IEEE-SEM, Volume 13, Issue 2, February-2025.
ISSN 2320-9151

AI-Oriented Micromanaged Development (AOMD): Revolutionizing Software Engineering through Intelligent Task Decomposition and Multi-Agent Collaboration

Mehmet Altuğ AKGÜL

Abstract- **The main purpose of this article is to introduce the basic components of the AI-Oriented Micromanaged Development (AOMD) approach and to discuss in detail how this approach is applied in software development processes. The article will examine how AOMD is used in processes such as project management, task allocation, coding, testing, integration and deployment, and will explain how these processes are optimized with AI technologies.**

The article focuses on understanding the basic components of AOMD and how these components come together. In particular, the processes included in the visual (microtask decomposition, AI-driven micromanagement, integration and quality assurance, collaboration with AI agents, deployment and monitoring, iteration and scaling) will be discussed in detail. Each of these processes will be examined in terms of how AI technologies are used and how these technologies are integrated into software development processes.

Index Terms- **artificial intelligence, ai-oriented development, ai agents, ai-assisted development**

I. INTRODUCTION

An approach called Artificial Intelligence-Oriented Micromanaged Development (AOMD) which name was determined by me, provides automatic delivery and management of development tasks by dividing them into smaller, easier-to-manage components within software development packages. This method offers an effective solution, especially for large changes and complex projects where traditional approaches are insufficient. As part of AOMD, artificial intelligence tools are integrated into various phases including project management task execution testing and delivery. Delivery can be done more quickly and efficiently with this method. By automating project processes and minimizing human error AOMD hopes to free up developers to work on more compilation-related tasks. The importance of AOMD has become even more clear as systems become more complex and traditional methods are unable to handle this complexity.

Multiple relationships between large packages and team coordination are among the challenges faced. AOMD has developed artificial intelligence tools to carry out these tasks automatically by breaking down projects into smaller easier-to-manage components to make this program widely available.

Software projects are getting more complicated these days. The system has become more complex and easier to use because of the widespread use of technologies like big data, cloud computing, IoT (Internet of Things) and machine advancements. These projects are typically challenging to distribute because they involve many products modules and relationships.

Scheduling tracking and manual task systems are frequently the foundation of traditional project management techniques. When it comes to the arrangement of complex components these parts fall short. To gain traction in this field AOMD is growing its AI tools. AI distributes and controls these computers automatically and projects are broken down into smaller, easier-to-manage microtasks. This project moves along more swiftly and efficiently in certain areas. By swiftly examining the modifications and new intervals that result from project design, AI tools can also dynamically update project plans.

To help readers understand the role that AOMD plays in software development, this article attempts to offer a general guide. The incorporation of AI technologies into software development and project management is thoroughly examined. The article’s discussion of the benefits, difficulties and possibilities of AOMD will assist you in developing the software for this strategy.

II. DECOMPOSITION INTO MICROTASKS

Complex dependencies and a multitude of module components are common in large-scale software projects. Dividing such projects into smaller, more manageable tasks allows for effective project management. This process enables better understanding and project management. Example: Take the development of an e-commerce platform project. Among the many components that comprise this project are user authentication, product catalog management, payment integration, order tracking and reporting.

The AOMD technique is used to divide each of these components into smaller tasks. As an example, the user authentication and password reset components. The complexity and priority of the task’s dependencies determine how they are categorized. To enable better task management and resource optimization, they are separated into simple, medium and complex categories. The task of user registration, for example may be simpler than the two-factor authentication task. and prioritized in accordance with the project’s main goals. Suppose that the payment integration module might be more significant than the reporting module. Completing one task may require completing another. Such as, order tracking may require the integration of payments. AI tools automatically separate the project into microtasks and categorize them in the AOMD approach. A project management program can automatically generate tasks, classify them according to dependencies, complexity, priority and analyze the project requirements.

Example Scenario: In a mobile app development project, AI tools automatically divide different components of the application (user interface, data management, API integration, etc.) into microtasks. For instance, the “user interface” component is divided into microtasks such as “home screen design” “menu navigation” “user profile screen.” These tasks are classified and prioritized based on their level of complexity. Also, dependencies are automatically identified. Suppose that the “user profile screen” task may require the completion of the “user authentication” task.

Each task can be better managed and controlled when large projects are divided into smaller ones. Project processes become more flexible when they are divided into smaller, more manageable tasks. It is simple to update or rearrange tasks to meet project specifications. Resources can be used more effectively by classifying and prioritizing tasks. Project procedures work more efficiently and quickly in this way. Partitioning into smaller tasks is one of the core phases of the AOMD methodology. This phase makes it possible to divide complicated projects into smaller, easier-to-manage tasks. Project processes can be managed more effectively and efficiently when tasks are categorized based on dependencies, complexity and priority. By automating this process with artificial intelligence tools project managers can reduce human error and maximize resource use. For AOMD to be implemented successfully, breaking down into microtasks is essential.

III. AI-DRIVEN MICROMANAGEMENT

One of the most important phases of the AOMD process is AI-oriented micromanagement. AI tools are used to automatically assign, manage and optimize project tasks during this phase. This procedure guarantees that tasks are assigned based on project requirements, resource usage and expertise. Task descriptions, dependencies and deadlines are also automatically generated, which improves the effectiveness and eliminates errors in project management procedures. To improve task assignment procedures, AI tools employ a variety of algorithms and machine learning models.

To assign tasks in the most effective manner, these tools examine project requirements, team member expertise and resource usage. To assign each task to the best candidate, AI tools examine team members prior performance areas of expertise and skills. Developers with experience in database administration can be given database optimization tasks and designers with UI/UX design experience can be given user interface design tasks. AI divides up the work evenly after analyzing team members present workload and resource usage. This keeps some team members from being idle or overworked. AI analyzes changes or new requirements that come up during the project process quickly and task assignments are updated appropriately. AI can transfer a task to a team member who can finish it more quickly if its priority changes.

AI-driven micromanagement enables the automatic determination of critical information such as task descriptions, dependencies, and deadlines. This automation makes project management processes more efficient and error-free.

AI tools examine project specifications and produce thorough task descriptions. The goal parameters and anticipated results of the task are all included in these descriptions. For an API integration task such as AI can automatically choose which data to transfer which APIs to use and which security precautions to take. To identify which tasks must be finished first AI examines the dependencies between them. Task prioritization and order are automatically scheduled as a result. For instance, completing a user authentication task might be necessary for a payment integration task. Artificial intelligence (AI) recognizes this dependency automatically, makes sure that tasks are finished in the right order and assesses team members workloads and project progress to establish reasonable deadlines for each task. When things change throughout the project, these dates can be updated dynamically.

AI can automatically change the due dates of other tasks if one takes longer than anticipated to finish. Team members can work more productively thanks to artificial intelligence tools that streamline task assignment procedures. As a result, project procedures move more quickly. Human error is reduced by automated task assignment and management procedures. To avoid confusion and delays, task descriptions, dependencies and deadlines are automatically established.

Task assignments and deadlines can be dynamically updated by artificial intelligence, which can also swiftly analyze changes that arise during the project process. This makes project management procedures more adaptive and flexible. Artificial intelligence allocates tasks equitably after analyzing team members present workload and resource consumption. This lowers project costs and makes better use of available resources.

IV. COLLOBORATIVE CODING BY AI-AGENTS

When working on individual tasks in software development processes, AI agents make use of machine learning models. To maintain code consistency and adjust to project architecture, these agents work together. Software development processes become more consistent, error-free and efficient because of this procedure. AI agents automate code completion procedures and offer developers code suggestions based on machine learning models. Suppose that, by examining related functions from the project history, an agent can recommend the best piece of code to use when writing a function. Error rates are decreased, and the development process is accelerated.

AI agents can automatically identify and correct possible coding errors thanks to machine learning models. Let's say an agent can identify a null reference error in code and offer a fix or preventative measure. AI agents suggest optimizations to enhance code performance using machine learning models. An agent can offer suggestions on how to write a loop more effectively or cut down on pointless database queries. AI agents improve their performance on subsequent tasks by applying the knowledge they acquire during the project process and the agent can work on a new task more rapidly and precisely by examining similar code snippets that have already been written. To maintain code consistency throughout the project, AI agents cooperate with one another while working on separate tasks. Maintaining code quality and being in line with the project architecture depend on this cooperation. AI agents interact with one another to uphold project-wide code standards and style.

To guarantee that code is written in accordance with a particular style guide such as an agent may interact with other agents. This guarantees a uniform code structure for the duration of the project. They cooperate to control dependencies that could affect one another's work. When an agent is working on an API integration task it keeps in touch with other agents that will be using the API to make sure it is integrated correctly and continuously checks on each other's work to avoid any mistakes or conflicts. For example, an agent can automatically step in to correct a mistake or notify the appropriate agent if it finds one in code written by another agent.

They communicate with one another all the time to adjust to the project architecture. An agent checks with other agents to see if a new module it has added conforms with the project architecture. Artificial intelligence systems that work together to accomplish complex tasks are known as multi-AI agents. When combined these agents can solve more complicated and large-scale problems even though they are made to carry out tasks on their own. Particularly in fields like software development, automation, data analysis and decision support systems the application of multi-AI agents holds enormous promise. Each AI agent can carry out its own tasks on its own. This enables agents to function without the need for human involvement.

Through information sharing and communication, AI agents can resolve complicated problems. By working together agents can pool their unique skills to tackle more complex issues. AI agents can learn from their experiences and adjust to novel circumstances by utilizing machine learning models. As a result, agents can function better in dynamic settings. The architecture of multi-AI agent systems is usually distributed. This enables agents to synchronize while operating in various cloud environments or systems. To make code more standardized, error-free and optimized AI agents work together. The projects overall code quality improves as a result.

Agent cooperation guarantees that possible mistakes are found and corrected as soon as possible. This lowers project process error rates. By assisting one another AI agents make sure that project procedures run more smoothly and swiftly. To align with the project architecture, agents must work together. As a result, the projects code structure remains constant.

V. INTEGRATION AND QUALITY ASSURANCE

Combining and integrating microtasks into a comprehensive codebase is a crucial step in the AOMD process. At this point automated tests are used to verify that the code is error, performance and functionality-free. We will go into detail about the integration and quality assurance procedures in this section. Project components are composed of small manageable tasks known as microtasks. Once these tasks are finished, they are integrated into a comprehensive codebase. This procedure guarantees that the projects structure is reliable and consistent. AI systems aggregate finished microtasks automatically. Task interactions and dependencies are considered during this merging process. For example, an e-commerce platform combines microtasks like payment integration product catalog management and user authentication into a single interoperable system.

To guarantee code consistency throughout the merging, process AI tools carry out several checks. Uniformity is guaranteed about function signatures, variable naming conventions and compliance with code style guides. As a result, the projects code structure is consistent. Integration tests are automatically carried out after microtasks have been combined. These tests verify that various parts function as intended. Let’s say the proper interaction between the payment integration and user authentication components is tested. Processes for quality assurance and integration cannot function without automated testing. Automated tests find bugs in the code early on. Errors are thus fixed prior to entering the production environment. For example, unit tests determine whether the functions generate the desired results.

Artificial intelligence tools could either directly intervene or provide recommendations for automatically correcting errors when they are detected. Performance tests examine the systems operation under specific load conditions. By identifying performance bottlenecks, these tests enable the code to be optimized for efficiency. Pretend that load tests determine whether the system can accommodate numerous users concurrently. Artificial intelligence tools can suggest optimization solutions to address performance issues when they are identified. Tests of functionality determine whether the system operates as intended. The requirements and user stories serve as the basis for these tests. Basic features like the user’s ability to add a product to the cart and make a purchase are tested to ensure they operate as intended in an e-commerce platform. Functional tests make it possible to find mistakes that have an immediate impact on the user experience. Continuous integration and continuous delivery (CI/CD) procedures are not complete without automated testing. These procedures automatically test and integrate new code changes into the production environment. Users can swiftly and safely receive updates and new features in this way.

Code optimization, error-free writing and consistency are guaranteed by automated testing and integration procedures. This guarantees that errors are found and fixed early on and improves the quality of the code throughout the projects. Error rates in project processes are thus decreased. Performance tests make sure the system operates more effectively. This facilitates the systems scalability and enhances the user experience. Users are guaranteed to receive updates and new features in a timely and dependable manner thanks to continuous integration and continuous delivery (CI/CD) procedures.

One of the most important phases of the AOMD process is integration and quality control. This step involves integrating microtasks into a comprehensive code base and using automated testing to guarantee the codes quality in terms of errors, performance and functionality. This procedure guarantees that the system functions more effectively lower error rates and improves code quality throughout the project. One significant illustration of how AI technologies can be applied successfully in software development processes is integration and quality assurance.

VI. DEPLOYMENT AND MONITORING

The deployment of the product to the production environment and ongoing monitoring are two of the last phases of the AOMD process. During this phase users will receive the product, and its functionality will be continuously monitored. Managing the systems scalability and gathering user feedback are also crucial steps in this process.

Among the most important phases of the software development process is the products deployment to the production environment. AI tools now automate the deployment procedures guaranteeing a timely and dependable product delivery to users. With the use of continuous integration (CI) and continuous delivery (CD) procedures AI tools allow the product to be automatically deployed to the production environment. Users can receive updates and new features more rapidly thanks to this process.

Automated deployment processes such as make it possible to quickly update an existing feature or add a new payment method on an e-commerce platform. [1]. AI technologies keep an eye on the systems performance after it is put into production. System errors, performance snags and user experience are all continuously monitored as part of this process. AI tools for example examine user interactions error rates and server response times in real time. AI systems automatically identify mistakes that happen in real-time environments and offer suggestions or direct assistance to correct them. Imagine AI can automatically switch to backup servers or notify the appropriate teams to address a server error when it is detected. The success of the product depends on gathering user feedback and controlling the systems scalability. AI tools allow the product to better meet user needs and meet increasing demands by automating these processes.

AI systems gather and evaluate user input automatically. This input can be gathered from several sources including user satisfaction surveys bug reports and user behavior. In a mobile application such as AI can analyze user reviews and in-app feedback to find significant trends and problems. They guarantee that the product is updated frequently in response to user input. Suppose that AI can automatically generate a new task to enhance a feature or suggest changes to the appropriate teams when it detects that users are having issues with it. In this way the systems scalability is managed through a variety of strategies and the product better meets user needs. For example, the systems load performance is continuously tracked and when required automatic scaling operations are carried out. This enables the system to accommodate increasing demands. An e-commerce platforms server capacity can be automatically raised to accommodate spikes in traffic during designated campaign times. Automatically scale resources based on the load status of the system of course. AI can automatically add more servers or increase server capacity when traffic increases ensuring that the system continues to function properly even during periods of high traffic.

Automatic deployment procedures guarantee that users receive updates and new features in a timely and dependable manner. Artificial intelligence tools track the systems performance continuously identifying and fixing problems early. The uninterrupted operation of the system is thus guaranteed. User feedback is automatically gathered and analyzed so that the product can better meet user needs. Resource management and automatic scaling make sure the system can handle increasing demands and continue to function properly even during periods of high traffic. The product is put into the production environment and closely watched during this phase. By automating procedures like real-time monitoring automatic deployment user feedback analysis and scalability management artificial intelligence tools improve product success. An important illustration of the successful application of artificial intelligence technologies in software development processes is this procedure.

VII. ITERATION AND SCALING

During this phase the product is continuously enhanced to satisfy increasing demands. Artificial intelligence (AI) tools streamline these procedures enabling more effective and efficient product scaling. A vital tool for ongoing product development is feedback. By analyzing user feedback AI tools can automate the process of adding updates or new features. AI systems automatically gather and examine user input from a variety of sources including surveys social media and in-app reviews.

These analyses show which features users are most interested in or which problems they are having. A new payment method could be added to an e-commerce platform to fix issues that are found during the checkout process. They make it possible to plan updates or new features based on user feedback. AI can automatically generate a new task to enhance a feature if users are having issues with it or suggest solutions to the appropriate teams. The product better suits user needs in this way.

The rapid and dependable delivery of new features to users is made possible by continuous integration and continuous delivery (CI/CD) procedures. A new feature that users have requested in a mobile application can be created and made available in a matter of days.[2] The process of expanding the system’s capacity to accommodate rising demands is known as scaling. Artificial intelligence (AI) tools streamline scaling procedures enabling more effective and efficient system scaling. AI systems scale resources automatically based on the load level of the system. Case in point, AI can automatically add more servers or increase server capacity when traffic increases. As a result, even during times of high traffic the system functions flawlessly. As proof, an e-commerce platforms server capacity can be automatically expanded to accommodate spikes in traffic during designated campaign times and the systems resource usage can be continuously monitored and optimized. AI can identify when resources are being used excessively and make sure they are used more effectively. As a result, the systems performance improves, and its costs decrease.

By forecasting the systems load status in the future these tools help plan scaling procedures beforehand. For instance, by anticipating the spike in traffic during special campaign times or holiday seasons AI can boost server capacity beforehand. This ensures uninterrupted system operation during periods of high traffic. They adjust and continuously assess the systems performance. AI is capable of automatically identifying performance bottlenecks and optimizing them to remove them.

As a result, the system operates more quickly and effectively. Feedback-driven iteration procedures enable the product to better meet user needs. Users are more satisfied as a result. New features and updates can be developed and released more quickly thanks to AI tools. This preserves the products edge over competitors. The system can accommodate increasing demands and continue to function properly even during periods of high traffic thanks to automatic scaling and resource management.

At this point the systems scaling is optimized, and new features and updates are added in response to user feedback. AI technologies streamline these procedures guaranteeing that the product is improved over time and satisfies expanding needs. This procedure serves as a significant illustration of how AI technologies can be applied successfully in software development processes.

VIII. BENEFITS AND CHALLENGES OF AOMD

By automating tasks like assignment coding testing and deployment AOMD frees up developers to work on more strategic projects. Project processes move more quickly as a result. New features and updates can be developed and released more quickly thanks to artificial intelligence tools. By doing this the products competitive advantage is preserved. Automated tests in AOMD allow for early error detection and correction. This greatly lowers the rate at which errors enter the production environment. The systems performance is continuously monitored by artificial intelligence tools which identify and fix problems instantly. This guarantees that the system will continue to function. AI tools make it possible to divide up tasks based on resource usage and expertise. Project costs are decreased, and resources are used more effectively in this way. AI automatically adjusts resource levels based on the systems load condition. As a result, the system can accommodate increasing demands. AI tool development or acquisition can be expensive initially.

These expenses may pose a major obstacle particularly for small and medium-sized businesses. AI tools can be expensive to maintain and update on a regular basis. Teams must receive training on AI tools to use them effectively. Time and resources are needed for this training process. Teams need to be continuously aware of new technologies and approaches because AI technologies are developing quickly. This calls for an ongoing process of learning. Large volumes of data are gathered and analyzed by AI tools.

This data’s security and privacy may raise serious ethical concerns. There may be severe repercussions especially if user data is misused or leaked. Certain jobs may be eliminated because of AI tools automation capabilities. This may result in social issues and detrimental effects on the workforce. Depending on the data they are trained on AI models may acquire biases. Injustice or poor choices in project procedures may result from these biases.

IX. FUTURE PERSPECTIVES

AOMD is bringing about a major change in the way software is developed. This strategy’s future development and effects on various industries could be greatly influenced by sectoral demands and technological advancements. This section will look at AOMDs future function and possible industries in which it could be used. Future developments of AOMD will be influenced by sectoral requirements as well as advancements in artificial intelligence technologies. Sectors will be significantly impacted as this method spreads throughout software development processes.

AOMD processes will be further automated and optimized in the future by deeper learning algorithms and more sophisticated machine learning models. These models will be faster and more accurate at handling increasingly complicated tasks. Real-time data analysis capabilities will be improved by artificial intelligence tools allowing for quicker and more efficient decision-making throughout project processes. In project environments that are dynamic and changing quickly this will be a huge advantage. In the software industry AOMD will improve the efficiency of project management coding testing and distribution procedures.

This will lower expenses and speed up software development processes. AOMD can be utilized by the banking and finance industry to facilitate the quicker and more dependable development of intricate financial systems. With AOMD automated reporting and risk management systems can be optimized. AOMD can be used by the biotechnology and healthcare industries to facilitate the quicker development of biotechnological systems and medical software.

X. CONCLUSION

In this article, we discussed the significance of the AI-Driven Micromanaged Development approach which demonstrates the efficient application of AI technologies in software development processes. This method enables more effective error-free and user-focused project management. However, challenges like the price of AI tools the need for training and potential moral dilemmas should also be considered. With careful management and planning these challenges can be overcome. In addition to bringing about a substantial change in the software sector AOMD finds extensive use across various industries.

Future developments in artificial intelligence technologies and sectoral requirements will determine how much more of an impact AOMD has. This method has a lot of potential for use in software development processes and other fields. AOMD has established itself as a significant player in the software development industry and is only going to grow in influence. This strategy uses the benefits of artificial intelligence technologies to make project processes more effective error-free and user-focused. A major shift in the software industry and other industries could be brought about by AOMD. For future projects to succeed AOMD adoption and development are therefore essential.

XI. REFERENCES

[1] Fischer, Mahsa & Lanquillon, Carsten. (2024). Evaluation of Generative AI-Assisted Software Design and Engineering: A User-Centered Approach. Lecture Notes in Computer Science. 14734. 31–47. 10.1007/978-3-031-60606-9_3.

[2] Kalava, Sudheer. (2024). AI-Powered Development: How Artificial Intelligence is Shaping Software Productivity. Journal of Artificial Intelligence & Cloud Computing. 3. 1-4. 10.47363/JAICC/2024(3)E148.

XII. ABOUT AUTHOR

Mehmet Altuğ Akgül is an Electrical-Electronics Engineer, Cloud Product Manager, and Solution Architect, born in 1995 in Ankara, Türkiye. Holding a Master's degree in Business Administration (MBA) and a second degree in Management Information Systems, he effectively combines technical expertise with strategic thinking. Since 2021, he has been an e-resident of Estonia. Certified as an AWS Cloud Solutions Architect and a Professional Scrum Product Owner, he specializes in cloud engineering, software architecture, and full-stack development. His passion lies in designing innovative cloud solutions and leading cross-functional teams to deliver impactful projects.