

Implementación de Métricas a través del Modelo GQM en un Centro de Desarrollo de Software

Implementation of Metrics through the GQM Model in a Software Development Center

J Jesús Minero, Elvia Lara, Josefina García

Instituto Tecnológico Superior de Nochistlán

Zacatecas, México

{jesus.minero, elvia.lara, josefina.garcia}@itsn.edu.mx

Resumen — El presente trabajo describe una propuesta de un sistema de métricas de software para medir y mejorar los procesos de un centro de desarrollo de software. Para ello se usó la metodología GQM, que tiene los siguientes niveles: nivel conceptual (Goal/Objetivo), nivel operativo (question/pregunta) y nivel cuantitativo (Metric/Métrica). Se analizaron las métricas definidas para proponer el uso de un prototipo para la recolección y análisis de las métricas y resumir los beneficios que se obtienen de su uso desde el punto de vista del mejoramiento continuo de procesos. Los temas tratados en este artículo pueden interesar a las organizaciones que desean mejorar sus procesos de software mediante el uso sistemático de métricas de calidad y productividad para desarrollo y mantenimiento de software.

Palabras Clave - métricas; GQM; prototipo; calidad.

Abstract — The present work describes a proposal for a software metrics system to measure and improve the processes of a software development center. For this purpose, the GQM methodology was used, which has the following levels: conceptual level (Goal), operational level (Question), and quantitative level (Metric). The defined metrics were analyzed to propose the use of a prototype for the collection and analysis of the metrics and to summarize the benefits obtained from its use from the perspective of continuous process improvement. The topics discussed in this article may interest organizations that wish to improve their software processes through the systematic use of quality and productivity metrics for software development and maintenance.

Keywords - metrics; GQM; prototype; quality.

I. INTRODUCCIÓN

En la actualidad, ha habido un notable aumento en el número de empresas desarrolladoras de software, acompañado por una creciente demanda de productos en este sector. Para estas empresas, la calidad del software es crucial como factor distintivo de competitividad y de imagen ante sus clientes, ya que la falta de calidad puede resultar en pérdidas económicas significativas. Por esta razón, las actividades relacionadas con la calidad del software y del proceso y su evaluación están ganando cada vez más relevancia [1].

Según Sarwosri et al. [2] obtener un producto de calidad requiere un proceso de calidad que debe ser cuidadosamente

planificado y gestionado de manera continua. Esta necesidad lleva a la adopción de metodologías que fomenten la disciplina tanto a nivel individual como de equipo, además es necesario definir métricas de desarrollo, y que sean aceptadas por los ingenieros. Aunque las metodologías están disponibles, aún faltan herramientas de soporte adecuado y que faciliten la recolección y gestión de métricas [3].

Por ello, es fundamental que las empresas de desarrollo formalicen sus procesos y modelos de medición para lograr productos de mayor calidad y sin defectos cuando se lancen al mercado [4]. Sin la implementación de métricas para medir los procesos, no sería posible demostrar de manera tangible el mejoramiento o la eficiencia en la ejecución de dichos procesos [5].

Sin embargo, hay varios problemas a considerar al elegir métricas. En primer lugar, deben ser seleccionadas con mucho cuidado, ya que las métricas definidas incorrectamente pueden desviar el proyecto de su objetivo y agotar el tiempo y los recursos asignados al proyecto. El segundo problema es que existe un número infinito de métricas y formas de categorizarlas. La sobrecarga en su número puede llevar a un cambio de prioridades y a la pérdida de enfoque en el proyecto [6]. Debido a que el problema de la selección de métricas es tan importante y complejo, han surgido varios enfoques estructurados y formales para derivar métricas y simplificar este proceso [7]. Entre ellos, el más conocido en el ámbito científico es el modelo de Objetivo-Pregunta-Métrica (Goal-Question-Metric, GQM), inventado por Victor Basili. Con esta metodología, es posible resolver el primer problema mencionado anteriormente, excluir la posibilidad de desviaciones de los objetivos del proyecto con métricas inapropiadas [6].

GQM, según Basili et al. [8], significa “objetivo, pregunta, métrica” y permite definir los objetivos aclarando las preguntas sobre cómo se puede alcanzar el objetivo con los datos recopilados. El enfoque GQM es un método diseñado para obtener una métrica que evalúe un objetivo específico mediante preguntas. Este método se desarrolla identificando uno o más objetivos de calidad y formulando preguntas que los describan. A continuación, se establecen métricas que proporcionarán un

resultado único basado en las respuestas a esas preguntas, con el propósito de determinar el grado de cumplimiento del objetivo definido [9].

El presente trabajo desarrolla una propuesta de un sistema de métricas que asegure el proceso de medición y análisis en un centro de desarrollo de software, y propone un prototipo con el fin de recopilar, almacenar y mostrar las estadísticas de las métricas definidas.

En la sección 2 del presente artículo se describe el marco teórico; en la sección 3 se describe la metodología GQM; en la sección 4 se presenta el prototipo y en la sección 5 se muestran las conclusiones.

II. MARCO TEÓRICO

Dentro del contexto de la ingeniería de software, una métrica de software proporciona una medida cuantitativa del grado en que un producto o proceso de software posee algún atributo. La medición permite ganar comprensión acerca del proceso y del proyecto, al proporcionar un mecanismo de evaluación objetiva [10].

El dominio de la métrica del software se divide en métricas de proceso, proyecto y producto [11]:

Métricas del proceso: estas métricas se utilizan para evaluar y mejorar los procesos de desarrollo y mantenimiento de software. Miden atributos como la eficiencia y calidad del proceso, permitiendo a los ingenieros de software optimizar la metodología de trabajo y desarrollar estrategias de mejora continua.

- Métricas del producto: son medidas cuantitativas utilizadas para evaluar las características de un producto de software, tales como tamaño, complejidad, diseño, rendimiento y nivel de calidad. Estas métricas ayudan a determinar si un producto está siendo desarrollado de acuerdo con los requisitos del usuario y permiten detectar y corregir problemas potenciales antes de que causen fallos críticos.

- Métricas del proyecto: son medidas que describen las características y la ejecución de un proyecto de software, como el número de desarrolladores, el costo, la programación y la productividad. Estas métricas ayudan a los gestores de proyectos a monitorear el progreso, comparar el esfuerzo, tiempo y costo real con las estimaciones iniciales, y tomar decisiones para reducir riesgos y mejorar la calidad del proyecto.

Existen múltiples metodologías y procesos que se han definido para determinar el conjunto apropiado de métricas para una organización de software. Una de ellas es Meta/Interrogante/Métrica (GQM en inglés). GQM es un método estructurado utilizado para definir y seleccionar métricas basadas en objetivos específicos. Involucra tres pasos clave [12]:

1. **Definir objetivos:** delinear claramente lo que se quiere lograr.
2. **Formular preguntas:** desarrollar preguntas que ayuden a determinar si los objetivos se están cumpliendo.

3. **Seleccionar métricas:** elegir métricas que proporcionen los datos necesarios para responder a estas preguntas.

GQM proporciona una manera eficiente de establecer mediciones tanto para el proceso como para los resultados de un proyecto [9].

La Fig. 1 muestra la representación del modelo GQM en forma de árbol, cuya raíz es el objetivo. Las ramas están representadas por preguntas que permiten especificar más el objetivo. Y las hojas son las métricas, el resultado medible de todo el modelo [6].

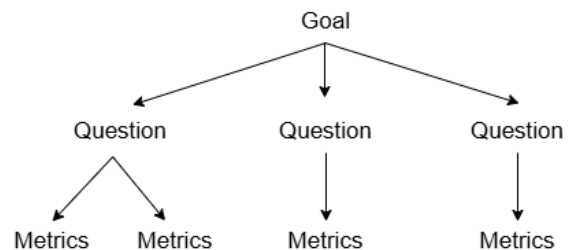


Figura 1. Estructura Jerárquica del modelo GQM [6]

El modelo de medición GQM tiene tres niveles:

Nivel Conceptual (Goal/Objetivo): un objetivo se establece para un objeto por diversas razones, en relación con distintos modelos de calidad, desde múltiples perspectivas y considerando un entorno específico. Los objetos de medición son: productos, procesos o proyectos.

Nivel Operativo (Question/Pregunta): a partir del objetivo, se formula un conjunto de preguntas con el fin de verificar su cumplimiento. Estas preguntas tienen como objetivo caracterizar el objeto de medición (ya sea producto, proceso o proyecto) en relación con un aspecto de calidad específico y determinar su calidad desde la perspectiva seleccionada.

Nivel Cuantitativo (Metric/Métrica): para cada pregunta se asocia un conjunto de datos, formulando métricas con el objetivo de proporcionar una respuesta de manera cuantitativa [13].

III. METODOLOGÍA

En esta sección, se presenta el desarrollo de la metodología GQM, así como la propuesta de métricas a nivel producto y proceso, que apoyen a las entidades muy pequeñas de desarrollo de software. Para esto se ha tomado como caso de estudio el Centro de Desarrollo de Software (CEDESOF) del Instituto Tecnológico Superior de Nochistlán certificado en el estándar ISO/IEC 29110.

Estas métricas permitirán conocer y mejorar la calidad de los productos de software y las estimaciones de tiempo y costo de los proyectos a partir de determinados indicadores de medición para tomar decisiones oportunas.

A continuación, se describe el desarrollo de los tres niveles de la metodología GQM.

A. Nivel Conceptual (Objetivos)

En este nivel se identificaron los objetivos del Centro de Desarrollo de Software.

TABLA I. OBJETIVOS

| No. Objetivo | Descripción |
|--------------|---|
| Objetivo 1 | Mejorar la estimación del tiempo del proyecto |
| Objetivo 2 | Mejorar la estimación del costo del proyecto |
| Objetivo 3 | Mejorar la calidad del software |

B. Nivel Operativo (Preguntas)

En este nivel se definieron un conjunto de preguntas a partir de los objetivos con el propósito de verificar su cumplimiento y determinar el objeto de medición (producto, proceso o proyecto). La Tabla II muestra la lista de preguntas definidas y su relación con los objetivos.

TABLA II. OBJETIVOS Y PREGUNTAS

| Objetivo | Pregunta |
|----------|---|
| 1 | ¿Cuál es el total de horas planeadas por proyecto? |
| | ¿Cuál es el total de horas reales por proyecto? |
| | ¿Cuál es el total de horas planeadas por fase? |
| | ¿Cuál es el total de horas reales por fase? |
| 2 | ¿Cuál es el costo total planeado por proyecto? |
| | ¿Cuál es el costo total real por proyecto? |
| 3 | ¿Cuál es el total de defectos durante el desarrollo? |
| | ¿Cuál es el total de defectos después de la entrega del software? |

C. Nivel Cuantitativo (Métricas)

En este nivel se definieron un conjunto de datos para cada pregunta formulando métricas con el fin de proporcionar una respuesta de manera cuantitativa.

TABLA III. PREGUNTAS Y MÉTRICAS

| Objetivo | Pregunta | Métrica |
|----------|--|---------------------------------------|
| 1 | ¿Cuál es el total de horas planeadas por proyecto? | Total de horas planeadas por proyecto |
| | ¿Cuál es el total de horas reales por proyecto? | Total de horas reales por proyecto |
| | ¿Cuál es el total de horas planeadas por fase? | Total de horas planeadas por fase |
| | ¿Cuál es el total de horas reales por fase? | Total de horas reales por fase |
| 2 | ¿Cuál es el costo total planeado por proyecto? | Costo total planeado por proyecto |
| | ¿Cuál es el costo total real por proyecto? | Costo total real por proyecto |

| | | |
|---|---|--|
| 3 | ¿Cuál es el total de defectos durante el desarrollo? | Total de defectos durante el desarrollo |
| | ¿Cuál es el total de defectos después de la entrega del software? | Total de defectos después de la entrega del software |

Las métricas propuestas para el logro de los objetivos del CEDESOF son:

- Total de horas planeadas por proyecto
- Total de horas reales por proyecto
- Total de horas planeadas por fase
- Total de horas reales por fase
- Costo total planeado por proyecto
- Costo total real por proyecto
- Total de defectos durante el desarrollo
- Total de defectos después del desarrollo

Una vez que se identificaron y definieron las métricas de calidad mediante la metodología GQM, el siguiente paso fue diseñar un prototipo para el registro y análisis de las métricas con la finalidad de mejorar el proceso de desarrollo del centro de desarrollo de software.

IV. PROTOTIPO

El prototipo propuesto es un sitio web para la recopilación, almacenamiento y análisis de métricas del Centro de Desarrollo de Software del Instituto Tecnológico Superior de Nochistlán. El objetivo principal del prototipo es mantener un historial detallado de las métricas definidas que permita analizar el comportamiento de los procesos, productos y proyectos. Este historial de métricas es clave para la mejora continua, facilitando la evaluación, identificación de áreas de mejora, y la toma de decisiones informadas basadas en datos cuantitativos. Además, proporciona una base sólida para auditorías y cumplimiento de estándares de calidad.

En la Fig. 2 se presenta el diseño de la interfaz principal del prototipo, la cual muestra la información general del Centro de Desarrollo de Software. El prototipo cuenta con dos perfiles principales: el perfil del cliente, que permite realizar una solicitud de un proyecto de software, y el perfil del administrador, que controla las peticiones del cliente y gestiona las métricas de los proyectos.

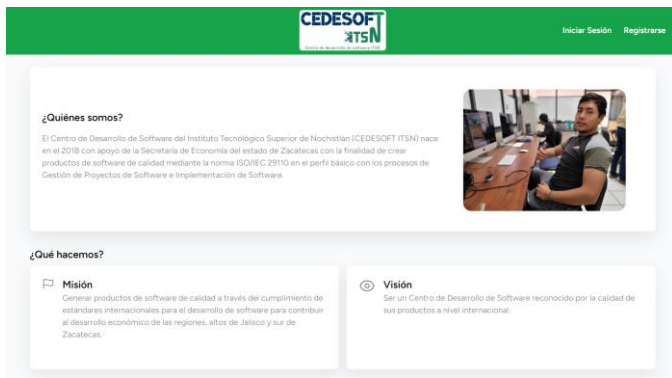


Figura 2. Interfaz principal del prototipo

En la Fig. 3 se muestra el perfil de usuario donde puede hacer la solicitud vía formulario de un proyecto de software y ver el estado de las solicitudes.

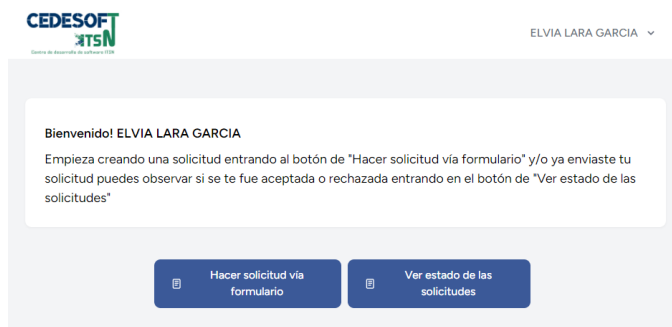


Figura 3. Interfaz principal del usuario cliente

En la Fig. 4 se muestran las acciones que puede realizar el perfil de administrador:

- **Dashboard:** muestra los datos globales de los proyectos.
- **Gestión de Solicitud de Proyectos:** permite ver los usuarios y las solicitudes realizadas, además de aceptar o rechazar dichas solicitudes.
- **Control de Proyecto:** muestra la solicitud del proyecto y su estado actual.
- **Informe de los Proyectos:** proporciona los datos generales del proyecto, incluyendo nombre, descripción, registro de tiempos, costos y defectos.
- **Generación de Reportes:** muestra el registro de los datos generales por proyecto y sus métricas.
- **Generación de Actas de Aceptación:** permite generar el acta de aceptación para la liberación de los proyectos.
- **Usuarios:** permite seleccionar usuarios registrados como solicitantes de proyectos o administradores, y eliminar usuarios.

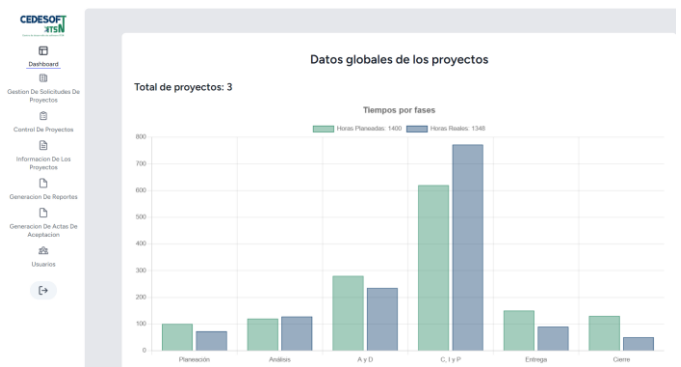


Figura 4. Interfaz principal del usuario administrador

La Fig. 5 muestra el registro de datos y métricas del proyecto, de la opción "Información de los proyectos" del menú del administrador, el formulario se divide en cuatro partes:

1. **Datos generales:** incluye el nombre del proyecto y una descripción.
2. **Registro de Tiempo:** abarca la fecha de inicio, fecha de finalización, horas planeadas, horas reales y horas invertidas por fase (Planeación, Análisis, Arquitectura y Diseño, Construcción, Integración y pruebas, Entrega y Cierre).
3. **Registro de Costos:** contiene el costo total planeado y el costo total real.
4. **Registro de Defectos:** registra el total de defectos durante el desarrollo y después del desarrollo.

| Datos generales | | |
|---|---|----------------------|
| Nombre del proyecto | <input type="text"/> | |
| Descripción del proyecto | <input type="text"/> | |
| Registro de tiempo | | |
| Fecha inicio | Fecha fin | |
| <input type="text" value="dd/mm/aaaa"/> | <input type="text" value="dd/mm/aaaa"/> | |
| Horas planeadas | Horas reales | |
| <input type="text"/> | <input type="text"/> | |
| Horas invertidas por fase | | |
| | Plan | Real |
| Planificación | <input type="text"/> | <input type="text"/> |
| Análisis | <input type="text"/> | <input type="text"/> |
| Arquitectura y diseño de software | <input type="text"/> | <input type="text"/> |

Construccion, integracion y pruebas

Entrega del proyecto

Cierre del proyecto

Registro de costos

Costo total planeado

Costo total real

Registro de defectos

Total de defectos durante el desarrollo

Total de defectos despues del desarrollo

ENVIAR

Figura 5. Registro de métricas del proyecto

La sección del Dashboard proporciona una visualización clara y concisa de las métricas definidas de los proyectos. Estos gráficos muestran los datos planeados en comparación con los datos reales, analizando tiempo por fase, tiempo total, costos y defectos.

En la Fig. 6 se visualizan los tiempos planeados y los tiempos reales de cada fase (Planeación, Análisis, Arquitectura y Diseño, Construcción, Integración y pruebas, Entrega y Cierre) de los proyectos. Estas gráficas permiten identificar rápidamente las áreas que consumen más tiempo y evaluar la eficiencia del ciclo de desarrollo, y para proyectos futuros mejorar la estimación de tiempo por fases.

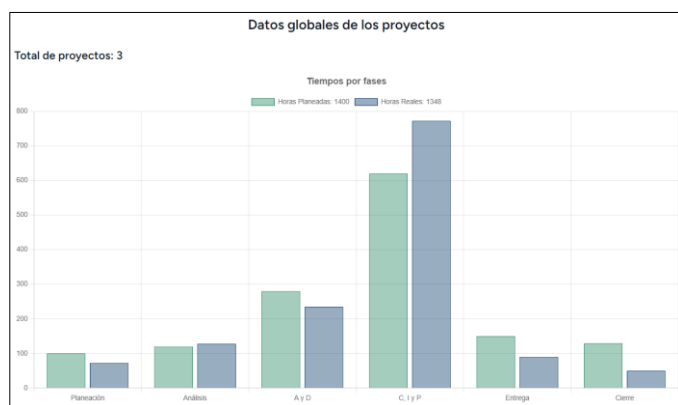


Figura 6. Ejemplo de análisis de tiempo por fases plan vs real

La Fig. 7 muestra tres gráficas circulares, la primera analiza el tiempo total planeado contra el tiempo total real para el desarrollo del proyecto, la segunda analiza el costo total planeado contra el tiempo total real, y la última gráfica muestra

una comparación de los defectos detectados durante el desarrollo del software contra los defectos encontrados después del desarrollo del proyecto. Esta comparación es necesaria para evaluar la efectividad del proceso de desarrollo y la calidad del producto final.

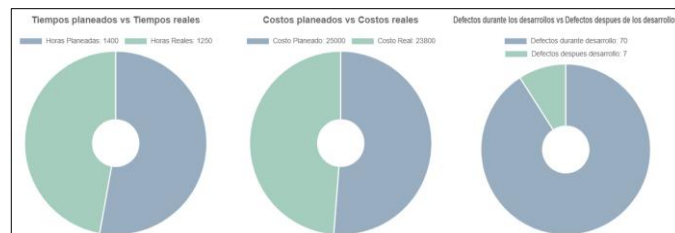


Figura 7. Ejemplo de análisis de tiempo, costos y defectos

V. CONCLUSIONES

A través de la metodología GQM se implementó un sistema de métricas de software para mejorar los procesos en un centro de desarrollo de software, se logró establecer una estructura clara y efectiva para la definición y recolección de métricas, abarcando desde los objetivos generales hasta las métricas cuantitativas específicas. Este enfoque permite una evaluación precisa y continua de los procesos, contribuyendo significativamente a la mejora de la calidad del software producido.

El análisis de las métricas definidas revela la necesidad de contar con una herramienta que facilite la recolección y el análisis de datos. El prototipo propuesto en este trabajo cumple con este propósito, proporcionando un sistema web que no solo recopila y almacena las métricas, sino que también ofrece visualizaciones claras y detalladas de los datos a través de gráficas.

Los beneficios de utilizar un sistema de métricas bien definido y una herramienta de apoyo se extienden más allá de la simple mejora del proceso de desarrollo. Las métricas permiten una mejor estimación del tiempo y costo de los proyectos, así como una evaluación continua de la calidad del software. Esto, a su vez, ayuda a las organizaciones a optimizar sus recursos, reducir costos y aumentar la satisfacción del cliente, posicionándose mejor en un mercado competitivo.

Finalmente, este estudio subraya la relevancia de las métricas de software en el contexto del mejoramiento continuo. Las organizaciones que adopten este enfoque podrán no solo mejorar sus procesos y productos, sino también fomentar una cultura de calidad y eficiencia. La metodología GQM y la herramienta propuesta representan un paso significativo hacia la profesionalización y optimización del desarrollo de software, ofreciendo un modelo replicable y adaptable para otras entidades en la industria del software.

REFERÊNCIAS BIBLIOGRÁFICA

- [1] J. Calabrese y R. Muñoz, «Asistente para la evaluación de calidad de productos de software según la familia de normas ISO/IEC 25000 utilizando el enfoque GQM,» 2018.
- [2] Sarwosri, M. Choirun Nisa, S. Rochimah, R. Januar Akbar y U. Laili Yuhana, «Measuring the quality of the Development process academic system with E-GQM method,» *International Journal on Informatics Visualization*, pp. 27-31, 2021.
- [3] J. C. Camacho Beltrán, «Prototipo web para la gestión de métricas como soporte a la metodología de desarrollo TSP,» Bogotá, 2017.
- [4] A. J. Caballero Alvarado, «Propuesta de métricas de calidad en el proceso de desarrollo de software en la oficina de sistemas e ingeniería de la información de la universidad privada Antenor Orrego,» Trujillo, 2017.
- [5] D. F. Veintimilla Alvarado y C. A. Chicaiza Caiza, «Dianóstico del uso de métricas de calidad de la norma ISO/IEC 25000 en mipymes de desarrollo de software de países miembros del Hastqb,» Quito, 2020.
- [6] A. Kruglov y G. Succi, *Developing sustainable and energy-efficient software systems*, Switzerland: Springer, 2023.
- [7] M. Farina, A. Gorb, A. Kruglov y G. Succi, «Technologies for GQM-Based metrics recommender systems: a systematic literature review,» *IEEE*, 2022.
- [8] Basili y R. Victor, *The goal question metric approach*, New York: Wiley, 1994.
- [9] J. Calabrese, S. Esponda, A. Pasini y P. Pesado, «Modelo de evaluación de datos utilizando el enfoque GQM,» *CACIC*, 2020.
- [10] R. S. Pressman y B. R. Maxim, *Ingeniería de software un enfoque práctico*, New York: McGraw-Hill, 2021.
- [11] S. Kumar, «Software metrics in software engineering,» 2022.
- [12] F. Yahya, R. J. Walters y G. B. Wills, «Using Goal-Question-Metric (GQM) approach to Assess Security in Cloud Storage,» *Springer Link*, pp. 223-240, 2017.
- [13] G. Kaplan, G. Riera, M. I. Bertín y G. Blanco, «Evaluación de requisitos de software utilizando GQM,» *CACiC*, pp. 342-351, 2023.