فصل نهم

مراحل و زمان انجام کار

هدف‌های رفتاری

۱- مراحل تجزیه و تحلیل سیستم را توضیح دهد.

۲- مراحل طراحی و پیاده‌سازی بانک اطلاعاتی را توضیح دهد.

۳- مراحل کدنویسی سیستم اطلاعاتی را نام ببرد.

۴- زمان تجزیه و تحلیل سیستم را به‌کار ببرد.

۵- زمان طراحی و پیاده‌سازی بانک اطلاعاتی را به‌کار ببرد.

۶- زمان کدنویسی و مستندسازی یک سیستم اطلاعاتی را به‌کار ببرد.

۷- مراحل و زمان نگهداری و پشتیبانی از سیستم اطلاعاتی را توضیح دهد.

۸- اصول تعیین مراحل و زمان انجام کار را توضیح دهد.

۹-۱ مقدمه

مراحل فرایند تولید نرم‌افزار از چندین بخش تشکیل می‌شود. مرحله نخست، تحلیل نیازها می‌باشد به‌طوری‌که استخراج نیازهای یک نرم‌افزار مطلوب، اولین مرحله تولید آن به‌شمار می‌رود. در حالی که بیشتر کارفرمایان (مشتریان) گمان می‌برند که می‌دانند نرم‌افزار موردنیازشان باید دارای چه توانایی‌ها و امکاناتی باشد. در حقیقت برای انگشت گذاشتن بر روی نیازهای ناقص و ناکافی، بحث برانگیز و مشکوک یا متناقض کارفرما، نیاز به تجربه و مهارت بسیار در فنون مهندسی نرم‌افزار می‌باشد. مرحله بعد، تشریح یا استخراج مشخصات فنی نرم‌افزار است. تشریح عبارت است از تعریف دقیق نرم‌افزاری که قرار است تولید شود با زبان دقیق ریاضی. متأسفانه معمولاً این مرحله از تولید نرم‌افزار معمولاً تنها زمانی انجام می‌شود که نرم‌افزار به پایان رسیده باشد. مرحله سوم طراحی و معماری است. طراحی و معماری به معنی معین‌کردن رفتار نرم‌افزار به صورت کلی و بدون درگیر شدن با جزییات آن است. در گام بعدی به مرحله کد نویسی می‌رسیم. تقلیل یک طراحی به کد ممکن است بدیهی‌ترین بخش مهندسی نرم‌افزار به نظر برسد، ولی لزوماً بزرگ ترین بخش آن نیست. بعد از مرحله کدنویسی، مرحله آزمایش قرار دارد که شامل آزمایش بخش‌های مختلف نرم‌افزار، خصوصاً در بخش‌هایی که کدی که توسط دو مهندس متفاوت نوشته شده باید به هم متصل شوند (با هم کار کنند) می‌شود. مرحله‌ای بسیار مهم و عموماً فراموش شده تولید نرم‌افزار، مستندسازی است. مستندسازی ساختار داخلی نرم‌افزار به منظور حفظ توانایی تصحیح و گسترش آن در آینده می‌باشد. در انتها، نگهداری و بهبود نرم‌افزار به منظور ایجاد توانایی مواجهه با مشکلات تازه کشف‌شده و نیازهای جدید است که می‌تواند زمانی بسیار بیشتر از تولید اولیه نرم‌افزار صرف خود کند.

۹-۲

مراحل تجزیه و تحلیل سیستم

سیستم چیست؟

در فصل‌های قبل گفته شد که سیستم مجموعه ای است که از اجزای به هم وابسته که وابستگی حاکم بر اجزای خود کلیت جدید را احراز کرده و از نظم و سازمان خاصی پیروی می‌نماید و در جهت تحقق هدف معینی که دلیل وجودی آن است فعالیت می‌کند.

واژه سیستم با مفاهیم بسیاری به‌کار برده می‌شود مانند سیستم اقتصادی، سیستم سیاسی، حمل و نقل، ارتباطات، حسابداری، مکانیزه (ماشینی)، سیستم اطلاعات مدیریت و... . در زبان فارسی نیز گاهی اوقات «سیستم» را به‌عنوان

واژه‌ای مترادف با کلمه‌های رشته (رشته جبال)، جهاز (جهاز هاضمه)، دستگاه (دستگاه گوارش، دستگاه گردش خون)، سلسله (سلسله اعصاب) و منظومه (منظومه شمسی) مورد استفاده قرار می‌دهند.

ترکیب سیستم: اجزای چهارگانه سیستم

۱- درونداد (ورودی): آنچه به‌نحوی وارد سیستم می‌شود و سبب تحرک سیستم می‌شود.

۲- فرایند تبدیل (پردازش): جریان تغییر و تبدیل آنچه وارد سیستم می‌شود.

۳- برونداد (خروجی): از تغییر و تبدیل در سیستم (به شکل کالایا خدمات) خارج می‌شود.

۴- بازخورد: فرایندی دورانی که قسمتی از ستاده به‌عنوان اطلاعات به درونداد پس خورانده می‌شود.

محیط سیستم: هر سیستم در محیطی قرار دارد. محیط سیستم شامل کلیه متغیرهایی است که می‌تواند در وضع سیستم مؤثر باشند و یا از سیستم تأثیر پذیرند. عوامل محیطی در برگیرنده عواملی همچون عوامل طبیعی، فرهنگی ایدئولوژی، اجتماعی، سیاسی، اقتصادی و غیره هستند.

طبقه‌بندی سیستم‌ها:

سیستم‌های اصلی: که در برگیرنده مجموعه‌ای از سیستم‌های فرعی می‌باشد.

سیستم‌های فرعی: که جزیی از یک سیستم بزرگ‌تر بوده و جهت تحقق هدف‌های سیستم اصلی فعالیت می‌کند.

سیستم‌های باز و بسته:

سیستم بسته : سیستمی‌ساده است که با محیط خود ارتباط برقرار نمی‌کند و در برخورد با محیط، سازمان خود را از دست می‌دهد.

سیستم باز : سیستمی‌است که با محیط خود در ارتباط است.

آنتروپی

در هر سیستم عواملی وجود دارند که بر خلاف جهت نظم سیستم عمل می‌کنند و مختل کننده انتظام سیستم هستند این عوامل را آنتروپی می‌نامند.

انواع آنتروپی:

۱- آنتروپی مثبت : عملکردش در خلاف جهت نظم سیستم است.

۲- آنتروپی منفی : عملکرش خلاف جهت آنتروپی مثبت است و برای ایجاد تغییرات تعدیلاتی در جهت اصلاح انحرافات به منظور بقاء سیستم در محیط عمل می‌کند.

خواص سیستم های باز:

۱- کلیت و جامعیت وجودی

۲- سلسله مراتب

۳- همبستگی بین اجزاء

۴- تناسب بین اجزاء

۵- گردش دایره وار

۶- خاصیت تولید مثل

۷- همپایانی

۸- گرایش به فنا

۹- گرایش به تکامل

۱۰- گرایش به تعادل یا خود نگهداری

۱- کلیت و جامعیت وجودی: سیستم در کلیت وجودی خود خواصی را ظاهر می‌سازد که در اجزای تشکیل‌دهنده آن به‌تنهایی وجود ندارد. این کلیت نتیجه ارتباط اجزاء با یکدیگر و نحوه ترکیب اجزاء و سازمان یافتن آنها نیز کلیت سیستم را به‌وجود می‌آورد.

۲- سلسله مراتب: در سیستم‌ها نوعی سلسله مراتب از نظر ساختاری،عملکرد و رفتاری وجود دارد. در هر سیستم عناصری وجود دارد که به نوبه خود عناصر کوچکتری هستند که ساخت و عملکرد ساده‌تری دارند.

۳- همبستگی بین اجزاء: هر جزء در سیستم به نحوی با سایر اجزاء مرتبط است و به علت وجود این همبستگی چنانچه در جزیی خللی وارد شود، سایر اجزاء نیز از آن خلل متأثر می‌شوند.

۴- تناسب بین اجزاء: بین اجزاءسیستم تناسب، سنخیت وکمال متقابل موجود است و وجود تناسب سبب حفظ هویت و کلیت سیستم می‌شود.

۵- گردش دایره وار:فرایند درونداد،تبدیل و برونداد جریانی مستمر و مداوم است. به این معنی که با صدور برونداد، سیستم بار دیگر آماده کسب نیرو و تجدید فعالیت گردیده و این جریان به شکل گردشی دایره‌وار ادامه می‌یابد.

۶- خاصیت تولید مثل: سیستم‌ها گرایش به جاودانه‌سازی خود دارند و تا جایی که امکان داشته باشد به حیات خویش ادامه می‌دهند و چنانچه در کار سیستم نقصی پدید آید در رفع آن می‌کوشند و برای ادامه حیات تلاش می‌کنند، در غیر این صورت از طریق تولید مثل وجود خود را در دیگری ادامه می‌دهند.

۷- همپایانی: همپایانی بدین‌معنی است که سیستم می‌تواند از راه‌ها ومسیرهای متفاوتی به هدف واحدی برسد.

۸- گرایش به فنا: در درون سیستم‌ها عواملی به‌وجود می‌آیند که سیستم‌ها را از جهت اصلی آن منحرف می‌سازند و به سمت عدم تعادل سوق می‌دهند. این عوامل را آنتروپی می‌خوانند.

۹- گرایش به تکامل: منظور از تکامل پیچیدگی ساخت و تنوع خواص است و چنانچه ساختار سیستم پیچیده‌تر شود و در اثر آن پیچیدگی، عملکردهای متنوع تری از سیستم به ظهور رسد و خواص بیشتری ارائه شود، سیستم متکامل‌تر شده است.

۱۰- گرایش به تعادل یا خود نگهداری پویا: این خصیصه که به هوموستاسیس معروف است بیانگر تلاش سیستم در حفظ متغیرهای ضروری خود در محدوده‌های معین به منظور ادامه حیات سیستم می‌باشد.

نظریه عمومی سیستم‌ها: این نظریه توسط برتالانفی ارائه گردید و بر اساس این نظریه یک ارگانیسم، صرفاً مجموع عناصر جداگانه‌ای نبوده بلکه سیستمی است که دارای نظام و کلیت می‌باشد که مرتباً در حال تغییر و تبدیل است. به اعتقاد وی ارگانیزم را نمی‌توان با شیوه تفکر و روش‌های معمول در مکتب مکانیسمی شناخت و باید طرز تفکر نوینی را برای شناخت موجودات ارگانیک ابداع کرد، این نظر برتالانفی به نظریه عمومی سیستم‌ها شهرت یافت.

نگرش سیستمی: این نگرش، چارچوبی منطقی و علمی‌ارائه می‌دهد که چندبعدی بوده و چارچوبی برای در نظر گرفتن عوامل محیطی، داخلی و خارجی سیستم به‌عنوان یک کل متشکل ارائه می‌دهد و به پدیده‌های اطراف به‌صورت یک کل به هم پیوسته می‌نگرد.

تعریف سیستم اطلاعات: تعاریف متعددی از سیستم اطلاعات در رشته‌هایی مانند مدیریت، علوم کامپیوتر، مهندسی نرم‌افزار، و علوم کتابداری و اطلاع‌رسانی ارائه شده است. تعریف زیر در واژه‌نامه انجمن کتابداری به عنوان (یانگ ۱۹۸۳) تعریفی جامع از سیستم‌های اطلاعات است که در اینجا با اندکی تغییر در جمله‌بندی ارائه می‌شود:

«یک سیستم کامل طراحی شده برای تولید، جمع‌آوری، سازماندهی (پردازش)، ذخیره، بازیابی و اشاعه اطلاعات در یک مؤسسه، سازمان یا هر حوزه تعریف شده دیگر از جامعه»

همچنین سیستم اطلاعات یک سیستم کامل است برای هدف در نظر گرفته شده. این تعریف سیستم اطلاعات را محدود یا ملزم به داشتن اجزایی مانند انسان، ماشین و یا غیره نمی‌کند چراکه یک سیستم اطلاعات ممکن است بتواند بدون داشتن هر یک از این اجزاء نقش خود را به‌صورت کامل ایفا کند. سیستم اطلاعات شامل فرایندهای تولید، جمع‌آوری، سازماندهی، ذخیره، بازیابی و اشاعه اطلاعات است. اگر یک سیستمی‌همه این فرایندها را انجام نمی‌دهد نمی‌توان به آن یک سیستم اطلاعات گفت. آن سیستم یا فرایند ممکن است یک زیر سیستم از یک سیستم اطلاعات

باشد. سیستم اطلاعات می‌تواند برای یک مؤسسه، سازمان یا هر حوزه تعریف‌شده دیگر از جامعه مطرح شود. اگر چه سیستم‌های اطلاعات در ابتدا برای استفاده در سازمان مطرح شده و شکل گرفته‌اند اما برای حوزه‌هایی فراتر از سازمان (به‌عنوان مثال جامعه پژوهشگران، شهروندان و...) نیز قابل کاربرد است. در این تعریف از واژه «داده» استفاده نشده است. خیلی از منابع دو واژه داده و اطلاعات را متفاوت تعریف می‌کنند و اطلاعات را داده‌های پردازش شده می‌دانند. مرز تفاوت بین داده و اطلاعات، نسبی است چرا که ممکن است اطلاعاتی دوباره پردازش، شوند و اطلاعات جدیدی تولید کنند. بنابراین در این شرایط نمی‌توان گفت اطلاعات قبل از پردازش، داده بوده‌اند.

یک سیستم اطلاعات چه کار می‌کند؟

یک سیستم اطلاعات سه فعالیت عمده انجام می‌دهد. ابتدا اطلاعاتی را از منابع درون‌سازمانی یا برون‌سازمانی به‌عنوان ورودی دریافت می‌کند. سپس بر روی اطلاعات دریافت شده کارهایی انجام می‌دهد تا اطلاعات مورد نظر سیستم را تولید کند. در نهایت، اطلاعات تولید شده را در اختیار کاربر در نظر گرفته شده مثلاً یک مدیر یا یک کارمند قرار می‌دهد. به‌عنوان مثال در یک سیستم اطلاعات رایانه‌ای که در بانک کار مربوط به حساب‌های پس‌انداز را انجام می‌دهد، اطلاعات توسط کارمند بانک وارد می‌شود و پس از پردازش و ذخیره، رسیدی چاپ شده و به مشتری تحویل داده می‌شود. همچنین مشتری می‌تواند گزارشی از کارکرد حساب خود را دریافت نماید.

۹-۳

تعریف تجزیه تحلیل سیستم‌ها

تجزیه تحلیل سیستم‌ها عبارت است از شناخت جنبه‌های مختلف سیستم و آگاهی از چگونگی عملکرد اجزای تشکیل‌دهنده آن و بررسی نحوه و میزان ارتباط بین اجزای آن به منظور دستیابی به مبنایی جهت طرح و اجرای یک سیستم مناسب‌تر.

تعریف سیستم در یک سازمان

در یک سازمان سیستم را مجموعه‌ای از روش‌ها نیز تعریف کرده‌اند که به یکدیگر وابسته بوده و با اجرای آنها قسمتی از هدف سازمانی محقق می‌شود.

تعریف روش: عبارت است از یک رشته عملیات که برای اجرای کل یا قسمتی از یک سیستم انجام می‌گیرد مانند

روش استخدام در یک سیستم پرسنلی یا روش انبارداری دریک سیستم تدارکاتی.

تعریف شیوه: عبارت است از تشریح جزییات و نحوه انجام دادن کار مثل استفاده از کارت جهت حضور و غیاب کارکنان

تجزیه و تحلیل سیستم‌ها در موارد زیر به مدیران کمک می‌کند:

۱- بررسی دوباره هدف‌های سازمانی.

۲- آشنایی بیشتر با نحوه کارها.

۳- کمک در پی بردن به کمبودها، نقایص و مشکلات.

۴- با استفاده از روش‌های علمی، راه‌ها و شیوه‌های بهتری را انتخاب و به مرحله اجرا بگذارند.

تشریح مراحل تجزیه و تحلیل سیستم، طراحی و پیاده‌سازی

مرحله اول : شناخت مشکل و تبیین آن: مشکل یا مشکلات مربوطه ممکن است از سوی مدیران، مقامات مسئول سازمانی و یا شخص تحلیل گر شناسایی گردد.

مشکل مربوطه بایستی :

- به‌اندازه کافی اهمیت داشته باشد که وقت و هزینه را بتوان صرف آن کرد.
- برای تشخیص مهم بودن مشکل باید ارتباط آن با هدف سازمان را بررسی کرد.
- در شناخت مشکل باید دقت کرد علت‌ها با معلول‌ها اشتباه نشوند.

مرحله دوم: ایجاد فرضیه: پس از شناخت مشکل بایستی درباره عواملی که سبب بروز مشکل شده‌اند حدس زد و فرضیاتی را مطرح کرد و فرضیه اهم (مهم‌ترین و محتمل‌ترین راه حل)را برگزید.

مرحله سوم: جمع‌آوری اطلاعات: در این مرحله بایستی اطلاعاتی را پیرامون مشکل و راه حل‌های آن کسب کرد.هر چه صحت و دقت اطلاعات بیشتر باشد،احتمال شناخت واقعیت و دستیابی به راه حل مناسب برای مشکل، بیشتر خواهد بود.

مرحله چهارم : طبقه‌بندی اطلاعات: در این مرحله تحلیل گر داده‌های پراکنده را طبقه‌بندی نموده و به آنها نظم می‌بخشد تا معنی‌دار شوند. این اطلاعات به روش‌های منطقی و عقلایی و با توجه به ماهیت و نوع آنها طبقه‌بندی و کدگذاری می‌شوند.

روش‌های طبقه‌بندی اطلاعات:

۱- استفاده از جدول

۲- استفاده از نمودارها

جدول: جدول بندی یکی از روش‌های طبقه‌بندی اطلاعات است که به تحلیل‌گر کمک می‌کند تا وجوه تشابه و همبستگی اطلاعات را که به کمک طبقه‌بندی منطقی به‌صورت ردیف‌ها و ستون‌های افقی و عمودی درآمده است به چشم ببیند.

نمودار: نمودارها از وسایل ترسیمی‌طبقه‌بندی و نظم بخشی به اطلاعات هستند و تحلیل گر با استفاده از آنها می‌تواند اطلاعات را به‌صورتی تنظیم و منعکس کند که درک آن برای بیننده و خواننده گزارش آسان‌تر شود و با صرف وقت کوتاهی از پیام آن مطلع گردد.

محاسن نمودارها:

۱- مقایسه اطلاعات را آسان می‌کنند.

۲- چون از علائم در آنها استفاده می‌شود از طولانی شدن کلام جلوگیری می‌کنند.

۳- با کمک آنها بهتر می‌توان روند تغییرات و تفاوت بین دو یا چند روند را مشاهده کرد.

برخی از مهم‌ترین انواع نمودارها:

الف) نمودار خطی

ب) نمودار میله‌ای یا ستونی

ج) نمودار دایره‌ای

چ)نمودار فضایی

ح) نمودار سازمانی

مرحله پنجم: تجزیه و تحلیل اطلاعات: دراین مرحله تحلیل‌گر می‌کوشدتا ارتباط بین اطلاعات را کشف کند. در این مرحله سؤالاتی از قبیل سؤالات ذیل درباره اطلاعات پرسیده می‌شود:

۱- چه فعالیتی انجام می‌شود؟

۲- چرا آن فعالیت انجام می‌شود؟

۳- آن فعالیت را چه کسی انجام می‌دهد؟

۴- آن فعالیت چگونه انجام می‌شود؟

۵- آن فعالیت در کجا انجام می‌شود؟

۶- آن فعالیت در چه زمانی انجام می‌شود؟

روش‌های اثبات منطقی در تجزیه و تحلیل اطلاعات:

الف) قانون توافق مثبت

ب) قانون توافق منفی

ج) متد ترکیبی با تغییرات ملازم

مرحله ششم: نتیجه‌گیری و ارائه راه‌حل: در این مرحله تحلیل‌گر به تعبیر و تفسیر یافته‌های خویش پرداخته و چنانچه فرضیه‌های اولیه او تأیید گردند او موفق به کشف علت شده است و اگر تأیید نشده باشند بایستی به‌دنبال راه‌حل‌های دیگری برای مشکل باشد.

نحوه ارائه راه‌حل: دراین مرحله تحلیل‌گر با کمک قدرت خلاقیت و ابتکار خویش و به مدد شناختی که نسبت به وضع موجود به‌دست آورده است پیشنهاداتی معقول و منطقی جهت رفع مشکلات و نقائص ارائه می‌دهد.

نکاتی که در زمینه ارائه راه‌حل باید رعایت گردند:

۱- همخوانی راه حل با برنامه‌های سازمان

۲- ارائه چند راه‌حل به‌جای یک راه حل

۳- مطابقت با قوانین و مقررات

۴- قابلیت اعمال

۵- تناسب بین هزینه اجرا و منافع حاصل از اجرای طرح

۶- مسئولیت اجرا

مرحله هفتم: تهیه و تنظیم گزارش: اقداماتی که تا این مرحله انجام گرفته‌اند توسط تحلیل‌گر در یک گزارش منظم تدوین و در دسترس مدیران و مقامات ذی‌ربط قرار می‌گیرد.

مرحله هشتم: اجرا : در این مرحله تحلیل‌گر طی برنامه‌ای پیش‌بینی‌های لازم را جهت اجرای پیشنهادات مصوب و پیاده کردن طرح‌های جدید و ایجاد تغییر در نظام قدیم انجام می‌دهد.

۱- تهیه برنامه زمان‌بندی اجرای راه‌حل مناسب

۲- تعیین شاخص‌های ارزیابی موفقیت

۳- اجرای برنامه زمان‌بندی حل شده

۴- ارزیابی موفقیت

۹-۴

مراحل کدنویسی

اهمیت طراحی برنامه

فرض کنید یک مسئله را برای تعدادی دانشجو تعریف و از آنان خواسته شده است که برنامه‌ای به‌منظور حل مسئله مورد نظر، طراحی و پیاده‌سازی نمایند. پس از صرف چند ثانیه، بلافاصله دانشجویان شروع به تایپ کد مورد نظر خود به‌منظور حل مسئله می‌نمایند. در بین دانشجویان، دانشجویی وجود دارد که کاغذی را بر می‌دارد و شروع به نوشتن موضوع می‌نماید. دقایقی سپری می‌گردد، اما همچنان دانشجویان مشغول تایپ برنامه خود و یا احتمالاً اشکال زدایی! آن هستند.تقریباً بدون استثناء، دانشجویی که دیرتر از دیگران آغاز نموده است، با سرعت بیشتری تکلیف خود را به پایان رسانده و حتی راه‌حل ارائه شده توسط وی، نیز از سایر دانشجویان به‌مراتب بهتر است.چرا؟ در صورتی‌که به کاغذی که در اختیار دانشجو قرار داده شده، دقت نمائید، یک طرح مناسب به‌منظور طراحی برنامه را برای مسئله، مشاهده خواهید کرد. برخی از دانشجویان نیز ممکن است چندین کاغذ را تکمیل و یک طراحی پیچیده را انجام داده باشند. نکته مهم در این رابطه این است که این دانشجویان (چه آنانی که یک طراحی ساده را انجام داده‌اند و چه آنانی که یک طراحی پیچیده را دنبال نموده‌اند)، دارای یک الگو(طرح) برای برنامه خود، می‌باشند.

الگوریتم: هر برنامه، می‌بایست دارای یک طرح و یا الگو بوده تا برنامه‌نویس بر اساس آن عملیات خود را دنبال نماید. از دیدگاه برنامه نویسان، هر برنامه نیازمند یک الگوریتم است. به‌عبارت ساده، الگوریتم، بیانیه‌ای روشمند به‌منظور حل یک مسئله بخصوص است. از منظر برنامه نویسان،الگوریتم به‌منزله یک طرح کلی و یا مجموعه دستورالعمل‌هایی است که با دنبال نمودن آنان، برنامه‌ای تولید می‌گردد.

الگوریتم‌های میکرو در مقابل ماکرو: الگوریتم‌ها دارای ویژگی‌های متفاوتی می‌باشند. ما می‌توانیم در رابطه با الگوریتم استفاده شده به منظور نوشتن یک برنامه مشخص صحبت نمائیم. از این زاویه، ما صرفاً در رابطه با الگوریتم در سطح ماکرو(macro level)، صحبت نموده‌ایم. در چنین مواردی، الگوریتم ارائه شده، سعی در به‌دست‌آوردن جنبه‌های عمومی‌برنامه از طریق یک مرور کلی به برنامه در مقابل درگیر شدن در جزئیات را دارد. ما می‌توانیم در رابطه با الگوریتم‌ها، از سطح «میکرو» صحبت نمائیم. از این زاویه، به سطوح پایین‌تر رفته و به عوامل اساسی و نگهدارنده‌ای که یک جنبه خاص از برنامه را با یکدیگر مرتبط می‌نماید، صحبت کرد. مثلاً در صورتی‌که شما دارای داده‌هایی هستید که می‌بایست قبل از استفاده مرتب گردند، الگوریتم‌های مرتب‌سازی متعددی در این زمینه وجود

داشته و می‌توان یکی از آنها را به‌منظور تأمین اهداف مورد نظر خود انتخاب نمود. انتخاب یک الگوریتم مرتب‌سازی، صرفاً باعث حل شدن یکی از جنبه‌های متفاوت برنامه می‌گردد. پس از مرتب‌سازی داده‌ها، می‌بایست از یک الگوریتم میکرو دیگر به‌منظور نمایش داده‌های مرتب شده استفاده گردد.

همان‌گونه که احتمالاً حدس زده‌اید، می‌توان تمام الگوریتم‌های میکرو را به‌منظور ایجاد یک الگوریتم ماکرو، جمع‌آوری کرد. اگر با الگوریتم‌های میکرو، آغاز نمائیم، و حرکت خود را به‌سمت نمایش ماکروی یک برنامه، پیش ببریم، کاری را انجام داده‌ایم که موسوم به طراحی «پایین به بالا» (buttom-up)، است. اگر فعالیت خود را با یک الگوریتم ماکرو آغاز و حرکت خود را به‌سمت پائین و الگوریتم‌های میکرو، ادامه دهیم، طراحی از نوع «بالا به پایین top-down)» را انجام داده‌ایم. شاید این سؤال مطرح گردد که کدام روش بهتر است؟ هر رویکرد، دارای نکات مثبت و منفی مربوط به خود است. صرفه نظر از رویکرد طراحی استفاده شده، می‌بایست دارای الگویی (طرحی) مناسب برای برنامه باشیم.حداقل، نیازمند یک اعلامیه از مسئله برنامه نویسی و یک طرح (الگو) برای برخورد با مسئله، خواهیم بود. پس از شناخت مسئله، می‌توان نحوه حل مسئله را ترسیم کرد. شناخت عمیق و مناسب نسبت به مسئله‌ای که قصد حل آن را داریم، شرط اساسی و ضروری برای طراحی یک برنامه است.

با توجه به اینکه این اعتقاد وجود دارد که شناخت جامع و کلی از مسئله‌ای که حل آن را داریم، بخشی ضروری در اولین مرحله برنامه نویسی است، ما در ادامه از رویکرد «بالا – پایین»، تبعیت می‌نمائیم. فراموش نکنید که رویکرد فوق، امکان مشاهده مجازی از هر مسئله برنامه‌نویسی را فراهم خواهد نمود. مراحل پنج گانه هر برنامه را صرف‌نظر از میزان پیچیدگی آن، می‌توان به پنج مرحله اساسی تجزیه کرد :

- مقداردهی اولیه
- ورودی
- پردازش
- خروجی
- پاکسازی

در ادامه به بررسی هریک از مراحل فوق، خواهیم پرداخت.

مرحله مقداردهی اولیه : اولین مرحله‌ای است که می‌بایست در زمان طراحی یک برنامه در رابطه با آن فکر کرد. مرحله فوق، شامل تمامی‌عملیات مورد نیازی است که برنامه می‌بایست قبل از برقراری ارتباط با کاربر، انجام دهد. در ابتدا ممکن است این موضوع که عملیاتی را قبل از برقراری ارتباط با کاربر می‌بایست انجام داد، تا اندازه‌ای عجیب به‌نظر رسد ولی

احتمالاً برنامه‌های زیادی را مشاهده نموده اید که در این راستا عملیات مشابهی را انجام می‌دهند. مثلاً، در زمان استفاده از برنامه‌هایی نظیر Word، Excel و یا برنامه‌های مشابه دیگر، با چنین مواردی برخورد نموده‌ایم. مثلاً با انتخاب گزینه منو File، می‌توان لیستی از فایل‌هایی را که با آنها کار کرده ایم در بخش انتهایی منوی فوق، مشاهده کرد. (مشاهده آخرین فایل‌های استفاده شده در یک برنامه خاص، با استفاده از جادو! میسر نشده است). برنامه مورد نظر شاید، لیست فایل‌های اخیر را از دیسک خوانده و آنها را به لیست مربوطه در منوی File، اضافه کرده باشد. با توجه به اینکه لیست فایل‌های فوق، می‌بایست قبل از اینکه برنامه هر چیز دیگر را برای کاربر نمایش دهد، خوانده و نمایش داده شوند، می‌توان انجام عملیات فوق را نمونه‌ای از مرحله مقداردهی اولیه، در نظر گرفت.

یکی دیگر از عملیات متداول که به این مرحله مرتبط می‌باشد، خواندن فایل‌های Setup است. چنین فایل‌هایی ممکن است حاوی اطلاعاتی در رابطه با نام مسیرهایی باشند که بانک‌های اطلاعاتی خاصی و یا فایل‌های ذخیره شده دیگری را بر روی دیسک مشخص می‌نمایند. با توجه به نوع برنامه‌ای که اجرا می‌گردد، فایل‌های Setup می‌توانند شامل اطلاعاتی در رابطه با فونت‌های نمایش، نام و محل چاپگر، رنگ‌های زمینه و رویه، وضوح تصویر صفحه نمایشگر و اطلاعات مشابهی دیگر باشند. سایر برنامه‌ها ممکن است مستلزم خواندن اطلاعاتی در رابطه با اتصالات شبکه، مجوزهای امنیتی و دستیابی به اینترنت، رمزهای عبور و سایر اطلاعات حساس دیگر باشند. در چنین مواردی فایل‌های Setup دارای نقشی مهم خواهند بود.

در زمان طراحی یک برنامه، همواره می‌بایست در رابطه با اطلاعاتی که یک برنامه قبل آغاز خدمات و عملیات خود به آنها نیازمند است، اندیشید و برای آنان در مرحله مقداردهی اولیه راهکار مناسب را انتخاب کرد. مرحله مقداردهی اولیه احتمالاً جایی است که می‌بایست از طریق آن اقدام به ارائه راهکار مناسب در جهت پاسخ به نیازهای فوق، کرد.

مرحله ورودی: در حقیقت چیزی است که انتظار دارید باشد! مرحله فوق، شامل اخذ (جمع آوری) هر آن چیزی است که یک برنامه برای انجام فعالیت‌های خود به آنها نیاز خواهد داشت. دراکثر موارد، اگر استنباط مناسبی از عملیاتی را که یک برنامه قصد انجام آن را دارد، حاصل گردد، مشخص نمودن لیستی از ورودی‌ها، کاری ساده خواهد بود. مثلاً اگر شما قصد نوشتن یک برنامه وام را دارید، می‌دانید که می‌بایست از کاربر، میزان وام درخواستی، بهره موردنظر و مدت زمان وام، درخواست گردد. در حالات دیگر، لازم است در رابطه با نوع ورودی‌هایی که می‌بایست از کاربر اخذ گردد، بررسی لازم و مبتنی براندیشه را دنبال نمود. مثلاً در صورتی که قصد نوشتن یک برنامه دفترچه آدرس را دارید، آیا می‌خواهید نام فایل حاوی دفترچه تلفن و محل ذخیره فایل مربوطه را در هر مرتبه که برنامه اجرا می‌گردد، از کاربر درخواست نمائید؟ به‌عبارت دیگر برخی از مراحل ورودی می‌توانند و شاید می‌بایست، توسط مرحله مقداردهی

انجام شوند. ماهیت واقعی میزان اطلاعاتی که می‌توان آنها را در مرحله مقداردهی خواند، بستگی به رفتار برنامه دارد. به‌عنوان یک قانون عمومی می‌توان به این مورد اشاره داشت که اکثر کاربران تمایل دارند که اطلاعات تکراری در یک فایل Setup و یا مقداردهی اولیه ذخیره گردد (در مقابل اینکه هر مرتبه که برنامه اجرا می‌گردد، مجبور به ورود اطلاعات تکراری باشند). فایل‌های Setup بسیار مناسب بوده و در هرموردی که امکان به‌خدمت گرفتن آنان منطقی به‌نظر می‌آید، می‌بایست از آنان استفاده گردد. برخی دیگر از اطلاعات اولیه دارای ماهیت خاص خود بوده و تا زمانی که کاربر آنها را تایپ ننماید، شناخته نمی‌گردند. در مثال وام اشاره شده، می‌توان از Textbox‌های متعددی به منظور اخذ اطلاعات از کاربر و استفاده از آنان در برنامه، کمک گرفت. با توجه به اینکه کاربر می‌بایست با این Textbox‌های مرتبط اطلاعات موردنیاز برنامه را وارد نماید، روشی که شما به‌منظور ارائه Textbox، Labels، Menus و سایر عناصر برنامه، استفاده می‌نمایید، یکی از بخش‌های مهم یک برنامه یعنی رابط کاربر (user interface) را مشخص خواهد کرد. فراموش نکنیم یکی از عوامل موفقیت هر نرم‌افزار، بخش رابط کاربر آن است. طراحی مناسب بخش فوق، امروزه به‌عنوان تخصصی خاص در طراحی و پیاده سازی نرم‌افزار مطرح و دارای جایگاه خاص خود است.

مرحله پردازش: شامل انجام عملیات برروی ورودی (ورودی‌ها)، به‌منظور تولید نتایج مورد نظر برای برنامه است. در مثال وام، برنامه پس از دریافت ورودی‌های مورد نظر (میزان وام، درصد بهره و زمان وام) آنها را از طریق یک معادله مالی به‌یکدیگر مرتبط و پس از حل معادله، نتیجه مورد نظر حاصل خواهد شد(میزان پرداخت ماهانه). به‌عبارت دیگر، مرحله پردازش قادر به دریافت ورودی، برخورد با آنها و تولید پاسخ مناسب به مسئله است. توجه داشته باشید که مرحله پردازش همواره باعث نمایش چیزی بر روی نمایشگر نخواهد شد. هدف، عمل (عملیات) برروی داده (داده‌ها) به‌منظور تولید یک نتیجه (نتایج) است. در این رابطه هیچگونه استثنایی وجود ندارد. در صورتی‌که در برنامه ای از قبل می‌دانیم که مرحله پردازش زمان زیادی طول خواهد کشید، منطقی است که فیدبک‌های لازم به‌منظور آگاهی کاربر از میزان و درصد انجام پردازش (پردازش‌ها) در اختیار وی گذاشته شود (در زمانی‌که برنامه در حال اجراء است). در این رابطه می‌توان از روش‌های متعددی استفاده کرد. (ارائه یک میله پیشرفت، براورد زمان تقریبی به‌منظور اتمام عملیات).

مرحله خروجی: مرحله فوق، پاسخ (پاسخ‌های) مناسب و مورد انتظار را به کاربران مبنی بر حل مسئله مورد نظر، ارائه می‌نماید. تعداد زیادی ازبرنامه‌ها، پاسخ نهایی (نتیجه) خود را از طریق یک Textbox، نمایش و در اختیار کاربر قرار می‌دهند، مثلاً اگر برنامه‌ای نوشته شده است که قصد محاسبه و نمایش میزان پرداخت ماهیانه یک وام دریافتی را داشته باشد، می‌توان نتیجه به‌دست آمده (پرداخت ماهانه) را از طریق یک Textbox، ارائه کرد تا پاسخی مناسب در ارتباط با مرحله خروجی یک برنامه، داده شده باشد. سایر برنامه‌ها ممکن است دارای وضعیتی به‌مراتب پیچیده‌تر باشند. مثلاً

می‌توان برنامه‌ای را در نظر گرفت که نام، آدرس، شماره تلفن و سایر اقلام اطلاعاتی را از بانک اطلاعاتی خوانده و در ادامه آنها را بر روی صفحه نمایشگر، نشان دهد. برنامه‌هایی این‌چنین، نیازمند شکل مناسب‌تری از نمایش خروجی بوده و نمی‌توان با استفاده از چند Textbox به خواسته خود دست یافت (ارائه یک خروجی مطلوب و انعطاف‌پذیر) در این‌گونه موارد می‌بایست از راهکارهای مناسب‌تری استفاده گردد. مثلاً می‌توان از جداول خاصی به‌منظور نمایش اطلاعات مورد نظر استفاده کرد.(استفاده از grid و یا List box که برنامه در صورت ضرورت آن را تکمیل نماید). نکته مهمی‌که می‌بایست در رابطه با مرحله خروجی رعایت گردد، آگاهی از این موضوع است که با توجه به نمایش نتایج خروجی برای کاربر، بخش فوق را می‌توان جزیی از بخش رابط کاربر یک نرم‌افزار در نظر گرفت. در زمان ورود اطلاعات (مرحله ورودی) از عناصر متفاوتی به‌منظور اخذ اطلاعات توسط کاربر در بخش رابط استفاده می‌گردد، در مرحله خروجی، بخش رابط کاربر با کاربر به‌گونه‌ای دیگر مرتبط خواهد شد (ارتباطی به‌مراتب غیرفعال‌تر نسبت به مرحله ورود اطلاعات).

مرحله پاکسازی (Cleanup): به‌منظور خاتمه بخشیدن مؤدبانه یک برنامه، پس از تکمیل عملیات مربوطه است. می‌توان این مرحله را به‌عنوان مکمل مرحله مقداردهی اولیه درنظر گرفت. با اینکه تعداد زیادی از برنامه‌های ساده قادرند به‌سادگی و بدون انجام عملیات تکمیلی توسط برنامه‌نویس، خاتمه یابند، ولی برنامه‌های پیچیده زیادی نیازمند برخی کمک‌ها در این زمینه می‌باشند. مثلاً اگر برنامه‌ای یک فایل Setup را به‌منظور مقداردهی برخی از متغیرها در مرحله مقداردهی اولیه، خوانده باشد، مرحله پاکسازی می‌تواند شامل بهنگام‌سازی آن دسته از متغیرهای موجود در فایل Setup باشد که نشان دهنده آخرین اطلاعات کاربر است. مرحله پاکسازی، اغلب شامل بستن فایل‌ها (فایل‌های Setup و بانک اطلاعاتی) است. برخی برنامه‌ها میزان استفاده از برنامه توسط کاربران را ثبت و اطلاعات مربوطه را در مکان‌هایی که Log file نامیده می‌شوند، ذخیره می‌نمایند (ثبت مشخصات افرادی که برنامه را اجرا نموده‌اند به‌همراه سایر اطلاعات مرتبط نظیر تاریخ و زمان آغاز و توقف برنامه، در خیلی از برنامه‌ها به امری ضروری تبدیل شده است). یکی دیگر از انواع فایل‌های Log به‌فایل‌های ثبت خطا برمی‌گردد (error log file). هدف این نوع از فایل‌ها، ثبت اطلاعاتی در رابطه با هر نوع خطایی است که ممکن است در مدت زمان اجرای یک برنامه، محقق گردد. برنامه‌نویسان با استفاده از محتویات این نوع فایل‌ها، قادر به اشکال‌زدایی برنامه خواهند بود.

عملیات واقعی و موردنظری که می‌بایست در مرحله پاکسازی، انجام گردد، به نیازهای یک برنامه بستگی خواهد داشت. معمولاً اگر در برخی برنامه‌ها عملیات خاصی را در مرحله مقدار دهی اولیه انجام دهیم، می‌بایست برخی از عملیات متناظر با آنان را در مرحله پاکسازی انجام داد. باز نمودن و بستن فایل‌های مورد نیاز در یک برنامه، نمونه‌ای

متداول از دو مرحله فوق می‌باشد.

آیا هر برنامه شامل پنج مرحله گفته شده است؟

در پاسخ به سؤال فوق می‌بایست با صراحت پاسخ منفی داده شود. در این راستا، برنامه‌های متعددی وجود دارد که مثلاً به مراحل مقداردهی اولیه و یا پاکسازی، نیاز نخواهند داشت. مراحل مقداردهی اولیه و پاکسازی در مرحله طراحی برنامه‌های پیچیده مورد توجه جدی قرار خواهند گرفت. به موازات افزایش تجربه در نوشتن برنامه، شناخت مناسبی در این رابطه به‌وجود می‌آید (کدام برنامه به تمام مراحل پنج‌گانه نیاز و کدامیک نیاز ندارند). طراحان می‌بایست همواره یک مسئله برنامه‌نویسی را با فرض وجود پنج مرحله یاد شده، دنبال نمایند. قطعاً حذف یک مرحله در زمان طراحی به‌مراتب ساده‌تر از نادیده گرفتن اولیه آن خواهد بود.

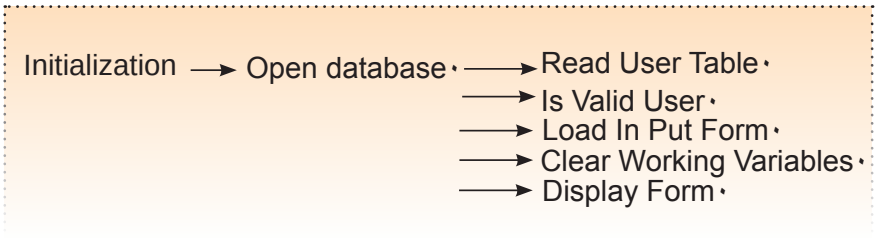
پالایش یک طرفه (Sideways Refinement): همان‌گونه که قبلاً اشاره گردید، ما علاقه‌مند به طراحی بالا به پایین می‌باشیم. (الگوریتم ماکرو به‌عنوان یک نقطه شروع در فرایند طراحی برنامه). پس از انتخاب رویکرد فوق، می‌بایست شناخت مناسبی نسبت به مسئله‌ای که قصد حل آن وجود دارد، ایجاد گردد تا رسیدن به سطح میکرو(ارائه الگوریتم‌های میکرو) به‌منظور حل مسئله مورد نظر راه زیادی را در پیش خواهیم داشت. به‌موازات حرکت از سطح مرور کلی برنامه به خصوصیات و ویژگی‌های یک برنامه، می‌بایست دانش خود را نسبت به جزئیات مربوطه افزایش داد.

از پنج مرحله گفته شده، می‌توان به‌منظور نقطه شروع دید ماکرو خود در زمان فرایند طراحی استفاده کرد. درادامه، می‌توان هر یک از مراحل را به‌دقت بررسی کرد تا جزئیات بیشتری در رابطه با مرحله مورد نظر، مشخص گردد (استخراج جزئیات لازم در رابطه با تحقق هر مرحله). فرایند فوق، «پالایش یک طرفه»، نامیده می‌شود. در ادامه، به‌منظور شناخت مناسب فرایند پالایش یک طرفه، به بررسی یک نمونه می‌پردازیم.

فرض کنید، کاربری دارای یک فایل بانک اطلاعاتی است که در آن تمام قرار ملاقات‌های وی، ذخیره شده‌اند. قرار ملاقات‌ها در بانک اطلاعاتی با نظم و ترتیب خاص (تاریخ قرار ملاقات) ذخیره شده‌اند. کاربر، می‌خواهد قادر به مشاهده قرار ملاقات‌های خود بر اساس حروف الفبایی و براساس نام خانوادگی اشخاص مورد نظری که قصد ملاقات با وی را دارند، باشد. چگونه می‌توان از پالایش یک طرفه، به‌منظور طراحی یک راه‌حل استفاده کرد؟

پالایش یک طرفه مرحله مقداردهی اولیه: می‌دانیم که کاربر دارای یک بانک اطلاعاتی شامل قرار ملاقات‌ها می‌باشد. ما همچنین می‌دانیم که کاربر می‌خواهد لیستی از قرار ملاقات‌های خود را به‌صورت مرتب شده و براساس نام‌خانوادگی مشاهده نماید. موارد فوق، دید ماکروی ما از الگوریتم است. بنابراین، در مرحله مقداردهی اولیه چه عملیاتی می‌بایست انجام داد؟ واضح است که ما نیازمند باز نمودن بانک اطلاعاتی قرار ملاقات‌ها می‌باشیم. ما همچنین

نیازمند یک فرم (مثلاً یک فرم مبتنی بر VB.NET و یا فرم وب) به‌منظور نمایش نتایج پس از مرتب‌سازی قرار ملاقات‌ها، خواهیم بود. (فرض می‌شود از مکان بانک اطلاعاتی بر روی شبکه آگاهی داریم، و می‌توان نام و رمز عبور کاربر را از بانک اطلاعاتی مربوطه به محض آغاز اجرای برنامه توسط کاربر، مشخص کرد). با استفاده از اطلاعات فوق، اولین «پالایش یک طرفه »، به‌صورت زیر خواهد بود :



همان‌گونه که در شکل فوق، مشاهده می‌گردد به‌موازات حرکت از سمت چپ به سمت راست، جزئیات مربوطه افزایش خواهد یافت. شکل فوق، پالایش یک طرفه، لیستی از برنامه‌های جانبی و توابع مورد نیاز به‌منظور انجام فعالیت‌های مربوطه در مرحله مقداردهی اولیه را نشان می‌دهد. هر روتین کوچک، مسئول انجام عملیاتی خاص خواهد بود.

شبه کد (Pseudo Code)

عملیات پالایش را می‌توان در رابطه با هر مرحله با استفاده از «شبه – کد»، دنبال کرد. شبه کد، الگوریتمی‌برای بیان عملیاتی است که می‌بایست توسط یک روتین محقق گردد. در این راستا از یک گرامر مشابه انگلیسی، استفاده می‌گردد. مثلاً شبه کد، روتین IsValid User به‌صورت زیر خواهد بود:

شبه کد روتین Is valid User
Is Valid User ()
If Current Username Not in Valid User List
Display Invalid User Error Message
Terminate Program
Else
Return Valid User ID Number
End

شبه کد، عملیاتی را که یک روتین می‌بایست انجام دهد، بدون اتکا به گرامر یک زبان برنامه‌نویسی خاص، تشریح می نماید. شبه کد، زبانی مبتنی بر گرامری خاص نبوده و الگوریتمی‌از عملیات مورد نظر که می‌بایست توسط یک روتین انجام شود را مشخص می‌نماید. مزیت شبه کد، شباهت زیاد آن به زبان انگلیسی است و می‌توان آن را با افرادی که برنامه نویس نبوده و به‌نوعی در فاز طراحی صاحب نظر می‌باشند، به اشتراک گذاشت تا صحت استنباطات حاصل شده تأیید و یا اصلاح گردد.(در فاز طراحی می‌بایست یک ارتباط مستمر با کاربران صاحب نظر برقرارگردد، ما قرار است مسئله آنان را حل نماییم نه مسئله خود را و یا نمی‌خواهیم مسئله‌ای دیگر را بر حجم مسائل آنان اضافه نماییم!) بدین ترتیب، امکان تشخیص خطا و اعمال تغییرات لازم در خصوص برخورد با خطاهای احتمالی در ابتدا فراهم می گردد (یکی از اصول مهندسی نرم‌افزار در این رابطه به این موضوع اشاره می‌نماید که به هر میزان که زمان کشف یک خطا در چرخه حیات یک برنامه سریع‌تر باشد، هزینه برخورد با خطا کاهش خواهد یافت).

پس از آگاهی از اهداف ارائه شده در شبه کد، می‌توان به‌سادگی اقدام به ترجمه شبه کد مربوطه به کدهای برنامه نویسی با استفاده از زبان مورد نظر نمود. فراموش نکنیم که طراحی خوب، همواره پیاده‌سازی ساده‌تر برنامه‌ها را به‌دنبال خواهد شد.

۹-۵

مستندات نرم‌افزار

مستندات نرم‌افزار، به هر مطلبی که به اطلاعاتی درباره ایجاد عملیات یا استفاده از نرم‌افزار اشاره دارد، گفته می‌شود و بایستی حداقل شامل موارد زیر باشد:

- توصیفی از نیازمندی‌ها و اهداف درخواست کننده نرم‌افزار
- امکان‌سنجی‌ها و بررسی نیازمندی‌های نرم‌افزار
- توصیفی از معماری سیستم که براساس آن تصمیم‌گیری می‌شود آیا سیستم پیاده‌سازی گردد یا خیر؟
- تعیین معماری و چارچوب سیستم، شامل: توصیفی از روش طراحی، مراحل طراحی، عملکرد برنامه‌های سیستم و داده‌هایی که بین قسمت‌های مختلف برنامه‌ها رد و بدل می‌شوند.
- لیست برنامه‌ها به همراه توضیحات لازم در قسمت‌هایی که کد برنامه پیچیده است.
- مستندات ارزیابی سیستم که شامل نتایج آزمون‌های انجام شده برروی سیستم می‌باشد.

– تعیین بودجه، زمان و نیروی انسانی مورد نیاز برای تولید نرم‌افزار

– راهنمای استفاده از سیستم، که به کاربر می‌گوید چگونه سیستم را نصب نماید و آن را به‌کار گیرد.

– مستندات نگهداری سیستم که مشکلات شناخته شده نرم‌افزاری و سخت‌افزاری را توضیح می‌دهد و نکاتی را که در مراحل طراحی سیستم جهت توسعه سیستم در نظر گرفته شده است، شرح می‌دهد.

نکته مهمی که باید در طول حیات یک نرم‌افزار در نظر داشت این است که هنگامی که تغییراتی در سیستم ایجاد می‌شود تمام مستندات مربوط به آن بایستی بهنگام گردد تا باز هم از طریق سازگاری مستندات با نرم‌افزار، کارایی لازم در استفاده از نرم‌افزار وجود داشته باشد. گارگ و اسکاچی پیشنهاد کردند به همراه مستندات از نرم‌افزاری استفاده شود که روابط بین مستندات در آن ثبت شده باشد و زمانی که تغییراتی در هریک از مستندات داده می‌شود، مستندات وابسته دیگر که بایستی بهنگام شوند در آن مشخص گردد. برای تولید مستندات بهینه بایستی به مراحل تولید نرم‌افزار توجه کرد. معمولاً تولید نرم‌افزار در روش‌های متداول شامل مراحل زیر است:

مرحله UR– بیان نیازهای کاربر یا User Requirements

مرحله SR– بیان نیازهای نرم‌افزار یا Software Requirements

مرحله AD– طراحی معماری یا Architectural Design

مرحله DD– طراحی تفضیلی و تولید برنامه یا Detailed Design

مرحله TR– انتقال و واگذاری نرم‌افزار برای بهره‌برداری یا Transfer of the Software

مرحله OM– بهره‌برداری و نگهداری سیستم یا Operation & Maintenance

در ادامه توضیحات هریک از مراحل فوق به‌صورت خلاصه بیان می‌گردد.

مرحله بیان نیازهای کاربر (UR): در این مرحله کلیه نیازهای کاربر توسط کاربر بیان می‌گردد. در روش‌های جدید سیستم‌های اطلاعاتی به آن مهندسی خواسته‌ها گفته می‌شود.

مرحله نیازهای نرم‌افزار (SR): در این مرحله تمام خواسته‌های کاربر از نظر تکنیکی و اقتصادی بررسی می‌گردد و تصمیمی مبنی بر اینکه آیا سیستم جدیدی پیاده‌سازی شود یا خیر، گرفته می‌شود.

مرحله طراحی معماری (AD): از طراحی معماری می‌توان به‌عنوان مهمترین مرحله از مراحل تولید نرم‌افزار نام برد که می‌بایست در آن تمام نیازهای نرم‌افزار را به‌صورت ساخت یافته و در قالب نمودارهای گردش اطلاعات نمایش داد. برای طراحی یک نرم‌افزار روش‌های مختلفی از جمله طراحی شیءگرا، طراحی تابعی، طراحی بلادرنگ، طراحی Form Base و طراحی Menu Driven وجود دارد که بسته به نظر طراح سیستم یکی از روش‌های بیان شده، برای

طراحی درنظر گرفته می‌شود.

مرحله طراحی تفضیلی و تولید برنامه (DD): در مرحله طراحی تفضیلی، طراحی‌های انجام شده در مرحله طراحی معماری بصورت نهایی درمی‌آید. هدف این مرحله، طراحی تکنیکی سیستم جدید با جزئیات کافی است. خروجی‌های این مرحله عبارتند از سند طراحی تفضیلی، برنامه، سند راهنمای کاربر (SUM) و مستندات آزمون‌های انجام شده برروی سیستم. پس از برنامه‌نویسی، رفع خطاها و بازبینی برنامه‌ها، آزمون‌های مختلفی انجام می‌شود که بایستی مستنداتی برای هر آزمون تهیه گردد. این آزمون‌ها عبارتند از: آزمون واحد، آزمون یکپارچگی، آزمون سیستم، آزمون پذیرش. تولیدکنندگان نرم‌افزاری بایستی طرح آزمون واحد را در مرحله طراحی تفضیلی، طرح آزمون یکپارچگی را در مرحله طراحی معماری و طرح آزمون سیستم را در مرحله بیان نیازهای نرم‌افزار تهیه کنند. همچنین طرح آزمون پذیرش، توسط کاربر بایستی در مرحله بیان نیازهای کاربر توسط درخواست کنندگان سیستم تهیه و مستند گردد.

مرحله انتقال و واگذاری نرم‌افزار برای بهره‌برداری (TR): در این مرحله نرم‌افزار در محیط عملیاتی نصب می‌گردد تا قابلیت‌هایی که برای نرم‌افزار در سند نیازهای کاربر ذکر شده، به نمایش گذارده شود. مرحله انتقال نرم‌افزار با پذیرش موقت نرم‌افزار آغاز و با شروع مرحله بهره‌برداری خاتمه می‌یابد.

مرحله بهره‌برداری و نگهداری سیستم (OM): این مرحله که تولیدکنندگان نرم‌افزار نیز در آن مشارکت دارند تا زمان پذیرش نهایی نرم‌افزار ادامه می‌یابد و عملکرد سیستم برای مدتی مورد بررسی قرار می‌گیرد تا از تطابق عملکرد واقعی سیستم با نیازهای کاربر اطمینان حاصل گردد و امکانات توسعه سیستم و قابلیت‌های جدید در آینده بررسی شود.

با توجه به مراحلی که در بالا اشاره شد، در هر مرحله مستنداتی که تولید می‌شوند عبارتند از:

- مرحله بیان نیازهای کاربر (UR): سند نیازهای کاربر (URD)
- مرحله نیازهای نرم‌افزار (SR): سند نیازهای نرم‌افزار (SRD)
- مرحله طراحی معماری (AD): سند طراحی معماری (ADD)
- مرحله طراحی تفضیلی و تولید برنامه (DD): سند طراحی تفضیلی (DDD)
- مرحله انتقال و واگذاری نرم‌افزار برای بهره‌برداری (TR): سند انتقال و واگذاری نرم‌افزار (STD)
- مرحله بهره‌برداری و نگهداری سیستم (OM): سند نگهداری سیستم (SMD)

با توجه به آنچه در مراجع مختلف به‌خصوص استانداردهای IEEE ذکر شده است، هریک از مستندات فوق‌الذکر شامل موارد زیادی است که مندرجات این مستندات به‌طور خلاصه در جدول شماره ۱-۹ ارائه شده است.

جدول شماره ۱-۹ مستندات مراحل تولید نرم‌افزار براساس استانداردهای بین‌المللی IEEE

نام سند	مندرجات
سند نیازهای کاربر	تعیین محیط عملیاتی، تشخیص نیازهای کاربر
سند نیازهای نرم‌افزار	هدف از تهیه سند، دامنه نرم‌افزار، تعاریف، مفاهیم، سرنام‌ها، مخفف‌ها، مأخذ و مراجع، مرور اجمالی سند، شرح ارتباط نرم‌افزار با پروژه‌های قبلی، جاری و آتی، شرح وظایف عمده نرم‌افزار، شرح این‌که نرم‌افزار در کجا و توسط چه کسی مورد استفاده قرار می‌گیرد و چه کسی آن را راهبری می‌کند، سخت افزار و سیستم عامل مورد نیاز نرم‌افزار، شرح ارتباط نرم‌افزار با سایر سیستم‌ها، شرح محدودیت‌های درخواستی از طرف کاربر، شرح مدل منطقی سیستم با استفاده از یک روش تجزیه و تحلیل شناخته شده، شرح نیازهای ویژهٔ نرم‌افزار، ارائهٔ یک ماتریس برای ردیابی نیازهای نرم‌افزار در قبال نیازهای کاربر
سند طراحی معماری	هدف از تهیه سند، دامنه نرم‌افزار، تعاریف، مفاهیم، سرنام‌ها، مخفف‌ها، مأخذ و مراجع، مرور اجمالی سند، مرور اجمالی نرم‌افزار، روش طراحی، شرح سیستم و طراحی آن همراه با نمودارهای لازم، مشخصات طراحی، امکان سنجی و براورد منابع، ارائهٔ یک جدول ارجاعات متقابل بین مؤلفه‌های طراحی شده و نیازهای نرم‌افزار
سند طراحی تفضیلی	هدف از تهیه سند، دامنه نرم‌افزار، تعاریف، مفاهیم، سرنام‌ها، مخفف‌ها، مأخذ و مراجع، مرور اجمالی سند، استانداردهای طراحی و مستندسازی و برنامه‌سازی، قراردادهای نام‌گذاری برنامه‌ها و فایل‌ها، ابزار تولید نرم‌افزار، مشخصات طراحی و مؤلفه‌ها، لیست متن برنامه‌ها، ارائهٔ یک جدول ارجاعات متقابل بین مؤلفه‌ها و نیازهای نرم‌افزار
راهنمای کاربر نرم‌افزار	۱- مقدمه شامل: شرح هدف سند، هدف نرم‌افزار، نحوهٔ استفاده از راهنما، مستندات وابسته و قراردادها، ۲- شرح نرم‌افزار ۳- بخش دستورالعمل‌ها شامل: احتیاط و هشدارها، روال‌های برپاسازی و راه‌اندازی سیستم، خطاهای احتمالی، دلایل آنها و چگونگی رفع خطاهای پیش آمده ۴- بخش مراجعه شامل ارائهٔ مثال‌هایی از عملیات نرم‌افزار، شرح روال‌های ترمیم سیستم، فهرست کلیهٔ پیغام‌های خطا، واژه‌نامه و نمایه

سند آزمون‌های نرم‌افزار	خلاصه‌ای از عناصر نرم‌افزار، فهرست عناصری که بایستی مورد آزمون قرارگیرند، شرح مشخصات داده‌های ورودی برای انجام آزمون و نتایج مورد انتظار، طرح کلی انجام آزمون، شرح معیارهای قبول یا رد یک آزمون، فهرست عناصری که بایستی قبل و بعد از آزمون تحویل داده شوند، شرح ویژگی‌های لازم برای محیط آزمون، شرح مشخصات کسانی که اجازه می‌دهند آزمون انجام شود، آزمون را انجام می‌دهند، نتایج آزمون را بررسی می‌کنند و کامل بودن آن را تأیید می‌نمایند، مشخصات نیروی انسانی لازم برای انجام آزمون‌ها همراه با سطح مهارت مورد نیاز، خلاصه‌ای از زمان انجام فعالیت‌های آزمون
سند انتقال نرم‌افزار	هدف از تهیه سند، دامنه نرم‌افزار، تعاریف، مفاهیم، سرنام‌ها، مخفف‌ها، مأخذ و مراجع، مرور اجمالی سند، شرح چگونگی نصب و راه‌اندازی نرم‌افزار، شرح چگونگی ساخت نرم‌افزار متن اصلی برنامه، فهرست عناصر پیکربندی قابل تحویل، خلاصهٔ گزارش آزمون پذیرش و نتیجهٔ آن، فهرست گزارش مشکلات نرم‌افزار در طی مرحلهٔ انتقال نرم‌افزار، فهرست در خواست‌های تغییر و اصلاح نرم‌افزار در طی مرحلهٔ انتقال نرم‌افزار
سند نگهداری سیستم	شرح محصول تولید شده، شرح برنامه ریزی اصلی و آنچه که عملاً اتفاق افتاده است، ارائهٔ حجم برنامه براورد شده در پایان مرحله طراحی معماری، ارائهٔ حجم برنامه واقعی در پایان مرحله انتقال نرم‌افزار، فهرست مستندات پروژه همراه با تعداد کلمات و صفحات، حجم کار براورده شده و واقعی، ارائهٔ براوردهای بهره‌وری به عنوان مثال تعداد خط برنامه نوشته شده در روز، خلاصهٔ فعالیت‌های تضمین کیفیت، فهرست منابع کامپیوتری اعلام شده در پایان مرحله طراحی معماری و منابع استفاده شده در پایان مرحله انتقال نرم‌افزار

برای مطالعه

مندرجات سند نیازهای کاربر (URD)

این سند همیشه قبل از آغاز یک پروژه نرم‌افزاری تهیه می‌گردد که در آن کلیه نیازهای کاربر گنجانده می‌شود و لازم است حداقل حاوی مطالب زیر باشد:

- ارزیابی درخواست کاربر با در نظر گرفتن حیطه کاری و امکان‌پذیری آن
- ارائه یک نمودار مفهومی که نشان‌دهنده درک اولیه از نیازهای کاربر است و در آن ورودی‌ها و خروجی‌های سیستم مشخص گردیده است.
- تخمین و ارزیابی بسیار کلی در زمینه هزینه، زمان، نیروی انسانی و تجهیزات مورد نیاز و منافعی که در صورت قبول

درخواست کاربر، پیش‌بینی می‌شود.

- مصاحبه‌ها، یادداشت‌ها و نتایج به‌دست آمده در خلال مرحله UR (مهندسی خواسته‌ها)
- شرحی از کلیه محدودیت‌هایی که کاربر برای هر راه‌حل پیشنهادی، تحمیل خواهد نمود.
- شرح قابلیت‌ها و عملکردهای اصلی نرم‌افزار
- شرح مسائل و مشکلات موجود
- توصیفی از خواسته‌های مربوط به توسعه و افزایش قابلیت نرم‌افزار

در نهایت پس از بررسی نیازهای کاربر کمیت‌هایی جهت تصمیم‌گیری در مورد قبول یا رد درخواست کاربر تشکیل می‌گردد و نتایج این تصمیم‌گیری در سند نیازهای کاربر ثبت می‌گردد. ضمناً کنترل و نظارت بر تغییرات این سند بر عهده کاربر می‌باشد.

سند نیازهای نرم‌افزار (SRD)

این سند دربرگیرنده نتایج حاصل از امکان‌سنجی‌ها، تجزیه و تحلیل‌هایی است که پیرامون سیستم نرم‌افزاری انجام گرفته است و بایستی حداقل حاوی مطالب زیر باشد:

- تخمینی از هزینه‌ها و منابع مورد نیاز شامل منابع سخت افزاری، نرم‌افزاری، نیروی انسانی، تخصص‌های مورد نیاز و تخمین نفر- ساعت هر تخصص (حداکثر تا ۲۰٪ خطا)
- شرح محدودیت‌های تعیین شده توسط مدیر پروژه و درخواست کننده سیستم نرم‌افزاری
- توصیفی از سیستم نرم‌افزاری موجود در قالب نمودار (مثلاً ترسیم نمودار جریان داده برای سیستم موجود)
- ارائه یک یا چند راه‌حل برای سیستم پیشنهادی
- بیان نیازهای سیستم پیشنهادی
- شرح عملیات و وظایفی که سیستم پیشنهادی انجام می‌دهد.
- ایجاد یک جدول ارجاع متقابل که در آن برای هر نیاز، نرم‌افزار ارجاعی به خواسته‌های کاربر وجود دارد.

سند طراحی معماری (ADD)

بسته به نوع طراحی، مستندات این مرحله می‌توانند در هر شکل و نوع اندکی متفاوت باشد. مثلاً در طراحی شیءگرا بهتر است نمودار سلسله مراتبی سیستم رسم و مستند گردد تا ارتباط بین اشیا مشخص شود و ترسیم یک شبکه کاری

که نشان می‌دهد چه اشیایی از سرویس‌های اشیا دیگر استفاده می‌کند، نیز مناسب است. اما در طراحی Form Base که سیستم شامل تعداد زیادی فرم می‌باشد، بایستی اطلاعات این فرم‌ها مستند گردد تا نشان دهد در هر فرم مجموعه‌های اطلاعاتی و فرم‌های قبلی و بعدی کدامند. همچنین فایل‌های در برگیرنده این فرم‌ها و برنامه‌هایی که از طریق فرم‌ها احضار می‌شوند، مشخص گردند. ولی به‌طور کلی سند طراحی معماری می‌بایست شامل موارد مشترک زیر باشد:

- تأییدیه کاربر برای طراحی سیستم
- تأیید و تعهد طراحان سیستم مبنی براینکه طراحی طبق جدول زمان‌بندی تعیین شده، انجام خواهد شد.
- برنامه زمان‌بندی برای انجام پروژه
- استانداردها و قراردادهای پروژه که شامل:

۱- استانداردهای طراحی که بایستی روش طراحی شرح داده شود و یا به یک روش استاندارد ارجاع داده شود.

۲- استانداردهای مستندسازی شامل شرح، شیوه و ابزار مستندسازی

۳- انتخاب زبان برنامه‌نویسی

۴- استانداردهای برنامه‌نویسی و قواعد نام‌گذاری فایل‌ها، جداول، فرم‌ها و برنامه‌ها

۵- ابزارهای تولید نرم‌افزار

- روش مورد استفاده در طراحی سیستم
- توصیفی از طراحی سیستم به همراه نمودارهای جریان داده سیستم پیشنهادی
- طراحی مقدماتی ساختار فایل‌ها، پایگاه داده‌ها و ساختمان داده ورودی و خروجی‌ها
- شرح وظایف سیستم
- امکان‌سنجی و براورد هزینه‌ها، منابع سخت‌افزاری و نیروی انسانی مورد نیاز (حداکثر با ۱۰٪ خطا)
- ارائه یک جدول ارجاع متقابل از مؤلفه‌های طراحی به نیازهای نرم‌افزار

سند طراحی تفضیلی (DDD)

در سند طراحی تفضیلی حداقل بایستی موارد زیر گنجانده شود:

- مشخصات طراحی که شامل:

۱- ترسیم نمودارهای جریان داده سیستم پیشنهادی که کلیه نمودارهای ترسیم شده در مرحله طراحی معماری در این مرحله با جزئیات کامل شکسته می‌شود و بایستی در فرم‌هایی برای هر پردازه، منطق اصلی پردازش اطلاعات شرح داده می‌شود.

۲- ترسیم نمودار ساختاری سیستم (SSC) که در آن زیرسیستم‌ها و ارتباط آنها با یکدیگر تعیین می‌شود و برای زیرسیستم‌ها مشخصات کامل فایل‌ها و جداول آنها و نحوه ارتباط آنها با یکدیگر مشخص می‌گردد.

۳- ترسیم نمودار ارتباط بین (ERD) که بایستی کلیه موجودیت‌های سیستم شناسایی و ارتباط بین آنها مشخص گردد.

۴- مشخصات و شرح ورودی‌ها و خروجی‌های سیستم

- لیست کد و مشخصات کامل برنامه‌ها، شامل نام برنامه، نام فایل‌ها و جداول ایجاد شده یا استفاده‌شده در برنامه، ورودی‌ها و خروجی‌های برنامه، نام برنامه‌های فراخواننده این برنامه یا برنامه‌های فراخوانی شده در این برنامه و در صورت لزوم فلوچارت هر برنامه

- تعیین پیکربندی نهایی سخت‌افزار، نرم‌افزار و شبکه مورد نیاز سیستم

- ارائه یک جدول ارجاع متقابل بین مؤلفه‌ها و نیازهای نرم‌افزار

سند راهنمای کاربر (SUM)

یکی از خروجی‌های مرحله طراحی تفضیلی، سند راهنمای کاربرمی‌باشد. این سند بایستی حداقل شامل موارد زیر باشد:

- نحوه استفاده از این سند

- بیان کلیه قراردادهای به‌کار رفته در ساختار نحوی و نگارشی مطالب

- شرح سیستم شامل اهداف سیستم، قابلیت‌ها و محدودیت‌های سیستم، ترتیب و توالی هریک از فعالیت‌های سیستم، شرح منوها، مشخصات کامل ورودی‌ها و خروجی‌های سیستم

- بیان نحوه تأمین امنیت سیستم

- پیغام‌های خطا و دلایل آنها و راهکارهای رفع آنها

- روال‌های ترمیم سیستم بعد از از کار افتادن سیستم

- روال پشتیبانی در صورت اتفاقات غیرمترقبه

برای سهولت کار، راهنمای کاربر می‌تواند به‌عنوان بخشی از یک راهنمای حین کار باشد، تا کاربر بتواند با انتخاب کلمات کلیدی، از توضیحات و راهنمای هر بخش استفاده نماید.

مستندات آزمون

مستندات آزمون واحد، یکپارچگی، سیستم و پذیرش شامل موارد مشترک زیر است:

- نام آزمون
- تاریخ شروع و خاتمه آزمون
- مشخصات افرادی که آزمون را انجام می‌دهند و نتایج کامل بودن آزمون را تأیید می‌کنند.
- محدودیت‌های خاص آزمون
- لیست متغیرها و ورودی‌ها به هر بخش
- لیست گزارشات و خروجی‌های هر بخش
- نتیجه آزمون

سند انتقال و واگذاری نرم‌افزار برای بهره‌برداری (STD)

سند انتقال نرم‌افزار بایستی حداقل شامل موارد زیر باشد:

- تاریخ نصب و نام نصب کننده سیستم نرم‌افزار
- حداقل سخت‌افزار موجود برای نصب سیستم
- نرم‌افزارهای لازم جهت نصب سیستم
- خلاصه‌ای از گزارش آزمون پذیرش
- شرح چگونگی نصب، راه‌اندازی و اجرای سیستم نرم‌افزاری برروی کامپیوتر
- گزارشی از درخواست تغییر یا اصلاح نرم‌افزار در طی مرحله انتقال نرم‌افزار
- نوع نصب سیستم نرم‌افزاری

سند نگهداری سیستم (SMD)

در مستندات این مرحله بایستی تاریخ بررسی و دوره بررسی سیستم مشخص گردد. سپس گزارشاتی که در این دوره تهیه می‌گردد مورد تجزیه و تحلیل قرار گیرد و خطاهای احتمالی و یا درخواست‌های جدید کاربر بررسی گردد. همچنین بایستی هزینه واقعی ایجاد و توسعه سیستم با هزینه پیش‌بینی شده مقایسه گردد و هرگونه پیشنهادی که باعث بهبود وضعیت سیستم در آینده خواهد شد، مشخص گردد. سپس اعلامیه پذیرش نهایی سیستم از طرف نمایندگان کاربران، تهیه و به تولیدکنندگان نرم‌افزار تحویل داده شود و به مستندات این مرحله ضمیمه گردد.

خلاصه فصل

سیستم مجموعه‌ای است که از اجزای به هم وابسته که وابستگی حاکم بر اجزای خود کلیت جدید را احراز کرده و از نظم و سازمان خاصی پیروی می‌نماید و در جهت تحقق هدف معینی که دلیل وجودی آن است فعالیت می‌کند.

یک سیستم اطلاعات سه فعالیت عمده انجام می‌دهد. ابتدا اطلاعاتی را از منابع درون‌سازمانی یا برون‌سازمانی به‌عنوان ورودی دریافت می‌کند. سپس بر روی اطلاعات دریافت شده کارهایی انجام می‌دهد تا اطلاعات مورد نظر سیستم را تولید کند. در نهایت، اطلاعات تولید شده را در اختیار کاربر در نظر گرفته شده مثلاً یک مدیر یا یک کارمند قرار می‌دهد.

سیستم‌های اطلاعات پتانسیل ارائه سه نوع مزیت به سازمان را دارا هستند: بهبود بهره‌وری، بهبود اثربخشی و مزیت رقابتی.

تجزیه تحلیل سیستم‌ها عبارت است از شناخت جنبه‌های مختلف سیستم و آگاهی از چگونگی عملکرد اجزای تشکیل دهنده آن و بررسی نحوه و میزان ارتباط بین اجزای آن به منظور دستیابی به مبنایی جهت طرح و اجرای یک سیستم مناسب تر.

طراحی پایگاه داده و ایجاد نمودار ارتباط موجودیت‌ها (ERD) یکی از مهم‌ترین بخش‌های چرخه حیات توسعه یک نرم‌افزار است که در برخی موارد از آن به عنوان مهمترین بخش نیز نام برده می‌شود.

هر برنامه، می‌بایست دارای یک طرح و یا الگو بوده تا برنامه‌نویس براساس آن عملیات خود را دنبال نماید. از دیدگاه برنامه‌نویسان، هر برنامه نیازمند یک الگوریتم است. به‌عبارت ساده، الگوریتم، بیانه‌ای روشمند به‌منظور حل یک مسئله به‌خصوص است. از منظر برنامه‌نویسان، الگوریتم به‌منزله یک طرح کلی و یا مجموعه دستورالعمل‌هایی است که با دنبال نمودن آنان، برنامه‌ای تولید می‌گردد.

هر برنامه را صرف‌نظر از میزان پیچیدگی آن، می‌توان به پنج مرحله اساسی تجزیه کرد: مقدار دهی اولیه، ورودی، پردازش، خروجی و پاکسازی.

مستندات نرم‌افزار، به هر مطلبی که به اطلاعاتی درباره ایجاد، عملیات یا استفاده از نرم‌افزار اشاره دارد، گفته می‌شود.

خودآزمایی

۱- آنتروپی چیست و چه انواعی دارد؟

۲- مراحل روش علمی‌تجزیه و تحلیل سیستم‌ها کدام است؟

۳- با یک مثال یک سیستم کوچک را تجزیه و تحلیل نمایید.