CDInsight

An AI-Driven Tool for Software Comprehension

Date: 19 January 2025

Wong Tsz Chun 12917491 Computer Science
Chan Wai Kwan 13339280 Computer Science
Supervisor: Dr. Ndudi Okechukwu EZEAMUZIE

Abstract

This report addresses the challenges faced by companies due to inadequate software documentation and ineffective process tracking, which often lead to operational inefficiencies and significant project delays. The project was initiated to tackle these issues by developing a software solution designed to streamline the software development process, enhance documentation and facilitate understanding of code bases for new developers. The aim of this report is to outline the research, objectives and methodologies used to create a practical tool that leverages artificial intelligence to automate documentation, analyze code structures, and track development workflows. By focusing on real-world applications, this project aims to provide immediate and long-term benefits to software development teams, including improved on-boarding efficiency, cost reduction and enhanced collaboration.

Chapter 1. Introduction

Introduction

Some companies lack well-constructed documentation for their software projects, regardless of their scale. When the original developers leave or the software is handed over to new employees, it becomes extremely difficult to understand the existing code. This complexity results in time-consuming and costly updates or revamps.

For example, in 2013, the United States government tried to develop a website called Healthcare.gov for the Affordable Care Act. However, due to a lack of clear documentation and understanding of the underlying code, the original developer left the project and a new team worked hard to diagnose and fix the problem, eventually finding another contractor to fix the problem, resulting in significant delays in access for many users. (Lee & Brumer, 2017)[1] Similarly, Knight, the largest stock trader in the United States, updated their automated routing system SMARS in 2012 and used the code "Power Peg" which had not been used for 8 years. However, the developers forgot to apply the code to every server, and no other Technicians reviewed the deployment. This resulted in one server processing orders abnormally and led to a $440 million loss, highlighting the risks of inadequate documentation (U.S. Securities and Exchange Commission, 2013)[2] These incidents demonstrate that poor process tracking and insufficient documentation can result in severe operational and financial consequences.

The software development process encompasses more than just documentation. It includes version control to manage code changes, automated testing to ensure stability, and continuous integration practices that streamline deployment. Effective bug tracking and issue management are also essential to maintaining project health over time. In addition, fostering team collaboration and implementing structured code review procedures are critical for preventing errors and ensuring knowledge is preserved during transitions. Without these practices, software teams face increased risks of operational disruption and financial loss. Therefore, addressing inefficiencies in the software development process through better documentation, process tracking, and collaboration tools is essential to minimizing risks, reducing costs and ensuring the long-term success of software projects.

[1] Lee, G., & Brumer, J. (2017). Government Software Projects: Lessons Learned from Healthcare.gov Project. The Business of Government, 1-7.
https://www.businessofgovernment.org/sites/default/files/Viewpoints%20Dr%20Gwanhoo%20Lee.pdf

[2] U.S. Securities and Exchange Commission (2013). SECURITIES EXCHANGE ACT OF 1934 Release No. 70694 / October 16, 2013.
https://www.sec.gov/files/litigation/admin/2013/34-70694.pdf

Project Aim

The aim of the project is to create a software solution that facilitates the tracking and understanding of software development processes. This tool will enable new developers to quickly catch up with existing codebases, thereby reducing the time and cost associated with updates and revamps. We will only use existing resources to give solutions instead of creating. We will not go too deep about the theory of AI, but we will take advantage of AI, and create real application and solutions.

Project Objectives

The software will achieve the following objectives:

1. Design the model and system workflow diagrams.
2. Design a user interface that is viewable and easy to use for developers.
3. Develop a method to analyze and decompose existing code structures to design databases.
4. Reconstruct information from database into secure data
5. Implement artificial intelligence to analyze code and documents.
6. Implement some functions that can automatically collect and organize relevant information from the code base. (e.g. Catch-Up, Development Suggestions. . .)
7. Evaluate the effectiveness of software solutions in terms of time before and after new developers use the tool.

Value Propositions

This project can provide the following benefits:
Immediate values of benefits

- **Enhanced On-boarding Efficiency:** New developers can easily understand existing code bases, reducing learning and research time and enabling faster system processing.
- **Reduced Time and Costs:** It simplifies documentation and analysis of legacy code, reducing the time and cost of updates and retrofits. Reduces the number of tokens required to be processed by AI, resulting in cost savings.
- **Improved Code Quality:** Using artificial intelligence to analyze code and files can help to identify hidden problems early and reduce the possibility of errors.

Ripple Effect

- **Long-term knowledge preservation:** This software helps to maintain project continuity by effectively documenting code and processes, ensuring that critical knowledge can be retained even if the original developer leaves.
- **Scalability:** Since new developers can quickly adapt to existing projects, companies can expand their teams faster and more efficiently, resulting in faster product development and deployment.
- **Enhancing Teamwork:** Since team members don't need to spend time on legacy code, they can focus more on collaboration and innovation, solve problems efficiently and develop more creative solutions.

Chapter 2. Background or Literature Review

Phases of System Development Life Cycle (SDLC)

Software development is a structured process involving designing, coding, testing, and maintaining software applications. This process is commonly referred to as the Software Development Lifecycle (SDLC). First, software development needs to understand the requirements of the stakeholders, such as functional or non-functional requirements. (OŔegan, 2010)[3] These requirements define the scope and objectives of what developers need to build, forming the foundation for the entire project.

The next phase is to involve analyzing and designing workflows to meet stakeholder needs. This stage often includes creating various Unified Modeling Language (UML) diagrams, such as use case diagrams, sequence diagrams and system diagrams. UML diagrams provide a comprehensive framework for structural and behavioral modeling, supporting the early stages of process design and visualization. (Jäger, D. et al., 1999)[4] These diagrams serve as essential deliverables, offering clear guidelines that help developers align on system functionality and design.

The coding and implementation phase transforms the design and workflow into functional software. This phase is often the longest in SDLC as developers write code to build the system using programming languages suited to the project's requirements. The development process typically involves breaking down the system into smaller, manageable units, which are coded individually and later integrated during subsequent phases. (Shylesh, 2017)[5] Following the completion of coding, the software enters the testing phase to validate its functionality and ensure it meets the defined requirements.

During the testing phase, developers deploy the software in a testing environment where the testing team evaluates the functionality of the entire system. Testing is a critical component of the SDLC as it identifies bugs, vulnerabilities and areas for improvement. It ensures the quality of the software by assessing its visibility, security, performance and usability. (Jindal, 2016)[6] When bugs or issues are discovered, testers communicate their findings to the development team, which then works to resolve them. The updated software is then retested to confirm that bugs have been fixed and no new problems have been introduced. This iterative process continues until the system meets the defined quality criteria. Finally, the software undergoes acceptance testing and is deployed to clients or end users to ensure that it meets their needs and expectations before final delivery.

[3]OŔegan, G. (2010). Introduction to software process improvement. Springer Science & Business Media. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2efe4d840631ebf026fede741e85195e36f8b134

[4]Jäger, D., Schleicher, A., & Westfechtel, B. (1999). Using UML for software process modeling. ACM SIGSOFT Software Engineering Notes, 24(6), 91-108.https://dl.acm.org/doi/pdf/10.1145/318774.318788

[5]Shylesh, S. (2017, April). A study of software development life cycle process models. In National Conference on Reinventing Opportunities in Management, IT, and Social Sciences (pp. 534-541). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2988291

[6]Jindal, T. (2016). Importance of Testing in SDLC. International Journal of Engineering and Applied Computer Science (IJEACS), 1(02), 54-56. https://www.researchgate.net/profile/Ijeacs-Uk/publication/312041152_Importance_of_Testing_in_SDLC/links/586bf4c508ae329d6212176b/Importance-of-Testing-in-SDLC.pdf

When the software is ready to be delivered to end users, it enters the deployment phase. This phase ensures that the software is properly installed, configured, and operates as intended within the production environment. Deployment is a critical post-production activity, often performed for or by software customers, during which all customer-specific customizations and configurations are finalized. (Dearle, 2007)[7] Depending on the complexity of the software, deployment can involve several steps, including preparing the production environment, migrating data, and conducting final verifications.

The maintenance phase is the final stage of the SDLC, where the software is fully operational and in use by end users. During this phase, teams monitor the system closely to address any unexpected issues, bugs, or user feedback that may arise. (Radack, 2009)[8] Maintenance ensures the software continues to function as intended in a real-world environment. This phase also involves updating the software to adapt to changes in the user environment, business needs, or emerging technologies. Updates may include enhancements to features, performance improvements or ensuring compatibility with evolving platforms and systems.

[7]Dearle, A. (2007, May). Software deployment, past, present and future. In Future of Software Engineering (FOSE'07) (pp. 269-284). IEEE. https://www.cs.tufts.edu/comp/250SA/papers/dearle2007.pdf

[8]Radack, S. (2009). The system development life cycle (sdlc) (No. ITL Bulletin April 2009 (Withdrawn)). National Institute of Standards and Technology. https://csrc.nist.gov/csrc/media/publications/shared/documents/itl-bulletin/itlbul2009-04.pdf

Overview of SDLC models

Waterfall Model

Waterfall Model is the oldest and well-know model in SDLC. It is the original standard for most software development that follows every step of the SDLC process, which is requirements analysis, design, coding, testing, and maintenance. According to Alshamrani and Bahattab (2015), this are ideal for quality control due to the extensive documentation and planning required.[9] This linear structure makes it easy to understand and manage, however it also has some limitations.

Advantages

The significant benefit of the Waterfall Model is its simplicity. Pargaonkar (2023) highlights that the Waterfall Model follows every phase of the SDLC with each phase building on information collected in the previous one.[10] This logical flow ensures a clear direction for the project, enabling teams to focus on one phase at a time without overlapping concerns. This step-by-step approach enables stakeholders to easily track progress and evaluate deliverables at each stage, promoting transparency and accountability throughout the process.

In addition, the framework ensures that all tasks are organized and planned, thus reducing confusion among developers and other team members. By clearly defining roles and responsibilities in each phase, the Waterfall Model minimizes communication errors and ensures coordination among different contributors.

Disadvantages

The disadvantages are also obvious, using this model is inflexible. Adenowo and Adenowo (2013) mention that processes and data are usually separated in the Waterfall Model, so if the data is to be modified, the code must also be changed. Once something has to be changed, the entire process needs to be modified.[11]This model is unsuitable for complex or agile environments because it struggles to accommodate rapidly evolving requirements and requires substantial effort to adapt to changes in requirements or scope after the project launch.

In addition, the client will only be involved at the requirement phase and at the end of the acceptance test, which may cause client feedback delays. Alshamrani and Bahattab (2015) state that clients often do not fully understand what they actually need and have little opportunity to preview the system during development when using this Waterfall Model. Because clients cannot clearly understand what they need, the expectations

[9]Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. International Journal of Computer Science Issues (IJCSI), 12(1), 106. https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model

[10]Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications (IJSRP), 13(08), 345-358.http://dx.doi.org/10.29322/IJSRP.13.08.2023.p14015

[11]Adenowo, A. A., & Adenowo, B. A. (2013). Software engineering methodologies: a review of the waterfall model and object-oriented approach. International Journal of Scientific & Engineering Research, 4(7), 427-434. https://www.researchgate.net/publication/344194737_Software_Engineering_Methodologies_A_Review_of_the_Waterfall_Model_and_Object-_Oriented_Approach

of clients and developers are inconsistent, which affects the progress of the project and leads to delays or even failure.

Agile Model

Agile Model is a model that adapts to rapidly changing requirements. In this model, developers and clients engage in close collaboration, face-to-face communication, and frequent delivery of new software versions. Erickson et al. (Ambler, 2001a, as cited in Erickson et al., 2005) states that the core principles of Agile Modeling (AM) include simplicity, incremental releases, staying on task, and producing a quality product.[12] Today, this model is particularly suitable for modern software development, especially for mobile applications and software that require rapid development cycles.

Advantages

Unlike the Waterfall Model, the Agile Model facilitates close collaboration between developers and customers, which helps to ensure that the final product aligns with client expectations and needs. Yahya and Maidin (2022) note that using this model can promote high-quality software development by focusing on client satisfaction through stakeholder engagement.[13] In the Agile Model, the project is divided into multiple sub-projects, and the finished product is presented to the client. If the sub-projects meet the client's requirements, developers can move on to the next stage. Therefore, using the Agile Model can minimize overall risk and enable the product to quickly adapt to changes.

In addition, the Agile Model can reduce both the time and cost of development. As mentioned earlier, the Agile Model is designed to meet the needs of rapid and frequent delivery, which can help to minimize overall risk. Raval and Rathod (2014) point out that one of the principles of the Agile manifesto is the frequent delivery of working software within weeks or months.[14] Gupta et al. (2022) mention that the root cause of many project failures is the lack of open communication and detailed documentation. [15] Without good communication, it becomes harder to evolve the application and requires much more effort in team coordination, ultimately leading to delayed software deployment. This delay increases both the time and cost of development.

[12]Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. Journal of Database Management (JDM), 16(4), 88-100. https://www.researchgate.net/profile/Keng-Siau-2/publication/220373708_Agile_Modeling_Agile_Software_Development_and_Extreme_Programming_The_State_of_Research/links/5984f29f458515605844f08d/Agile-Modeling-Agile-Software-Development-and-Extreme-Programming-The-State-of-Research.pdf?origin=journalDetail&_tp=eyJwYWdlIjoiam91cm5hbERldGFpbCJ9

[13]Yahya, N., & Maidin, S. S. (2022, September). The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team. In 2022 10th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-5). IEEE. https://ieeexplore.ieee.org/document/9936036

[14]Raval, R. R., & Rathod, H. M. (2014). Improvements in Agile model using hybrid theory for software development in software engineering. International Journal of Computer Applications, 90(16). https://research.ijcaonline.org/volume90/number16/pxc3894677.pdf

[15]Gupta, A., Poels, G., & Bera, P. (2022). Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. IEEE Access, 10, 119745-119766. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9945932

Disadvantages
The significant drawback is the limited documentation in the project, which means that there is a lack of comprehensive documentation when using the model. Gumiński et al. (2023) indicate that agile methods significantly reduce the amount of documentation and rely heavily on tacit knowledge, making this model unsuitable for highly stable projects.[16] Since the project is subject to constant changes in this model, each version of the document may undergo major revisions. If the developer does not have proper document management, it can easily lead to confusion during development.

Also, documentation in Agile focuses only on critical paths, meaning that it often lacks comprehensiveness. This can be a disadvantage for projects requiring detailed records for maintenance or regulatory compliance. For example, industries such as healthcare or finance often mandate comprehensive documentation to meet legal and regulatory standards. In such cases, the lack of detailed records can result in non-compliance, fines, or increased operational risks.

Core Deliverables in Software Engineering

Use Cases Diagrams
Use Case Diagrams are a fundamental component of the Unified Modeling Language (UML), widely used in software engineering to model system functionality from the user's perspective. According to von der von der Maßen and Lichter (2002), Use Cases are tools used to capture functional requirements for software systems, providing a way to specify the interaction between a particular software system and its environment.[17] Use Case Diagrams provide a high-level visual representation of how users, known as "actors," interact with the system to achieve specific goals. These diagrams help stakeholders understand system requirements, identify key functionalities and ensure alignment between development teams and client expectations.

Sequence Diagrams
Sequence Diagrams, a core element of UML, play a vital role in modeling the dynamic interactions between system components. Sequence Diagrams capture the time-ordered flow of messages exchanged between objects and actors to achieve a specific goal. Il-Yeol (2001) states that Sequence Diagrams focus on time sequencing or the order in which messages are sent and represent graphically.[18] The primary components of a Sequence Diagram include external users or systems (actors), system entities (objects) and interactions between actors and objects. Sequence Diagrams are particularly valuable in designing and understanding complex workflows, debugging system logic, and optimizing

[16]Gumiński, A., Dohn, K., & Oloyede, E. (2023). Advantages and disadvantages of traditional and agile methods in software development projects–case study. Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska, (188 Nowoczesność przemysłu i usług= Modernity of industry and services), 191-206. https://www.researchgate.net/publication/377763200_Advantages_and_disadvantages_of_traditional_and_agile_methods_in_software_development_projects_-_case_study

[17]T. von der Maßen, H. Lichter. (2002, September). Modeling variability by UML use case diagrams. In Proceedings of the International Workshop on Requirements Engineering for product lines (pp. 19-25). Citeseer. https://swc.rwth-aachen.de/docs/2002_PLE_Essen.pdf

[18]Il-Yeol, S. (2001, November). Developing sequence diagrams in UML. In International Conference on Conceptual Modeling (pp. 368-382). Berlin, Heidelberg: Springer Berlin Heidelberg. https://cci.drexel.edu/faculty/song/publications/p_ER2001-SQD.pdf

performance.

Class Diagrams
Class Diagrams are a component of UML and represent the static structure of a system. These diagrams provide the architecture of the system by illustrating system classes, properties, methods and the relationships between them. Berardi et al. (2005) indicate that class diagrams represent the domain of interest by modeling information through objects grouped into classes and the relationships that connect them.[19] The primary components of a Class Diagram include classes, represented as rectangles divided into sections for the class name, attributes and methods. In addition, the relationships between classes are depicted using lines and annotations, which specify associations, inheritance, aggregation, or composition. These relationships enable teams to model complex interactions and dependencies within the system.

Entity Relationship (ER) Diagrams
Entity Relationship (ER) Diagrams are graphical representations used to model data and its relationships within a system, which is essential in database design and provides a visual framework that helps teams understand how entities (objects or concepts) interact. According to Btoush and Hammad (2015), an ER data model is a high-level conceptual framework that represents information in terms of entities, attributes and relationships, specifically designed to streamline and support database design.[20] ER Diagrams consists of three main parts: entities, attributes, and relationships. Entities are represented by rectangles and represent objects or concepts within the domain. Attributes are represented by ellipses and define properties or characteristics of an entity. Relationships are represented by diamonds and illustrate how entities are connected. These diagrams are important for ensuring the logical structure of the database and help identify redundancies, inconsistencies, and dependencies.

[19]Berardi, D., Calvanese, D., & De Giacomo, G. (2005). Reasoning on UML class diagrams. Artificial intelligence, 168(1-2), 70-118. https://www.sciencedirect.com/science/article/pii/S0004370205000792

[20]Btoush, E. S., & Hammad, M. M. (2015). Generating ER diagrams from requirement specifications based on natural language processing. International Journal of Database Theory and Application, 8(2), 61-70. https://www.researchgate.net/profile/Eman-Btoush/publication/275952818_Generating_ER_Diagrams_from_Requirement_Specifications_Based_On_Natural_Language_Processing/links/554a878e0cf21ed21358e791/Generating-ER-Diagrams-from-Requirement-Specifications-Based-On-Natural-Language-Processing.pdf

Highlight of the proposed solution

Since the purpose of this project is to reduce development and debugging time and costs, our proposed solution will have the following features:

- **Use diverse resources:** Our tools focus on how to effectively solve time and cost issues, and leveraging existing resources such as Open AI, Claude AI, and MySQL is critical. In addition, we hope to automatically generate flowcharts through artificial intelligence, just like how program visualization tools create flowcharts.

- **Focus on security:** Security is a paramount concern in software development. Our proposed solution will only read the full source code and obtain some information such as class name and function name and hand them over to AI for analysis. It does not store any sensitive data such as API keys in the database and all data is only used to provide information to the AI.

- **Provide code suggestion and explanation in the form of business:** Since most of the time when using our tools is to deal with some legacy code, What makes it even harder is to understand the complicate business logic and different actors from the code when documentation are missed or incomplete. Often times the business objective changes without documentation, causing the incoherence and misleading between code and documentation. We want this application not only to provide code suggestions and explanations, but also to show the developer that what is going on with the current project. This is of great help in understanding operations and development. For this feature, we can refer to how Github Copilot provides code suggestions and how it works

Overall, the proposed solution not only reduces the time spent dealing with legacy code but also helps improve software development.

Chapter 3. Preliminary Methodology

Basic Idea

The core idea of the software is to generate reports and documentation, such as user diagrams, workflows, and the architecture of existing software. To define a milestone, the first step we want is to create a use case diagram from a scaled project with multiple folders and files with Python only. As the final result, we hope that Python, JavaScript and HTML will be supported Therefore, we have the following structure.

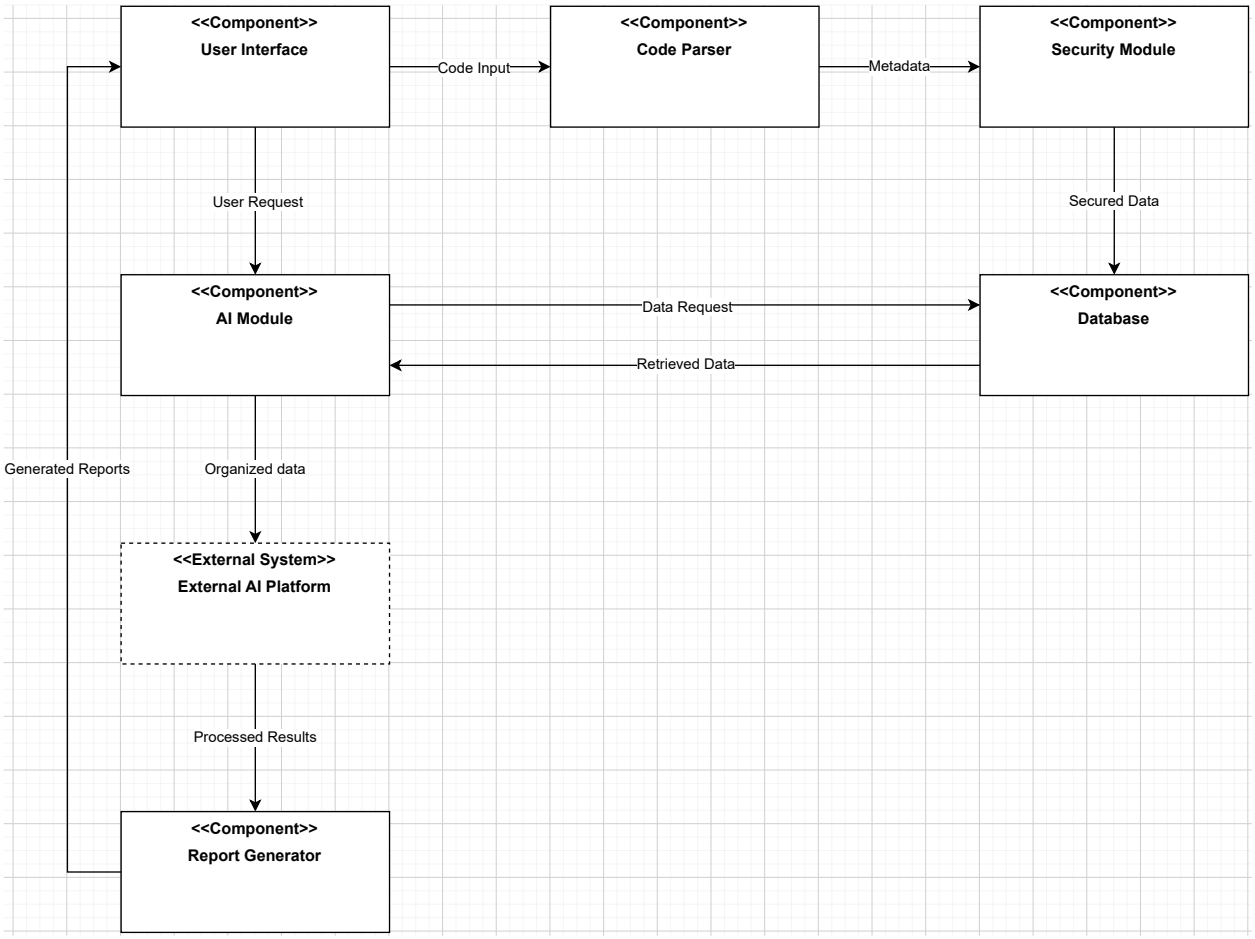

Figure 1: Component Diagram

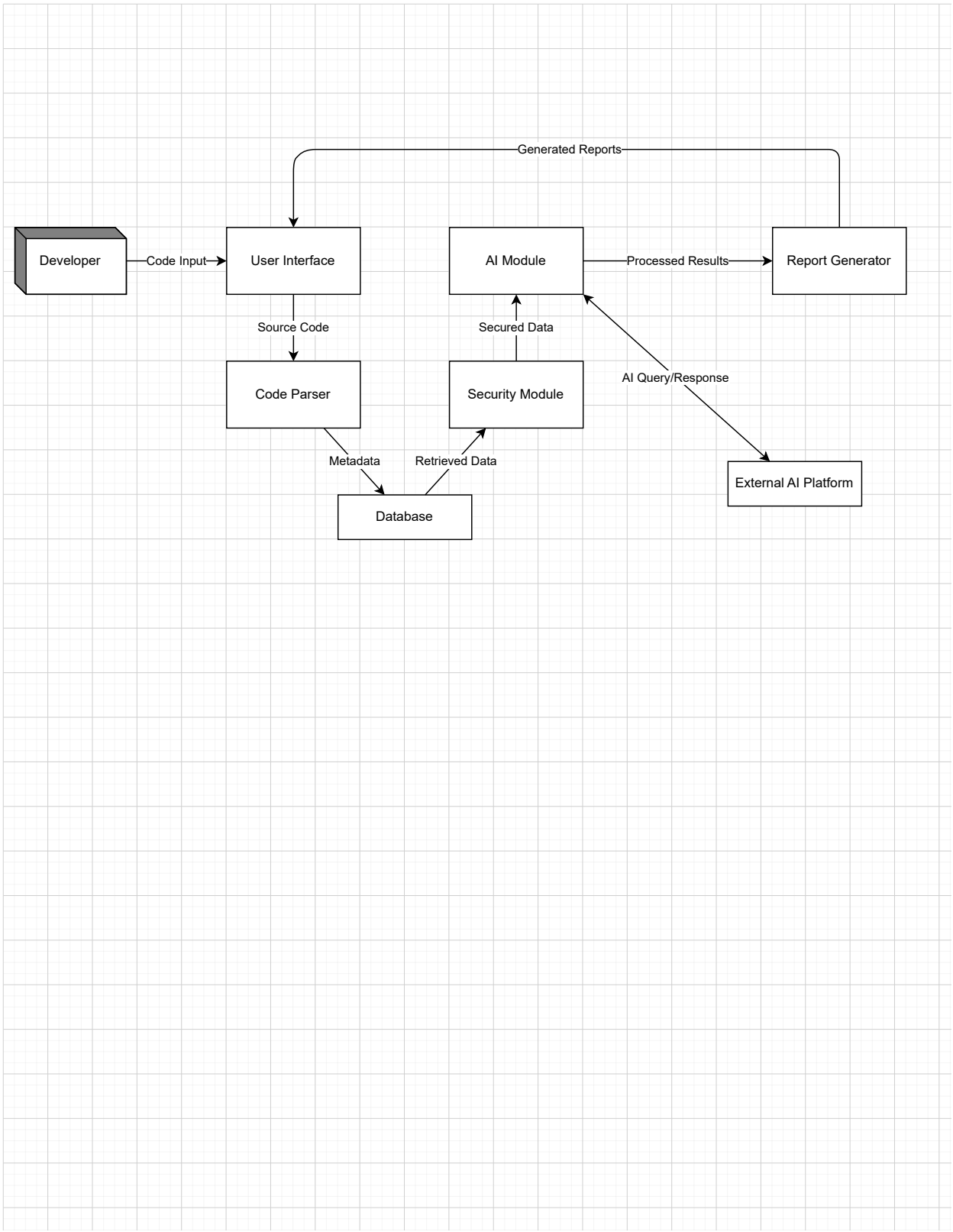

Figure 2: Data flow diagram

In order to generate the use case diagram, the work flow is as follow.

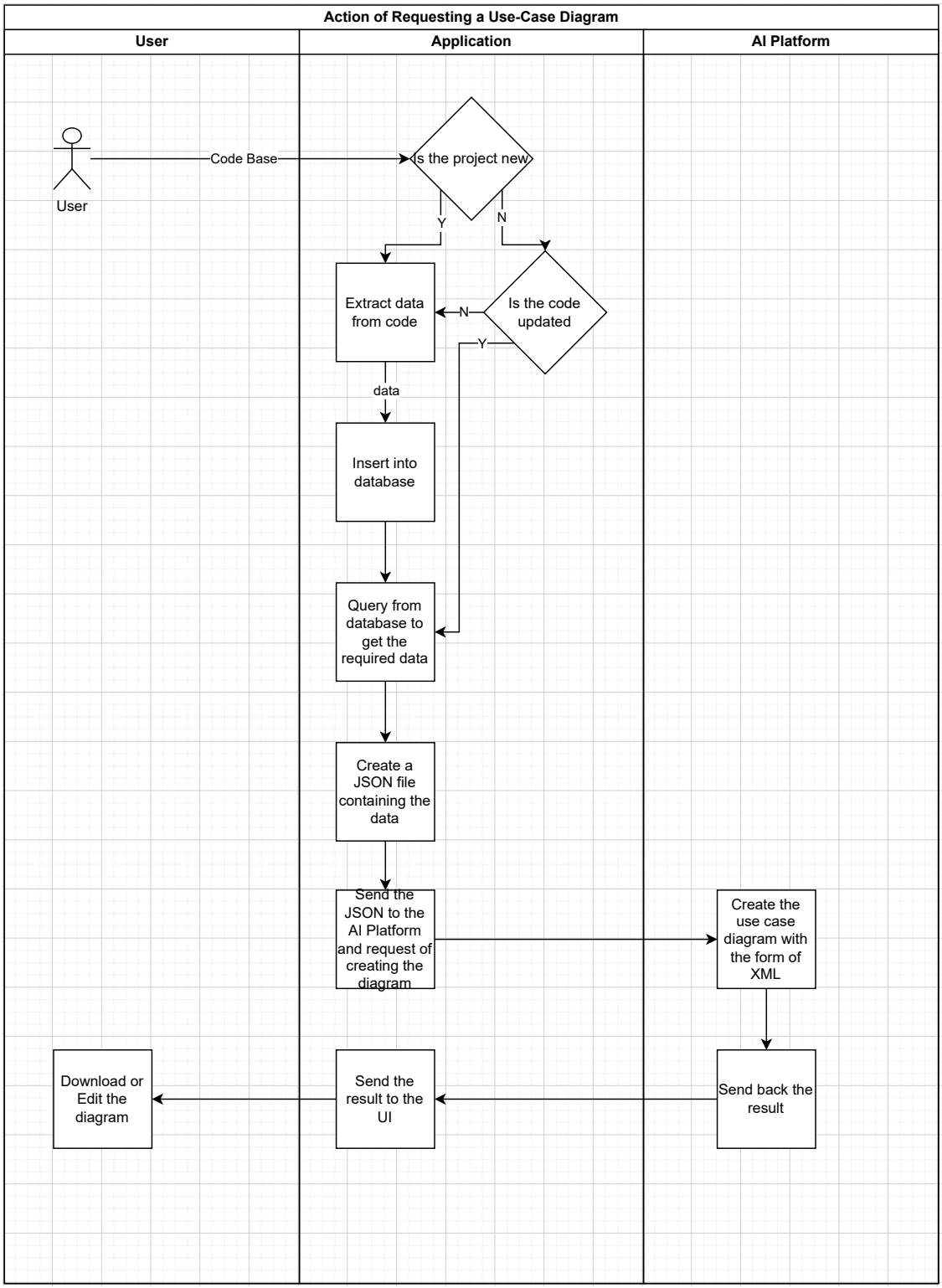

Figure 3: Workflow

User Interface

At the beginning of the project, we only want some essential function to receive user actions. A button to update codebase, a button to generate the use case diagram, a field to fill-in diagram output location, a box to show token used after each request, and a field to fill-in the root location of the source code. Specifically for the button to update codebase, once it is clicked, the source code will turn into metadata and send to the database. That means it will not automatically transfer to metadata if there are any changes.

Code Parser

To make the code into metadata, we will use the python library- astroid. It is a powerful tool to break down or bring up the scale for analysis, therefore it is able to show the entire picture macroly and microly to fulfill our requirement. We know that there is limitation making the some part of code could not be analysed such as complex matrix operations and variable type mutations during runtime.We want to find out if it is possible to outline the code and tell the user these code could not be analysed and the result could be affected.

Security Module

This is to prevent data sent to AI platform contain sensitive information, We will discuss it later

Database Design Principle

For the database, we have designed a possible and simple approach that allow the process of metadata transferred between code parser and the AI Module.

As we would like to solve problem not only technically, but also in the way of business communication with development department, so that developer could focus on development more. The database schema should include information as much as possible in which AI could build a complete use-case diagram along with the stakeholders.

To show the overview of the development process and the development result, even if a tiny relation between a variable and a function matter. For example, at retail industry, there are two variables, basePrice (decimal) and discount(float) with a function calculateFinalPrice(basePrice, discount), which return the price after discount. However, the developer misuse the data type of decimal and float, causing the final result has a $0.01 difference. After the discount, if there are over 1 million orders, there will be $10000 discrepancy. After that, there is still a function to calculate the taxes for the company, and the resulted report show inconsistencies. After investor or stock trading algorithm trade because of the revenue is higher than expected, it makes its competitors research on the reason of its high revenue and legal action could be made.

Although it is just in theory, but just few months ago, CrowdStrike , a cybersecurity company, making 500+ companies lost more than 5 billion dollars because of one single bug. If the bug could be found automatically through the relation between code, it could save 5 billion dollars.

Database Structure

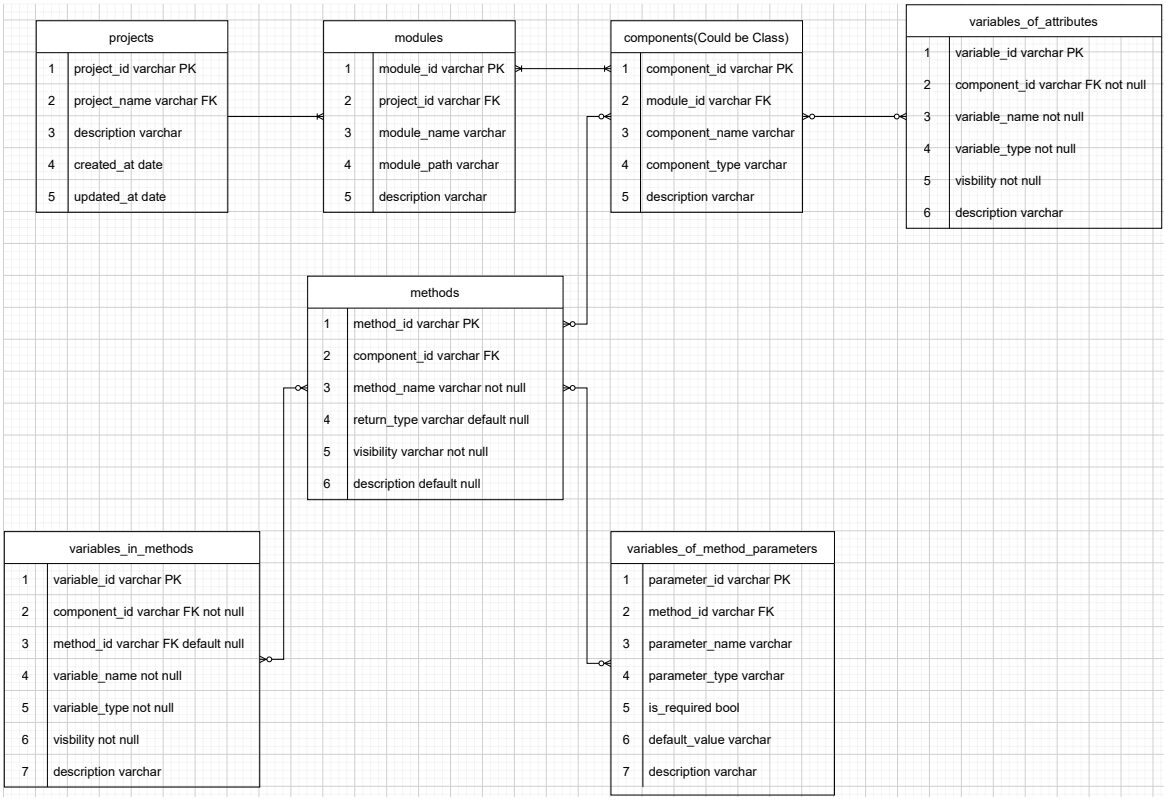

Figure 4: Database Design

The diagram shows a simplified version of the database which includes the core function.

Core Structure and Hierarchy

The database follows a hierarchical organization with three primary entities:

- **Projects:** Top-level container for all related components
- **Modules:** Logical groupings within projects (uniquely identified by project and path)
- **Components:** Specific implementation units (classes, interfaces, services, etc.)

Actor-Based System Modeling

- Actors table categorizes system users (human, system, external_service)
- UseCases table documents system behaviors with pre/post conditions
- Actions and Steps tables break down complex operations
- Relationship mapping through UseCaseActors and UseCaseComponents

Component-Level Documentation

- Methods table captures function signatures and visibility
- Endpoints table manages API endpoints and HTTP methods
- MethodParameters tracks detailed parameter information
- ComponentDependencies maps relationships between components

Variable and Data Flow Tracking

- Variables table stores definitions with scope and visibility
- VariableUsages tracks usage patterns (read/write/parameter/return)
- VariableFlow monitors data movement between methods
- VariableParameterMapping links variables to method parameters

Report Generator

After testing, it is difficult for AI to generate PDF at the moment, or the result is not satisfied. Therefore, we would like to have an other approach.

For claude, it is able to understand JSON which will be generated by the AI module and analyse the metadata from it. To output an ideal result, the best ways we found is to ask it exporting XML codes of the diagram. Not only it is able to do it, user is able to edit it with the generated report. For example, all diagrams are generated with this method. We truely believe this is the solution.

The responsibility of the report generator is to visualize the XML, provide an user interface to edit the report, and allow user to output it.

Security

Companies and governments are aware of the security concerns associated with AI, leading some to ban or limit its use. During development, developers might accidentally send confidential or sensitive data to AI, such as login methods, payment handling, and more.

To address this, we propose two parts with three approaches each for handling sensitive code, from database storage to code delivery.

From Database

1. **Approach 1:** Sensitive code is not stored. Instead, descriptions are used to ensure the code is impossible to leak.
2. **Approach 2:** Only directly sensitive content is hidden; other parts are stored.
3. **Approach 3:** All code is stored, including sensitive content.

From Code Delivery

1. **Approach 1:** Sensitive information is sent as descriptions.
2. **Approach 2:** Only sensitive parts are sent as descriptions; others may be sent as code.
3. **Approach 3:** All information is sent as actual code.

Definitions

- **Related:** Code that handles or processes sensitive data.
- **Unrelated:** Code that does not directly interact with sensitive data.
- **Sensitive Content:** Code that generates sensitive data.

Example

Consider three functions: `get_pwd(user)`, `login(user,pwd)`, and `redirect_main_page`.

- `get_pwd`**:** Returns the password for a user. It is the sensitive content, as it retrieves passwords.
- **login:** Checks if the provided password matches and returns a boolean. It is related to sensitive content,as it deals with password verification.
- `redirect_main_page`**:** Calls `login()` and redirects based on the result. It is unrelated to sensitive content, as it only processes boolean results.

Scenario

Different approach could be implement or change based on the user requirement. Based on the requirement, the data in database could be send as actual code.

- **Database Handling:**
  - **Approach 1:** Related and sensitive code (`get_pwd` and `login`) are stored as descriptions.
  - **Approach 2:** Related code (`login`0 is stored as actual code, but sensitive code (`get_pwd`) is not.
  - **Approach 3:** All code are stored as actual code.
- **Code Delivery:**
  - **Approach 1:** Related and sensitive code (`get_pwd` and `login`) are sent as descriptions.
  - **Approach 2:** Related code (`login`0 could be sent as actual code, but sensitive code (`get_pwd`) is not.
  - **Approach 3:** All code could be sent as actual code.

1 Prototyping

Currently, we successfully extract data from code to database, the next step is how we should query so that the AI could generate an accurate use case diagram.

2 Overview

The essential of the projects is to turn code into tiny information and then turning it back. When information are separated, we can choose the data we want and other left behind, therefore reducing energy consumed and making AI work efficiently. With this approach, it is much easier to add additional function as we only need to define what information will AI need.
It is user create information (code) , the software sends those information from the database to AI, and it is us to make those meaningless data into a group of information provided to user with actual applications and interface.With all these objective combine, this is the approach we proposed.

3 Requirement

Based on what the result we look forward, we need to have a local database, an AI platform. For the language, as AI is implemented, python is preferred. Some library like ast (A library to identify and get element with python files)

4 Limitation and Constraint

As different languages have different syntax, it is impossible to support all languages even with language parser. For now, python is focused.

Conclusion

This project aims to create a comprehensive software management solution that helps companies maintain and update their codebase efficiently. By leveraging AI and structured metadata, the software will provide valuable insights and recommendations while ensuring data security and cost-efficiency. The actual ways of fulfilling the objectives are not mentioned here and will be discussed soon. An example of "Move-On" is that when developers want to add something new, they can express the idea to the AI through this software, and the AI could give advice on what existing functions developers can call, reducing code redundancy and improving readability.

References

1. Lee, G., & Brumer, J. (2017). Managing Mission-Critical Government Software Projects: Lessons Learned from Healthcare. gov Project. The Business of Government, 1-7.
https://www.businessofgovernment.org/sites/default/files/Viewpoints%20Dr%20Gwanhoo%20Lee.pdf

2. U.S. Securities and Exchange Commission (2013). SECURITIES EXCHANGE ACT OF 1934 Release No. 70694 / October 16, 2013.
https://www.sec.gov/files/litigation/admin/2013/34-70694.pdf

3. OŔegan, G. (2010). Introduction to software process improvement. Springer Science & Business Media.
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2efe4d840631ebf026

4. Jäger, D., Schleicher, A., & Westfechtel, B. (1999). Using UML for software process modeling. ACM SIGSOFT Software Engineering Notes, 24(6), 91-108.
https://dl.acm.org/doi/pdf/10.1145/318774.318788

5. Shylesh, S. (2017, April). A study of software development life cycle process models. In National Conference on Reinventing Opportunities in Management, IT, and Social Sciences (pp. 534-541).
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2988291

6. Jindal, T. (2016). Importance of Testing in SDLC. International Journal of Engineering and Applied Computer Science (IJEACS), 1(02), 54-56.
https://www.researchgate.net/profile/Ijeacs-Uk/publication/312041152_Importance_of_Testing_in_SDLC/links/586bf4c508ae329d6212176b/Importance-of-Testi
pdf

7. Dearle, A. (2007, May). Software deployment, past, present and future. In Future of Software Engineering (FOSE'07) (pp. 269-284). IEEE.
https://www.cs.tufts.edu/comp/250SA/papers/dearle2007.pdf

8. Radack, S. (2009). The system development life cycle (sdlc) (No. ITL Bulletin April 2009 (With-drawn)). National Institute of Standards and Technology.
https://csrc.nist.gov/csrc/media/publications/shared/documents/itl-bulletin/itlbul2009-04.pdf

9. Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. International Journal of Computer Science Issues (IJCSI), 12(1), 106.
https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model

10. Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications (IJSRP), 13(08),345-358.
http://dx.doi.org/10.29322/IJSRP.13.08.2023.p14015

11. Adenowo, A. A., & Adenowo, B. A. (2013). Software engineering methodologies: a review of the waterfall model and object-oriented approach. International Journal of Scientific & Engineering Research, 4(7), 427-434. https://www.researchgate.net/publication/344194737_Software_Engineering_Methodologies_A_Review_of_the_Waterfall_Model_and_Object_Oriented_Approach

12. Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. Journal of Database Management (JDM), 16(4), 88-100. https://www.researchgate.net/profile/Keng-Siau-2/publication/220373708_Agile_Modeling_Agile_Software_Development_and_Extreme_Programming_The_State_of_Research/links/5984f29f458515605844f08d/Agile-Modeling-Agile-Software-D pdf?origin=journalDetail&_tp=eyJwYWdlIjoiam91cm5hbERldGFpbCJ9

13. Yahya, N., & Maidin, S. S. (2022, September). The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team. In 2022 10th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-5). IEEE. https://ieeexplore.ieee.org/document/9936036

14. Raval, R. R., & Rathod, H. M. (2014). Improvements in Agile model using hybrid theory for software development in software engineering. International Journal of Computer Applications, 90(16). https://research.ijcaonline.org/volume90/number16/pxc3894677.pdf

15. Gupta, A., Poels, G., & Bera, P. (2022). Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. IEEE Access, 10, 119745-119766. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9945932

16. T. von der Maßen, H. Lichter. (2002, September). Modeling variability by UML use case diagrams. In Proceedings of the International Workshop on Requirements Engineering for product lines (pp. 19-25).Citeseer. https://swc.rwth-aachen.de/docs/2002_PLE_Essen.pdf

17. Il-Yeol, S. (2001, November). Developing sequence diagrams in UML. In International Conference on Conceptual Modeling (pp. 368-382). Berlin, Heidelberg: Springer Berlin Heidelberg. https://cci.drexel.edu/faculty/song/publications/p_ER2001-SQD.pdf

18. Berardi, D., Calvanese, D., & De Giacomo, G. (2005). Reasoning on UML class diagrams. Artificial intelligence, 168(1-2), 70-118. https://www.sciencedirect.com/science/article/pii/S0004370205000792

19. Btoush, E. S., & Hammad, M. M. (2015). Generating ER diagrams from requirement specifications based on natural language processing. International Journal of Database Theory and Application, 8(2), 61-70. https://www.researchgate.net/profile/Eman-Btoush/publication/275952818_Generating_ER_Diagrams_from_Requirement_Specifications_Based_On_Natural_Language_Processing/links/554a878e0cf21ed21358e791/Generating-ER-Diagrams-from-Requirement-Specifications-pdf

20. Btoush, E. S., & Hammad, M. M. (2015). Generating ER diagrams from requirement specifications based on natural language processing. International Journal of Database Theory and Application, 8(2), 61-70. https://www.researchgate.net/profile/Eman-Btoush/publication/275952818_Generating_ER_Diagrams_from_Requirement_Specifications_Based_On_Natural_Language_Processing/links/554a878e0cf21ed21358e791/Generating-ER-Diagrams-from-Requirement-Specifications-pdf

Appendix A. Overview of Project Progress

Tasks Completed and Tasks Ongoing:

The following lists the tasks completed:

- Develop a method to analyze and decompose existing code structures to design databases. (Objective 3)
- Research and analyze the required materials like essays and reports for this project, such as SDLC process and reports on the company's losses due to lack of documentation.
- Developed a preliminary version of the tool that can analyze and decompose existing code structures.

The following lists the ongoing tasks:

- Design the model and system workflow diagrams. It is about 70% completed.
- Implement artificial intelligence to analyze code and documents. It is about 10% completed.

Appendix B Revised Project Plan

Task	Description	Estimated Completion Time	Assigned To	Current Status
Build a Prototype	Build a prototype to demonstrate how it works	2-3 weeks	Matthew & Joe	In Progress
Test the Prototype with Database and AI	Check whether the data is successfully saved to the database and forwarded to AI	1-2 weeks	Matthew & Joe	Not Started
Interim Report	A progress update on the project after completing the prototype and testing	2-3 weeks	Matthew & Joe	Not Started
Build an AI-driven Tool Software	Implement the initial version with AI and Database	4-6 weeks	Matthew & Joe	Not Started
Design Software User Interface	Design the User Interface so that it can be easy to know how to use	2-3 weeks	Matthew & Joe	Not Started
Implement the final version of the Software	Complete all the function and User Interface	1-2 weeks	Matthew & Joe	Not Started
Software Testing	Test all the function and Use Case testing	1-2 weeks	Matthew & Joe	Not Started
Final Report & Presentation	Integrate all the things we make and prepare for the presentation	2-3 weeks	Matthew & Joe	Not Started

Appendix C1 Matthew Wong's Interim Report

Summary
Although AI becomes more popular, there are not many applications that are game-changing. While focusing on the development of AI, engineering AI application is essential so that the value of AI could actually moving into the society. Turning my thought into an action is interesting, like I am actually inventing something.

Current Progress
The overall design is completed and we are applying it to the application. Although it is tough but the first step , which is extracting metadata from the code, is almost completed. The database setup is also finished. To make the development smoother, I create a git repository so my teammate could constantly update the code without the worry of version control. I completed the data extraction from code.

Challenges Faced
We proposed a new way to use AI efficiently and securely. There are not much similar application on the market. It is challenging for us to reference and learn on how other people doing it

Future Plans
We still need to organised the extracted data from the database, and send it to the AI. After doing it,we will test the diagram generation by different kind of code. The code could be an open source application. Next, we will integrate a UI, and perhaps we could provide a UI allowing user to edit the result of the diagram.

Appendix C2 Joe Chan's Interim Report

Summary

During this phase of the project, my role was similar to that of a support person, responsible for reading different essays and reports and analyzing research materials related to our project. In addition, I also tried different methods to obtain the AI API keys required for the project, such as the ChatGPT API key and the Claude.AI API key. Although I successfully obtained the ChatGPT API key for free account, I encountered a problem when upgrading to the Plus version, which temporarily hindered my progress. Since I recently received the prototype, I will assist the team with testing and UI design in the future and will keep in touch with Matthew.

Current Progress

I have completed the reference collection and analysis of different essays and reports and have integrated them into this report. In addition, I successfully registered a free ChatGPT account and obtained the ChatGPT API key.

Challenges Faced

The main challenge I faced was the inability to upgrade to the Plus version, which limits the number of times the API key can be used. Free API key users have strict usage limits, and once the limit is reached, payment is required to continue using the service.

Future Plans

In the future, I plan to fix this by finding a way to pay for the ChatGPT API key, thereby removing the restriction on upgrading to the Plus version. In addition, I will be assisting Matthew in testing and integrating the program. Once the prototype is complete, we will start designing the UI and implementing any features we want to add.