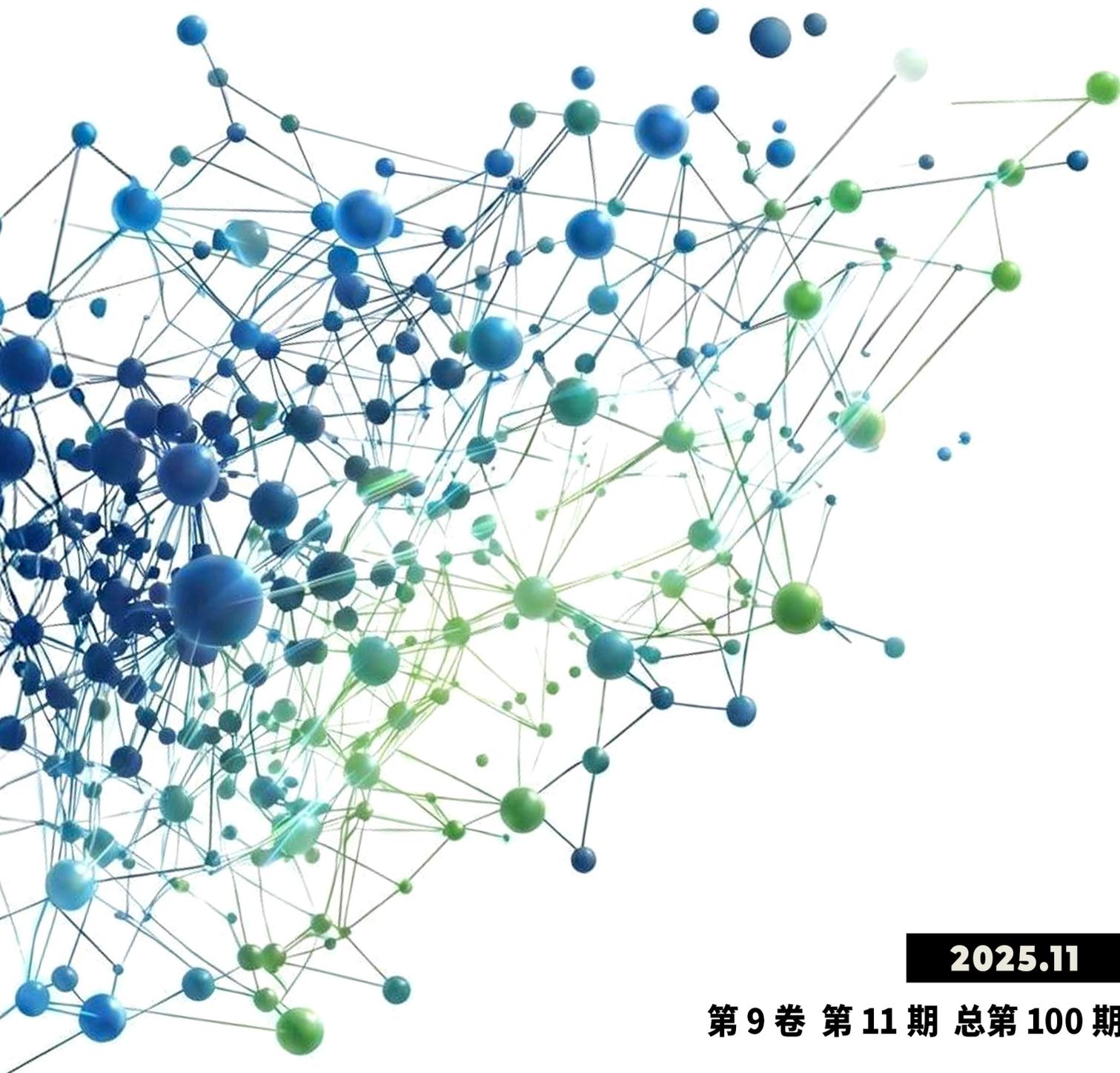


# 信息世界： 连接与创新

The Information World: Connection and Innovation



2025.11

第9卷 第11期 总第100期

# 信息世界： 连接与创新

The Information World: Connection and Innovation



## 出版社信息

主管：香港星源出版社

主办单位：香港星源出版社

主编：裴景川

执行主编：姜若尘

社内编辑：

程牧言 顾廷舟 周奕 唐昊临

杜景尧 邢若航 钟昱廷 林尧

罗修远 苏承远 贺云策 林峻尧

陆昊然 任知衡 罗晋 傅云衡

温奕川

网址：<https://hksspub.com/>

电话：+852 6855 8145

邮箱：[hksspub2022@163.com](mailto:hksspub2022@163.com)

刊期：月刊

STAR SOURCE PUBLISHING  
香港星源出版社



6378 9010

# 目 录

# CONTENTS

智能数据编排框架在分布式环境中的构建与实践 .....	蒋牧川	001
高可扩展流处理系统的容错机制与性能调优研究 .....	霍景尧	008
面向复杂业务场景的微服务治理策略与架构演进 .....	谭书远	015
基于多任务学习的模型协同训练方法研究 .....	柳承晗	022
跨平台应用生命周期管理体系的自动化实现路径 .....	郑泽川	029
企业级数据资产治理平台的架构设计与实施分析 .....	乔沐辰	036
端侧推理模型的轻量化压缩与性能优化研究 .....	陆修齐	043
分布式图数据库在智能知识管理中的工程应用 .....	顾承逸	043
面向高并发业务的缓存一致性控制机制研究 .....	梁景澄	050
云边协同环境下任务卸载策略与资源调度优化 .....	宋知远	056
大模型推理服务的弹性扩缩容架构与负载预测方法 .....	冉启航	063
客户端系统日志分析与异常检测自动化框架设计 .....	贺昱辰	069
基于人工智能的客户端系统工程效率提升研究 .....	顾言舟	075
跨终端同步系统中的数据一致性保障机制研究 .....	苏昱廷	080
容器化部署环境中的安全隔离与访问控制策略 .....	叶景曜	088
基于强化学习的智能运维决策模型与实践验证 .....	邱若川	093
面向实时渲染的客户端图形管线优化策略研究 .....	韩靖衡	100
数据隐私保护框架在跨域协作平台中的工程实现 .....	林奕辰	106
面向可观测性的分布式追踪系统设计与性能评估 .....	唐予衡	111
智能软件工程生态的协同演进与技术趋势展望 .....	魏子骞	119

# 基于人工智能的客户端系统工程效率提升研究

顾言舟  
(江苏省 东南大学 211189)

## 摘要:

在客户端系统规模持续扩大、应用场景日益复杂的背景下,客户端工程在研发过程中逐步暴露出协作成本上升、流程协调困难以及工程效率受限等问题。人工智能技术在软件工程领域的应用,为改善客户端系统工程效率提供了新的技术路径。本文以客户端系统工程为研究对象,从工程效率视角出发,分析人工智能技术参与客户端工程过程的应用方式及其对工程效率的影响机制。

在梳理客户端系统工程组织方式与效率影响因素的基础上,结合人工智能在软件工程中的典型应用范式,对客户端工程中高成本环节和关键协作场景进行了系统分析。通过对人工智能在需求管理、开发支持和协作决策等环节中应用特征的讨论,阐明其在减少重复性工作、辅助工程决策以及改善协作透明度方面的作用方式。在此基础上,构建了面向工程效率的客户端系统架构设计思路,分析人工智能能力在系统架构中的嵌入方式及其对工程流程的支撑机制。研究表明,人工智能对客户端系统工程效率的影响具有情境依赖特征,其作用效果受到工程规模、协作复杂度和系统架构设计的共同制约。

关键词:人工智能辅助软件工程;客户端系统架构;工程效率优化;智能化研发支持;软件工程自动化

## 第一章 客户端系统工程复杂化背景下的效率问题分析

### 1.1 客户端系统规模扩展与工程复杂性演变

客户端系统正逐步从单一平台应用向多平台、多终端适配演进,涵盖桌面端、移动端、Web等多种形态,并融合高频交互、实时响应与复杂前后端协同特性。系统规模扩展引发依赖管理难度上升、模块间耦合增强、构建流程冗长等问题,工程复杂性持续累积。

高度迭代的产品开发模式对工程的响应速度提出更高要求,客户端团队面临需求快速变更、交付节奏压缩的常态。传统线性开发流程难以有效应对这一变化,工程管理效率与协作稳定性成为制约研发质量与成本控制的关键因素。

### 1.2 客户端系统工程效率约束的具体表现

多平台并行开发中存在大量重复性工作,代码逻辑难以共享,接口标准不统一,导致任务分配效率低,逻辑冗余频发。工程工具链的分散部署使开发、测试、上线等环节各自独立,协作流程断裂,工程透明性与追踪能力下降。

研发过程中仍依赖大量人工决策,包括任务调度、缺陷排查、版本控制等,导致决策效率和准确性受限。在持续集成、快速迭代要求下,这些低效环节易引发开发进度波动、产出质量不稳,工程能力与系统交付能力之间形成张力。

### 1.3 人工智能技术介入客户端工程过程的现实动因

人工智能技术在代码生成、测试覆盖、缺陷预测等方面已展现出提升效率与降低人为干预的潜力。在客户端系统中,AI可辅助界面布局优化、性能瓶颈识别、行为路径建模与需求变更预测,构建更具数据驱动特征的研发流程。

当前研究已证明,人工智能在软件工程中的应用正在从辅助工具向过程嵌入机制转变,具备良好的可拓展性与落地基础。在研发节奏加快、系统复杂性提升的背景下,AI赋能客户端工程具备现实驱动与方法支撑,有助于构建面向效率优化的智能化研发体系。

## 第二章 客户端系统工程与人工智能应用的理论基础

### 2.1 客户端系统工程的组织形态与效率影响因素

客户端系统工程作为软件开发的重要组成部分,其组织形态在近年来发生显著变化。随着多终端、多平台需求的增长,客户端研发从原本的单端开发模式逐步演化为高度协作、多团队并行推进的复杂体系。这种演变使得客户端工程不仅需要解决功能实现问题,还必须面对组件复用、版本迭代、平台适配、质量保障等多重挑战。

在现代客户端系统中,工程结构往往呈现出分层

模块化特征，包括基础组件层、业务逻辑层、平台接口层和用户交互层等。每一层的开发任务通常由不同小组承担，形成多个角色之间的协作链条。工程效率受制于这些协作关系的协调程度，而非单一技术环节的优化。例如，接口不一致、版本依赖冲突、文档不完善等问题，都可能在工程中造成重复沟通、返工和时间损耗，最终影响整体进度。

此外，客户端工程与服务端开发、产品管理、质量保障等环节高度耦合，使得任何一个环节的效率问题都可能引发系统性瓶颈。尤其在持续集成与快速迭代机制下，效率不仅关乎代码产出速度，更体现于流程流转的连贯性和错误处理的及时性。从整体工程组织形态出发，深入理解协作复杂性与效率构成要素，是实现后续智能优化的基础。

## 2.2 人工智能在软件工程中的应用范式

人工智能在软件工程中的应用日益丰富，尤其在开发流程日益数据化、自动化的背景下，其介入方式已超越传统工具支持的范畴。当前的研究与实践主要集中在任务预测、代码生成、异常检测、测试用例生成、代码评审建议等方面，形成了一系列基于数据驱动的工程辅助机制。

在任务层面，人工智能可以通过历史版本数据分析任务关联与依赖，辅助进行更合理的任务分配与进度预测。在编码阶段，基于大模型的代码补全与生成工具已进入实际应用，为开发者提供实时智能建议，提高编码速度的同时降低语义错误率。在测试与运维方面，AI 可识别回归路径、生成测试脚本、检测异常指标，帮助团队提前发现潜在缺陷。

客户端系统开发本身具有交互性强、模块变化频繁、平台适配复杂等特点，因此更依赖上下文理解与非结构化信息处理能力。人工智能在这些场景下不仅承担辅助执行的角色，更通过模型的预测能力介入协作协调与质量控制。已有研究显示，将 AI 嵌入客户端工具链，有助于构建具备知识感知与反馈能力的开发环境，从而提升整体工程响应效率 [1]。

## 2.3 工程效率的内涵界定与分析维度

工程效率在客户端系统背景下不应仅以完成速度衡量，其内涵更涉及资源利用、过程稳定性、错误恢复能力与知识传承机制等多个层面。以客户端项目为例，一个高效的工程不仅要在短时间内完成既定开发任务，还需保证成果的可维护性、可扩展性和跨团队协作的顺畅性。

当前关于工程效率的分析框架多从流程视角出发，强调任务流转速度和缺陷闭环周期。但在复杂系统中，仅以时间维度评估往往难以全面反映实际效率

状态。例如，任务执行速度虽快，但因前置信息不完整或协作配合不到位，导致交付结果质量低、返工频繁，其整体效率实则低于看似慢但稳健的流程。

更准确的效率评估，应结合任务间依赖结构、协作路径长短、工具链耦合度、决策透明性等因素进行建模。客户端系统工程中经常出现的场景包括多模块并行开发、版本管理复杂、UI/UX 迭代频繁等，这些都要效率评估方式具备情境感知能力。

## 2.4 本文研究的技术视角与工程分析框架

在探讨如何借助人工智能提升客户端工程效率时，必须首先明确技术介入的角度。本文立足于客户端系统的工程实践，提出一种面向协作密度高、模块交互复杂的开发环境的智能分析框架。该框架意在通过数据驱动手段捕捉影响效率的关键因素，并以智能化方法对开发流程进行优化建议输出。

整体框架由三部分构成。第一部分是工程过程建模，通过静态代码分析、任务调度图谱提取、历史协作行为记录等方式，构建描述开发活动的结构化模型。第二部分是模型驱动的智能分析，结合机器学习与图模型方法，对关键流程节点的潜在风险、任务拥塞点和模块依赖问题进行预测。第三部分是辅助决策与反馈机制，将分析结果映射为具体操作建议，如任务重排序、接口优化建议、测试策略推荐等，以插件或交互组件形式集成至客户端开发平台。

技术路径上，本文重点关注具有可解释性与轻量化特点的人工智能方法，避免引入高训练成本或黑盒机制所带来的工程风险。在数据维度上，结合版本控制日志、代码提交记录、评审反馈与自动化测试结果，构建多源协同数据视角，使模型分析更贴近工程实际情境 [4]。

## 第三章 人工智能参与客户端系统工程的应用场景分析

### 3.1 客户端系统工程中的高成本环节识别

客户端系统的开发流程覆盖需求理解、UI 设计、功能实现、测试验证及持续集成等多个阶段，其中多环节呈现出明显的资源投入密集和效率瓶颈特征。与传统后端开发相比，客户端工程更依赖于交互逻辑的细致实现和多平台适配支持，这一特性在面对频繁迭代和版本更新时，容易导致重复性劳动增加。

具体来看，UI/UX 设计与实现之间的信息转换往往存在断层，设计意图难以准确传达给开发人员，造成界面还原效率低下。其次，在任务协同阶段，功能模块间的接口变更频繁、依赖项耦合度高，也会使得代码合入和测试流程受阻，增加沟通与协调成本。此

外，由于客户端代码对用户体验要求较高，功能上线前需反复验证交互逻辑、动画流畅性和性能边界，测试成本居高不下，成为制约工程效率的重要因素。

### 3.2 人工智能在客户端工程关键环节中的应用场景

人工智能在客户端系统开发中的应用，已从早期的语法层次辅助，逐步拓展至任务分解、代码生成、测试建议等多个层面。在编码阶段，基于大模型的智能补全工具可结合上下文环境实时给出函数结构、组件参数和交互逻辑建议，提升开发者操作效率的同时降低重复劳动。

在需求与设计衔接环节，部分研究尝试将自然语言处理能力引入设计意图识别与代码转换中，通过设计与控件识别模型，将 UI 设计稿自动转化为前端组件代码，缩短设计 - 实现之间的距离。在复杂组件复用场景中，AI 亦可通过代码克隆检测与接口模式识别，自动推荐可复用模块及其集成方式，缓解重复造轮子的现象。

测试环节是人工智能应用较为集中的领域之一。自动化测试用例生成、用户行为路径预测、UI 异常识别等技术均已进入实用阶段，尤其适用于处理覆盖面广但验证逻辑相对明确的场景。在异常日志分析与性能瓶颈定位方面，机器学习模型能够通过聚类和异常检测方式辅助开发者快速定位潜在缺陷 [2]。

### 3.3 智能辅助工具对客户端工程协作模式的影响

人工智能工具的引入，不仅在具体任务上提升了开发效率，更在团队协作模式上引发了潜在转变。在传统客户端工程中，任务拆解、接口协商与版本同步多依赖于人工经验，协作效率受限于成员之间的熟悉程度与沟通频次。引入智能分析系统后，任务依赖图谱可以自动构建，接口变更影响范围能够实时标注，协同精度与响应速度得以显著提升。

同时，AI 工具的嵌入使得部分原本依赖人工判断的环节实现流程化、数据化。例如，通过对历史任务数据建模，系统可以预测任务难度与风险，辅助项目负责人进行资源分配与进度管控。在代码评审环节，智能工具可以提前发现潜在缺陷并推荐修改建议，减少评审工作量，提升协作透明度。

人机联合的协作形态正在取代传统分工模式，使得团队成员能够在更高层次聚焦系统设计与问题抽象，而将重复性操作交由智能系统承担，从而推动客户端研发流程向智能化、闭环化方向演进 [4]。

### 3.4 应用场景下工程效率提升的潜在约束

尽管人工智能在客户端工程中展现出明显的效率优势，但其实际应用仍受到多种约束条件的影响。在

数据基础方面，多数 AI 模型依赖历史代码、版本记录与评审数据进行训练，而部分团队由于流程规范程度不足，难以提供高质量训练数据，导致模型预测准确性下降。

此外，客户端工程涉及多平台适配、设备性能差异、UI 多样性等复杂背景，对 AI 模型的泛化能力提出更高要求。一个在特定架构或语言环境下表现优异的智能工具，可能难以直接迁移至其他平台或技术栈。在工程实践中，这类工具往往需结合具体上下文进行二次训练或规则微调，增加部署与维护成本。

另一方面，AI 工具介入还会带来团队角色与流程的重构需求。开发人员需具备一定的数据理解与模型反馈能力，团队也需调整原有协作规范以适应智能辅助机制的融入。在转型初期，这可能引发一定的效率波动与协作摩擦。因而，如何平衡工具能力与工程稳定性，建立适配的工具引入策略，是推动客户端系统智能化发展的关键议题。

## 第四章 面向工程效率提升的客户端系统架构设计思路

### 4.1 工程效率导向下的客户端系统架构设计目标

在客户端系统工程不断复杂化的背景下，系统架构的设计目标逐渐从单一的功能实现转向对研发过程的整体支撑。工程效率导向的架构设计强调在满足业务需求和运行稳定性的前提下，尽可能降低开发、协作与维护过程中的隐性成本，使系统能够在高频迭代环境中保持良好的演进能力。

这一目标要求架构具备清晰的结构层次与合理的职责划分，使开发人员能够在明确边界内开展工作，减少因依赖不清或接口不稳定带来的协作阻碍。同时，架构需要对变化保持足够的包容性，使需求调整、功能扩展和技术升级不至于引发大范围重构，从而降低工程调整的代价。

### 4.2 人工智能能力在客户端系统架构中的嵌入方式

将人工智能引入客户端系统架构，并非简单地叠加智能工具，而是要求在架构层面为智能能力的运行与反馈提供稳定支点。人工智能只有在与系统结构深度结合的情况下，才能持续发挥对工程效率的促进作用。因此，其嵌入方式应围绕系统数据流、控制流和协作流展开，使智能分析结果能够直接作用于工程行为。

在架构层面，人工智能能力通过持续获取开发过程数据，对任务状态、代码变更和协作关系进行建模，从而形成对工程运行状态的动态认知。这种认知并不

以替代人为决策为目标，而是通过提供预测结果和趋势判断，辅助开发人员与管理者进行更为理性的决策。当人工智能的输出能够直接影响构建流程、测试策略或协作节奏时，其价值便从“辅助工具”转变为“结构性能力”，成为架构运行逻辑的一部分。

#### 4.3 面向工程支撑的系统功能模块划分

面向工程效率的客户端系统架构，其功能划分应以支撑研发活动为核心逻辑，而非仅围绕业务功能展开。模块划分的关键不在于数量的多少，而在于是否能够清晰映射工程流程中的核心活动，并在不同活动之间建立稳定而低成本的协作关系。

在这种架构设计思路下，系统功能的组织方式更强调对开发、协作、验证和反馈等工程行为的承载能力。各功能模块在逻辑上相互独立，但在运行过程中通过统一的数据通道和接口规范形成协同关系，从而避免模块之间的紧耦合。通过这种方式，系统既能够支持局部能力的独立演进，又不会破坏整体工程流程的连续性，为后续引入新的工具或智能能力保留空间。

#### 4.4 架构设计对客户端工程流程的支撑机制

客户端系统架构对工程流程的支撑，并非体现在对某一环节的局部优化，而是通过对整体流程结构的塑造，影响研发活动的组织方式与执行效率。合理的架构设计能够将复杂的工程流程拆解为一系列可感知、可控制的阶段，使系统具备对流程状态的理解能力。

在这一机制下，架构不仅承担运行时支撑角色，还通过与开发工具链和自动化系统的联动，参与到工程流程的调度与反馈之中。代码提交、构建触发、测试执行和结果分析等行为被纳入统一的架构视角，形成闭环运行逻辑。这种闭环使工程活动的结果能够反向作用于后续决策，为流程调整和效率优化提供依据。

面向工程效率的客户端系统架构并不是静态的技术结构，而是一种与研发过程持续互动的动态体系。通过在架构层面嵌入对流程的感知与调节能力，客户端系统能够在复杂协作环境中保持稳定运行，并为工程效率的持续提升提供长期支撑。

### 第五章 人工智能辅助下的工程效率影响机制分析

#### 5.1 人工智能对客户端工程决策效率的影响

客户端系统工程中的决策活动贯穿研发全过程，包括任务拆分、资源分配、接口调整、版本发布时机等多个层面。这类决策通常具有信息不完全、时效性要求高和依赖经验判断等特征，在工程复杂度不断提升的背景下，单纯依赖人工判断容易导致响应滞后或

决策偏差。人工智能技术的引入，为提升工程决策效率提供了新的技术路径。

通过对历史项目数据、代码演进轨迹和协作行为的持续学习，人工智能模型能够对工程状态形成结构化认知，并在此基础上提供预测性支持。例如，在客户端模块频繁迭代的场景中，模型可基于过往变更记录预测接口调整可能引发的影响范围，从而辅助开发人员在决策阶段评估潜在风险。这种基于数据驱动的决策支持机制，有助于缩短决策周期，提高决策一致性，使工程管理从经验导向逐步转向模型导向 [3]。

需要强调的是，人工智能在工程决策中的作用并非替代人工判断，而是通过降低信息获取与分析成本，提高决策的可达性与可解释性。尤其在多团队协作环境中，智能分析结果能够作为统一参考依据，减少沟通成本和理解偏差，从机制层面提升客户端工程的整体决策效率。

#### 5.2 客户端工程中重复性任务的智能化处理方式

客户端系统工程中存在大量重复性任务，如代码模板编写、接口适配、基础组件调用、测试用例构建等。这些任务在不同项目或版本中高度相似，但长期以来主要依赖人工完成，既占用研发资源，又容易引入不一致问题。人工智能在这一领域的介入，使重复性任务具备被系统化处理的可能。

在编码阶段，基于大模型的代码生成与补全能力能够根据上下文自动生成符合规范的代码片段，减少开发人员在基础实现层面的投入。在测试阶段，智能化测试生成工具可以依据功能变更自动调整测试范围，避免人工维护测试脚本所带来的高成本。这类处理方式通过将重复劳动转化为模型推理过程，有效释放开发人员精力，使其更多关注系统设计 with 问题抽象。

从工程机制角度看，重复性任务的智能化处理不仅提升了单次任务效率，更在长期运行中促进了工程规范的统一。模型在学习过程中不断强化对既有规范和最佳实践的遵循，使系统输出趋于一致，有助于降低客户端工程中因个人差异带来的质量波动。

#### 5.3 工程效率改善的适用条件与边界讨论

尽管人工智能在提升客户端工程效率方面展现出显著潜力，但其效果并非在所有场景下均能充分发挥。工程效率改善依赖于若干前提条件，包括工程数据的可获取性、流程规范程度以及系统架构对智能能力的适配水平。在缺乏稳定数据来源或工程流程高度非结构化的情况下，人工智能模型难以形成有效学习基础，其辅助效果将受到明显限制。

客户端系统具有平台多样、技术栈差异大、业务变化频繁等特点，这对人工智能模型的泛化能力提出

较高要求。当模型训练所依赖的场景与实际应用环境差异较大时，其输出结果可能失去参考价值，甚至对工程决策产生误导。

从工程管理角度看，人工智能工具的引入还可能引发角色分工与流程调整的问题。开发人员对模型结果的信任程度、团队对智能建议的接受机制，都会影响技术落地效果。在缺乏合理引导的情况下，过度依赖智能系统可能削弱工程人员的主动判断能力，形成新的风险点。因此，工程效率提升应建立在人机协同的基础之上，通过明确边界和责任分配，使人工智能成为稳定、可控的工程支撑力量。

人工智能对客户端工程效率的影响具有明显的条件依赖性。只有在工程流程相对规范、数据基础较为完备、架构设计具备良好扩展性的前提下，其效率提升机制才能持续发挥作用。

## 第六章 工程视角下的研究结论与系统演进思考

### 6.1 人工智能辅助客户端工程的实践特征总结

通过对客户端系统工程复杂性与人工智能介入方式的系统分析可以发现，人工智能在客户端工程中的实践形态呈现出明显的工程导向特征。其作用并非集中于单一技术环节，而是通过对开发过程数据的持续感知与分析，逐步嵌入工程运行逻辑之中，从而对决策效率、任务执行方式和协作结构产生影响。

在实际工程场景中，人工智能更倾向于承担“辅助者”而非“替代者”的角色。其核心价值体现在对高频决策场景的支持、对重复性工程活动的消解以及对流程状态的持续反馈。通过这种方式，工程活动中的不确定性得以降低，开发过程的可预期性和一致性有所增强。与此同时，人工智能能力的有效发挥高度依赖工程流程的规范程度和系统架构的开放性，这使其呈现出明显的工程适配特征，而非通用技术能力。

### 6.2 工程效率视角下的系统设计结论

从工程效率视角审视客户端系统架构设计，可以得出一个较为明确的结论，即效率提升并非来源于单点技术突破，而是依赖于系统结构、流程机制与智能能力之间的协同作用。客户端系统若仅在局部引入人工智能工具，而缺乏对整体工程流程的重构，其效率改善往往难以持续。

研究表明，将工程效率作为架构设计的内在目标，有助于在系统层面提前约束模块边界、协作路径和数据流向，从而为人工智能能力的嵌入提供稳定载体。在这一框架下，架构不再只是运行环境的支撑结构，而成为工程活动的组织者与调节者。人工智能通过与架构协同运行，使工程效率的提升由“人为经验积累”逐步转向“系统能力演进”，体现出明显的工程系统化特征。

### 6.3 研究局限与后续研究方向

尽管本文从工程视角对人工智能参与客户端系统工程的效率影响进行了系统分析，但仍存在一定局限性。研究主要基于理论分析与工程机制抽象，尚未结合具体企业级项目进行长期、定量的实证验证，相关结论在不同规模、不同技术栈的工程环境中仍需进一步检验。

人工智能模型本身的可解释性与稳定性问题，在工程实践中仍具有现实挑战。模型输出结果在复杂工程情境下的可信度、适用范围以及责任边界划分，均有待更深入的研究和规范化探索。这些问题不仅涉及技术层面，也与工程管理和组织协作方式密切相关。

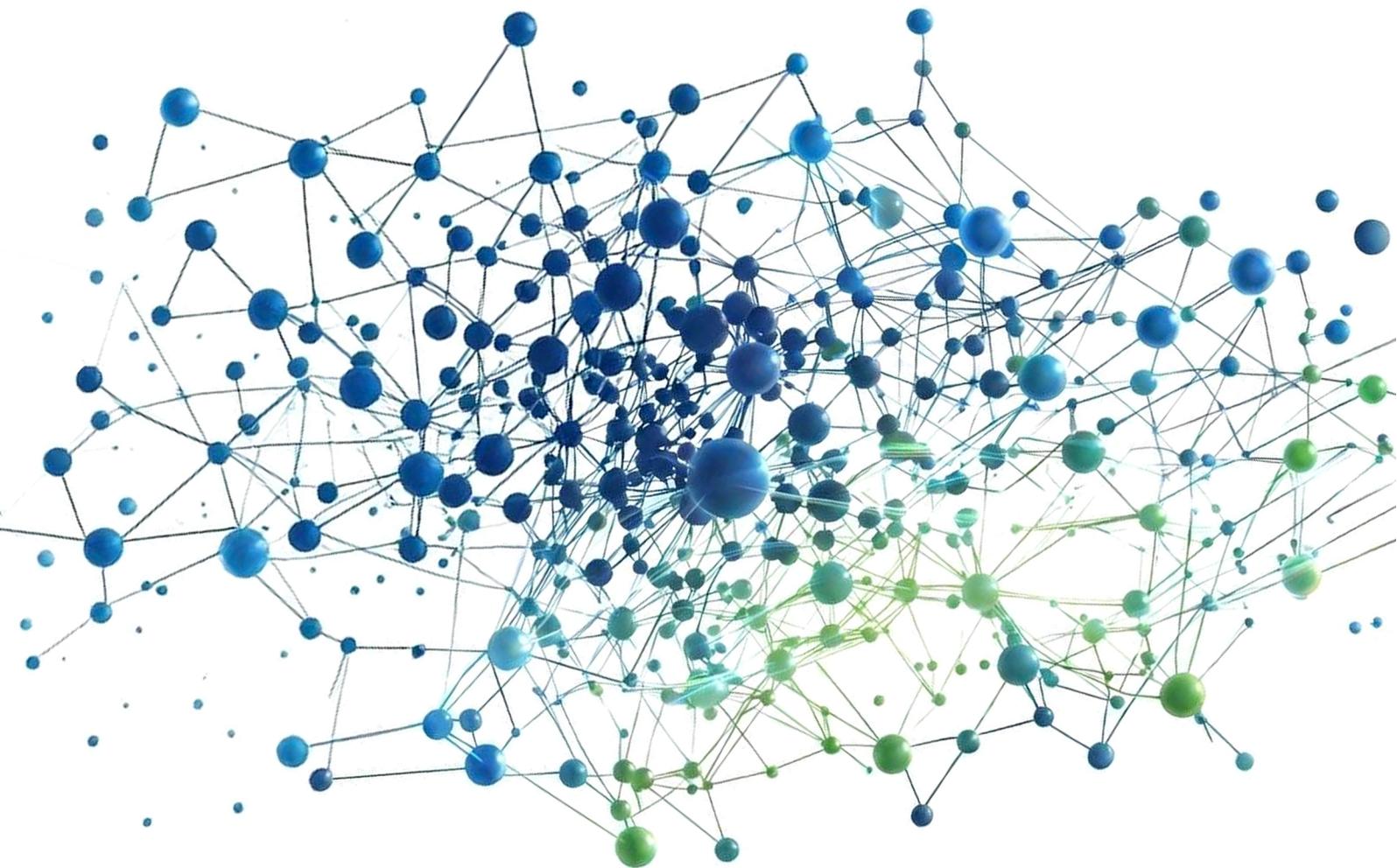
未来研究可在本文分析基础上，结合真实客户端工程数据构建实验平台，对人工智能介入前后的工程效率变化进行对比评估。同时，可进一步探索面向不同工程规模和团队结构的智能化架构演进路径，使人工智能在客户端系统工程中的应用更加可控、可持续，并具备更强的工程适配能力。

### 参考文献：

- [1] 吴强. 人工智能在软件工程中应用研究 [J]. 网络安全技术与应用, 2025(08):52-54.
- [2] 邢颖. 基于可解释性人工智能的软件工程技术方法综述 [J]. 计算机科学, 2023, 50(5):9-18.
- [3] Crawford T, Duong S, Fueston R, et al. AI in Software Engineering: A Survey on Project Management Applications[J]. arXiv, 2023.
- [4] Cao S, Sun X, Widyasari R, et al. A Systematic Literature Review on Explainability for Machine/Deep Learning-based Software Engineering Research[J]. arXiv, 2024.

# 信息世界： 连接与创新

The Information World: Connection and Innovation



STAR SOURCE PUBLISHING  
香港星源出版社



6378 9010