

Propuesta de Métricas para la implementación del estándar ISO/IEC 29110

Jezreel Mejía

jmejia@cimat.mx

Centro de Investigación en Matemáticas- Unidad Zacatecas, Parque Quantum, Ciudad el Conocimiento
Avenida Lassec, Andador Galileo Galilei, Manzana, 3 Lote 7, 98160. Zacatecas, México.

DOI: 10.17013/risti.45.24-47

Resumen: El estándar ISO/IEC 29110 tiene como objetivo proveer y contribuir una serie de diferentes actividades, tareas y productos de trabajo que buscan mejorar la calidad de los diferentes productos de software desarrollado por Entidades Muy Pequeñas (EMPs). Sin embargo, la calidad no debe ser solo cualitativa sino cuantitativa a través del uso de métricas y aún en muchas EMPs existe el desconocimiento de la correcta aplicación de las diferentes métricas que pueden ser implementadas en este estándar. Este artículo presenta una propuesta de Métricas que apoyen a las EMPs en la implementación de este estándar y tener una visión cuantitativa en la calidad a nivel proceso, producto y proyecto. Para identificar las métricas viables, se analizaron dos proyectos presentados por dos EMPs para obtener su certificación, identificando el conjunto de métricas a través de la documentación presentada por las EMPs.

Palabras-clave: ISO/IEC 29110, entidades muy pequeñas, métricas de producto, proceso y proyecto.

Metrics proposal for the implementation of the ISO/IEC 29110 standard

Abstract: The ISO/IEC 29110 standard aims to provide and contribute to a series of different activities, tasks and work products that seek to improve the quality of the different software products developed by Very Small Entities (VSEs). However, quality should not only be qualitative but quantitative through the use of metrics and even in many VSEs there is a lack of knowledge of the correct application of the different metrics that can be implemented in this standard. This article presents a metrics proposal that support VSEs in the implementation of this standard and have a quantitative vision of quality at the product, process and project levels. To identify viable metrics, two projects submitted by two VSEs to obtain their certification were analyzed, identifying the set of metrics through the documentation submitted by these.

Keywords: ISO/IEC 29110, Very Small Entities, Product, Process and Project Metrics.

1. Introducción

De acuerdo a los resultados obtenidos de INEGI en 2018, existen más de 4 millones de empresas en México, de las cuales el 97.3% corresponden a microempresas (Mipymes), y 2.7% a PYMES (INEGI, 2018). Debido a su importancia en México el 76% de estas empresas fueron apoyadas por parte de PROSOFT, lo que indica su importancia en la industria del software, en este país (Laporte, et al 2017). Además, es importante resaltar que tienen una gran presencia en la industria del software en algunos países, donde alrededor del 94% está formado por este tipo de empresas (Mejia, et al 2020) (Galván-Cruz, et al 2021).

A nivel mundial, se encuentran asociada al desarrollo de software alrededor del 92% de las empresas (Mejia, et al 2021). Por lo tanto, este tipo de empresas están cada vez más interesadas en implementar estándares o modelos orientado a procesos que permitan guiar el desarrollo de sus productos y servicios de software adecuados a su tamaño y características. En respuesta a esta necesidad para las Entidades Muy Pequeñas (EMPs) o Micro, pequeña y medianas empresas (Mipyme), se crea el estándar ISO/IEC 29110 (ISO/IEC, 2011). El objetivo de este estándar es mejorar la calidad del producto a través del uso de dos procesos: el proceso de Gestión de Proyectos (GP) y el proceso de Implementación del Software (IS).

Sin embargo, alrededor del 15% de las EMPs han indicado que los estándares son complejos y no proporcionan una guía para su uso en un entorno pequeño (Laporte & Mejia, 2020). Este contexto, puede desencadenar problemas como productos de una calidad baja, por lo que deben evaluarse y mejorarse continuamente los procesos implementados (Choras, et al 2020). Para lograrlo, se requiere hacer uso de diferentes métricas de software, necesarias para evaluar tanto el proyecto, proceso y producto en este tipo de empresas y durante la implementación del estándar ISO/IEC 29110. Por lo tanto, la medición permite que la mejora continua no sea solo cualitativa sino cuantitativa.

Este artículo tiene como objetivo, proponer un conjunto de métricas viables identificadas a través de un estudio de caso en dos EMPs que han permitido identificar métricas que pueden ser implementadas bajo el estándar ISO/IEC 29110.

Después de la introducción, este artículo se estructura como sigue: la sección 2 describe de manera general el estándar ISO/IEC 29110; la sección 3 presenta la clasificación de métricas que fueron seleccionadas; la sección 4 muestra el estudio de caso que sirvió para identificar y proponer el conjunto de métricas seleccionadas; y finalmente la sección 5 presenta las conclusiones y trabajo futuro.

2. ISO/IEC 29110

La ISO/IEC 29110 (2011) es una serie de estándares titulado “Ingeniería de Software – Perfiles de Ciclo de Vida para Pequeñas Organizaciones. Una Entidad Muy Pequeña (VSE por sus siglas en inglés – Very Small Entities) se define como una entidad (empresa, organización, departamento o proyecto) que tiene menos de 25 personas. La mayoría de las PYMES de software pertenecen a la categoría VSE [3]. Esta norma contiene dos áreas de proceso, las cuales se describen a continuación:

1. **Proceso de Gestión de Proyectos (GP):** El propósito de GP es establecer y llevar a cabo de manera sistemática las tareas del proyecto de implementación de software, lo que permite cumplir con los objetivos del proyecto respecto a la calidad, tiempo y costos esperados (ISO/IEC, 2011). Esta parte del estándar ISO/IEC 29110 está destinada a ser utilizada por las EMPs para establecer procesos para implementar cualquier enfoque o metodología de desarrollo que incluya, por ejemplo, un enfoque ágil, evolutivo, incremental, impulsado por pruebas desarrollo, etc., en función de las necesidades de la organización o el proyecto de VSE. Esta área de proceso contiene 7 objetivos (Galván-Cruz, et al 2021). El proceso de GP consta de cuatro actividades y cada actividad con un número de tareas específicas. Cada una de estas actividades con sus tareas producen los siguientes productos.

Productos de Entrada: los productos de Entrada son necesarios para realizar las tareas del proceso de GP.

- Enunciado del Trabajo.
- Configuración del Software.
- Solicitud de Cambio.

Productos Internos: los productos internos para el proceso de GP son generados y consumidos por el mismo proceso.

- Solicitud de Cambio.
- Acciones Correctivas.
- Acta de Reunión.
- Resultados de Verificación.
- Reporte de Avance.
- Respaldo del Repositorio del Proyecto.

Productos de Salida: los productos de salida son aquellos que se generan al realizar las tareas del proceso de GP:

- Plan de Proyecto.
- Acta de Aceptación.
- Repositorio del Proyecto.
- Acta de Reunión.
- Configuración de Software.

2. **Proceso de Implementación de Software (IS):** Su propósito es la realización sistemática del análisis, el diseño, actividades de construcción, integración y pruebas de productos de software nuevos o modificados de acuerdo con los requisitos (ISO/IEC, 2011). Esta parte del estándar ISO/IEC 29110 está destinada a ser utilizada por el VSE para establecer procesos para implementar cualquier enfoque o metodología de desarrollo que incluya, por ejemplo, un enfoque ágil, evolutivo, incremental, impulsado por pruebas desarrollo, etc. en función de las necesidades de la organización o el proyecto de VSE. El proceso IS, a través de la realización de cada una de sus actividades, pretende cumplir 7 objetivos al igual que el proceso de GP. Cada una de estas actividades con sus tareas producen los siguientes productos.

Productos de Entrada: los productos de Entrada son necesarios para realizar las tareas del proceso de IS.

- Enunciado del Trabajo.
- Plan del Proyecto Gestión de Proyectos.
- Repositorio del Proyecto Gestión de Proyectos.

Productos Internos: los productos internos para el proceso de IS son generados y consumidos por el mismo proceso.

- Resultados de Verificación.
- Resultados de Validación.

Productos de Salida: los productos de salida señala son aquellos que se generan al realizar las tareas del proceso de IS.

- Configuración de Software:
 - Especificación de Requisitos.
 - Diseño de Software.
 - Registro de Trazabilidad.
 - Componente de Software.
 - Software.
 - Casos de Prueba y Procedimientos de Prueba.
 - Reporte de Pruebas.
 - Manual de Operación.
 - Manual de Usuario.
 - Manual de Mantenimiento.
- Solicitud de Cambio.

La Figura 1, muestra la estructura general de los procesos GP e IS con sus actividades y total de tareas.

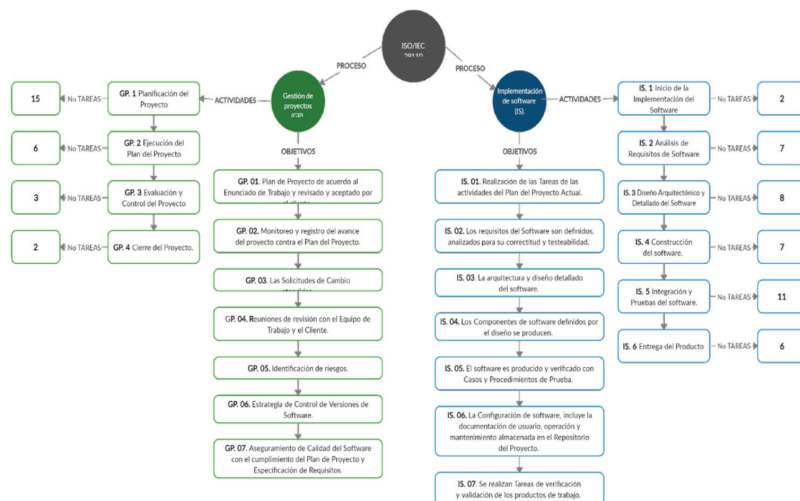


Figura 1 – Estructura de los procesos GP e IS

Como puede observarse en la Figura 1, cada área de procesos tiene 7 objetivos. El proceso GP consta de 4 actividades y la sumatoria de sus tareas es de 26, donde la actividad 1 es la que contiene el máximo de tareas asignadas, 15 en total. Con respecto al proceso IS consta de 6 actividades y la sumatoria de sus tareas es de 41, siendo la actividad 5 es la que contiene el máximo de tareas asignadas, 11 en total. Como resultado se debe de realizar la implementación de 67 tareas que producen en total 22 productos de trabajo.

3. Métricas

Para realizar la identificación de métricas se tomo como referencias las métricas de Proceso, de producto y de Proyecto (Baso, 2014; Constanzo, et al 2014; Gehlot, et al 2019; Haindl, et al 2021; Noor, et al 2020; Rashid, et al 2019; Vogel, et al 2020) Ver Anexo A:

- *Métricas del proceso:* Evalúan la vialidad y la naturaleza del proceso de programación (Rashid, et al 2019), permiten la obtención de indicadores que buscan a analizar avances en el proceso y el ambiente de desarrollo de software, utilizada para evaluar la eficiencia de un proceso, encontrándose atributos como el costo del desarrollo, esfuerzo y tiempo dedicado a pruebas, cantidad de personas por día, por mes, interrupciones, entre otros (Noor, 2020).
- *Métricas del producto:* Evalúan la calidad de los productos entregables y la estimación de los elementos del trabajo obtenido en los diferentes periodos del desarrollo de software (Rashid, et al 2019), considerando atributos como el tamaño, calidad, complejidad, esfuerzo, volatilidad, entre otros (Noor, 2020).
- *Métricas del proyecto:* Describen características propias del proyecto y su ejecución, reflejando las cualidades y la ejecución de la empresa, con el objetivo de reducir la planificación de desarrollo o minimizar defectos, para lo que consideran atributos como la duración real del proyecto, esfuerzo real por proceso, subprocesso, por proyecto, costo total invertido, tamaño del proyecto, entre otros (Noor, 2020).

4. Estudio de caso

El desarrollo del estudio de caso se realizó en dos Mipymes de desarrollo de productos y servicios de software en el estado de Zacatecas, México. Para poder llevar a cabo la propuesta y viabilidad de las métricas, se solicitó la carpeta de los proyectos presentados, para obtener la certificación en el ISO/IEC 29110 de estas empresas.

La primera EMPs fue creada en el año 2005 y es una organización privada dedicada a la venta de productos y soluciones informáticas. Tiene como objetivo proveer de soluciones tecnológicas más eficaces y de mejorar el cumplimiento de los estándares de calidad deseados. Uno de los enfoques de dicha empresa el desarrollo de software a medida, tomando en cuenta estrictos estándares de calidad. La documentación presentada se enlista a continuación con la siguiente estructura:

- 1REQ
 - 1.02_ EspecificaciónRequerimientos
 - MAT_Rastreabilidad

- 2DIS
 - 3.01_CU
 - 3.02_ModeloArquitectónico
 - 3.03_BD-DD
 - 3.04_PrototiposManualPreliminar
- 3DEV
 - UT
 - Acceso
 - Alta de Usuarios
 - Aretes
 - Médicos
 - Productores
 - Pruebas
 - UPP
- 4TST
 - IT
 - PlanPruebasIntegracion
- ADM
 - Minutas
 - 1.02_MIN_JuntaValidaciónRequerimientos20191003
 - 6.04_MIN_EntregaAlCliente20200331
 - MIN_JuntaAvanceCliente20191107
 - MIN_JuntaAvanceCliente20191205
 - MIN_ReporteAvance20191007
 - MIN_ReporteAvance20191104
 - MIN_ReporteAvance20191202
 - MIN_ReporteAvance20200106
 - MIN_ReporteAvance20200203
 - SolicitudCambio
 - Reportes
 - ReporteAvance20191007
 - ReporteAvance20191104
 - ReporteAvance20191202
 - ReporteAvance20200106
 - ReporteAvance20200203
 - 1.01_EnunciadoTrabajo
 - 1.03_PlanProyecto
 - 1.04_PAC
 - 1.05_PlanRiesgos
 - 1.06_Cronograma
 - 4.01_BitácoraVersiones
 - 4.02_AccionesCorrectivas
 - 6.05_ActaEntrega
 - 6.05_ActaEntrega.pdf
 - CriteriosVerificaciones
 - SESIAJ-EstrategiaControlVersiones

- ETC
 - Linea Base
 - LB_1REQ
 - LB_2DIS
 - LB_3DEV
 - LB_4TST
 - LB_5Final
 - Manuales
 - 6.01_Manual Usuario
 - 6.02_Manual Mantenimiento
 - 6.03_Manual Operación
- Finalconfig
 - 1.02_EspecificaciónRequerimientos
 - 6.01_Manual Usuario
 - 6.02_Manual Mantenimiento
 - 6.03_Manual Operación

La segunda Mipyme fundada en el año 2008. Tiene como objetivo ser pionera en el desarrollo de nuevas soluciones basadas en la innovación, constante perfeccionamiento, así como la total satisfacción del cliente a través de Ingeniería, Investigación y Tecnología de Clase Mundial. Las áreas de desarrollo Incluyen: Minería, Construcción, Pirotecnia, Academia e Instrumentación. Sus principales desarrollos se enfocan a: Desarrollo de Aplicaciones Mecatrónicas, Sistemas Embebidos e Internet de las Cosas; totalmente compatible con la plataforma de desarrollo de Atmel. La documentación presentada para la obtención de la certificación está relacionadas a 18 productos de trabajo que a continuación se enlistan:

- Gestión del Proyecto.
 - Acciones Correctivas.
 - Acta de Aceptación.
 - Actas de Reunión.
 - Cronograma de actividades.
 - Descripción del Proyecto.
 - Plan de adquisición y capacitación.
 - Plan del Proyecto.
 - Reportes de Avance.
 - Solicitud de Cambio.
- Implementación de Software.
 - Análisis y Diseño.
 - Casos de Prueba y Procedimientos de Prueba.
 - Especificación de Requisitos.
 - Manual de Mantenimiento y Operación.
 - Manual de Usuario.
 - Registro de Trazabilidad.
 - Reporte de Pruebas.
 - Reporte de Verificación y Validación.
 - Software.

4.1. Identificación de métricas

Para llevar a cabo la identificación de métricas que se proponen se realizó las siguientes actividades:

- Análisis de las tareas y los productos de trabajo: el análisis de las tareas y la relación que tienen entre ellas desde el proceso de GP e IS fueron analizadas.
- Análisis de documentación presentada: se tomo como referencia la documentación presentada por las dos empresas de desarrollo de software en consideración de la aplicación del estándar ISO/EIC 29110.

Tras la realización de las actividades anteriores, se detectaron 63 métricas viables agrupados de acuerdo a la Figura 2.

PRODUCTO:

Tamaño:

Líneas de Código (Medida en miles - KLDC), .

Puntos de Función (PF).

Páginas de Documentación.

Complejidad:

Complejidad Ciclomática.

Nivel de Acoplamiento de los Módulos.

Nivel de Modularidad (Cohesión de Módulos).

Calidad:

Cantidad de defectos por KLDC.

Cantidad de errores encontrados por KLDC.

Cantidad de defectos/errores que encuentran los usuarios después de la entrega.

Tipo y origen de los defectos (requerimientos, análisis y diseño, construcción, integración y pruebas).

Mantenimiento:

Cantidad de Componentes.

Volatilidad de los Componentes.

Complejidad de los componentes.

Cantidad de requerimientos nuevos, de cambios o mejoras.

Cantidad de requerimientos de corrección de defectos.

Tiempo promedio de corrección de errores ó defectos.

Tiempo promedio de cambios.

Porcentaje del código corregido.

Confiabilidad:

Tiempo transcurrido entre fallas

Tiempo esperado entre fallas

Tiempo requerido para corregir una falla

Nivel de severidad de la falla

Usabilidad:

Facilidad de aprendizaje de uso.

Errores cometidos por los usuarios con el uso.

Tiempo requerido para realizar las tareas.

Rendimiento:

Tiempos de respuesta (Acuerdos SLA).

Utilización de recursos (Troughput o Thruput) – cantidad de transacciones que pueden ejecutarse concurrentemente con un tiempo de respuesta razonable.

Tiempo de recuperación.

PROYECTO:

Esfuerzo

Cantidad de horas trabajadas.

Cantidad de personas que trabajan en el proyecto.

Tiempo transcurrido.

Distribución del esfuerzo por fase.

Costo:

Costo del Desarrollo.

Costo del Soporte.

Costo de hs/persona.

Productividad

Cantidad de software desarrollado por unidad de trabajo.

Tamaño/Esfuerzo.

Ritmo de entrega del software por unidad de tiempo transcurrido.

Seguimiento

- Cronograma real vs Cronograma estimado.
- Porcentaje de tareas completadas.
- Porcentaje de requerimientos implementados por unidad de tiempo.
- Porcentaje de tiempo total dedicado a las pruebas.
- Porcentaje de error en la estimación de tiempo.
- Costo sobre el valor agregado.

Estabilidad

- Origen de los cambios en los requerimientos.
- Cambios de los requerimientos en el desarrollo.
- Cambios en los requerimientos en producción.

PROCESO:

Esfuerzo:

- Distribución del esfuerzo por fase del proceso.
- Cantidad de personas requeridas.
- Esfuerzo requerido para corregir un defecto.
- Esfuerzo requerido para mejorar un defecto.

Reusabilidad:

- Cantidad de componentes reutilizados.
- Grado de reusabilidad de los componentes.

Calidad:

- Cantidad de defectos sin corregir.
- Costo de corrección de defectos.
- Eficacia en la eliminación de defectos.
- Cantidad de veces que un módulo fue probado.
- Tamaño de un módulo.
- Tiempo promedio en corrección de defectos.

Soporte a clientes:

- Tamaño del back log de defectos.
- Tiempo de respuesta en atender los defectos.
- Tiempo de resolución de defectos.

Herramientas:

Soporte de herramientas para procesos propuestos.

Asociado a la documentación analizada de las dos empresas, se identificó cuales elementos permiten obtener información asociada a las métricas identificadas. El Anexo A, muestra los productos de trabajo del estándar que inciden en la extracción de información para poder alimentar los datos de las métricas, las cuales se originan en el proceso de Gestión de Proyectos y reciben información a partir de datos generados en el mismo proceso o bien del proceso de Implementación de Software.

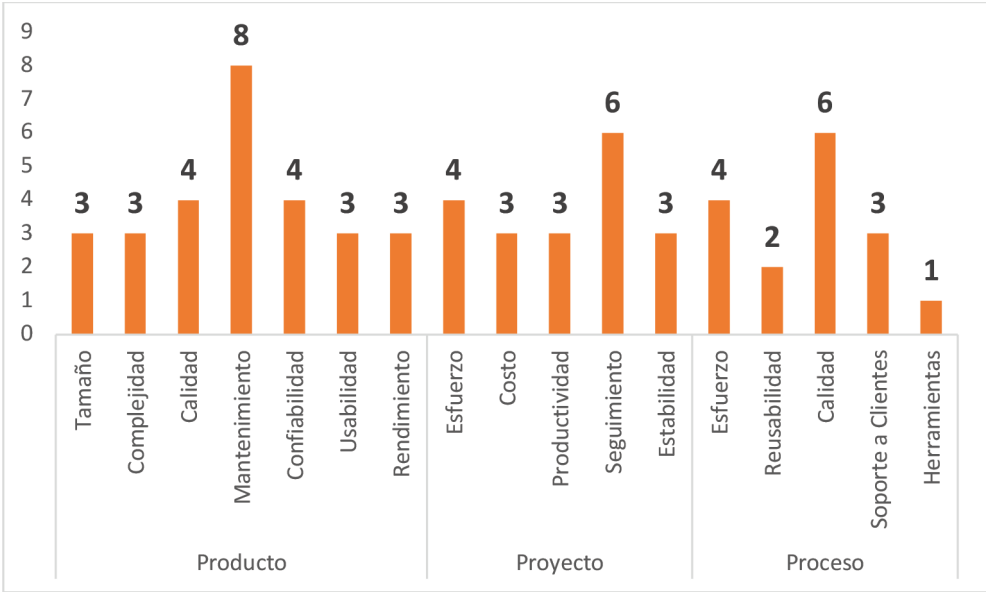


Figura 2 – Total de métricas que pertenecen a cada categoría según su atributo

Respecto a la información que se obtiene a partir del análisis de la documentación, y de acuerdo a la Figura 3, se observa que existe una gran cantidad de información que se obtiene a partir del análisis o implementación de casos y procedimientos de prueba, de donde se obtiene información para alimentar 26 de las 63 métricas.

Además, en la Figura 3 se observa que, para el caso de la documentación asociada a los componentes de software, se obtiene información para alimentar 25 de las 63 métricas y adicionalmente, mientras qué, asociado a los diferentes reportes de avance y reporte de pruebas, permite alimentar con información a 18 y 23 de las 63 métricas analizadas respectivamente, y finalmente del plan de proyecto se pueden cubrir 17 métricas.

En relación con la cantidad de tareas que cubren las diferentes métricas, es posible obtener que existen 7 tareas asociadas al proceso de Gestión de Proyectos que permiten alimentar estas métricas. Para el proceso de Implementación de Software, existen 15 tareas que permiten alimentar métricas, donde adicionalmente. Por lo tanto, se identificó

que, para el proceso de Gestión de Proyectos, la tarea que presenta una mayor frecuencia en relación a su aportación de información a las métricas corresponde a la tarea 2.1 (Monitorear la ejecución del Plan de Proyecto y registrar la información actual en el Reporte de Avance), 28 de las 63 métricas presentadas en la Anexo A.

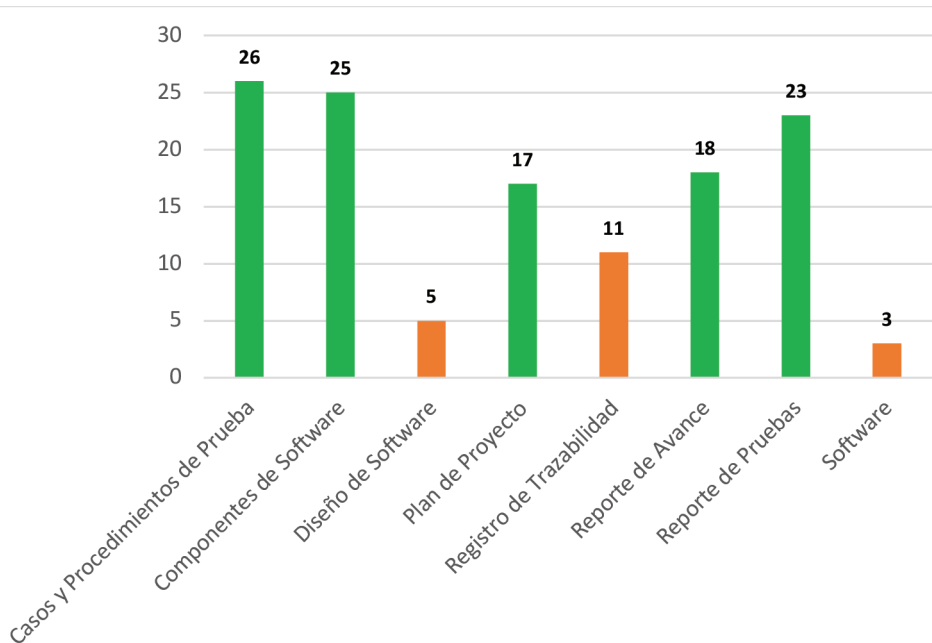


Figura 3 – Cantidad de métricas que pueden ser analizadas a partir del producto de trabajo generado

4.2. Propuesta de Mejoras para la implementación de productos de trabajo

Después del análisis de la documentación presentada por ambas EMPs y de los resultados obtenidos a través de la identificación de métricas y establecer su viabilidad que se presenta en el Anexo A, se identificaron mejoras con respecto a los productos de trabajo relacionadas a las métricas propuestas:

- a. **Tiempo estimado:** la estimación de tiempo es la actividad de la planificación del proyecto que intenta determinar cuánto dinero, esfuerzo, recursos y tiempo tomará construir un sistema o producto. De acuerdo al análisis de las actividades de GP e IS, la estimación de tiempo debe realizarse a tres niveles para ser presentado en un plan de proyecto mas cercano al tiempo que se empleará en el desarrollo del proyecto, estos tres niveles son:
 1. Tiempo estimado a nivel gestor y líder técnico.
 2. Tiempo estimado del equipo de proyecto.
 3. Tiempo estimado global del proyecto.

- b. **Tiempo real:** en el cronograma el Líder Técnico y el equipo de trabajo analizan los tiempos de acuerdo con el reporte de avance de cada miembro del equipo, y si hay algún pendiente en una tarea asignada poder tomar una decisión. Cabe resaltar que, si hay una solicitud de cambio y es aceptado, esta métrica será afectado, ya que cambiaría los tiempos de algunas tareas o en el peor de los casos el tiempo de todas las tareas. El cronograma real debe ser alimentado con los reportes de avance de cada miembro del equipo. Por lo tanto, puede existir algún ajuste en la estimación de tareas y en el tiempo real de cada tarea.

Por lo tanto, en relación con la generación de reportes de avances, esclarecer que debe existir a lo menos tres niveles de reportes de avance, siendo estos a nivel individual, para realizar el análisis del cumplimiento de las tareas por cada miembro del equipo, a nivel de equipo, siendo este el conjunto un análisis a partir de los reportes levantados por los miembros del equipo y reportes a nivel de proyectos, considerando los elementos anteriores para realizar el análisis del cumplimiento que se lleva como equipo y así realizar una comparativa con los demás proyectos en ejecución.

- c. **Costo Estimado:** la estimación de costo es la actividad de la planificación del proyecto que intenta determinar cuánto dinero, esfuerzo y recursos tomará construir un sistema o producto. De acuerdo al análisis de las actividades de GP e IS, la estimación de costos debe realizarse a tres niveles para ser presentado en un plan de proyecto mas cercano al costo que implica todo el desarrollo del proyecto, estos tres niveles son:

1. Costo estimado a nivel gestor y líder técnico.
2. Costo estimado del equipo de proyecto.
3. Costo estimado global del proyecto.

- d. **Costo Real:** en el cronograma el Líder Técnico y el equipo de trabajo analizan los costos de acuerdo con el reporte de avance de cada miembro del equipo, y si hay algún pendiente en una tarea asignada poder tomar una decisión. Cabe resaltar que si hay una solicitud de cambio y es aceptado, esta métrica será afectado, ya que cambiaría los costos de algunas tareas o en el peor de los casos el costo de todas las tareas. El cronograma real debe ser alimentado con los reportes de avance de cada miembro del equipo. Por lo tanto, puede existir algún ajuste en la estimación de tareas y en el tiempo real de cada tarea.

Con respecto a las métricas anteriores puede concluirse que es necesario que se deben desarrollar 3 planes de proyectos, uno de alto nivel (Requisitos y algunos requerimientos del proyecto), uno de bajo nivel, con el detalle de los requerimientos analizados por parte de los miembros del equipo de trabajo, para realizar la evaluación de los costos asociados para el desarrollo del proyecto y poseer información que permita alimentar el plan de proyectos de alto nivel para no incurrir en malas proyecciones respecto a lo planificado. Y finalmente, el plan de proyecto global, que es la versión que se entregará al cliente para ser validado por él.

5. Conclusiones y trabajo futuro

El estándar ISO/IEC 29110 fue creado para apoyar a las Entidades Muy Pequeñas (EMPs) en la mejora de sus procesos de desarrollo de software, debido al rol que tienen muy importante en el desarrollo de software, las cuales tienen en promedio el 92% de representación a nivel mundial. Por lo tanto, es importante que las EMPs implementen de manera adecuada los procesos de este estándar. Sin embargo, debido a la falta de experiencia previa en el uso de estándares, la falta de comprensión del mismo y el desconocimiento de métricas asociadas al producto, proyecto y proceso que faciliten la evaluación cualitativa a lo largo de los proyectos desarrollados puede dificultar la correcta evaluación de la calidad.

En este artículo se muestra un conjunto de métricas que apoyan en la implementación del estándar. El hacer uso de diferentes métricas permite mejorar la calidad desde la perspectiva de las tareas asociadas a cada una de las actividades del estándar ISO/IEC 29110, así como permitir realizar un análisis respecto de la calidad desde el punto de vista del producto, proyecto y proceso. De acuerdo al Anexo A, se describe el tipo de métrica, su descripción; así como, en que tarea y producto de trabajo del estándar puede alimentarse.

El estudio de caso que se llevo a cabo, permitió proponer un conjunto de métricas que a través de la documentación presentada para obtener una certificación en el estándar ISO/IEC 29110 se identificó su viabilidad a través de la realización de los productos de trabajo. Como resultado, se lograron identificar 63 métricas. Además, se identificó que existe una gran cantidad de información que se obtiene a partir del análisis o implementación de casos y procedimientos de prueba, de la documentación asociada a los componentes de software, de reportes de avance y reporte de pruebas, y finalmente del plan de proyecto. Por lo que, es importante resaltar la importancia de estos productos de trabajo y su incidencia en la generación de información para implementar métricas. Como resultado del análisis de la documentación de los proyectos presentados fue posible obtener información para el 48% de las 63 métricas a partir de su documentación. Lo que resalta la importancia de capacitar a las EMPs en el uso de métricas de producto, proyecto y del proceso. Finalmente, como resultado de este análisis a través de las métricas se identificaron hallazgos de mejora asociadas a los productos de trabajo o formatos entregados para mejorarlos y asociar éstos a diferentes datos que son necesarios para implementar métricas, las cuales permitirán mejorar la base de conocimientos de las EMPs en relación con los proyectos realizados.

Como trabajo futuro, es analizar la correcta implementación en los productos de trabajo de las empresas que presenten proyectos para la obtención de la certificación en el estándar ISO/IEC 29110. Además, de incluir de manera explícita el uso de métricas asociadas a los productos de trabajo indicados en el Anexo A.

Anexo A

Anexo A - Métricas existentes en ingeniería de software y análisis de las métricas y su obtención a partir de la documentación entregada.

Métricas		Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Producto	Tamaño	Líneas de Código (Medida en miles - KLDC). Esta métrica busca medir la cantidad de instrucciones ejecutables, excluyendo comentarios, líneas en blanco y toda línea no ejecutable. Loc: Líneas de código. # LE: Líneas de código ejecutables. #Cmtsblks: Número de líneas de código no ejecutables.	IS.4.3 - 5.7	Componentes de Software - Software.
	Puntos de Función (PF).	Los puntos de función se utilizan en el desarrollo de software como base para la estimación de costos, evaluación comparativa y, en general, para la derivación de métricas de productividad y calidad. Una clasificación de puntos de función es independiente de la tecnología subyacente de la aplicación.	IS.4.3 - 5.7	Componentes de Software - Software.
	Páginas de Documentación.	Este atributo indica la calidad de la documentación suministrado junto con el componente. Al ser una medida subjetiva utilizaremos una métrica con varias categorías. Se utiliza una métrica de tipo Nivel que indique la calidad de la documentación.	IS.5.10 - 5.11 - 6.3	Manual de Operación - Manual de Usuario - Manual de Mantenimiento.
Complejidad	Complejidad Ciclomática.	Esta métrica proporciona una indicación del código complejidad basada en el número de ramas en los flujos de control. #Aristas: Número de aristas. # Nodos: Número de nodos del grafo correspondientes a sentencias del programa.	IS.4.4 - 4.5 - 5.3 - 5.4 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba.
	Nivel de Acoplamiento de los Módulos.	Esta métrica calcula el número de dependencias entre los componentes de software contenidos dentro del paquetes.	IS.3.3	Diseño de Software.
	Nivel de Modularidad (Cohesión de Módulos).	Esta métrica proporciona una indicación de las funciones asignadas a un solo módulo de software. Cohesión Número de requisitos asociados con un componente de software. Número de funciones por componente de software Interacción promedio de funciones dentro de los componentes. AFI: Interacción de la función promedio.	IS.3.3	Diseño de Software.
Calidad	Cantidad de defectos por KLDC.	Esta métrica busca medir la cantidad de defectos identificados, según la cantidad de líneas de código, lo que provee una medida para la evaluación de las métricas de calidad productiva. # DE: # de defectos identificados. KLDC: Cantidad total de líneas de código.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.

Métricas	Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Cantidad de errores encontrados por KLDC.	Esta métrica busca medir la cantidad de errores identificados, según la cantidad de líneas de código, lo que provee una medida para la evaluación de las métricas de calidad productiva. # E: # de errores identificados. KLDC: Cantidad total de líneas de código.	IS.5.4	Componentes de Software - Casos y Procedimientos de Prueba.
Cantidad de defectos/errores que encuentran los usuarios después de la entrega.	Esta medida busca proveer al equipo de desarrollo de un parámetro de calidad de sus productos posterior a la entrega al usuario. En otras palabras, corresponde al # de defectos o errores encontrados por los usuarios después de la liberación de la aplicación. En otras palabras, el defecto posterior a la prueba.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Tipo y origen de los defectos (requerimientos, análisis y diseño, construcción, integración y pruebas).	Esta medida busca proveer un análisis de la causa de los defectos en la producción, identificando el proceso de desarrollo de software en el cual se está incurriendo en fallas.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Mantenimiento	Cantidad de Componentes.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Volatilidad de los Componentes.	IS.3.3 - 4.3	Componentes de Software - Diseño de Software - Registro de Trazabilidad
	Complejidad de los componentes.	IS.3.3	Diseño de Software.
	Cantidad de requerimientos nuevos, de cambios o mejoras.	# de requerimientos cambiados. IS.3.3	Diseño de Software.
	Cantidad de requerimientos de corrección de defectos.	# de requerimientos corregidos. IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.

Métricas	Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Tiempo promedio de corrección de errores ó defectos.	Esta medida busca proveer la información de cuánto tiempo se está empleado en realizar cambios o corregir defectos del software. Errores descubiertos durante el trabajo para hacer el cambio. Defectos descubiertos después de liberar el cambio al cliente base.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Tiempo promedio de cambios.	T1 = Tiempo de operación T2 = Sumatoria de los intervalos de tiempo entre fallos consecutivos. A = # total de fallos detectados durante el tiempo de operación observado. Nota: 0<X, Y; Mientras más grande mejor.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Porcentaje del código corregido.	-	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Confiabilidad	Tiempo transcurrido entre fallas.	T1 = Sumatoria de los intervalos de tiempo entre fallos consecutivos. A = # total de fallos detectados durante el tiempo de operación observado.	IS.4.3 - 5.5 - 5.6
	Tiempo esperado entre fallas.	TTF: Tiempo fallos de las características del software en un marco de tiempo determinado.	IS.4.3 - 5.5 - 5.6
	Tiempo requerido para corregir una falla.	Contar el número de fallas que no volvieron a suceder durante el período de prueba bajo condiciones similares. A1= # de fallos resueltos. A2= # total de fallos detectados.	IS.4.3 - 5.5 - 5.6
	Nivel de severidad de la falla.	La gravedad de los defectos se puede medir mediante el índice de gravedad de los defectos, una métrica utilizada para medir la calidad del producto directamente. Se mantienen cuatro niveles para comprobar la gravedad de los defectos, es decir, Nivel 4: Crítico, nivel 3: Grave, nivel 2: Medio y nivel 1: Bajo	IS.5.4
Usabilidad	Facilidad de aprendizaje de uso.	T = Tiempo promedio tomado para aprender a utilizar una función correctamente. Nota: 0<T; mientras más pequeño mejor.	IS.4.3 - 5.5 - 5.6

Métricas		Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Rendimiento	Errores cometidos por los usuarios con el uso.	T = Período de tiempo de operación durante la observación (o la suma de tiempo de operación entre operaciones con error humano. N = # de ocurrencias de errores humanos de operación.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Tiempo requerido para realizar las tareas.	Tiempo necesario para completar tareas representativas y predefinidas con funciones de software. Mide todo el tiempo necesario para completar la tarea, independientemente del éxito o el fracaso.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Tiempos de respuesta (Acuerdos SLA).	Ejecutar concurrentemente una gran cantidad de tareas y medir los tiempos de espera como resultado de operaciones de E/S.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Utilización de recursos (Throughput o Thruput) – cantidad de transacciones que pueden ejecutarse concurrentemente con un tiempo de respuesta razonable.	Capacidad: número de solicitudes entrantes de funciones y servicios de software que se pueden gestionar en un período de tiempo determinado. Define la tasa máxima de solicitudes que se pueden atender en este período de tiempo.	IS.4.3 - 5.5 - 5.6	Componentes de Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Tiempo de recuperación.	T = Tiempo de recuperación en cada oportunidad. N = # de casos que se observó al sistema entrar en recuperación.	IS.4.3 - 5.5 - 5.6	Software - Casos y Procedimientos de Prueba - Reporte de Pruebas.
Proyecto Esfuerzo	Cantidad de horas trabajadas.	Mide el tiempo que es efectivo en el proyecto a partir de la relación entre la horas-hombre productivas y las horas-hombre totales en el punto del proyecto.	GP.2.1	Plan de Proyecto (Bajo Nivel).
	Cantidad de personas que trabajan en el proyecto.	# de personas que trabajan en el proyecto.	GP.2.1	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Tiempo transcurrido.	Tiempo transcurrido en el desarrollo del proyecto.	GP.2.1	Plan de Proyecto (Bajo Nivel).

Métricas		Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Costo	Distribución del esfuerzo por fase.	Corresponde al tiempo promedio que tarda el proceso de desarrollo en ir desde el inicio hasta la finalización. Por lo tanto, la reducción del tiempo de ciclo es beneficiosa para la organización ya que le permite a la empresa cumplir con sus plazos en un lapso más rápido.	GP.2.1	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Costo del Desarrollo.	Funcionalidad desarrollada: Comprende desde los requerimientos del cliente hasta la obtención del software.	GP.3.1 - 3.2 -3.3	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Costo del Soporte.	\$ costes totales por soporte	GP.2.1	Plan de proyecto (Bajo Nivel).
	Costo de hs/ persona.	Mide cuánto debe trabajar el equipo de proyecto con el tiempo restante para finalizar el proyecto.	GP. 2.1	Plan de proyecto (Bajo Nivel).
Productividad	Cantidad de software desarrollado por unidad de trabajo.	Tiene como objetivo medir la capacidad de entrega en relación con el tiempo empleado (o unidad de trabajo).	GP.2.1	Plan de Proyecto (Bajo Nivel) - Reporte de Avance - Software.
	Tamaño/Esfuerzo.	El tamaño funcional permite calcular la tasa de productividad esperada, comparándola con datos históricos de la organización. Tasa de productividad puede expresarse como horas/PF.	GP.2.1	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Ritmo de entrega del software por unidad de tiempo transcurrido.	Permite medir el tiempo transcurrido hasta la entrega al usuario o bien medir la capacidad de entrega por unidad de tiempo.	GP.2.1	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
Seguimiento	Cronograma real vs Cronograma estimado.	Mide la variación en términos de cronograma, resaltando el porcentaje de trabajo que aún debe completarse de acuerdo con el cronograma.	GP.2.1	Plan de Proyecto (Bajo Nivel).
	Porcentaje de tareas completadas.	-	GP.2.1	Plan de Proyecto (Bajo Nivel).
	Porcentaje de requerimientos implementados por unidad de tiempo.	-	GP.2.1	Plan de Proyecto (Bajo Nivel).

Métricas		Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
	Porcentaje de tiempo total dedicado a las pruebas.	TCP: Productividad total de casos. TPPSP: Total de pasos de prueba sin procesar. E: Esfuerzos (Horas).	GP.3.3	Casos y Procedimientos de Prueba - Registro de Trazabilidad - Reporte de Avance.
	Porcentaje de error en la estimación de tiempo.	-	GP.3.1 - 3.2 -3.3	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Costo sobre el valor agregado.	El BAIDT (beneficio antes de intereses, pero después de impuestos) b) El WACC (costo promedio del capital) y el capital en cada ejercicio. c) La retribución que esperan los inversores por aportar el capital. Este concepto se determina multiplicando el costo del capital o tasa WACC por el volumen de capital o activos que genera dicho BAIDT.	GP.2.1	Plan de Proyecto (Bajo Nivel).
Estabilidad	Origen de los cambios en los requerimientos.	Análisis del origen de los cambios en los requerimientos.	GP.3.1 - 3.2 -3.3	Plan de Proyecto (Bajo Nivel) - Reporte de Avance.
	Cambios de los requerimientos en el desarrollo.	-	IS.4.4 - 4.5 - 5.3 - 5.4 - 5.5 - 5.6	Casos y Procedimientos de Prueba - Reporte de Pruebas.
	Cambios en los requerimientos en producción.	Número de defectos posteriores a la liberación: número de defectos que se hacen evidentes en el entorno productivo después de la prueba. Un defecto posterior al lanzamiento es un problema informado por el usuario en una versión de software, que debe corregirse.	GP.2.1	Software.
Proceso Esfuerzo	Distribución del esfuerzo por fase del proceso.	-	IS.4.3	Componentes de Software.
	Cantidad de personas requeridas.	# de personas requeridas para el desarrollo del proyecto.	GP.2.1 - 3.1 - IS.4.1 - 5.1 - 6.1	Plan de Proyecto (Bajo Nivel).
	Esfuerzo requerido para corregir un defecto.	Esfuerzo requerido (persona-horas) para hacer el cambio.	GP.1.5 - 1.6	Plan de Proyecto.

Métricas	Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
Esfuerzo requerido para mejorar un defecto.	Muestra la relación entre el número de un tipo de defecto detectado en una fase de prueba específica y el número total de este tipo de defecto en el producto. Cuanto mayor sea este número, que indica una mayor proporción de defectos o defectos importantes detectados en la fase de prueba “apropiada”, mayor será la eficacia de esta fase de prueba para eliminar el tipo de defectos objetivo.	GP.1.5 - 1.6	Plan de Proyecto.
Reusabilidad	Cantidad de componentes reutilizados.	Esta métrica identifica y monitorea los niveles de reutilización del software mediante la evaluación del porcentaje de objetos del ciclo de vida de reutilización durante un período de tiempo. LOCR: Líneas de código de reutilización en el módulo del sistema. TLOCM: Total de líneas de código en el sistema o módulo.	IS.5.5 - 5.6 Reporte de Pruebas - Registro de Trazabilidad - Reporte de Avance.
	Grado de reusabilidad de los componentes.	IS.4.4 - 4.5 - 4.6 - 5.3 - 5.4 - 5.5 - 5.6	Casos y Procedimientos de Prueba - Registro de Trazabilidad - Reporte de Avance.
Calidad	Cantidad de defectos sin corregir.	Defectos encontrados sin corregir durante las pruebas.	IS.4.4 - 4.5 - 4.6 - 5.3 - 5.4 - 5.5 - 5.6 Casos y Procedimientos de Prueba - Registro de Trazabilidad - Reporte de Avance.
	Costo de corrección de defectos.		IS.4.4 - 4.5 - 4.6 - 5.3 - 5.4 - 5.5 - 5.6 Casos y Procedimientos de Prueba - Registro de Trazabilidad - Reporte de Avance.
	Eficacia en la eliminación de defectos.	E: Número de errores encontrados antes de entregar al usuario final. D: Número de defectos encontrados después de la entrega.	IS.4.4 - 4.5 - 4.6 - 5.3 - 5.4 - 5.5 - 5.6 Casos y Procedimientos de Prueba - Registro de Trazabilidad - Reporte de Avance.
	Cantidad de veces que un módulo fue probado.		IS.4.3 Componentes de Software.

	Métricas	Descripción	Tarea ISO/IEC 29110	Formato o producto de trabajo
	Tamaño de un módulo.	Cantidad de líneas de código del componente.	IS.5.5 - 5.6	Componente de software. [Corregido]
	Tiempo promedio en corrección de defectos.	Mide el tiempo que se tarda en resolver los defectos asociados a la ejecución del proyecto.	IS.5.5 - 5.6	Reporte de Pruebas - Registro de Trazabilidad - Reporte de Avance.
Soporte a los clientes	Tamaño del backlog de defectos.	NPCDM: Número de problemas cerrados durante el mes. NLLPDM: Número de problemáticas durante el mes.	IS.4.3	Componentes de Software.
	Tiempo de respuesta en atender los defectos.	TACD: Tiempo actual para resolver defectos. TPCD: Tiempo planificado para la resolución de defectos.	IS.5.5 - 5.6	Reporte de Pruebas - Registro de Trazabilidad - Reporte de Avance.
	Tiempo de resolución de defectos.	TACD: Tiempo actual para resolver defectos.	IS.5.5 - 5.6	Reporte de Pruebas - Registro de Trazabilidad - Reporte de Avance.
Herramientas	Soporte de herramientas para procesos propuestos.	-	IS.5.5 - 5.6	Reporte de Pruebas - Registro de Trazabilidad - Reporte de Avance.

Agradecimientos

Este trabajo de investigación ha sido posible gracias a la participación de Mipymes de desarrollo de software en Zacatecas, México y por el trabajo aportado por los estudiantes de **México** (*Edgar Bonilla, Israel Faustino, Einar Jhordany, Elizabeth Villanueva del Centro de Investigación en Matemáticas, Unidad Zacatecas, México*) y de **Chile** (*Octavio Zepeda Valero de la Escuela de Ingeniería Universidad Católica del Norte*).

Referencias

Basso, D. (2014). Propuesta de Metricas para Proyectos de Explotacion de Informacion. Revista Latinoamericana de Ingenieria de Software, 2(4), 157. <https://doi.org/10.18294/relais.2014.157-218>.

- Constanzo, M. A., Casas, S. I., Marcos, C. A. (2014). Comparación de modelos de calidad, factores y m.tricas”, Informes Cient.ficos T.cnicos - UNPA, 6(1), 1–36. Accedido el 14 de enero de 2022. <https://doi.org/10.22305/ict-unpa.v6i1.89>
- Choras, M., Springer, T., Kozik, R., López, L. (2020). Measuring and Improving Agile Processes in a Small-Size Software Development Company”, IEEE Access, 8, 78452–78466. <https://doi.org/10.1109/access.2020.2990117>
- Galván-Cruz, S., Muñoz, M., Mejía, J., Laporte, C.Y., Negrete, M. (2021). Building a Guideline to Reinforce Agile Software Development with the Basic Profile of ISO/IEC 29110 in Very Small Entities. In: Mejia J., Muñoz M., Rocha Á., Quiñonez Y. (eds) New Perspectives in Software Engineering. CIMPS 2020. Advances in Intelligent Systems and Computing, vol 1297. Springer, Cham. https://doi.org/10.1007/978-3-030-63329-5_2
- Gehlot, S. (2019). Complexity Metrics for Component Based Software – A Comparative Study. Journal of Computers, 14(6), 389–396. <https://doi.org/10.17706/jcp.14.6.389-396>
- Haindl, P & Plosch, R. (2021). Value-oriented quality metrics in software development: Practical relevance from a software engineering perspective, IET Software, pp. 1–18. <https://doi.org/10.1049/sfw2.12051>
- ISO/IEC. (2011). ISO/IEC 29110-4-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 4-1: Profile specifications: Generic profile group. <https://www.iso.org/standard/51154.html>
- ISO/IEC 29110. (2011). Software engineering- Lifecycle profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. ISO/IEC TR 29110-5-1-2:2011. Technical report. <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>.
- INEGI. (2018). INEGI Presenta Resultados de la Encuesta Nacional Sobre Productividad y Competitividad de las Micro, Pequeñas y Medianas Empresas (ENAPROCE) 2018. En A. Asociación Mexicana de Secretarios de Desarrollo Económico (Ed.), *Comunicado de Prensa Num. 448/19*, (pág. 16).
- Laporte, C. Y. & Mejia, J. (2020). Delivering Software- and Systems-Engineering Standards for Small Teams. Computer, 53(8), 79–83. <https://doi.org/10.1109/mc.2020.2993331>
- Laporte, C. Y., Muñoz M., Gerançon, B. (2017). The education of students about ISO/IEC 29110 software engineering standards and their implementations in very small entities. IEEE Canada International Humanitarian Technology Conference (IHTC), pp. 94-98. <https://doi.org/10.1109/IHTC.2017.8058208>.
- Mejía, J., Bonilla, E., Faustino, I., Jhordany, E., Villanueva, E. (2020). GPIS:Tool to assess internal processes and projects based on ISO/IEC 29110,” 2020 9th International Conference On Software Process Improvement (CIMPS), pp. 44-51. <https://doi.org/10.1109/CIMPS52057.2020.9390129>.

- Mejia, J., Bonilla, E., Faustino, I., Jhordany, E., Villanueva, E. (2021). Apoyando a las Mipymes en la Evaluación Interna de Procesos y Proyectos para la Certificación en la Norma ISO/IEC 29110”, RISTI - Revista Iberica de Sistemas e Tecnologias de Informação, (41), 80–96. <https://doi.org/10.17013/risti.41.80-96>
- Noor, H., Hayat, B., Hamid, A., Wakeel T., Nasim, R. (2020). Software Metrics: Investigating Success Factors, Challenges, Solutions And New Research Directions. International Journal Of Scientific & Technology Research, 9(8), 38–44. <http://www.ijstr.org/final-print/aug2020/Software-Metrics-Investigating-Success-Factors-Challenges-Solutions-And-New-Research-Directions.pdf>
- Rashid, J., Mahmood, T., Nisar, M. W. (2019). A Study on Software Metrics and its Impact on Software Quality. Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan, 24(1), 69–82.
- Vogel, M., Knapik, P., Cohrs, M., Szyperrek, B., Pueschel, W., Etzel, H., Fiebig, D., Rausch, A., Kuhrmann, M. (2020). Metrics in automotive software development: A systematic literature review. Journal of Software: Evolution and Process, e2296. <https://doi.org/10.1002/smr.2296>