ISSN-e: 1856-9811

QUEST FOR QUALITY IN AI-ENABLED SDI-TYPE TOOLS: DOMAIN MODELING

SEARCH FOR QUALITY IN AI-ENABLED IDE TOOLS: DOMAIN MODEL

GIOVANNA CIANFAGLIONE
cianfaglione.giovanna@correo.unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela

FRANK CAICEDO
frank.caicedo@correo.unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela)

MARÍA PÉREZ
maperez@unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela)

DINARLE M ORTEGA
dortega@ucab.edu.ve
Universidad Católica Andrés Bello (Venezuela)

Summary

Technological advancement has driven the growth of artificial intelligence (AI)-enabled Integrated Development Environment (IDE)-type software tools, offering developers unlimited opportunities to address problems, automate tasks and enhance creativity in a variety of areas, from data analysis to content generation. In this article, a domain model of AI-enabled SDI-type tools is specified considering the ISO 25059 standard as a fundamental basis in the search for a quality estimation model for this type of tools. This domain model is built through an exhaustive analysis, literature review, study of the most relevant plugins and using the UML standard language, in particular the class diagram, due to its semantic richness to formulate it in a precise and understandable way. It is important to recognize that this model may have certain limitations associated with the information available, provided by manufacturers and bibliographic sources. This model provides a standardized structure that facilitates communication and contributes to the success of a target project by facilitating the understanding of complex realities of quality in AI-enabled tools, especially in the context of SDI.

Keywords: DevSecOps, Integrated Development Environment (IDE), AI-enabled tools, quality, ISO 25059.

Abstract

Technological advancement has driven the growth of Artificial Intelligence (AI)-enabled Integrated Development Environment (IDE) software tools, offering developers unlimited opportunities to address issues, automate tasks, and enhance creativity across various domains, from data analysis to content generation. This article specifies a domain model for AI-enabled IDE tools considering ISO 25059 as a fundamental basis in the quest for a quality estimation model for such tools. This domain model is constructed through comprehensive analysis, literature review, examination of the most relevant plugins, and utilizing the standard UML language, particularly that of the class diagram, due to its semantic richness for formulating it precisely and understandably. It's important to acknowledge that this model may have certain limitations associated with available information provided by manufacturers and bibliographic sources. Nonetheless, this model provides a standardized structure that facilitates communication and contributes to the success of a target project by aiding in the understanding of complex realities inherent in AI-enabled tools' quality, especially in the context of IDEs.

Key Words: DevSecOps, Integrated Development Environment (IDE), AI-enabled tools, quality, ISO 25059.

RECEIVED: 09-01-2024 ACCEPTED: 11-07-2024 PUBLISHED: 30-09-2024

How to quote: Cianfiaglione et al. (2024). Quest for quality in AI-Enabled SDI-TypeTools: domain modeling. *Anales*, 41, 15 - 32. https://doi.org/10.58479/acbfn.2024.111

ISSN-e: 2244-8276

INDEX

1. Introduction

In an increasingly digitized world, where software has become a fundamental part of technology, software quality is considered an important aspect to take into account (Techopedia, 2023). From the operation of critical systems to personal entertainment, confidence in the accuracy, security and efficiency of software is crucial. In this sense, this article aims to propose a domain model of the context of AI-enabled IDEs (Integrated Development Environment), considering the ISO 25059 standard, respectively (ISO 25000, 2022). Quality is a complex aspect since the domain model promotes its understanding. Faced with complex situations, software engineers need to use some techniques to help them better understand the problem they are analyzing before starting with the software implementation. In this case, the use of the domain model serves to identify and represent the concepts of the problem domain and thus support its analysis (Jardiel, 2015).

Therefore, this article is divided into three sections, in addition to the Introduction, Conclusions and Recommendations. It begins with the Context section in which the crucial definitions for the proposal and creation of the domain model are addressed, this is found in the second section, how each of the relationships of the definitions represented with a class diagram is modeled complying with the standards of the UML language and how each of them is crucial for the development of a quality software product. The next section reflects on what is found in the literature about the AI plugins that exist today in the market for coding, integration or other functions that can be performed by IDE tools and are of great help to programmers today. This section includes a table containing each plugin with its most relevant characteristics and how each of these relate to the ISO 25059 standards according to the characteristics and subcharacteristics proposed by this standard.

This is a research in progress that seeks to formulate a model to estimate the quality for this type of AI-enabled tools and thus establish and prescribe the characteristics and sub-characteristics that should be present in such systems. Having a quality model facilitates technical criteria that provide guidance and references to stakeholders when selecting them.

2. Context

Four topics are analyzed to reflect on the reality of quality in AI-enabled SDI-type software tools.

2.1 Software Development Process

Based on the authors ITE Corp (2022) and Prieto (2023), the software development process is a set of steps or stages that are followed to create a software product, such as requirements gathering, analysis and specification; design and development; testing, installation and deployment; maintenance and support. Following these steps allows the development of efficient, secure and useful software programs for users and implies planning, carrying out and efficiently managing a project in order to execute it successfully and ensure that it fulfills the objective for which it was designed. On the other hand, IBM (n/d) also states that "Software development refers to a set of IT activities dedicated to the process of creating, designing, deploying and supporting software."

It is concluded then, that the software development process is a structured and orderly process of activities consisting of several stages or steps, which range from its conception to its maintenance. In other words, it is a set of interrelated activities with an agreed order whose final objective is the creation of a software product that satisfies the users' needs. Finally, with reference to software efficiency, security and usability, which are key to software development, they imply an adequate planning, execution and management of the project to ensure that it meets the established objectives and is beneficial to users (Maida and Paciencia, 2015).

2.2 DevSecOps

Dynatrace (2024), states that "DevSecOps is a collaborative framework that extends the impact of DevOps by adding security practices to the software development and delivery process. It resolves the tension between DevOps teams that want to release software quickly and security teams that prioritize security above all else." He further notes that by integrating application security principles and practices into software development and operations, this enables teams to deliver new services and software development at a much faster rate without compromising application security (Dynatrace, 2024). Also, RedHat (2023) notes that "DevSecOps stands for development, security and operations, defined as an approach that addresses culture, automation and platform design, and integrates security, as a shared responsibility throughout the IT lifecycle."

On the other hand, according to AWS (2023), "DevSecOps is the practice of integrating security testing into every stage of the software development process. It includes tools and processes that foster collaboration between developers, security specialists, and operations teams to create software that is efficient and secure." The goal of DevSecOps is to help development teams address security issues effectively, thus contributing to an alternative to supplant older software security practices that did not accommodate rapid updates and tighter timelines. In this way DevSecOps became a cultural transformation that makes security a shared responsibility for all software developers (AWS, 2023).

Note then that DevSecOps is a philosophy, a framework and a set of practices that seeks to improve collaboration and communication between software development and IT operations and security teams. Its main objective is to shorten the software development lifecycle, streamline the delivery of applications and improve the quality of the final product, and the use of these practices is based on three fundamental pillars: people, processes and technology in order to develop and deliver software faster, more reliably and efficiently. In another sense, DevSecOps, according to Ingeno (2023) is based on these four principles: it is a culture and philosophy, automates the entire development process, uses the Lean approach development (Information-technologies, 2018), finally seeks to measure the entire process and share. In turn, it contemplates this set of stages: testing, planning, delivery, monitoring, configuration, feedback, operate, build, deploy and code. Among the tools to be used in the development process, particularly in the "code" and "build" stage of DevSecOps, IDE tools stand out.

2.3 IDE

AWS (2023), states that "An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as editing, creating, testing, and packaging software into one easy-to-use application. Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to facilitate their work." On the other hand, RedHat (2013) and Datascientest (2022) define an IDE as an integrated software system for application design that combines common developer tools into a single graphical user interface (GUI).

Thus, an IDE can be defined as a software application that provides a set of tools to facilitate software development that helps programmers make their code more efficient. Its main objective is to increase programmers' productivity by integrating into a single graphical user interface (GUI) that combines various capabilities, such as editing, creating, testing and packaging software, in a single easy-to-use application. In other words, it is an essential tool for any programmer looking to increase their productivity and improve the quality of their work, which may have integrated some type of AI component.

2.4 AI Enabled Tools

RedHat (2024), indicates that "artificial intelligence (AI) refers in general to Computer Science processes and statistical algorithms that are capable of simulating and improving human intelligence. In other words, AI describes systems capable of acquiring knowledge and applying information to solve problems". AI-enabled tools, in turn, are systems that include data and components that implement algorithms that mimic learning and problem solving, have inherently different characteristics than software systems alone (Horneman, et al, 2019).

The events we record may be security-related, so an AI-enabled system combines the concepts of enabled systems with the processes of Computer Science and statistical algorithms of Artificial Intelligence. In turn, this technology infrastructure is configured to acquire, process and analyze data in order to make intelligent decisions, make predictions and automate tasks (Roymo, 2023). That said, it can be said that an Artificial Intelligence (AI) enabled system (AI Enable System) refers to a technology infrastructure configured with the necessary components, software and resources to leverage and utilize AI capabilities in its functions and operations. These systems are designed to apply AI algorithms and techniques to simulate and enhance human intelligence, acquiring knowledge and applying information to solve problems and address software quality assurance.

2.5 ISO 25010 and ISO 25059

Agencia (2021), states that "software quality is directly related to the fulfillment of the requirements formulated by the user, so that if a program does not meet any of these, it is a low quality software". On the other hand, Perez (2019), states that "Quality in software development is a crucial aspect for any successful software project. The goal of any project is to ensure the quality of the results and satisfy the needs of the customer and other stakeholders. To achieve this, it is important to plan for delivery and process quality."

In conclusion, software quality is considered a fundamental concept for the success of any software development project. It refers to the degree to which the software meets the requirements established by the customer and other stakeholders, and its ability to satisfy their needs (Rodriguez, 2016). In turn, software quality is an essential factor for the success of any software development project, through compliance with ISO 25010 and ISO 25059 standards (ISO 25000, 2022).

According to the ISO 25010 (2022) page, the ISO 25010 standard represents "a quality model, which is fundamental to product evaluation because it establishes the core system on which it is based". They also point out that "this model determines the quality characteristics that are considered when evaluating the properties of a given software product whose quality can be interpreted as the degree to which the product satisfies the requirements of its users, thus providing value. It is precisely these requirements (functionality, performance, security, maintainability, etc.) that are represented in the quality model, which categorizes product quality into characteristics and sub-characteristics."

On the other hand, Ormeńo (2019), indicates that "ISO 25010 is a family of standards aimed at creating a common framework for assessing software product quality", composed of 8 characteristics and 31 sub-characteristics that relate to the static properties of software and the dynamic properties of the computer system. The model is applicable to both computer systems and software products".

On the other hand, according to the ISO 25000 (2022) page, the ISO 25059 standard "is an extension of the ISO/IEC 25010 quality model that focuses on addressing additional aspects that arise in AI systems. As organizations incorporate more and more AI systems, it has become necessary to adapt the existing quality model by incorporating the specific characteristics of these systems." According to, iTeh Standard Preview (2023), this standard is "a document that describes a quality model for AI systems and is an application-specific extension of the SQuaRE standards. The characteristics and subcharacteristics detailed in the model provide a consistent language and terminology for specifying, measuring, and evaluating the quality of AI systems" and are used as a set of quality criteria against which established quality requirements can be compared to determine their completeness (Iso/iec 25059:2023, 2023).

It is concluded then that ISO 25010 and ISO 25059 play a fundamental role in the formulation of a quality model for AI-enabled tools, which will establish and prescribe the characteristics and subcharacteristics that must be present in AI-enabled tools. ISO 25050 has new characteristics and each of them has sub-characteristics. The first characteristic is Functionality, which has adaptability, functional relevance, functional correctness and functional completeness as subcharacteristics. The Security characteristic has intervisibility, non-repudiation, authenticity, accountability, integrity and confidentiality. Reliability has 5 sub-characteristics which are: resilience, robustness, maturity, availability and fault tolerance (Iso/iec 25059:2023, 2023).

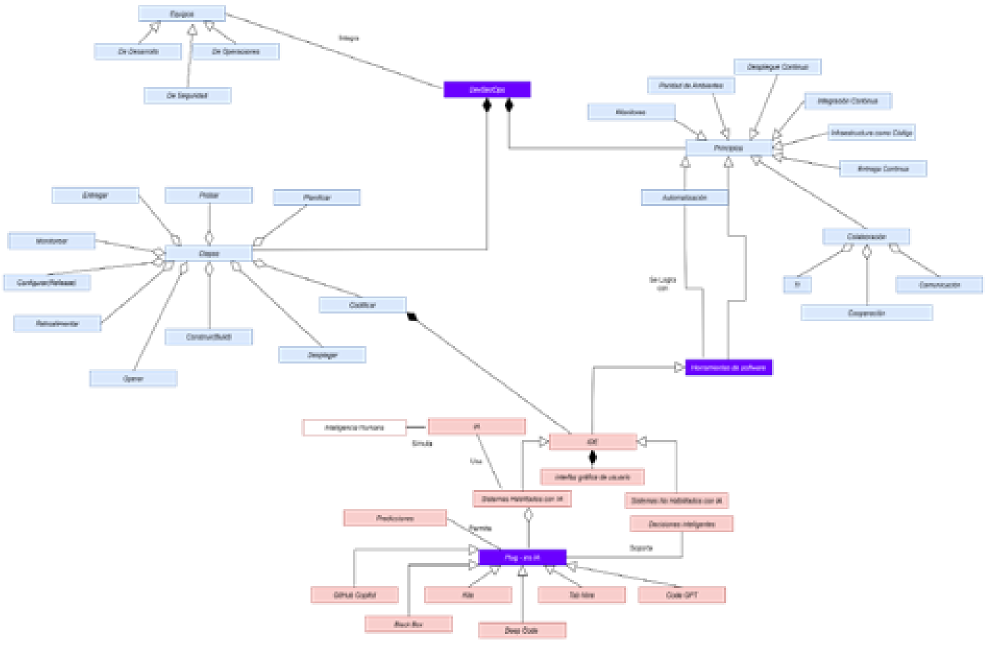
In turn, the Interaction Capability has recognizability of suitability, operability, accessibility, protection against user errors, aesthetics, transparency and user controllability. Protection has five sub-characteristics, which are: operational restriction, risk identification, failure protection, hazard warning and secure integration. Another characteristic is Maintainability, which has the capacity to be tested, analysis capacity, confidentiality, modularity and capacity to be modified. There is also Portability which as a sub-characteristic has adaptability, ease of installation, ability to be replaced and scalability. Efficiency is another characteristic with temporal behavior, resource utilization and capacity. As a last characteristic, there is Compatibility which has interoperability and coexistence. These characteristics and sub-characteristics provide a complete evaluation framework to measure and evaluate the quality of a software product, allowing to ensure that it meets the established requirements (ISO 25000, 2022 and ISO 25010, 2022).

According to the above, the explained topics could be presented as a domain model expressed visually with a UML class diagram.

3. Domain Model

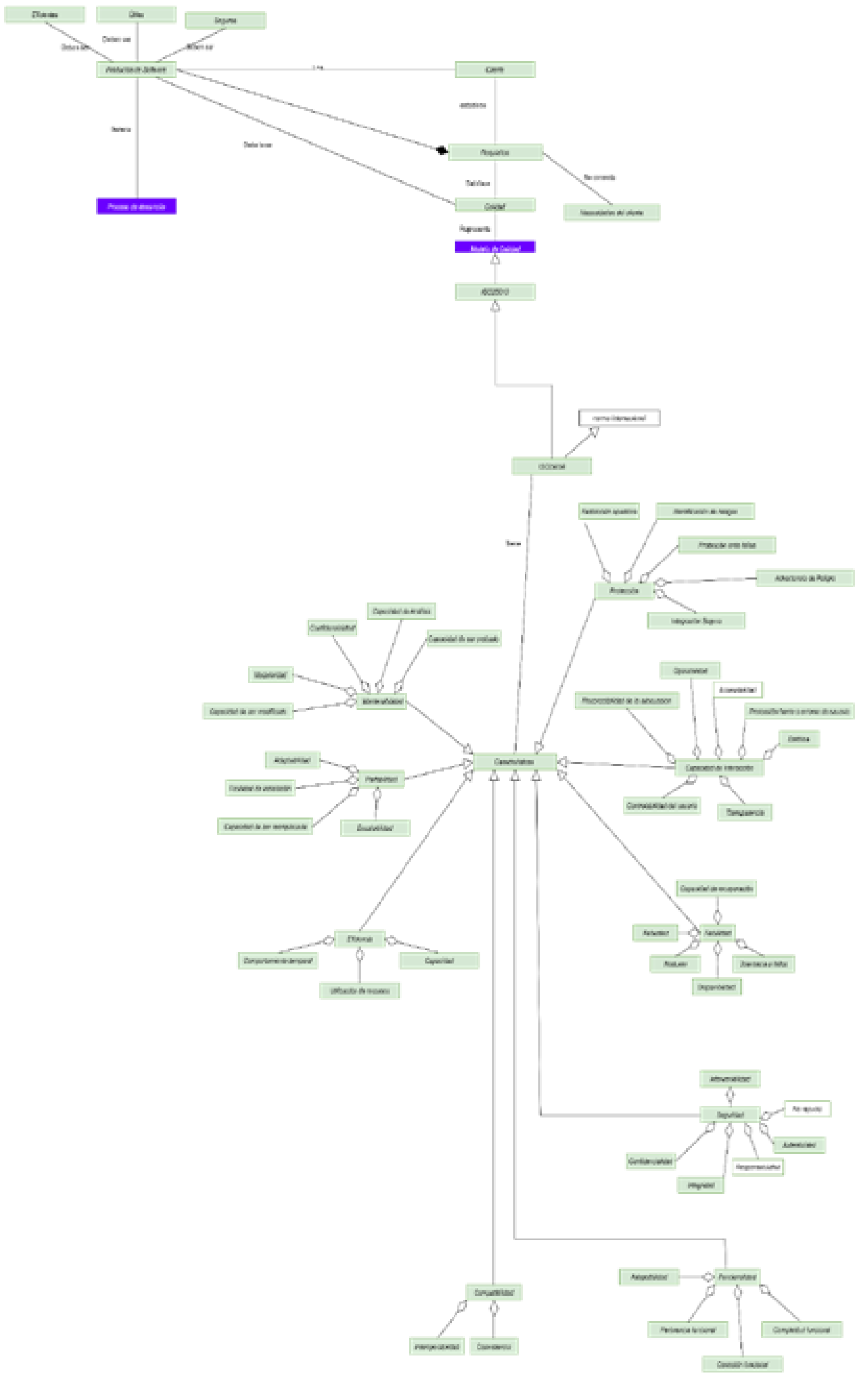
Therefore, it can be seen below in the Domain Model proposed and elaborated with a Class Diagram expressed in the standard UML language, that it highlights three worlds such as DevSecOps, ISO and Quality Models, and AI-enabled Tools-Plugins. These three worlds are shown below, identifying the concepts reviewed in the literature and the proposed relationships; note that the concepts that integrate these three worlds (the dark purple classes) were also identified (Figures 1, 2 and 3). These classes link the three worlds described.

Figure 1: First DevSecOps world



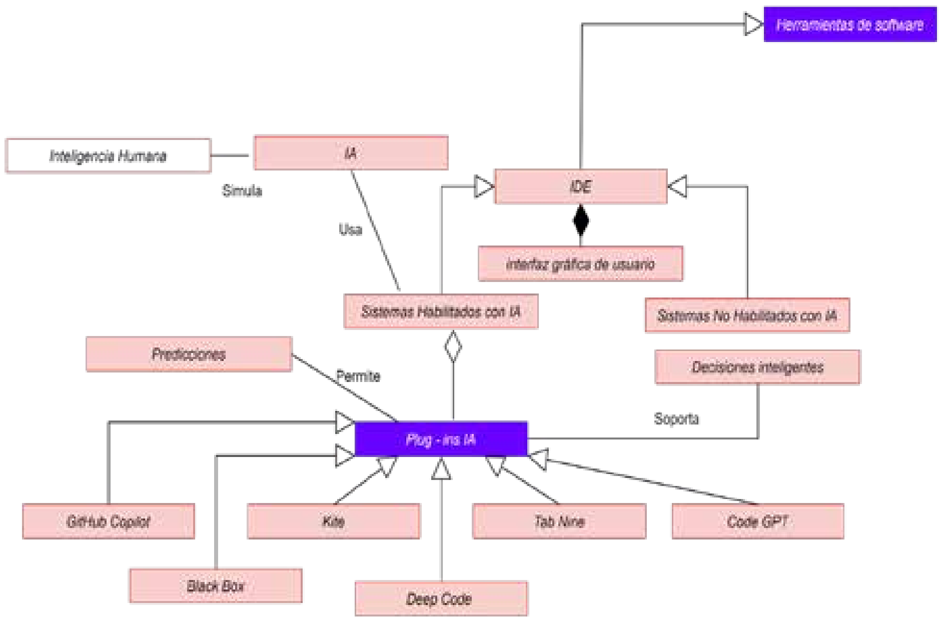
The binding concepts found here are: DevSecOps and Software Tools, through the Automation principle.

Figure 2: Second World ISO and Quality Models



Here the binding concepts of this world with DevSecOps are: Development Process and Quality Model, since DevSecOps is a philosophy about the development process.

Figure 3: Third World Enabled Tools with AI-Plugins



This world is directly linked to DevSecOps through the Software Tools concept.

The domain model is then the integration of these three worlds, given its complexity, it was presented in three parts; however, thanks to it, the complexity of the subject treated is received, and relationships are already identified that allow to specify the characteristics and sub-features that should be present in the IDE type AI-enabled software tools. A fundamental part of this type of tools are the AI-enabled *plug-ins.*

4. Plug - Ins

According to Correa (2024), AI *plug-ins* for IDEs have become increasingly important tools for programmers. These *plug-ins* harness the power of AI to improve developer productivity, code quality, security, and the overall development experience. Therefore, in the field of software product development, there are several *plug-ins* that are the most common, currently in use and interoperate with different IDEs such as Visual Studio Code, Eclipse, Sublime Text and Atom.

In reference to the help for coding and more efficient code development by programmers, Github Copilot, according to Fernandez (2023), is a help system created by GitHub that allows you to write code in real time, since as the programmer writes the code, GitHub Copilot suggests the code related to what is being developed. On the other hand, Fernandez (2024), defines Black Box as an AI tool that helps to find and use code snippets efficiently. Thus, while the programmer is coding, he will have suggestions to complete tasks that are being performed.

In turn, Fernandez et al. (2019) argues that Kite is another *plug-in* that stands out in the coding process, as it is an AI add-on that is placed in the code editor which has the auto-completion function like those mentioned above. Additionally, TabNine like GitHub Copilot and Kite works by generating code or offering code suggestions in real time, performing code context analysis and providing auto-completion.

To close, it is of utmost importance to analyze security, errors and vulnerabilities that may arise during development, and that is when Deep Code, according to Correa (2024), stands out, since it allows analyzing the code to improve the quality and security of the software by performing an exhaustive analysis using continuous learning techniques to detect vulnerabilities, errors or bad practices in the code. Finally, Stork (2024) states that Code GPT aims to enhance the ability to code and streamline the workflow, asking questions about problems with the code being worked on.

The following link shows a summary of the plugins analyzed so far, in search of which features and subfeatures according to ISO 29059 are present in them.

https://doi.org/10.6084/m9.figshare.25648779.v2

After analyzing the Table, a trend in the presence of the Interoperability subfeature, within the compatibility feature, is observed in most of the evaluated *plugins*, such as Github Copilot, Kite, DeepCode, TabNine and Code GPT. However, the only *plugin* that does not present this subfeature is Black Box. In addition, it is observed that for the Functionality feature each of them has different associated sub-features. Github Copilot, Kite, Code GPT and TabNine plugins have the Functional Relevance subfeature within the Functionality feature. On the other hand, Black Box, Kite, DeepCode, Code GPT and TabNine have the Functional Adaptability sub-characteristic. The Code GPT plugin and Kite also have the Functional Correctness sub-characteristic.

Regarding Interaction Capability, it is observed: Kite and TabNine have the Fitness Recognizability subfeature, while DeepCode has the Operability and User Controllability subfeatures. Finally, the Reliability feature is present only in the Code GPT *plugin.* In support of this, it can be said that the evaluated *plugins* differ in terms of the subfeatures present within each of the evaluated features, which evidences the different capabilities and approaches of each of them.

5. Conclusions

Among the conclusions, the following are presented:

A bibliographic review of the context related to AI-enabled IDE-type tools and the DevSecOps development process was conducted, taking into account the contribution of a quality model inspired by the ISO 25059 standards for the estimation of their quality.

Software development is a complex process that requires the collaboration of diverse teams and the integration of multiple practices. In this context, the adoption of AI-enabled tools within the DeSecOps framework emerges as a promising strategy to optimize the efficiency and quality of developed software. DevSecOps, as a philosophy, integrates security into the software lifecycle, promoting collaboration between development, operation and security teams. AI-enabled tools, meanwhile, offer a range of possibilities to automate tasks, improve efficiency and optimize software quality. ISO 25059 provides a solid framework for assessing the quality of AI-enabled IDE-type tools.

A domain model is proposed, elaborated with the UML language according to the notation of class diagrams, which allows to interpret, analyze and reflect on the context of AI-enabled IDE-type tools under the DevSecOps philosophy in the search for a model to estimate the quality of this type of AI-enabled tools, as well as the IDEs.

Analysis of the AI plugins table reveals a variety of capabilities and approaches. Interoperability is present in most of the plugins, allowing their integration with other development tools. Functionality offers sub-features such as functional relevance, functional adaptability and functional correctness. Interoperability, also present, facilitates the recognition of the adequacy of suggestions, as well as their operability and user controllability. Finally, reliability offers features that guarantee the reliability of the suggestions. The choice of the ideal *plug-in* will depend on the specific needs of the project, considering the various capabilities and approaches that each one offers.

6. Recommendations

Among the recommendations, the following are proposed:

Conduct a more comprehensive review of *plug-ins* and SDI-type AI-enabled tools in order to deepen and propose a quality estimation model for them.

Operationalize through the proposal of a set of metrics that allow the developer to evaluate the different tools according to the characteristics and sub-characteristics present in this quality model.

7. Bibliographic References

AWS (2023). What is DevSecOps? https://aws.amazon.com/es/what-is/devsecops/#:~:text=building%20the%20software.-,What%20does%20DevSecOps%20stand%20for%3F,they%20are%20building%20software%20applications.

AWS (2023). What is an integrated development environment (IDE). Amazon.com. https://aws.amazon.com/es/what-is/ide/

Agency, F. (2021, August 1). Software Quality. eHealth. https://saludelectronica.com/calidad-del-software/

Correa, J. (2024, April 15). AI for developers: A complete guide to AI for software developers. Developer Blog. https://developero.io/ai-for-developers

Datascientest (2022, September 2). IDE : What is an Integrated Development Environment? Training in science

Dynatrace (27, March 2024). DevSecOps-Development, security and operations https://www.dynatrace.com/monitoring/solutions/devsecops-1/?utm_source=google&utm_medium=cpc&utm_term=devsecops&utm_campaign=us-appsec-application-security&utm_content=none&utm_campaign_id=11810003001&gclsrc=aw.ds&gad_source=1&gclid=Cj0KCQjwztOwBhD7ARIsAPDKnkChL4ZftHqjOpxPwYUUoA5pBUaLT4EGIpRYbsLjsXFp-BN0MQWqfh7caAmRVEALw_wcB&mrasn=1160682.1439354.4uUpreyi

Fernandez, E., Pardo, L., & Ferzzola, M. (2019, February 8). Kite: AI-based code autocompletion. Neoteo.com. https://www.neoteo.com/kite-autocompletado-de-codigo-basado-en-ai/.

Fernández, Y. (2024, March 27). Blackbox AI: what is it and what tools does this artificial intelligence that helps you program by completing code offer. Xataka.com; Xataka Basics. https://www.xataka.com/basics/blackbox-que-que-herramientas-ofrece-esta-inteligencia-artificial-que-te-ayuda-a-programar-completando-codigo.

Fernández, Y. (2023, February 16). What is GitHub's Copilot and how this artificial intelligence that helps you program works. Xataka.com; Xataka Basics. https://www.xataka.com/basics/que-copilot-github-como-funciona-esta-inteligencia-artificial-que-te-ayuda-a-programar.

Horneman. A, Mellinger A and Ozkaya I. (2019). AI Engineering: 11 Foundational Practices. Pittsburgh: Carnegie Mellon University Software Engineering Institute, 2019. https://resources.sei.cmu.edu/asset _files/WhitePaper/2019_019_001 _634648.pdf.

Ibm.com. (n/d) What is software development?. https://www.ibm.com/es-es/topics/software-development

Ingeno, J. (2023). Software Architects Handbook: Become a successful software architect by implementing effective architecture concepts. Packt Publishing.

ISO 25000 (2022). ISO/IEC 25059. https://iso25000.com/index.php/normas-iso-25000/iso-25059

ISO 25010. (2022). Iso25000.com. https://iso25000.com/index.php/normas-iso-25000/iso-25010

Iso/iec 25059:2023. (2023). ISO. https://www.iso.org/standard/80655.html

iTeh Standard Preview (2023). Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI systems. https://cdn.standards.iteh.ai/samples/80655/168addf09e0a4d8181b9172dc7404fab/ISO-IEC-25059-2023.pdf

Jardiel, L. (2015, January 27). The Domain Model. Blogspot.com. https://modeladodesoftware.blogspot.com/2015/01/9-el-modelo-de-dominio.html

Maida, E and Paciencia, J. (2015). Software development methodologies. https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf

ITE Solutions by ITE Corp. (2022, November 9) What is the software development process? Linkedin.com. https://es.linkedin.com/pulse/cu%C3%A1l-es-el-proceso-de-desarrollo-software-itesolucionesmx

Ormeńo, N. (2019, May 15). ISO 25010 and Software Development. https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341.

Perez, J. B. (2019). 3digits - We have a solution - Information Technologies. https://www.3digits.es/blog/gestion-de-la-calidad-en-el-desarrollo-de-software.html

Prieto, E. (2023, November 16). What Are The Stages Of Software Development?. https://global.tiffin.edu/noticias/cuales-son-las-etapas-del-desarrollo-de-software

Red Hat (2024). Open source artificial intelligence in your favor. https://www.redhat.com/es/products/ai

Redhat.com (2013, July 31). What is an IDE and what is it for. https://www.redhat.com/es/topics/middleware/what-is-ide.

RedHat.com (2023, March 15). What is DevSecOps. https://www.redhat.com/es/topics/devops/what-is-devsecops.

Rodríguez, A. (2016, March 23). How to evaluate the quality of a SDI. Blogspot.com. https://blog-idee.blogspot.com/2016/03/como-se-evalua-la-calidad-de-una-ide.html

Roymo, D. (2023, July 31). AI and process automation: increasing efficiency and reducing costs. GAMCO, SL. https://gamco.es/ia-automatizacion-de-procesos-eficiencia-reducir-costes/ https://www.stork.ai/es/ai-tools/codegpt-efbb2

Techopedia.com (2023). Importance of Software Development in Technology. https://www.techopedia.com/

Information-technologies. (2018) Lean Method of Software Development. https://www.tecnologias-informacion.com/metodo-lean.html