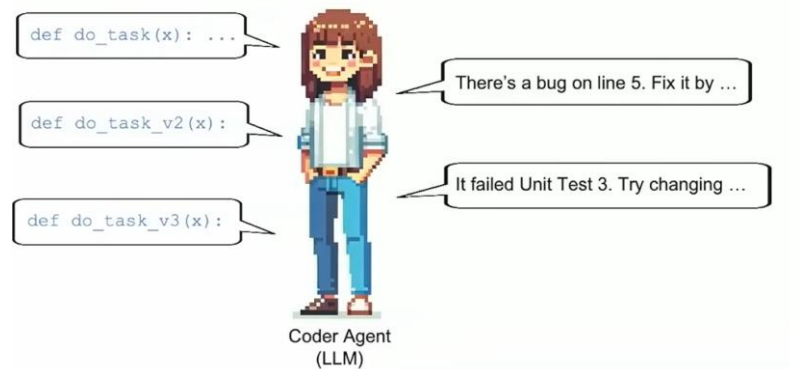


智能体：能够自我感知环境，能独立做出决策，并能够主动执行的人工智能系统。

智能体会让解决任务的过程更和人类相似。人类再面对问题时候，会把问题**拆分**成不同的部分，然后去逐个击破，在分步完成任务的时候，会针对这个任务去做反思，去检查，然后再去补充，整个过程是一个**动态的过程**，是一个**交流的过程**，这一点有别于目前的非智能体，非智能体在生成的时候，他不会去反思自己的生成过程。



ChatDev

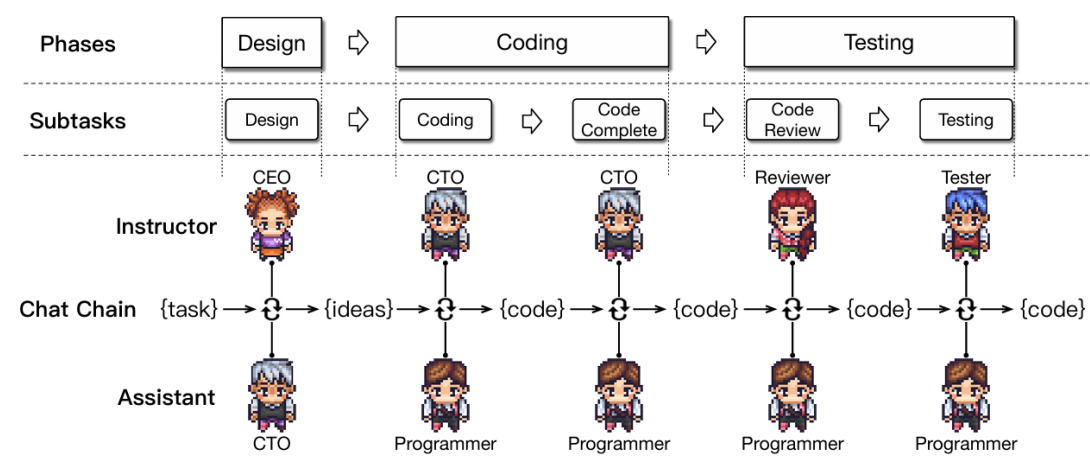
ChatDev 拟作一个由多智能体协作运营的虚拟软件公司，在人类“用户”指定一个具体的任务需求后，不同角色的智能体将进行交互式协同，以生产一个完整软件。



ChatDev 框架概要：让多个智能体分工合作。

ChatDev 分为 软件设计，系统开发，集成测试，文档编制四个主要环节。

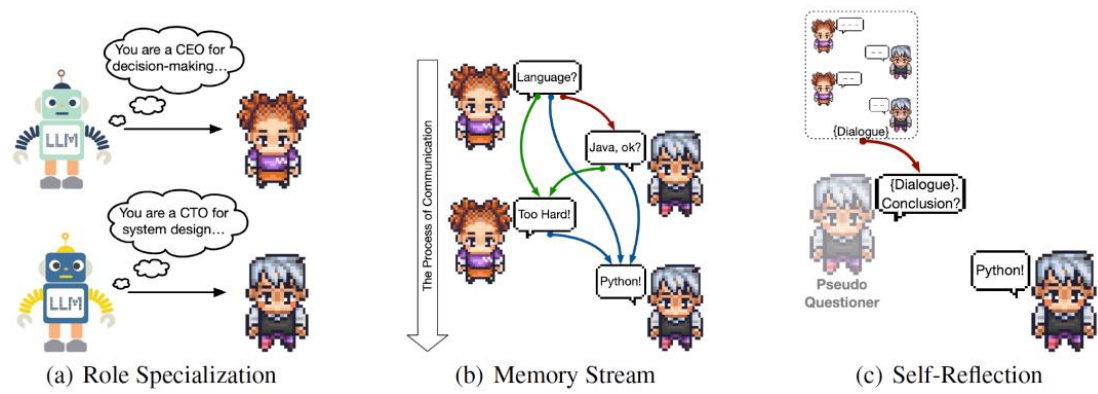
聊天链：



多轮对话和角色协作来完成软件开发的各个阶段，确保每个阶段都有明确的任务和责任分配。

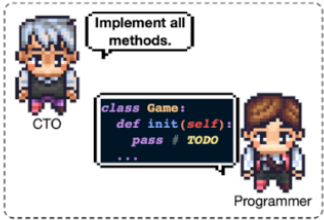
驱动智能体交流对话的主要机制为：角色专业化（Role Specialization）、记忆流（Memory Stream）、自反思（Self-Reflection）

- 角色专业化通过角色扮演机制（Role-Playing）确保每个智能体各司其职，在指定的专业角色下完成相应的方案提议和决策讨论。
- 记忆流通过将历史对话进行呈现，保证上下文感知的对话过程，并动态地对对话历史信息进行汇总和决策。
- 自反思机制在对话没有自动触发结束协议时生效，通过整轮对话进行文本摘要，在软件开发过程中不断评估和改进自己的工作摘录出对话过程达成一致后的最终决议。

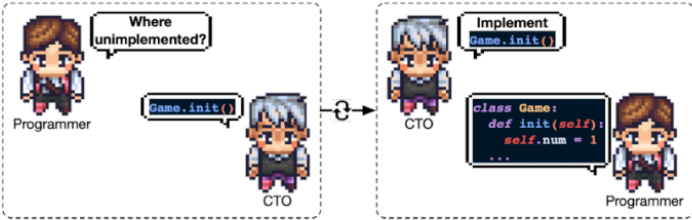


在编码和测试阶段，为了缓解代码幻觉问题

具体通过角色翻转（Role Flipping）来获取交流过程中的代码审查员（Instructor）实际想让程序员（Assistant）修改的代码块精确片段，后将此思维注入到提示中以更加细粒度地将信息指示给程序员让其进行代码优化。



(a) Naive Instruction in Coding



(b) Thought Instruction in Coding

