها تنها یک طراحی از سیستم ارائه نمی‌دهد بلکه می‌تواند چندین مدل طراحی جایگزین برای یک سیستم مطرح کند. در واقع توصیه می‌شود که مهندس سیستم گزینه‌های مختلف را ارزیابی کرده و وزن‌دهی کند.

به‌ندرت پیش می‌آید که یک راه‌حل مشخص برای تمام مسائل مربوط به سیستم مناسب باشد و بهتر است شرایط و اولویت‌های مختلف در آن مورد توجه قرار گیرند. مواردی از جمله هزینه، اندازه، وزن، زمان توسعه، زمان ارائه به بازار (که بستگی به بازاریابی دارد) و همچنین ریسک باید مورد ارزیابی قرار گیرند. شبیه‌سازی پارامترها از مدل‌های سیستم مشتق می‌شوند تا جنبه‌های مورد نیاز مستقیماً قابل مشاهده و ارزیابی و اندازه‌گیری باشند.

ج) مدل‌سازی سیستم: طراحی مدل سیستم در مرحله انتخاب راه‌حل مناسب ایجاد می‌شود. در این مرحله تنها جزئیات راه‌حل‌های انتخاب‌شده مطرح می‌شوند. دیدگاه مهندس سیستم هم در مدیریت‌کردن کل چرخه تولید سیستم و هم در طراحی هر بخش خاص (زیر سیستم) از سیستم تأثیرگذار است. با استفاده از زبان مدل‌سازی (SYSML) که یک زبان مدل‌سازی گرافیکی و استاندارد است و به‌منظور مدل‌سازی فرایندها در مهندسی سیستم طراحی شده است می‌توان نیازمندی‌ها را در این مرحله به‌درستی تشخیص داده و پیمایش کرد.

د) یکپارچه‌سازی: یک سیستم نمی‌تواند تنها در یک محیط بسته به حیات خود ادامه دهد و فقط به خودش سرویس برساند، بلکه در یک محیط باز قرار می‌گیرد و می‌تواند با آن محیط تعامل داشته باشد. در این مرحله باید سیستم یکپارچه‌سازی شده و این کار به کمک تعریف واسطه‌های سیستمی انجام می‌شود. با چنین نگاهی به اصطلاح، یک سری جزایر نرم‌افزاری به‌دست می‌آید که باید سعی در به‌هم پیوستگی آن‌ها با هدف یکپارچگی کل سیستم داشت. اما یک راهکار جایگزین و مناسب‌تر در رویکرد مهندسی نرم‌افزار این است که به‌جای اینکه اصل بر فرایندها باشد، اصل بر اجزای سیستمی یک نرم‌افزار قرار گیرد چراکه در مهندسی نرم‌افزار اصل بر سیستم

نرم‌افزاری است. به این ترتیب که به جای فرایندها، اشیا یا کلاس‌ها به عنوان عامل اصلی قرار گیرد. تنها در این صورت است که دید تحلیلی‌گر یا طراح سیستم از اجزای زیر سیستمی که پیوند سستی باهم دارند به سیستمی تغییر می‌کند که در آن تحلیل یا طراحی براساس اجزای مشترک (مثلاً اشیا و کلاس‌ها) صورت می‌پذیرد.

در دیدگاه ساخت یافته به دلیل عدم انعطاف، همان جزیره‌های نرم‌افزاری که اشاره شدند پدید می‌آیند و یکپارچه‌سازی در آن کاری بسیار دشوار است. اما در دیدگاه شیء‌گرا منظور از یک سیستم، مجموعه‌ای از اشیا یا کلاس‌هاست که فارغ از اینکه به کدام بخش یا زیر سیستم از یک سیستم اصلی تعلق دارند، با همدیگر همکاری کرده و هدف واحدی را که همانا نرم‌افزار یکپارچه است تعقیب می‌کنند.

ه) شروع به کار سیستم: همان‌طور که اشاره شد، سیستم براساس یک مدل طراحی خاص ایجاد شده و شروع به کار می‌کند. مدل‌سازی در مراحل قبل، نیازمندی‌های پیاده‌سازی سیستم مانند نرم‌افزار، سخت‌افزار، بخش‌های مکانیکی و سایر قسمت‌ها را فراهم آورده‌اند، بنابراین با وجود این شرایط اولیه، سیستم می‌تواند شروع به کار کند.

و) ارزیابی کارایی سیستم: در ابتدا که سیستم آماده به کار می‌شود، باید موردآزمون قرار گرفته و ارزیابی شود و همچنین نتایج به دست آمده از این ارزیابی‌ها با نیازمندی‌های سیستم مطابقت داده می‌شود. در مدل SYSML ارتباط بین مدل تست و مدل نیازمندی‌ها با استفاده از بازبینی ارتباطات به وجود می‌آید.

ز) بازبینی ارزیابی مجدد و ترمیم سیستم در صورت نیاز: این فعالیت پس از اتمام مراحل فوق انجام می‌شود و به‌طور دقیق فرایند را مورد بررسی و ارزیابی قرار می‌دهد. بنابراین دنباله‌ای از باز خوردهای به دست آمده برای عناصر سیستم فراهم می‌شود.

۱-۴ مدیریت ریسک

یک مسئله بسیار مهم در مدیریت سیستم، مدیریت ریسک است. تضمین مدیریت ریسک در کل مراحل پروژه نقش کلیدی ایفا می‌کند. در انجام یک پروژه هیچ‌گاه رسیدن به یک حالت ایده‌آل و کامل که در ذهن مدیران پروژه وجود دارد، اتفاق نمی‌افتد. در حالت نرمال تنها می‌توان کاری کرد که پروژه در مسیری به پیش رود که کمتر دچار خطا شود.

مدیریت ریسک مسئول شناسایی ریسک‌های بالقوه و تعریف معیارهایی برای کاهش میزان ریسک است. همچنین مدیریت ریسک در مواجهه با ریسک‌های به وجود آمده باید به دنبال راه حل مناسب باشد. مدیریت ریسک بخشی از فرایند تصمیم‌گیری در یک پروژه است و نیاز به توانایی مشاهده و تشخیص در مواجهه با ریسک و داشتن عکس العمل مناسب در برخورد با آن را دارد. (پرسمن، راجر اس^۱ ۱۳۹۲ ترجمه جعفر نژاد قمی)

۱-۵ وظایف مهندس سیستم

مهندس سیستم یک رابط و مسئول برقراری نظم در کل پروژه است که در طول کل پروژه مستقل از دیدگاه نرم‌افزاری یا سخت‌افزاری و یا سایر دیدگاه‌ها عمل می‌کند و می‌اندیشد. از دیدگاه ساخت یافته این نظم در مشخص کردن وظایف مهندس سیستم، در ارتباط با واحدهای کاری مطرح می‌شود که به طور مستقیم به مدیر هماهنگی کل پروژه گزارش می‌دهند، در حالی که خود با دیگر بخش‌های توسعه، تبادل اطلاعات دارند.

مهندسين سیستم در واقع واسطه بين مدير پروژه و ديگر بخش‌های توسعه هستند، بلکه در سطح سیستم ایفای نقش می‌کنند. آن‌ها باید بتوانند با بخش‌های مختلف توسعه تبادل اطلاعات داشته باشند نه اینکه تنها ناظر ارتباط بخش‌ها و تبادل پیام بین آن‌ها باشند. در حقیقت مهندس سیستم نقش تصمیم‌گیرنده اصلی در سیستم را داراست و مطابق با استاندارد INCOSE، مهندس سیستم ۲۰ تا ۳۰ درصد کل پروژه را به دوش می‌کشد. بسیاری از پروژه‌ها فاقد بخش ساخت یافته مهندسی سیستم هستند که نمونه‌ای از آن در شکل ۱-۳ دیده می‌شود. وظایف موجود در این نوع پروژه‌ها توسط مدیریت کل پروژه و بخش‌های توسعه منفرد اداره می‌شوند. این نوع روش انجام پروژه در دسرهای زیادی را برای مدیر پروژه به همراه می‌آورد. بنابراین مدیریت کل پروژه باید این توانایی را داشته باشد تا وظایف مهندسی سیستم را به دور از محتوای مورد توجه در آن حدس بزند و این امر منجر به کاهش زمان در دسترس‌پذیری مدیر نسبت به سایر وظایف مدیریتی‌اش می‌شود. (ولکینز، تیم ۲۰۱۴)