

# USO DE MODELOS DE CALIDAD EN LAS MYPES PRODUCTORAS DE *SOFTWARE* DE LIMA

MIRIAM AMABLE CIUDAD, ROSA MILLONES RIVALLES\*  
Universidad de Lima

Recibido: 7 de mayo de 2018 / Aprobado: 11 de septiembre del 2018

doi: 10.26439/ing.ind2019.n037.4543

**RESUMEN:** Las micro y pequeñas empresas representan un 85 % en la industria del *software*. Se les caracteriza por el uso de los modelos de calidad tanto del proceso como del producto. En una investigación descriptiva con 80 mypes se observa mayor uso de modelos de calidad de proceso y menor uso para calidad de producto, aunque muestran un conocimiento de los factores de calidad.

**Palabras clave:** *software*-control de calidad / control del proceso / calidad del producto / pequeña y mediana empresa / micro y pequeña empresa (mype) productora de *software*

## USE OF QUALITY MODELS BY SOFTWARE DEVELOPMENT *MYPES* FROM LIMA

**ABSTRACT:** Micro and small businesses (*mypes*) account for 85% of the software industry. They characterize by the use of quality models for both the process and the product. In a descriptive research conducted with 80 *mypes*, it was observed that they used more process quality models and less product quality models, and they demonstrated to know the quality factors.

**Keywords:** software quality control / process control / product quality / micro and small businesses / software development micro and small business

---

\* miriama@ulima.edu.pe, rbmillon@ulima.edu.pe

## 1. INTRODUCCIÓN

La industria de *software* tiene gran relevancia en la economía porque presenta un buen nivel de crecimiento, como el obtenido en el 2017, con un 5,3 % (Loya Páez, 2018). Por ello es importante obtener productos de calidad provenientes de procesos de calidad para lograr un buen nivel de competitividad. Dada su importancia, se investiga sobre la adopción de modelos de calidad tanto de procesos como de productos en las mypes (micro y pequeñas empresas) productoras de *software*, las cuales representan el 85 % de la industria de *software*. En este artículo se presenta inicialmente una revisión de literatura de los diferentes descriptores temáticos, calidad de *software*, calidad de proceso, calidad de producto, metodologías de desarrollo, modelo de calidad y mypes productoras de *software*. A continuación se muestran los resultados de la investigación cuantitativa realizada a una muestra aleatoria representativa de 80 mypes productoras, para finalmente hacer el análisis y sacar las conclusiones correspondientes.

## 2. REVISIÓN DE LITERATURA

### 2.1 Módulo de calidad de *software*

En primer lugar, Estayno, Dapozo, Cuenca y Greiner (2009) definen la calidad de *software* partiendo del concepto de calidad planteado por la norma UNEEN ISO 8402, la cual expresa que “la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas” (p. 2). Asimismo, Miguel, Mauricio y Rodríguez (2014) presentan la definición de calidad de *software*, de acuerdo con el glosario estándar IEEE de Terminología de Ingeniería de *Software*, el cual la define como “1) el grado en que un sistema, componente o proceso cumple con los requisitos especificados y 2) el grado en que un sistema, componente o proceso cumple con las necesidades o expectativas de un usuario” (p. 31). Además, Pressman (2010) la refiere como el cumplimiento que posee el producto *software* respecto a los requisitos tanto funcionales como no funcionales, así como las características implícitas que se espera de todo producto *software* desarrollado profesionalmente. Como consecuencia, en este contexto se resalta el hecho de que, para lograr la calidad una vez que es especificada, esta pueda ser medida, por lo que, Estrada (2014) define la calidad del *software* como “un conjunto de cualidades medibles y específicas que varía de un sistema a otro, dependiendo del tipo de *software* que se va a desarrollar, para determinar su utilidad y existencia” (p. 1).

Por último, Acosta, Espinel y García (2017) expresan:

La calidad del *software* hoy en día juega un papel importante dentro de las empresas, por esto es necesario tener un punto de comparación y mirar si el

producto entregado cumple con los requerimientos y necesidades que tienen los usuarios o clientes. (p. 76)

## 2.2 Calidad de proceso

En principio, un modelo importante en la calidad de proceso es el brindado por la norma internacional ISO/IEC 12207 *Software Life Cycle Processes*, que comprende un modelo de referencia de procesos del ciclo de vida del *software*. Por ello, Sánchez, Sicilia y Rodríguez (2012) explican que dicha norma internacional contempla dos categorías de procesos, la primera relacionada a los procesos de contexto del sistema y la segunda vinculada a los procesos específicos del *software*, los cuales abarcan siete procesos de implementación, representando las etapas del ciclo de vida del *software*, entre las cuales figuran el análisis de requisitos, el diseño arquitectural, el diseño detallado, la construcción, la integración, las pruebas de calificación y la implementación, más ocho procesos de soporte del *software*: la gestión de la documentación, la gestión de la configuración, el aseguramiento de la calidad, la verificación, la validación, la revisión, la auditoría y la resolución de problemas del *software*; además, otros tres procesos de reutilización del *software*: la ingeniería del dominio, la gestión de activos de reúso y la gestión de programas de reúso.

Acosta, Espinel y García (2017) destacan que la organización productora de *software*, para lograr calidad, debe adoptar un estándar para que los entregables cumplan con las expectativas del negocio. Proponen el *Capability Maturity Model Integration* (CMMI), creado por la Universidad Carnegie Mellon, que refleja principalmente las prácticas relacionadas con la gestión de procesos, la gestión de requisitos, el manejo de la configuración de *software* y prácticas tales como la verificación, la validación y la medición. Asimismo resaltan que es un modelo certificable con dos representaciones de madurez equivalentes, la continua y la escalonada, y que cada organización puede adoptar la que se adapte mejor a sus características y prioridades de mejora (Oktaba, Piattini, Pino y Orozco, 2009). Al respecto, Estrada (2014) menciona que el CMMI resulta ser adoptado por las grandes empresas de *software*, siendo un modelo de calidad que clasifica a las empresas en niveles de madurez de los procesos que se ejecutan para producir *software*. Sin embargo, no representa un modelo hecho para pequeñas organizaciones, aunque, el modelo CMMI ha servido de base para varias propuestas de modelos flexibles aplicables en pymes.

Por otro lado, tanto Navarro y Garzás (2010) como Oktaba, Piattini, Pino y Orozco (2009) refieren que la industria del *software* en América Latina posee una alta concentración en pequeñas y medianas empresas. Por ello, con el objetivo de mejorar sus procesos, se desarrolla una iniciativa a través de la construcción del modelo Moprosoft, pensando en la estandarización de la gran mayoría de empresas de *software* mexicanas. Entre sus

características, resalta Oktaba (2003), se trata de un modelo definido como un conjunto de procesos de naturaleza práctica además de ser fácil de entender y aplicar, sobre todo en organizaciones pequeñas. Su estructura se divide en tres categorías: la primera, de alta dirección, con el proceso de gestión del negocio que define y hace cumplir el plan estratégico de la organización; la segunda se refiere a la categoría de gerencia que comprende la gestión de procesos, la gestión de proyectos y la gestión de recursos; integra tres subprocesos, el de recursos humanos y ambiente de trabajo, el de bienes, servicios e infraestructura, y el de conocimiento de la organización. Por último se presenta la categoría de operación, que comprende dos procesos que constituyen el centro productivo del *software*, el de la administración de proyectos específicos, como el de desarrollo y mantenimiento de *software*. Además, Cánepa (2010) resalta que Moprosoft posee un marco de referencia de certificación y un mecanismo de evaluación, el cual indica un estado real de una organización durante un periodo de vigencia específico.

En otro contexto, Fernández y Piattini (2012) describen el proyecto Competisoft como una iniciativa iberoamericana, financiada por el Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED), que integra diferentes propuestas relacionadas con la mejora de procesos para microempresas y pymes desarrolladoras de *software*. Su objetivo general fue incrementar el nivel de competitividad de las pymes iberoamericanas productoras de *software* mediante la creación y difusión de un marco metodológico común, ajustado a su realidad socioeconómica y orientado a la mejora continua de sus procesos.

Finalmente, Amable-Ciudad (2017) destaca que:

La industria de *software* reconoce la contribución que dan las pequeñas organizaciones en los productos y servicios que brindan. Asimismo, se menciona que las normas ISO/IEC no estaban dirigidas a las pequeñas organizaciones hasta que se desarrolló la norma ISO/IEC 29110. Como resultado de la creación de esta norma, se han planteado modelos de evaluación y mejora de procesos de *software*, paquetes de despliegue, proyectos piloto y estrategias de implementación como apoyo a la norma, considerando herramientas de documentación y gestión del conocimiento para apoyar la adopción de la norma. (p. 62)

Por ello, en nuestro país se ha creado la norma técnica peruana NTPRTISO/IEC TR 29110512, basada en la guía de la norma internacional, que según el Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual (Indecopi, 2012), representa un reporte técnico de gestión e ingeniería para el modelo de referencia del perfil básico de la pequeña organización (PO), considerando PO a una empresa, organización, departamento o proyecto de hasta 25 personas, dedicada al desarrollo de *software*. Se considera que este reporte técnico puede ser usado por la PO para establecer procesos a ser implementados con cualquier enfoque o metodología de desarrollo basado en las necesidades de la PO o del proyecto, y comprende dos procesos

del nivel operativo: el de la gestión del proyecto y el de la implementación del *software*. Además, Coque-Villegas, Jurado-Vite, Avendaño-Sudario y Pizarro (2017) destacan que la ISO/IEC 29110 fue especialmente elaborada para pequeñas organizaciones de hasta 25 personas, siendo además certificable, y demostrando que se adoptan las prácticas de la norma a través de proyectos de mejora de procesos.

### 2.3 Modelo de calidad de producto

En primer lugar, Cataldi, Lage, Pessacq y García (1999) destacan que el modelo de McCall se basa en el concepto de calidad visto bajo la perspectiva del usuario respecto a las características de operación, a la capacidad para soportar cambios y a la adaptabilidad a nuevos entornos, considerando una serie de factores de calidad, tales como: corrección, confiabilidad, eficiencia, seguridad, usabilidad, flexibilidad, facilidad de prueba, facilidad de mantenimiento, portabilidad, reusabilidad e interoperabilidad. Asimismo, cada factor se descompone en criterios o propiedades internas del *software* que determinan su calidad, los cuales pueden ser evaluados mediante un conjunto de métricas.

En otro contexto, Reyes, Ampuero y González (2015) citan el *Standish Group Report CHAOS* del 2014, en el que se indica que:

en el 43 % de los casos, los proyectos de *software* no cumplen con el cronograma, el presupuesto o las funciones requeridas. Además, el 18 % del total de los proyectos se cancelan antes de la terminación, o se entregan, pero nunca son utilizados. (p. 1)

Esto significa que se debe garantizar tanto la calidad de proceso como la calidad de producto. Por ello comparan distintos modelos de calidad de producto de *software*, estableciendo una serie de criterios, tales como estructura, propósito, elementos internos y externos, métricas de características de calidad, tipo de proyecto, tipo de modelo y calidad, identificando como las más utilizadas a las normas ISO/IEC 9126 e ISO/IEC 25010.

Asimismo, Acosta, Espinel y García (2017) destacan el modelo de la norma ISO/IEC 9126 (*software engineering-product quality*), que tiene como finalidad cuantificar los productos de *software* basándose en las características de calidad. Al respecto, dicho modelo abarca la calidad del proceso, la calidad del producto, la calidad del *software* y la calidad de uso distribuido en los modelos de calidad interna, calidad externa y calidad en uso. Otro modelo identificado corresponde a la norma ISO/IEC 14598 (*software product evaluation*) que muestra una metodología para evaluar el producto de *software*.

En relación a la serie de normas ISO 25000, se genera como una evolución de las normas ISO/IEC 9126 y la ISO/IEC 14598, y comprende una serie de tópicos relacionados con la gestión de la calidad, el modelo de calidad, las medidas de calidad, los requerimientos de calidad y la evaluación de la calidad. Por lo tanto, la implementación de esta norma alineada a los objetivos del *software* permitirá lograr la calidad del producto

*software*, una mayor rentabilidad y la eliminación de las ineficiencias (Estayno, Dapozo, Cuenca y Greiner, 2009).

## 2.4 Métodos de desarrollo

Teniendo como punto de partida que el proceso de *software* es “una estructura de actividades, acciones y tareas que se requieren a fin de construir un *software* de calidad” (Pressman, 2010, p. 26), es importante analizar las más representativas.

En primer término, el Instituto de Normas Técnicas de Costa Rica (INTECO, 2009) refiere que una metodología es “un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo” (p. 39). Además considera que otra característica importante de una metodología es que en ella se definen artefactos, roles y actividades, junto con las prácticas y las técnicas recomendadas.

Cada metodología de desarrollo de *software* propone una visión de cómo establecer una serie de pasos predecibles para la gestión de un proyecto de *software* y también define un conjunto de actividades, roles, controles y productos de trabajo destinados a la reducción de la posibilidad del caos y a la minimización del riesgo de fracaso del proyecto. Por ello las metodologías han evolucionado a la par del desarrollo de la disciplina de la ingeniería de *software* (Pressman, 2010).

En relación a la metodología *Rational Unified Process* (RUP), Despa (2014) destaca que se desarrolla el *software* a través de un desarrollo iterativo, priorizando el manejo de riesgos, trabajando con un modelado adecuado, utilizando el lenguaje de notación UML (*Unified Modeling Language*), lo que implica desarrollar un conjunto de artefactos en diagramas como el de casos de uso, entre otros, gestionando el cambio y realizando las pruebas de rendimiento. Además se indica que RUP genera una documentación precisa y exhaustiva y que permite la reutilización de código como uno de los componentes del *software*. Sin embargo, el método RUP requiere de profesionales entrenados.

INTECO (2009) expresa que RUP resulta de la combinación de varias metodologías y se ve influenciado por métodos previos como el modelo en espiral. Asimismo, surge como respuesta a las fallas de proyectos que utilizaban métodos del modelo en cascada. Pressman (2010) resalta que RUP reconoce la importancia de la comunicación con el cliente y los métodos directos para describir su punto de vista respecto de un sistema. Adicionalmente hay una búsqueda por la minimización de los riesgos del proyecto, repitiendo una serie de ciclos o iteraciones que constituye la vida de un sistema, cada uno de los cuales produce una nueva versión del mismo, considerando que las iteraciones deben desarrollarse en las fases de inicio, elaboración, construcción y transición.

En relación a los métodos ágiles, se considera como base a la Alianza Ágil, que surge en el 2001 a partir de una reunión de destacados desarrolladores, consultores y escritores que firmaron el llamado *Manifiesto ágil*, que define 12 principios que caracterizan a este movimiento (Agile Alliance, 2001). Entre ellos resalta que la mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de *software* con valor, que se acepte que los requisitos cambien, que se entregue *software* funcional frecuentemente, que los responsables del negocio y los desarrolladores trabajen de manera coordinada continuamente durante todo el proyecto, que los proyectos se desarrollen en torno a individuos motivados, que el método más eficiente y efectivo de comunicación resulte en la conversación directa entre miembros, que el *software* funcionando resulte ser la medida principal de progreso, entre otros.

Álvarez García, De las Heras del Dedo y Lasa Gómez (2012) destacan que el conjunto de métodos ágiles para el desarrollo de *software* posee como principios fundamentales compartir un especial valor por la colaboración del equipo, por la construcción incremental y por la sencillez de sus reglas. El proceso de desarrollo de *software* resulta ser adaptativo en cada una de sus etapas.

Por ejemplo, Yepes, Pardo y Gómez (2015) realizan la siguiente observación como resultado de una investigación de estudios de casos de adopción de metodologías ágiles en mipymes (micro, pequeñas y medianas empresas) desarrolladoras de *software*:

De los 29 estudios, 11 siguieron el método SCRUM y 9 el método XP, asimismo, algunos estudios han utilizado enfoques tradicionales como el modelo en cascada y espiral. Por otro lado, el 34.5% de los estudios analizados proponen nuevas propuestas que permiten integrar metodologías tradicionales, estándares y modelos de facto con enfoques ágiles, por ejemplo, CMMI e ISO/IEC 29110. Asimismo, se puede observar una tendencia de los trabajos realizados con relación al uso de modelos como SCRUM y XP con el 23.9% y 19.6% respectivamente, en este sentido, de acuerdo al análisis, éstos son los modelos ágiles más populares y de mayor implementación en la industria de *software* hasta el momento. (p. 470)

Asimismo, identifican intereses por integrar prácticas ágiles con modelos tradicionales, convencionales, de facto y estándares internacionales, siendo los métodos ágiles SCRUM y XP los más utilizados, ya que han sido ampliamente implementados e integrados por mipymes y grandes empresas junto a otros modelos, como el CMMI, la norma ISO/IEC 29110, el modelo en espiral y el modelo en cascada, entre otros.

En relación a los métodos ágiles, Lean y TDD (Test Driven Development), Despa (2014) los explica. En el primero se realiza un desarrollo iterativo descartando todos los componentes que no agregan valor al producto; está orientado hacia el cliente, propiciando la mejora continua con el objetivo de incrementar el aprendizaje y empoderar al equipo que desarrolla el producto *software*. En relación a TDD, destaca que se focaliza en la etapa de pruebas, empezando con la construcción de los escenarios de las pruebas

antes de iniciar la codificación del producto *software* y luego refactorizar (reestructurar) de forma continua el código construido.

## 2.5 La micro y pequeña empresa (mype)

Según la Ley de Promoción y Formalización de la Micro y Pequeña empresa (Ley 28015), se define a la mype como la unidad económica que puede estar constituida por una persona natural o jurídica, mediante cualquier forma de organización o gestión empresarial, cuyo objetivo es trabajar en actividades de extracción, transformación, producción, comercialización de bienes o prestaciones de servicios. Además, el artículo 11.o de la ley vigente (Ley 30056), reemplaza al texto del artículo 5.º de la Ley 28015, eliminando lo correspondiente al número de trabajadores, quedando así solo la limitación para el monto máximo de ventas anuales. Al respecto, existe legislación vigente cuyo objetivo es facilitar la inversión e impulsar el desarrollo productivo y el crecimiento empresarial introduciendo importantes modificaciones en el régimen laboral especial de las micro y pequeñas empresas.

Por otro lado, Amable-Ciudad (2015) explica que las mypes poseen fortalezas debido a que son "incubadoras e instrumentos de aprendizaje y capacitación para el capital humano, demostrado en el conocimiento y experiencia de los integrantes de este tipo de organizaciones en el desarrollo de *software*" (p. 81). Además, considera como otra fortaleza en este sector:

El emprendimiento empresarial que genera creatividad y por tanto innovación, así como factores para la sostenibilidad de la empresa, la cual es también una cualidad de las mypes productoras de *software*, debido a que varios de los emprendedores nacen en las universidades con los conocimientos que adquieren en las aulas e iniciativas que surgen a través de los trabajos que realizan. (pp. 81-82)

## 3. METODOLOGÍA

Se identifica la caracterización de las mypes productoras de *software* a través del nivel de calidad del proceso y del producto *software*. Para ello se considera que la población sea conformada por todas aquellas empresas micro o pequeñas productoras de *software* ubicadas en Lima Metropolitana, lo que resulta en un total de 329 organizaciones. En relación a la definición del tamaño de la muestra, se considera el valor del 90 % como nivel de confianza, además del valor de la probabilidad de que las mypes aplican factores de calidad en el desarrollo del producto de *software*, el cual resulta con el valor de 0,6, en base a una encuesta piloto con un error de estimación del 8 %, así como de un 2 % de no respuesta. Como consecuencia se aplica una encuesta a los representantes de una muestra de 80 mypes con un muestreo estratificado de acuerdo al número de trabajadores (Amable-Ciudad, 2015).

### 3.1 Instrumento de medición

Se diseñó un cuestionario dirigido a los empresarios o gerentes generales, siendo la mayoría de las preguntas de tipo cerrado. El cuestionario consta de 11 módulos (tabla 1).

Tabla 1  
*Módulos de la encuesta*

| Módulo | Contenido                                |
|--------|--|
| I      | Datos de la empresa                      |
| II     | Personal                                 |
| III    | Proyectos anuales                        |
| IV     | Productos que comercializa               |
| V      | Clientes                                 |
| VI     | Calidad del proceso                      |
| VII    | Calidad del producto                     |
| VIII   | Gestión de proyectos                     |
| IX     | Proceso de desarrollo de <i>software</i> |
| X      | Roles                                    |
| XI     | Procesos del ciclo de vida               |

Elaboración propia

En el módulo VI se consideran preguntas cerradas sobre calidad del proceso de *software* respecto al uso de modelos de calidad de procesos, con la opción de indicar el modelo en caso de ser utilizado alguno.

El módulo IX comprende preguntas cerradas sobre procesos de desarrollo de *software*, específicamente sobre metodologías y artefactos que utilizan las mypes.

El módulo XI, cuyas preguntas se presentan en la tabla 2, trata sobre procesos del ciclo de vida del *software*, consultando sobre los métodos de desarrollo y técnicas empleadas en el ciclo de vida del *software*, permitiendo así relacionarlas con las etapas del ciclo de vida.

**Tabla 2**  
**Cuestionario, parte XI**  
**XI. Procesos del ciclo de vida del software (si aplican marque con una x. Puede marcar más de uno):**  
**Señale los métodos utilizados:**

| Procesos del ciclo de vida del software        | Métodos |       |    |        |     | Otros métodos (especifique) |
|--|---------|-------|----|--------|-----|-----------------------------|
|  | RUP     | Scrum | XP | Kanban | TDD |                             |
| Creación del concepto del producto de software |         |       |    |        |     |                             |
| Especificación de requisitos                   |         |       |    |        |     |                             |
| Arquitectura y diseño                          |         |       |    |        |     |                             |
| Codificación / programación                    |         |       |    |        |     |                             |
| Pruebas e integración                          |         |       |    |        |     |                             |

**Señale las técnicas utilizadas:**

| Procesos del ciclo de vida de software      | Técnicas                             |                 |                |                          |                                       |           |                        | Otras técnicas (especifique) |                       |
|---|--------------------------------------|-----------------|----------------|--------------------------|---------------------------------------|-----------|------------------------|------------------------------|-----------------------|
|   | Matriz de trazabilidad de requisitos | Product Backlog | Sprint Backlog | Diagrama de casos de uso | Diagramas de colaboración / secuencia | Prototipo | Diagramas conceptuales |                              | Diagramas de procesos |
| Creación del concepto del producto software |                                      |                 |                |                          |                                       |           |                        |                              |                       |
| Especificación de requisitos                |                                      |                 |                |                          |                                       |           |                        |                              |                       |
| Arquitectura y diseño                       |                                      |                 |                |                          |                                       |           |                        |                              |                       |
| Codificación / programación                 |                                      |                 |                |                          |                                       |           |                        |                              |                       |
| Pruebas e integración                       |                                      |                 |                |                          |                                       |           |                        |                              |                       |

Fuente: Amable, Millones y Checa, 2014

#### 4. RESULTADOS

En la figura 1 se presenta una gráfica en la que se observa el que el 13,8 % de las empresas no usan modelos de procesos de *software*.

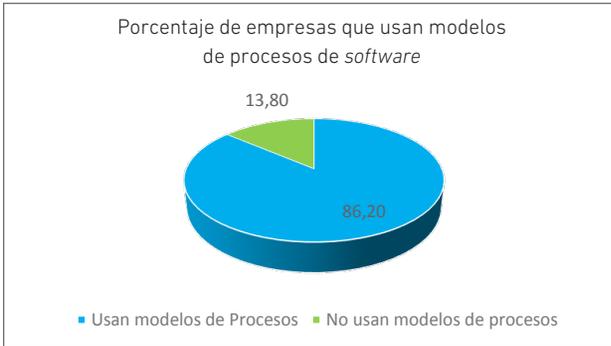


Figura 1. Uso de modelos de procesos de *software*

Elaboración propia

Se observa en la figura 2 que el 53,3 % usa algún modelo que conduce a una certificación CMMI, Moprosoft, Competisoft, ISO/IEC29110 e ISO/IEC 12207, mientras que el 46,7 % de las organizaciones usan otras prácticas, tanto propias como pertenecientes a otros modelos no referidos en el cuestionario.

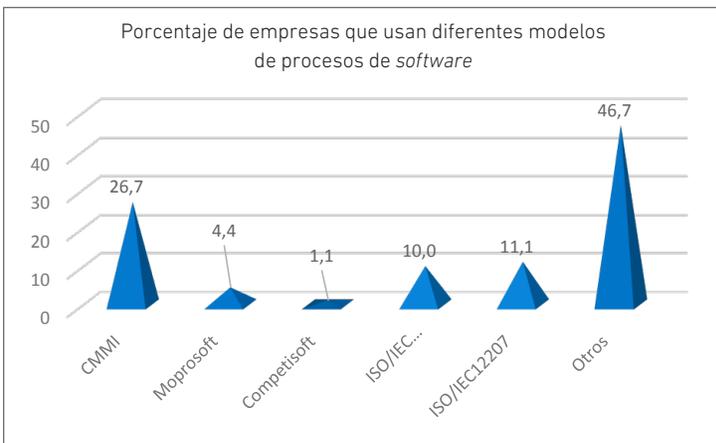


Figura 2. Uso de diferentes modelos de procesos de *software*

Elaboración propia

En relación al uso de metodologías o algunas prácticas, se muestra en la figura 3 que los representantes de las mypes encuestadas manifiestan usarlas en el desarrollo en un 92,5 %.



Figura 3. Uso de diferentes modelos de procesos de software

Elaboración propia

En la figura 4 se observa que el 26,5 % usa RUP, y las metodologías ágiles (conformadas por Scrum, Programación Extrema, Scrum / XP, TDD, Lean, Kanban y Scrumban) representan el 63,3 %.

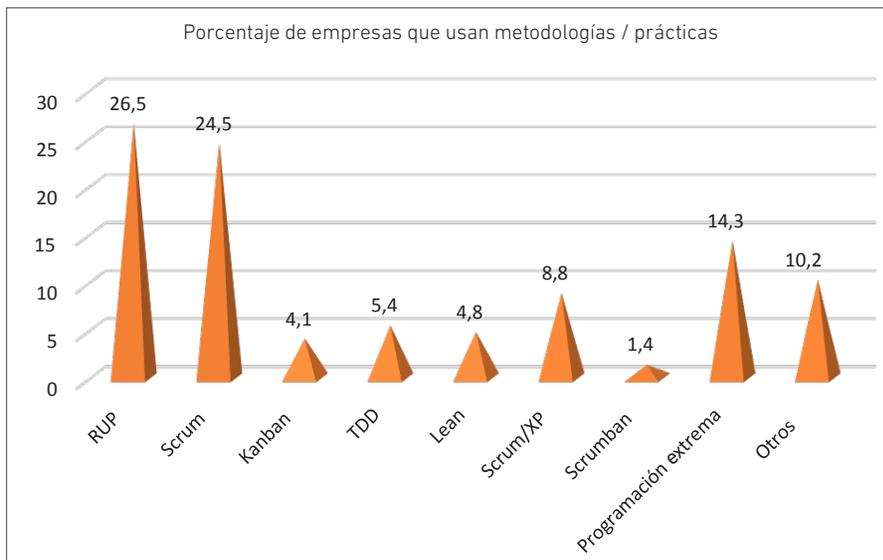


Figura 4. Uso de diferentes modelos de procesos de software

Elaboración propia

En lo que respecta al uso de modelos de calidad de producto, el 61,30 % manifestó usarlos, mientras que el 38,7 % no los usan (figura 5).

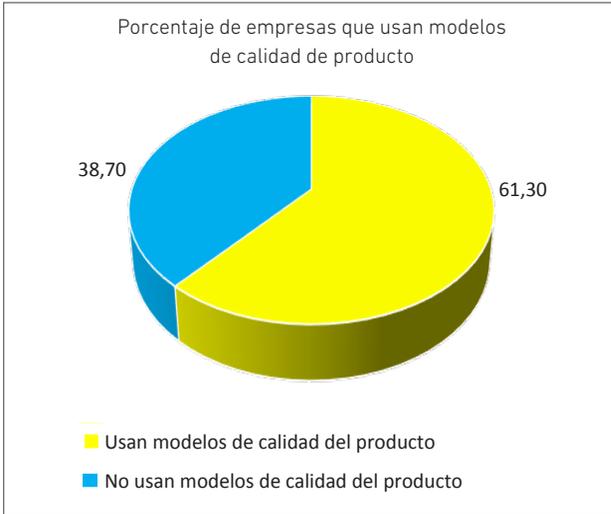


Figura 5. Uso de modelos de calidad de producto

Elaboración propia

En la figura 6 se muestra que solo el 39,3 % considera dentro de sus actividades el uso de los modelos de calidad de producto de McCall, la norma ISO 9126 y la serie ISO 25000. Un 60,7 % manifiesta utilizar otras prácticas.

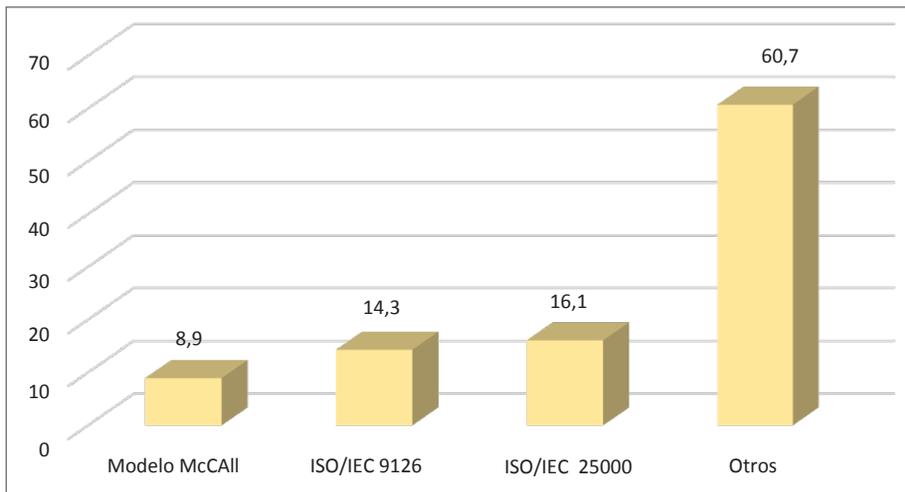


Figura 6. Uso de diferentes modelos de calidad de producto

Elaboración propia

Asimismo, los representantes de las empresas desarrolladoras de *software* manifiestan que usan factores de calidad en sus actividades en un 97,5 % (figura 7).

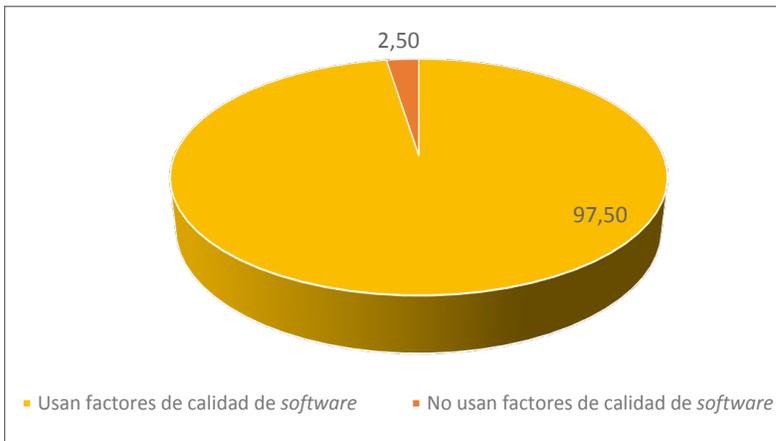


Figura 7. Uso de factores de calidad

Elaboración propia

En la figura 8 se muestra que las empresas utilizan en mayor porcentaje el factor de funcionalidad (16,7 %), seguido por la usabilidad (14,9 %), la seguridad (14,2 %), entre otros factores.

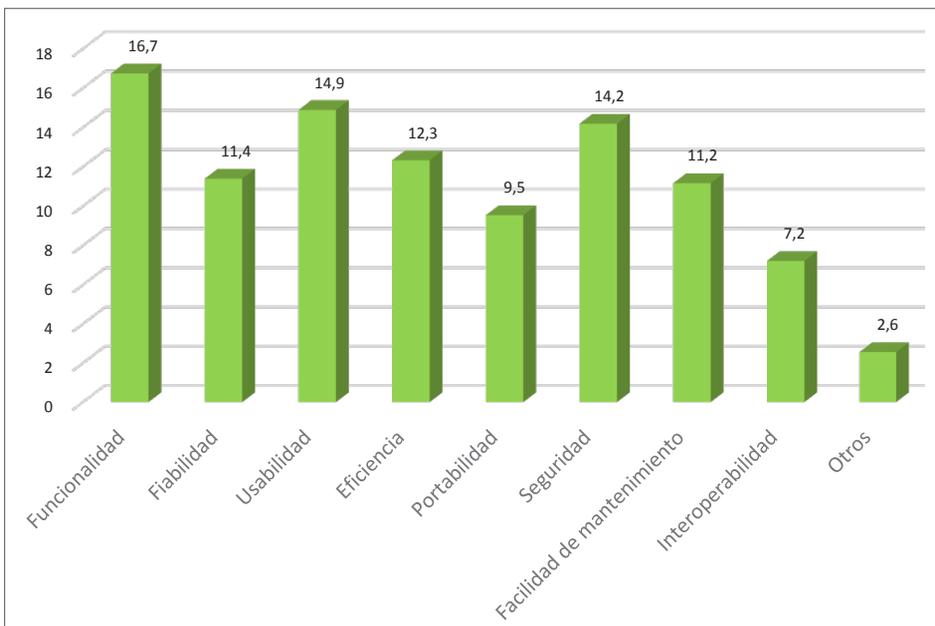


Figura 8. Uso de factores de calidad de *software*

Elaboración propia

En la figura 9 se muestra el diagrama de simetría obtenido con el *software* Minitab entre los procesos del ciclo de vida del *software* y métodos de desarrollo de *software*, primera sección del módulo XI del cuestionario, del cual se interpreta lo siguiente:

- Las etapas del ciclo de vida de creación del concepto del producto de *software*, así como la arquitectura y el diseño, están asociadas a la metodología de desarrollo RUP.
- La etapa del ciclo de vida especificación de requisitos está asociada al método ágil Scrum.
- Los procesos de ciclo de vida codificación / programación y pruebas e integración están asociados al método XP.

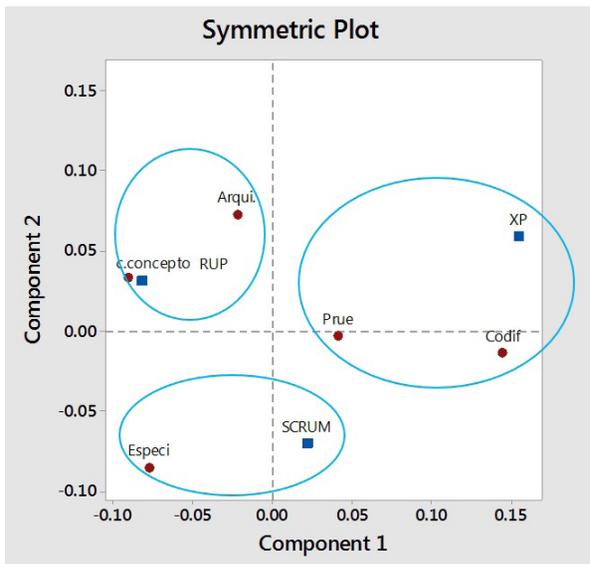


Figura 9. Procesos de ciclo de vida y métodos de desarrollo de *software*

Fuente: Amable, Millones y Checa, 2014

En la figura 10 se muestra un diagrama de simetría obtenido con el *software* Minitab entre procesos del ciclo de vida del *software* y uso de técnicas en el desarrollo de *software*, segunda sección del módulo XI del cuestionario, del cual se interpreta lo siguiente:

- El proceso de creación del concepto del producto de *software* está asociado a las técnicas diagrama de casos de uso, diagrama de colaboración/secuencia, diagrama conceptual y diagrama de procesos.
- El proceso especificación de requisitos está asociado con las técnicas matriz de trazabilidad, Product Backlog y Sprint Backlog.

- El proceso arquitectura y diseño está asociado con la técnica del prototipo.
- El proceso pruebas e integración está asociado a la técnica verificación y validación de *software*.

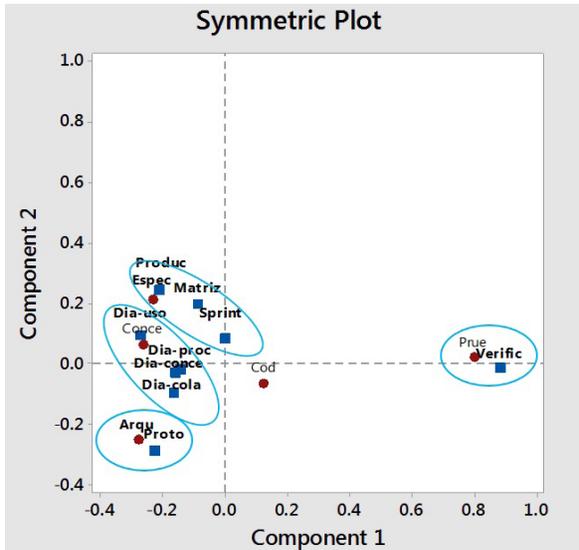


Figura 10. Procesos de ciclo de vida y técnicas de desarrollo de *software*

Fuente: Amable-Ciudad, 2015

## 5. CONCLUSIONES

En relación al uso de modelos de procesos se evidencia que un 53,3 % de las mypes encuestadas adopta algún modelo certificable como CMMI, Moprosoft, Competisoft, ISO/IEC29110 o ISO/IEC 12207. Sin embargo, solo el 15,5 % usa modelos relacionados a pequeñas organizaciones, representado por los modelos Moprosoft y Competisoft, así como la norma ISO/IEC 29110, debido a que, como dice la literatura, el modelo CMMI no fue creado para dichas organizaciones.

En relación al uso de métodos en el desarrollo de *software* se evidencia una mayor utilización de las metodologías ágiles, debido a que en conjunto suman el 63,3 %, siendo los más seguidos el método Scrum y el método XP (*eXtreme Programming*). Por otro, lado es representativo el uso de RUP en el desarrollo de *software*.

En relación al uso de los modelos de calidad de producto (el de McCall, la norma ISO 9126 y la serie ISO 25000), se evidencia un bajo uso de los mismos, con un 39,3 %. Sin embargo, al preguntar sobre la utilización de los factores de calidad, que son un

componente de los modelos de calidad de producto, los representantes de las mypes encuestadas manifestaron utilizarlos, lo cual significa que hay un conocimiento y una práctica sobre los factores de calidad del producto *software*, más no el uso de algún modelo de calidad de producto relacionado.

A través de los diagramas de simetría obtenidos entre las etapas del proceso del ciclo de vida, tanto con los métodos de desarrollo como con las técnicas, se evidencia la relación correcta entre ellos. En primer lugar se encuentra la relación entre la etapa del ciclo de vida, creación del concepto del producto *software* con el método RUP, y por el lado de las técnicas se vincula la misma etapa del ciclo de vida con los diagramas de casos de uso, de colaboración/secuencia, diagrama conceptual y de procesos, los cuales corresponden al método RUP. En segundo lugar se relaciona la etapa del ciclo de vida especificación de requisitos con el método ágil Scrum y con las técnicas matriz de trazabilidad, Product Backlog y Sprint Backlog, las cuales corresponden al método Scrum. Lo mismo se evidencia en las otras etapas.

En base a lo identificado se recomienda realizar investigaciones que apoyen a este tipo de organizaciones en la adopción, tanto de modelos de calidad de procesos adecuados para su tamaño, como de modelos de calidad de productos, y relacionarlos con las metodologías de desarrollo de *software* que utilicen, teniendo en cuenta las características del producto *software* que requiera construir.

## REFERENCIAS

- Acosta, N. J., Espinel, L. A. y García, J. L. (2017). Estándares para la calidad de *software*. TIA 5(1), pp. 75-84.
- Agile Alliance. (2001). *Agile Manifesto*. Recuperado de [www.agilealliance.com](http://www.agilealliance.com).
- Álvarez García, A., De las Heras del Dedo, R. y Lasa Gómez, C. (2012). *Métodos ágiles y Scrum*. Madrid: Anaya.
- Amable, C. M., Millones, R. R., y Checa, F. R. (2014). *Análisis del uso de modelos de calidad de software. Una propuesta de mejora de procesos en las mypes productoras de software de Lima*. Lima: Universidad de Lima.
- Amable, C. M., Millones, R. R. y Checa, F. R. (2015). Calidad de *software* en las Mypes productoras de *software* en Lima. En: *VII Congreso Internacional de Computación y Telecomunicaciones. Memoria COMTEL 2015*. Lima: Universidad Inca Garcilaso de la Vega, Fondo Editorial.
- Amable-Ciudad, M. (2015). Propuesta de un proceso de investigación cuantitativa. Aplicación en la caracterización de las mypes productoras de *software*. *Interfases 8*, pp. 71-92. Recuperado de <http://revistas.ulima.edu.pe/index.php/Interfases/article/view/574>

- Amable-Ciudad, M. (2017). Propuesta de modelo de mejora para mypes productoras de *software*. *Interfases 10*, pp. 57-73. doi:<http://dx.doi.org/10.26439/interfases2017.n10.1769>
- Cataldi, Z., Lage, F., Pessacq, R. y García Martínez, R. (1999). Revisión de marcos teóricos educativos para el diseño y uso de programas didácticos. En: *Proceedings del V Congreso Internacional de Ingeniería Informática*. Buenos Aires.
- Coque-Villegas, S., Jurado-Vite, V., Avendaño-Sudario, A. y Pizarro, G. (2017). Análisis de experiencias de mejora de procesos de desarrollo de *software* en PYMEs. *Revista Ciencia UNEMI 10(25)*, pp. 13-24.
- Despa, M. L. (2014). Comparative study on *software* development methodologies. *Database Systems Journal 5(3)*, pp. 37-56.
- Estrada, A. C. (2014). Modelo de calidad de *software*. *Innovación, Ingeniería y Desarrollo 1(1)*, pp. 47-52.
- Estayno, M. G., Dapozo, G. N., Cuenca Pletsch, L. R. y Greiner, C. L. (2009). Modelos y métricas para evaluar calidad de *software*. En: *XI Workshop de Investigadores en Ciencias de la Computación*. San Juan, Argentina.
- Fernández Sánchez, C. M. y Piattini Velthuis, M. (2012). *Modelo para el gobierno de las TIC basado en las normas ISO*. Madrid: Asociación Española de Normalización y Certificación.
- Instituto de Normas Técnicas de Costa Rica (Inteco) (2009). *Ingeniería del software: metodologías y ciclos de vida*. San José: Laboratorio Nacional de Calidad del Software de Inteco.
- Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual (Indecopi) (2012). *Ingeniería de software. Perfiles del ciclo de vida para las pequeñas organizaciones. Parte 5-1-2: Guía de gestión e ingeniería grupo de perfil genérico. Perfil básico (NTP-RT\*ISO/TEC TR2\*110-5-1-2)* Lima: Indecopi.
- International Organization for Standardization, International Electrotechnical Commission y Institute of Electrical and Electronics Engineers (2008). *Systems and software engineering. Software life cycle processes. ISO/IEC 12007:2008*. Recuperado de <https://www.iso.org/home.html>
- Ley 28015 (3 de julio del 2003). Ley de Promoción y Formalización de la Micro y Pequeña empresa. Recuperado de <http://www.leyes.congreso.gob.pe/Documentos/Leyes/28015.pdf>
- Loya Páez, A. (2018). Industria de *software* en Perú. *Software Gurú*. Recuperado de <https://sg.com.mx/revista/53/industria-software>

- Miguel, J. P., Mauricio, D. y Rodríguez, G. (2014). A review of *software* quality models for the evaluation of *software* products. *International Journal of Software Engineering & Applications* 5(6), pp. 31-53.
- Navarro, J., y Garzás, J. (2010). Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y *software* libre. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software* 6(1), pp. 6-15.
- Oktaba, H. C. (2003). *Modelo de procesos para la industria del software versión 1.3*. México: Secretaría de Economía.
- Oktaba, H., Piattini, M., Pino, F. J. y Orozco, M. J. (2009). *Competisoft: Mejora de procesos software para pequeñas y medianas empresas y proyectos*. México: Alfaomega.
- Pressman, R. S., Campos, O. V. y Enríquez, B. J. (2010). *Ingeniería del software: Un enfoque práctico*. México: McGraw-Hill.
- Reyes, A. G., Ampuero, M. A. y González, A. H. (2015). Análisis comparativo de modelos y estándares para evaluar la calidad del producto de *software*. *Revista Cubana de Ingeniería* 6(3), pp. 43-52.
- Sánchez, A. S., Sicilia, U. M. A. y Rodríguez, G. D. (2012). *Ingeniería del software: Un enfoque desde la guía SWEBOK*. México: Alfaomega.
- Yepes González, J., Pardo Calvache, C. y Gómez Gómez, O. (2015). Revisión sistemática acerca de la implementación de metodologías ágiles y otros modelos en micro, pequeñas y medianas empresas de *software*. *Revista Tecnológica ESPOL* 28(5), pp. 464-478.

