

ارائه‌ی الگوی پیشنهادی استانداردسازی در چرخه‌ی حیات نرم‌افزار

محمد رضا گودرزی

مصطفی تمناجی

چکیده:

تاریخ دریافت: ۹۳/۱۱/۱۱

تاریخ پذیرش: ۹۳/۱۱/۲۵

چرخه‌ی حیات، یک چارچوب فرآیندی مشترک در سامانه‌های ساخته‌ی بشر از شکل‌گیری ایده تا وارهایی آن است. برای دستیابی به یک سامانه‌ی نرم‌افزاری کارآمد و اثربخش لازم است در هر یک از مراحل چرخه‌ی حیات، الزامات و مشخصه‌های کیفیتی و عملکردی تعریف و اجرا شوند. لذا در این مقاله پس از بیان مفاهیم اولیه و تشریح مراحل چرخه‌ی حیات سامانه‌های نرم‌افزاری، الزامات عمومی هر مرحله شناسایی شده است. سپس در ادامه با روش کتابخانه‌ای و توصیفی و با بهره‌گیری از تجربیات برتر و استانداردهای موجود، الگوی استانداردسازی مراحل توسعه‌ی نرم‌افزار در چرخه‌ی حیات ارائه شده است. الگوی استانداردسازی ارائه شده در این مقاله، با لحاظ کردن موارد مذکور به دنبال ارائه‌ی راهکاری برای چارچوب‌دهی به طیف گسترده از الزاماتی است که در این حوزه مطرح هستند. این الگو به توسعه‌دهندگان نرم‌افزار کمک می‌کند تا الزامات متداول مورد انتظار از آن‌ها را شناسایی کرده و فرایندهای توسعه‌ی خود را متناسب با این الزامات طرح‌ریزی و اجرا نمایند. همچنین این الگو به کارفرمایان (و کاربران نهایی) کمک می‌کند تا بتوانند نیازهای خود را در قالب ادبیات مشترک به توسعه‌دهندگان ارائه کنند. با بهره‌گیری از این الگو اطمینان حاصل می‌شود که حداقل الزامات کیفی، امنیتی و به روز بودن، در آن لحاظ شده و توسعه دهنده‌ی زیرساخت‌های لازم برای انجام این کار را در اختیار دارد و همچنین نرم‌افزار تهیه شده به‌طور مستمر در حال پشتیبانی و بروز رسانی است.

واژگان کلیدی:

چرخه‌ی حیات نرم‌افزار، استانداردسازی سامانه‌ی نرم‌افزاری، روش‌شناسی توسعه‌ی نرم‌افزار

۱) مقدمه

فرهنگ استفاده از این فناوری در بخش وسیعی از حوزه‌ها، رعایت‌نشدن حقوق تولیدکنندگان محصولات نرم‌افزار، عدم سرمایه‌گذاری مناسب برای پژوهش و تحقیق در حوزه‌ی نرم‌افزار و غیره، مطرح هستند که در این مقاله به آن‌ها پرداخته نشده و این امور، به تدریج در قالب ارائه‌ی طرح‌هایی توسط نهادهای حاکمیتی در حال اصلاح هستند [۱]. در این مقاله به چالش‌های فنی و تخصصی موضوع تولید و توسعه نرم‌افزار پرداخته خواهد شد. نکته‌ی حائز اهمیت اینکه هر تولید نرم‌افزاری در قالب تعریف پروژه بوده و لذا هر پروژه نرم‌افزاری از نظر فنون مدیریت

در حال حاضر با پیشرفت فناوری و افزایش سطح انتظارات کاربران فناوری اطلاعات^۱، مشکلات بیشتری نیز مطرح شده‌اند. حوزه‌ی فناوری اطلاعات و به‌خصوص بخش توسعه‌ی نرم‌افزار در ایران سیر تکاملی بلوغ را تجربه می‌نماید. در مسیر روبه‌رشد، مسائل و چالش‌هایی وجود دارد که برخی از آن‌ها مشکلات ذاتی صنعت تولید و توسعه‌ی نرم‌افزار بوده و برخی دیگر تأثیر عوامل خارجی فضای کاری فناوری اطلاعات است. عوامل بیرونی از قبیل: کامل‌نبودن زیرساخت‌های ضروری IT، فراگیرنشدن

پروژه‌ی عمومی شباهت زیادی با دیگر پروژه‌ها داشته و ریسک‌های پروژه (هزینه، زمان، کیفیت) را به‌طور ذاتی دارا است. نکته‌ی بعدی اینکه، عدم به‌کارگیری استانداردها در حوزه‌ی فناوری اطلاعات، از نظر مدیران و کارشناسان حوزه‌ی نرم‌افزار، تهدیدی جدی است. از دیگر مشکلات مهم در تولید سامانه‌های بزرگ به‌روش سنتی، عدم وجود روش منسجم و یکپارچه برای طراحی و توسعه و آزمون و پشتیبانی نرم‌افزار است [۵].

تمامی شرکت‌ها درصدد تولید نرم‌افزار با کیفیت بالا و هزینه‌ی پایین و تخمین زمان نسبتاً دقیق و تحویل در زمان مورد توافق است. اما در عمل افزایش کیفیت ملزم به بالا رفتن هزینه بوده و تنظیم زمان نیازمند برنامه‌ریزی دقیق و رعایت تحقق فازهای پروژه‌ی نرم‌افزاری است و این به معنی جذب منابع انسانی خبره و نهایتاً افزایش هزینه پروژه است. نکته‌ی قابل توجه در مدیریت پروژه‌ها، کنترل متعادل هر سه عامل به‌صورت متوازن است. اینجاست که لزوم وجود استاندارد نرم‌افزار به‌وضوح احساس می‌شود. کارشناسان نرم‌افزار معتقدند که پشتیبانی قدرتمند محصولات و همچنین مستندسازی و تخمین کیفیت تنها با وجود رعایت استانداردهای منظم مقدور است. به‌عنوان نمونه نرم‌افزاری که بدون توجه به استانداردهای کیفی و مدیریتی و عدم مستندسازی تولید شده باشد امکان بازسازی، اصلاح و پشتیبانی را از تولیدکننده خواهد گرفت و در واقع دوباره نوشتن یک چنین نرم‌افزاری از بازسازی و عیب‌یابی آن راحت‌تر است.

۲) چالش‌های تولید و توسعه‌ی محصولات نرم‌افزاری

"نرم‌افزار" به تعبیری، پیچیده‌ترین صنعتی است که تاکنون توسط بشر ساخته شده است. وابستگی دنیای امروز به نرم‌افزار، فوق تصورات ما در دهه‌های

گذشته است. دنیای امروز بدون نرم‌افزار معنایی ندارد، دنیای امروز با نرم‌افزار حرکت کرده و پیش می‌رود [۵]. در مسیر استفاده از محصولات نرم‌افزاری، روزبه‌روز بر پیچیدگی آن‌ها افزوده و چالش‌های نرم‌افزاری بیشتری به‌وجود آمده‌اند. آقای سامرویل^۱ در سال ۱۹۸۶ مهندسی نرم‌افزار و چالش‌های آن را معرفی کرد که همچنان نیز مطرح هستند:

- نرم‌افزارهای تولید شده دارای تأخیر است. معمولاً زمان ارائه نرم‌افزار بیشتر از زمان تخمین زده شده است.

- نرم‌افزار بدون خطا نیست.

- نرم‌افزار گران است.

- نگهداری نرم‌افزار مشکل است. [۵]

اگرچه این مشکلات کاهش یافته‌اند، ولی هنوز حل نشده‌اند. چرخه‌ی حیات محصول و یا چرخه‌ی حیات پروژه به‌طور عام از شروع ایده تا مرحله‌ی وارهایی تعریف می‌شود. به‌طور خاص چرخه‌ی حیات نرم‌افزار از مراحل گوناگونی تشکیل یافته است که پایداری بخش نرم‌افزار و میزان بهبود مورد نیاز، قبل از ارائه نسخه‌ی نهایی را توضیح می‌دهد. اولین نسخه‌ای که برای آزمایش کنندگان نرم‌افزار که اشخاصی غیر از مهندسين نرم‌افزار هستند، تهیه می‌شود به نسخه‌ی آلفا^۲ معروف است. مرحله‌ای که آن را رفع اشکالات می‌نامند، مرحله‌ی بتا^۳ و در نهایت مرحله‌ای که همه‌ی اشکالات برطرف شده‌اند و تحت آزمون داده‌های واقعی قرار می‌گیرد، مرحله‌ی تثبیت^۴ نامیده می‌شود. در روش تولید سامانه‌های کوچک نرم‌افزاری، ابتدا شناسایی کلیه‌ی نیازهای مشتری دریافت و فرایندهای طراحی و تولید به‌صورت مشترک و هم‌زمان با هم انجام می‌پذیرد، که در اکثر مواقع نیز با موفقیت به اتمام می‌رسد. در به‌کارگیری این سبک هر تولیدکننده‌ی نرم‌افزاری یک روش تولید خاص خود اتخاذ می‌کند و در مراحل ساخت از روش‌شناسی‌های^۵ متفاوت استفاده می‌کند.

1. Sommerville
2. Alfa
3. Beta

4. Stable
5. Methodologies

تحت این شرایط، نتیجه‌ی کار شرکت‌های مختلف به لحاظ روند کیفی و اجرایی شکل و شمایی گوناگون خواهد داشت و این امر اصولاً یکپارچگی محصولات در این بازار را فدای خود خواهد کرد. اجرای چنین روشی در سامانه‌های بزرگ به علت گستردگی فعالیت‌ها غیرممکن بوده و نیاز به استفاده از روش‌ها و ابزارهای جدید است. برخی تهدیدات حوزه‌ی توسعه‌ی نرم‌افزار در ادامه مطرح شده است:

• **محصول نامحسوس:** در پروژه‌های نرم‌افزاری می‌توان به ویژگی‌هایی مانند غیر قابل دیدن^۱ محصول، پیچیدگی‌های^۲ تولید و پشتیبانی و انعطاف‌پذیری نرم‌افزار^۳، اشاره نمود که هر کدام به تنهایی تهدیدی جدی برای پروژه‌های نرم‌افزاری تلقی می‌شوند.

• **منابع محدود شرکت نرم‌افزاری (منابع انسانی متخصص):** فعالان حوزه‌ی تولید نرم‌افزار در کشور ما، اغلب شرکت‌های کوچک و با پشتوانه مالی کم هستند. منابع انسانی تیز هوش و خلاق از سرمایه‌های با ارزش این‌گونه شرکت‌ها هستند که جذب و نگهداشت این نیروها نیاز به حمایت مالی دارد. در چنین فضایی شرکت با توان اندک، اقدام به قبول پروژه‌ای بزرگ و عقد قرارداد می‌نماید که در اکثر مواقع منجر به وصل این شرکت به یک شرکت بزرگتر و یا لغو قرارداد می‌شود.

• **تغییرات سریع تکنولوژی:** شرکت‌های نرم‌افزاری برای حضور در بازار پر تلاطم فناوری اطلاعات نیاز به همراهی با تغییر و تحولات در حوزه‌ی کسب و کار را داشته و باید دائماً نوآوری‌ها در این حوزه را رصد نمایند. در واقع شاهد آن هستیم که شرکت‌های کوچک به دلیل محدودیت منابع از توسعه‌ی فناوری‌های روز به منظور آشنایی با ابزار و روش‌های نوین تولید و توسعه‌ی نرم‌افزار فاصله می‌گیرند.

• **تنوع پروژه‌های نرم‌افزاری:** در حوزه‌ی فناوری اطلاعات، تنوع پروژه‌ها بسیار و تمایز آن‌ها با یکدیگر در روش طراحی و تولید فراوان است. تولیدات

در زمینه‌های مختلف مانند نرم‌افزارهای کاربردی، بانک‌های اطلاعاتی، سیستم‌عامل، شبیه‌ساز، سامانه‌های کنترلی و صنعتی و کاربردهای متنوع دیگر انجام می‌پذیرد که هر کدام ویژگی‌های خاص و ابزارهای تولیدی مخصوص به خود را دارا هستند.

• **ضعف انجام کار گروهی:** انجام پروژه یک کار تیمی بوده و در آسیب‌شناسی پروژه‌های نرم‌افزاری، توصیه می‌شود که در کلیه‌ی مراحل چرخه‌ی حیات محصول باید تمامی ذی‌نفعان (تأمین‌کنندگان، کارکنان، مشتریان، ...) مشارکت داشته باشند.

• **عدم آگاهی کامل مشتری:** به دلیل کمبود تجربه، و عدم تعریف دقیق کار مورد نیاز توسط مجری و از طرفی عدم اشراف کاربر نهایی به موضوعات جدید فناوری اطلاعات، قراردادهای منعقد شده کامل نبوده و منجر به شکست می‌شوند.

• **عوامل ریسک:** از مهمترین مخاطرات پروژه‌های نرم‌افزاری می‌توان به چهار عامل مهم: نیروی انسانی، فرآیندکار، فناوری، دسترسی به ابزار روز اشاره کرد. با مراقبت متوازن از این عوامل مهم، موفقیت دستیافتنی و عدم توجه به هر کدام، تهدیدی جدی برای پروژه محسوب می‌شود.

مسائل فوق طی دهه‌های اخیر باعث شده‌اند تا متخصصان نرم‌افزاری (از حوزه‌های خصوصی، دولتی، دانشگاهی و نظامی) به دنبال یافتن روش‌ها و الگوهایی باشند تا بتوان بر اساس آن‌ها توسعه‌ی نرم‌افزار را روشمند و منظم نمود. برخی از اهداف آرمانی این تلاش، در قالب ارائه روش‌هایی برای توسعه و پشتیبانی و طرح کیفیت نرم‌افزار^۴ بوده که نتایج موفقیت‌آمیز آن را در پروژه‌های بزرگ و پیچیده^۵ به خصوص در نرم‌افزارهای کاربردی، صنعتی، نظامی و فضایی محقق شده‌اند.

چالش‌های مطرح شده در این بخش، نشان‌دهنده‌ی یک بی‌نظمی در تعریف و ارائه راهکار در زمینه‌ی توسعه‌ی نرم‌افزار و به‌طور کلی سامانه‌های نرم‌افزاری است. ایجاد و رعایت این نظم، از یک طرف به

1. Invisibility
2. Complexity
3. Flexibility

4. Software Quality Plan
5. Complexity

سفارش دهندگان نرم افزار برمی گردد و از سوی دیگر فعالیت های توسعه ای نرم افزار را تحت تأثیر قرار می دهد. گرچه در دنیای متلاطم و ارگانیک توسعه ای نرم افزار، ایجاد نظم کامل، غیرمنطقی و غیرعملیاتی به نظر می رسد، لیکن امکان ایجاد چارچوب های باز و پویا وجود دارد. استانداردهای این حوزه عمدتاً به ارائه ی چنین الزامات کلی و چارچوب هایی می پردازند. براین اساس در این مقاله تلاش شده است تا الگوی استانداردسازی فعالیت های چرخه ای حیات نرم افزار، به عنوان یکی از راهکارهای کاهش یا رفع چالش های موجود در این حوزه پیشنهاد شود.

۳) کیفیت نرم افزار

طبق استاندارد ISO "کیفیت نرم افزاری درجه ای است که نرم افزار ترکیبی از خصوصیات مطلوب را دارا است." [۱] همچنین در استاندارد ISO8402 "کیفیت، توانایی برآوردن نیازهای تصریحی و تلویحی است." [۹] تضمین کیفیت نرم افزاری را می توان به منزله ی چتری بر تمامی مراحل فرایند توسعه ای نرم افزار (طراحی، پیاده سازی، توسعه) در نظر گرفت. در دهه های گذشته، مدیران شرکت های بزرگ جهانی دریافته اند که تولید با کیفیت بالا منجر به صرفه جویی در هزینه ها و محصول پایانی بهتر، می شود. از طرفی کیفیت پایین به جهت امکان نیاز به پیکربندی، کدنویسی مجدد و حتی طراحی دوباره، بر روی هزینه و زمان بندی تأثیر می گذارد. لذا کشف سریع تر مشکلات کیفی در نرم افزار مقرون به صرفه تر است. اولین گام تولید نرم افزار معماری است که ویژگی های کیفیتی در آن قابل ردیابی است. کیفیت نرم افزار، شامل موارد زیر است: کیفیت عملکردی^۱ و کیفیت ساختاری نرم افزار.^۲

کیفیت عملیاتی نرم افزار، شاخصی جهت نشان دادن میزان تطابق نرم افزار با نیازمندی های عملیاتی تعریف شده برای نرم افزار بوده و کیفیت ساختاری

نرم افزار، منعکس کننده ی میزان دستیابی به نیازمندی های غیر عملیاتی مانند پایداری و قابلیت نگهداری نرم افزار و ... است [۶]. میزان دستیابی به نیازمندی های غیر عملیاتی معیاری است که روی محورهای مدنظر بیشتر سازمان ها، تأثیر داشته و از آن می توان به عنوان عاملی مؤثر جهت کاهش مشکلات، یاد کرد، به نحویکه نرم افزار تولیدی عاری از خطا و اشتباه باشد، در ظرف مدت زمانی تعیین شده و هزینه ای پیش بینی شده به اتمام برسد، رضایت تمامی ذی نفعان برآورده شود، فرآیند توسعه ای نرم افزار از کیفیت مناسبی برخوردار باشد، توسعه ای دانش مستند شده و متکی به افراد نباشد و پشتیبانی از محصول تولیدی آسان و کم هزینه باشد [۶].

۴) ضرورت استانداردسازی

پیچیدگی سامانه های جدید در تجهیزات فناوری اطلاعات و لوازم بازار دقیق و الکترونیکی و کنترلی بسیار زیاد شده است. این مسئله با وجود اینکه مشکلاتی را به همراه دارد، زمینه های بهبود را نیز به ارمغان می آورد. این مشکلات بیشتر ناشی از الزام سازگاری سخت افزار و نرم افزار در کنار یکدیگر هستند. مدیریت هماهنگی و یکپارچگی بین اعضای سیستم بسیار مشکل و نیاز به زمان بندی های دقیق دارد. برای بهبود ارتباطات و تعاملات اعضای یک سیستم نیاز به تعیین چارچوب و تعریف فرآیندهای قوانین مشترک و رعایت الزامات وضع شده است. استاندارد، ملزومات واقعی یک سیستم به همراه ویژگی های آن را مشخص می کند و تحت این شرایط مجری طرح به راحتی می تواند اقدامات لازم جهت پیشبرد موفق طرح و رسیدن به یک نتیجه ی مطلوب را شناسایی و یک محصول قدرتمندتر و با پشتیبانی بهتر تولید نماید. وقتی استاندارد صحیحی تدوین شود، آنگاه تقسیم کار و در انتها، یکپارچه سازی محصول نهایی در کار گروهی با دقت

و صحت بیشتری همراه خواهد بود. با وجود چنین استانداردی، کارفرما در هنگام برون‌سپاری به‌راحتی می‌تواند پروژه را میان چند مجری تقسیم کند و به‌علت تعامل مناسب مجریان از طریق این استاندارد و امکان نظارت مناسب از سوی کارفرما نتیجه‌ی قابل قبولی حاصل خواهد شد. برای پاسخگویی به این نیاز مبرم در جهت حفظ و توسعه‌ی بازار داخلی و همچنین زمینه‌سازی حضور در بازارهای هدف خارجی، شرکت‌های نرم‌افزاری داخلی نیاز به اخذ گواهی‌نامه استانداردهای معتبر داخلی و بین‌المللی را دارند. در این وضعیت استانداردسازی فرآیندهای جاری شرکت و نهادینه‌سازی روح کیفیت در استانداردهای محصولات تولیدی، تنها راه‌حل هماهنگ‌سازی فعالیت‌های انبوه و رهایی از مسائل و مشکلات پیچیده است. اما رابطه‌ی کیفیت و استاندارد چیست و چگونه می‌توانند مکمل یکدیگر شوند؟ به‌منظور درک همگرایی کیفیت و استاندارد می‌توان به برخی از اهداف استانداردسازی مانند: ایجاد فهم مشترک از الزامات بین کلیه‌ی ذی‌نفعان، طراحی پایدار و مطمئن، ارتقاء پودمانگی^۱، بهبود قابلیت‌اعتماد (RAMS)^۲، ایجاد قابلیت کنترل فرآیند تحقق محصول، ارتقاء فرآیندها و در نهایت تولید اقلام تجاری استاندارد اشاره نمود.

۵) روش‌شناسی توسعه‌ی نرم‌افزار^۳

از طرفی هدف از معماری فرآیند تحقق ایده، استانداردسازی فرآیندهای کلان چرخه‌ی تحقق محصولات، از شکل‌گیری ایده تا ساخت نمونه‌ی محصول است. در این راستا استفاده از استاندارد IDS-1107 باعث ایجاد یکپارچگی و زبان مشترک در تعریف، اجراء، نظارت، ارزیابی و ممیزی فرآیندهای مذکور خواهد شد. [۱] در توسعه‌ی سامانه‌های بزرگ نرم‌افزاری استفاده از روش‌شناسی‌ها اجتناب‌ناپذیر است. در روش‌شناسی‌های جدید، فعالیت‌های مشترک

و تکراری در مراحل چندگانه در چرخه‌ی حیات نرم‌افزار قرار گرفته و با استفاده از فنونی خاص و توزیع مراحل انجام کار در فازهای مختلف از افزایش هزینه و اتلاف منابع جلوگیری به‌عمل می‌آید. موفقیت در این مسیر جز با تبعیت از استانداردها حاصل نخواهد شد. زیرا استانداردها همان تجربیات برتر هستند که قانونمند شده‌اند.

برای تولید یک نرم‌افزار، اصول و قوانین خاصی وجود دارد که در مجموع، فرآیند تولید را مطابق با استانداردها و برای دستیابی به محصولی با بهترین کیفیت هدایت می‌کند. چرخه‌ی حیات محصول، سیر تکاملی یک محصول، از زمان پیدایش ایده تا زمان از رده خارج شدن است. چرخه‌ی حیات، دربرگیرنده‌ی تمامی مراحل مختلف توسعه‌ی نرم‌افزار (تعریف انتظارات ذی‌نفعان - تعریف الزامات فنی - تجزیه‌ی منطقی و شکست الزامات - ارائه راه‌حل فنی - اکتساب محصول - مونتاژ و یکپارچه‌سازی - تصدیق محصول - صحت‌گذاری محصول - انتقال) است. پروژه‌های مختلف نرم‌افزاری اگر چه به‌طور قابل ملاحظه‌ای با هم متفاوتند ولی به‌طور معمول دارای مراحل اجرایی فوق هستند [۳]. مسئله‌ی اساسی و مهم در توسعه‌ی نرم‌افزار، ارتباط فرآیندهای مختلف تولید نرم‌افزار در چرخه‌ی حیات است که این ارتباطات با استفاده از مدل‌ها قابل پیاده‌سازی هستند. پیروی از این مدل‌ها در استقرار موفق مراحل چرخه‌ی حیات نقش اساسی ایفاء می‌نماید. بر همین اساس از گذشته تاکنون مدل‌های متفاوتی به‌منظور طراحی و پیاده‌سازی نرم‌افزار ارائه گردیده، که می‌توان به مدل‌هایی نظیر مدل آبشاری^۴، مدل^۵، مدل حلزونی^۶، فرآیند منطقی یکپارچه (RUP)^۷، فرآیند نرم‌افزار چابک (ASP)^۸ و ... اشاره نمود. هر مدل دارای شاخص‌ها و ویژگی‌های منحصربه‌فرد خود بوده و نرم‌افزارهایی که با اتکاء بر هر یک از مدل‌های فوق پیاده‌سازی می‌شوند، خصایص خود را از مدل به‌کارگرفته‌شده به ارث خواهند برد. این

1. Maturity
2. Reliability, Availability, Maintainability, safety & Security
3. Software Development Methodology
4. Waterfall model

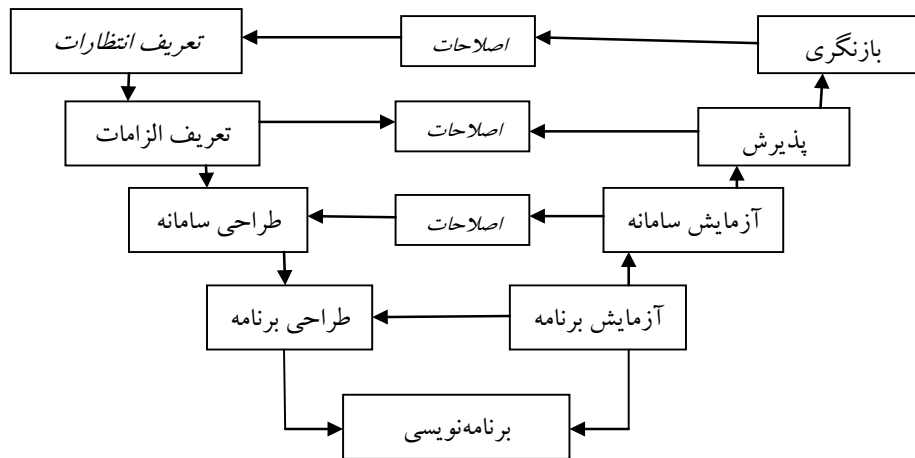
5. V model
6. Spiral model
7. Rational Unified Process
8. Agile Software Process

مدل‌ها نسبت به یکدیگر دارای مزایا و معایبی هستند و طراحان با توجه به نیاز پروژه‌های نرم‌افزاری و در نظر گرفتن ویژگی‌های مدل‌های ارائه شده از مدل مطلوب استفاده می‌نمایند. با توجه به اینکه دو مدل فرآیند منطقی یکپارچه (RUP) و مدل V کاربرد بیشتری داشته و از طرفی مفاهیم این حوزه را به طرز کامل‌تری در برمی‌گیرند، در ادامه به اختصار این دو مدل تشریح شده‌اند.

۵-۱) معرفی مدل V

شکل (۱) نمایش نموداری از مدل V را نشان می‌دهد. این مدل یک شکل گسترده از مدل آبشاری است

و تأکید بر ضرورت تطابق فعالیت‌ها با فعالیت‌های متناظری است که محصولی تولید می‌کنند. در مدل V نسبت به مدل آبشاری فعالیت آزمایش، گسترش یافته‌تر است. هر مرحله دارای یک فعالیت تطبیق مختص به خود است که در صورت وجود مغایرت، موجب بازگشت به فاز مربوطه و انجام مجدد مراحل بعدی آن خواهد شد (تداعی چرخه‌ی دمینگ)^۱. به‌طور ایده‌آل، فقط در صورت وجود مغایرت بین آنچه که در فعالیت خاصی مشخص گردیده است با آنچه که واقعاً در پیاده‌سازی به‌وقوع پیوسته است، بازگشت به عقب اتفاق خواهد افتاد.^[۳]



شکل ۱: نمایش نموداری از مدل V

همان‌طور که در شکل مشاهده می‌شود در سمت چپ بعد از شناسایی نیازهای مشتری، شکست فعالیت‌ها آغاز و جزئیات برای انجام طراحی مهیا می‌شود. در پائین‌ترین مرحله پس از اتمام برنامه‌نویسی عمل یکپارچه‌سازی شروع و آزمون‌های خاص صورت پذیرفته و اصلاحات برای شروع مرحله‌ی بعد و یا بازگشت به فعالیت مقابل صورت می‌پذیرد. مدل V به‌علت کارایی و سادگی آن در تعریف پروژه و یا تحویل پروژه کاربرد فراوان دارد.

۵-۲) فرآیند منطقی یکپارچه (RUP):

از معروفترین و به‌روزترین رویکردهایی که در سال‌های اخیر به‌عنوان ابزار معماری در فرآیندهای نرم‌افزاری به‌کار گرفته می‌شود، می‌توان به فرآیند یکپارچه‌ی

رشنال رز^۲ (RUP) اشاره کرد که با دیدگاهی مشتری‌گرا در بهینه‌سازی فرآیندهای تولید محصول، به تیم‌های نرم‌افزاری کمک می‌کند. RUP با فراهم آوردن اصول و قواعدی یکپارچه، روی تولید بهترین محصولات تمرکز می‌کند^[۴]. فرآیند یکپارچه‌ی RUP در حقیقت یک ابزار مهندسی نرم‌افزار است که زمینه‌های توسعه مانند دستورالعمل‌ها، کدها، مدل‌ها و غیره را با مواردی از قبیل تکنیک‌ها، سازوکارها، مراحل تعریف شده و شیوه‌ها، در چارچوبی یکپارچه ترکیب می‌نماید. این متدولوژی یک چارچوب برای پروژه فراهم می‌آورد که کلاس‌هایی از فرآیندها را به‌طور افزایشی^۳ و چرخشی مطرح می‌کند. بر اساس رویکرد RUP، توسعه‌ی نرم‌افزار در نگاهی کلی، شامل

1. Damming Cycle
2. Rational Rose

3. Incremental

چهار فاز می‌شود:

۱. فاز شناخت^۱: مبتنی بر ارزیابی نیازها، قابلیت دوام، امکان‌پذیری و نیازمندی‌های یک پروژه یا برنامه است.

۲. فاز تفصیل^۲: معماری مناسب سامانه را بر اساس نیازهای پروژه اندازه‌گیری می‌کند.

۳. فاز پیاده‌سازی^۳: نسخه اولیه از نرم‌افزار توسعه یافته را ارائه می‌دهد.

۴. فاز انتقال^۴: چرخه توسعه نرم‌افزار را در صورتی که تمامی اهداف پروژه تأمین شود به پایان می‌رساند [۴].

سازمان باید در پیاده‌سازی RUP فرآیندهایی را طرح‌ریزی و اجراء نماید تا به کمک آن‌ها بتواند خرابی‌های بالقوه در محصول را در مراحل اولیه چرخه حیات به‌ویژه در مرحله طراحی و توسعه شناسایی کند تا از وقوع آن‌ها در حوزه مصرف به‌نحومقتضی پیشگیری شود. این روش‌شناسی بسیار قدرتمند بوده و در پروژه‌های بزرگ کارایی بسیاری دارد. در سال‌های گذشته کشورها و شرکت‌های زیادی اقدام به بومی‌سازی و بهره‌برداری از این روش کرده‌اند. در اغلب موارد پس از بومی‌سازی، ساختار شرکت بر اساس مراحل RUP سازماندهی شده و یکسان‌سازی روش کار و مستندات انجام می‌پذیرد.

۶) مدل‌های چرخه حیات

در ابتدا استانداردهای موجود در زمینه چرخه حیات در حوزه استانداردهای دفاعی مطرح و سپس با توجه به فوائد بی‌شمار آن به حوزه تجاری راه یافت.

کاربرد چرخه حیات سیستم در یک سازمان و یا یک پروژه بزرگ، مثال‌های متنوعی از تأثیر مهم آن به‌عنوان ابزاری مفید در شکست کار (WBS)^۵ و رصد نمودن مراحل انجام امور پیچیده است. از ویژگی‌های چرخه حیات می‌توان به موارد زیر اشاره نمود:

• فازبندی چرخه حیات با هدف برنامه‌پذیر و

کنترل‌پذیر نمودن فعالیت‌ها انجام می‌شود.

• ورودی‌ها، فعالیت‌ها و خروجی‌های هر فاز به‌صورت شفاف بیان می‌شوند.

• برای هر فاز یک هدف مشخص بیان می‌شود.

• هر فاز دارای یک یا چند خروجی است.

• پایان هر فاز به‌گونه‌ای تعیین می‌شود که نقطه‌ای ارزیابی پیشرفت فرآیند باشد.

• جهت تصمیم‌گیری در نقاط حساس، امکان بازنگری و تصمیم‌سازی تعیین می‌شود.

با توجه به اهمیت موضوع برای شرکت‌ها، تاکنون موسسه‌ی بین‌المللی استاندارد (ISO) استانداردهای متنوعی برای چرخه حیات محصول و سامانه‌ها معرفی نموده است. سازمان جهانی استاندارد با نگاه فرآیندی و فارغ از ساختار سازمانی، استاندارد چرخه حیات را تدوین و منتشر نموده است.

همان‌طور که چرخه حیات پروژه برای شرکت‌ها و سازمان‌های پروژه‌محور حائز اهمیت است، چرخه حیات محصول برای بسیاری از سازمان‌های بزرگ نظامی و دفاعی مانند NASA^۶ و DOD^۷ مهم است.

درواقع با پیشرفت تکنولوژی، فرصت‌های بهبود جدیدی ایجاد می‌شود و سازمان‌های بزرگ، چرخه حیات محصولات را به‌چالش کشانده و با بازنگری چرخه حیات محصول، بهبود مستمر را برای محصول محقق می‌نمایند. این سازمان‌ها چرخه حیات را به‌عنوان یک فرآیند جاری تصویب و با هدف بهبود روش‌ها و کاهش هزینه‌ها در مقاطع زمانی متفاوت، مورد بازنگری قرار می‌دهند.

استانداردهایی که مؤسسه‌ی ISO برای چرخه حیات ارائه نموده ISO/IEC 12207:2008 و ISO/IEC 15288:2008 هستند، که در ادامه به توضیح آن‌ها پرداخته خواهد شد.

۱-۶) مدل فرآیندهای چرخه حیات ISO/IEC 15288 اولین نسخه‌ی استاندارد ISO/IEC 15288:2008 با عنوان چرخه حیات سیستم منتشر گردید. این استاندارد

۱. Inception

۲. Elaboration

۳. Construction

۴. Transition

۵. Work Breakdown Structure

۶. National Aeronautics and Space Administration

۷. Department Of Defense

به ارائه‌ی چارچوب چرخه‌ی حیات برای اجرای فعالیت‌ها به صورت منظم و سازمان‌یافته و همچنین ارائه‌ی چارچوبی برای ارزیابی بلوغ فرآیند می‌پردازد. در این استاندارد از منظر سیستمی به چرخه‌ی حیات محصول پرداخته شده و لذا تفکر سیستمی و فرآیندی به همراه تعریف زیرسامانه‌ها به عنوان عناصر اصلی هستند. رویکرد سیستمی در چرخه‌ی حیات محصول باعث می‌شود که کلی‌نگری انجام و لذا عوامل

تأثیرگذار و میزان تأثیر آن‌ها شناسایی شوند. از نقاط قوت این استاندارد لزوم رعایت نظام مهندسی سیستم^۱ است. مهندسی سیستم برای اطمینان از تبدیل صحیح الزامات مشتری به الزامات فنی که در محصول مورد نظر باید متبلور شود، با به‌کارگیری سه مهارت (مهارت‌های مهندسی، مدیریت پروژه، مدیریت کسب و کار) مدیریت فعالیت‌ها را در طول اجرای پروژه جاری می‌سازد.



شکل ۲ - فرآیندهای چرخه‌ی حیات سیستم بر اساس ISO/IEC 15288

همان‌گونه که در شکل ۲ نشان داده شده است، استاندارد ISO/IEC 15288:2008 دارای ۴ فرآیند اصلی و ۲۵ فرآیند فرعی است. فرآیندهای اصلی در چرخه‌ی حیات عبارتند از:

۳. فرآیندهای پروژه‌ای: فرآیندهای پروژه مربوط به عملیات مدیریتی و کنترلی پروژه‌ها است.
 ۴. فرآیندهای فنی: مهمترین فرآیندها در ستون فرآیندهای فنی قرار دارند که به نوعی تداعی کننده‌ی مدل V هستند. [۷]

۱. فرآیندهای توافق: توافق مربوط به موضوعات تأمین و عرضه و پیگیری‌های اولیه‌ی شروع پروژه و مباحث مالی و بودجه‌بندی
۲. فرآیندهای سازمانی توانمندساز پروژه: فرآیندهای دوم، توانمندسازهای پروژه است که کنترل چرخه‌ی حیات^۲ خاص هر سازمان است.

فرآیندهای فنی، نیاز ذی‌نفعان را ابتداءً به محصول تبدیل نموده و سپس با به‌کارگیری این محصول، خدمت پایداری را در زمان و مکان لازم در راستای رضایت مشتری ارائه می‌نمایند. فرآیندهای فنی برای ایجاد و استفاده از یک سیستم چه به شکل مدل و

1. System Engineering
 2. Project Life Cycle

چه در فرم محصول نهایی به کار می‌روند. همان‌گونه که در شکل ۲ مشاهده می‌شود این فرایند شامل ۱۱ فرایند فرعی است. در این استاندارد تأکید بر انجام تصدیق و صحت‌گذاری در اتمام هر فاز و صدور مجوز برای ورود به فاز بعدی است. همچنین در مراحل طراحی به دو قابلیت تکرارپذیری^۱ و بازگشت‌پذیری^۲ که از ضروریات چرخه‌ها هستند، اشاره گردیده است.

۶-۲) مدل فرآیندهای چرخه‌ی حیات ISO IEEE/EIA 12207

هدف اصلی چرخه‌ی حیات ISO/IEC 15288، ارائه‌ی چارچوبی برای تحت پوشش قرار دادن فرآیندها و تحقق مراحل ایده تا محصول است. چرخه‌ی حیات ISO/IEC 15288 به صورت کلی برای انواع محصولات و پروژه‌ها کاربرد داشته و استفاده از آن مزایای زیادی به همراه خواهد داشت. از آنجایی که نرم‌افزار جزء جدایی‌ناپذیر از سیستم‌های بزرگ بوده و از طرفی پروژه‌ها و محصولات نرم‌افزاری دارای پیچیدگی‌ها و ظرافت‌های خاصی هستند، لذا نیاز به چرخه‌ی خاصی نیز هستند. از اوایل دهه ۱۹۷۰، استانداردهای مختلفی در خصوص چرخه‌ی حیات تولید نرم‌افزار ارائه شده است. نکته‌ی قابل تأمل، سرعت تدوین و منسوخ شدن این استانداردها است. دلیل این امر، رشد فزاینده استفاده از نرم‌افزار در دهه‌های اخیر، توسعه نرم‌افزارهایی با پیچیدگی و کاربردهای مختلف بوده است. نکته‌ی حائز اهمیت اینک، نرم‌افزار دارای ماهیتی پویا بوده و در هر زمان باید خود را با خیل عظیم نیازها و انتظارات جدید کاربران تطبیق نماید. بدیهی‌است از گذشته تاکنون، طیف خواسته‌های انسانی تغییر کرده و در حال تغییر است، همچنین، سخت‌افزارها دچار تغییر و تحول گسترده‌ای بوده و در حال پیشرفت هستند. در این راستا لازم است نرم‌افزار نیز با رعایت کامل اصل انعطاف‌پذیری، پذیرای تمامی تحولات از گذشته تاکنون بوده و بتواند در هر زمان رسالت خود را به‌خوبی انجام دهد. بنابراین باید نرم‌افزارهای تولیدی به‌سادگی قابل تغییر بوده و به‌توان به سادگی قابلیت‌های جدید

را به آن‌ها اضافه و یا از آن حذف نمود. این تغییرات می‌توانند پیشگیرانه، اصلاحی و یا بهبوددهنده باشند و لازمه‌ی این امر تبعیت از استانداردهای چرخه‌ی حیات در حین تولید است.

به‌منظور برآورده‌سازی این نیاز سازمان بین‌المللی استاندارد، چرخه‌ی حیات نرم‌افزار ISO/IEC 12207 را ارائه نموده است. این استاندارد در هماهنگی و سازگاری با استاندارد چرخه‌ی حیات ISO/IEC 15288 بوده و در آن فعالیت‌ها و وظایف مرتبط با چرخه‌ی حیات نرم‌افزار از ابتدا تا انتها مشخص شده است. شکل (۳) جریان فرآیندی و ارتباط آن‌ها با فرآیندهای اصلی را نشان می‌دهد. این استاندارد براساس طبقه‌بندی فرایندها، سازماندهی شده است. فرایندهای ISO/IEC 12207 به سه طبقه‌ی اصلی فرایندهای اولیه، فرایندهای پشتیبانی، فرایندهای سازمانی تقسیم‌بندی شده‌اند که در ادامه به شرح مختصری از آن‌ها پرداخته خواهد شد.

۶-۲-۱) فرایندهای اولیه

این فرایندها در طول حیات یک پروژه‌ی نرم‌افزاری انجام می‌شوند. در این قسمت پنج فرایند تعریف شده‌اند که عبارتند از: تقاضا، تأمین، تولید و توسعه، اجرا، نگهداری. این پنج فرایند چرخه‌ی حیات نرم‌افزار را از تعریف اولیه‌ی سیستم تا بازنشستگی سیستم پوشش می‌دهد.

۶-۲-۲) فرایندهای پشتیبانی

فرایندهای پشتیبانی برای بهبود کلیه‌ی فرایندها تعریف می‌شوند. به‌طور کلی، فرایندهای پشتیبانی در سطح پروژه، مدیریت گردیده و شامل ۸ فرایند است که فرایندهای فرعی هشت‌گانه‌ی: مستندات، مدیریت پیکره‌بندی، تضمین کیفیت، تصدیق، صحت‌گذاری، بازنگری دسته جمعی، ممیزی، شفاف‌سازی مشکلات هستند.

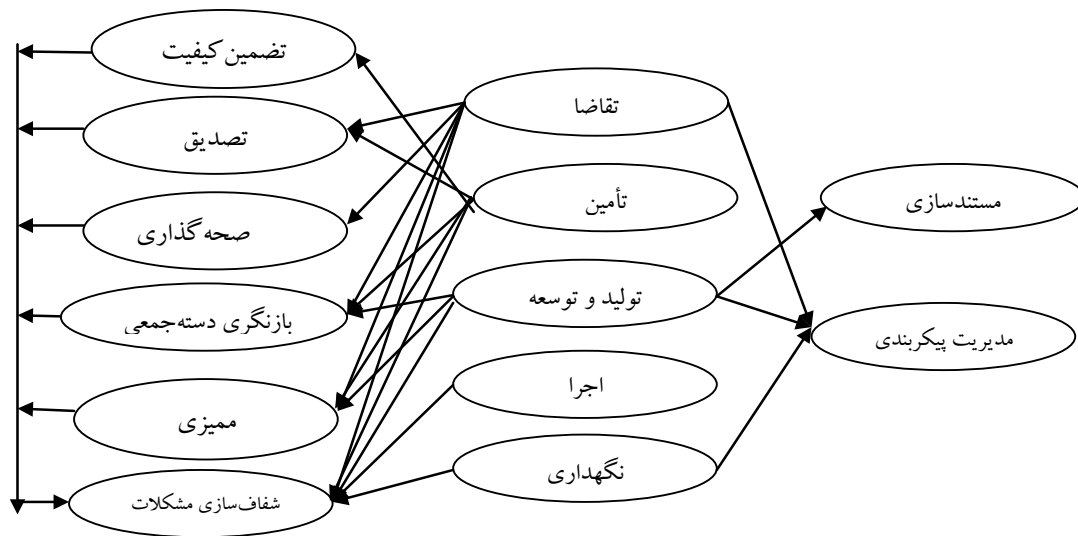
۶-۲-۳) فرایندهای سازمانی

فرایندهای سازمانی در سطوح شرکتی استقرار می‌یابند، اما در سطح پروژه اعمال می‌شوند. این

1. Recursive
2. Iteration

فرایندهای ستون فقرات، یک پروژه‌ی نرم‌افزاری را تشکیل می‌دهند. چهار فرایند سازمانی در اینجا

تعریف شده‌اند که عبارتند از: فرایند مدیریت، فرایند زیرساخت، فرایند بهبود، فرایند آموزش. [۸]



شکل ۳: جریان فرآیندهای پشتیبانی IEEE/EIA 12207 و ارتباط آن‌ها با فرآیندهای اصلی

نکته‌ی قابل توجه، انعطاف این استاندارد برای پروژه‌ها یا سازمان‌های خاصی است که احتمالاً نیازی در به‌کارگیری تمام فرآیندهای فراهم شده توسط این استاندارد را نداشته باشند. به همین دلیل، اجرای این استاندارد می‌تواند شامل گزینش مجموعه‌ای از فرآیندهای مناسب برای پروژه یا سازمان باشد. دو روش پیشنهادی (انطباق کامل - انطباق متناسب) مطرح است. ادعای انطباق کامل مجموعه‌ی فرایندهایی را اعلام می‌کند که انطباق با آن‌ها وجود دارد. انطباق کامل توسط نشان دادن اینکه تمامی الزامات مجموعه اعلام شده‌ی فرآیندها با استفاده از نتایج به‌عنوان شواهد، برآورده شده‌اند، به‌دست می‌آید. زمانی که این استاندارد برای برقراری مجموعه‌ای از فرآیندهایی که واجد شرایط برای انطباق کامل نیستند، انطباق متناسب استفاده می‌شود. [۸] هر چرخه‌ی حیاتی که این موارد را رعایت نموده باشد از نظر مؤسسه‌ی IEEE استاندارد بوده و تا حدی قابل اعتماد است. از این استاندارد به‌منظور ایجاد چارچوبی برای تبیین چرخه‌ی حیات نرم‌افزار، استفاده شده است. هدف این استاندارد آن است که فرآیندهای ۱۷ گانه‌ی فوق توسط سازمان‌ها بر اساس نیازمندی‌های خاص هر پروژه

متناسب‌سازی و هم‌همی فعالیت‌های غیر مرتبط حذف شود. این استاندارد سازگاری کارایی فرآیندها، فعالیت‌ها، وظایف انتخابی را پس از متناسب‌سازی تعریف می‌نماید.

۳-۶ استفاده از استانداردها در چرخه‌ی حیات نرم‌افزار

در دو قسمت قبلی به موضوعات استانداردسازی و نحوه‌ی تداوم چرخه‌ی حیات نرم‌افزار اشاره شد. در این قسمت به تلفیق این دو موضوع، که نتیجه‌ی آن به‌کارگیری استاندارد در چرخه‌ی حیات نرم‌افزار است پرداخته خواهد شد و انتظار است که خروجی چنین ادغامی محصولی با کیفیت باشد. کیفیت در یک پروژه را می‌توان تا حد زیادی وابسته به استانداردهایی دانست که در آن به‌کار گرفته می‌شود. استانداردهای متعددی در زمینه‌های مختلف وجود دارد که باید از هر کدام به درستی و در زمان و مکان مقتضی از آن‌ها بهره برد. پس می‌توان نتیجه گرفت که یکی از مؤلفه‌های اصلی در بحث کیفیت، استفاده درست و مناسب از استانداردها است. در حقیقت وجود استاندارد و کیفیت در یک محصول نرم‌افزاری تأثیر متقابل داشته و به‌کارگیری استانداردها در چرخه‌ی حیات نرم‌افزار، کیفیت را تضمین می‌نماید. به‌منظور استانداردسازی چرخه‌ی حیات استانداردهایی

در مورد فعالیت‌های داخل چرخه‌ی حیات تدوین و ارائه گردیده، همچنین برای تمامی فعالیت‌های مربوط به فرآیندهای پشتیبانی و فرآیندهای اصلی، به‌طور خاص استانداردهایی پیشنهاد شده است. در جدول (۱) نگاشت فعالیت‌های ISO IEEE/EIA

12207 به برخی از استانداردهای رایج را مشاهده می‌نمایید. استانداردهای ذکرشده در این جدول منحصر به فعالیت مربوطه نبوده بلکه شرکت‌های معتبر در حوزه‌ی تدوین استانداردهای جهانی، استانداردهای مشابه را ارائه نموده‌اند.

| Software Specific Processes (ISO /IEC 12207) | | | |
|--|--|---|---|
| SW Implementation Processes | استانداردهای پیشنهادی | SW Support Processes | استانداردهای پیشنهادی |
| 1- Software Implementation Processes | ISO/IEC DIS 27013 | 1-Software Documentation Management Process | ISO/IEEE 829 |
| 2-Software Requirements Analysis Processes | ISO/IEEE 830 AAQMI/ISO 13485 ISO/IEC 27001 | 2-Software Configuration Management Process | ISO/IEEE 1042 ISO/IEEE 828 ISO/IEEE 10007 |
| 3-Software Architectural Design Processes | ISO/IEEE 42010 :2011 | 3-Software Quality Assurance Process | ISO/IEEE 720 |
| 4-Software Construction Process | ISO/IEC23004-1 | 4-Software Verification Process | ISO/IEEE 1012 |
| 5-Software Integration Processes | ISO/IEC DIS 27013 | 5-Software Review Process | ISO/IEEE 1028 |
| 6-Software Qualification Testing Processes | AAQMI/ISO 13485 ISO/IEEE 829 | 6-Software Audit Process | ISO/IEEE 1028 |
| | | 7-Software Problem Resolution Process | ISO/IEEE STDMM 95844 |

جدول ۱: استانداردهای متناظر با فعالیت‌های چرخه حیات

۴-۶) بهبود مستمر در چرخه‌ی حیات

به‌منظور استقرار هرچه مطلوب‌تر نظامات استانداردهای بین‌المللی از جمله ISO 9000 و ISO/IEC 12207 رعایت دو نکته، فرآیندگرایی و اصول چرخه‌ی دمینگ حائز اهمیت هستند. فرآیندگرایی دارای اهمیت است زیرا، فرآیندهای سازمان می‌توانند تمام فعالیت‌های سازمان را در جهت حرکت به سمت اهداف کسب‌وکار، همسو و هم‌گرا کنند. فرآیندهای سازمان امکان بهره‌گیری صحیح از منابع را فراهم کرده و بستر لازم را برای به‌کارگیری تکنولوژی‌های جدید فراهم می‌کنند. از این رو است که در صنایع تولیدی، مدت‌ها است که اهمیت و نقش فرآیندهای کارآمد و اثربخش، بر کیفیت محصول شناخته شده است. فرآیندهای با کیفیت بالا، به نیروی انسانی سازمان‌ها کمک می‌کند که با همگرایی بیشتری، در جهت اهداف سازمان کار کنند. فرآیندهای کارآمد، موتور محرکه‌ی به‌کارگیری تکنولوژی‌های نو در سازمان هستند. همچنین اجرای حلقه PDCA بسیار ساده و مفید بوده و برای تثبیت فعالیت‌های مطلوب گذشته کاربرد دارد. این حلقه از ۴ جزء تشکیل یافته است.

۱. طرح، بدانیم که چه کاری را می‌خواهیم انجام دهیم.

۲. اجرا کنیم.

۳. در مقاطع مشخصی از زمان فعالیت‌های انجام شده را بررسی و با راه در پیش‌رو ارزیابی نماییم.

۴. اقدامات لازم به‌منظور اصلاح روند را تعیین و اجرا کنیم.

مراحل فوق اگرچه ساده و آسان به‌نظر می‌آیند لیکن به‌کارگیری آن‌ها، متضمن پایداری سیستم خواهد بود.

۷) الگوی پیشنهادی استانداردسازی در چرخه‌ی حیات نرم‌افزار

تنها گزینه‌ی شرکت تولیدی برای ماندگاری در بازار داخلی و رقابت در بازار جهانی، ارائه محصول با کیفیت و ارزان و به‌موقع به مشتری است. تحقق این سه خواسته با رعایت مراحل چرخه‌ی حیات محصول امکان‌پذیر خواهد بود. در بازار نرم‌افزار، علاوه بر مراحل توسعه نرم‌افزار و انتقال آن به محیط عملیاتی، پشتیبانی و به‌روزرسانی آن است.

به‌منظور ارائه الگوی پیشنهادی، ابتدا با مطالعات

کتابخانه‌های فرایندهای چرخه‌ی حیات که در بین متخصصان مختلف اشتراک دارند، شناسایی شده و در سند تشریح فعالیت‌های چرخه‌ی حیات، ذکر شده‌اند. سپس با مطالعه‌ی استانداردهای موجود در سطح بین‌المللی و تحلیل محتوای آن‌ها توسط کارشناسان، کامل‌ترین استانداردها انتخاب شدند. در شکل ۴ طرح الگوی پیشنهادی نمایش داده شده است. همان‌گونه که در شکل مشاهده می‌شود، فعالیت‌های درونی الگو را می‌توان به چهار بخش تقسیم نمود:

۱. چرخه‌ی حیات سیستمی ISO/IEC 15288 و چرخه‌ی حیات ISO/IEC 12207 محیط پیرامونی الگو را تشکیل و اعتبار پویایی سیستم هستند.

در ابتدائی‌ترین مرحله که شروع و تعریف پروژه مطرح است، هماهنگی با ISO/IEC 15288 به صورت مطلوبی ایفای نقش نموده و دیدگاه سیستماتیک بر آن حاکم می‌شود. از طرفی مطابق رویه‌های ISO/IEC 12207 هر پروژه جدید مستلزم طرح‌ریزی مشخص جهت انجام آن خواهد بود. نیازمندی‌های نیروی انسانی، تجهیزات، ابزارها، روش‌های اجرایی ویژه، آموزش‌های ویژه و ... و به‌طور کلی مطابق پیش‌بینی‌های اولیه منعکس در رویه‌ها باید بررسی، برآورد و تأمین گردند، که در این فاز کلیه‌ی مستندات مرتبط تهیه می‌گردند.

۲. مهندسی قابلیت‌اعتماد در چرخه‌ی حیات محصول مانند چتری بر تمام فعالیت‌های تولید و توسعه سایه افکنده است و با رعایت اصول (RAMS) ضریب اطمینان عملیاتی محصول نهایی فراهم خواهد شد.

۳. مدل V به‌عنوان پایه و اساس گردش فعالیت‌ها بوده و در هر مرحله از مدل، استاندارد مربوطه معرفی گردیده و حلقه PDCA در هر سطح تکرار شده است.

۴. مدیریت کیفیت سری ISO 9000 و امنیت ISO/27000 IEC به‌عنوان استانداردهای زیربنایی در چرخه‌ی حیات محصول ذکر شده‌اند.

در اولین گام اجرایی، باید شناخت اولیه و ارزیابی کلی از فعالیت‌های مورد نظر شرکت به‌عمل آمده و الزامات استانداردهای سیستم مدیریت کیفیت ISO 9000 و ISO 12207 در زمینه‌ی فعالیت‌های مربوطه تفسیر شوند. مطابق رویه‌های ISO 9000، هر شرکت دارای خط‌مشی

کیفیت، طرح کیفیت، رویه‌های مختلف جهت پوشش عناصر ۲۰ گانه‌ی ISO با توجه به ویژگی‌های صنعت نرم‌افزار و مستندات پشتیبان مربوطه خواهد شد به گونه‌ای که کل فرایند چرخه‌ی حیات تولید نرم‌افزار تحت پوشش رویه‌های مذکور قرار می‌گیرد.

با توجه به اینکه استاندارد ISO/IEC 12207 به‌عنوان پایه و اساس الگو در نظر گرفته شده است برای هرچه بهتر پیاده‌سازی منبع (Data Pro 12207) قابل بهره‌برداری است. با توجه به ساختار کلی این استاندارد چندین وجه را می‌توان برای پیاده نمودن این استاندارد در نظر گرفت که انتخاب این وجوه به عوامل مختلفی اعم از فرهنگ، منابع، تکنیک و برنامه‌ریزی بستگی دارد. برخی از توصیه‌های لازم جهت استقرار ۱۲۲۰۷ به شرح زیر هستند:

- تعیین اینکه چه فعالیت‌ها و وظایفی از ISO/IEC 12207 برای سازمان ضروری است و نیازها را می‌پوشاند.

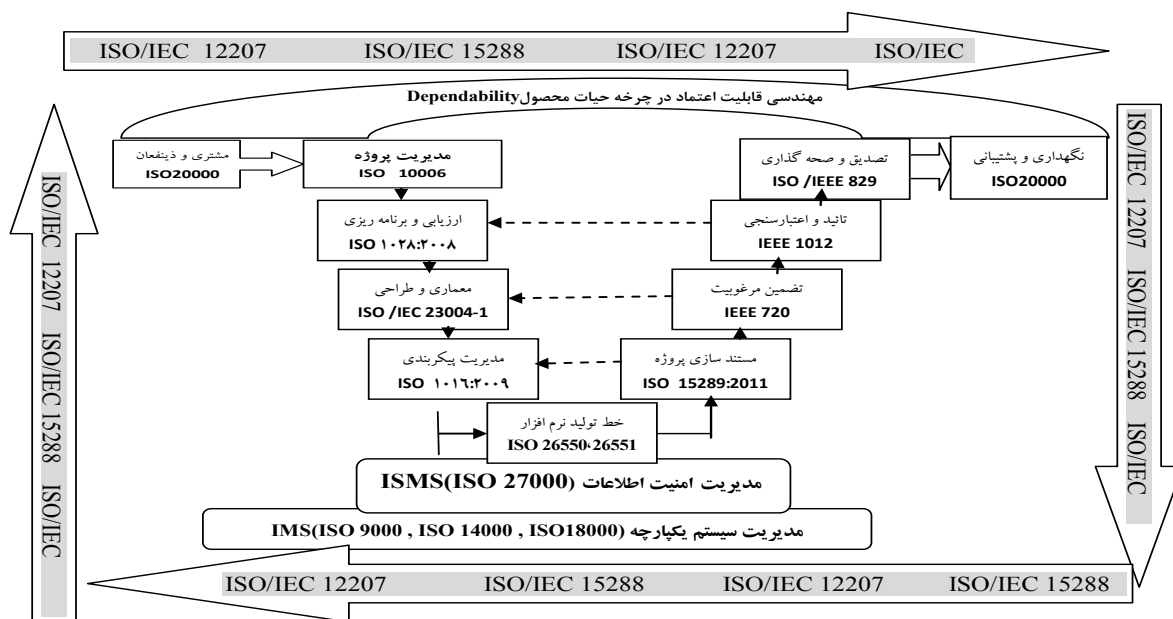
- انطباق فعالیت‌ها و وظایف انتخاب شده از ISO/IEC 12207 با فعالیت‌ها و پروسه‌های جاری سازمان و تعیین فعالیت‌هایی که در سازمان انجام نمی‌گیرد و یا به‌صورت کامل اجرا نمی‌شود.

- جلب موافقت مدیر اصلی بسیار مهم می‌باشد زیرا در مرحله پیاده‌سازی ISO/IEC 12207 اعمال تغییرات براساس نیازهای ISO/IEC 12207 در سطوح مختلف سازمانی قابل انجام خواهد بود.

- آماده‌سازی یک طرح از منابع و برنامه‌ریزی برای ایجاد وظایف و شناخت دقیق پیاده‌سازی وظایف جدید بر مبنای ISO/IEC 12207 بسیار مهم است.

- آموزش همه سطوح سازمان. طراحی سطوح مختلف آموزش از سطح آگاهی‌های عمومی تا سطح جزئیات وظایف، ضروری است. آموزش و ارتباطات کلید موفقیت پیاده‌سازی ISO/IEC 12207 می‌باشد.

- اطلاع‌رسانی به کل سازمان وقتی پروژه با موفقیت به پایان می‌رسد. در این مرحله می‌توان مقایسه‌ای بین مزایای واقعی و مزایای پیش‌بینی شده استاندارد ISO/IEC 12207 تهیه نمود.



شکل ۴: الگوی پیشنهادی استانداردسازی در چرخه‌ی حیات نرم‌افزار

با رعایت اصول فرآیندگرایی و چرخه‌ی دمینگ در هر مرحله، تعریف مسأله و شناسایی نیازهای مشتری، تغییرات نیازها و اثرات تغییرات نیاز در فرایند شناخت، طراحی، برنامه‌نویسی و آزمایش برنامه‌ها و سیستم و همچنین کنترل و مدیریت ویرایش‌های مختلف نرم‌افزار عرضه شده به مشتری و ... همگی تحت پوشش و کنترل قرار خواهند داشت.

۸) بحث و نتیجه گیری

اهمیت سامانه‌های نرم‌افزاری و حساسیت مشخصه‌هایی از قبیل قابلیت اطمینان، امنیت، ایمنی و در دسترس بودن آن‌ها، سبب شده است تا توجه به استانداردسازی در چرخه‌ی حیات این سامانه‌ها امری ضروری و اجتناب‌ناپذیر شود. در این مقاله ابتدا مشکلات و چالش‌های حوزه‌ی تولید و توسعه‌ی نرم‌افزار مورد تحلیل و بررسی قرار گرفت. در گام دوم با تشریح وضعیت موجود و تعریف کیفیت محصولات نرم‌افزاری به‌عنوان شاخصی برای رسیدن به وضع مطلوب طرح موضوع گردید. در گام سوم، روش‌شناسی نرم‌افزار به‌عنوان عنصر کلیدی برای پیش‌بینی و همچنین ایجاد

تغییرات اساسی و بنیادی در محصول پرداخته و با مقایسه‌ی مزایا و معایب روش‌شناسی‌های متداول، روش‌شناسی ۷ برای ارائه‌ی مدل پیشنهاد گردید. در گام چهارم، استانداردسازی محصول به‌عنوان مکملی برای ارتقاء کیفیت محصول نرم‌افزاری اشاره شد. اشراف بر روش‌شناسی‌های نرم‌افزار و رعایت اصول استاندارد برای تحقق بهبود کیفیت لازم و ملزوم بوده و لیکن کافی نیستند. در این راستا نیاز به ابزاری پیرامونی به‌منظور ایجاد هماهنگی و رشد متوازن و رعایت عملکرد زنجیروار مراحل مختلف است که چرخه‌ی حیات ISO/IEC 12207 برای حوزه‌ی نرم‌افزار معرفی گردید. مطالعات کتابخانه‌ای انجام‌شده همچنین نشان می‌دهد که هر یک از فعالیت‌های چرخه‌ی حیات بسته به ماهیت و نوع کاربرد نرم‌افزار می‌تواند بیشتر یا کمتر مورد توجه طراحان و برنامه‌نویسان قرار گیرد، لیکن در هر فعالیت حداقل الزامات باید رعایت شود که این الگو راهنمای مناسبی برای تعیین چارچوب حداقل الزامات هر فرایند است.

با رعایت موارد فوق‌الذکر دغدغه‌های مطرح شده در پروژه (کیفیت، امنیت، زمان، هزینه) کاهش و

یا مرتفع خواهند شد. کیفیت استاندارد ISO 6030 (مهندسی قابلیت اعتماد در چرخه‌ی حیات محصول) به مانند چتری بر تمامی مراحل تولید تا پشتیبانی است. همچنین استقرار سیستم مدیریت امنیت اطلاعات (ISMS) در سازمان منجر به استانداردسازی فرایندها و محصول می‌شود. استفاده از استانداردهای مورد نیاز خانواده ISO 27000، متضمن امن‌سازی محصول و خدمات خواهند بود. ممکن است استقرار چنین الگویی در ابتدا به لحاظ صرف زمانی و نیز هزینه‌ی اولیه توجیه فنی و اقتصادی نداشته باشد. لیکن با کمی ژرفنگری و در نظر گرفتن مزایای ناشی از برقراری الزامات کیفی کار و نیز جلب رضایت مشتریان و پاسخ‌دهی مناسب و سریع به نیازهای آنان در بلند مدت، نه تنها توجیه اقتصادی خواهد داشت بلکه در عین حال امکان توسعه و بسط بازارهای

داخلی و حرکت به سمت بازارهای بین‌المللی را مهیا می‌سازد. الگوی پیشنهادی کمک می‌کند تا درک مشترکی از فعالیت‌های چرخه‌ی حیات سامانه‌ی نرم‌افزاری ایجاد شود. این درک و ادبیات مشترک به کارفرمایان و کاربران نهایی نرم‌افزار کمک می‌کند تا بتوانند خواسته‌های خود را به‌درستی مطرح کرده و نرم‌افزار مورد نظر خود را به‌درستی تحویل بگیرند. همچنین این الگو به توسعه‌دهندگان نرم‌افزار کمک می‌کند تا با طرح‌ریزی فرایندهای مناسب در محیط توسعه خود، الزامات مشتریان را برآورده کرده و زیرساخت‌های مورد نیاز برای پشتیبانی محصولات خود را تأمین کرده و نهایتاً بقای خود در بازار رقابتی توسعه‌ی نرم‌افزار را تضمین کنند.

۹) منابع و مراجع

۱. معماری فرایند تحقق ایده (از ایده تا ساخت نمونه) - الزامات عمومی IDS 1107:2012
۲. بهکمال، بهشید، کاهانی، محسن (۱۳۸۸). "ارائه‌ی یک روش مبتنی بر مدل کیفیت برای ارزیابی سیستم‌های تجارت الکترونیکی" مجموعه مقالات همایش تجارت الکترونیکی، دانشگاه تهران.
۳. ع. احمدی جوقی، مدیریت پروژه های نرم‌افزاری/نویسنده مایک و باب هیوز، تهران: انستیتوایزایران، ۱۳۷۸
۴. ا. هدایت‌فر، "روش فراگیری آسان یکپارچه‌ی منطقی" نویسنده کرول پر، تهران مؤسسه‌ی فرهنگی هنری دیباگران سال ۱۳۸۴

5. Software Engineering, Sommerville – Ninth Edition
6. Boehm, B.W, Brown, J.R., Kaspar, H, Lipow, M., McLeod, G. and Merritt, M. (1978). "Characteristics of Software Quality", North Holland.
7. ISO/IEC 15288:2008 systems and software engineering – system life cycle processes
8. ISO/IEC 12207:2008 systems and software engineering – system life cycle processes
9. ISO/IEC 8402:2008 Quality Management and Quality Assurance