- **AgentDropout** [8]

5. Development and Testing Ecosystems

For code generation-specific tasks, systems like **AgentCoder** [15], **CodeCoR** [14], and **AdaCoder** [16] integrate:

- Code interpreters
- Automatic testers (via Python/JS frameworks)
- Optimization modules to iteratively refine generated software components.

6. Orchestration Tools and APIs

Tools like **LangChain**, **LLM-Orchestration Engines**, **Prompt Layer**, and custom pipelines using **Python multiprocessing** and **task queues** are widely used to implement the control flow between agents. These systems ensure communication, context sharing, and asynchronous task management.

2.2 Software Requirement Specification

2.2.1 Introduction

2.2.1.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to define and formalize the requirements, goals, and expectations of the proposed system *CodeCodez*. This system aims to provide an automated, end-to-end solution for software project generation using natural language specifications. By leveraging multiple specialized Large Language Models (LLMs) and structured task decomposition, CodeCodez seeks to significantly streamline the software development lifecycle, reduce manual effort, and accelerate project delivery.

The document ensures a common understanding among all stakeholders—developers, testers, researchers, and end-users—by clearly outlining the system objectives, functional and non-functional requirements, interfaces, and constraints. It serves as a contractual agreement between the project team and the stakeholders, providing a reference point to guide the entire software development process.