

浅谈计算机应用软件的的开发和维护

杨瑞青

硅湖职业技术学院, 江苏 昆山 215300

[摘要]从个人日常生活到企业关键运营, 应用软件已经渗透到几乎所有领域, 因此迫切需要深入了解如何有效地开发和维护这些软件, 满足不断变化的需求和挑战。文章旨在探讨计算机应用软件的的开发和维护过程中涉及的关键方面, 无论是从项目规划、需求分析、编码阶段, 还是后续的维护、性能优化和问题修复, 将研究如何建立高效的开发团队, 采用最佳实践, 从而保障软件的可靠性和质量。

[关键词]计算机; 应用软件; 开发; 维护

DOI: 10.33142/sca.v6i10.10203

中图分类号: TP31-4

文献标识码: A

Brief Discussion on the Development and Maintenance of Computer Application Software

YANG Ruiqing

Silicon Lake Vocational & Technical Institute, Suzhou, Jiangsu, 215330, China

Abstract: From personal daily life to critical operations of enterprises, application software has penetrated almost all fields, so it is urgent to have a deep understanding of how to effectively develop and maintain these software to meet constantly changing needs and challenges. The article aims to explore the key aspects involved in the development and maintenance process of computer application software, whether from project planning, requirement analysis, coding stage, or subsequent maintenance, performance optimization, and problem repair, which will study how to establish an efficient development team and adopt the best practices to ensure the reliability and quality of the software.

Keywords: computer; application software; development; maintenance

随着科技的不断演进, 软件在生活中扮演着越来越重要的角色, 涵盖了从日常工作到娱乐和社交的各个领域。软件开发不仅需要技术专长, 还需要深刻理解用户需求和业务流程。维护软件同样重要, 因为软件的寿命通常远远超出其初始开发阶段, 维护涉及修复漏洞、添加新功能、适应新的硬件和操作系统等任务。通过研究软件开发和维护的关键方面, 希望为开发人员、项目经理和决策者提供有价值的见解, 帮助他们在这个不断发展的领域取得成功。

1 计算机应用软件的概述

1.1 定义

计算机应用软件是一种以代码形式存在的程序, 旨在计算机上执行各种任务和功能。这些软件程序可以用来处理数据、提供娱乐、自动化业务流程、解决问题以及执行各种其他任务。应用软件通常设计用于特定领域或任务, 满足用户的需求。

1.2 分类

操作系统软件: 操作系统是计算机的核心软件, 负责管理硬件资源, 提供用户界面, 并允许其他软件程序在计算机上运行。

应用软件: 应用软件是为特定任务或需求而设计的软件, 包括办公套件 (如 Microsoft Office)、图形设计工具 (如 Adobe Photoshop) 和数据库管理系统 (如 Oracle) 等^[1]。

娱乐软件: 这些软件用于提供娱乐和休闲活动, 例如视频游戏、多媒体播放器和社交媒体应用。

2 计算机应用软件开发的现存问题

2.1 技术问题

2.1.1 跨平台兼容性问题

在多平台的环境中, 开发人员必须确保他们的应用程序能够在不同操作系统、浏览器和设备上正常运行。不同平台之间的差异导致应用程序在某些情况下无法正常工作, 这需要大量的测试和调试工作。

2.1.2 技术债务

随着软件的迭代开发, 一些开发团队会忽略代码的质量和可维护性, 导致技术债务的积累。技术债务会降低开发效率, 增加维护成本, 并导致应用程序的不稳定性。

2.1.3 快速技术变革

计算机领域的技术变革非常迅速, 新技术和框架不断涌现。开发人员需要不断学习和适应这些新技术以保持竞争力, 这需要额外的时间和资源。

2.2 管理问题

2.2.1 不合理的时间表压力

有时开发项目会受到不合理的时间表压力, 导致开发团队不得不以牺牲质量为代价来满足紧迫的截止日期。这会导致软件质量下降和项目失败的风险增加。

2.2.2 范围蔓延

项目范围蔓延是指在项目进行过程中不断添加新功能或要求,而未充分评估其影响。这导致项目超出预算和进度延误。

2.2.3 团队沟通问题

团队之间的不良沟通或合作问题导致信息不对称、决策延误和项目延期。良好的沟通和协作对于项目的成功至关重要^[2]。

2.3 质量问题

2.3.1 稳定性问题

软件中的错误和漏洞导致应用程序崩溃或不稳定。这会损害用户体验,降低用户满意度。

2.3.2 性能问题

软件的性能问题导致响应时间过长、资源占用过多,或无法处理大规模数据,性能问题会限制应用程序的可用性和扩展性。

2.3.3 可维护性问题

如果软件代码缺乏结构、文档和规范,将导致后续维护和更新的困难,这会增加维护成本并降低软件的寿命。

3 计算机应用软件开发

3.1 有效处理技术问题

计算机应用软件开发中有效处理技术问题至关重要,因为这些问题影响软件的性能、可维护性和用户体验。

跨平台兼容性是当今应用软件开发中的一个主要挑战,在一个多元化的计算环境中,确保应用程序在各种操作系统、浏览器和设备上能够正常运行,需要采取一系列策略。首先,开发人员需要采用跨平台的开发框架和技术以减少不同平台之间的差异。例如,使用 Web 技术创建跨平台的 Web 应用程序,在不同设备上提供一致的用户体验。此外,跨平台应用程序开发工具如 React Native 和 Flutter 帮助开发人员创建一次代码,多平台运行的应用程序。其次,测试是解决跨平台兼容性问题的关键,开发团队应建立全面的测试策略包括各种设备和浏览器的测试,从而保障应用程序在不同环境下稳定运行,自动化测试工具可以帮助减少测试的重复性,提高测试覆盖率。最重要的是,开发团队应密切关注平台供应商的更新和发布,不同平台会定期推出新的版本和更新,开发人员需要及时适应这些变化,从而保障应用程序在最新的环境中仍然可用^[3]。

技术债务是指在软件开发过程中,为了迅速交付功能而采取的折中措施,牺牲代码质量和可维护性。技术债务的累积会导致后续开发和维护工作的困难,因此需要采取措施来处理和减少技术债务。首先,开发团队应定期进行代码审查和重构,代码审查可以帮助发现潜在的问题和改进代码质量,而重构是指重新设计和重写代码,提高可读性和可维护性,这些实践有助于降低技术债务的积累。其次,采用最佳实践和设计原则可以预防技术债务的产生,

例如采用面向对象编程、分层架构和单一职责原则可以提高代码的清晰度和可维护性。此外,开发团队应充分了解项目的需求和目标,以便权衡时间、资源和质量,在制定项目计划时应充分考虑技术债务的管理,从而保障项目在截止日期内交付。

计算机技术领域的迅猛发展导致了新技术和框架的不断涌现。如何有效地处理这种快速技术变革是技术问题的另一个关键方面。首先,开发团队需要建立学习和适应新技术的文化,持续学习和培训对开发人员来说至关重要,跟踪新技术的发展,工作中的学习和知识分享会帮助团队保持竞争力。其次,开发团队应保持对新技术趋势的敏感性,参与技术社区、参加行业会议和关注技术新闻是了解新技术趋势的好方法,以便团队及时了解最新的技术,决定是否将其引入项目中。

3.2 规范处理管理问题

时间表压力是软件开发项目中常见的管理问题,经常导致项目延期、质量下降和团队不满。如何规范处理时间表压力是至关重要的。首先,项目管理团队应合理制定项目时间表,时间表应基于项目的实际需求和复杂性来制定,而不是仅仅根据上级要求的截止日期,合理的时间表应包括项目各个阶段的时间估计包括需求分析、设计、开发、测试和上线,同时应考虑风险因素,以便在时间表中留出适当的缓冲时间。其次,时间表的管理需要持续的监控和调整,项目管理团队应定期跟踪项目的进展,识别潜在的延期风险并采取措​​施来应对这些风险。另外,沟通也是处理时间表压力的关键,开发团队应与项目管理团队保持密切的沟通,及时报告问题和进度,从而能够及早发现问题并采取纠正措施。最后,项目管理团队和利益相关者应理解软件开发的复杂性和不确定性,在一些情况下的时间表需要调整,从而保障项目的成功^[4]。

范围蔓延是管理问题中的另一个常见挑战。首先,项目管理团队应建立清晰的需求管理流程,确保所有需求都经过充分的需求分析和评估以确定其优先级和影响,需要建立变更控制程序以便在项目进行过程中处理新需求,并确保其与原始范围的一致性。其次,项目管理团队和开发人员需要了解范围蔓延的潜在风险以及如何防止和管理它,培训可以帮助团队更好地理解需求管理和变更控制的重要性。另外,项目管理团队需要与利益相关者建立有效的沟通渠道,定期的需求审查会议,从而保障所有相关方都了解项目的需求和范围,在变更请求时沟通可以帮助相关方理解变更的影响,以便做出明智的决策。最后,范围蔓延的风险评估和决策应该在项目计划中考虑,项目管理团队应预留适当的资源和时间来处理潜在的范围蔓延,减少其对项目的负面影响。

团队之间的不良沟通或合作问题导致信息不对称、决策延误和项目延期。首先,建立清晰的沟通渠道和流程是解决团队沟通问题的第一步,团队成员应明确了解如何报

告问题、寻求支持和与其他团队成员合作，因此要制定沟通计划、定义沟通责任和建立问题追踪系统。其次，培训和培训可以帮助团队成员提高沟通技巧和协作能力，培训包括冲突解决、团队建设和有效沟通的课程，以便有效改善团队之间的合作和信息共享。

3.3 解决质量问题

稳定性问题是指软件中的错误和漏洞，导致应用程序崩溃或不稳定。解决稳定性问题需要采取一系列措施。首先开发团队应实施全面的测试计划，这包括单元测试、集成测试和系统测试，从而保障所有功能都经过充分测试，自动化测试工具可以帮助减少测试的重复性，提高测试覆盖率。其次，错误和漏洞的管理是解决稳定性问题的关键，开发团队应建立问题跟踪系统以记录、跟踪和分配问题，问题应经过仔细的分析来确定其严重性和紧急性，解决问题的优先级应基于其影响程度和用户需求。另外，持续集成和持续交付（CI/CD）实践可以帮助早期发现和解决稳定性问题，CI/CD 流程可从而保障每次代码更改都经过自动化测试和部署，从而降低错误进入生产环境的风险。最重要的是，团队应建立质量文化，将质量置于首要位置，质量意识应贯穿整个开发团队，从需求分析到设计和编码，直到测试和维护，团队成员应明白质量的重要性，并努力达到高质量的软件产品^[5]。

性能问题导致软件响应时间过长、资源占用过多，或无法处理大规模数据。解决性能问题需要采取多方面的措施。性能测试是解决性能问题的关键，开发团队应建立性能测试计划包括负载测试、性能剖析和压力测试，这些测试可以帮助发现性能瓶颈和瓶颈，以便采取相应的措施。其次，开发团队应使用性能监控工具来监视应用程序的性能，识别瓶颈并进行性能优化，优化包括改进代码、数据库查询、缓存和硬件资源的使用。另外，使用适当的缓存策略可以减少数据库查询和数据传输的次数，从而提高性能，负载均衡可从而保障应用程序在多个服务器上均匀分布以处理高流量。

可维护性问题涉及到软件代码的结构、文档和规范。解决可维护性问题需要采取一系列的措施。首先，开发团队应采用最佳的编码实践，包括命名约定、单一职责原则和面向对象编程，代码审查是一种有效的方式，用来发现和改进代码质量。其次，开发团队应编写清晰和详尽的文档，包括代码注释、技术文档和用户手册，这有助于后续开发和维护工作的顺利进行。另外，采用合适的开发工具和集成开发环境可以提高可维护性，这些工具可以帮助开发人员更好地管理代码、版本控制和代码重构。

4 如何做好计算机应用软件维护工作

4.1 建立清晰的维护策略

在进行软件维护之前，首要任务是建立清晰的维护策略。维护策略应明确定义维护的目标、范围和优先级，从而保障工作有重点地进行。维护策略的目标包括性能优化、

错误修复、功能升级或安全增强，同时策略还应明确规定维护的范围，即需要维护哪些部分、改进哪些功能以及解决哪些问题。最重要的是，策略应明确确定维护工作的优先级，从而保障最关键的问题首先得到解决。

4.2 建立高效的维护团队

一个高效的维护团队是成功维护软件的核心。这个团队应该包括经验丰富的开发人员、测试人员和技术支持人员，他们应具备广泛的技术知识和问题解决能力。团队成员之间的有效沟通和协作也是非常重要的。他们需要迅速分享信息、讨论问题并协同解决挑战，良好的协作有助于提高维护工作的效率^[6]。

4.3 采用版本控制和文档管理

在维护过程中，版本控制和文档管理是必不可少的，版本控制系统可以帮助团队跟踪代码变更，记录每个版本的修改，以便追溯和恢复以前的状态，文档管理系统则有助于组织和维护项目文档、技术文档和用户手册。版本控制系统应具备以下特点：支持多用户协作，允许团队成员同时编辑和提交代码；提供版本历史记录，了解每个版本的修改内容；支持分支和合并，从而保障维护工作不会影响主干开发。

5 结语

综上所述，计算机应用软件的开发和维护是一项复杂而不断演进的工作，在这个数字时代，软件已经成为人们生活 and 商业运营的不可或缺的一部分，因此了解如何有效地开发和维护软件至关重要。在这个过程中，建立清晰的维护策略、拥有高效的维护团队、采用版本控制和文档管理、进行系统性测试以及建立持续改进的文化都是关键要素。只有通过综合考虑这些因素，软件开发和维护才能取得成功。通过不断学习和适应新技术、工具和最佳实践，更好地满足用户需求，提高软件质量并保持竞争力。

[参考文献]

- [1] 张昊. 计算机应用软件的开发及维护研究[J]. 造纸装备及材料, 2021, 50(1): 177-178.
- [2] 吴金玲. 刍议计算机应用软件的开发和维护[J]. 计算机产品与流通, 2020(2): 23.
- [3] 辛立新. 计算机应用软件的开发和维护[J]. 数码世界, 2019(3): 28.
- [4] 顾正庶. 计算机应用软件的开发和维护[J]. 山东工业技术, 2019(5): 154.
- [5] 曾嵘娟. 试谈计算机应用软件的开发和维护[J]. 电脑编程技巧与维护, 2018(10): 46-48.
- [6] 万波. 计算机应用软件的开发和维护策略分析[J]. 无线互联科技, 2018, 15(19): 33-34.

作者简介：杨瑞青（1994.9—）女，汉族，中央民族大学硕士毕业，现就职于硅湖职业技术学院，职位为计算机学院专任教师，研究方向：计算机应用技术。