

# 基于用户需求的计算机软件设计方法探析

梁雪芹 李爱超

河北科技学院 河北唐山 063000

**摘要：**在数字化转型加速的时代背景下，计算机软件设计的核心竞争力已从技术实现转向用户需求满足。本文系统探讨基于用户需求的软件设计理论框架与实践路径，通过梳理用户需求分析的核心方法、设计原则及技术实现策略，结合典型案例验证方法有效性，提出融合敏捷开发与用户中心设计的创新路径。研究表明，精准的需求捕获、系统化的设计架构及持续的用户反馈机制，是提升软件可用性与商业价值的关键要素。

**关键词：**用户需求分析；软件设计方法；敏捷开发；用户中心设计；需求验证

## 引言

在数字经济浪潮下，计算机软件成为产业变革核心，但软件项目供需失衡问题突出。Statista数据显示2024年全球软件市场规模超2.5万亿美元，而Standish Group报告指出，仅32%的软件项目能完全满足用户需求，近半数存在功能冗余或需求遗漏问题，严重影响用户体验与企业竞争力。国外自20世纪80年代起发展用户中心设计、敏捷开发等理论，国内也在需求建模等方面取得进展，但现有研究仍存在动态需求捕获不足、跨平台适配方法缺失、量化评估体系不完善等问题。本研究采用“理论建构—案例验证—方法创新”框架，运用文献计量、多案例分析等方法，结合DSRM范式构建需求驱动模型。创新点在于提出需求优先级评估矩阵，构建可视化三维设计框架，设计包含眼动追踪等技术的立体化需求验证体系。

## 一、用户需求分析的理论基础

### （一）用户需求的定义与分类

用户需求是用户对目标软件在功能、性能、体验等方面的期望集合，可分为三个维度：

**功能性需求：**明确的业务操作要求，如电商系统的订单管理、支付结算功能（ISO/IEC 25010 标准定义）。

**非功能性需求：**系统运行的质量属性，包括性能（响应时间<200ms）、安全性（数据加密等级 AES-256）、可用性（容错恢复时间<30秒）等。

**隐性需求：**用户未明确表达的潜在期望，如移动应用的手势交互习惯、界面视觉舒适度。

Kano模型将需求划分为**必备需求**（Must-be Requirements）、**期望需求**（One-dimensional Requirements）、**魅力需求**（Attractive Requirements），为需求优先级排序提供理论依据。例如手机银行的指纹登录属于期望需求，而基于用户行为的智能风险预警则属于魅力需求。

### （二）用户需求分析的重要性

需求分析是软件设计的逻辑起点，直接决定项目成败。经典案例显示：某物流管理系统因未识别仓库人员的移动端操作需求，导致现场扫码效率降低40%，最终项目延期18个月，成本超支65%。NASA的火星气候轨道器事故，根源在于需求规格说明书中单位转换错误（英制/公制混淆），造成1.25亿美元损失。这些案例印证了Gartner的观点：项目后期修改需求的成本是前期的100倍以上。

### （三）用户需求获取的方法体系

方法类型	具体工具	适用场景	优势	局限性
直接沟通	用户访谈、焦点小组	需求探索阶段	深度挖掘隐性需求	样本偏差风险
观察法	现场跟测、屏幕录制	操作场景还原	捕获真实行为数据	耗时耗力
数据驱动	日志分析、A/B测试	需求验证阶段	量化需求优先级	依赖历史数据
原型法	低保真原型、高交互Demo	需求确认阶段	可视化需求表达	过度关注界面细节

实际应用中需采用混合方法：某教育软件团队通过120小时课堂观察发现教师的分组教学需求，结合200份

**作者简介：**梁雪芹，女，1987，安徽省宿州市人，汉，工程师，本科，从事计算机科学与技术方向研究。

教师问卷验证后，开发出智能分组管理模块，使课堂效率提升35%。

### 三、基于用户需求的软件设计原则

#### (一) 用户中心设计原则 (User-Centered Design, UCD)

核心思想是“在设计过程中始终以用户为中心”，包含四个实施步骤：

1. 用户建模：构建包含人口统计学、行为特征、心理动机的用户画像。如为老年用户设计健康管理APP时，需特别关注视力衰退导致的字体大小需求（建议 $\geq 16\text{px}$ ）。

2. 场景分析：通过用户旅程地图 (User Journey Map) 描绘使用场景，识别痛点与机会点。某银行在分析企业客户的跨境转账流程时，发现多系统切换导致的信息重复录入问题，进而开发出统一数据中台。

3. 交互设计：遵循尼尔森可用性十原则，如系统状态可见性（操作反馈即时呈现）、防错原则（关键操作二次确认）。微信的红包功能通过动态红包雨动画和清晰的拆包反馈，将用户误操作率降低至0.3%。

4. 可用性测试：采用think-aloud法（出声思考）进行用户测试，记录任务完成时间、错误率等指标。某电商APP通过可用性测试发现，地址管理模块的层级深度导致38%的用户放弃下单，优化后转化率提升22%。

#### (二) 模块化设计原则 (Modular Design)

将软件系统分解为独立可复用的模块，通过定义清晰的接口规范实现松耦合架构。典型应用如MVC（模型—视图—控制器）架构，将业务逻辑、用户界面、数据处理分离，使模块复用率提升60%以上。某ERP系统采用模块化设计，将采购、库存、销售模块封装为微服务，支持企业按需组合功能，实施周期缩短40%。模块划分需遵循“高内聚、低耦合”原则，运用功能聚类分析（如基于UML包图）和依赖关系矩阵进行优化。

#### (三) 可扩展性设计原则 (Scalable Design)

通过弹性架构设计应对用户规模与业务需求的变化，包含：

1. 水平扩展：分布式系统通过增加节点提升处理能力，如淘宝双11期间将应用服务器扩展至10万台，支撑每秒58.3万笔订单创建。

2. 垂直扩展：通过功能分层（表现层—业务层—数据层）实现模块独立扩展，某金融风控系统将规则引擎模块分离，支持策略更新不影响核心交易。

3. 接口预留：采用RESTful API设计标准，为未来

功能扩展预留接口。微信开放平台通过千余个API接口，支持超2000万开发者接入，构建庞大生态系统。

设计时需运用容量规划模型（如Little定律）预测性能瓶颈，某出行平台通过建立用户增长与服务器资源的数学模型，将资源利用率从65%提升至85%，成本降低30%。

### 四、软件设计方法与用户需求的融合路径

#### (一) 传统设计方法的需求适配分析

方法	需求处理特点	优势	局限性
结构化设计	自顶向下分解，数据流驱动	适合流程明确的业务系统	需求变更适应性差（变更成本随进度呈指数增长）
面向对象设计	以用户业务对象建模，封装需求	支持需求迭代，代码复用性高	复杂系统的对象关系难以管理（可能出现需求遗漏）
基于模型的设计 (MBD)	通过UML模型可视化需求	提升需求沟通效率，减少歧义	模型维护成本高，依赖专业建模能力

实践表明，传统方法在处理模糊需求、动态变更时存在天然缺陷。某电子政务系统采用结构化设计，因政策调整导致需求变更，最终返工量占总开发量的45%，工期延长12个月。

#### (二) 敏捷开发方法的需求响应机制

敏捷开发通过“迭代增量”模式实现需求的动态管理，核心实践包括：

1. 用户故事拆分：将复杂需求分解为可在2-4周完成的最小单元（如“用户故事=角色+目标+场景”格式）。

2. 冲刺规划会议：团队与用户共同确定迭代周期内的需求优先级，使用燃尽图 (Burndown Chart) 跟踪进度。

3. 持续集成/持续部署 (CI/CD)：通过自动化测试确保需求变更的系统稳定性，某互联网公司实现每天500+次代码部署，需求响应速度提升70%。

某医疗SaaS平台运用敏捷方法，将需求评审周期从传统瀑布模型的45天缩短至7天，客户满意度从62%提升至89%。但需注意，过度简化需求文档可能导致知识流失，建议采用“轻量文档+可视化模型”的混合策略。

#### (三) 需求驱动的软件架构设计

架构设计的核心是建立需求到技术方案的映射关系，关键步骤如下：首先，通过需求分析矩阵，将功能性需求对应业务模块，非功能性需求关联技术组件，如高并发需求适配Redis与Nginx。其次，依据需求特征选择架构风格，微服务适合复杂多变业务，单体架构适用于功

能单一领域。再者，运用ATAM方法进行质量属性设计与决策。Netflix案例表明，微服务架构可应对用户规模增长实现快速响应，但也导致服务治理复杂，需借助服务网格技术应对挑战。

## 五、实证研究：典型案例对比分析

### （一）成功案例：微信的需求生态构建

微信从即时通讯工具发展为超级APP，关键在于建立三级需求响应体系：基础需求层满足核心通讯需求；扩展需求层通过开放平台转化用户多元需求；隐性需求层满足社交认同等需求。需求管理采用“灰度发布+用户反馈闭环”机制提升功能接受度。

### （二）失败案例：某跨境电商软件的需求陷阱

某跨境电商平台投入2000万元开发的ERP系统上线6个月后用户流失率达70%，根源在于需求分析失误，存在伪需求误判、场景缺失、技术债累积等问题。该案例警示，需求分析需构建三方评审机制，防止技术导向的需求失真。

## 六、用户需求的验证与持续优化

### （一）数据驱动的持续优化

建立“反馈收集—需求分析—迭代开发”的闭环：

1. 反馈渠道：客服工单（占比35%）、应用内反馈（40%）、社交媒体监测（25%），某社交APP通过NLP分析用户评论，自动识别出12类高频需求。

2. 需求优先级排序：采用WSJF（加权最短作业优先）算法，综合考虑需求价值、开发成本、风险系数，某电商平台将“直播带货”需求优先级提升，使其成为GMV增长的核心驱动力。

3. 版本管理：使用语义化版本号（SemVer）规范需求变更，如1.0.0代表重大需求升级1.1.0代表功能增强，

1.0.1代表缺陷修复。

### （二）新兴技术赋能需求验证

眼动仪测试显示，用户对软件界面的注意力80%集中在左上区域（F型浏览模式），某办公软件据此调整功能入口布局，使核心功能使用率提升40%。脑电仪（EEG）测试可捕获用户的认知负荷，帮助优化复杂操作流程，如某CAD软件通过脑电数据发现，三维建模功能的界面复杂度导致用户疲劳度上升30%，通过分步引导设计降低认知成本。

## 结论

随着数字化转型的深入，软件设计将从“功能实现”走向“价值创造”，基于用户需求的设计方法论需要持续吸收认知科学、行为经济学等跨学科理论，在技术快速迭代中保持对用户本质需求的洞察力。这既是学术研究的前沿领域，也是产业实践的核心竞争力所在。

## 参考文献

- [1] Cooper A, Reimann R, Cronin D. About Face 4: The Essentials of Interaction Design[M]. Wiley, 2014.
- [2] 梅宏, 王千祥, 黄罡. 软件工程的现状与未来[J]. 中国科学: 信息科学, 2020, 50(1): 1-26.
- [3] 李兵, 吴吉义. 微服务架构设计与实践[M]. 机械工业出版社, 2021.
- [4] 周志明. 深入理解Java虚拟机: JVM高级特性与最佳实践(第3版)[M]. 机械工业出版社, 2019.
- [5] Fowler M, Lewis M. Microservices: Lightweight Architecture for the Enterprise[J]. IEEE Software, 2015, 32(1): 19-25.