



# 电信模式下的敏捷开发交付——业务UI驱动式开发

张剑, 张娜, 张波, 杜若彬, 孙占岭, 罗佳佳, 项磊

(中国电信股份有限公司上海分公司云中台/数字集成部, 上海 201203)

**摘要:** 随着计算机技术的快速发展, 软件项目的开发迭代愈发频繁, 项目规模与数量也在不断增加, 这也对项目管理的科学性、高效性提出了更高的要求。自20世纪60年代软件生命周期这一概念出现至今, 诸如瀑布模型、螺旋模型、敏捷开发等拥有不同特点的软件生命周期理论被相继提出, 然而, 这些典型理论已无法很好地适应当下各企业不同的项目管理需求。提出了一种名为用户体验驱动开发(UDD)的敏捷开发新模式, 旨在实现适用于中国电信的现代化项目管理模式。UDD在“明厨亮灶”项目中的实践表明, 其拥有成本可控、研发周期较短、客户满意度高的理想表现, 十分契合中国电信的实际开发需要。

**关键词:** UDD; 敏捷开发; 软件生命周期; 瀑布模型; 可视化开发工具

**中图分类号:** TP393

**文献标志码:** A

**doi:** 10.11959/j.issn.1000-0801.2025080

## 0 引言

软件在生活中扮演越来越重要的角色, 人们愈发依赖计算机提供的各种便捷服务, 软件开发的需求量与规模也在不断上升。然而, 软件产品的复杂程度之高也直接反映了开发与测试流程的困难与不确定性, 这导致了很多时候软件会发生无法预料的行为, 甚至可能导致严重的问题和损害。出于这些原因, 人们将目光投向了软件开发过程中的管理策略, 希望创造一套能够对人员、组织、程序进行规范, 从而提高软件质量的管理模式。由此, 3个目标被首次提出: 开发结构化编程语言(如Pascal、C语言)、开发设计方法与设计原则(如自顶向下细化、功能分解)、定义软件生命周期<sup>[1]</sup>。

在以上3个目标中, 软件的生命周期在项目开发中所起到的作用尤为突出。软件生命周期定义了软件产品生命周期中的不同阶段, 通常包括需求分

析和规范、设计、开发、验证和确认、部署、操作、维护和停用。同时, 软件生命周期模型(过程模型)也定义了执行这些不同阶段必须遵循的原则和准则, 如瀑布模型建议仅在前一个阶段的可交付成果完成后才开始下一阶段; 而螺旋模型则将软件开发视为由风险分析驱动的多次系统迭代<sup>[1]</sup>。这些模型的诞生使得软件项目的风险得以降低, 开发效率得到提升, 有效确保了项目按预期且高质量地完成。从软件生命周期理论诞生至今, 为了适应项目管理的不同需求, 多种软件生命周期模型应运而生, 其中比较著名的有瀑布模型与敏捷开发。

### (1) 瀑布模型

瀑布模型如图1所示, 瀑布模型大致将开发阶段分为6步, 且各步骤间具有明确的依赖关系<sup>[2]</sup>。基于该模型的顺序依赖关系, 一旦在靠后的位置处发现先前步骤中的错误与质量问题, 就需要整体回到该步骤重新开始, 造成大量重复劳

动。为了尽可能减少该情况的出现，瀑布模型提出了“推迟实现”与“质量保证”两个观点，前者要求尽可能推迟程序的物理实现，先将程序的逻辑模型考虑清楚；后者要求在每个开发步骤中均有对应的规范文档产出，并组织质量保证小组对产出的文档进行质量评审，保证每个阶段的稳定输出、避免后续返工，也正是因为这一要求，瀑布模型可以说是一种由文档驱动的模式。不过文档驱动这一特点也导致了在最终产品出现之前，用户只能通过文档了解该产品的具体样式，无法对该产品有一个动态、全面的了解，一旦用户发现最终的产品同自身的想法产生冲突、无法满足自身需要，该产品就会面临推倒重来的局面。因此，瀑布模型由文档驱动的特点使得产品的维护成本得以降低、每阶段均有稳定的产出，但一旦出现早期步骤错误或未能满足用户需求的情况，就会面临规模空前的返工；同时，由于在每一步骤中均涉及验收环节，所以该模型的开发周期往往较长。

(2) 敏捷开发（以极限编程为例）

相比于瀑布模型，敏捷开发的交付周期十分短，且不会出现牵一发而动全身的颠覆性重构，其整体步骤如图2所示。敏捷开发的核心在于快速开发、不断迭代<sup>[3]</sup>，项目组首先会对可能出现的难点提出解决方案，再结合用户需求，根据项目实际与用户设定的优先级订立交付计划，随后

就进入多个迭代过程，通过多次快速交付新版本的方式更正老版本错误、快速让用户看到成品、满足用户提出的新要求。因此，该开发模型可以在灵活满足用户不断改变的需求的同时兼顾产品质量与交付速率，不过频繁的迭代也导致了其对人员的依赖性很强，一旦发生人员的更换，就需要进行大量的交接工作以便新人员快速融入产品的开发节奏中。用户需求的不断改变、产品的不断迭代同样会导致开发成本的提升，无论是在人力还是物力上，该开发模型均会产生较大的投入。

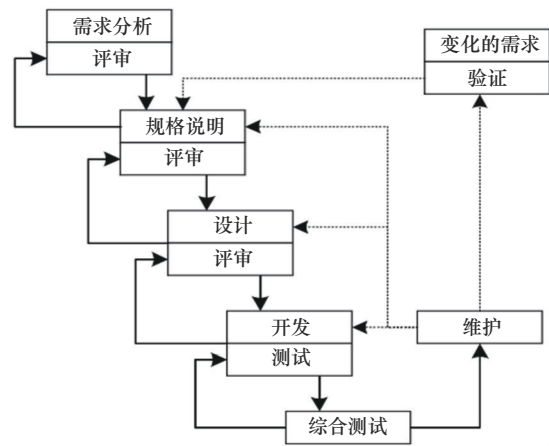


图1 瀑布模型

综上所述，瀑布模型与敏捷开发虽然各有明显的优点，但因其开发周期长、成本投入高的痛点，二者均无法很好地满足身为运营商的电信公

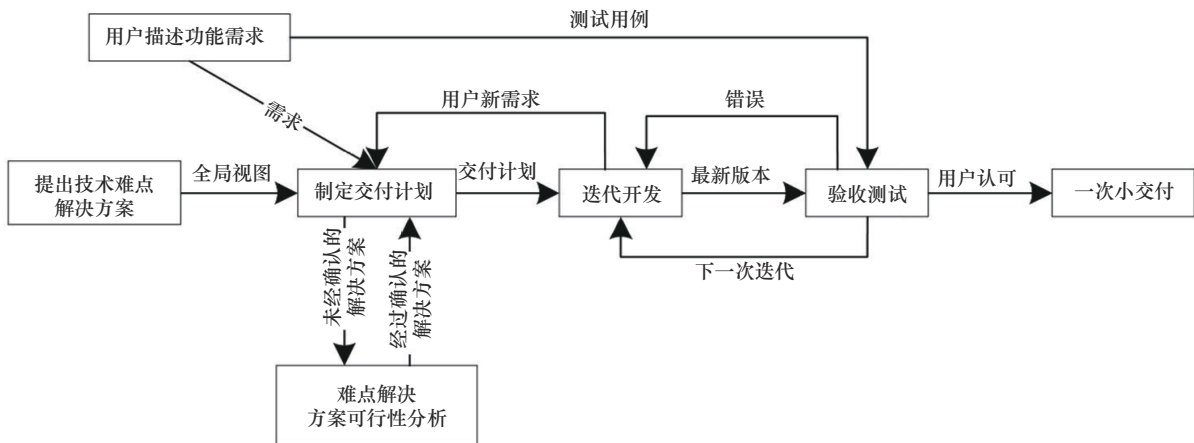


图2 敏捷开发(极限编程)



司的开发需要。而一些较新的开发模型，如混合开发模型，则只是简单地分阶段使用瀑布模型与敏捷开发进行项目管理，并未对上述问题提出具有建设性的解决方案<sup>[4]</sup>。为了适应日益复杂的企业项目管理需求，更好地进行电信日常开发的管理，针对以上缺陷，本文提出了名为用户体验驱动开发（UE-driven development, UDD）的敏捷开发新模式。UDD是以用户界面（UI）、用户体验（UE）为驱动，结合Figma、Axure、可视化开发平台等成熟工具，聚焦电信实际特点、低成本、短交付周期的新型开发模型，能够在保证客户满意度的同时有效降低企业非必要的研发投入。

## 1 UDD敏捷开发模型

### 1.1 模型整体设计

UDD敏捷开发模型（后称UDD）整体设计如图3所示，由5个主要步骤组成，首先根据客户不断变更的需求对产品设计进行迭代，直到用户满意后再开始制定开发计划、进行开发、完成测试验证并最终发布。不同于传统敏捷开发模式，UDD会对前期的概念模块进行迭代而非对计划、开发、验证这3个开发相关模块进行整体迭代，这也体现了UDD的核心思想——谋定而后动，即在开始正式开发前先同客户完成对UI、UE设计的确认，以大幅减少重复迭代开发所导致的研发资源的浪费。

在上述UDD敏捷开发模型开发过程中，需求确认阶段需要采用迭代方式，即在迭代过程中

时刻与用户保持沟通，并根据每次迭代过程中用户提出的新需求，迅速修改产品的设计细节，在一轮又一轮迭代的过程中不断完善产品的设计，使得最终得到的系统版本可以在最大限度上贴近用户的需求。

在需求确认阶段结束后，会进入UDD开发模式的确认阶段，该阶段主要为后续的开发工作做准备，确保开发团队针对该项目具有清晰的开发方向和开发计划，为后续系统开发工作的顺利开展打下基础。主要内容如下。

（1）细化需求。在需求确认阶段的基础上，进一步细化和明确各个功能模块的具体需求，并确定每个功能模块的具体实现细节和技术方案。

（2）确定迭代周期。根据项目规模和团队能力确定项目的开发周期，并根据周期将项目划分为多个开发部分，每个开发部分完成一部分功能的开发、测试和交付工作。

（3）制定详细计划：为每个开发周期制定详细的开发计划，如确定项目的功能列表以及功能开发优先级、确定预计开发工作量、明确各个开发阶段的目标和交付成果等。

（4）对团队成员进行开发任务分配、选择适合项目的开发与管理工具、及时进行开发进程跟踪和调整等。

计划阶段结束后，会进入UDD开发模式的开发阶段，开发阶段是将用户需求落地，并转化为实际产品的重要过程。UDD的开发过程主要包括：根据需求确认阶段得到的产品设计稿，在开发过程中进一步细化产品设计，根据得到的设计

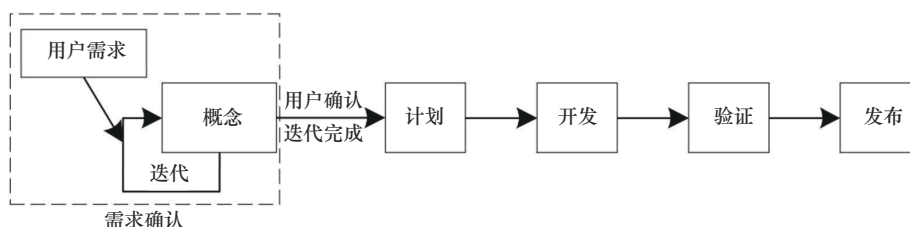


图3 UDD敏捷开发模型整体设计

稿和用户功能需求进行代码编写，并确定代码风格良好，易于扩展和维护、使用持续集成工具自动化测试和集成流程，并实现持续交付，使得每次提交的代码均可快速部署到生产和测试环境中，对代码进行严格评审，确保代码质量等。

在开发阶段结束后，项目流程会进入 UDD 开发模式中的验证阶段。验证阶段的意义在于确保开发出的软件系统符合用户需求、功能完整并且质量可靠。验证阶段的具体内容主要如下。

(1) 功能测试。验证每个功能是否按照需求文档的要求正确实现并达到预期的效果。

(2) 集成测试。测试各个模块或组件之间的接口是否可以正常工作以及各个部分协同工作时，确保整体系统可以正常运作。

(3) 系统测试。模拟实际运行环境，验证软件系统整体能否正常工作，性能参数是否正常。

(4) 回归测试。加入新功能或新组件后，重新测试旧功能确保其运行正常。

(5) 进行缺陷管理和修复。记录测试过程中发现的问题，并对其进行分类和优先级排序；同时跟踪缺陷修复的进度，并重新测试已修复的缺陷等。

验证阶段结束后开发流程进入 UDD 开发模式的最后一个阶段——发布阶段。在 UDD 开发模式中，发布阶段是产品生命周期的关键环节，标志着软件系统从开发和验证阶段过渡至面向用户/市场的状态。发布阶段的内容主要如下。

(1) 构建与打包。将开发完成并通过验证的代码进行编译和打包，形成可发布的版本。

(2) 部署规划。制定详细的部署计划，包括部署的时间表、步骤和回滚策略等。

(3) 部署实施。将软件系统部署至生产环境，并执行部署计划，确保发布过程顺利进行。

UDD 开发模式通过将客户体验置于开发过程的核心地位，确保软件产品不仅能满足技术要求，更能贴合客户的实际需求。UDD 开发模式强

调迭代式的开发、持续的测试与验证以及频繁的客户参与，有助于打造更加符合市场需求、更加令客户满意的软件产品。此外，UDD 模型还促进了开发团队之间的协作，提高了项目的透明度和响应能力，使得软件开发更加高效、灵活，相比起其他开发模式，无疑是更加适合运营商公司的开发方法。

## 1.2 UDD 统筹下的研发流程

### 1.2.1 各流程具体任务

UDD 研发流程如图 4 所示。在项目开始之初，设计人员首先会根据产品经理的要求完成产品的 UI 设计；随后，产品经理会根据已完成的 UI 设计图同客户进行需求确认，若客户认为仍须修改，则设计人员继续根据其需求进行 UI 界面的迭代；客户完成确认后，UI 板块即完成定稿，需求经理将根据 UI 设计图创建动态 UE 图，对开发目标进行细化分解；为了让客户更加直观地了解最终产品的交互方式与实际样式，前端开发人员将根据 UI、UE 设计图，并借助可视化开发平台快速完成前端界面的实现，并使用该样例界面同客户进行最终确认，若客户认为仍须修改，则需求经理将继续根据其需求调整 UE 设计；若客户认可当前样例，则认定客户的需求及成品要求已经明确无误，此时会开始制定开发计划并开始数据库的设计、后端业务及前端数据交互模块的开发；完成开发后，专业测试团队将根据 UI、UE 编写测试用例，对产品进行全方位的正确性测试，以保证产品在业务上的最终质量。

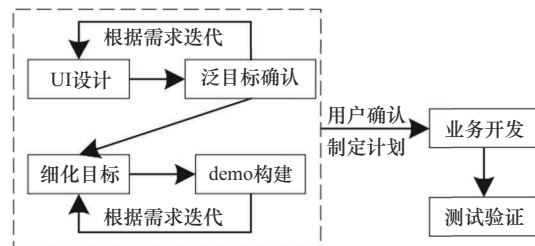


图 4 UDD 研发流程



在各步骤的实施过程中，UDD始终贯穿其中。UI与UE不仅在明确客户需求时起到了关键作用，也为前端开发人员提供了明确的开发规范，从而有力地保证了开发流程的可靠性，有效地降低了风险。

### 1.2.2 相比于其他模型的优势

首先，UDD将瀑布模型“推迟实现”的观点发挥到了极致。瀑布模型仅仅要求“尽可能地推迟实现”“将逻辑模型考虑清楚”，并没有具体规定应如何做到以上两点，而UDD则明确规定了将物理实现推迟至用户完成UI、UE的整体确认后再进行，使得“推迟实现”得以真正提高项目的完成质量、降低完成风险。同时，UDD也沿用了瀑布模型的质量评审机制，有效地保证了项目各阶段的文档产出质量，保证了代码的可维护性，这也使得人员的更换并不会对项目的推进造成太大的影响。

其次，UDD也吸纳了敏捷开发“快速沟通”的特点，通过同客户频繁、简短的沟通明确客户需求，从而做到及时调整产品设计以对客户需求进行响应，使产品更加贴合用户心理预期，有效地提升客户满意度。同时，UDD在沿用敏捷开发迭代思想的基础上对其进行了改进，不再对烦琐的开发模块进行迭代，而是着眼于开发前的UI、UE迭代，这使得研发人员的工作量与项目的整体成本得以下降、研发效率得以提高，且不会影响最终的客户满意度，起到降本增效的作用。

综上，UDD同时具备了瀑布模型与敏捷开发各自的优势，并弥补了其二者原有的不足之处，做到了对电信公司实际开发需求的适配：成本可控、研发周期较短、客户满意度高。

### 1.2.3 各阶段使用的工具

在基于UDD进行软件系统开发的过程中，不仅需要借助其先进的开发理论来规范开发流程，也需要借助外界工具进行辅助开发，提高开发效率。在第1.2.1节介绍的整个开发流程中，主

要使用的工具为Figma、Axure及可视化开发工具，三者分别负责UI界面设计、快速原型设计及提供前端可视化操作平台。

#### (1) Figma

Figma是一个基于浏览器的协作式UI设计工具，具有强大的协同能力，并且支持跨平台使用，如Windows、Chrome、Linux、Mac、TNT等系统。

与其他设计工具相比，Figma具有显著的使用优点：同时支持多人协同操作，可以让多个UI设计师在同一个屏幕上集思广益，并且支持对他人的设计风格和样式进行重复使用，从而实现页面设计风格统一，进而显著提升设计效率；具备强大的版本控制能力，可追踪、管理设计稿的变更历史，确保团队沟通顺畅；支持实时预览与反馈，设计过程中可随时查看并调整设计效果，并通过批注实时反馈设计建议。

在UDD中，Figma的使用几乎贯穿整个研发流程。在前期设计阶段，其被用于向客户展示UI设计界面，明确客户需求，而这也为后期的UE、前端界面开发提供了可依据的开发蓝本；在最后的测试阶段，Figma设计的UI界面也被当作检测开发正确性的重要依据，严格保证了成品样式贴合客户对UI界面的实际需求。

#### (2) Axure

Axure是一款专业的快速原型设计工具，用于快速创建应用软件或网站的线框图、流程图、交互原型和规格说明文档。该软件支持多人协作设计和版本控制管理，能够制作高保真或低保真的互动原型，模拟真实的用户体验。Axure的组件库功能使得设计组件可以重复使用，从而提高设计的一致性和效率，同时支持复杂的交互和动画设计。此外，Axure拥有强大的插件生态系统，能够满足多样化的设计需求，进一步扩展其功能。

在本文提出的软件开发模式中，Axure也扮

演着十分重要的角色。在UI设计确定后, UE会根据客户不断变更的需求进行多次迭代, 而Ax-ure则可快速构建出直观的UE设计图供客户进行检验, 为后续的前端交互逻辑与后端接口的开发设计提供重要参考。

### (3) 可视化开发工具

本文提出的软件开发模式还使用了一种基于数据大屏展示的方法。该展示方法主要包括: 在检测到用户在操作界面上输入的页面创建操作时, 新建页面; 在检测到用户对组件仓库中任一标准组件的拖动操作时, 将被拖动的标准组件显示在所新建的页面上; 在检测到用户在所述操作界面上输入的保存操作时, 根据所新建的页面上的标准组件的源码, 生成数据大屏文件和相应的数据大屏页面的访问地址; 在检测到对所述访问地址的访问请求时, 读取所述数据大屏文件, 并渲染出相应的数据大屏页面。

由于第1.2.1节提到的样例构建步骤需要前端人员完成对UE设计图的具体实现, 所以样例的开发效率极大地影响着整个迭代过程的推进速度, 而前文提到的可视化开发平台能够极大地提升开发人员的开发效率, 使得样例构建不会成为整个研发流程的效率瓶颈。

## 1.3 可视化开发界面

随着浏览器的快速迭代以及前端技术的发展, 公司的数据报表看板、城市的数据治理等都需要在web端展示数据大屏, 用户对数据大屏页面的人机交互及美观形成基本的共识。用户对需要展示的数据有自己的诉求和想法, 但项目的迭代升级周期长, 无法及时地解决用户的展示诉求。

目前在数据大屏展示领域, 传统的做法是定制化开发。即根据用户需求, 设计、开发、测试、发布, 此做法需要开发人员熟悉前端技术及图表库、地图库等使用, 前期开发周期长, 对开发人员专业水平要求较高, 且后期需求变更及维

护成本较高。

鉴于以上提出的种种问题, 本文的可视化开发界面部分提供了一种数据大屏页面展示方法, 通过借助可视化装置、设备和介质, 来克服或者部分解决上述问题。

在使用可视化界面进行开发的过程中, 借助数据大屏页面进行方法展示的主要步骤可以划分如下。

(1) 在检测到用户在操作界面上输入的页面创建操作时, 新建页面。

(2) 在检测到用户对组件仓库中任一标准组件的拖动操作时, 将被拖动的标准组件显示在所新建的页面上, 所述组件仓库中的各个标准组件是采用统一规范封装而成的。

(3) 在检测到用户在所述操作界面上输入的保存操作时, 根据所新建的页面上的标准组件的源码, 生成数据大屏文件和相应的数据大屏页面的访问地址。

(4) 在检测到对所述访问地址的访问请求时, 读取所述数据大屏文件, 并渲染出相应的数据大屏页面。

在利用可视化大屏进行实际开发的过程中, 要求在基于可视化设计的设计稿基础上, 利用可视化工具, 进行大屏页面的构建和开发。在基于可视化大屏进行可视化开发的过程中, 严格的开发标准有助于项目的顺利交付。主要的开发标准包如下。

(1) 需求理解。要求准确理解可视化项目的需求和目标, 并可清晰列出大屏页面需要展示的内容和信息点。

(2) 工具应用。要求可以充分利用工具的特点和优势, 进行优化大屏页面的构建。

(3) 布局设计。若要设计出合理、美观的大屏页面布局, 必须要求组件摆放有序, 符合用户的视觉习惯和阅读逻辑。

(4) 数据可视化。在基于大屏展示的可视化



构建方法中，若要准确实现设计稿中的数据可视化组件，必须要求数据展示要直观易懂，且能清晰传达信息。

(5) 交互设计。需要设计简洁明了的交互功能，方便用户操作。

(6) 数据展示。应当确保大屏数据展示合理。

(7) 文档编写。应当编写条理清晰的作业说明文档，并详细说明设计思路、实现过程及遇到的问题等。

针对软件系统开发过程中的具体开发需求，可视化界面还提供了一系列功能模块，最大限度地简化了开发过程、提高了开发效率。

## 2 应用案例

目前在“明厨亮灶”这一旨在提升食品安全透明度的创新项目中，用户体验驱动的开发模式——UDD展现出了其无与伦比的优势与巨大价值。通过将UDD模式与“明厨亮灶”系统开发相结合，明确了UDD在项目开发中的具体实施细节，证明了UDD模式可以极大地优化开发流程，显著提升项目的效率与质量。

在使用UDD模式进行“明厨亮灶”项目的具体开发过程中，设计人员首先给出UI、UE界面的设计初稿，用户针对设计初稿提出修改意见，随后设计人员根据用户修改意见，完善UI、UE界面。不断重复上述过程，直到UI、UE界面的设计情况达到用户期望，在该案例中，UI、UE界面共迭代三轮。具体迭代情况如下。

在第一轮沟通当中，根据开始的设计原图第一稿确定客户的原始诉求，项目中客户诉求内容主要包括覆盖松江全区企业情况，按照优良差展示最终效果，显示食安指数排名、厨房红黑榜，显示相关告警及风险预警。

在第二轮沟通当中，通过“以图换图”的沟通方式促进了信息的准确传递与理解，确保双方对项目的理解保持高度一致，从而加速决策与执

行的进程。

在第三轮沟通当中，引入多样化的轮播样式与动态展示方式，从而有效避免因页面空间限制而可能导致的信息遗漏或压缩。

在上述基于UDD开发模式的系统开发过程中，通过多轮沟通细化需求至样例目标，紧密对齐客户，防重构，版本实现了高达80%的浪费减少。在多轮的沟通过程中，用户也意识到了初版目标存在一些不合理的地方，提出了相应的修改意见。相比于第一轮中的UI、UE界面，最终的结果去除了许可证号、检查日期等用户不需要的冗余信息，优化了页面布局。比起未使用UDD模式进行开发的情况，极大提升了项目开发效率，最大限度地避免了实际开发情况与用户需求相冲突导致的多轮迭代开发，大大降低了项目彻底重构的风险，同时增强了客户满意度与信任度。

根据上述迭代过程，结合大屏可视化开发工具，极大地提高了开发过程的效率，简化了“明厨亮灶”系统的开发过程。由于“明厨亮灶”系统中包含的功能数量较多，涉及的交互逻辑较为复杂，使用到的开发类技术知识较为广泛，因此若不能针对开发模式进行简化，则很容易导致开发效率低下、人力成本高、软件系统存在缺陷等问题。在最初使用瀑布模式进行开发时，瀑布模式灵活性较低，导致需求变更困难，一旦需求变更，就需要回到前期阶段重新设计甚至重做，极大增加了开发成本和时间消耗，因此时间人力成本高达280人·天；在使用敏捷开发模式进行开发时，由于敏捷开发模式具有较高灵活性，可以快速响应变化，因此在时间人力成本方面有显著下降，但由于敏捷开发模式对沟通、人力成本提出了较高要求，因此整体时间人力成本仍然较大；在使用UDD开发模式进行开发时，由于UDD开发模式完美结合了瀑布开发模式和敏捷开发模式的优点，因此可以灵活面对需求变更，并

显著降低开发过程中的人力和沟通成本，将时间人力成本降低至60人·天，实现了效率显著提高的同时，也极大降低了开发成本。

经对比可知，终稿界面设计较为清晰简洁，用户的需求以及修改意见得到了最大限度满足。

### 3 结束语

UDD开发模式在本案例中的实践充分凸显了其用户至上的核心优势。通过不断迭代循环过程中的多轮沟通，UDD极大地增强了需求把握的精准度和效率，有效跨越了传统开发中常见的需求误解障碍。这种紧贴用户真实需求、不断迭代优化的开发策略，不仅成功将项目周期从冗长的280人·天压缩至高效的60人·天，实现了资源的最优化配置，还通过创新的动态界面与多样化展示手段，极大地提升了用户体验，使软件功能更加贴近用户需求。

与瀑布模型的开发周期长和敏捷开发的开发成本投入高相比，UDD模式以其卓越的灵活性和对用户需求的敏锐捕捉能力脱颖而出。它融合了敏捷开发的快速迭代精神与瀑布模型中推迟实现的观点，走出了一条低成本、高质量、短周期的软件开发之路。

然而，UDD模式在实际应用过程中还具有一定的提升空间。它要求项目团队具备良好的与客户进行沟通协同的能力，确保及时掌握用户需求并给予反馈。因此，面对复杂多变的用户需求，

为了进一步提升UDD模式的效能，未来可以考虑引入先进的AI辅助工具，以自动化手段提升需求分析的效率和准确性。

### 参考文献：

- [1] FUGGETTA A. Software process: a roadmap[C]//Proceedings of the Conference on The Future of Software Engineering. New York: ACM Press, 2000: 25-34.
- [2] 闫帅, 许鹏翔. 基于瀑布模型与敏捷开发相结合的项目管理方法探讨[J]. 电子技术与软件工程, 2013(18): 67.
- [3] 贺重勉. 基于Scrum模型的软件项目敏捷开发研究[D]. 北京: 北京邮电大学, 2022.
- [4] 蒋鹏, 刘钧钧, 秦炜. 可信软件混合开发模型[J]. 信息技术与标准化, 2024(S1): 70-74.

### [作者简介]

张剑 (1980-), 男, 中国电信股份有限公司上海分公司云中台/数字集成部工程师, 主要研究方向为AI创新。

张娜 (1992-), 女, 中国电信股份有限公司上海分公司云中台/数字集成部研发经理, 主要研究方向为软件开发。

张波 (1989-), 男, 中国电信股份有限公司上海分公司云中台/数字集成部需求研发经理, 主要研究方向为软件开发。

杜若彬 (2001-), 男, 现就职于中国电信股份有限公司上海分公司云中台/数字集成部, 主要研究方向为软件开发。

孙占岭 (1999-), 男, 现就职于中国电信股份有限公司上海分公司云中台/数字集成部, 主要研究方向为软件开发。

罗佳佳 (1991-), 男, 现就职于中国电信股份有限公司上海分公司云中台/数字集成部, 主要研究方向为软件开发系统架构设计。

项磊 (1992-), 男, 中国电信股份有限公司上海分公司云中台/数字集成部研发经理, 主要研究方向为软件开发。