

# Estudio de herramientas para alcanzar niveles de calidad de software a nivel de proceso

Yadira Jazmín Pérez Castillo, Sandra Dinora Orantes Jiménez

Instituto Politécnico Nacional, Centro de Investigación en Computación,  
Ciudad de México, México  
yaddy9011@gmail.com, dinora@cic.ipn.mx

**Resumen.** El área de la ingeniería de software se ha encargado de proporcionar herramientas para mejorar los procesos de construcción de software y, por ende, la calidad en el mismo, por lo tanto, para lograr esto, debe guiarse de métodos, modelos y/o prácticas que permitan alcanzarla. Este trabajo de investigación intenta proveer una revisión de los instrumentos existentes en el mercado, que permitan a una empresa desarrolladora de software alcanzar niveles de madurez a nivel proceso o bien, contar con estrategias utilizadas para aumentar la calidad de los productos de software.

**Palabras clave:** ingeniería de software, calidad de productos de software, procesos, herramientas, madurez de procesos.

## Study of Tools to Reach Software Quality Levels at the Process Level

**Abstract.** The area of software engineering has been responsible for providing tools to improve the processes of software construction and, therefore, the quality in it, therefore, to achieve this, it should be guided by methods, models and / or practices that allow to reach it. This research work tries to provide a review of the instruments in the market, which allow a software developer to reach maturity levels at the process level or to have strategies used to increase the quality of software products.

**Keywords:** software engineering, quality of software products, processes, tools, process maturity.

### 1. Introducción

La calidad de software y las contrariedades que lleva consigo conseguir un nivel de calidad aceptable, es un problema que tiene sus raíces desde la llamada crisis del software y para mitigar sus estragos, varias estrategias han surgido en los últimos años y han proveído técnicas que ayudan a una mejora de la calidad en los procesos de desarrollo de software. Según [1], la creciente preocupación por la calidad en la industria del software tiene como objetivo principal el desarrollo sistemático de productos y servicios de mejor calidad y el cumplimiento de las necesidades y

expectativas de los clientes. La calidad de software se puede representar de acuerdo con el enfoque de evaluación, ya sea a nivel de proceso, producto o calidad en uso [2].

Por lo tanto, el presente trabajo se centrará en el tipo de evaluación a nivel de proceso, tomando en consideración que así se asegura el desarrollo de un producto de software de calidad y por ende, que posee la característica de usabilidad; por lo anterior, el objetivo de este trabajo es comparar las herramientas existentes en el mercado actual, que permiten acceder a una guía para alcanzar un nivel de calidad aceptable en los procesos de desarrollo, es por ello, que resulta favorable conocer cuáles modelos o estándares poseen una visión integradora de la calidad del producto de software o cuáles resultan adecuados para el ámbito de una organización.

Este documento se encuentra distribuido de la siguiente manera: en la *sección 2*, se dan a conocer algunos modelos importantes a nivel de proceso para la calidad de software, en la *sección 3*, se proporciona una discusión que permite dar resultados de esta investigación a través de un cuadro comparativo con base en características principales de los modelos descritos en la *sección 2* y sus apartados. Finalmente, en la *sección 4*, se establecen las conclusiones del trabajo realizado y comentarios acerca de trabajos futuros.

## **2. Modelos a nivel de proceso para la calidad de software**

Realizando el análisis de investigación de la bibliografía existente, correspondiente al tema a abordar en este trabajo, es posible encontrar algunos modelos actualmente reconocidos por la industria de la ingeniería de software, entre ellos, ISO/IEC 15504 (SPICE), BOOTSTRAP, DROMEY, IEEE / EIA 12207, COBIT (Control Objectives for Information and related Technology, Objetivos de Control para Información y Tecnologías Relacionadas), ISO 90003, ISO 20000, CMMI (Capability Maturity Model Integration, Modelos de Madurez de Capacidades Integrados), PSP (Personal Software Process, Proceso de Software Personal), TSP (Team Software Process, Equipo de Procesos de Software), cuyo objetivo en común es proveer una evaluación de la calidad del software a nivel proceso; dentro de los siguientes apartados, se realiza una breve descripción de cada uno de ellos, con el fin de conocerlos, resaltando sus principales características, objetivos y funciones.

### **2.1. ISO/IEC 15504 (SPICE)**

Permite adaptar la evaluación para procesos en Pequeñas y Medianas Empresas (PYMES) y grupos de desarrollo pequeños, mediante la estructuración en 6 niveles de madurez: Nivel 0- Organización inmadura, Nivel 1- Organización básica, Nivel 2- Organización gestionada, Nivel 3- Organización establecida, Nivel 4- Organización predecible y Nivel 5- Organización optimizando. Su objetivo es llegar a que la organización logre ser madura, lo cual, conlleva que la organización tenga procesos definidos, responsabilidades definidas, predicción de resultados, productos entregados con calidad, que las entregas se den en los tiempos pactados, incrementar la productividad, clientes satisfechos, y empleados felices [2].

ISO/IEC 15004 (SPICE) trata de una herramienta con los siguientes objetivos determinados en [3]:

- Proponer y desarrollar un estándar de evaluación de procesos de software.
- Evaluar su desempeño mediante su experimentación en la industria emergente del desarrollo de software.
- Promover la transferencia de tecnología de la evaluación de procesos de software a la industria del software a nivel mundial.

## **2.2. BOOTSTRAP**

Es una metodología de valoración de la calidad que permite la mejora de procesos a partir de 6 actividades básicas: examinar la necesidad, iniciar proceso de mejora, preparación y dirección de la evaluación, análisis de resultados, implantación y finalización de mejoras [2].

*Bootstrap Institute* es el instituto que dirige los destinos de la metodología y está constituido por: miembros fundadores, corporativos y demás integrantes; dentro de sus principios están: que la metodología sea accesible a todos y crezca de forma que permita mejoras, que la evolución de la metodología sea democrática, que provea un servicio a la industria y que opere como una empresa no lucrativa [4].

## **2.3. DROMEY**

Es un modelo adaptable para evaluar varias etapas del proceso de desarrollo, tales como levantamiento de requisitos, diseño e implementación. Se estructura con características y sub-características de calidad; propone tres modelos distintos para cada etapa de construcción del producto: modelo de requerimientos, modelo de diseño y modelo de calidad de la implementación, a partir de la evaluación establecida en cinco etapas, para características como: eficiencia, confiabilidad, mantenibilidad, portabilidad, facilidad de uso y funcionalidad [2].

Es un modelo de calidad a medida, propuesto por Robert Geoff Dromey en 1995. Su propósito es el de trabajar con una estructura que permite construir y utilizar un modelo de calidad práctico para evaluar las etapas de determinación de los requerimientos, diseño e implementación. Esta información puede ser usada para elaborar, comparar y evaluar la calidad de los productos de software. Este modelo plantea la calidad del producto por medio de la definición de sub-características que pueden ser medidas y evaluadas como características. También, permite aumentar el entendimiento respecto de la relación entre los atributos (características) y los sub atributos de calidad [5].

## **2.4. IEEE / EIA 12207**

Este estándar establece un marco de trabajo común para el ciclo de vida del desarrollo de software, a partir del planteamiento de procesos, actividades y tareas que pueden ser aplicadas durante la adquisición, suministro, desarrollo, operación, mantenimiento y/o despliegue de un producto software [2]. IEEE/EIA 12207: Procesos del Ciclo vital del software, se beneficia de una herencia que atraviesa un cuarto del siglo del trabajo de estándares de proceso del software, el estándar extensamente se considera como el suministro de una base para el comercio mundial en servicios del software. Al lado de la ISO 9.000, es probablemente el estándar más relacionado con la información importante con relación a la tecnología [6]. La Fig. 1, muestra la

estructura del modelo, como se puede observar, consta de 3 partes, las cuales permiten a una empresa ir incorporando el modelo progresivamente.

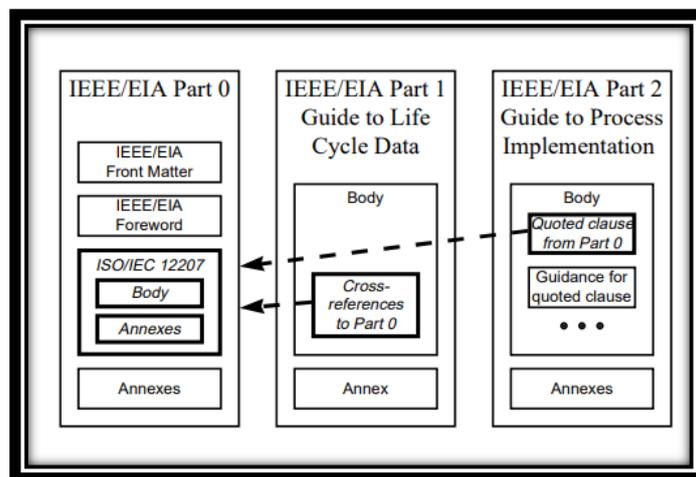


Fig. 1 Estructura IEEE / EIA 12207 [2].

## 2.5. COBIT

Se caracteriza por ser orientado a negocios y procesos, además de ser basado en controles, trabaja con siete criterios de información que son definidos como requerimientos de control del negocio: efectividad, eficiencia, confidencialidad, integridad, disponibilidad, cumplimiento y confiabilidad [2].

COBIT se ha puesto en práctica en muchos países desde su introducción en 1996. Uno de los factores que dan popularidad a COBIT es que su extenso resumen, marco, objetivos de control, guías de la dirección y juego de herramientas de realización son gratuitos. El pago sólo se requiere para las guías de auditoría. COBIT parece hacerse un marco influyente para el control y el gobierno de TI (Tecnologías de la Información) se certifican a por el significado y diversidad de las organizaciones en cual se ha utilizado. Además, COBIT está actualmente en su quinta edición y consta de 3 niveles de aplicación [6].

## 2.6. ISO 90003

Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software, cuyo propósito es ofrecer una guía de aplicación de la norma 9001 que pretende ser utilizada para demostrar o soportar que la entidad (proyecto/empresa) está en capacidad de desarrollar software con criterios de calidad [2]. La ISO 9001 cuando es aplicada a auditorías del software, tiene cinco categorías [6].

La sección 7.3 de la ISO 90003:2004 cubre una serie de subdivisiones que proporcionan pautas detalladas para las fases del diseño y desarrollo de software, las cuales son:

- 7.3.1. *Design and development planning (Diseño y planeación del desarrollo).*

- 7.3.2. *Design and development inputs (Diseño y entradas del desarrollo).*
- 7.3.3. *Design and development outputs (Diseño y salidas del desarrollo).*
- 7.3.4. *Design and development review (Diseño y revisión del desarrollo).*
- 7.3.5. *Design and development verification (Diseño y verificación del desarrollo).*
- 7.3.6. *Design and development validation (Diseño y validación del desarrollo).*
- 7.3.7. *Control of design and development changes (Cambios del desarrollo y control del diseño).*

## **2.7. ISO 20000**

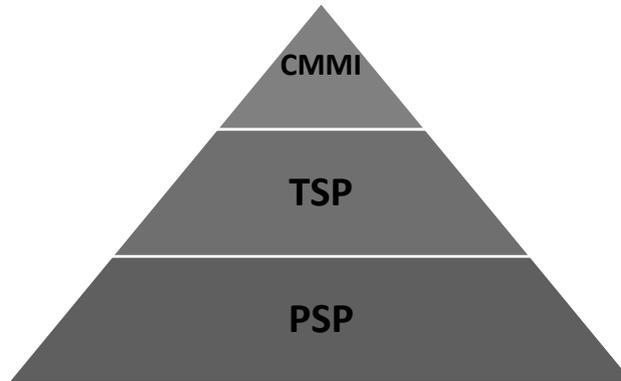
El objetivo principal de esta norma es el de avalar que la prestación de servicios gestionados de TI (Tecnologías de la información) de una empresa cuentan con la calidad necesaria para brindar dichos servicios a los clientes. Se subdivide en dos partes: "Especificaciones", publicada como ISO 200001:2005, y "Código de buenas prácticas" publicada como ISO 20000-2:2005 [2].

La ISO 20000 es un instrumento potente para los que buscan la dirección del servicio de calidad, si son empresas del sector de la TI, departamentos de la TI o abastecedores internos dentro de alguna otra clase de la organización o hasta otros tipos de organizaciones orientado a servicios. Es relevante notar que, aunque el origen del estándar se arraigue en una intención de manejar servicios de la TI, la versión actual no contiene ningún impedimento para la utilización de ello para otros tipos de servicios.

La ISO 20000 se forma de una serie de documentos. El primer documento, formalmente conocido como ISO/IEC 20000-1 (ISO/IEC 2011), es la única parte que establece requisitos para un sistema de gestión del servicio, por lo tanto, estos requisitos son los únicos que deben cumplirse para obtener la certificación. Los otros documentos, aunque sean muy valiosos y provechosos para poner en práctica el estándar, se pueden considerar complementarios [7].

## **2.8. PSP, TSP Y CMMI**

En esta sección se describe el uso de tres herramientas, que en conjunto permiten a una empresa dedicada al Desarrollo de Software (DS), alcanzar niveles de madurez, se consideran las más actuales y utilizadas en el área de la IS (Ingeniería de Software), la Fig. 2, muestra hablando en términos de calidad y madurez de software, una pirámide de la IS relacionada con los modelos de madurez PSP, TSP Y CMMI, en la que se detalla el nivel de utilidad y de pautas para el uso de las estrategias para la gestión de DS, partiendo desde la vista personal hasta la empresarial; primero en la parte más baja, se debe de comenzar aplicando la gestión a nivel personal, posteriormente a nivel de equipos de trabajo, para poder lograr una integración a nivel organizacional y obteniendo un nivel de madurez como resultado de esta integración de estrategias [8].



**Fig. 2** Pirámide que expresa el nivel de aplicación de las estrategias para la calidad de software PSP, TSP, CMMI [Elaboración propia].

### **Personal Software Process (PSP)**

La demanda de la industria del software para lograr previsibilidad y consistencia frente a los rápidos cambios es significativa. El marco de trabajo PSP ayuda al individuo a satisfacer estas demandas [9]. PSP es un proceso personal que, al estar basado en los principios de mejora, ayuda a la gente a establecer sus metas personales, identificar qué métodos utilizarán, medir su trabajo y analizar los resultados, para ajustar los métodos que utilizan para cumplir sus metas [9].

PSP es un proceso definido para ayudar a realizar mejor el trabajo, cuyo objetivo es obtener y reportar datos precisos y completos del trabajo que se realiza a nivel individual, con el fin de mejorar el proceso individual, afectando de esta manera al desempeño de todo el equipo [9].

PSP amplía el proceso de mejora a la gente que realiza el trabajo de desarrollo de software, concentrándose en las prácticas de trabajo de los ingenieros en una forma individual, enseñando cómo manejar la calidad desde el principio de un producto. PSP métricas propias de cada individuo, que permiten estructurar y ordenar el trabajo del día a día. El resultado del trabajo personal, además puede ser llevado a un trabajo en equipo TSP (Team Process Software), el cual es “comandado” por un sistema de gestión de la configuración y por supuesto, un Jefe de Proyecto quien evalúa los resultados y avances de los miembros del equipo [10].

Algunos de los objetivos del proceso PSP son:

- Introduce la disciplina y la medición del proceso.
- Introduce estimación y planeación.
- Introduce al manejo de calidad.

Los principios de PSP, recopilados por [8] son:

1. Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
2. Usar procesos bien definidos y cuantificados.

3. Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
4. Cuanto antes se detecten y corrijan los errores menos esfuerzo será necesario.
5. Es más efectivo evitar los defectos que detectarlos y corregirlos.
6. Trabajar bien es siempre la forma más rápida y económica de trabajar.

Algunas de las principales características se describen, de manera resumida, en el siguiente listado:

- Concentra la práctica de una forma individual.
- Es de uso personal.
- Se aplica en proyectos pequeños.
- Produce software de calidad.
- Administra tiempo.
- Demuestra cómo manejar el control de calidad.

Los procesos, prácticas y herramientas PSP se diseñaron para ayudar a los ingenieros del software a hacer bien su trabajo, por tanto, permite aplicar métodos avanzados de ingeniería a tareas personales diarias. Proporcionando métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo [6].

### **Team Software Process (TSP)**

TSP inició como una herramienta capaz de ayudarle a los equipos de gerentes de proyectos, así como a los ingenieros, a organizar y producir proyectos de software a gran escala; se dio a conocer en 1996 y fue desarrollado por el ingeniero y físico Watts S. Humphrey, el cual publicó el primer reporte técnico en el año 2000 [11].

Humphrey buscaba proporcionar a sus estudiantes de ingeniería de software, una visión total del ciclo de vida del software [11].

Esta novedosa herramienta es considerada como una metodología para administrar el trabajo de mejora y desarrollo de los procesos de software, además de garantizar un entorno de trabajo agradable y natural para los equipos. TSP brinda un conjunto de pasos bien estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo, además brinda una ayuda acerca de cómo conformar equipos para el desarrollo de software de calidad [11]. Como toda herramienta, los objetivos que rigen la implementación de TSP [12], se listan a continuación:

- Maximizar la calidad del software en detrimento de los costos.
- Formar equipos que sean capaces de planear y registrar su trabajo, establecer metas bien definidas y sean aptos para realimentar su propio trabajo mediante la medición de éste.
- Brindar un punto de vista a los gerentes y líderes de proyecto acerca de cómo monitorear y como motivar a sus equipos de trabajo para sacar su máximo potencial.
- Establecer una guía para el mejoramiento en organizaciones maduras; así como acelerar la mejora continua de procesos.

Como toda metodología que busca la continua mejora de procesos, TSP posee fases en donde se describen una serie de pautas para ayudar a realizar un buen desarrollo de software por parte del equipo de trabajo. Las fases descritas por [12] son mostradas en la Fig. 3.



**Fig. 3** Fases de TSP, descritas en [12], [elaboración propia].

Las principales características de la herramienta TSP, descritas por Humphrey en [11], son:

- Miembros expertos en papeles de liderazgo y pertenencia.
- Relaciones tranquilas y establecidas entre los miembros.
- Los miembros se sienten atraídos por el grupo y son fieles.
- Los valores y metas del grupo son los de sus integrantes.
- Los miembros están motivados por hacer lo que puedan por el grupo.
- La interacción y toma de decisiones tiene lugar en el ambiente adecuado.
- El grupo desea ayudar a cada miembro a adquirir su pleno potencial.

### Capability Maturity Model Integration (CMMI)

CMMI es un modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo, mantenimiento, adquisición y operación de productos y servicios. Su enfoque es la mejora de procesos, que provee a las organizaciones de los elementos esenciales para un proceso efectivo y provee guía para la calidad de los procesos. CMMI fue desarrollado por el SEI: Software Engineering Institute, Instituto de Ingeniería de Software [2].

El objetivo de CMMI es crear productos de calidad que permita obtener resultados medibles y repetibles. CMMI mide la madurez del proceso de desarrollo del software en una escala del 1 al 5 (véase Fig. 4) e integra disciplinas como sistemas y software en un solo marco de trabajo, describe formas efectivas y probadas de hacer las cosas, no es un enfoque radical [8].

El modelo CMMI consta de 22 áreas de proceso. Las áreas de proceso de Nivel 2 y 3 son las siguientes: Planificación de Proyecto (PP), Seguimiento y Control de Proyectos (PMC), Desarrollo de Equipos Integrado (IT), Gestión de Riesgos (RSKM), Desarrollo de Requisitos (RD), Gestión de Requisitos (RM), Solución Técnica (TS), Integración del Producto (SI), Verificación (VER), Validación (VAL), Medición y

Análisis (MA), Gestión de la Configuración (CM), Aseguramiento de Calidad de Proceso y Producto (PPQA). CMMI es utilizado en más de 5000 empresas de las cuales 24 son empresas ubicadas en el Perú. En su gran mayoría las empresas peruanas se encuentran en nivel 2 y 3 [8].



Fig. 2 Ilustración de los Niveles de Madurez CMMI. [Recuperado de <http://rupandcmmi.blogspot.com/p/niveles-de-madurez.html>].

En la Tabla 1 se muestra el cuadro comparativo correspondiente a los modelos de calidad a nivel proceso, descritos en este trabajo, por consiguiente, es posible determinar que los modelos relacionados muestran, como principal objetivo, proveer a una empresa desarrolladora de software las armas para construir software de calidad, además de que cuentan con niveles para alcanzar la calidad de una manera progresiva, del mismo modo se pueden adaptar a equipos de trabajo, tanto pequeños como grandes. Lo ideal es conocer más a fondo cada herramienta y realizar un estudio detallado de cada una, es decir, aplicar los modelos presentados en este trabajo o bien, entrevistar a las empresas que los utilizan, para conocer los factores de éxito o fracaso de estos modelos en la industria del software.

Por otro lado, un ejemplo de modelos para alcanzar un nivel de calidad de tipo híbrido, es decir modelos de calidad a nivel proceso, producto y/o funcionalidad, Luis E Mendoza, ver [13], propone MOdelo Sistémico de CALidad (MOSCA), para evaluar la calidad de los sistemas de software, integrando el modelo de Calidad del Producto y el modelo de Calidad del Proceso, MOSCA fue probado en dos empresas venezolanas desarrolladoras de software, utilizando el Método "Análisis de Características por Caso de Estudio", con 3 fases para ir incrustando la calidad en los sistemas de software. Lo anterior para destacar otra oportunidad de estudio, es decir, integrar aquellos modelos que permitan implementar la calidad de software en sus tres horizontes; nivel de proceso, producto o calidad en uso.

### 3. Conclusiones y trabajos futuros

Es evidente que los procesos de desarrollo en una industria emergente, como el software, que continuamente aporta nuevos escenarios tecnológicos, deben de contar con una herramienta de evaluación de la madurez de sus procesos, para que continuamente este siendo evaluada; de tal forma que la entidad pueda nutrirse y así crecer y aportar conocimiento al resto del panorama industrial.

**Tabla 1.** Cuadro Comparativo: Modelos de Calidad Nivel Proceso.

|   | ISO 9003   | ISO/IEC 2000   | CMMI  | TSP   | PSP  |
|---|--|--|---|---|--|
| <b>Objetivo</b>   | Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software. | Avalar que la prestación de servicios gestionados de TI de una empresa cuenta con la calidad necesaria para brindar dichos servicios a los clientes. | Modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo. | Metodología para administrar el trabajo de mejora y desarrollo de los procesos de software. | Proceso personal que ayuda a la gente a establecer sus metas personales. |
| <b>Ámbito</b>   | Organizacional   | Organizacional   | Organizacional  | Equipo  | Personal   |
| <b>Niveles/ actividades para alcanzar madurez y calidad</b> | cinco categorías   | Documentación  | Escala del 1 al 5   | Procesos, prácticas   | Procesos, prácticas  |
| <b>Tipo de proyectos aplicables</b>                         | Grandes  | Grandes  | Grandes   | Grandes   | Pequeños   |
| <b>Certificación</b>  | SI   | SI   | SI  | SI  | SI   |

|   | ISO/IEC 15504   | BOOTSTRAP   | DROMEY  | IEEA/E/A 12207  | COBIT   |
|---|---|---|---|---|---|
| <b>Objetivo</b>   | Proponer y desarrollar un estándar de evaluación de procesos de software. | Metodología de valoración de la calidad que permite la mejora de procesos a partir de 6 actividades básica. | Es un modelo adaptable a evaluar varias etapas del proceso de desarrollo. | Marco de trabajo para el ciclo de vida del desarrollo de software a partir del planteamiento de procesos, actividades y tareas. | Marco influyente para el control y el gobierno de las TI. |
| <b>Ámbito</b>   | Pequeñas y medianas empresas  | Equipo  | Organizacional  | Organizacional  | Equipo  |
| <b>Niveles/ actividades para alcanzar madurez y calidad</b> | 6 niveles   | 6 actividades   | Tres modelos distintos para cinco etapas 5 etapas.                        | Se compone de 3 partes  | Siete criterios   |
| <b>Tipo de proyectos aplicables</b>                         | Grandes /pequeños   | Grandes   | Grandes   | Grandes   | Grandes   |
| <b>Certificación</b>  | SI  | -   | -   | -   | SI  |

En este artículo, se proporciona una revisión breve de herramientas que permiten ir evaluando la calidad de los procesos de desarrollo de software. Dentro de la literatura se encuentran algunos trabajos como en [2] and [5], que se ha encargado realizar un revisión más exhaustiva a nivel de proceso, producto y uso.

Como conclusión es posible afirmar que una empresa, del ámbito del desarrollo de sistemas informáticos, que no cuente con algún sistema de gestión de calidad o de evaluación de sus procesos de desarrollo de software tiene la tarea difícil de situarse en un mercado competitivo para tener éxito en cualquier licitación pública o privada; por tanto, se vuelve indispensable que conozca aquellas herramientas que le permitan iniciarse en el camino hacia la calidad en el desarrollo de software.

Un trabajo futuro para esta investigación es aplicar los modelos presentados en este trabajo o bien, entrevistar a las empresas que los utilizan, para conocer los factores de éxito o fracaso de estos modelos en la industria del software.

## Referencias

1. López-Echeverry, A.M., Cabrera, C., Valencia-Ayala, L.E.: Introducción a la calidad de software. *Sci. Tech.* 2(39), 326–331 (2008)
2. Callejas-Cuervo, M., Alarcón-Aldana, A.C., Álvarez-Carreño, A.M.: Modelos de calidad del software, un estado del arte. *Entramado* 13(1), 236–250 (2017)
3. ISO 15504 Norma de desarrollo de software SPICE ISO/IEC 15504, <https://www.normas-iso.com/iso-iec-15504-spice/>
4. Estándares de Calidad en el Mercado Estándar, (1993)
5. Das, S.R., Yaylacicegi, U., Canel, C.: Using ISO 90003 for software ‘design and development’ in large virtual teams. *Ind. Manag. Data Syst.* 108(6), 775–793 (2008)
6. Cots, S., Casadesús, M., Marimon, F.: Benefits of ISO 20000 IT service management certification. *Inf. Syst. E-bus. Manag.* 14(1), 1–18 (2016)
7. Chavarría, A.E., Oré, S.B., Pastor, C.: Aseguramiento de la calidad en el proceso de desarrollo de software utilizando CMMI, TSP y PSP. *RISTI - Rev. Iber. Sist. e Technol. Inf.*, vol. 2016, no. 20, pp. 62–77 (2016)
8. Kamatar, J., Hayes, W.: Experience report on the personal software process. *IEEE Softw.* 17(6), 85–89 (2000)
9. Humphrey, W.S.: Introducing the personal software process. *Ann. Softw. Eng.* 1(1), 311–325 (1995)
10. Humphrey, W.S.: Team software process (2000)
11. Díaz Sánchez, D.Y., Maldonado Reyes, I.M., Reveles Pérez, O.D., Escamilla Camargo, A.: TSP-Team Software Process (2017)
12. Mendoza, L.E., Pérez, M.A., Grimán, A.C.: Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software / Prototype of Software Quality Systemic Model (SQSM) vol. 8, no. 3, Centro de Investigación en Computación, IPN (2005)