Modern Engineering and Application 现代工程与应用, 2026, 4(3), 16-24
Published Online August 2026.
https://ufascience.com/mea/index

利用 OpenAI API 实现软件开发各阶段的自动化

高宇辰

四川大学商学院，成都 610065，四川，中国

摘要：近年来，随着各类组织力求提升效率、缩短产品上市时间，自动化已成为软件开发的重点方向。人工智能（AI）工具——尤其是 ChatGPT 这类使用自然语言处理（NLP）的工具——的引入，为软件开发生命周期中各阶段的自动化开辟了新的可能。本研究的主要目标是评估 ChatGPT 在软件开发各阶段自动化中的有效性。本文基于 OpenAI 应用程序接口（API）研制了一款 AI 工具，具备两项关键功能：（1）根据案例或流程输入生成用户故事；（2）估算执行每个用户故事所需的工作量。此外，本文还使用 ChatGPT 生成应用程序代码。该 AI 工具在三个案例研究中进行了测试，每个案例均按两种不同的开发策略展开：一是借助 AI 工具的半自动流程，二是传统的人工方式。结果表明，开发总时间显著缩短，降幅在 40%至 51%之间。但同时也观察到，所生成的内容可能存在不准确与不完整之处，在应用于项目前需要审查与调试。综上，鉴于软件工程日益转向自动化，进一步研究以提升 AI 工具（尤其是利用 NLP 技术的工具）的效率与可靠性至关重要。

关键词：人工智能工具；用户故事；ChatGPT；AI 工作量估算；软件开发流程自动化

Automation of Software Development Stages with the OpenAI API

YuChen Gao

Business School, Sichuan University, Chengdu 610065, Sichuan, China

Abstract: In recent years, automation has become a key focus in software development as organizations seek to improve efficiency and reduce time-to-market. The integration of artificial intelligence (AI) tools, particularly those using natural language processing (NLP) like ChatGPT, has opened new possibilities for automating various stages of the development lifecycle. The primary objective of this study is to evaluate the effectiveness of ChatGPT in automating various phases of software development. An AI tool was developed using the OpenAI application programming interface (API), incorporating two key functionalities: (1) generating user stories based on case or process inputs, and (2) estimating the effort required to execute each user story. Additionally, ChatGPT was employed to generate application code. The AI tool was tested in three case studies, each explored under two different development strategies: a semi-automated process utilizing the AI tools and a traditional manual approach. The results demonstrated a significant reduction in total development time, ranging from 40% to 51%. However, it was observed that the generated content could be inaccurate and incomplete, necessitating review and debugging before being applied to projects. In conclusion, given the increasing shift towards automation in software engineering, further research is

文章引用: 高宇辰. 利用 OpenAI API 实现软件开发各阶段的自动化[J]. 现代工程与应用, 2026, 4(3): 16-24.
DOI: https://doi.org/10.61784/mea2028

critical to enhance the efficiency and reliability of AI tools, particularly those that leverage NLP technologies.

Keywords: Artificial intelligence tools; User stories; ChatGPT; AI estimation; Automation of software development processes

一、引言

人工智能（AI）并非新概念，但数据量的指数级增长与算力的持续提升正在深刻改变企业的运营方式。深度学习推动了计算机视觉与自然语言处理等领域的重大进展。生成式人工智能技术（即利用算法生成新数据的技术，如 ChatGPT 等语言模型）的快速普及，极大地降低了基于 AI 的产品与服务的使用门槛。

ChatGPT、Bard 等生成式 AI 模型在敏捷软件开发中的重要性日益凸显。在以协作与适应性为要的敏捷方法中，这类模型可在若干关键阶段增强并加速软件开发过程。在项目规划与定义阶段，它们能够高效生成用户故事，帮助开发团队准确解读客户需求并将其转化为清晰、详尽的描述，从而促进对项目目标的理解以及团队成员与利益相关方之间的沟通。在设计阶段，生成式模型可用于编写技术文档与规格说明，简化原型设计流程并为软件架构的关键决策提供依据。在开发阶段，这类模型可协助生成源代码，有望显著加快实现过程；虽然仍需人工监督与审查，但这一能力可提升生产率并降低出错风险。

本文提出一种利用 OpenAI“gpt-3.5-turbo”模型的方法，用以应对敏捷软件开发中的一项现实挑战：用户故事（US）的自动生成。用户故事是软件开发项目规划与执行中的基本要素，本文的方法探索了 AI 如何有效地服务于这一基础流程，力求为借助先进模型提升敏捷软件开发的效率与质量开辟新的思路。

为此，本文基于 OpenAI 的应用程序接口设计了一款工具，用于生成用户故事并估算各项任务所需的工作量。这些能力在三个不同的实际场景中接受了检验：开发就诊与用药提醒系统、实现酒店客房预订系统，以及设计车辆预订与租赁系统。此外，本文还使用 ChatGPT 自动生成应用程序代码。与此并行，相同的用例也以传统方法人工开发。随后本文对两种方式的开发时间进行了详细比较，从而评估自动化相较常规做法在实现所需方案时的相对效率与有效性。

本文结构如下：第一部分阐明主要目标、简要概述流程并说明全文结构；第二部分全面介绍与 ChatGPT 相关的基础议题；第三部分详述所采用的研究方法；第四部分给出所得结果；第五部分讨论研究结论。

二、文献综述

（一）人工智能

1955 年提出“人工智能”（AI）一词的 John McCarthy 把它定义为机器运用语言、进行抽象、应对挑战并自我改进的能力 [1]。自诞生以来，AI 已发展为一门旨在复现人类思维与行为的独立学科。AI 的范畴不断扩展，吸引了各领域研究者，这种跨学科合作把 AI 应用于包括教育在内的众多领域 [2]。AI 系统借助受人脑神经网络启发的算法来模拟人类推理。机器学习与数据处理工具的日益普及，持续推进 AI 在商业、政务等行业中的作用，帮助解决复杂问题并促进可持续发展 [3]。

（二）自然语言处理

自然语言处理（NLP）是 AI 的重要分支，聚焦于计算机与人类语言之间的交互，使机器能够理解、解释并生成类人的文本或语音。早期 NLP 模型出现于 20 世纪 50 年代，而深度学习的近期进展显著提升了其性能 [4]。现代 NLP 应用包括聊天机器人与虚拟助手，可在语言学习、科研等领域提供个性化支持。例如，先进的生成式语言模型 ChatGPT 自 2023 年初发布以来被广泛使用，在各种情境中为用户提供帮助 [5]。

（三）ChatGPT

聊天机器人已从简单脚本演进为部署在服务器与云平台上的高级系统。这类机器人可管理软件仓库、回答问题并支持协作，采用对话式与语音激活式界面 [5]。其主要目标是自动化重复性任务，但成效取决于良好的设计与透明的沟通能力。

由 OpenAI 研制的 ChatGPT 是运用机器学习技术进行类人对话的高级 AI 聊天机器人的典型代表。它构建于 2018 年首次提出的 GPT（生成式预训练 Transformer）架构之上。GPT 的目标是通过在海量人类生成文本上训练来预测序列中的后续词，并在机器翻译与文本生成等领域取得成功 [6]。尽管 ChatGPT 已相当先进，但它并无意识，仅依靠既定算法与强化学习来模仿特定的书写风格与对话模式；模型通过学习用户提问并优化回答而不断改进 [7]。ChatGPT 在生成类人文本与检索信息方面表现出色，是一种用途广泛的工具，但其输出可能反映训练数据中固有的偏见 [8]。

ChatGPT 的训练借助了大量数据与算力，推动了 AI 内容生成的发展。其架构基于深度学习框架，融合预训练模型与算法等多个组件，但数据处理与偏见方面的伦理问题仍是突出挑战。OpenAI 提供的 API 使开发者能够把先进的自然语言处理能力集成到自身应用中，可完成文本生成、语言翻译与问答等任务，从而提升各行业的效率与用户交互体验。ChatGPT 的若干替代方案（如谷歌 Bard 与微软 Bing Chat）同样提供对话式 AI 能力；此外，Rasa、Botpress 等开源平台还为开发者提供了更高的定制化空间。

（四）软件开发

软件开发是涵盖调研、规划、设计、开发、测试、配置与维护的多层面过程。在过去 50 年里，该过程已从非正式的“编码—修补”方式演进为旨在提升可预测性与效率的规范化方法，随着系统复杂度上升尤其如此。在软件工程中，开发过程并非刚性指令，而是一个灵活框架，使团队能够选择最合适的行动以保证按时交付高质量软件。该框架通常包含五项核心活动：沟通、规划、建模、构建与部署，它们以迭代循环方式实施，每轮循环带来功能上的增量改进。

敏捷方法是现代软件开发中的主流路径，其设计目标是适应产品需求与团队构成两方面的变化。敏捷团队强调灵活性、协作与客户的持续参与。敏捷方法建立在三条核心原则之上：未来需求难以预测；设计与实现紧密耦合；分析、设计与测试各阶段本身具有不可预测性。敏捷过程通过持续适应项目与技术条件的迭代开发周期来应对这种不可预测性，客户反馈在其中至关重要。通过把复杂项目拆分为更小、更易管理的增量，敏捷方法可实现更快交付、范围上的灵活性以及基于用户反馈的持续改进。

（五）用户故事

“用户故事”概念由 Kent Beck 于 1997 年提出、Rachel Davies 于 2002 年正式化，采用“作为{角色}，我希望{行为}，以便{理由 / 价值}”的格式，从客户视角刻画其问题。用户故事是描述用户与系统交互的简明叙述，强调用户所获得的价值而非技术细节。用户故事把客户目标拆分为可快速实现并进行质量保证（QA）评审的小型任务；当某个故事复杂或耗时较长时，会进一步拆分为更小的任务，以保证尽早测试与反馈。

（六）相关工作

评估 ChatGPT 在各领域有效性的实验正在推进。GREENGARD 评估了 ChatGPT 辅助唾液腺治疗临床决策的能力；研究显示其前景可期，但要确保医疗场景下的可靠性与安全性仍需进一步开发。ALTAMIMI 考察了 ChatGPT 在作文自动评分中的应用，聚焦其准确性与可靠性，发现 ChatGPT 显著提升了评分效率。ChatGPT 提供了有价值的思路，能生成想法并调试代码，但作者强调需与人类专业知识交叉核验。另一项研究评估了 ChatGPT 对软件工程教育的影响，指出其在向学生提供个性化反馈方面的潜力。

PETROVIĆ展示了 ChatGPT 在 DevSecOps 运行时日志分析中的应用，发现其精度可与传统分类方法相当。

SAID 等推出了基于 ChatGPT 的动画机器人头“Adam”，在问答任务中识别率高达 96%，展示了其在客户服务中的有效性。在基础设施即代码（IaC）领域，PETROVIĆ把 ChatGPT 用于静态分析，识别 Terraform 与 Ansible 脚本中的安全与语法问题；虽有前景，但研究也指出处理时间与相关成本方面的局限。UZAIR 提出了面向软件开发自动化的六层 AI 架构，利用 GPT 等大语言模型处理不同抽象层级的任务。

上述近期研究表明，AI 工具（尤其是 ChatGPT 这类大语言模型）有望加速软件开发的多个方面，包括用户故事生成、自动编码与调试。然而，重大挑战依然存在：当前模型在准确性、偏见与处理速度方面仍有局限，在安全自动化与 DevOps 分析等任务中尤为明显。这些问题说明 AI 工具尚未完全自主，也不足以完全取代人类开发者。因此，应把这类模型视为增强人类专业能力的有价值的补充资产，而非独立的解决方案。

三、研究方法

本部分说明本研究所用的流程步骤，共分八个关键阶段：问题识别、工具构建、测试、纠正与调试、工具方案的提出、用工具实施案例研究、人工实施案例研究，以及最后的结果比较与文档记录（见图 1）。

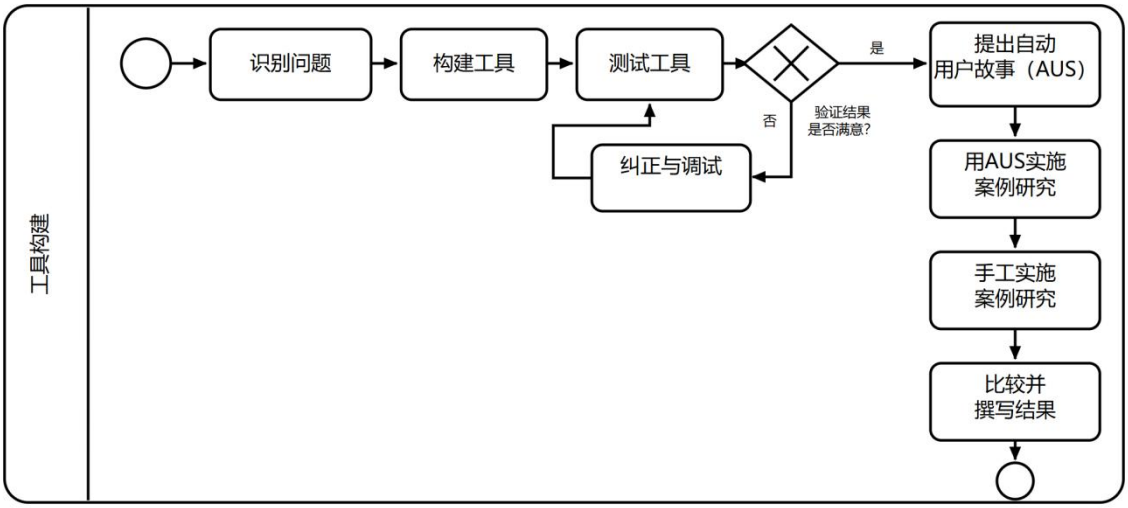

Figure 1 General Stages of the Investigation

图 1 研究的总体阶段

（一）问题识别

本研究旨在评估 ChatGPT 在软件开发各阶段自动化中的有效性，重点关注用户故事的生成以及每个故事所需工作量的估算；同时，本文还力求量化使用 AI 工具相较传统流程所节省的时间。

（二）人工智能工具的构建

该工具基于 OpenAI API 研制，具备两项主要功能：（1）为给定流程或场景生成用户故事；（2）估算执行这些用户故事所需的工作量。其开发过程如下：（1）需求收集与初步规划——明确项目目标与需求以及 AI 工具的具体要求，并确定团队角色；（2）定义用户故事——把需求拆分为用户故事；（3）待办列表优先级排序——产品负责人依据价值与复杂度对用户故事排序；（4）开发迭代——以称为冲刺（Sprint）的短开发周期推进工作，每个冲刺中团队选取一组高优先级用户故事加以实现；（5）开发——开发人员实现各项功能，包括与人工智能相关的部分。

（三）工具测试

所研制的工具经过单元测试与集成测试以确保正确运行。在每个开发周期中，依据测试结果进行纠正与调整。随后把该工具应用于所选案例研究，测量其执行时间并与人工方法相比较，以凸显使用 AI 所带来的效率提升。

（四）提出自动用户故事（AUS）

在全部测试顺利完成并获得利益相关方认可后，该工具被部署至生产环境；其运行表现受到持续监测，项目结束后还会开展回顾评审以评估流程并汲取经验。

API 内部建模：经由 API 的内部处理可能涉及多个分析与文本生成阶段。模型采用神经网络与注意力机制等自然语言处理技术来理解上下文、生成连贯文本并给出用户故事：它把输入分解为数值表示，通过多层注意力与处理理解上下文，并据此生成连贯文本；此外，模型在大量文本数据上预训练，以便生成与上下文相关且有用的回答。

（五）案例研究的实施

本阶段包含两个不同的过程：用 AUS 执行案例研究，以及人工实施案例研究。两个过程均测量各项任务的执行时间；对结果的对比分析可判定哪种过程在软件开发中更为敏捷，并量化开发时间的缩短幅度。

该实验于 2023 年 4 月至 8 月的学期内，在厄瓜多尔拉塔昆加科托帕希技术大学信息系统专业第六周期课程“软件工程基础（FIS）”中开展，所执行的活动如下。

（1）案例研究的选择：从学生建议的案例池中选出三个案例研究（见表 1）。选择过程考虑了总体需求，确保所选问题不过于复杂，且能够支撑项目的完整实施。

Table 1 Case Studies

表 1 案例研究

编号	流程
案例 1（P1）	就诊与用药提醒
案例 2（P2）	库存管理系统
案例 3（P3）	场地预订与租赁

（2）工作团队的组建：该学期共有 35 名学生选修 FIS 课程，全部学生均已修完先修课程（包括其他编程课程），并至少参与过两个开发项目。团队依据以往项目成果组建，识别出编程能力最强的学生。共组建 6 个团队，每队 4 人，并尽量保证各队实力均衡；未加入开发团队的学生则参与各项指标的测量工作。每个案例由两个团队完成：第一个团队在 AI 工具辅助下执行流程，该工具自动生成其中两个阶段，并另用 ChatGPT 生成应用程序代码，共计三个自动化阶段；第二个团队则按常规敏捷实践人工执行各项活动。

（3）开发阶段的实施：在本阶段，各团队以主要遵循 Scrum 实践的敏捷方法论开展软件开发的各个阶段（见表 2），唯一区别在于其中三个阶段使用 AUS 与 ChatGPT 实现特定流程的自动化。

Table 2 Stages of Software Development

表 2 软件开发的各个阶段

阶段	活动
初始规划	定义用户故事；对产品待办列表排序并确定冲刺时长
冲刺规划	从产品待办列表中选取本次冲刺的条目并拆分任务；估算各任务的工作量

开发	编码；打磨 AI 生成的代码；每日站会以同步团队进度与障碍
冲刺评审	演示已完成功能；评估、反馈并调整产品待办列表
冲刺回顾	为下一次冲刺调整并实施改进计划
下一冲刺准备	团队依据反馈与新需求更新产品待办列表；根据优先级与所收到的反馈选定下一冲刺的任务
产品交付	冲刺完成且产品功能可用后，进行最终评审并做好发布准备

（4）执行时间的测量：在软件开发生命周期各阶段的执行过程中，测量团队记录每个阶段的时长，记录以电子表格形式保存，时间以秒计。这一选择基于此前对 ChatGPT 的测试——此类内容的生成很少超过一分钟。在执行自动化阶段时，测量平均重复三次以保证数据可靠，记录中采用全部重复测量的平均值。

（5）对实施结果进行比较：为比较所得结果，本文采用自动化与人工两种流程下各开发阶段的执行时间测量值，以秒为单位记录，并进行多次测试以保证数据一致性；此外还计入了把自动生成的代码调整为符合各案例具体需求所需的纠正与调试时间。

四、结果与讨论

结果表明，自动执行任务与人工执行相同任务之间存在显著的时间差异。结果以秒表示，反映使用 AI 工具（AUS 与 ChatGPT）执行任务所需的极短时间；为保持一致，人工任务的测量值也换算为秒。

围绕就诊提醒与用药过程设计了一套系统，帮助患者高效跟踪就诊与服药情况。结果显示，自动化任务耗时介于 15.1 s 至 18 015.1 s 之间，而同样的任务人工完成则需 14 400 s 至 432 000 s（见表 3）。

Table 3 P1. Reminders for Medical Appointments and Medication Intake

表 3 P1．就诊与用药提醒

自动化任务	TA	TM
定义用户故事	45.18 s	14 400 s
估算各任务所需工作量	15.1 s	68 400 s
编码	18 015.1 s	432 000 s

注：TA＝自动化任务；TM＝人工任务；s＝秒。

在案例 2 中，涉及库存流程，即用于详尽、实时记录组织产品、物料或资产库存的系统；结果同样凸显出自动化与人工任务时长之间的显著差异：自动化任务耗时 3.84 s 至 14 403.84 s，人工任务则需 10 800 s 至 288 000 s（见表 4）。

Table 4 P2. Inventory System

表 4 P2．库存系统

自动化任务	TA	TM
定义用户故事	29.39 s	10 800 s
估算各任务所需工作量	3.84 s	25 200 s
编码	14 403.84 s	288 000 s

注：TA＝自动化任务；TM＝人工任务；s＝秒。

最后，在案例 3 中，涉及场地预订与租赁流程，即用于管理并简化体育场地预订与使用的应用；自动化任务耗时 8.23 s 至 14 408.23 s，人工任务则介于 10 800 s 至 288 000 s 之间（见表 5）。

Table 5 P3. Court Reservation and Rental

表 5 P3．场地预订与租赁

自动化任务	TA	TM
定义用户故事	27.1 s	10 800 s
估算各任务所需工作量	8.23 s	32 400 s
编码	14 408.23 s	288 000 s

注：TA＝自动化任务；TM＝人工任务；s＝秒。

对测量数据的分析可以推断，使用 AI 工具时执行时间最短的是工作量估算任务；该任务意义重大，因为它有助于确定实现每个用户故事所需的时间。相反，最耗时的任务是编码，即为应用或流程生成代码。值得注意的是，ChatGPT 可为多种编程语言生成代码，但开发者仍需额外时间分析、调试并调整这些代码以保证其可用；因此各团队据此另行计入了 4 至 5 小时。

表 6 全面汇总了结果，列出各案例按流程类型（半自动或人工）所耗费的总时间，并给出时间差与所实现的缩减比例。

Table 6 General Summary of Results

表 6 结果总汇

问题	TA 时间	TM 时间	差值	缩减比例
P1	446 475.38 s	874 800 s	428 324.62 s	51%
P2	205 237.07 s	507 600 s	302 362.93 s	40%
P3	244 843.56 s	518 400 s	273 556.44 s	47%

注：TA＝自动化任务；TM＝人工任务；s＝秒。

缩减幅度可观：最低为 40%，最高可达 51%。这意味着，每 10 小时的人工工作量，在 AI 工具辅助下以半自动流程执行至少可缩短 4 小时=。可见，借助 AI 工具缩短开发时间的效果十分显著。不过必须强调，AI 工具所生成的内容应由开发者加以分析，以确保其在项目中的有效性与适用性。

本研究对 ChatGPT 与 AUS 等自动化工具相较人工开发方法的表现作了细致评价，可从时间效率、代码可读性、可测试性与可维护性等若干关键维度加以分析。

时间效率：案例研究结果表明，使用自动化工具可显著缩短开发时间。三个案例中，自动化相较人工开发把时间缩短了 40%至 51%；用户故事生成、工作量估算与代码编写等任务在基于 AI 的自动化下完成得快得多。例如案例 1（就诊提醒系统）中，自动化流程完成任务耗时 15.1 s 至 18 015.1 s，而人工开发需 14 400 s 至 432 000 s；案例 2（库存管理系统）中，自动化任务耗时 3.84 s 至 14 403.84 s，人工则为 10 800 s 至 288 000 s。这一显著提升主要归因于 ChatGPT 等 AI 工具生成可用代码与内容的速度。不过研究也指出，初始结果虽生成迅速，却往往需要调试与打磨等后处理，这会略微削减总体的时间收益。

代码可读性：本研究指出，自动化的主要缺点之一是 AI 工具所生成代码的可读性较低。自动化系统虽能快速产出可用代码，但这些代码往往缺乏人工编码实践通常具备的结构性与清晰度。可读性对软件的长期维护至关重要，因为后续开发者必须能够轻松理解并修改代码。研究表明，AI 生成的代码往往含有结构不清或逻辑冗杂之处，需要开发者进一步干预。

可测试性：ChatGPT 等自动化工具虽能快速生成代码，但这些代码往往需要额外测试以确认其满足系统功能需求。由于 AI 工具存在固有局限、可能无法完全理解复杂需求，所生成代码常含有错误或遗漏，必须在测试中查出并纠正。开发者往往需额外投入时间对 AI 生成代码进行单元测试与集成测试，这凸显出测试阶段人工监督的重要性。

可维护性：可维护性指软件在其生命周期内被更新或修改的难易程度。研究强调，自动化工具虽擅长快速产出代码，但这些代码未必针对长期可维护性作过优化。人类开发者通常遵循模块化且文档完善的编码实践，便于日后修改；相比之下，AI 生成的代码可能前后不一致且缺乏模块化，更难适应新需求或变更。

对上述维度的整体评价，使得对自动化与人工开发总体有效性的判断更为可靠。本研究的结论表明，自动化工具虽显著提升效率，却并未消除为保证代码质量而进行人工干预的必要；人工监督与自动生成相结合能产生最可靠的结果。未来研究应聚焦于提升 AI 工具的准确性以减少人工干预，并引入实时反馈机制；自动化的范围还应扩展至测试与部署等软件生命周期的其他阶段。从长远看，AI 模型须在更具针对性的数据集上训练，以改善上下文理解并减轻偏见。

五、结论

研究结果表明，利用 AI 工具实现任务自动化可使开发时间缩短 40%至 51%，具体取决于场景。但必须认识到，这些工具的输出需要调试、打磨与适配；因此，整体开发过程以及由 AI 工具辅助的任务实际上构成了半自动化工作流，需要持续监督并辅以补充措施，才能确保所生成结果在软件开发项目中的准确性与适用性。

在编码阶段，为使用 ChatGPT 生成的代码片段，需要作多处调整，主要原因在于初始输出存在不准确之处。为提高精度，必须把提问细化得更为具体以获得更准确的回答；此外还需开展大量审查与调试，才能使生成的代码适于实际应用。这些额外步骤自然增加了耗时，并已计入任务总时长。

AUS 工具利用自然语言处理算法生成文本内容，能够实现其预定功能，如产出用户故事与估算开发工作量；但所生成的内容常常需要审查与调试，因为它在特定情境下往往不够完整与精确。要提升此类工具的可靠性，还需进一步研究，尤其是探索 AI 增强软件工程与 Software 2.0 等新兴范式。

ChatGPT 等 AI 驱动工具虽在软件开发早期阶段带来显著的时间节省，但在代码质量、可读性与可维护性方面存在明显取舍。人工开发对于确保长期可靠性仍不可或缺，因为它产出的代码结构更好、更易读也更易维护。因此，有效的开发策略应把自动化的效率与人类专业判断的精确性结合起来。

参考文献

[1] 李全兴, 吴国林. 论人工智能的理解——以DeepSeek-R1等新兴人工智能为例. 华南理工大学学报（社会科学版）, 2025, 27(2): 9-18.

[2] 毛新军. 自主机器人软件工程的研究综述. 计算机学报, 2021, 44(8): 1661-1678.

[3] Harika J, Baleeshwar P, Navya K, et al. A review on artificial intelligence with deep human reasoning. 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022: 81-84.

[4] Yadav A, Sondhi H. Systematic literature review on sustainable marketing and artificial intelligence. Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development, 2023: 583-588.

[5] 张军爱, 刘海军, 李高峰, 等. 人工智能ChatGPT赋能初中化学教学设计的创新功能与价值定位. 化学教育（中英文）, 2025, 46(5): 86-88.

[6] Santhanam S, Hecking T, Schreiber A, et al. Bots in software engineering: a systematic mapping study. PeerJ Computer Science, 2022, 8(3): e866.

[7] Allam H, Dempere J, Akre V, et al. Artificial intelligence in education: an argument of ChatGPT use in education. 2023 9th International Conference on Information Technology Trends (ITT), 2023: 151-156.

[8] 陈阳, 姜凯星. 国内ChatGPT应用于教育领域的研究现状、分析与启示——基于CiteSpace文献计量. 中小学电教（综合）, 2025(12): 20-26.