

第3章

软件需求获取

在软件开发的初期阶段,获取准确且全面的软件需求至关重要。需求的错误或遗漏可能会导致项目失败或资源浪费。因此,软件需求获取作为软件工程中的核心步骤,直接影响项目的成败。本章将深入探讨软件需求获取的定义、目标以及不同的需求获取方法。将介绍如何通过访谈、问卷调查、观察法、工作坊等多种手段,从不同利益相关者那里收集准确的需求信息。通过学习本章内容,读者将掌握需求获取的核心步骤和方法,为软件项目的成功奠定基础。

本章目标

- 了解软件需求获取的定义和目标。
- 掌握软件需求获取过程的步骤。
- 学习不同的软件需求获取方法,包括访谈、问卷调查、观察法、工作坊等。
- 掌握软件需求获取过程中的技巧和策略。
- 了解软件需求获取的工具和技术。



观看视频:
需求获取
补充知识

3.1 软件需求获取过程概述

在深入探讨具体的方法之前,了解软件需求获取的整体过程是非常重要的。需求获取过程通常包含多个步骤,每一步都在确保需求完整性和准确性方面发挥着关键作用。接下来将介绍软件需求获取的定义与目标,并详细说明这一过程的各个步骤。

3.1.1 软件需求获取的定义与目标

1. 定义

软件需求获取是软件工程中一项关键的初始活动,其主要目的是从客户、用户和其他利益相关者处收集和理解他们对软件系统的需求。这一过程不仅涉及询问和记录用户的软件需求,还包括帮助用户明确和澄清他们的软件需求,以确保开发团队能够准确理解并满足这些软件需求。

软件需求获取涉及从抽象的用户需求到具体的软件需求的转化过程。这一过程需要多种方法和技术,以确保收集到的信息是全面、准确且可行的。

2. 目标

软件需求获取的目标是确保最终的软件产品能够满足所有利益相关者的期望和需求,以下是软件需求获取的主要目标。

- 识别利益相关者: 确定所有相关的利益相关者,包括客户、最终用户、开发团队、支

持人员等,以确保他们的需求和期望能够被充分考虑和理解。

- 理解用户需求: 深入理解用户的实际需求和业务目标,而不仅仅是用户提出的功能性需求。这需要了解用户的工作环境、业务流程和面临的挑战。
- 明确系统范围: 确定系统的边界和范围,避免范围蔓延问题。明确系统的边界有助于设定合理的开发目标和管理用户期望。
- 收集全面软件需求: 收集和记录所有类型的软件需求,包括功能性需求、非功能性需求、业务需求、性能需求、安全需求等,确保软件需求的全面性和准确性。
- 达成共识: 在利益相关者之间达成对软件需求的共识,解决任何冲突和分歧,确保所有人对软件需求的理解一致,并同意这些软件需求指导系统的开发。
- 验证软件需求的可行性: 评估软件需求的技术可行性、经济可行性和时间可行性,确保软件需求可以在现有的技术和资源条件下实现,并在预定的时间和预算内完成。
- 建立优先级: 为软件需求建立优先级,确定哪些软件需求是必需的,哪些软件需求是可选的或可以在后续版本中实现。这有助于在资源有限的情况下,确保最重要的软件需求得到优先实现。
- 记录软件需求: 将收集到的软件需求清晰、准确地记录下来,形成软件需求规格文档。这些文档将作为后续系统设计、开发和测试的基础。

通过达成这些目标,软件需求的获取过程有助于确保软件产品能够满足用户的实际软件需求,同时遵守预算和时间限制,减少开发过程中的风险和不确定性。

在实际应用中,软件需求获取是一个动态和迭代的过程,它涉及多种技术和工具,以及与利益相关者的持续沟通。随着项目的进展,新的软件需求可能会被发现,而原有的软件需求也可能改变,因此软件需求获取并不是一次性的活动,而是贯穿整个项目生命周期的持续过程。

接下来将详细介绍软件需求获取的各种方法、过程、技巧与策略,以及支持这一过程的工具和技术,旨在为软件工程师、项目经理和所有参与软件开发的人员提供一个全面的软件需求获取框架。

3.1.2 软件需求获取的步骤

软件需求获取是一项系统性的工作,需要经过多个步骤才能确保软件需求的完整性和准确性。以下是软件需求获取的步骤(见图 3-1),每一步都有其特定的任务和目标。

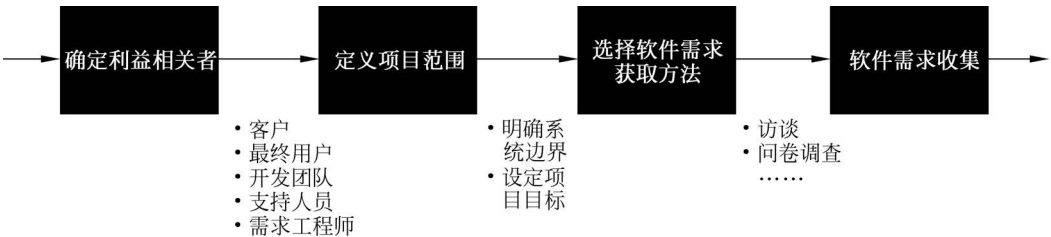


图 3-1 软件需求获取的步骤

1. 确定利益相关者

首先,需要识别和确定所有的利益相关者。利益相关者包括那些对系统有软件需求或会受到系统影响的个人和群体。常见的利益相关者如下。

- 客户：支付费用的组织或个人。
- 最终用户：实际使用系统的人。
- 开发团队：包括项目经理、开发人员、测试人员等。
- 支持人员：提供技术支持和维护的人。
- 需求工程师：负责软件需求分析和管理的专业人员。

2. 定义项目范围

在确定利益相关者之后,需要明确项目的范围。这一步的目的是防止软件需求蔓延,确保项目在可控范围内进行。

- 明确系统边界:确定系统的功能性和非功能性需求,明确哪些功能在项目范围内,哪些功能不在项目范围内。
- 设定项目目标:与利益相关者一起制定明确的项目目标,确保所有参与者对项目预期有一致的理解。

3. 选择软件需求获取方法

选择适合的软件需求获取方法是确保软件需求准确的重要步骤。常用的软件需求获取方法包括以下几种。

- 访谈:与利益相关者进行一对一或小组访谈,深入了解他们的需求和期望。
- 问卷调查:通过书面问卷收集大量用户的需求信息。
- 观察法:观察用户的工作流程和实际操作,理解他们的需求。
- 工作坊:组织利益相关者共同讨论和定义软件需求。
- 原型开发与迭代验证:通过原型展示系统功能,收集用户反馈,逐步完善软件需求。
- 焦点小组:小规模用户群体讨论,收集具体软件需求和意见。
- 调查分析与文档研究:分析现有系统文档、用户手册、业务流程图等,获取软件需求信息。
- 用户故事和敏捷方法:使用用户故事和敏捷方法,灵活应对软件需求变化。
- 数据分析法:通过对现有数据进行收集、整理、分析和解释,来获取软件需求的方法。

4. 软件需求收集

选择合适的方法后,开始实际的软件需求收集工作。

- 安排和执行会议:组织访谈、工作坊或焦点小组会议,收集软件需求。
- 分发和收集问卷:设计和分发问卷,收集用户反馈。
- 进行现场观察:观察用户在实际工作中的操作,记录观察结果。
- 开发原型:创建系统原型,与用户一起评审和迭代。

3.1.3 软件需求获取过程中的关键角色及其责任

软件需求获取是一个复杂且协作性很强的过程,涉及多个关键角色,每个角色在软件需求获取过程中承担不同的责任。明确这些角色及其责任对于确保软件需求获取的质量和效率至关重要。以下是软件需求获取过程中的关键角色及其责任。

1. 客户

客户是软件需求的主要来源,客户通常是支付费用的组织或个人,期望通过软件系统解

决特定的业务问题或实现某些目标。客户的主要责任包括以下几点。

- 提供需求：描述客户业务需求和期望。
- 参与软件需求评审：确认和评审软件需求,确保开发团队理解准确。
- 决策支持：在软件需求变更或优先级调整时提供决策支持。
- 反馈与验证：提供反馈和验证软件需求的实现情况。

2. 最终用户

最终用户是实际使用软件系统的人,最终用户对系统的功能和使用体验有最直接的软件需求和期望。最终用户的主要责任包括以下几点。

- 提供使用场景：描述日常工作中的实际操作和业务流程。
- 反馈软件需求：提供详细的功能性需求和非功能性需求。
- 参与原型评审：评审系统原型并提供反馈。
- 软件需求验证：确认软件需求是否被准确实现,系统是否符合预期。

3. 需求工程师

需求工程师是软件需求获取过程中非常重要的角色,需求工程师负责在客户、最终用户和开发团队之间架起沟通的桥梁。需求工程师的主要责任包括以下几点。

- 软件需求收集：使用访谈、问卷、观察等方法收集需求。
- 软件需求分析：分析和整理收集到的软件需求,确保软件需求的完整性和一致性。
- 软件需求文档化：编写详细的软件需求规格文档,包括软件需求规格说明书、用户故事等。
- 软件需求验证：与客户和最终用户一起验证和确认软件需求。
- 软件需求管理：持续跟踪和管理软件需求变更,保持软件需求的可追溯性。

4. 项目经理

项目经理负责软件需求获取过程的整体协调和管理,确保软件需求获取过程按计划进行并满足项目目标。项目经理的主要责任包括以下几点。

- 规划和调度：制订软件需求获取的计划和时间表。
- 资源分配：分配和管理软件需求获取所需的资源。
- 进度跟踪：跟踪软件需求获取的进度,确保按时完成。
- 冲突管理：解决软件需求获取过程中出现的冲突和问题。
- 沟通协调：在客户、最终用户和开发团队之间进行沟通和协调。

5. 开发团队

开发团队包括软件工程师、设计师、测试人员等,他们负责将软件需求转化为实际的软件系统。开发团队的主要责任包括以下几点。

- 技术可行性评估：评估软件需求的技术可行性,提供专业建议。
- 原型开发：开发系统原型,帮助客户和最终用户理解软件需求。
- 软件需求实现：根据软件需求规格文档进行系统设计和开发。
- 软件需求验证：通过测试验证软件需求的实现情况,确保系统符合软件需求。

6. 测试人员

测试人员负责验证软件需求的实现情况,确保系统功能和性能符合预期。测试人员的主要责任包括以下几点。

- 软件需求分析：分析软件需求文档，编写测试用例。
- 测试执行：执行功能测试、性能测试等，验证系统的各项软件需求。
- 缺陷报告：记录和报告测试过程中发现的缺陷和问题。
- 回归测试：在软件需求变更后进行回归测试，确保系统的稳定性。

7. 产品经理

产品经理负责产品的整体规划和策略，确保软件需求与产品的长期目标和市场需求一致。产品经理的主要责任包括以下几点。

- 产品愿景：定义和维护产品愿景，确保软件需求与产品战略一致。
- 市场分析：进行市场分析和用户调研，识别用户需求和市场机会。
- 优先级设定：根据业务目标和市场需求为软件需求设定优先级。
- 软件需求沟通：与开发团队、客户和最终用户沟通软件需求，确保各方理解一致。

8. 利益相关者

利益相关者包括所有对项目有影响或受项目影响的个人或组织。利益相关者的主要责任包括以下几点。

- 软件需求提供：提供软件需求和期望。
- 参与评审：参与软件需求评审和确认过程。
- 提供反馈：在软件需求获取和实现过程中提供持续的反馈和建议。

通过明确以上角色及其责任，软件需求获取过程可以更加有序和高效地进行。每个角色在软件需求获取中扮演的关键作用和承担的责任都至关重要，只有各方密切合作，才能确保软件需求的完整性、准确性和可行性，从而为后续的系统设计、开发和测试奠定坚实的基础。

3.2 软件需求获取的方法

在明确了软件需求获取的过程之后，接下来将探讨各种具体的方法。这些方法各有特点，适用于不同的场景和需求类型。通过了解这些方法，读者可以选择最合适的方式来收集和分析需求信息，从而提高项目的成功率。

3.2.1 访谈

1. 访谈的定义

访谈是一种软件需求获取的方法，通过面对面的对话从利益相关者那里收集软件需求信息。访谈可以是正式的或非正式的、结构化的或非结构化的。它是一种直接、互动的交流方式，能够深入了解用户的软件需求、期望、业务流程以及面临的挑战。

根据访谈的结构和目的，访谈可以分为以下几种类型（见图 3-2）。

- 结构化访谈：使用预先设计好的问题列表，访谈内容和顺序严格按照计划进行。结构化访谈能够确保所有受访者回答相同的问题，便于比较和分析。
- 半结构化访谈：结合结构化和非结构化访谈的特点，有一些预设问题，但访谈过程中允许根据具体情况自由扩展和深入探讨。半结构化访谈既有一定的控制性，又具有灵活性，适用于软件需求获取中的大多数情况。

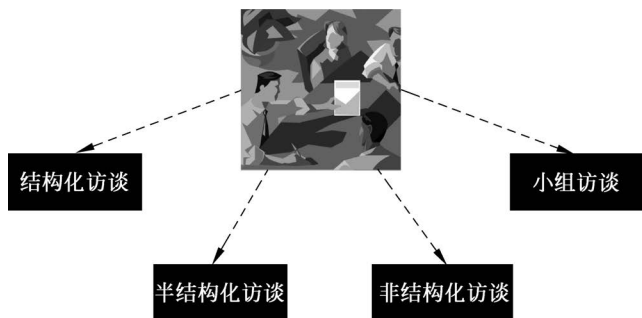


图 3-2 访谈的类型

- 非结构化访谈：没有固定的问题列表，访谈内容完全依靠访谈者和受访者的自由对话。这种访谈方式适合在初期探索软件需求、了解问题背景或挖掘潜在软件需求时使用。
- 小组访谈：对一个小组的利益相关者进行集体访谈，通过互动讨论收集软件需求。这种方式可以获取不同视角的软件需求，同时促进利益相关者之间的沟通和共识。

2. 访谈的步骤

进行有效的访谈需要经过精心的准备和执行，图 3-3 所示是访谈的步骤。



图 3-3 访谈的步骤

1) 确定访谈目标

在开始访谈之前，明确访谈的具体目标和期望结果。这包括：

- 明确要收集哪些类型的软件需求信息(如功能性需求、非功能性需求、业务流程等)。
- 确定访谈的焦点和范围，避免偏离主题。

2) 选择访谈对象

选择合适的受访者非常关键，受访者应当是对项目有深入了解和实际需求的人。通常包括以下几类对象。

- 最终用户。
- 系统管理员。
- 业务负责人。
- 技术支持人员。

3) 设计访谈问题

设计清晰、有针对性的问题列表，确保问题覆盖所有重要方面。通常包括以下问题。

- 开放性问题，如“请描述一下您一天的工作流程?”
- 封闭性问题：如“您是否需要系统提供数据导出功能?”
- 澄清性问题：如“能否详细说明一下您提到的‘数据处理速度慢’具体指什么情况?”

4) 安排访谈

与受访者约定访谈时间和地点，确保环境安静、不受干扰。同时，提前告知访谈的目的和流程，取得受访者的同意和支持。

5) 进行访谈

实际访谈中,访谈者需要注意以下几点。

- 建立信任: 开场时简要介绍自己和访谈目的,消除受访者的紧张感。
- 主动倾听: 认真倾听受访者的回答,不打断、不急于下结论,给予充分表达的机会。
- 灵活调整: 根据受访者的回答,适时调整和补充问题,深入挖掘有价值的信息。
- 记录与总结: 详细记录受访者的回答,访谈结束前对关键需求进行总结和确认。

6) 分析访谈结果

访谈结束后,对收集到的信息进行整理和分析,包括以下几点。

- 整理记录: 将访谈记录整理成书面文档,确保信息的完整性和准确性。
- 识别软件需求: 从访谈记录中识别出具体的软件需求,并进行分类和优先级排序。
- 编写访谈报告: 总结访谈结果,形成访谈报告,分享给利益相关者进行确认和反馈。

3. 访谈的技巧

为了确保访谈的有效性,访谈者需要掌握一些关键技巧。

- 准备充分: 在访谈前对项目背景、业务流程和受访者情况进行充分了解,准备好相关材料和问题。
- 保持中立: 访谈过程中保持客观、中立,不带个人偏见,不诱导受访者回答。
- 善于倾听: 注重倾听而不是只关注提问,捕捉受访者的真实软件需求和潜在问题。
- 灵活应变: 根据受访者的反应灵活调整访谈策略,避免过于死板或机械化。
- 控制节奏: 掌控访谈的节奏和时间,确保在规定时间内完成预定的问题和目标。
- 鼓励表达: 鼓励受访者自由表达,避免受访者因紧张或顾虑而不敢说出真实想法。

4. 访谈的优缺点

访谈作为软件需求获取的一种重要方法,具有其独特的优点和缺点。

1) 优点

- 互动性强: 可以通过互动深入了解软件需求,及时澄清和解答疑问。
- 灵活性高: 能够根据实际情况调整访谈内容和方向,灵活应对各种问题。
- 深度挖掘: 可以通过追问和探讨,挖掘出潜在软件需求和隐性软件需求。

2) 缺点

- 时间成本高: 需要花费大量时间进行准备、执行和分析,效率较低。
- 主观性强: 访谈结果可能受到受访者个人观点和情绪的影响,存在一定的主观性。
- 依赖技巧: 访谈效果很大程度上取决于访谈者的技巧和经验,新手容易出现遗漏或误导。

5. 访谈的实例

假设一个软件项目团队正在开发一个新的客户关系管理(CRM)系统,项目团队决定通过访谈来获取软件需求。他们进行了以下一系列访谈。

(1) 与销售经理进行访谈: 了解销售团队的业务流程、现有系统的不足以及对新系统的期望。

问题示例: 您目前使用的 CRM 系统有哪些主要不足? 新系统应如何改进?

(2) 与客服代表进行访谈: 收集客服代表在处理客户问题时遇到的痛点和软件需求。

问题示例: 您在处理客户投诉时,现有系统有哪些限制? 您希望新系统有哪些功能来

帮助您更有效地工作？

(3) 与 IT 支持人员进行访谈：了解技术支持和系统维护方面的软件需求和挑战。

问题示例：现有系统在维护和技术支持方面有哪些问题？新系统应如何设计以提高维护效率？

通过这些访谈，项目团队不仅收集到了详细的功能性需求，还发现了一些之前未曾考虑的非功能性需求，如系统性能和数据安全需求，从而为后续的系统设计和开发提供了全面和可靠的软件需求基础。

访谈作为软件需求获取的重要方法，通过面对面的交流能够深入理解用户需求，捕捉到其他方法难以获取的信息。在软件需求获取过程中，合理安排和运用访谈方法将大幅提高软件需求的准确性和完整性，为软件项目的成功打下坚实的基础。

3.2.2 问卷调查

1. 问卷调查的定义

问卷调查是一种通过设计和分发结构化问卷来系统地收集用户和利益相关者软件需求信息的方法。它特别适用于需要从大量用户中获取标准化、可量化数据的场景。问卷调查可以以纸质或电子形式分发，受访者可以在其方便的时间填写并提交。

2. 问卷调查的目标

问卷调查的主要目标如下。

- 广泛覆盖：收集来自大量分散用户的软件需求，确保各类用户群体的软件需求被充分了解。
- 标准化数据：通过预定义的问题获取标准化数据，便于分析和比较。
- 定量分析：提供可量化的数据，有助于后续的统计分析和软件需求优先级设定。
- 多样反馈：通过问卷能够收集多种类型的软件需求信息，包括功能性需求、非功能性需求和用户期望等。

3. 设计问卷的步骤

成功的问卷调查依赖于精心设计的问卷。图 3-4 所示是设计问卷的步骤。



图 3-4 设计问卷的步骤

(1) 明确调查目标：确定调查的具体目标和预期结果，确保问卷的问题与目标一致。

(2) 定义目标受众：明确问卷的目标受众，确保问题设计适合受众的背景和理解水平。

(3) 设计问题。

- 问题类型：包括开放式问题和封闭式问题。开放式问题允许受访者自由回答，适合收集详细意见；封闭式问题提供固定选项，便于量化分析。
- 问题顺序：问题的顺序应从简单到复杂，循序渐进地设计，以引导受访者顺畅地回答问题。
- 问题清晰：确保问题简明扼要，不含歧义，避免复杂的专业术语。

(4) 设计回答选项：对于封闭式问题，设计合理的回答选项，确保涵盖所有可能的回

答,避免受访者感到选项不够全面。

(5) 预测问卷: 在正式分发前,对问卷进行预测试,收集反馈并进行修改和优化。

4. 问卷的类型

根据问题的性质和调查目标,问卷可以分为以下几种类型。

- 开放式问卷: 受访者可以自由回答问题,适用于收集详细的意见和建议。
- 封闭式问卷: 提供固定的回答选项,适用于获取标准化和可量化的数据。
- 半开放式问卷: 结合开放式和封闭式问题,既能收集详细意见,又能获取量化数据。

5. 实施问卷调查

设计好问卷后,接下来的步骤是实施问卷调查。

- 选择分发方式: 确定问卷的分发方式,可以是纸质问卷、电子邮件、在线调查工具等。
- 分发问卷: 将问卷分发给目标受众,确保覆盖所有利益相关者。
- 收集问卷: 设定合理的截止日期,收集受访者填写好的问卷。
- 跟进和提醒: 在问卷分发后,适时跟进和提醒受访者填写和提交问卷,确保获得足够的反馈。

6. 数据分析与处理

问卷收集完成后,需要对数据进行分析 and 处理。

- 数据整理: 将收集到的问卷进行整理和录入,确保数据的完整性和准确性。
- 定量分析: 对封闭式问题的数据进行统计分析,使用频率分析、平均值分析、交叉分析等方法。
- 定性分析: 对开放式问题的回答进行内容分析,提取关键主题和观点。
- 结果总结: 总结分析结果,形成数据报告,提炼出主要软件和共性问题。

7. 问卷调查的优缺点

(1) 优点。

- 覆盖面广: 可以覆盖大量分散的受众,收集广泛的软件需求信息。
- 成本低: 相比其他方法,问卷调查的成本相对较低,尤其是在线问卷。
- 标准化: 通过统一的问题设计,可以获得标准化的数据,便于分析和比较。
- 匿名性: 受访者可以匿名回答,有助于获取真实意见。

(2) 缺点。

- 深度有限: 难以深入了解复杂软件需求,回答通常较为简短和表面。
- 响应率低: 如果问卷设计不合理或分发方式不合适,响应率可能较低。
- 误解风险: 问题设计不当可能导致受访者误解,影响数据质量。
- 缺乏互动: 缺乏与受访者的互动,无法及时澄清或深入挖掘软件需求。

8. 提高问卷调查效果的策略

为了提高问卷调查的效果,可以采取以下策略。

- 设计简洁: 问卷应尽量简洁,控制在合理的长度内,避免过多的问题。
- 提供激励: 可以提供一些小礼品或抽奖机会,激励受访者参与填写问卷。
- 清晰说明: 在问卷开头提供清晰的说明,解释调查的目的和问卷填写的重要性。
- 预测试: 通过预测试收集反馈,优化问卷设计,确保问题清晰和易于理解。

- 及时跟进：在问卷分发后,及时跟进和提醒受访者,鼓励他们尽早填写和提交问卷。

9. 问卷调查的实例

为了更好地理解问卷调查在需求获取中的应用,以下是一个关于医院信息管理系统开发项目的需求获取实例。

1) 项目背景

一家大型医院计划开发一个信息管理系统,以提高医院的管理效率和医疗服务质量。项目团队需要通过问卷调查收集不同部门和用户的需求,以确保系统功能全面并满足实际需求。

2) 需求获取过程

(1) 设计问卷。

- 问卷内容：根据不同用户群体(如医生、护士、管理人员、技术支持人员等)的需求,设计不同的问卷内容,涵盖系统功能性需求、用户体验、数据安全等方面。
- 示例问题。
 - 医生：您希望系统有哪些功能可以帮助您更高效地管理患者信息？
 - 护士：在日常工作中,现有系统有哪些不足？新系统应如何改进？
 - 管理人员：您对系统的报表功能有何需求？希望系统能提供哪些数据分析功能？
 - 技术支持人员：系统在维护和技术支持方面应具备哪些特性？

(2) 分发问卷。

- 问卷形式：采用电子问卷和纸质问卷相结合的方式,确保覆盖所有用户群体。
- 分发方式：通过电子邮件、内部通信平台和现场分发等方式,将问卷分发给相关人员。

(3) 收集和分析问卷。

- 数据收集：回收问卷后,整理和录入数据,确保数据的完整性和准确性。
- 数据分析：对收集到的数据进行分类和统计分析,识别出主要的功能性需求和非功能性需求。

(4) 需求确认和反馈调整。

- 需求确认：根据分析结果,整理出初步的需求清单,与利益相关者进行确认和讨论,确保需求的准确性和可行性。
- 反馈调整：根据利益相关者的反馈,对需求清单进行调整和完善,最终确定系统的详细需求。

问卷调查是一种高效、经济的软件需求获取方法,适用于从大量用户中收集标准化和可量化的数据。通过精心设计和合理实施,问卷调查可以为软件开发提供宝贵的软件需求信息,确保系统能够满足广泛用户的需求和期望。在实际应用中,结合其他软件需求获取方法,可以更全面地理解和满足用户需求,提高软件项目的成功率。

3.2.3 观察法

观察法是一种通过直接观察用户在其实际工作环境中如何使用现有系统或执行相关任务来获取软件需求信息的方法。这种方法能够帮助需求工程师理解用户的实际工作流程、操作习惯和遇到的问题。观察法特别适用于了解用户日常工作中的非言语行为和潜在软件

需求,这些软件需求往往是通过访谈或问卷难以获取的。

1. 观察法的目标

观察法的主要目标包括以下几个。

- 获取真实反馈:通过观察用户的实际操作,获取真实的使用行为和软件需求。
- 发现隐性软件需求:发现用户自己未能意识到的软件需求或使用问题。
- 理解工作流程:深入理解用户的工作流程和操作习惯,识别流程中的瓶颈和改进点。
- 验证软件需求:验证从其他方法(如访谈或问卷)获取的软件需求,确保其准确性和完整性。

2. 观察法的类型

观察法可以分为多种类型,主要包括以下几种。

- 参与式观察:观察者直接参与到用户的工作中,与用户共同完成任务,能够获得第一手的操作体验和软件需求信息。
- 非参与式观察:观察者不参与用户的工作,仅作为旁观者观察用户的操作,避免干扰用户的正常工作。
- 直接观察:观察者在现场直接观察用户的操作,能够实时记录用户行为和环境。
- 间接观察:通过视频录制等方式进行观察,观察者在事后回看视频进行分析。

3. 实施观察法的步骤

实施观察法需要精心计划和准备,以确保观察结果的准确性和有效性。图 3-5 所示是实施观察法的步骤。

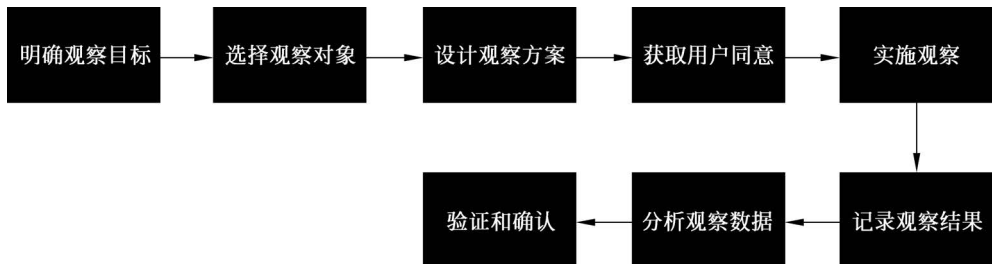


图 3-5 实施观察法的步骤

(1) 明确观察目标:确定观察的具体目标和预期结果,确保观察内容与目标一致。

(2) 选择观察对象:选择典型的用户作为观察对象,确保观察结果具有代表性。

(3) 设计观察方案。

- 观察内容:确定需要观察的具体内容和行为,如操作步骤、使用工具、交互方式等。
- 观察方法:选择适当的观察方法(参与式或非参与式、直接或间接)。
- 观察记录:设计记录表或使用录音机、录像机等工具记录观察内容。

(4) 获取用户同意:在进行观察前,必须获得用户的知情同意,解释观察的目的和内容,确保用户配合。

(5) 实施观察:按照设计好的方案进行观察,注意尽量不干扰用户的正常工作。

(6) 记录观察结果:实时记录用户的操作行为、问题和任何显著的细节,确保记录的准确性和完整性。

(7) 分析观察数据：对记录的观察数据进行整理和分析,提取关键软件需求和改进点。

(8) 验证和确认：将观察结果与用户和其他利益相关者进行验证和确认,确保软件需求的准确性和可行性。

4. 观察法的优缺点

1) 优点

- 真实操作行为：通过直接观察用户的实际操作行为,获取反映真实软件需求和使用情况的一手信息。
- 发现隐性软件需求：能够发现用户自己未能意识到的软件需求和问题。
- 全面理解：深入理解用户的工作流程和操作环境,有助于全面分析软件需求。
- 现场验证：能够现场验证软件需求和假设,确保软件需求的准确性。

2) 缺点

- 时间和成本高：观察法通常需要花费较多的时间和资源,特别是参与式观察。
- 有干扰风险：观察者的存在可能会干扰用户的正常操作,影响数据的真实性。
- 主观性强：观察结果可能受到观察者的主观判断影响,需要谨慎分析。
- 样本代表性不足：如果被观察的用户样本选择不合理,可能导致所收集信息偏离整体用户需求,影响分析结果的普适性和有效性。

5. 提高观察法效果的策略

为了提高观察法的效果,可以采取以下策略。

- 精心选择对象：选择具有代表性的用户和典型的使用场景,确保观察结果的广泛适用性。
- 明确记录标准：制定明确的记录标准和方法,确保记录的客观性和一致性。
- 减少干扰：尽量减少对用户正常工作的干扰,观察者应保持低调,避免影响用户操作。
- 多次观察：通过多次观察不同用户和不同场景,获取更加全面和可靠的数据。
- 结合其他方法：将观察法与访谈、问卷等方法结合使用,相互验证,确保软件需求的全面性和准确性。

6. 观察法的实例

为了更好地理解观察法在需求获取中的应用,以下是一个关于银行客户服务系统开发项目的实例。

1) 项目背景

一家银行计划开发一个新的客户服务系统,以提升客户服务效率和客户满意度。项目团队需要通过观察法收集前台客服人员的实际操作和 workflows,了解他们的具体需求。

2) 需求获取过程

(1) 确定观察目标。

- 目标：了解客服人员在处理客户事务时的实际操作流程,识别其中的瓶颈和痛点,收集对新系统的需求。
- 观察对象：前台客服人员,包括柜员和客户经理。

(2) 设计观察方案。

- 观察内容。
 - 客户办理不同业务(如开户、存款、贷款申请)时的操作流程。

- 客服人员使用现有系统的具体操作步骤。
- 客服人员与客户的互动和交流方式。
- 处理异常情况和问题的过程。

• 观察方式：非参与式观察，观察者不参与操作，只记录和分析客服人员的操作过程。

(3) 实施观察。

- 安排观察时间：在业务高峰期和非高峰期进行观察，以获取全面的操作数据。
- 记录观察结果：使用记录表详细记录每个操作步骤、遇到的问题以及客服人员的反馈。

(4) 分析观察数据。

- 整理记录：将观察记录整理成书面文档，确保信息的完整性和准确性。
- 识别需求：从记录中识别出具体的功能性需求和非功能性需求，如系统响应速度、界面友好性、操作简便性等。
- 编写观察报告：总结观察结果，形成观察报告，并与相关利益相关者分享，进行确认和反馈。

通过以上步骤，项目团队能够深入了解客服人员的实际工作流程和需求，为新系统的设计和开发提供翔实的需求数据。同时，观察法能够发现一些潜在的问题和需求，为系统的优化提供依据。

观察法是一种重要的软件需求获取方法，通过直接观察用户的实际操作行为，能够获取真实的软件需求信息和使用情况。尽管观察法需要较高的时间和资源投入，但其在发现隐性软件需求、全面理解用户工作流程等方面具有独特优势。通过精心设计和合理实施，观察法可以为软件需求获取提供宝贵的数据和洞见，确保系统能够更好地满足用户需求和期望。结合其他软件需求获取方法，观察法能够帮助需求工程师全面、准确地了解用户需求，提高软件项目的成功率。

3.2.4 工作坊

工作坊(Workshop)是一种通过组织利益相关者和开发团队在特定时间和地点进行集中讨论、交流和协作的软件需求获取方法。工作坊旨在通过面对面的互动，快速、有效地收集软件需求、解决问题、达成共识，促进团队之间的沟通与协作。工作坊通常由专业的主持人引导，采用多种互动方式，如头脑风暴、小组讨论、角色扮演等，确保所有参与者积极参与、充分表达意见。

1. 工作坊的目标

工作坊的主要目标包括以下几个。

- 快速收集软件需求：在短时间内收集到全面、详细的软件需求信息。
- 促进沟通与协作：通过面对面的互动，促进团队成员和利益相关者之间的沟通与协作。
- 达成共识：通过集中讨论，解决分歧，达成一致的软件需求和解决方案。
- 激发创意：通过互动和讨论，激发创新思维，提出创造性的解决方案。

2. 工作坊的类型

根据具体的软件需求和目标，工作坊可以分为以下几种类型。

- 软件需求收集工作坊：专注于收集和识别软件需求，通常在项目初期进行。
- 软件需求优先级设定工作坊：集中讨论和确定软件需求的优先级，确保关键软件的需求优先实现。
- 问题解决工作坊：针对特定的问题或挑战，集思广益，寻找解决方案。
- 原型评审工作坊：展示和评审系统原型，收集反馈，改进设计。

3. 组织工作坊的步骤

成功的工作坊需要精心策划和组织。图 3-6 所示是组织工作坊的步骤。

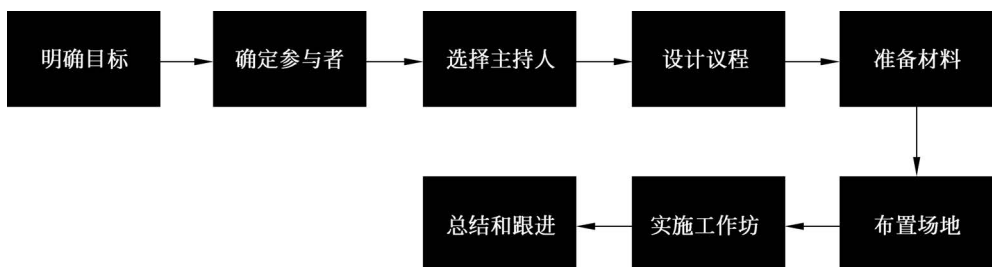


图 3-6 组织工作坊的步骤

- 明确目标：确定工作坊的具体目标和预期结果，确保活动的焦点明确。
- 确定参与者：选择合适的参与者，包括需求工程师、开发团队、用户代表、利益相关者等，确保各方代表参与。
- 选择主持人：选择经验丰富的主持人，负责引导讨论和控制会议进程。
- 设计议程：制定详细的工作坊议程，包括活动安排、讨论主题、时间分配等。
- 准备材料：准备必要的材料和工具，如白板、便笺、原型等。
- 布置场地：选择合适的场地，布置工作环境，确保参与者的舒适和便利。
- 实施工作坊：按照议程进行工作坊，引导讨论，记录关键信息，确保活动顺利进行。
- 总结和跟进：总结工作坊的成果，形成正式的行动计划，确保后续跟进和实施。

4. 工作坊的关键活动

在工作坊中，通常会采用多种互动方式和活动，以确保参与者的积极参与和讨论的深入。以下是一些常见的工作坊活动。

- 头脑风暴：鼓励参与者自由发表意见和创意，广泛收集软件需求和解决方案。
- 小组讨论：将参与者分成小组，围绕特定主题进行深入讨论，形成小组意见。
- 角色扮演：通过模拟用户场景和操作，帮助参与者理解用户需求和情境。
- 投票表决：对软件需求或方案进行投票表决，确定优先级和决策方向。
- 原型演示：展示系统原型，收集参与者的反馈和建议，改进设计。

5. 工作坊的优缺点

(1) 优点。

- 高效沟通：通过面对面的互动，促进高效沟通，快速收集软件需求和解决问题。
- 达成共识：集中讨论，有助于解决分歧，达成一致的软件需求和解决方案。
- 激发创意：通过集体讨论和头脑风暴，激发创新思维，提出创造性的解决方案。
- 增强团队合作：加强团队成员和利益相关者之间的合作与理解，增进团队凝聚力。

(2) 缺点。

- 组织复杂：工作坊的组织和实施需要精心策划和协调,涉及的准备工作较多。
- 时间成本高：需要投入大量的时间和人力资源,可能影响正常工作进程。
- 参与者依赖：工作坊的效果高度依赖参与者的积极性和参与度,可能存在个别参与者不积极或不配合的情况。
- 信息过载：在短时间内收集大量信息,可能导致信息过载,需要后续整理和分析。

6. 提高工作坊效果的策略

为了提高工作坊的效果,可以采取以下策略。

- 精心策划：提前制订详细的工作坊计划,明确目标、议程和参与者,确保活动的组织有序。
- 选择合适的主持人：选择有经验的主持人,引导讨论,控制会议进程,确保工作坊高效进行。
- 营造积极氛围：通过鼓励和引导,营造积极、开放的讨论氛围,激发参与者的积极性和创造力。
- 合理时间安排：合理安排工作坊的时间,确保每个环节有充足的时间进行讨论和总结。
- 有效记录：实时记录工作坊的讨论内容和关键结论,确保信息完整准确,便于后续整理和分析。
- 及时跟进：在工作坊结束后,及时整理和总结成果,形成正式的软件需求规格文档或行动计划,确保后续的跟进和实施。

7. 工作坊的实例

1) 项目背景

一家银行计划开发新一代手机应用,以提供更便捷和丰富的功能,提升用户体验。为此,项目团队组织了一次需求获取工作坊,邀请了各相关方共同参与讨论和分析需求。

2) 工作坊过程

(1) 准备阶段。

- 目的：明确工作坊的目标,准备相关材料。
- 参与者：项目经理、产品经理、需求工程师、用户代表(不同年龄段的客户)、技术团队、市场团队、客服团队。
- 材料准备：现有应用分析报告、用户反馈、市场调研数据、竞品分析资料。

(2) 工作坊实施。

- 介绍与目标设定：项目经理介绍工作坊目的和目标。
- 头脑风暴。
 - 用户故事：用户代表分享使用现有应用的体验和改进建议。
 - 痛点分析：讨论现有应用的不足和用户痛点。
- 需求优先级划分。
 - 功能性需求：通过讨论和投票确定关键功能性需求的优先级,如账户管理、转账汇款、贷款申请、理财产品购买等。
 - 非功能性需求：性能、安全性、易用性等。

- 原型设计与反馈。
 - 原型展示：设计团队展示初步原型,收集各方反馈。
 - 讨论与改进：根据反馈进行调整和改进。

(3) 总结与后续步骤。

- 总结会议：记录各方达成的一致意见和未解决的问题。
- 后续计划：撰写详细的后续计划,计划下一步工作,如需求验证和确认。

3) 实例结果

通过此次工作坊,项目团队不仅明确了手机应用的功能性需求和非功能性需求,还发现了一些潜在的问题和改进方向,为后续开发奠定了坚实的基础。

工作坊是一种高效的软件需求获取方法,通过组织利益相关者和开发团队在特定时间和地点进行集中讨论和协作,能够快速收集软件需求、解决问题、达成共识。尽管工作坊的组织和实施需要精心策划和投入较多的时间和资源,但其在高效沟通、达成共识、激发创意和增强团队合作等方面具有独特优势。通过精心设计和合理实施,工作坊可以为软件需求获取提供宝贵的数据和洞见,确保系统能够更好地满足用户需求和期望,提高软件项目的成功率。结合其他软件需求获取方法,工作坊能够帮助需求工程师全面、准确地了解用户需求,确保项目顺利进行。

3.2.5 原型开发与迭代验证

原型开发是软件开发过程中的一种关键方法,它通过创建系统的简化版本(即原型)来展示系统的主要功能和界面。这些原型可以是低保真度的草图,也可以是高保真度的可交互模型,其目的是让用户和利益相关者在开发初期就能够了解系统的外观和功能,从而提供反馈和意见。

原型开发与迭代验证是一种高效的软件需求获取和系统设计方法,通过创建系统的简化版本,让用户和利益相关者在开发初期就能够了解系统的外观和功能,从而提供反馈和意见。通过不断地迭代验证和改进,确保系统软件需求的准确性和完整性,提高用户满意度和项目成功率。尽管原型开发需要投入一定的时间和资源,但其在验证软件需求、减少风险、促进沟通等方面具有独特优势。在实际项目中,结合其他软件需求获取方法,原型开发可以帮助需求工程师全面、准确地了解用户需求,确保系统能够更好地满足用户期望和软件需求。

有关原型开发与迭代验证的具体内容请参看第8章。

3.2.6 焦点小组

焦点小组是一种定性研究方法,通常由一个训练有素的主持人带领一组特定的参与者进行深入讨论。其目的是通过互动和讨论,深入了解参与者的观点、感受、需求和动机。这种方法特别适用于软件需求获取过程中的探索性研究,以便更好地理解用户的真实软件需求和期望。

1. 焦点小组的目标

- 深入理解用户需求：通过参与者的讨论和互动,深入了解用户对系统功能和设计的真实软件需求。

- 发现潜在问题：在讨论过程中发现用户可能面临的潜在问题和挑战，提前识别软件需求和设计中的潜在风险。
- 获取多样化视角：通过多样化的参与者群体，获取不同背景、角色和经验的视角，确保软件需求的全面性和多样性。
- 促进用户参与：鼓励用户积极参与软件需求获取过程，增加用户对系统的认同感和满意度。

2. 焦点小组的组成

一个典型的焦点小组包括以下几个关键角色。

- 主持人：主持人负责引导讨论，确保讨论围绕既定主题展开，并鼓励所有参与者积极发言。主持人需要具备良好的沟通技巧和管理能力，以确保讨论的有序进行。
- 参与者：参与者是焦点小组的核心，他们通常是系统的潜在用户或利益相关者。参与者的选择应具备代表性，以确保讨论结果的广泛适用性。
- 记录员：记录员负责记录讨论的主要内容和关键点，可以通过笔记、录音或视频等方式记录讨论过程，以便后续分析和总结。
- 观察员：在一些情况下，项目团队的成员或其他利益相关者可以作为观察员参与，观察讨论过程，并记录自己的观察和见解。

3. 焦点小组的步骤

焦点小组的步骤如图 3-7 所示。

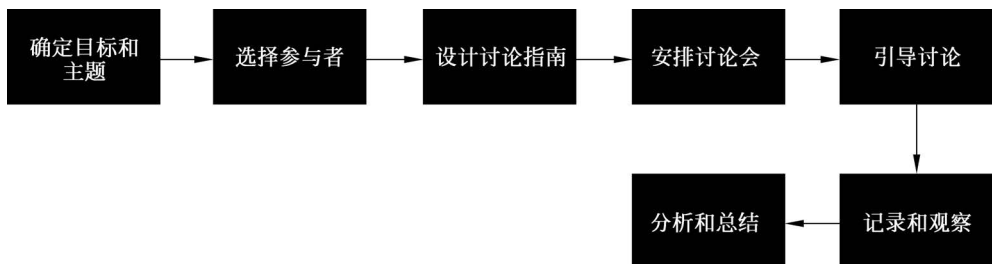


图 3-7 焦点小组的步骤

- 确定目标和主题：明确焦点小组的目标和讨论主题，确保讨论内容与软件需求获取的目标一致。
- 选择参与者：根据项目软件需求选择具有代表性的参与者，确保参与者的多样性和代表性。
- 设计讨论指南：制定详细的讨论指南，包括讨论的主要问题和话题，确保讨论围绕既定主题展开。
- 安排讨论会：选择合适的时间和地点安排讨论会，确保参与者能够方便地参加讨论。
- 引导讨论：主持人引导讨论，确保所有参与者都有机会发言，并鼓励参与者之间的互动和交流。
- 记录和观察：记录员和观察员记录讨论的主要内容和关键点，以便后续分析和总结。
- 分析和总结：对讨论记录进行分析，总结出用户的主要软件需求、问题和建议，形成软件需求规格文档和分析报告。

4. 焦点小组的优缺点

1) 优点

- 深入了解：通过互动和讨论，能够深入了解用户的真实软件需求、动机和期望。
- 快速获取反馈：在短时间内获取大量的用户反馈，有助于快速识别软件需求和问题。
- 多样化视角：参与者的多样性能够提供不同的视角和见解，确保软件需求的全面性和多样性。
- 用户参与：通过用户参与软件需求获取过程，增加用户对系统的认同感和满意度。

2) 缺点

- 组织复杂，资源投入高：焦点小组需要精心组织，包括招募合适的参与者、安排场地、记录与整理讨论内容等，整体协调成本较高。
- 结果易受主观影响：讨论过程易受到参与者主观偏见或群体心理影响，可能导致数据偏差。
- 主持人依赖性强：讨论质量在很大程度上取决于主持人的引导能力，其经验和技巧直接影响信息的准确性与完整性。
- 数据处理复杂：小组讨论通常产生大量非结构化的定性数据，需耗费大量精力进行整理、编辑与分析。

5. 提高焦点小组效果的策略

为了提高焦点小组的效果，可以采取以下策略。

- 选择合适的参与者：根据项目软件需求和目标选择具有代表性的参与者，确保讨论结果的广泛适用性。
- 精心设计讨论指南：设计详细的讨论指南，包括讨论的主要问题和话题，确保讨论围绕既定主题展开。
- 培训主持人：确保主持人具备良好的沟通技巧和管理能力，通过培训提高主持人的引导和管理能力。
- 充分准备：提前准备好讨论所需的材料和设备，确保讨论顺利进行。
- 鼓励互动：在讨论过程中鼓励参与者之间的互动和交流，确保所有参与者都有机会发言。
- 详细记录：通过笔记、录音或视频等方式详细记录讨论过程，确保数据的完整性和准确性。
- 系统分析：对讨论记录进行系统的分析和总结，提炼出用户的主要软件需求、问题和建议，形成软件需求规格文档和分析报告。

6. 实例：移动银行应用的焦点小组

1) 项目背景

一家大型银行计划推出一款移动银行应用，旨在为客户提供方便的移动金融服务。项目团队决定通过焦点小组方法收集用户需求，以确保应用能够满足用户期望。

2) 焦点小组过程

(1) 组建焦点小组。

- 参与者选择：银行现有客户，包括不同年龄段、职业和收入水平的代表，共 12 人。

- 组长：项目团队中的一名高级需求工程师负责主持讨论。

(2) 会议准备。

- 议题设计：准备一系列开放性问题,涵盖用户对移动银行应用的功能期望、使用习惯、体验反馈等。
- 场地与设备：选择舒适的会议室,准备录音设备和白板。

(3) 会议实施。

- 引导讨论：组长引导参与者讨论他们在使用其他移动银行应用时的体验、遇到的问题和期望的功能。
- 收集反馈：鼓励参与者分享具体的使用场景和需求,并记录关键点。

(4) 结果分析。

- 汇总意见：整理参与者的反馈,识别出高频需求和共性问题。
- 优先级划分：根据需求的重要性和实现难度对需求进行优先级排序。

(5) 总结。

- 用户期望功能：实时账户余额查询、转账功能、账单支付、投资理财服务。
- 常见问题：用户担心安全性、操作复杂性、页面加载速度等。

通过焦点小组,项目团队能够深入了解用户对移动银行应用的实际需求和习惯。这种方法不仅提供了丰富的定性数据,还帮助团队识别出用户关注的关键问题,为后续的需求分析和设计提供了宝贵的参考。

焦点小组是一种有效的定性研究方法,通过组织一组具有代表性的参与者进行深入讨论,能够深入了解用户的真实需求和期望。尽管焦点小组需要投入一定的时间和成本,但其在深入理解用户需求、发现潜在问题、获取多样化视角和促进用户参与等方面具有独特优势。在实际项目中,结合其他软件需求获取方法,焦点小组可以帮助需求工程师全面、准确地了解用户需求,确保系统能够更好地满足用户期望和软件需求。

3.2.7 调查分析与文档研究

1. 调查分析

调查分析是一种通过系统地收集和分析数据来了解软件需求的方法。它涉及设计调查问卷、分发问卷、收集回复以及分析数据。这种方法可以定量地衡量用户需求和期望,特别适用于需要从大量用户中收集意见和反馈的情况。

1) 调查分析的目标

- 获取广泛意见：从大量用户中收集数据,了解不同用户群体的需求和期望。
- 量化软件需求：通过定量分析,确定软件需求的优先级和重要性,为决策提供数据支持。
- 识别趋势和模式：分析数据中的趋势和模式,发现用户需求的共性和差异。
- 评估满意度：评估用户对现有系统或产品的满意度,识别需要改进的领域。

2) 调查分析的步骤

调查分析的步骤如图 3-8 所示。

- 定义目标：明确调查的目的和目标,确定需要了解的主要问题和软件需求。
- 设计问卷：设计结构合理、问题明确的问卷,确保问题易于理解并能有效获取所需信息。

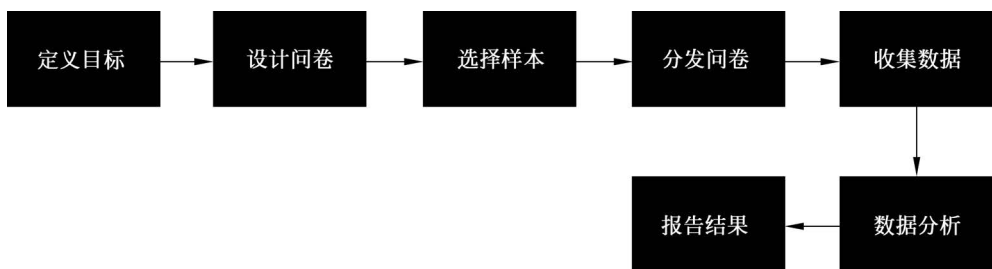


图 3-8 调查分析的步骤

- 选择样本：确定调查的目标用户群体，选择具有代表性的样本，确保数据的广泛性和代表性。
- 分发问卷：通过电子邮件、在线平台或纸质方式分发问卷，确保覆盖足够多的用户。
- 收集数据：收集用户填写的问卷数据，确保数据的完整性和准确性。
- 数据分析：使用统计方法对数据进行分析，提取有意义的结论和洞见。
- 报告结果：编写调查分析报告，展示数据分析的结果，为软件需求决策提供依据。

3) 问卷设计的要点

- 明确问题：确保每个问题都明确、具体，避免模棱两可的表达。
- 简洁明了：问题和选项应简洁明了，避免复杂的术语和长句。
- 避免偏见：设计问题时避免引导性语言，以免影响用户的回答。
- 多样化题型：使用多种题型（如选择题、评分题、开放式问题等），全面了解用户需求。
- 逻辑顺序：按逻辑顺序安排问题，确保用户回答时流畅自然。
- 测试问卷：在正式分发前进行测试，确保问卷设计合理，问题清晰易懂。

4) 调查分析的优缺点

(1) 优点。

- 广泛覆盖：能够覆盖大量用户，获取广泛的意见和反馈。
- 定量分析：通过定量数据分析，提供客观、量化的软件需求信息。
- 高效便捷：使用在线问卷工具可以高效、便捷地分发和收集数据。
- 用户匿名性：用户可以匿名回答问卷，减轻心理负担，获取真实反馈。

(2) 缺点。

- 设计难度大：问卷设计需要专业知识，设计不当可能影响数据质量。
- 响应率低：问卷调查的响应率可能较低，需要激励措施提高参与度。
- 有数据偏差：部分用户可能随意填写问卷，导致数据偏差和不准确。
- 定性不足：问卷调查主要提供定量数据，难以深入了解用户的具体软件需求和背景。

2. 文档研究

文档研究是通过查阅和分析现有文档资料来获取软件需求的方法。这些文档包括用户手册、业务流程文档、系统规格说明书、技术报告、市场研究报告等。文档研究是一种有效的辅助方法，可以为软件需求获取提供重要背景信息和参考资料。

1) 文档研究的目标

- 了解现状：通过查阅现有文档，了解当前系统或业务流程的现状和存在的问题。

- 获取背景信息：提供项目相关的背景信息，帮助理解软件需求的上下文和环境。
- 识别历史软件需求：通过历史文档，识别之前未解决或遗留的软件需求，为当前项目提供参考。
- 补充信息：补充其他软件需求获取方法的不足，提供全面、准确的软件需求信息。

2) 文档研究的步骤

文档研究的步骤如图 3-9 所示。

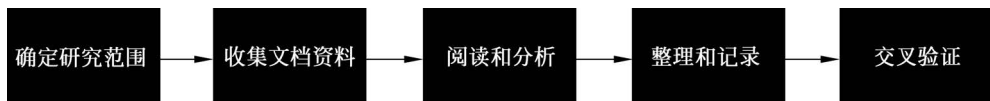


图 3-9 文档研究的步骤

- 确定研究范围：明确需要查阅和分析的文档类型和范围，确保文档研究的针对性。
- 收集文档资料：收集相关的文档资料，包括用户手册、业务流程文档、系统规格说明书等。
- 阅读和分析：仔细阅读和分析文档内容，提取与软件需求相关的信息和数据。
- 整理和记录：将提取的信息和数据整理和记录，形成系统的文档研究报告。
- 交叉验证：将文档研究的结果与其他软件需求获取方法的结果进行交叉验证，确保信息的准确性和一致性。

3) 文档研究的优缺点

(1) 优点。

- 历史参考：通过查阅历史文档，可以了解之前的软件需求和设计思路，为当前项目提供参考。
- 信息全面：文档通常包含详细的信息和数据，提供全面的背景资料。
- 成本较低：文档研究主要依赖已有资料，成本较低，适合辅助软件需求获取。
- 补充验证：可以作为其他软件需求获取方法的补充和验证，提高软件需求信息的准确性。

(2) 缺点。

- 信息过时：部分文档的信息可能过时，不适用于当前软件需求获取。
- 内容冗长：文档内容冗长，查阅和分析耗时较多，需要耐心和细致。
- 不够具体：文档研究主要提供背景信息和参考资料，难以获取具体、详细的用户需求。
- 依赖文档质量：文档质量不一，内容完整性和准确性直接影响研究结果。

4) 提高调查分析与文档研究效果的策略

为了提高调查分析与文档研究的效果，可以采取以下策略。

- 明确目标：在进行调查分析和文档研究前，明确目标和软件需求，确保研究的针对性和有效性。
- 专业设计：设计问卷时聘请专业人员，确保问卷设计合理，问题清晰易懂。
- 提高响应率：通过激励措施（如奖励、抽奖等）提高问卷调查的响应率，确保数据的代表性。
- 多渠道分发：利用多种渠道（如电子邮件、社交媒体、公司内部网等）分发问卷，覆盖

更多用户。

- 详细记录：在文档研究过程中，详细记录和整理提取的信息，确保数据的完整性和系统性。
- 交叉验证：将调查分析和文档研究的结果与其他软件需求获取方法的结果进行交叉验证，确保信息的准确性和一致性。
- 持续改进：根据项目进展和反馈，不断改进调查分析和文档研究的方法和策略，提高软件需求获取的质量和效果。

3. 实例：图书馆管理系统需求获取

1) 项目背景

某大学计划开发一个新的图书馆管理系统，以替代现有的老旧系统并提升管理效率和用户体验。项目团队需要通过调查分析和文档研究来获取系统需求。

2) 调查分析过程

(1) 问卷调查。

- 对象：学生、教职工、图书馆工作人员。
- 内容。
 - 学生和教职工：了解他们在借阅、查询、预约图书等方面的需求和痛点。
 - 图书馆工作人员：了解他们在书籍管理、借还流程、库存管理等方面的需求和改进意见。

(2) 文档研究。

- 现有系统文档：阅读并分析现有系统的使用手册、维护文档、需求规格说明书等，了解系统的功能、限制和问题。
- 相关标准和规范：查阅图书馆管理相关的国家或行业标准，确保新系统符合必要的规范要求。

3) 需求获取结果

通过调查分析和文档研究，项目团队收集并整理了如下需求。

(1) 功能性需求。

- 借还书流程简化，支持自助借还书。
- 提供在线预约、续借功能。
- 实现书籍库存管理的自动化，减少手工操作。

(2) 非功能性需求。

- 系统需具备高可用性，支持大并发访问。
- 数据安全性需得到保障，确保用户隐私和书籍信息不被泄露。

(3) 用户界面需求。

- 界面简洁易用，支持移动端访问。
- 提供多语言支持，满足国际学生的需求。

该实例展示了如何通过调查分析和文档研究，有效获取并整理系统需求，为后续的系统设计和开发提供坚实的基础。

调查分析与文档研究是两种有效的软件需求获取方法，分别通过定量数据和现有文档资料为软件需求获取提供支持。调查分析通过系统地收集和分析数据，能够广泛了解用户

需求和期望,特别适用于需要从大量用户中收集意见和反馈的情况。文档研究通过查阅和分析现有文档资料,提供重要的背景信息和参考资料,辅助软件需求获取过程。尽管两种方法各有优缺点,但在实际项目软件需求获取过程中,常常需要综合运用,以达到全面和准确的软件需求收集效果。这种综合运用不仅能够弥补单一方法的不足,还能通过相互验证提高软件需求获取的可靠性和有效性。

3.2.8 用户故事和敏捷方法

用户故事是一种简洁的软件需求描述方式,以非技术语言描述系统功能或特性,旨在通过用户的视角来明确软件需求。每个用户故事通常由以下 3 部分组成。

- 角色: 用户是谁?
- 目标: 用户想要做什么?
- 原因: 用户为什么需要这样做?

用户故事的格式为:“作为一个[角色],我想要[目标],因为[原因]。”这种简洁、清晰的描述方式有助于项目团队和利益相关者快速理解和讨论软件需求。

1. 用户故事的目标

- 明确软件需求: 通过用户故事,明确系统需要实现的功能和特性。
- 促进沟通: 用户故事使用非技术语言,促进开发团队与用户之间的有效沟通。
- 指导开发: 用户故事为开发团队提供具体的开发目标和验收标准。
- 迭代改进: 在敏捷开发过程中,用户故事可以不断细化和调整,以应对软件需求的变化。

2. 敏捷方法的定义

敏捷方法是一种迭代和增量的软件开发方法,强调灵活性、快速交付和用户参与。敏捷方法通过短周期的迭代(通常为 1~4 周的“冲刺”),逐步实现系统功能,并在每个迭代结束时交付可用的产品增量。敏捷方法的核心原则包括以下几个。

- 以客户为中心: 强调用户和客户的参与,确保开发的系统满足用户需求。
- 快速交付: 通过短周期的迭代,快速交付可用的产品增量。
- 持续改进: 不断收集反馈,调整和改进系统功能和特性。
- 团队协作: 强调团队成员之间的紧密协作和沟通。

3. 用户故事和敏捷方法的步骤

用户故事和敏捷方法的步骤如图 3-10 所示。

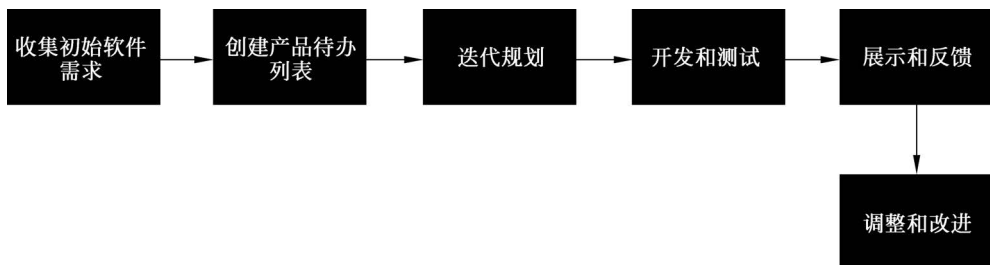


图 3-10 用户故事和敏捷方法的步骤

- 收集初始软件需求：与用户和利益相关者沟通，收集初步的软件需求和期望，形成用户故事。
- 创建产品待办列表：将收集到的用户故事整理成产品待办列表，按优先级排序。
- 迭代规划：在每次迭代开始前，开发团队与产品负责人一起选择高优先级的用户故事，制定迭代目标。
- 开发和测试：在迭代周期内，开发团队实现选定的用户故事，并进行测试和验证。
- 展示和反馈：在每次迭代结束时，开发团队向用户和利益相关者展示迭代成果，收集反馈意见。
- 调整和改进：根据反馈意见，调整和改进用户故事和产品待办列表，进入下一个迭代。

4. 用户故事的撰写技巧

- 明确角色：确保每个用户故事中都明确描述用户角色，了解用户的背景和软件需求。
- 具体目标：描述用户希望完成的具体任务或操作，确保目标清晰可实现。
- 突出价值：阐明用户完成任务后的预期收益，强调软件需求的商业价值和用户价值。
- 小而精：将用户故事分解为小而精的单元，确保每个用户故事在一个迭代内可以完成。

5. 敏捷方法的关键实践

- 每日站会：团队每天进行简短的站会，交流进展、问题和计划，确保信息透明和团队协作。
- 迭代回顾：每次迭代结束后进行回顾，总结经验教训，提出改进建议，持续优化开发过程。
- 看板和任务板：使用看板或任务板管理迭代中的任务，跟踪任务的状态和进展，确保任务按计划完成。
- 持续集成：在开发过程中持续集成代码，进行自动化测试，确保系统稳定并保证其质量。
- 用户参与：邀请用户和利益相关者参与迭代展示和反馈，确保系统满足实际软件需求。

6. 用户故事和敏捷方法的优缺点

1) 优点

- 灵活应变：敏捷方法强调灵活应对软件需求变化，能够快速响应用户反馈和市场变化。
- 用户参与：用户故事和敏捷方法强调用户和利益相关者的参与，确保系统满足用户需求。
- 快速交付：通过短周期的迭代，快速交付可用的产品增量，缩短产品上市时间。
- 高效沟通：用户故事使用非技术语言，促进开发团队和用户之间的有效沟通 and 理解。
- 持续改进：通过迭代回顾和持续反馈，不断优化和改进系统功能和特性。

2) 缺点

- 初期规划不足：敏捷方法强调迭代开发,可能导致初期规划不足,需要不断调整和改进。
- 团队要求高：敏捷方法对团队的自组织能力和协作要求高,需要经验丰富的团队成员。
- 用户依赖：敏捷方法依赖用户和利益相关者的持续参与和反馈,用户的参与度和反馈质量直接影响开发效果。
- 文档不足：敏捷方法强调工作软件而非文档,可能导致文档不够翔尽,需要额外的文档工作。

7. 提高用户故事和敏捷方法效果的策略

- 加强用户参与：确保用户和利益相关者持续参与软件需求获取和开发过程,提供及时的反馈和意见。
- 优化团队协作：通过培训和团队建设,提升团队成员的协作能力和自组织能力,确保高效开发。
- 精细用户故事：将用户故事分解为小而精的单元,确保每个用户故事在一个迭代内可以完成,提高开发效率。
- 制定验收标准：为每个用户故事制定清晰的验收标准,确保开发成果可以被验证和接受。
- 持续改进：定期进行迭代回顾,总结经验教训,提出改进建议,持续优化开发过程和团队协作。

8. 实例：青年租房管理系统的用户故事和敏捷方法

请扫描下方二维码查看本实例。



用户故事和敏捷方法是获取软件需求的有效工具,通过简洁的用户故事描述软件需求,使用敏捷方法迭代开发和验证系统功能,能够灵活应对软件需求变化,快速响应用户反馈,提高系统设计的准确性和用户满意度。尽管这两种方法对团队的协作能力和用户的参与度有较高要求,但通过合理的策略和实践,可以显著提高软件需求获取的效果,确保系统满足用户需求和商业目标。在实际项目中,结合其他软件需求获取方法,用户故事和敏捷方法能够帮助需求工程师全面、准确地了解用户需求,确保系统能够更好地满足用户期望和软件需求。

3.2.9 数据分析法

数据分析法是一种通过对现有数据进行收集、整理、分析和解释,来获取软件需求的方法。它依赖于对各种数据源的深入分析,包括系统日志、用户行为数据、业务数据、市场数据等,以发现用户需求、使用模式和潜在改进点。这种方法特别适用于大型系统和平台,在用户基数大、数据量丰富的情况下,通过数据分析可以提供精确的软件需求洞察。

1. 数据分析法的目标

- 理解用户行为：通过分析用户的操作记录和行为数据，了解用户的实际使用情况和软件需求。
- 发现使用模式：识别用户群体的使用模式和趋势，找出常见的使用场景和问题。
- 优化系统性能：通过分析系统性能数据，发现瓶颈和性能问题，提出改进建议。
- 提高用户满意度：基于数据分析的结果，优化用户体验，提升用户满意度。
- 支持决策：为产品和业务决策提供数据支持，指导产品开发和改进。

2. 数据分析法的步骤

数据分析法的步骤如图 3-11 所示。

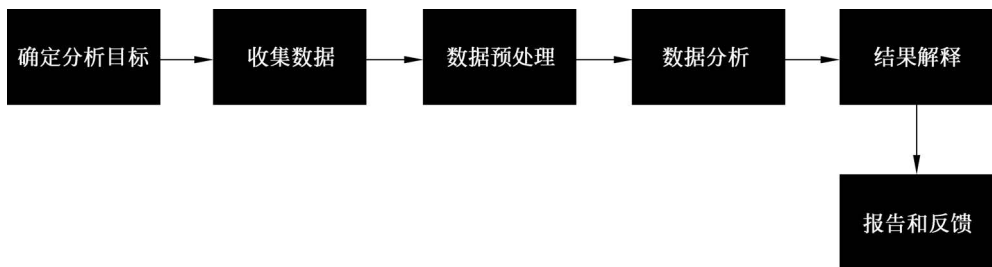


图 3-11 数据分析法的步骤

- 确定分析目标：明确数据分析的目的和目标，确定需要解决的问题和获取的软件需求。
- 收集数据：从不同的数据源收集相关数据，包括系统日志、用户行为数据、业务数据等。
- 数据预处理：对收集到的数据进行清洗、整理和格式化，处理缺失值和异常值，确保数据的质量和一致性。
- 数据分析：应用统计分析、数据挖掘、机器学习等方法对数据进行深入分析，提取有价值的信息和洞见。
- 结果解释：对分析结果进行解释和总结，形成具体的软件需求和改进建议。
- 报告和反馈：编写数据分析报告，展示分析过程和结果，向利益相关者汇报，并收集反馈意见。

3. 数据分析的方法和技术

- 描述性分析：通过描述性统计方法，对数据的基本特征进行总结和描述，包括均值、中位数、标准差等指标。
- 探索性分析：通过可视化和数据挖掘技术，探索数据中的模式和关系，发现潜在的软件需求和问题。
- 诊断性分析：通过因果分析和关联分析，查找问题的根本原因和影响因素，提出针对性的改进建议。
- 预测性分析：利用机器学习和统计建模技术，预测未来的趋势和软件需求变化，为产品规划和决策提供支持。
- 规范性分析：通过优化和仿真技术，提出最优的解决方案和改进措施，指导系统优化和改进。

4. 数据分析的工具和技术

- 数据收集工具：包括日志收集工具(如 ELK Stack)、数据仓库(如 Amazon Redshift)、API 等,用于收集和存储数据。
- 数据预处理工具：包括数据清洗和整理工具(如 Pandas、SQL),用于处理和准备数据。
- 数据分析工具：包括统计分析工具(如 R、SPSS)、数据挖掘工具(如 RapidMiner)、机器学习工具(如 Scikit-Learn、PyTorch)等,用于分析和挖掘数据。
- 数据可视化工具：包括可视化工具(如 Tableau、Power BI)、图表库(如 Matplotlib、D3.js),用于展示分析结果和洞见。
- 协作和报告工具：包括协作平台(如 Jupyter Notebook、Google Colab)、报告工具(如 Markdown、LaTeX),用于编写和分享数据分析报告。

5. 数据分析法的优缺点

1) 优点

- 基于客观数据：数据分析法依赖于客观数据提供准确和可靠的软件需求洞察。
- 发现隐藏的软件需求：通过深入分析数据,可以发现隐藏的软件需求和潜在问题,并提出针对性的改进建议。
- 支持决策：数据分析为产品和业务决策提供数据支持,帮助决策者做出明智的选择。
- 持续改进：数据分析是一个持续的过程,可以不断收集和分析新数据,持续优化和改进系统。

2) 缺点

- 数据质量依赖：数据分析的结果高度依赖于数据的质量,数据缺失、错误或不一致会影响分析结果的准确性。
- 技术要求高：数据分析需要掌握一定的统计、数据挖掘和编程技术,对分析人员的技能要求较高。
- 结果解释与落地难：分析所得结果往往较为抽象,若缺乏对业务背景的深入理解,可能难以准确解读并有效转化为可执行的需求或改进措施,降低实际价值。
- 隐私与合规风险：在处理用户数据时,若未妥善管理数据隐私和安全,将面临泄露风险及法律合规问题,带来潜在的法律责任和用户信任危机。

6. 提高数据分析法效果的策略

- 确保数据质量：在数据收集和预处理阶段,确保数据的准确性、一致性和完整性,处理缺失值和异常值,提高数据质量。
- 选择合适的方法：根据分析目标 and 数据特征,选择合适的分析方法和技术,确保分析结果的有效性和准确性。
- 多角度分析：从多个角度和维度对数据进行分析,全面了解软件需求和问题,避免片面和单一的结论。
- 结合业务背景：在解释和应用分析结果时,结合业务背景和用户需求,提出实际可行的软件需求和改进措施。
- 注重数据隐私和安全：在数据收集和分析过程中,遵守数据隐私和安全规定,确保

用户数据的保护和合规性。

- 持续监控和改进：定期收集和分析新数据，持续监控系统性能和用户需求，进行持续改进和优化。

7. 实例：社交媒体应用的数据分析

1) 项目背景

一家初创公司计划开发一款新的社交媒体应用，旨在为用户提供更具互动性和个性化的社交体验。项目团队需要通过数据分析了解用户需求，确保应用功能符合市场需求。

2) 数据分析过程

(1) 数据收集。

- 从现有的社交媒体平台收集用户行为数据，包括用户活跃时间、互动频率、内容偏好等。
- 使用在线问卷和调查工具，收集潜在用户对新应用的期望和需求。
- 访问公开的行业报告和市场研究，获取社交媒体市场的最新趋势和统计数据。

(2) 数据清洗。

- 删除重复数据和无效数据。
- 处理缺失数据，确保数据完整性。
- 规范数据格式，统一数据单位和类型。

(3) 数据分析。

- 描述性分析：使用统计工具对用户行为数据进行描述性统计，计算平均值、中位数、标准差等，描述用户行为的总体特征。
- 探索性数据分析(EDA)：使用可视化工具(如 Python 的 Matplotlib、Seaborn)对数据进行可视化分析，发现潜在的模式和关系。例如，分析用户活跃时间分布，发现高峰使用时间。
- 相关性分析：分析不同变量之间的相关性，如用户活跃时间与互动频率之间的关系，帮助识别关键影响因素。

(4) 需求提炼。

- 根据数据分析结果，提炼出用户对新应用的核心需求。例如，用户希望有更多的实时互动功能、更好的隐私保护措施、个性化推荐内容等。
- 确定优先级，根据用户需求的重要性和技术实现的可行性，制订需求实现计划。

(5) 报告和反馈。

- 撰写数据分析报告，详细描述数据收集、清洗、分析的过程及结果。
- 向项目团队和利益相关者汇报分析结果，征求反馈意见，进一步完善需求文档。

通过数据分析法，项目团队能够系统地收集和分析用户的行为数据，深入了解用户需求和市场趋势，从而为社交媒体应用的开发提供科学依据和数据支持。这种方法不仅提高了需求获取的准确性，还帮助团队识别潜在的市场机会和确定产品改进方向。

数据分析法是一种通过对现有数据进行深入分析来获取软件需求的方法，适用于大型系统和平台。通过数据分析，可以客观、准确地了解用户需求和模式，发现隐藏的问题和改进点，为产品和业务决策提供数据支持。尽管数据分析对数据质量和分析技能有较高要求，但通过合理的策略和实践，可以显著提高软件需求获取的效果，确保系统设计符合用

户需求和商业目标。在实际项目中,结合其他软件需求获取方法,数据分析法能够帮助需求工程师全面、准确地了解用户需求,确保系统能够更好地满足用户期望和软件需求。

3.3 软件需求获取的技巧与策略

请扫描下方二维码查看本节的内容。



3.4 软件需求获取工具和技术

请扫描下方二维码查看本节的内容。



本章小结

本章对软件需求获取的核心过程进行了系统性梳理,重点分析了需求获取的各种方法和技术手段。首先,明确了软件需求获取在项目初期的重要性,接着详细探讨了不同需求获取技术的应用场景与优缺点,包括访谈、问卷调查、观察法、用户故事等方法的具体操作步骤。还讨论了利益相关者的角色及其在需求获取中的关键作用,强调了与最终用户、客户及其他相关方的密切沟通对需求准确性的影响。

通过本章的学习,读者能够理解并掌握需求获取的基本方法,深入认识到需求获取不仅是简单的信息收集,更是一个迭代、不断完善的过程。系统性地获取准确且全面的需求,为后续的软件设计、开发和验证奠定了坚实的基础。

习 题

请扫描下方二维码在线答题。

