Passive Defence
Vol. 15, No. 4, Winter 2024, Serial No. 60
ISSN: 2980-8030 , E-ISSN: 2008-6849

Improving Software Quality Assurance Using Common International Standards

Ali Karimi, *, Ali Tolui Far
*Assistant Professor, Imam Hossein (AS) University, Tehran, Iran
(Received: 03/06/2024, Revised: 27/07/2024, Accepted: 22/09/2024, Published: 15/12/2024)
DOR: 20.1001.1.20086849.1403.15.4.6.9

ABSTRACT

The software development and maintenance environment is called as software quality assurance environment. In today's world where numerous complex and critical software applications are used in daily life, ensuring the quality and reliability, or in other words, the quality assurance of these products is essential. On the other hand, to increase the likelihood of projects' success, utilizing software quality standards to guide the quality assurance process is of great importance. Passive defense in software refers to a set of measures taken to enhance security and reduce software vulnerabilities against threats. Software quality assurance standards, such as ISO 9001 and ISO 25010, emphasize the development of software with features like security, reliability, and flexibility. These features align with the objectives of passive defense, as secure and flexible software can resist various threats and ensure continued functionality. ISO standards such as ISO-9001 and ISO-25010, as well as IEEE standards like IEEE 730 and IEEE 828, define basic concepts and key principles for software quality assurance. These standards play a significant role in the software quality assurance process by improving the performance and quality of software. In this article, we examine 8 common and widely used software quality assurance standards, categorized into two groups: "Project Process Standards" and "Quality Management Standards." This article can be utilized by software development organizations and software project managers to enhance the quality of their products.

Keywords: Software Standards, Software Quality Assurance, Software Development, Software Quality Assurance Environment, International Standards, Project Process, Quality Management

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Publisher: Imam Hussein University © Authors

* Corresponding Author Email: a.karimi@ihu.ac.ir

پدافند غیرعامل

سال پانزدهم، شماره ۴، زمستان ۱۴۰۳، (پیاپی ۶۰): صص ۹۷-۷۷

شاپای چاپی: ۶۹۴۹-۲۰۰۸ | **شاپای الکترونیکی:** ۸۰۳۰-۲۹۸۰

علمی – ترویجی

بهبود تضمین کیفیت نرم‌افزار با استفاده از استانداردهای رایج جهانی

علی کریمی[1*]، علی طلوعی‌فر[2]

DOR: 20.1001.1.20086849.1403.15.4.6.9

تاریخ دریافت: ۱۴۰۳/۰۳/۱۴ تاریخ پذیرش: ۱۴۰۳/۰۷/۰۱

تاریخ بازنگری: ۱۴۰۳/۰۵/۰۶ تاریخ انتشار: ۱۴۰۳/۰۹/۲۵

چکیده

محیط توسعه و نگهداری نرم‌افزار، به‌عنوان محیط تضمین کیفیت نرم‌افزار نامیده می‌شود. در دنیای امروز که نرم‌افزارهای پیچیده و حیاتی بسیاری در زندگی روزمره استفاده می‌شوند، اطمینان از کیفیت و قابلیت اطمینان و به عبارتی تضمین کیفیت این محصولات امری ضروری است. ازسوی‌دیگر، برای افزایش احتمال موفقیت پروژه‌ها، استفاده از استانداردهای کیفیت نرم‌افزار برای هدایت فرایند تضمین کیفیت، بسیار حائز اهمیت است. پدافند غیرعامل در نرم‌افزار به مجموعه اقداماتی اشاره دارد که برای افزایش امنیت و کاهش آسیب‌پذیری نرم‌افزار در مقابل تهدیدات، انجام می‌شود. استانداردهای تضمین کیفیت نرم‌افزار همچون ISO 9001 و ISO 25010، بر ایجاد نرم‌افزارها با ویژگی‌هایی مانند امنیت، قابلیت اطمینان و انعطاف‌پذیری تأکید دارند. این ویژگی‌ها، همسو با اهداف پدافند غیرعامل هستند؛ زیرا نرم‌افزارهای امن و انعطاف‌پذیر می‌توانند در مقابل تهدیدات مختلف مقاوم بوده و امکان ادامه فعالیت را فراهم کنند. استانداردهای ISO مانند ISO 9001 و ISO 25010 و نیز استانداردهای IEEE از قبیل IEEE 730 و IEEE 829، مفاهیم مبنایی و اصول کلیدی را برای تضمین کیفیت نرم‌افزار تعریف می‌کنند. این استانداردها در فرایند تضمین کیفیت نرم‌افزار، برای بهبود عملکرد و کیفیت نرم‌افزار، نقش قابل‌توجهی ایفا می‌کنند. در این مقاله، به بررسی ۸ استاندارد رایج و متداول تضمین کیفیت نرم‌افزار در دو دسته‌بندی «استانداردهای فرایند پروژه» و «استانداردهای مدیریت کیفیت» می‌پردازیم. این مقاله می‌تواند توسط سازمان‌های توسعه‌دهنده نرم‌افزار و مدیران پروژه‌های نرم‌افزاری، در جهت ارتقای کیفیت محصولات خود مورد استفاده قرار گیرد.

کلیدواژه‌ها: استانداردهای نرم‌افزار، تضمین کیفیت نرم‌افزار، توسعه نرم‌افزار، محیط تضمین کیفیت نرم‌افزار، استانداردهای بین‌المللی، فرایند پروژه، مدیریت کیفیت.

[1] استادیار دانشگاه جامع امام حسین (ع)، تهران، ایران (a.karimi@ihu.ac.ir)- نویسنده مسئول

[2] دانشجوی کارشناسی ارشد مهندسی کامپیوتر، دانشگاه جامع امام حسین (ع)، تهران، ایران

* این مقاله یک مقاله با دسترسی آزاد است که تحت شرایط و ضوابط مجوز Creative Commons Attribution (CC BY) توزیع شده است.

© نویسندگان

ناشر: دانشگاه جامع امام حسین (ع)

۱- مقدمه

ظهور عصر دیجیتال باعث گسترش سریع سامانه‌های نرم‌افزاری شده است و آنها را با هر جنبه‌ای از زندگی مدرن درهم می‌آمیزد. ازآنجایی‌که جهان دستخوش یک تحول عمیق دیجیتالی می‌شود، تقاضا برای نرم‌افزار باکیفیت بالا هرگز به این اندازه حیاتی نبوده است. کیفیت نرم‌افزار فقط یک ویژگی مطلوب نیست، بلکه یک ستون اساسی است که رضایت کاربر، قابلیت اطمینان سامانه و موفقیت کلی کسب‌وکار را پایه‌ریزی می‌کند. در میان پیچیدگی توسعه سامانه‌های نرم‌افزاری بزرگ-مقیاس، معماران و طراحان با وظیفه چالش‌برانگیز تضمین کیفیت نرم‌افزار روبه‌رو هستند، درحالی‌که نیازها و نیازمندی‌های در حال تکامل کاربر را برآورده می‌کنند [۱].

تضمین کیفیت نرم‌افزار[1] بخش مهم و جدایی‌ناپذیر فرایند توسعه نرم‌افزار محسوب می‌شود. در طول تکامل روش‌های توسعه نرم‌افزار، عوامل متعددی به اهمیت نقش تضمین کیفیت نرم‌افزار کمک کرده‌اند. افزایش اتصال اینترنت به بسیاری از برنامه‌های کاربردی نرم‌افزاری یک عامل، کمی قدیمی، اما مهم است؛ زیرا نیازمند توجه بیشتر به تجزیه‌وتحلیل محصولات برای ویژگی‌های امنیتی و عملکرد است. عامل دوم، افزایش تقاضا برای انتشار نرم‌افزارهای مکرر، نیاز به بهینه‌سازی فرایندهای اعتبارسنجی و تأیید را افزایش می‌دهد تا محصول سریع‌تر و باکیفیت بالاتر تحویل داده شود [۲].

به‌طورکلی، یک مهندس عالی SQA باید مجموعه‌ای از توانایی‌های متفاوت از مهندسان توسعه را داشته باشد تا به طور مؤثر بر کل فرایند توسعه محصول از ابتدا تا انتها نظارت کند. رون پتون[2] شرح زیر را در مورد یک متخصص SQA ذکر کرد: «مسئولیت اصلی یک فرد تضمین‌کننده کیفیت نرم‌افزار بررسی و اندازه‌گیری فرایند توسعه فعلی و یافتن راه‌هایی برای بهبود آن با هدف جلوگیری از بروز خطا است.» [۲].

سازمان‌ها توجه زیادی به توسعه نرم‌افزار دارند؛ زیرا محصولات با کیفیت بالا نیازهای مشتریان را برآورده می‌کنند، از شهرت و اعتبار آنها محافظت می‌کنند و عملیات تجاری را بهبود می‌بخشند [۳]. یک مطالعه اخیر نشان می‌دهد که نرخ موفقیت در پروژه‌های نرم‌افزاری ۳۷٪ است و در طول توسعه نرم‌افزار، تلاش‌های بیشتری باید انجام شود [۴].

روش SQA شامل فعالیت‌های مختلفی است که در طول فرایند توسعه نرم‌افزار انجام می‌شود. هنگامی که این فعالیت‌ها به‌خوبی انجام شوند، نتایج، سامانه‌های نرم‌افزاری هستند که الزامات و انتظارات را باکیفیت و رضایتمندی بیشتر کاربر فراهم می‌کنند. مزایای مختلفی از توسعه شیوه‌های معتبر و مطرح (به‌خوبی تثبیت‌شده) SQA مانند کاهش هزینه‌های توسعه و آزمون، بهبود کیفیت و بهره‌وری سامانه‌های نرم‌افزاری توسعه‌یافته و افزایش رضایتمندی مشتری وجود دارد [۵].

فعالیت‌های تضمین کیفیت نرم‌افزار شامل برنامه‌ریزی، تعریف و اجرای استانداردهای کیفیت، ممیزی و بازبینی[3]، آزمون و گزارش‌دهی درباره نتایج می‌باشد. اعتبارسنجی نیازمندی‌ها به‌عنوان سنگ‌بنایی در نظر گرفته می‌شود؛ با استفاده از عناصر مهمی نظیر نمونه‌سازی[4]، بازرسی، مبتنی بر دانش، مبتنی بر آزمون، مدل‌سازی و ارزیابی، و مدل‌های رسمی [۶].

طبق گزارش Chaos در سال ۲۰۲۰، تنها ۳۵٪ از پروژه‌ها از نظر زمان و بودجه کاملاً موفق بوده‌اند. اتخاذ شیوه‌های خوب برای انجام فرایند توسعه نرم‌افزار جایگزین مناسبی برای افزایش احتمال موفقیت است. اقدامات خوب شامل به‌کارگیری دستورالعمل‌ها برای انجام هر مرحله از چرخه‌حیات نرم‌افزار و همچنین به‌کارگیری استانداردهای بین‌المللی است. این استانداردها در سطح بین‌المللی مورد توافق کارشناسان قرار گرفته و توسط سازمان‌های مهم صنعت نرم‌افزار ترویج داده می‌شوند. به‌عنوان‌مثال، سازمان بین‌المللی استاندارد[5] همراه با کمیسیون بین‌المللی الکتروتکنیک[6]، برخی از مهم‌ترین استانداردهای بین‌المللی را ارائه می‌کنند. به‌طورمشابه، موسسه مهندسی نرم‌افزار[7] و موسسه مهندسان برق و الکترونیک[8]، دو سازمان مرتبط در تعریف استانداردهای بین‌المللی هستند. استاندارد سندی است که توسط یک مقام، عرف یا موافقت عمومی به عنوان یک الگو یا نمونه تعیین می‌شود [۷].

استانداردهای مهندسی نرم‌افزار منابع دانش مدون هستند. مطالعات نشان داده‌اند که استفاده از استانداردها مزایایی مانند قابلیت همکاری محصول، افزایش بهره‌وری، افزایش سهم بازار و بهبود تعامل با ذینفعان نظیر شرکت‌ها، سازمان‌های دولتی و عموم مردم را به همراه دارد. استانداردها و اسناد فنی مرتبط را می‌توان نوعی انتقال فناوری در نظر گرفت و در صورت انتخاب و استفاده صحیح از استانداردها، باید تأثیرات اقتصادی در سازمان داشته باشند [۸].

پیاده‌سازی استانداردهای مهندسی نرم‌افزار در سازمان‌های بسیار کوچک از اهمیت بالایی برخوردار است؛ زیرا آنها درصد قابل‌توجهی از سازمان‌های نرم‌افزاری در سراسر جهان را نشان

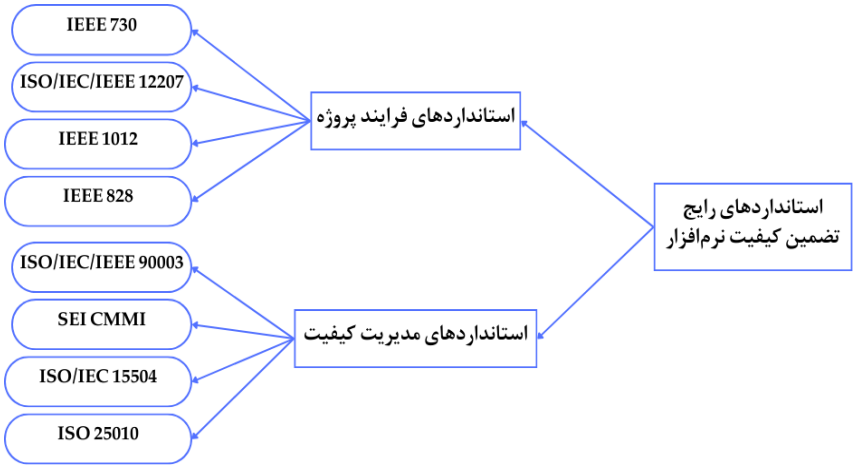
[1] Software Quality Assurance (SQA)
[2] Ron Patton
[3] Reviewing and auditing
[4] Prototyping
[5] International Organization for Standardization (ISO)
[6] International Electrotechnical Commission (IEC)
[7] Software Engineering Institute (SEI)
[8] Institute of Electrical and Electronics Engineers (IEEE)

می‌دهند. آنها اغلب تأمین‌کنندگان شرکت‌های کوچک و متوسط و سازمان‌های بزرگ‌تر هستند؛ بنابراین، توسعه محصولات یا خدمات باکیفیت بالا برای بقا و رشد آنها ضروری است [۸].

مهندسی نرم‌افزار، مانند سایر رشته‌های مهندسی، مبتنی بر استفاده از شیوه‌های کاملاً تعریف‌شده برای اطمینان از کیفیت محصولات یا خدمات ارائه‌شده است. مجموعه گسترده‌ای از استانداردهای IEEE و ISO وجود دارد که تمام جنبه‌های توسعه چرخه حیات نرم‌افزار، نگهداری و مدیریت را پوشش می‌دهد. بیش از ۲۰۰ استاندارد ISO در زمینه سامانه‌ها و مهندسی نرم‌افزار منتشر شده‌اند که توسط کارشناسان بیش از ۶۰ کشور و سازمان‌های حرفه‌ای مانند IEEE توسعه داده شده‌اند. به‌عنوان‌مثال، استاندارد ISO/IEC/IEEE 12207 فرایندهایی را برای یک سازمان یا پروژه فراهم می‌کند که قابل استفاده برای تعریف، کنترل و بهبود فرایندهای چرخه حیات نرم‌افزار هستند. همچنین، استاندارد ISO/IEC/IEEE 29148 فرآیندهایی را فراهم می‌کند که منجر به تعیین نیازمندی‌ها در طول چرخه حیات می‌شوند [۸].

باتوجه‌به مطالعات انجام‌شده در حوزه متداول‌ترین استانداردهای تضمین کیفیت نرم‌افزار با بررسی مقالات پیشین، آرایه‌شناسی استانداردها در دو دسته استانداردهای فرایند پروژه و استانداردهای کنترل کیفیت به‌صورت شکل (۱) ارائه می‌شود. در این شکل، ۸ استاندارد موجود و متداول تضمین کیفیت نرم‌افزار در دو دسته جداگانه بررسی شده‌اند.

بخش باقی‌مانده از کار به شرح زیر تنظیم شده است. بخش دوم، مفهوم تضمین کیفیت نرم‌افزار را مورد بحث قرار می‌دهد. در بخش سوم، مفهوم استانداردهای تضمین کیفیت نرم‌افزار در دو دسته فرایند پروژه و کنترل کیفیت بررسی و اصول و رویکردهای هر یک از ۸ استاندارد، ارائه خواهد شد. بخش چهارم شامل نتیجه‌گیری است.



شکل (۱): آرایه‌شناسی استانداردهای رایج و متداول تضمین کیفیت نرم‌افزار

۲- تضمین کیفیت نرم‌افزار

در این بخش ابتدا با مفاهیم نرم‌افزار، خطا[1]، نقص[2]، شکست[3]، کیفیت نرم‌افزار[4] آشنا شده و سپس مفهوم تضمین کیفیت نرم‌افزار مورد بررسی قرار خواهد گرفت.

مطابق تعریف ISO 24765 نرم‌افزار عبارت است از [۹]:

۱) تمام یا بخشی از برنامه‌ها، رویه‌ها، قوانین و مستندات مربوط به یک سامانه پردازش اطلاعات.

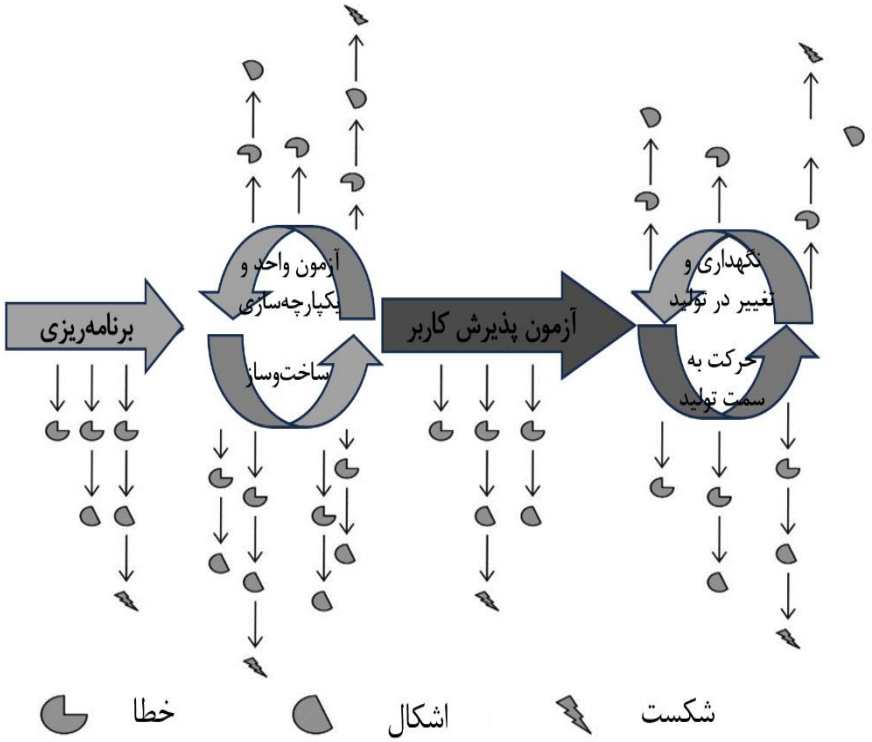
۲) برنامه‌های رایانه‌ای، رویه‌ها، و احتمالاً اسناد و داده‌های مرتبط با عملکرد یک سامانه رایانه‌ای.

طبق استاندارد ISO 24765 خطا یعنی عمل انسانی که نتیجه نادرستی به بار می‌آورد. نقص شامل دو تعریف می‌باشد که تعریف اول مطابق با استاندارد ISO 24765 و تعریف دوم مربوط به هیئت بین‌المللی صلاحیت آزمون نرم‌افزار[5] می‌باشد [۹]:

۱) یک مشکل (مترادف خطا) که در صورت عدم تصحیح، می‌تواند باعث ازکارافتادن برنامه یا نتایج نادرست شود.

۲) یک کاستی یا عیب در یک نرم‌افزار یا جزء سامانه که می‌تواند منجر به عدم انجام عملکرد آن مؤلفه شود، به‌عنوان‌مثال تعریف نادرست داده‌ها یا دستورالعمل کد منبع. یک نقص، اگر اجرا شود، می‌تواند باعث خرابی یک نرم‌افزار یا مؤلفه سامانه شود.

[1] Error
[2] Defect
[3] Failure
[4] Software quality
[5] International Software Testing Qualifications Board (ISTQB)



شکل (۲): خطاها، نقص‌ها و شکست‌ها در چرخه حیات نرم‌افزار [۹]

بر اساس استاندارد ISO 25010 شکست عبارت است از [۹]: خاتمه توانایی یک محصول برای انجام یک عملکرد موردنیاز یا ناتوانی آن در انجام در محدوده‌های مشخص‌شده قبلی.

خطا، نقص و شکست در شکل (۲) نشان داده شده است [۹].

کیفیت نرم‌افزار به مدت طولانی یک نگرانی مهم در توسعه نرم‌افزار بوده و برای ارزیابی موفقیت محصولات نرم‌افزاری استفاده شده است [۱۰]. کیفیت پایین به‌ویژه، در سامانه‌های حیاتی و بی‌درنگ می‌تواند منجر به ضرر مالی، ناتوانی دائمی، شکست در انجام ماموریت یا خسارات جبران‌ناپذیر شود [۱۱].

استاندارد ISO 8402:1986، کیفیت نرم‌افزار را این‌گونه تعریف می‌کند: «مجموعه ویژگی‌ها و مشخصات یک محصول یا خدمت که بر توانایی آن در برآوردن نیازهای مشخص یا ضمنی تأثیر می‌گذارد». استاندارد ISO 9001:2008 این دیدگاه را به شرح زیر گسترش داد:« کیفیت چیزی را می‌توان با مقایسه مجموعه‌ای از ویژگی‌های ذاتی با مجموعه‌ای از نیازمندی‌ها تعریف کرد. کیفیت بالا یا عالی در صورتی حاصل می‌شود که ویژگی‌های ذاتی همه نیازمندی‌ها را برآورده کنند. اگر آن ویژگی‌ها همه نیازمندی‌ها را برآورده نکنند، سطح کیفیت پایین یا ضعیفی حاصل می‌شود» [۱۲]. چندین مدل کیفیت و ویژگی برای اندازه‌گیری کیفیت نرم‌افزار پیشنهاد شده است [۱۳-۱۵].

تعریف IEEE از کیفیت نرم‌افزار به شرح زیر است [۱۶]:

۱) درجه‌ای که یک سامانه، مؤلفه یا فرایند مطابق با نیازمندی‌های مشخص‌شده عمل کند.

۲) درجه‌ای که یک سامانه، مؤلفه یا فرایند مطابق با نیازمندی‌ها یا انتظارات مشتری و یا کاربر عمل کند.

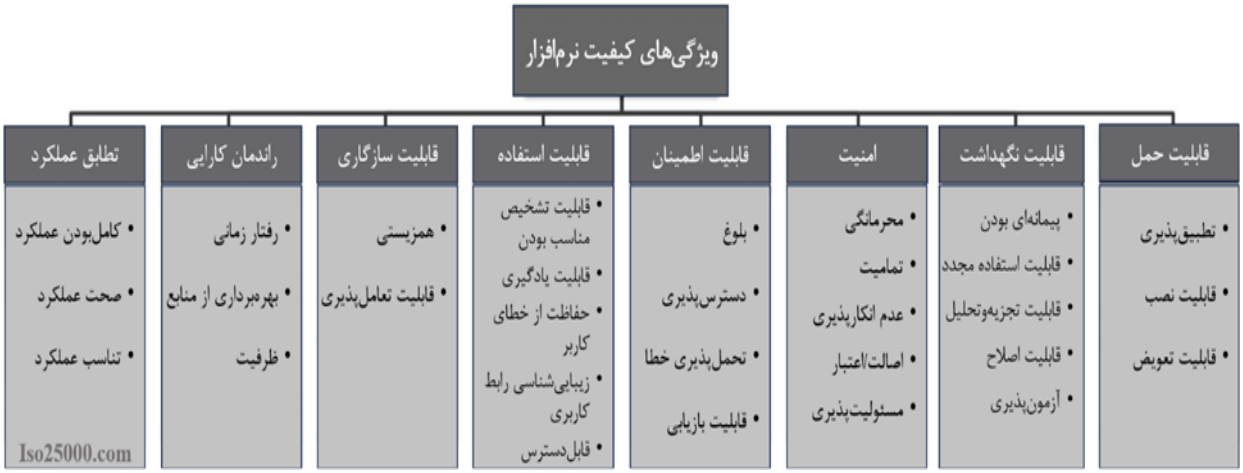
تضمین کیفیت مطابق تعریف استاندارد ISO 24765 عبارت است از [۹]:

۱) یک الگوی برنامه‌ریزی‌شده و نظام‌مند از کلیه اقدامات لازم برای ایجاد اطمینان کافی مبنی بر انطباق یک کالا یا محصول با الزامات فنی تعیین‌شده.

۲) مجموعه‌ای از فعالیت‌ها که برای ارزیابی فرایند توسعه یا تولید محصولات طراحی شده‌اند.

۳) فعالیت‌های برنامه‌ریزی‌شده و نظام‌مند که در چارچوب سامانه کیفیت اجرا شده و به میزان لازم به نمایش درآمده‌اند تا اطمینان کافی ارائه دهند که یک موجودیت نیازهای کیفیت را برآورده خواهد کرد.

کیفیت نرم‌افزار دارای هشت ویژگی سطح بالا مطابق با استاندارد ISO/IEC 25010 است که هر یک از آنها از مجموعه‌ای از ویژگی‌های فرعی مرتبط تشکیل شده است که در شکل (۳) قابل مشاهده است [۱۷].



شکل (۳): آرایه‌شناسی ویژگی‌های کیفیت نرم‌افزار بر اساس استاندارد ISO/IEC 25010 [۱۷]

هدف برنامه‌های توسعه نرم‌افزار مدیریت یک نرم‌افزار در جهت توسعه یک محصول است؛ بنابراین، یک برنامه مدیریت پروژه باید وجود داشته باشد که فرایندهای فنی و مدیریتی لازم برای برآورده‌کردن الزامات پروژه را تعریف کند. الزامات شامل انطباق با استانداردها و الزامات قانونی برای برآوردن هدف نهایی توسعه نرم‌افزار باکیفیت مناسب برای استفاده از آن در سامانه‌های ایمن است [۱۸].

بر اساس تعریف ارائه‌شده توسط استاندارد IEEE Std. 730-2014، تضمین کیفیت نرم‌افزار یک الگوی برنامه‌ریزی‌شده و نظام‌مند از تمام اقدامات لازم برای ایجاد اطمینان کافی در مورد مطابقت کالا یا محصول با الزامات فنی تعیین شده است. تضمین کیفیت[1] باید در طول تمام فعالیت‌های توسعه‌ای به‌کار گرفته شود تا در اعلام اینکه محصول نیازهای مشخص‌شده را برآورده می‌کند، سطح بالایی از اطمینان فراهم شود [۱۸].

متداول‌ترین تعریف از تضمین کیفیت نرم‌افزار، تعریفی است که توسط IEEE پیشنهاد شده است که عبارت است از [۱۶]:

- یک الگوی برنامه‌ریزی‌شده و نظام‌مند از همه اقدامات ضروری برای فراهم‌کردن اطمینان کافی که یک محصول مطابق نیازمندی‌های فنی بنا شود.
- یک مجموعه از فعالیت‌های طراحی‌شده برای ارزیابی فرایند توسعه یا ساخت محصولات.

تعریف توسعه‌یافته تضمین کیفیت نرم‌افزار، SQA عبارت است از [۱۶]:

- یک مجموعه نظام‌مند و برنامه‌ریزی‌شده از اقدامات لازم برای حصول اطمینان کافی از اینکه:
- فرایند توسعه نرم‌افزار یا فرایند نگهداری نرم‌افزار با موارد زیر منطبق است:
  - ✓ نیازمندی‌های فنی وظیفه‌مندی
  - ✓ نیازمندی‌های مدیریتی با حفظ زمان‌بندی
  - ✓ عملیات در محدوده بودجه تعیین‌شده

این تعریف توسعه‌یافته، قویاً با موارد زیر مرتبط است:

- مفاهیم استانداردهای ISO 9000 در مورد SQA
- طرح اصلی مدل بلوغ قابلیت[2] برای نرم‌افزار (این مدل می‌تواند در راستای هدایت و بهبود فرایندها در یک پروژه، بخش یا یک سازمان، استفاده شود).

مقایسه تعریف توسعه‌یافته تضمین کیفیت نرم‌افزار با نسخه‌های دیگر در جدول (۱) گفته شده است.

فرایند تضمین کیفیت نرم‌افزار نیازمند اقدامات برنامه‌ریزی‌شده و نظام‌مند است که برای ایجاد اطمینان کافی در مورد مناسب‌بودن فرایندهای توسعه نرم‌افزار و تولید محصول نرم‌افزاری باکیفیت مناسب برای هدف موردنظر ضروری است. فرایند SQA با تأیید اینکه طرح‌ها (از جمله طرح مدیریت پروژه) طبق استانداردها توصیف شده‌اند، رویه‌ها بر اساس برنامه‌ها اجرا می‌شوند و همچنین محصولات (نرم‌افزار) مطابق با استانداردها توسعه یافته‌اند، این امر را تضمین می‌کند [۱۸].

همکاری بین مهندسین نرم‌افزار و تیم‌های SQA، یک راه‌حل مناسب برای دستیابی به فعالیت‌های توسعه، نگهداری کارآمد و اقتصادی نرم‌افزار است؛ درعین‌حال، تضمین کیفیت محصولات را در پی دارد [۱۶].

[1] Quality Assurance (QA)

[2] Capacity Maturity Model (CMM)

جدول (۱): مقایسه تعریف توسعه‌یافته تضمین کیفیت نرم‌افزار در مقایسه با نسخه‌های دیگر [۱۶]

ردیف	تعریف توسعه‌یافته SQA	تعریف SQA توسط IEEE	بخش‌های مربوط به استاندارد ISO 9000-3	نیازمندی‌های مرتبط SEI-CMM
۱	اقدامات نظام‌مند و برنامه‌ریزی‌شده موردنیاز است	+	مسئولیت‌های مدیریتی (۴,۱) سامانه کیفیت (۴,۲) بازبینی قرارداد (۴,۳)	✓ مدیریت کیفیت نرم‌افزار ✓ مدیریت نیازمندی ✓ برنامه‌ریزی پروژه ✓ پیگیری و نظارت بر نرم‌افزار
۲	سروکار داشتن با فرایند توسعه نرم‌افزار	+	بررسی و بازبینی قرارداد (۴,۳) کنترل طراحی (۴,۴) کنترل محصولات عرضه‌شده به مشتری (۴,۷) کنترل فرایند (۴,۹) بازرسی و آزمون (۴,۱۰) کنترل محصول نامنطبق (۴,۱۳) کنترل سوابق و مدارک کیفیت (۴,۱۶) فنون آماری (۴,۲۰)	• مدیریت نیازمندی‌ها • برنامه‌ریزی پروژه‌های نرم‌افزاری • پیگیری و نظارت نرم‌افزار • مدیریت پیکره‌بندی نرم‌افزار • مهندسی محصولات نرم‌افزاری • بازبینی همتایان • مدیریت قراردادهای فرعی نرم‌افزار • مدیریت کیفیت نرم‌افزار
۳	سروکار داشتن با فرایند نگهداری نرم‌افزار	-	بازبینی قرارداد - دغدغه مدیریتی (۴,۳,۲c) کنترل فرایند (۴,۹) سرویس‌دهی (۴,۱۹) فنون آماری (۴,۲۰)	○ مدیریت نیازمندی ○ برنامه‌ریزی پروژه‌های نرم‌افزاری ○ پیگیری و نظارت نرم‌افزار ○ مهندسی محصولات نرم‌افزاری ○ مدیریت کیفیت نرم‌افزار
۴	سروکار داشتن با نیازمندی‌های فنی وظیفه‌مندی	+	بازبینی قرارداد (۴,۳) کنترل طراحی (۴,۴) کنترل محصول عرضه‌شده به مشتری (۴,۷) بازبینی و آزمون (۴,۱۰) کنترل محصول نامنطبق (۴,۱۳)	■ مدیریت نیازمندی‌ها ■ برنامه‌ریزی پروژه‌های نرم‌افزاری ■ پیگیری و نظارت نرم‌افزار ■ مدیریت پیکره‌بندی نرم‌افزار ■ مهندسی محصولات نرم‌افزاری ■ بازبینی همتایان ■ مدیریت قراردادهای فرعی نرم‌افزار
۵	سروکار داشتن با نیازمندی‌های زمان‌بندی	-	نگرانی‌های مدیریت بازبینی قرارداد (۴,۳,۲c) شناسایی برنامه زمان‌بندی (۴,۴,۲g) بررسی تأمین‌کنندگان از پیشرفت توسعه نرم‌افزار (۴,۴,۳) شناسایی برنامه زمان‌بندی (۴,۴,۲g)	▫ مدیریت نیازمندی‌ها ▫ برنامه‌ریزی پروژه نرم‌افزاری ▫ پیگیری و نظارت بر نرم‌افزار
۶	سروکار داشتن با کنترل-های بودجه‌ای	-	شناسایی برنامه زمان‌بندی (۴,۴,۲g)	◆ مدیریت نیازمندی‌ها ◆ برنامه‌ریزی پروژه نرم‌افزاری ◆ پیگیری و نظارت بر نرم‌افزار

۳- استانداردهای تضمین کیفیت نرم‌افزار

امروزه کیفیت سامانه‌های نرم‌افزاری برای شرکت‌ها در ارائه خدمات و محصولات باکیفیت بسیار مهم است. بااین‌حال، تعداد زیادی از پروژه‌های نرم‌افزاری هنوز با شکست مواجه می‌شوند. برای افزایش احتمال موفقیت پروژه‌ها، استفاده از استانداردهای کیفیت نرم‌افزار برای هدایت فرایند تضمین کیفیت نرم‌افزار مناسب است [۷].

اهداف استانداردسازی به‌خوبی شناخته‌شده هستند؛ زیرا یک رویکرد یکپارچه در انجام فعالیت‌های مختلف ارائه می‌دهند. به طور ساده، استانداردها بهترین روش انجام کارها را نشان می‌دهند [۱۷].

معرفی استانداردهای ISO مربوطه به یک شرکت، علاوه‌بر هزینه، مزایایی را پس از مدت معینی به همراه دارد. به‌طورمثال، مزایای سازمان از اجرای ISO 9001 و صدور گواهینامه سامانه را می‌توان به خارجی و داخلی تقسیم کرد [۱۹-۲۲]. مزایای داخلی ممکن است شامل اطمینان از پایداری و تکرارپذیری فرآیندها، تضمین و حفظ حاکمیت سازمانی از طریق رویه‌های عملیاتی به‌طور واضح تعریف‌شده و مسئولیت‌ها و اختیارات مرتبط با آن‌ها، اطمینان از داده‌ها و اطلاعات بدون ابهام استخراج‌شده از نظارت و اندازه‌گیری محصولات و خدمات و همچنین فرآیندهایی برای استفاده در مدیریت سازمان و بهبود مستمر آن باشد [۲۳].

برای نمونه، بکارگیری استانداردهای مربوط به ISO 9001 می‌تواند کنترل مدیریت را در حیاتی‌ترین حوزه‌های ساختار سرمایه کاری بهبود بخشد [۲۴-۲۶]. رویه‌های ISO امکان جلوگیری از اشتباهات را فراهم می‌کند که به‌ویژه در مواقع بحران بسیار مهم است [۲۳].

یکی از شش کلاس در دسته‌بندی معماری SQA، عبارت است از: استانداردهای SQA، صدور گواهینامه و مولفه‌های ارزیابی سامانه SQA. این مولفه‌ها، استانداردهای مدیریتی و حرفه‌ای بین‌المللی را در سازمان پیاده‌سازی و اجرا می‌کنند. اهداف اصلی این گروه از مولفه‌ها عبارت‌اند از [۱۶]:

۱) استفاده از دانش حرفه‌ای بین‌المللی؛

۲) بهبود همکاری با سامانه‌های کیفیت سازمان‌های دیگر؛

۳) سنجش و ارزیابی حرفه‌ای اهداف و دستاوردهای سامانه‌های کیفیت سازمان.

استانداردهای موجود به دو زیر کلاس طبقه‌بندی می‌شوند که هر دو ممکن است موردنیاز مشتریان باشند:

۱) استانداردهای فرایند پروژه[1]

۲) استانداردهای مدیریت کیفیت[2]

به‌عنوان یک مثال از کاربرد استانداردها در تضمین کیفیت نرم‌افزار، استانداردهای قابل اجرا برای هر مرحله از چرخه حیات نرم‌افزار، در جدول (۲) آورده شده است [۱۸].

مزایای اصلی حاصل از به‌کارگیری استانداردها به شرح زیر فهرست شده است [۱۶]:

- ✓ توانایی درخواست برای پروژه‌های نرم‌افزاری و توسعه و نگهداری نرم‌افزارها در بخش‌های مربوطه با استفاده از روش‌ها و رویکردهای به‌روز و حرفه‌ای در سطح بالاتر؛
- ✓ درک متقابل و هماهنگی بهتر بین تیم‌های توسعه و به‌ویژه بین تیم‌های توسعه و نگهداری؛
- ✓ همکاری بهتر بین توسعه‌دهنده نرم‌افزار و مشارکت‌کنندگان خارجی در پروژه؛
- ✓ درک و همکاری بهتر بین تأمین‌کنندگان نرم‌افزار و مشتریان/خریدکنندگان، بر اساس پذیرش استانداردهای شناخته‌شده توسعه، نگهداری و تضمین کیفیت.

این مزایا، همراه با پیچیدگی و دامنه روزافزون پروژه‌های نرم‌افزاری، منجر به استفاده گسترده‌تر از استانداردها در صنعت شده است.

توسعه استانداردهای SQA توسط چندین مؤسسه استاندارد ملی و بین‌المللی انجام شده است؛ سازمان‌های حرفه‌ای و صنعت‌محور که منابع قابل‌توجهی را در توسعه و به‌روزرسانی پروژه‌های استاندارد سرمایه‌گذاری می‌کنند. مؤسسات و سازمان‌های زیر از برجسته‌ترین توسعه‌دهندگان استانداردهای SQA و مهندسی نرم‌افزار محسوب می‌شوند و در این زمینه شهرت و جایگاه بین‌المللی کسب کرده‌اند [۱۶]:

- IEEE (مؤسسه مهندسین برق و الکترونیک) انجمن کامپیوتر
- ISO (سازمان بین‌المللی استاندارد)
- ANSI (مؤسسه استاندارد ملی آمریکا[3])
- IEC (کمیسیون بین‌المللی الکتروتکنیکی)
- EIA (اتحادیه صنایع الکترونیک[4])

[1] Project Process Standards
[2] Quality management standards
[3] American National Standards Institute (ANSI)
[4] Electronic Industries Alliance (EIA)

جدول (۲): استانداردهای قابل اجرا برای مراحل/فرآیندهای توسعه نرم‌افزار [۱۸]

فرایند توسعه نرم‌افزار	استانداردهای قابل اجرا	شرح
تجزیه‌وتحلیل نیازمندی‌های نرم‌افزار	IEEE 830 [27]	محتویات و ویژگی‌های مشخصات نیازمندی-های نرم‌افزار را ارائه می‌دهد.
طراحی معماری نرم‌افزار	ISO/IEC/IEEE 42010 [28]	ارائه چارچوب مفهومی و محتوا برای توصیف معماری سامانه‌های مبتنی بر نرم‌افزار
طراحی دقیق نرم‌افزار	IEEE 1016 [29]	محتوا و سازماندهی توصیف طراحی نرم‌افزار را فراهم می‌کند.
ساخت نرم‌افزار	ISO/IEC/IEEE 29119 [30-32]	مجموعه‌ای از استانداردها را برای آزمون نرم-افزار ارائه می‌دهد که فرایند آزمون، فنون آزمون و مستندات آزمون را شرح می‌دهد.
صلاحیت نرم‌افزار	IEEE 730 [33]	فرمت و محتوای یک طرح تضمین کیفیت نرم-افزار را ارائه می‌دهد.
مدیریت پیکربندی نرم‌افزار	IEEE 828 [34]	محتوای یک برنامه مدیریت پیکربندی نرم‌افزار را ارائه می‌دهد.
تصدیق و اعتبارسنجی نرم‌افزار	IEEE 1012 [35]	رهنمودهایی را برای فعالیت‌های تصدیق و اعتبارسنجی نرم‌افزار ارائه می‌دهد.
بازبینی نرم‌افزار	IEEE 1028 [36]	الزامات بازبینی نظام‌مند و ممیزی نرم‌افزار را فراهم می‌کند.
مستندات کاربر	ISO/IEC/IEEE 26512 [37]	فرایند مستندسازی را از دیدگاه کاربر و همچنین از دیدگاه تأمین‌کننده ارائه می‌دهد.

سیر توسعه استانداردها و مدل‌ها در شکل (۴) نشان داده شده است.

خصوصیات دو کلاس استاندارد فوق‌الذکر، در جدول (۳) خلاصه شده است.

۳-۱- استانداردهای فرایند پروژه

استانداردهای این کلاس، بر روش‌های انجام پروژه‌های توسعه و نگهداری نرم‌افزار تمرکز دارند و کیفیت آنها را تضمین می‌کنند، به‌عبارت‌دیگر، بر «نحوه» اجرای یک پروژه نرم‌افزاری تمرکز دارند. این استانداردها مراحلی که باید انجام شود، نیازمندی‌های مستندسازی طراحی، محتوای اسناد طراحی، بازبینی‌های طراحی و مسائل مربوط به بازبینی، آزمون نرم‌افزار که باید انجام شود، موضوعات آزمون و سایر موارد را تعریف می‌کنند. طبیعتاً به‌دلیل ویژگی‌های آنها، بسیاری از استانداردها در این دسته، می‌توانند به‌عنوان کتاب‌های درسی مهندسی نرم‌افزار و تضمین کیفیت نرم‌افزار استفاده شوند [۱۶]. این استانداردها به‌طورکلی، استانداردهای تخصصی هستند که رهنمودهای روش‌شناسانه را برای تیم توسعه‌دهنده فراهم می‌کنند.

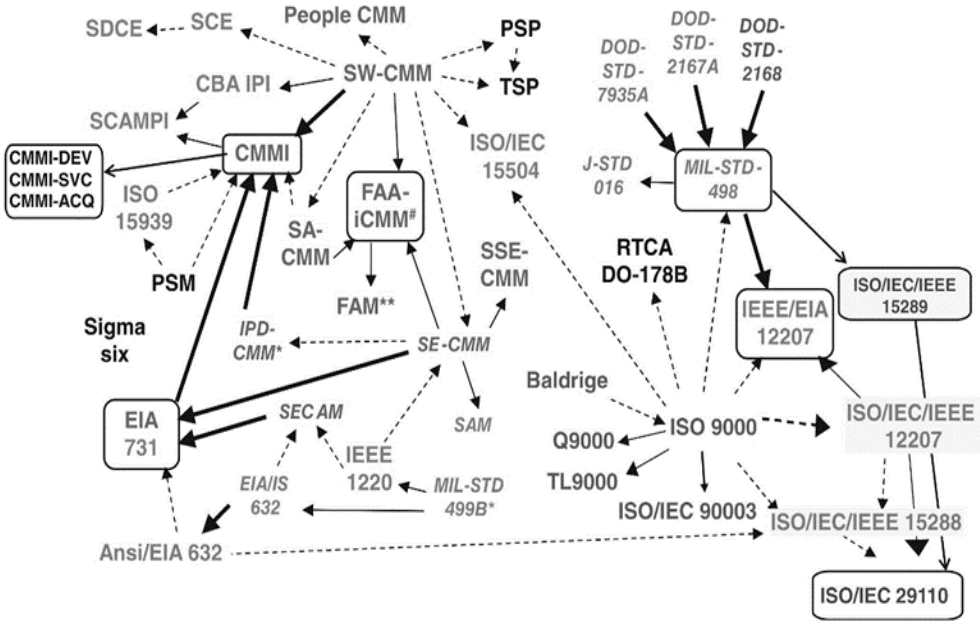
با اینکه این استانداردها توضیح می‌دهند که چه مراحلی باید برداشته شوند و چه محصولات کاری باید به‌عنوان بخشی از فرایند، توسعه داده شوند، اما هنوز فنون مشخص (مانند روش‌ها و ابزارها) را برای سازمان‌ها باز می‌گذارند. هدف اصلی استانداردهای فرایند پروژه، تعریف چارچوب مشترکی از بهترین روش‌ها است. این بهترین روش‌ها، در قالب فعالیت‌ها و زیرفعالیت‌هایی است که باید به‌عنوان بخشی از یک پروژه انجام شوند [۳۸].

برای مثال، IEEE/EIA 12207 کل چرخه حیات پروژه را پوشش می‌دهد و IEEE 1012 به‌طورخاص بر فعالیت‌های تأیید و اعتبارسنجی[1] تمرکز می‌کند.

۳-۱-۱- استاندارد IEEE 730

استاندارد IEEE 730-2014 برای فرایندهای تضمین کیفیت نرم-افزار شامل وظایف ارزیابی است که به‌طور خاص شامل وظایف تأیید و اعتبارسنجی است یا به‌طور خلاصه به‌عنوان V&V شناخته می‌شود. در ضمیمه E از [۳۳] برای راهنمای ویژه صنعت برای استفاده از IEEE 730-2014، تعریف تأیید نرم‌افزار برای صنعت تجهیزات پزشکی شامل می‌شود: «آزمون نرم‌افزار یکی از بسیاری از فعالیت‌های راستی‌آزمایی است، برای تایید اینکه خروجی توسعه نرم‌افزار نیازمندی‌های ورودی آن را برآورده می‌کند» [۳۹].

[1] verification and validation activities (V&V)



شکل (۴): توسعه استانداردها و مدل‌ها [۹]

جدول (۳): مقایسه کلاس‌های استاندارد [۱۶]

مشخصات	استانداردهای فرایند پروژه	استانداردهای مدیریت کیفیت
واحد هدف	یک تیم پروژه توسعه و/یا تعمیر و نگهداری نرم‌افزار	مدیریت توسعه و/یا نگهداری نرم‌افزار و واحدهای خاص SQA.
تمرکز اصلی	روش‌شناسی برای انجام پروژه‌های توسعه و نگهداری نرم‌افزار	سازماندهی سامانه‌ها، زیرساخت‌ها و الزامات SQA
مقصود استاندارد	«چگونه» اجرا شود	«چه چیزی» باید به دست آورد
هدف استاندارد	تضمین کیفیت یک پروژه نرم‌افزاری خاص	تضمین کیفیت نرم‌افزار تأمین‌کننده و ارزیابی قابلیت فرایند نرم‌افزاری آن
مثال‌ها	IEEE 730 Standard ISO/IEC/IEEE12207 Standard IEEE 1012 Standard IEEE 828	ISO/IEC/IEEE 90003 Standard SEI CMMI ISO/IEC 15504 Standard ISO 25010

به‌طورکلی، این استاندارد، فرمت و محتوای یک طرح تضمین کیفیت نرم‌افزار را ارائه می‌دهد. فهرست محتویات یک طرح تضمین کیفیت نرم‌افزار[1] (SQAP) مطابق با استاندارد IEEE 730 در جدول (۴) نشان داده شده است [۱۸].

۳-۱-۲- استاندارد ISO/IEC/IEEE 12207

استاندارد ISO/IEC/IEEE 12207، با عنوان مهندسی سامانه‌ها و نرم‌افزار - فرایندهای چرخه حیات نرم‌افزار، استانداردی است که که انطباق فرایند توسعه نرم‌افزار در شرکت را با استاندارد، موردارزیابی قرار می‌دهد. اگرچه این استاندارد به‌صورت مستقیم یک استاندارد ایمنی نیست، اما با ویژگی‌های کیفیت حیاتی مرتبط است. ویژگی‌های کیفیت شامل جوانب مرتبط با سلامتی، ایمنی، تضمین امنیت، قابلیت اطمینان، قابلیت دسترسی و قابلیت پشتیبانی می‌شوند. در این استاندارد، ایمنی معمولاً به‌عنوان انتظاری برای عدم تهدید جان انسان، سلامتی، اموال یا محیط‌زیست شناخته می‌شود؛ بنابراین، استاندارد ISO/IEC/IEEE 12207 برای ارزیابی فرایندهای توسعه نرم‌افزار برای موارد موردنظر، به دلیل اقدامات ایمنی آن علاوه بر استفاده از آن توسط شرکت، مورد استفاده قرار می‌گیرد [۴۰].

سه سطح بالایی از معماری فرایند استاندارد، به‌عنوان کلاس‌های فرایند، دسته‌بندی‌های فرایند و فرایندهای جامع آنها، در نمودار استخوان ماهی در شکل (۵) نشان داده شده‌اند [۱۶].

[1] Software Quality Assurance Plan (SQAP)

جدول (۴): فهرست محتویات یک SQAP مطابق با استاندارد IEEE 730 [۱۸]

ردیف	فعالیت	زیربخش‌ها
۱	هدف و دامنه	-
۲	تعاریف و کلمات اختصاری	-
۳	اسناد مرجع	-
۴	نمای کلی طرح SQA	۱,۴ سازماندهی و استقلال ۲,۴ مخاطرات محصول نرم‌افزاری ۳,۴ ابزار ۴,۴ استانداردها، رویه‌ها و قراردادها ۵,۴ تلاش، منابع و برنامه
۵	فعالیت‌ها، نتایج و وظایف	۱,۵ تضمین محصول ۱,۱,۵ ارزیابی برنامه‌ها برای انطباق ۲,۱,۵ ارزیابی محصول برای انطباق ۳,۱,۵ ارزیابی برنامه‌ها برای پذیرش ۴,۱,۵ ارزیابی چرخه حیات محصول برای انطباق ۵,۱,۵ اندازه گیری محصولات ۲,۵ تضمین فرآیند ۱,۲,۵ ارزیابی فرآیندهای چرخه حیات برای انطباق ۲,۲,۵ ارزیابی محیط‌ها برای انطباق ۳,۲,۵ ارزیابی فرآیندهای پیمانکار فرعی برای انطباق ۴,۲,۵ اندازه‌گیری فرآیندها ۵,۲,۵ ارزیابی مهارت و دانش کارکنان
۶	ملاحظات اضافی	۱,۶ بازبینی قرارداد ۲,۶ اندازه‌گیری کیفیت ۳,۶ چشم‌پوشی‌ها و انحرافات ۴,۶ تکرار کار ۵,۶ مخاطرات برای انجام SQA ۶,۶ راهبرد ارتباطات ۷,۶ فرآیند عدم انطباق
۷	سوابق SQA	۱,۷ تجزیه‌وتحلیل، شناسایی، جمع آوری، پرونده‌سازی، نگهداری و دفع ۲,۷ در دسترس‌بودن سوابق

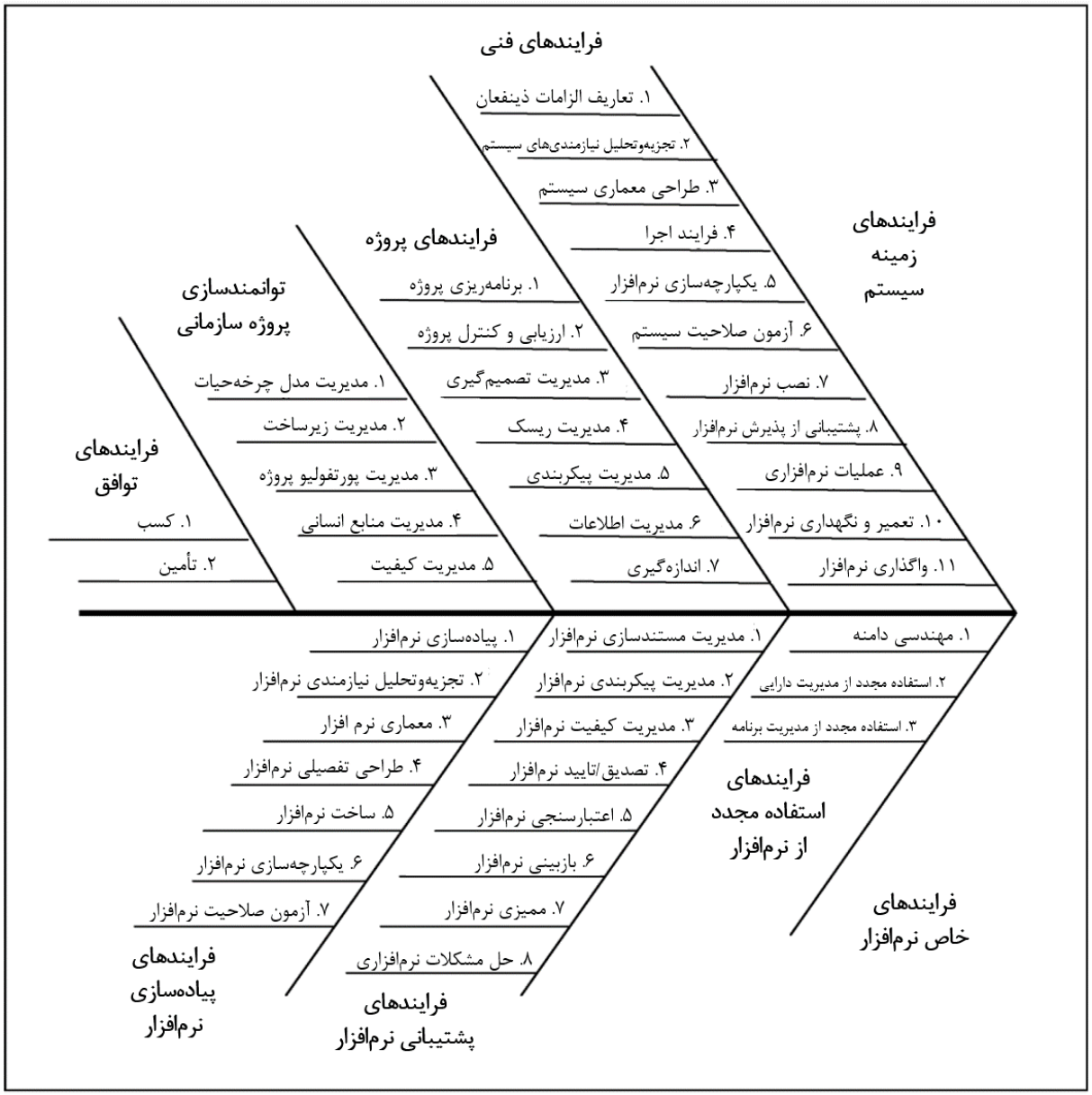
۳-۱-۳- استاندارد IEEE 1012

استاندارد IEEE 1012:2016، فهرست مطالب موردنیاز برای تولید یک برنامه تأیید و اعتبارسنجی نرم‌افزار[1] (SVVP) و نیز رویکردهایی را برای برنامه‌ریزی تأیید و اعتبارسنجی (V&V) توصیه می‌کند. به‌طورکلی این استاندارد موارد زیر را شرح می‌دهد [۱۸]:

- فعالیت‌های تأیید و اعتبارسنجی نرم‌افزار و محتوای طرح تأیید و اعتبارسنجی نرم‌افزار را تعیین و توضیح می‌دهد و رویکرد تأیید و اعتبارسنجی چرخه حیات نرم‌افزار بر اساس مدل V را توضیح می‌دهد.
- استفاده از شواهد رسمی برای تأیید صحت نرم‌افزار را توصیه می‌کند.
- استفاده از ابزارهای CASE را برای تجزیه‌وتحلیل استاتیک کد و آزمون توصیه می‌کند.
- فهرست مطالب[2] را برای مشخصات SVVP برای آزمون واحد (SVVP/UT)، آزمون یکپارچه‌سازی (SVVP/IT)، آزمون سامانه (SVVP/ST) و آزمون پذیرش (SVVP/AT) توصیه می‌کند.

[1] Software verification and validation plan (SVVP)

[2] Table of contents (TOC)



شکل (۵): نمودار استخوان ماهی استاندارد ISO/IEC 12207 [۱۶]

شرح وظایف ارائه‌شده توسط این استاندارد بسیار دقیق است و شامل جزئیات وظایف و ورودی‌ها و خروجی‌های موردنیاز است. نمونه‌ای از شرح وظایف در جدول (۵) نشان داده شده است [۱۶].

به‌طورکلی در این استاندارد، تجزیه‌وتحلیل، ارزیابی، بازبینی، بازرسی، تخمین و آزمون محصول و فرایند همگی در ارزیابی گنجانده شده است [۴۱].

۳-۱-۴- استاندارد IEEE 828

در سال ۱۹۸۳، مؤسسه IEEE اولین استاندارد برای مدیریت پیکربندی نرم‌افزار[1] را منتشر کرد. استاندارد IEEE 828 برای برنامه‌های مدیریت پیکربندی نرم‌افزار که در سال‌های ۱۹۹۰، ۱۹۹۸ و ۲۰۰۵ بازنگری شد. آخرین نسخه این استاندارد در سال ۲۰۱۲ منتشر شد. استاندارد IEEE 828 حداقل نیازمندی‌ها را برای فرآیندهای مدیریت پیکربندی در سامانه‌ها و مهندسی نرم‌افزار تعیین می‌کند [۴۲].

استاندارد IEEE 828 برای تهیه طرح مدیریت پیکربندی نرم‌افزار[2] تأیید شده است. هر گونه تغییر در نرم‌افزار باید مطابق با SCMP تجزیه‌وتحلیل، مستند و تأیید شود. به‌طورکلی این

[1] Software Configuration Management (SCM)
[2] Software Configuration Management Plan (SCMP)

استاندارد، محتوای یک طرح مدیریت پیکربندی نرم‌افزار را ارائه می‌دهد. این استاندارد موارد زیر را در بر می‌گیرد [۱۸]:

- ✓ مدیریت پیکربندی نرم‌افزار و محتویات طرح مدیریت پیکربندی نرم‌افزار را تعریف و توضیح می‌دهد و همچنین دستورالعمل‌هایی در مورد نحوه انجام همین کار ارائه می‌دهد.
- ✓ رابطه‌ای بین فعالیت‌های SCM و توسعه اجزای نرم‌افزار و یکپارچه‌سازی سامانه برقرار می‌کند. علاوه‌براین، فعالیت‌های کنترل تغییر پیکربندی کد و سند را شناسایی و توضیح می‌دهد.
- ✓ استفاده از ابزارهای CASE را برای مدیریت پیکربندی توصیه می‌کند.
- ✓ TOC را برای SCMP توصیه می‌کند.

جدول (۵): شرح وظایف استاندارد IEEE Std 1012 - یک مثال [۱۶]

فرایند: فعالیت‌های رایج V&V فعالیت: برنامه‌ریزی تأمین تأیید و اعتبارسنجی وظیفه (۲): تأیید قرارداد		
شرح وظیفه	ورودی موردنیاز	خروجی موردنیاز
تأیید مشخصات زیر در قرارداد: ۱. نیازمندی‌های سامانه (از RFP (Request for proposal) یا مناقصه و قرارداد) راضی‌کننده و سازگار با نیازهای کاربر هستند. ۲. رویه‌ها برای مدیریت تغییرات نیازمندی‌ها و شناسایی سلسله مراتب مدیریت برای مدیریت مشکلات به صورت سندی ثبت شده‌اند. ۳. رویه‌های ارتباط و همکاری بین طرفین مستند شده است، از جمله مالکیت، ضمانت، کپی رایت و محرمانگی. ۴. معیارهای پذیرش و رویه‌ها مطابق با نیازمندی‌ها مستند شده است.	۱. طرح V&V (VVP) ۲. درخواست پیشنهادیه مناقصه ۳. قرارداد ۴. برنامه‌ها و زمان‌بندی‌های توسعه تامین‌کننده	۱. گزارش کار(ها) ۲. تأیید قرارداد ۳. VVP به‌روز شده ۴. گزارش(های) ناهنجاری

استاندارد IEEE 828 از استاندارد ISO 12207 نیز پشتیبانی می‌کند. حداقل الزامات قابل قبول برای CM در هر دو سامانه و نرم‌افزار را بیان می‌کند. در نتیجه، برای همه کلاس‌ها اعمال می‌شود. استاندارد EEE 828 فعالیت‌های CM را که باید در چه نقطه‌ای از چرخه حیات انجام شود و برنامه‌ریزی و منابع موردنیاز را توصیف می‌کند. طبق استاندارد IEEE 828، هدف مدیریت پیکربندی به شرح زیر است که [۹]:

- شناسایی و مستندسازی ویژگی‌های عملکردی و فیزیکی یک محصول، مؤلفه، خروجی یا یک سرویس
- کنترل هرگونه تغییری در این ویژگی‌ها
- ثبت و گزارش هر تغییر و وضعیت اجرای آن
- پشتیبانی از ممیزی محصولات، نتایج، سرویس‌ها یا مؤلفه‌ها برای تأیید انطباق با نیازمندی‌ها

۳-۲-استانداردهای مدیریت کیفیت

استانداردهای این کلاس بر روی توسعه نرم‌افزار سازمان و زیرساخت‌ها و الزامات مدیریت تضمین کیفیت نرم‌افزار تمرکز دارند و انتخاب روش‌ها و ابزارها را به سازمان واگذار می‌کنند. با رعایت استانداردهای مدیریت کیفیت، سازمان‌ها می‌توانند به طور پیوسته اطمینان حاصل کنند که محصولات نرم‌افزاری آنها به سطح قابل قبولی از کیفیت دست می‌یابند. برخی از مناقصه‌های توسعه نرم‌افزار فعلی، شرکت‌کنندگان را ملزم می‌کنند که گواهینامه یکی از استانداردهای مدیریت کیفیت را داشته باشند [۱۶].

این استانداردها راهنمای موارد زیر هستند:

- مدیریت توسعه نرم‌افزار
- نگهداری نرم‌افزار
- زیرساخت نرم‌افزار

این استانداردها بر آنچه موردنیاز است، تمرکز دارند، نه تصمیم‌گیری در مورد چگونگی دستیابی به آنها. سامانه مدیریت کیفیت، ارزیابی هدفمند از دستاوردهای سازمان را فراهم می‌سازد. سازمان‌هایی که نیازمندی‌های دستیابی به کیفیت را برآورده کرده‌اند، می‌توانند گواهینامه SQA را دنبال کنند.

استانداردهای مدیریت کیفیت معمولاً حداقل دو هدف را دنبال می‌کنند. در مرحله اول، امکان ارزیابی بلوغ زیرساخت SQA یک سازمان؛ این معمولاً شامل صدور گواهینامه توسط یک نهاد معتبر خارجی است. چنین گواهی‌نامه‌ای به نمونه‌های OEM[1] (سازنده تجهیزات اصلی) اجازه می‌دهد تا فرایند SQA

[1] Original Equipment Manufacturer

تـأمین‌کننـده را ارزیـابی کننـد و در نتیجـه نشـانه‌ای از توانـایی سازمان برای ارائه محصولات باکیفیت بالا را ارائه می‌دهـد. ثانیـاً، استانداردهای مدیریت کیفیت را می‌توان برای راهنمایی سـازمان در جهت بهبود سامانه SQA استفاده کرد [۳۸].

استانداردهای مـدیریت کیفیـت معمـولاً بـر آنچـه کـه یـک سـازمان توسـعه‌دهنـده بایـد قـادر بـه انجـام آن باشـد، بـدون مشخص‌کردن چگونگی دستیابی به آن قابلیت، متمرکز است. این رویکرد عمدتاً قابـل قبـول اسـت؛ زیـرا مناسـب‌تـرین روش‌هـا و ابزارهای خاص برای دستیابی بـه قابلیـت‌هـا، ممکـن اسـت بـین سازمان‌ها متفاوت باشد (به‌عنوان‌مثال، بسته به حـوزه محصـول). بااین‌حال، روش‌ها و ابزارهای انتخاب‌شـده بایـد بـه دسـتیابی بـه قابلیت‌هایی که توسط استانداردها تعریف شده‌انـد، کمـک کننـد [۳۸].

۳-۲-۱- استاندارد ISO/IEC/IEEE 90003

در ابتدا، استاندارد ISO 9000 پشتیبانی خاصی از فرایندهای توسعه نرم‌افزار نداشت؛ از ایـن‌رو، اسـتاندارد ISO 9000-3 بـرای برآورده کردن نیازهای صنعت نرم‌افـزار منتشـر گردیـد. در سـال ۲۰۰۴ اولین نسخه ISO/IEC 90003 جایگزین استاندارد فوق شد و دستورالعمل‌هایی را در مورد به‌کارگیری آخـرین ISO 9000 در توسعه نرم‌افزار ارائه کرد. اگرچه دو تفسیر از کیفیـت (مبتنـی بـر فرآیند و محصول) اساساً متفاوت هستند، اما در بسیاری از مـوارد یکدیگر را پوشش مـی‌دهنـد. اسـتاندارد ISO 90003 مبتنـی بـر فرآیند توصیه می‌کنـد کـه ISO/IEC 25010 بایـد بـرای تعریـف ویژگی‌های کیفی نرم‌افزار تولیدشده با فرآیندهای توسعه نرم‌افزار منطبق با ISO 9000 استفاده شود [۱۷].

این سند راهنمایی‌هایی را برای سازمان‌ها در استفاده از ISO 9001:2015 برای کسب، تأمین، توسعه، بهره‌بـرداری و نگهـداری نرم‌افزارهای کامپیوتری و خدمات پشتیبانی مرتبط ارائه می‌دهـد. به نیازمنـدی‌هـای اسـتاندارد ISO 9001:2015 اضـافه نکـرده یـا تغییری نمی‌دهد. دستورالعمل‌های ارائه‌شده در ایـن سـند بـرای استفاده به‌عنوان معیار ارزیابی در ثبت/گواهینامه سامانه مـدیریت کیفیت در نظر گرفته نشده است. بااین‌حال، برخی از سـازمان‌هـا می‌توانند اجرای دستورالعمل‌های پیشنهادی در این سند را مفید بدانند و می‌توانند بدانند که آیا سامانه مدیریت کیفیت حاصل، با این سند مطابقت دارد یا خیر. در این حالت، یک سازمان می‌تواند از این سـند و ISO 9001 بـه‌عنـوان معیـار ارزیـابی سـامانه‌هـای مدیریت کیفیت در حوزه نرم‌افزار استفاده کند [۴۳].

احتمـالاً اسـتاندارد ISO/IEC 90003 یکـی از مناسـب‌تـرین استانداردهای بین‌المللی برای استفاده به‌عنوان یـک مـدل بـرای پذیرش به‌عنوان پایه‌ای برای سـامانه SQA سـطح بـالا و توسـعه دستورالعمل‌های کیفیـت مـوردنیـاز اسـت. موضـوعات موردنیـاز SQA، همان‌طور که در نسخه ۲۰۱۴ استاندارد بیان شـده اسـت، به عنوان موضوعات رویه‌ای است که باید توسط سازمان‌ها اتخـاذ شود [۱۶].

جـامع‌تـرین تعـاریف اسـتاندارد ISO/IEC 90003:2014 از اقدامات اصلاحی و پیشـگیرانه بـا توجـه بـه توسـعه و نگهـداری نرم‌افزار عبارت است از [۱۶]:

- اقدامات اصلاحی:

یک فرایند سازمانی بـازخوردی کـه بـه طـور مـنظم اعمـال می‌شود که اقـداماتی را بـرای از بـین بـردن علـل عـدم انطبـاق (نقص‌های نرم‌افزار) آغاز و انجام می‌دهد.

- اقدامات پیشگیرانه:

یک فرایند سازمانی بـازخوردی کـه بـه طـور مـنظم اعمـال می‌شود که اقداماتی را برای جلوگیری از وقوع عدم انطباق بـالقوه (نقص‌های نرم‌افزاری) آغاز و انجام می‌دهد.

گروه‌های نیازمندی و حوزه‌های موردنیـاز آنهـا در جـدول ۶ ارائه شده است [۱۶].

۳-۲-۲-استاندارد SEI CMMI

یکپارچه‌سازی مدل بلوغ قابلیـت[1] بـرای توسـعه، یـک مـدل مرجع است که فعالیت‌هایی را برای توسعه محصولات و خـدمات پوشـش مـی‌دهـد [۴۴]. موسسـه مهندسـی نـرم‌افـزار (SEI) در دانشـگاه کـارنگی ملـون در پیتسـبرگ، پنسـیلوانیا، چـارچوب یکپارچه‌سازی مدل بلوغ قابلیت را برای بهبود فرآینـد در ایـالات متحـده ارائـه کـرد کـه SEI آن را سـاخته اسـت. ایـن مـدل، مجموعه‌ای از بهترین شیوه‌ها برای بهبـود نحـوه عملکـرد کارهـا است [۴۵].

پروژه یکپارچه‌سازی CMM برای حل مشکل تکراری استفاده از چنـدین CMM ایجـاد شـده اسـت. CMMI مـدلی اسـت کـه مـی‌توانـد سـه مـدل منبـع را ترکیـب کنـد - اتحادیـه صـنایع الکترونیک/استاندارد موقت[2] ۷۳۱، مدل بلوغ قابلیت برای نرم‌افزار و مدل بلوغ قابلیت توسعه محصول یکپارچه[3]- در یـک چـارچوب بهبود واحد که می‌تواند رشته‌های مختلف زیادی را در خود جای دهد و انعطاف‌پذیری برای پشتیبانی از دو نمـایش متفـاوت (هـم پیوسته و هم مرحله‌ای) را دارد [۴۶].

[1] Capability Maturity Model Integration (CMMI)
[2] Electronic Industries Alliance/Interim Standard (EIA/IS)
[3] Integrated Product Development Capability Maturity Model (IPD-CMM)

جدول (۶): حوزه‌های نیازمندی و طبقه‌بندی آنها برای استاندارد ISO/IEC/IEEE 90003 [۱۶]

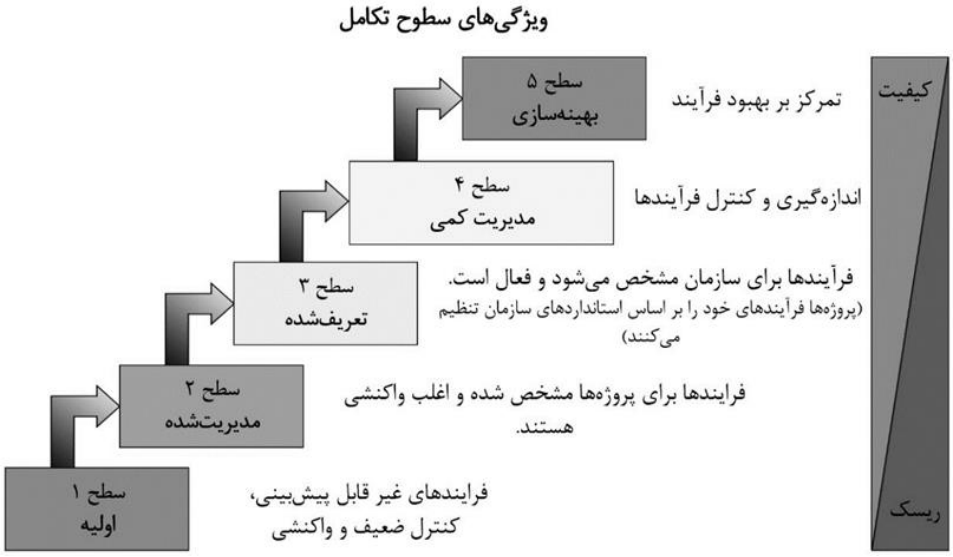
حوزه‌های موردنیاز	گروه‌های موردنیاز
۴. سامانه مدیریت کیفیت	۱,۴ نیازمندی‌های عمومی ۲,۴ مدارک موردنیاز
۵. مسئولیت‌های مدیریتی	۱,۵ تعهدات مدیریت ۲,۵ تمرکز بر مشتری ۳,۵ خط‌مشی کیفیت ۴,۵ برنامه‌ریزی ۵,۵ مسئولیت، اختیار و ارتباطات ۶,۵ بازبینی مدیریت
۶. مدیریت منابع	۱,۶ تامین منابع ۲,۶ منابع انسانی ۳,۶ زیرساخت ۴,۶ محیط کار
۷. تحقق محصول	۱,۷ برنامه‌ریزی برای تحقق محصول ۲,۷ فرآیندهای مرتبط با مشتری ۳,۷ طراحی و توسعه ۴,۷ خرید ۵,۷ تولید و ارائه خدمات ۶,۷ کنترل دستگاه‌های نظارت و اندازه‌گیری
۸. اندازه‌گیری، تجزیه‌وتحلیل و بهبود	۱,۸ بهبود کلی ۲,۸ نظارت و اندازه‌گیری ۳,۸ کنترل محصولات ناسازگار ۴,۸ تجزیه‌وتحلیل داده‌ها ۵,۸ بهبود

در سال ۲۰۱۰، SEI نسخه CMMI ۱,۳ را منتشر کرد. این با استفاده از اطلاعات به‌دست‌آمده از تعدادی از مدل‌های معتبر و محبوب و همچنین دانش در مورد بهترین شیوه‌های توسعه نرم‌افزار و بهترین سامانه‌های سازمان‌های CMM با بلوغ بالا که بیش از ۱۰ سال از این مدل استفاده کرده بودند، ایجاد شد [۴۶]. مدل CMMI پنج سطح رشد[1] دارد که در شکل ۶ نشان داده شده است. سطح ۱ پایین‌ترین سطح است که یک فرایند مدیریت نامناسب را نشان می‌دهد، درحالی‌که سطح ۵ بالاترین سطح است و یک قابلیت مهندسی نرم‌افزار بسیار موثر و پیشرفته را نشان می‌دهد.

۲۲ حوزه فرآیند[2] وجود دارد که با پنج سطح مرتبط هستند، به جز سطح ۱. هر حوزه فرآیند دارای مجموعه‌ای از شیوه‌های مرتبط است که باید برای دستیابی به تعدادی از اهداف واضح تعریف‌شده، انجام شود [۴۶]. برای اینکه یک شرکت توسعه نرم‌افزار به سطح خاصی از تکامل برسد، ابتدا باید تمام اهداف حوزه فرایند را در آن سطح خاص و همچنین هر سطح پایین‌تر، برآورده کند [۴۷].

هر سطح شامل چندین حوزه فرایند است و برای هر یک از این حوزه‌ها، اهداف «عمومی» و «ویژه» شرح داده شده است که همه آنها متفاوت هستند. روش‌های مختلفی تحت هر هدف مشخص می‌شوند. این شیوه‌ها به سازمان کمک می‌کنند تا به طور کامل درک کند که چگونه می‌تواند به اهداف بلوغ خود دست یابد. آنها همچنین نمونه‌هایی از فعالیت‌هایی هستند که هر زمان که یک برنامه بهبود فرایند نرم‌افزار[3] انجام می‌شود، باید مورد توجه قرار گیرند [۴۶].

[1] Maturity levels (MLs)
[2] process areas (PAs)
[3] Software Process Improvement (SPI)



شکل (۶): یکپارچه‌سازی مدل بلوغ قابلیت (CMMI)

مدل CMMI یک چارچوب بهبود مهم و باارزش است که دارای PA، سطوح تعریف‌شده و اقدامات کلیدی است. بااین‌حال، CMM همانند نسل قبلی خود نشان نمی‌دهد که شیوه‌های کلیدی چگونه باید اجرا شوند و همچنین توصیه نمی‌کند که چگونه سازمان‌ها می‌توانند استراتژی‌های پیاده‌سازی خود را بهبود دهند [۴۸, ۴۹].

مدل CMMI مشابه ISO 9000-3، مدلی است که برای ارزیابی قابلیت یک سازمان توسعه‌دهنده نرم‌افزار استفاده می‌شود. تمرکز ویژه CMMI بر روش‌های مدیریت مبتنی بر رویکردهای کمی است. این امر به‌عنوان‌مثال در سطوح بالاتر بلوغ، سازمان ملزم است از معیارهای محصول و فرایند برای کنترل عملکرد و شناسایی نیازها و فرصت‌های بهبود استفاده کند [۳۸].

جدول (۷) حوزه‌های فرآیند مرتبط با سطوح بلوغ را در نسخه مرحله‌ای از مدل توسعه CMMI نشان می‌دهد [۹].

۳-۲-۳- استاندارد ISO/IEC 15504

استاندارد ISO/IEC 15504 (معروف به SPICE[1]) رویکرد محبوب دیگری است که به سازمان‌ها اجازه می‌دهد تا بلوغ فرایندهای توسعه نرم‌افزار خود را از طریق بهبود مستمر افزایش دهند. این مدل توسط شرکت‌های توسعه نرم‌افزار با فرایندهای رسمی پذیرفته‌شده و به طور گسترده استفاده می‌شود و سازگاری گسترده و آسانی برای یکپارچه‌سازی با سایر مدل‌های فرایندی دارد. این استاندارد، بزرگ‌ترین مدل آزمایش‌شده و مورد توافق است که دو بعد مستقل را برای فرایندها و قابلیت ارائه می‌دهد. بااین‌وجود، به طور خاص بر قابلیت استفاده و دسترسی متمرکز نیست. این استاندارد توسط ISO/IEC 33000 جدید بازنگری شده است [۵۰].

مدل ارزیابی فرایند ۱۵۵۰۴ یک مدل دو بعدی است:

- بعد قابلیت
- بعد فرایند

مدل بُعد قابلیت از شش سطح قابلیت تشکیل شده است که سطح صفر کمترین و سطح ۵ بالاترین است. این مدل، ویژگی‌های فرآیند[2] را تعریف می‌کند که برای دستیابی به هر سطح قابلیت باید به آنها دست یافت. ویژگی‌های فرآیند، عمومی هستند و «چه چیزی» را تعریف می‌کنند، نه «چگونگی». مدل بُعد قابلیت و ویژگی‌های فرآیند موردنیاز برای هر سطح در شکل (۷) نشان داده شده است [۱۶].

مدل ابعاد قابلیت از موارد زیر تشکیل شده است:

- سطوح قابلیت و نیازمندی‌های حوزه فرایند برای هر سطح
- شاخص‌هایی برای هر PA که به‌عنوان مبنایی برای جمع‌آوری شواهد عینی استفاده می‌شوند که ارزیاب را قادر می‌سازد تا رتبه‌بندی را تعیین کند.
- مقیاس درجه‌بندی پیشرفت برای ویژگی‌های فرایند
- نیازمندی‌های پیشرفت تجمعی برای هر سطح قابلیت

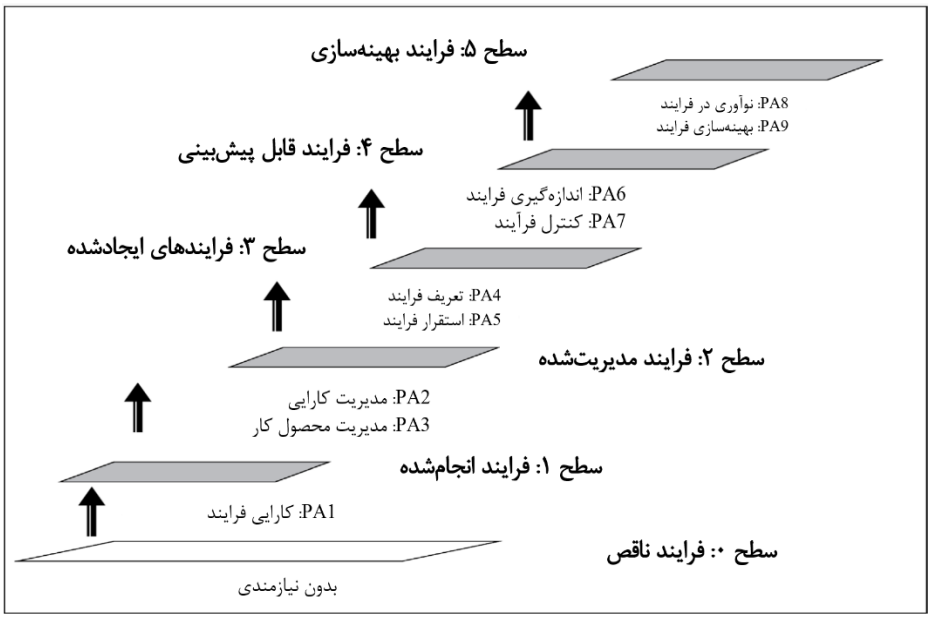
بُعد فرایند ۱۵۵۰۴ به طور کامل مدل مرجع فرایند استاندارد ۱۲۲۰۷ ارائه‌شده در شکل ۵ را پذیرفته است. مدل بُعد فرایند از بیش از ۴۰ فرآیند تشکیل شده است (تعداد فرایندها برای نسخه‌های مختلف استاندارد ۱۲۲۰۷ کمی متفاوت است) [۱۶].

[1] Software Process Improvement and Capability Evaluation

[2] Process attributes (PAs)

جدول (۷): نمایش مرحله‌ای از مدل CMMI برای توسعه [۹]

سطح	تمرکز	حوزه فرایند کلیدی	
۵ بهینه‌سازی	بهبود مستمر فرایند	مدیریت عملکرد سازمانی تجزیه‌وتحلیل و رفع علل	**کیفیت** **بهره‌وری**
۴ مدیریت کمی	مدیریت کمی	عملکرد فرایند سازمانی مدیریت کمی پروژه	
۳ تعریف‌شده	استانداردسازی فرایند	توسعه نیازمندی‌ها راه‌حل فنی یکپارچه‌سازی محصول تصدیق اعتبارسنجی تمرکز فرایند سازمانی تعریف فرایند سازمانی آموزش سازمانی مدیریت یکپارچه پروژه مدیریت مخاطرات تجزیه‌وتحلیل تصمیم و حل‌وفصل	
۲ مدیریت‌شده	مدیریت پروژه پایه	مدیریت نیازمندی‌ها برنامه‌ریزی پروژه نظارت و کنترل پروژه مدیریت قرارداد تأمین‌کننده اندازه‌گیری و تجزیه‌وتحلیل تضمین کیفیت فرایند و محصول مدیریت پیکربندی	**مخاطرات** **دوباره‌کاری**
۱ اولیه	-	-	



شکل (۷): مدل ابعاد قابلیت ISO/IEC 15504 [۱۶]

فرآیندها به هفت دسته طبقه‌بندی می‌شوند [۱۶]:

گروه دسته‌بندی چرخه حیات سامانه:

- AGR: روند توافق[1]
- ORG: فرایندهای فعال‌کننده پروژه سازمان[2]
- PRO: فرایندهای پروژه[3]
- ENG: فرایندهای فنی[4]

گروه دسته‌بندی چرخه حیات نرم‌افزار:

- DEV: فرایندهای توسعه نرم‌افزار[5]
- SUP: فرایندهای پشتیبانی نرم‌افزار[6]
- REU: فرایندهای استفاده مجدد از نرم‌افزار[7]

رابطه بین بُعد قابلیت و بُعد فرایند توسط استاندارد در جدولی تعریف می‌شود که PAهای مربوطه را برای هر فرایند نشان می‌دهد. طبق جدول، یک فرایند می‌تواند برای یک تا چهار PA مرتبط باشد. مدل ارزیابی فرایند دوبُعدی و روابط بین ابعاد در شکل ۸ ارائه شده است [۱۶].

اهداف و ساختار استاندارد ISO/IEC 15504 مشابه CMMI است که تعدادی از سطوح بلوغ را برای مجموعه‌ای از حوزه‌های فرایند کلیدی تعریف می‌کند. شباهت بین CMMI و ISO/IEC 15504 از طریق نگاشت بین دو استاندارد نشان داده شده است (به‌عنوان‌مثال [۵۱, ۵۲]).

۳-۲-۴- استاندارد ISO 25010

استاندارد ISO/IEC 25010 آخرین مورد از سری استانداردهای ISO/IEC است که به‌عنوان ارزیابی و نیازمندی‌های کیفیت نرم‌افزار (SQuaRE[8]) شناخته می‌شود [۵۳].

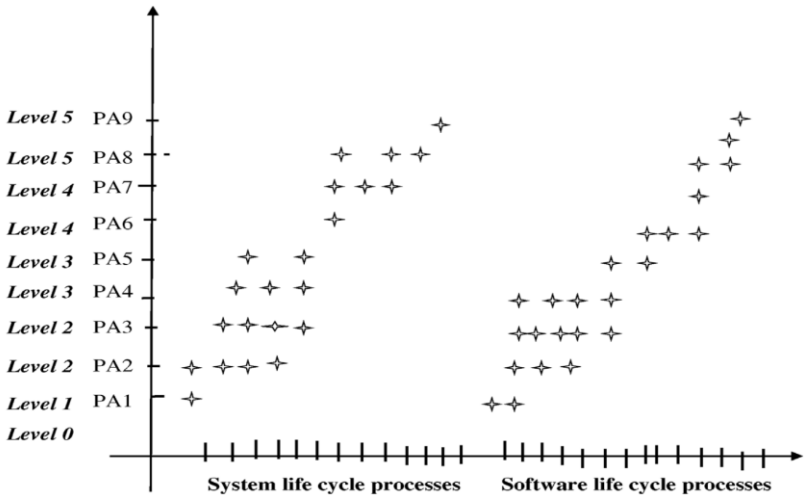
پارادایم SQuaRE کیفیت سامانه‌های نرم‌افزاری را در زمینه فارنزیک (جمع‌آوری و بررسی شواهد) نرم‌افزار، ارزیابی کرده و یک مدل «کیفیت در استفاده[9]» ارائه می‌دهد [۴۱].

استاندارد بین‌المللی ISO/IEC 25010 مجموعه‌ای از ویژگی‌ها را پیشنهاد می‌کند که در هنگام اندازه‌گیری «کیفیت در استفاده[10]» یک سامانه نرم‌افزاری باید در نظر گرفته شوند. استاندارد ISO/IEC 25010 دو مدل را برای اندازه‌گیری کیفیت تعریف می‌کند: «کیفیت در استفاده» و «کیفیت محصول[11]». اولی شامل ویژگی‌هایی است که به نتیجه تعامل مربوط می‌شود، زمانی که یک محصول در زمینه استفاده خاصی استفاده می‌شود که برای سامانه کامل انسان - رایانه، از جمله رایانه و نرم‌افزار در حال استفاده، قابل استفاده است. اولی شامل ویژگی‌هایی است که به نتیجه تعامل میان محصول و کاربر در یک زمینه خاص استفاده ارتباط دارد که قابل اعمال بر روی سامانه کامپیوتری - انسانی کامل است، شامل هم کامپیوتر و هم نرم‌افزار در حال استفاده. هدف دوم ارزیابی ویژگی‌های کیفی واقعی محصول نرم‌افزار/ سامانه است. علاوه‌براین، ازآنجایی‌که «کیفیت در استفاده» نیز ارتباط نزدیکی باکیفیت اطلاعاتی دارد که به کاربر منتقل می‌شود، مهم است که برخی از ویژگی‌های تعریف‌شده در استاندارد ISO/IEC 25012 و اقدامات مرتبط گزارش‌شده در استاندارد ISO/IEC 25024 را در نظر بگیرید [۵۴].

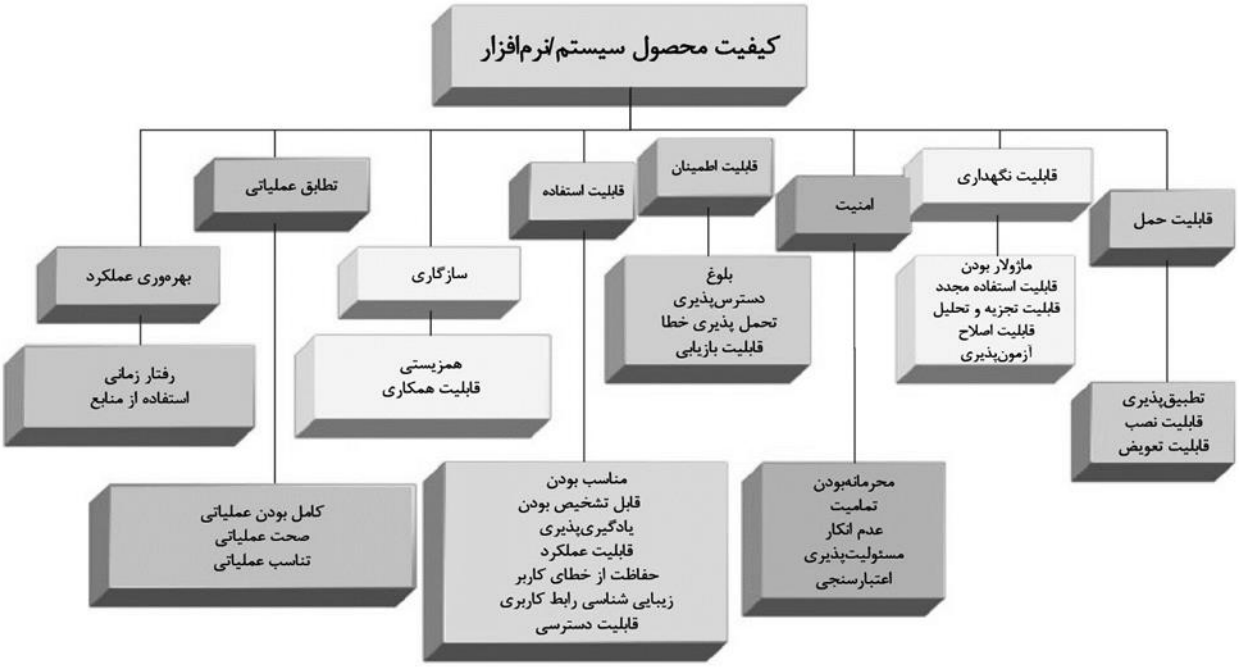
استاندارد ISO 25010:2011 هشت ویژگی کیفیت را برای نرم‌افزار مشخص می‌کند، همان‌طور که در شکل ۹ نشان داده شده است. مدل کیفیت ارائه شده توسط استاندارد ISO 25010 مشابه مدل IEEE 1061 است. یک مشخصه کیفیت مانند «بهره‌وری» را برای ارزیابی انتخاب کنید، در مثال زیر (شکل ۱۰). سپس یک یا چند ویژگی فرعی کیفیت را برای ارزیابی انتخاب کنید (به‌عنوان‌مثال، رفتار زمانی). آخرین مرحله این است که اندازه‌گیری را به وضوح مشخص کنید تا هیچ‌گونه تفسیر نادرستی از نتیجه آن وجود نداشته باشد. یک خط نقطه‌چین نشان می‌دهد که در صورت لزوم می‌توان از ویژگی‌های فرعی عبور کرد [۹].

این استاندارد ویژگی‌های مختلف کیفیت نرم‌افزار مانند قابلیت اطمینان، کارایی، امنیت و رضایت کاربر را پوشش می‌دهد. استاندارد ISO/IEC 25010:2023 شامل ۹ ویژگی کیفی است: مناسب‌بودن عملیات، بهره‌وری عملکرد، سازگاری، قابلیت تعامل، قابلیت اطمینان، امنیت، قابلیت نگهداری، انعطاف‌پذیری و ایمنی [۵۵].

[1] Agreement process
[2] Organization project-enabling processes
[3] Project processes
[4] Technical processes
[5] Software development processes
[6] Software support processes
[7] Software reuse processes
[8] Systems and software Quality Requirements and Evaluation
[9] Quality in use
[10] Quality in use
[11] Product quality



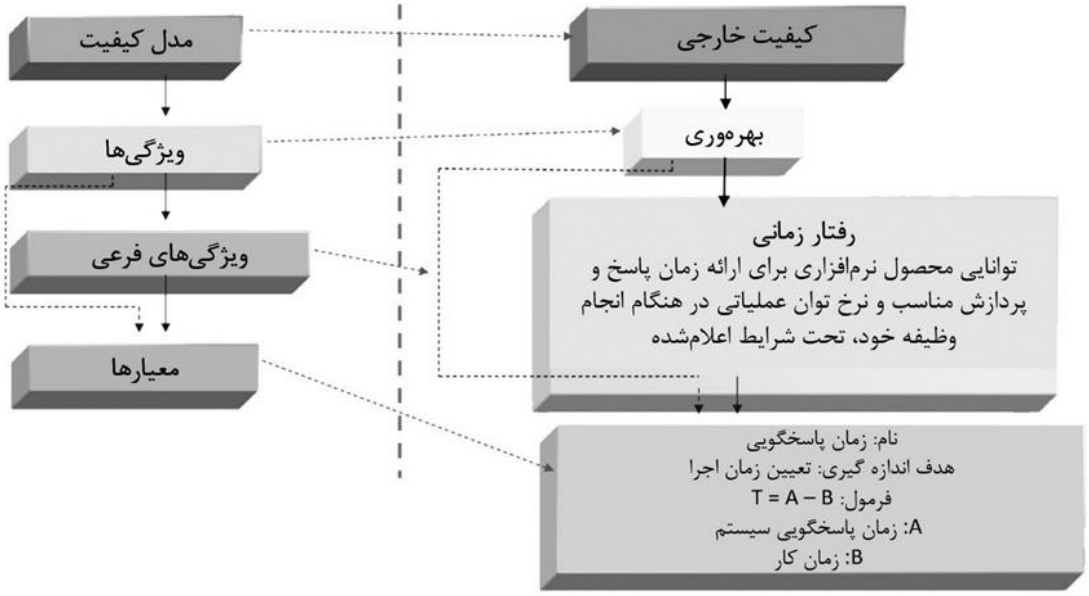
شکل (۸): مدل ارزیابی فرایند دو بعدی استاندارد ISO/IEC 15504 [۱۶]



شکل (۹): مدل کیفیت برای محصول نرم‌افزار استاندارد ISO 25010 [۹]

این مدل می‌تواند برای تعیین الزامات و ارزیابی کیفیت محصولات هدف در طول چرخه حیات آنها توسط چندین ذینفع از جمله توسعه‌دهندگان، خریداران، کارکنان تضمین کیفیت و کنترل و ارزیاب‌های مستقل مورداستفاده قرار گیرد. فعالیت‌هایی در چرخه حیات محصول که می‌توانند از استفاده از این مدل بهره‌مند شوند، عبارت‌اند از [۵۶]:

- ✓ استخراج و تعریف نیازمندی‌های سامانه اطلاعاتی و محصول
- ✓ اعتبار بخشیدن به جامعیت تعریف نیازمندی‌ها
- ✓ شناسایی اهداف طراحی سامانه محصول و اطلاعات و طراحی فرایند لازم برای دستیابی به کیفیت
- ✓ شناسایی اهداف آزمون محصول و سامانه اطلاعاتی
- ✓ شناسایی معیارهای کنترل کیفیت به‌عنوان بخشی از تضمین کیفیت
- ✓ شناسایی معیارهای پذیرش برای یک محصول و/یا یک سامانه اطلاعاتی
- ✓ ایجاد معیارهایی برای ویژگی‌های کیفیت محصول در حمایت از این فعالیت‌ها



شکل (۱۰): ارزیابی کیفیت نرم‌افزار با مدل ISO 25010 [۹]

۴- نتیجه‌گیری

یکی از ۶ کلاس در دسته‌بندی معماری تضمین کیفیت نرم‌افزار، عبارت است از: استانداردهای SQA، صدور گواهینامه و مولفه‌های ارزیابی سامانه SQA. این مؤلفه‌ها، استانداردهای مدیریتی و حرفه‌ای بین‌المللی را در سازمان پیاده‌سازی و اجرا می‌کنند. اهدافی که این گروه از مؤلفه‌ها دنبال می‌کنند شامل استفاده از دانش حرفه‌ای بین‌المللی، بهبود همکاری با سامانه‌های کیفیت سازمان‌های دیگر و سنجش و ارزیابی حرفه‌ای اهداف و دستاوردهای سامانه‌های کیفیت سازمان می‌باشد.

با توسعه و پیشرفت مستمر زمان، استاندارد کیفیت نرم‌افزار کمک شایانی به توسعه و بهبود مهندسی نرم‌افزار کرده و شرایط مساعدی را برای تضمین کیفیت نرم‌افزار ایجاد کرده است. نه‌تنها به استخراج روش‌های آزمون نرم‌افزار، ایجاد ابزارهای خودکار آزمون نرم‌افزار و بهبود دائمی سازوکار مدیریت فرایند آزمون نیاز داریم، بلکه توسعه مهندسی نرم‌افزار نیز باید با استانداردهای کیفیت نرم‌افزار مطابقت داشته باشد که به‌طور مداوم در جامعه امروزی در حال توسعه و به‌روزرسانی هستند. تنها در چنین سازوکار مهندسی نرم‌افزار کاملی، می‌توان نرم‌افزار باکیفیت بالا توسعه داد.

در دهه اخیر، استفاده از استانداردهای نرم‌افزاری توسط سازمان‌های خصوصی و دولتی در توسعه محصولات باکیفیت در چارچوب بودجه و زمان‌بندی مصوب افزایش یافته است. متأسفانه پیاده‌سازی و استفاده از استانداردهای مهندسی نرم‌افزار هزینه‌بر است و کار ساده‌ای نیست؛ زیرا مقاومت مدیران و توسعه‌دهندگان در برابر تغییر ممکن است منجر به کاهش سرعت یا حتی مانع استفاده از استانداردها شود.

در این مقاله، با ارائه یک آرایه‌شناسی به بررسی استانداردهای بین‌المللی رایج برای تضمین کیفیت نرم‌افزار پرداخته شد. ابتدا نیازمندی‌ها و اهمیت تضمین کیفیت نرم‌افزار در سامانه‌های نرم‌افزاری مورد بحث قرار گرفت. سپس در مورد مزایای استفاده از استانداردهای متداول حوزه تضمین کیفیت نرم‌افزار برای رسیدن به کیفیت مطلوب، صحبت شد. آرایه‌شناسی مذکور، شامل دو دسته استانداردهای فرایند پروژه و استانداردهای مدیریت کیفیت بود. در دسته استانداردهای فرایند پروژه، استانداردهای IEEE 730، ISO/IEC/IEEE 12207، IEEE 1012 و IEEE 828 بررسی شدند. همچنین در دسته استانداردهای مدیریت کیفیت، استانداردهای ISO/IEC/IEEE 90003، SEI CMMI، ISO/IEC 15504 و ISO 25010 مورد بررسی قرار گرفتند.

به‌طورکلی، استفاده از استانداردهای متداول حوزه تضمین کیفیت نرم‌افزار به سازمان‌ها کمک می‌کند تا فرایندهای خود را استانداردسازی کنند، کیفیت نرم‌افزار را افزایش دهند، اعتماد مشتریان را بالا ببرند، رقابت‌پذیری خود را ارتقاء دهند و فرایندهای خود را بهبود بخشند. این مزایا در نهایت بهبود عملکرد سازمان و ارائه نرم‌افزارهای باکیفیت و قابل اعتماد به مشتریان را ممکن می‌سازند. لازم به ذکر است که هر سازمان باید به نیازها و شرایط خاص خود توجه کند و استانداردهای مناسبی را برای تضمین کیفیت نرم‌افزار خود انتخاب نماید. همچنین، مدیران و تیم‌های توسعه نرم‌افزار باید به‌طور مداوم این استانداردها را مورد بررسی و ارزیابی قرار داده و در صورت نیاز، تغییرات لازم را ایجاد نمایند.

۵- مراجع

[1] S. Pargaonkar, “Enhancing Software Quality in Architecture Design: A Survey-Based Approach,” Int. J. Sci. Res. Publ. (IJSRP), vol. 13, no. 08, 2023, http://dx.doi.org/10.29322/IJSRP.13.08.2023.p14014

[2]R. Silva Farias, I. Ahmed, and E. Santana de Almeida, “What Makes a Great Software Quality Assurance Engineer?,” arXiv e-prints, p. arXiv: 2401.13623, 2024, https://doi.org/10.48550/arXiv.2401.13623

[3]W. Y. Wong, T. H. Sam, C. W. Too, and W. F. Pok, “Software quality assurance plan: Setting quality assurance checkpoints within the project life cycle and system development life cycle,” 2022 IEEE 18th Int. Colloq. Signal Process. Appl. (CSPA), 2022: IEEE, pp. 214-219.

[4] S. Ibarra and M. Muñoz, “Support tool for software quality assurance in software development,” 2018 7th Int. Conf. Softw. Process Improv. (CIMPS), 2018: IEEE, pp. 13-19, https://doi.org/10.1109/CIMPS.2018.8625617

[5]Y. Zhao, Y. Hu, and J. Gong, “Research on International Standardization of Software Quality and Software Testing,” 2021 IEEE/ACIS 20th Int. Fall Conf. Comput. Inf. Sci. (ICIS Fall), 2021: IEEE, pp. 56-62, https://doi.org/10.1109/ICISFall51598.2021.9627426

[6]I. Atoum, M. K. Baklizi, I. Alsmadi, A. A. Otoom, T. Alhersh, J. Ababneh, J. Almalki, and S. M. Alshahrani, “Challenges of software requirements quality assurance and validation: A systematic literature review,” IEEE Access, vol. 9, pp. 137613-137634, 2021, https://doi.org/10.1109/ACCESS.2021.3117989

[7]N. Silega, G. F. Castro Aguilar, I. M. Alcívar, K. M. Faggioni, Y. I. Rogozov, and V. S. Lapshin, “An ontology-based approach to support the knowledge management of software quality standards,” Enfoque UTE, vol. 14, no. 3, pp. 49-56, 2023, https://doi.org/10.29019/enfoqueute.946

[8]C. Y. Laporte, G. Verret, and M. Muñoz, “A software project that partially failed: A small organization that ignored the management and technical practices of software standards,” Computer, vol. 56, no. 5, pp. 138-144, 2023, https://doi.org/10.1109/MC.2023.3253979

[9]C. Y. Laporte and A. April, “Software quality assurance.” John Wiley & Sons, 2018.

[10]R. E. Al-Qutaish, “Quality models in software engineering literature: an analytical and comparative study,” J. Am. Sci., vol. 6, no. 3, pp. 166-175, 2010.

[11]D. Jamwal, “Analysis of software quality models for organizations,” Int. J. Latest Trends Comput., vol. 1, no. 2, pp. 19-23, 2010.

[12]I. G. Ndukwe, S. A. Licorish, A. Tahir, and S. G. MacDonell, “How have views on software quality differed over time? Research and practice viewpoints,” J. Syst. Softw., vol. 195, p. 111524, 2023, https://doi.org/10.1016/j.jss.2022.111524

[13]A. Kumar and D. Gupta, “Paradigm shift from conventional software quality models to web based quality models,” Int. J. Hybrid Intell. Syst., vol. 14, no. 3, pp. 167-179, 2017, https://doi.org/10.3233/HIS-180249

[14] J. P. Miguel, D. Mauricio, and G. Rodríguez, “A review of software quality models for the evaluation of software products,” Int. J. Softw. Eng. Appl. (IJSEA), 2014, https://doi.org/10.48550/arXiv.1412.2977

[15]S. S. Thapar, P. Singh, and S. Rani, “Challenges to development of standard software quality model,” Int. J. Comput. Appl., vol. 49, no. 10, 2012.

[16]D. Galin, “Software quality: concepts and practice.” John Wiley & Sons, 2018.

[17]N. Nyári and A. Kerti, “Review of software quality related iso standards,” Biztonságtudományi Szemle, vol. 3, no. 2, pp. 61-72, 2021.

[18]G. Karmakar, A. Wakankar, A. Kabra, and P. Pandya, “Development of Safety-Critical Systems: Architecture and Software.” Springer Nature, 2023.

[19]D. Zimon and S. Dellana, “A longitudinal exploratory study of ISO 9001 certification abandonment in small-and medium-sized enterprises,” Int. J. Qual. Reliab. Manag., vol. 37, no. 1, pp. 53-67, 2020, https://doi.org/10.1108/IJQRM-10-2018-0284

[20]D. Zimon, “Quality Management Systems' impact on the functioning of distribution channels in the FMCG market,” Calitatea, vol. 18, no. 156, p. 52, 2017.

[21]D. Zimon, “The influence of quality management systems for improvement of logistics supply in Poland,” Oeconomia Copernicana, vol. 8, no. 4, pp. 643-655, 2017.

[22]F. M. Aarts and E. Vos, “The impact of ISO registration on New Zealand firms’ performance: a financial perspective,” TQM Mag., vol. 13, no. 3, pp. 180-191, 2001.

[23]G. Zimon, A. M. Habib, and D. Haluza, “Does the quality management system affect working capital management efficiency? Evidence from Polish firms,” Cogent Bus. Manag., vol. 11, no. 1, 2024, https://doi.org/10.1080/23311975.2023.2292787

[24]G. Zimon and D. Zimon, “Quality management systems and working capital SMEs in GPO—A Case of Poland,” Adm. Sci., vol. 10, no. 4, p. 76, 2020, https://doi.org/10.3390/admsci10040076

[25]D. Zimon and G. Zimon, “The impact of implementation of standardized quality management systems on management of liabilities in group purchasing organizations,” Qual. Innov. Prosperity, vol. 23, no. 1, pp. 60-73, 2019, https://doi.org/10.12776/qip.v23i1.1210

[26]M. Salehi, N. Mahdavi, S. Zarif Agahi Dari, and H. Tarighi, “Association between the availability of financial resources and working capital management with stock surplus returns in Iran,” Int. J. Emerg. Mark., vol. 14, no. 2, pp. 343-361, 2019, https://doi.org/10.1108/IJoEM-11-2017-0439

[27]IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830:1998, 20 Oct. 1998.

[28]IEEE/ISO/IEC international standard for software, systems and enterprise–architecture description, ISO/IEC/IEEE 42010:2022(E), 2022.

[29]IEEE standard for information technology–systems design–software design descriptions, IEEE Std. 1016:2009, 2009.

[30]IEEE/ISO/IEC international standard—software and systems engineering—software testing—Part 3: Test documentation, ISO/IEC/IEEE 29119-3:2021(E), 2021.

[31]ISO/IEC/IEEE international standard—software and systems engineering—software testing—Part 2: test processes, ISO/IEC/IEEE 29119-2:2021(E), 2021.

[32]ISO/IEC/IEEE international standard—software and systems engineering—software testing—Part 1: general concepts, ISO/IEC/IEEE 29119-1:2022(E), 2022.

[33]IEEE standard for software quality assurance processes, IEEE Std 730:2014, (Revision of IEEE Std 730-2002), 2014.

[34]IEEE Standard for Configuration Management in Systems and Software Engineering, IEEE Std 828:2012, (Revision of IEEE Std 828-2005), 16 March 2012.

[35]IEEE Standard for System, Software, and Hardware Verification and Validation, IEEE Std 1012:2016, (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017), 2016.

[36]IEEE Standard for Software Reviews and Audits, IEEE Std 1028:2008, 2008.

[37]ISO/IEC/IEEE International Standard—Systems and software engineering—Requirements for acquirers and suppliers of information for users, ISO/IEC/IEEE 26512:2017(E), 2017.

[38]N. Mellegård, “Improving Defect Management in Automotive Software Development, LiDeC—A Light-weight Defect Classification Scheme.” Chalmers Tekniska Hogskola (Sweden), 2013.

[39]R. Oberhauser, “VR-V&V: Immersive Verification and Validation Support for Traceability Exemplified with ReqIF, ArchiMate, and Test Coverage,” Int. J. Adv. Syst. Meas., vol. 16, no. 3 & 4, pp. 103-115, 2023.

[40]L. Pourvatan, “Test Process Assessment of Industrial Control Systems via Safety Standards,” ed, 2021.

[41]S. K. Sharma and M. Khaliq, “Design and development of software quality forensics framework and model,” Multidiscip. Sci. J., vol. 6, no. 7, pp. 2024111-2024111, 2024, https://doi.org/10.31893/multiscience.2024111

[42]S. Fahmy, A. Deraman, J. Yahaya, A. Nasir, and N.

Shamsudin, "The evolution of software configuration management," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 1.3, 2020, https://doi.org/10.30534/ijatcse/2020/0891.32020
[43]Software engineering-Guidelines for the application of ISO 9001:2015 to computer software, ISO/IEC/IEEE 90003:2018, (Revision of ISO/IEC 90003:2014), 2018.
[44]S. K. Sundaram and M. Suresh, "Partial CMMI V2. 0 Assessment Using Multi-grade Fuzzy for Healthcare and Insurance Segment in Software Services," in Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES 2022: Springer, 2023, pp. 401-412. https://doi.org/10.1007/978-981-19-8497-6_37
[45]J. Mona, R. H. A. Al-Sagheer, and S. M. Alghazali, "Software Quality Assurance Models and Application to Defect Prediction Techniques," Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 1, pp. 169–178-169–178, 2023.
[46]I. Keshta, "A model for defining project lifecycle phases: Implementation of CMMI level 2 specific practice," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 2, pp. 398-407, 2022, https://doi.org/10.1016/j.jksuci.2019.10.013
[47]B. Day, S. C. Ke-Zun, L. Lovelock, and C. Lutteroth, "Climbing the ladder: CMMI level 3," 2009 IEEE Int. Enterp. Distrib. Object Comput. Conf., 2009: IEEE, pp. 97-106, https://doi.org/10.1109/EDOC.2009.29
[48]P. Clarke and R. V. O'Connor, "An empirical examination of the extent of software process improvement in software SMEs," J. Softw. (Malden), vol. 25, no. 9, pp. 981-998, 2013, https://doi.org/10.1002/smr.1580
[49]C. Vivatanavorasin, N. Prompoon, and A. Surarerks, "A process model design and tool development for supplier agreement management of CMMI: Capability Level 2," 2006 13th Asia Pac. Soft. Eng. Conf. (APSEC'06), 2006: IEEE, pp. 385-392, https://doi.org/10.1109/APSEC.2006.12
[50]C. Quintal and J. A. Macías, "Measuring and improving the quality of development processes based on usability and accessibility," Univers Access Inf. Soc., vol. 20, no. 2, pp. 203-221, 2021, https://doi.org/10.1007/s10209-020-00726-7
[51]S. Peldzius and S. Ragaisis, "Comparison of maturity levels in CMMI-DEV and ISO/IEC 15504," Appl. Math. and Comput. Eng., pp. 117-122, 2011.
[52]T. P. Rout and A. Tuffley, "Harmonizing iso/iec 15504 and cmmi," Softw. Process: Improv. Pract., vol. 12, no. 4, pp. 361-371, 2007, https://doi.org/10.1002/spip.329
[53]M. I. F. Nuzula and S. Rochimah, "Evaluation of Service Quality in Human Resource Information Systems Using the ISO/IEC 25010," Int. Semin. Appl. Technol. Inf. Commun. (iSemantic), 2023: IEEE, pp. 215-220, https://doi.org/10.1109/iSemantic59612.2023.10295365
[54]V. Santa Barletta, D. Caivano, L. Colizzi, G. Dimauro, and M. Piattini, "Clinical-chatbot AHP evaluation based on "quality in use" of ISO/IEC 25010," Int. J. Med. Inform., vol. 170, p. 104951, 2023, https://doi.org/10.1016/j.ijmedinf.2022.104951
[55]A. Sopandi, N. A. Yahaya, and A. Subiyakto, "Developing the Readiness and Success Model of Information System Implementation in the Indonesian Equestrian Industry," J. Appl. Data Sci., vol. 5, no. 1, pp. 133-145, 2024, https://doi.org/10.47738/jads.v5i1.145
[56]Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Product quality model, ISO/IEC 25010:2023, International Organization for Standardization (ISO), 2023.