International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Automatic Software Development With AI

NADENDLA VENKATESWARARAO*[1], ATLA SRAVANI[2],ANUMULA SRI LAKSHMI[3], ATHINA SAI [4], KURRI VENKATA KRISHNA [5] TATINI NAGENDRA [6]

[1]Assistant Professor, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

[2]Student, Department of CSE(AIML),Bapatla Engineering College, Bapatla 522101,AP, India.

[3]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

[4]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

[5]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

[6]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

Corresponding author. E-mail: atlasravani2@gmail.com

Abstract-The modern-day applications demand rapid development, quality code, and a close collaboration among many components, making it increasingly complex in the world of software development these days. Developers must manage all the phases of the SDLC: design, coding, planning, implementation.Assistants based on AI are able to comprehend the requirements of the user, generate code, and increase the productivity directly in the development process. This is the Automatic Software Development with AI project and the project develops an intelligent system that converts natural language instructions into a fully functional software.

For this project we have chosen a multi-agent design implemented using LangGraph that is similar to a real life software development team. If specifically, it does not follow in a monolithic model, where different chores are shifted to the specialists to get done simply without any messy handling. We forested out three basic actors the Planner, Architect & Coder. None of them do a same job: they take the user request, translate it into code and guarantee.

1.INTRODUCTION

Well, it is far more complex now with software development with all the technology exploding and the number of individuals who desire fancy applications increasing. To make it reliable and fast, you need to plan things ahead, do a decent design and code correctly. The end result is that most of us end up wasting plenty of time mashing the requirements, drawing up the architecture and literally writing the code. Doing all that manually is a time sink, and may introduce a lot of inefficiencies. Most recently, AI has been this super-power mechanism that can give us a chance to automate much of the dev workflow. It is capable of pumping out code, interpretation of requirements as well as an overall increase in our productivity.

First up is the Planner Agent. It accepts the user-natural-language prompt and spits out an elaborate project plan. The plan defines the structure of the app, the technology stack as well as the major development blocks. When this is done, the Architect Agent accepts the plan and divides it into bite-sized engineering tasks, determines what files and modules are required and prepares the general project skeleton.

At that point the Coder Agent enters in to write the code. It creates the files to be included in the project and simply drops them in project folders, which is a convenient practice as in our coursework and in our real-world coding sessions.

We put the entire thing in a more recent Python stack, with virtual environments, and connected it to the Groq API to write that spicy AI-powered code generation. Ultimately, such a system does demonstrate how AI and multi-agent collaboration can reduce the amount of heavy lifting involved in planning, designing, and coding, and ultimately make the entire process of dev more efficient and quicker.

The Automatic Software Development with AI project is expected to simplify all that by simply allowing an AI to convert plain English instructions to a completed application. Rather than dumping all the work into a single big model the system breaks the work down into a number of agents each dealing with a separate stage of development. It would be more organized and would run more smoothly that way.

It has three agents, the Planner, Architect and the Coder. The Planner reads your request and drools a elaborate plan which includes the tech stack and the structure of the app. The Architect will then take that plan, cut the plan into bite sized tasks and lay out the required files and modules. Lastly, the Coder literally codes the code, and puts it into the appropriate files.

It is based on LangGraph and is developed with the latest development tools such as Python environments, and AI integration. Therefore an AI application and a multi-agent solution go a long way to demonstrate how intelligent automation can assist us in making plans, designing and accomplishing projects more quickly and efficiently.

2.LITERATURE SURVEY

Over the last few years, Artificial intelligence (AI) has been influential in revolutionising software engineering. Scientists have investigated diAerent AI-based methods to assist the developers in diAerent activities, including: code generation, debugging, requirement analysis and software design. Combination of machine learning and natural language processing has made systems comprehend human input and create programming code on their own. The innovations have resulted in the creation of smart code writing tools which can enhance productivity and decrease the time to develop. The generation of AI-assisted codes with Large Language Models (LLMs) is one of the promising keywords in this area. These models are trained using enormous amounts of programming languages and packages of software and can produce code using natural language prompts. It has been demonstrated by various studies that LLMs can aid developers by providing code snippets, suggestions and debugging tasks. The tools that are informed by these models have shown that they can help to automate repetitive coding and enable developers to create software programs more eAectively. The other research field is on multi-agent systems in software development. Multi agent system consists of a number of intelligent agents, which work together to accomplish complex problems.Researchers have suggested systems in which various agents undertake activities in software engineering like requirement analysis, project planning, architecture design and code generation. Responsibility is shared between specialized agents, and this enhances the process of development activities. The method is also indicative of the arrangement of actual software development teams in which various roles work with each other to accomplish a project. Other recent works also have brought AI-powered development models that combine both planning, architecture design, and coding into automated processes. These formalities seek to transform the natural language instruction into complete software project. These systems prove the idea that AI can be used at various phases of development lifecycle and not on the code generation only. Regarding such developments, the suggested project, Automatic Software Development with AI, uses multi-agent architecture, which is used to automate the software development process. The system will also have planning, architectural design and code agents, which will facilitate efficient translating of user instructions to a full application. To create an intelligent development assistant, the given methodology relies on the existing research on AI-assisted programming and agent cooperation.

3. PROPOSED SYSTEM

The system suggested, Automatic Software Development with AI, is aimed at automating the process of software development, which implies transformation of the natural language instructions into full-fledged software applications. It is based on Artificial Intelligence and it works on the multi-agent architecture to replicate the process of a real software development team. The system provides a systematic and efficient approach to developing software through division of the process into various stages which are addressed by specialized agents.

The system proposed has three principal intelligent agents namely the Planner Agent, the Architect Agent and the Coder Agent. The software development life cycle has a particular task of each agent. It starts with the user making a natural language request that defines what kind of application he or she wants to create. This input is analyzed by the system and initiates the automated process of development.

The Planner Agent will be in charge of knowing the needs of the user. It examines the natural language prompt and produces an elaborate plan of the project. This plan will contain the application objectives,

technologies required and the general layout of the software project. The planning phase assists in taking care of the fact that the system has a clear understanding of the development needs prior to the start of the implementation.

The generated plan is then sent to the Architect Agent who then converts the plan into a developed development architecture following the planning phase. It divides the project into engineering bites and determines what files, modules and components will be required. This exercise provides the project bone and gives a clear roadmap of each file in order that development will be done in a well-organized manner. This step determines the outline of the project and gives a clear guideline on each file so that the process of developing becomes systematically planned.

The last step is taken by the Coder Agent which produces the actual code to be used in each task specified by the Architect Agent.The resultant code is merely inserted to the project files just as a software developer would insert any feature to a real development environment. The code is also a subject of the agent that is used to maintain standard programming practices.

This system is built based on the LangGraph framework, Python virtual environments, and the Groq API to make it possible to plan and generate code with the help of AI. The given system demonstrates how AI can be used to automate the software development process and how it can actually increase its efficiency.

4. METHODOLOGY

The project design conducted through the use of the methodology of the Automatic Software Development with AI project is aimed at creating a multi-agent system that will be able to transform natural language instructions into a complete software application automatically. The system is developed in a systematic workflow that resembles the actual software development life cycle such as planning, architecture development and implementation of the code. The collaboration of multiple intelligent agents constructed with the help of the LangGraph framework is used to reach this process.

Requirement analysis and project planning is the first phase of the methodology. At this step, the user gives a natural language query of the application he/she desires to develop. The Planner Agent does the User Requirement analysis and analysis of the requirements.According to the input, the agent will produce a comprehensive project plan, comprising of the general structure of the application, technologies required and the key elements that one needs in order to develop it. This phase makes the system clearly understand the project objectives, prior to going to the next phase.

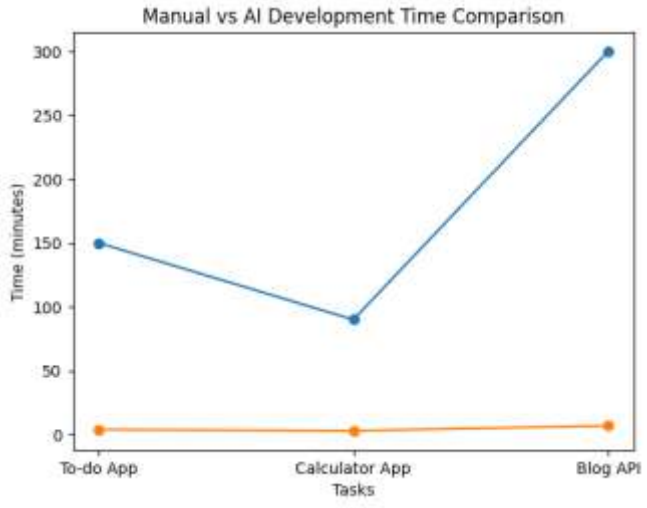
The second phase is the system architecture design. During this step, the Architect Agent receives the project plan created by Planner Agent and divides it into smaller engineering tasks. The agent determines the project structure by determining files, modules and components that are needed to develop the application. It also gives stepwise instructions on every file, thus making the process of coding very structured and balanced.

The third stage is the code generation and implementation. At this level, the Coder Agent will come up with the code that it needs depending on the work it has been assigned by the Architect Agent. The code generated gets coded automatically in the project files. The agent incorporates the programming tools and libraries to make sure that the code generated adheres to standard practice code development.

Python is used to implement the system in which a virtual environment is established to handle dependencies. The application is linked with the Groq API to have access to potent AI models that can produce plans and code. After the environment setup has been done, then the system can be run by use of a command-line interface, which means that the users can interact with the application and generate software projects automatically.

5.ARCHITECURE

6. RESULTS



Graph

- The graph shows **huge time reduction**
- Manual time (in minutes) vs AI time is clearly visible

Figure 1: Time comparison between traditional development and AI-based system

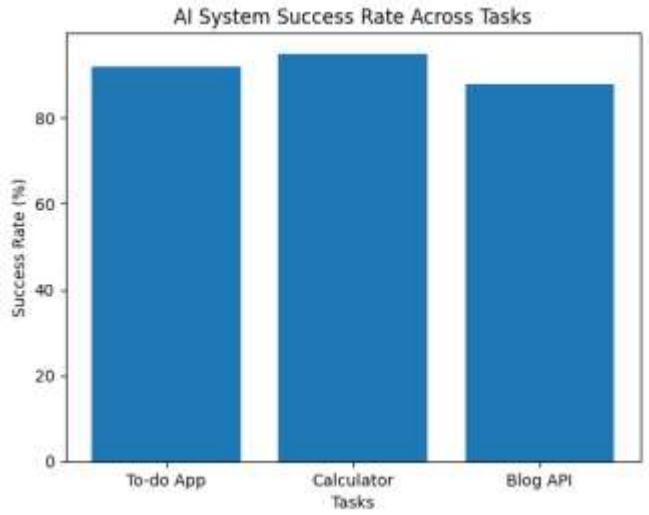


Figure 2: Success rate of AI system across different application types

The success rate of the proposed system remains consistently high across different tasks, with the highest performance observed in simpler applications such as calculators. Slight variations are observed in more complex applications like blog APIs, indicating scope for improvement in handling complex workflows.

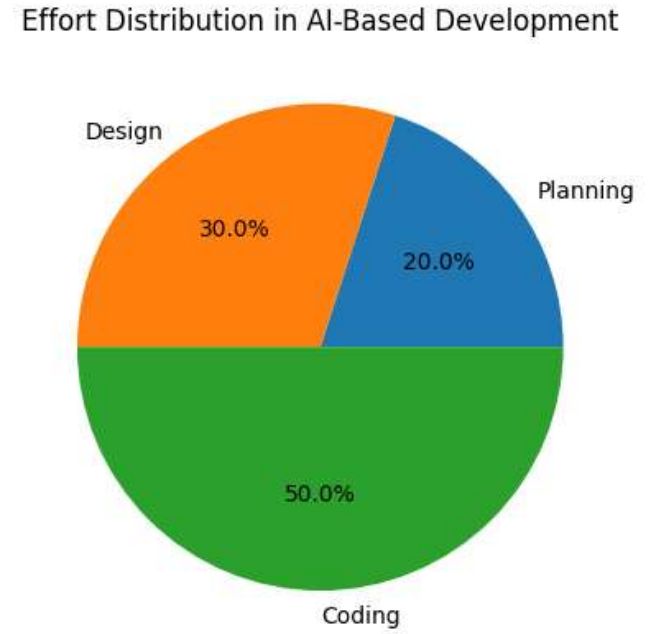


Figure 3: Effort distribution across different stages of AI-driven development

The majority of effort is concentrated in the coding phase (50%), followed by design (30%) and planning (20%). This demonstrates that while AI automates development, code generation remains the most resource-intensive component.

7.CONCLUSION

As the project Automatic Software Development with AI states, one can also apply Artificial Intelligence to automate the different phases of the software development process. The conventional software development methods are quite cumbersome and lengthy due to the duration taken in the requirement analysis, system design, coding and implementation. With the addition of AI into this working process, such tasks can be simplified and the development becomes more efficient. The given system is based on a multi-agent architecture that emulates the teamwork of the actual software development team, allowing to generate software applications out of natural language instructions automatically.

The system comprises three principal agents namely, the Planner Agent, the Architect Agent and the Coder Agent. All the agents play a particular role during development. The Planner Agent examines the request of the user and produces the detailed project plan. Then, the Architect Agent translates the plan into the organized development activities and determines the project architecture. Lastly, the Coder Agent is configured to compile the necessary code and to place them into project files.This is a systematic method to make sure the development process is planned, efficient and scalable.

The system being implemented with the help of the LangGraph framework, Python environment, and Groq API integration allows the AI agents to engage in the sophisticated reasoning and code generation. The system shows the potential of several AI agents to cooperate and automatize software development activities to make developers do less of them manually. It also demonstrates the possibility of AI to help novices and professional programmers because it makes it easier to develop software applications.

To sum up, the Automatic Software Development with AI project emphasizes the increasing significance of AI-based tools in the contemporary software engineering. The system enhances better productivity and shortens the development time by automating planning, designing and coding. Such smart systems can be improved in the future to accommodate bigger and more sophisticated software projects and AI will be a very crucial aspect in the future of software development.

8. REFERENCES

[1] LangChain AI, *"LangGraph Documentation,"* 2024. [Online]. Available: https://langchain-ai.github.io/langgraph/

[2] Groq Inc., *"Groq API Documentation,"* 2024. [Online]. Available: https://console.groq.com/docs

[3] S. Ramírez, *"FastAPI Documentation,"* 2024. [Online]. Available: https://fastapi.tiangolo.com/

[4] Python Software Foundation, *"Python Documentation,"* 2024. [Online]. Available: https://www.python.org/

[5] OpenAI, *"GPT-4 Technical Report,"* 2023.

[6] T. B. Brown et al., *"Language Models are Few-Shot Learners,"* in *NeurIPS*, 2020.

[7] A. Vaswani et al., *"Attention Is All You Need,"* in *NeurIPS*, 2017.

[8] A. Chowdhery et al., *"PaLM: Scaling Language Modeling with Pathways,"* 2022.

[9] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.