- **CodeCoR** [14] emphasizes reflective agents for self-improvement in code quality. While effective, the system does not explicitly handle modularity or real-time task decomposition, which are core to CodeCodez.
- **Context Engineering for Multi-Agent Code Assistants** [17] explores maintaining long-term coherence among collaborating agents. CodeCodez extends this concept by embedding context at the task-tree level and ensuring it flows dynamically across subtasks.

4. Surveys and Meta Frameworks

Recent surveys such as **J. Park et al.** [10] and **Y. Dong et al.** [19] highlight common architectural patterns in LLM-based agent systems. These include pipeline-based, debate-based, and consensus-based models. However, none explicitly formalize the end-to-end software creation lifecycle. CodeCodez fills this gap by combining hierarchical decomposition with a role-based multi-agent architecture that spans from requirement intake to deployment-ready code.