

软件工程管理与经济虚拟仿真实验平台——软件规模度量组件的设计与实现

摘 要

本文以软件工程经济学虚拟实验教学平台的开发为背景，设计并实现了一个度量软件规模的虚拟仿真实验模块。对国内外现有基于软件的度量方法的现状、优缺点进行比较分析，对基于软件的度量模型进行相关的理论研究，本文探索了软件规模度量对软件开发的重要性。对常用软件模型的测量方法进行总结和分类，比较分析其使用的优缺点，并对软件工程经济学虚拟实验教学平台的软件规模度量组件提出了合理的实验设计。通过对 IFPUG 和 NESMA 方法的详细分析和分类，将研究五种用户功能类型以及应用手段，并将 IFPUG 和 NESMA 方法用于整个实验设计。同时，结合工作量评估和软件成本计算方法，总结了实验模型，使参与实验的学生能够理解和熟悉常用的软件规模度量的方法与计算过程，加深课程知识点的掌握。

关键词：虚拟仿真实验，软件规模度量，软件工程经济学

装

订

线

Software Engineering Management and Economic Virtual Simulation Experiment Platform-Design and Implementation of Software Scale Measurement Components

ABSTRACT

Based on the development of the software engineering economics virtual experiment teaching platform, this paper designs and implements a virtual simulation experiment module that measures the software scale. The current situation, advantages and disadvantages of the existing software-based measurement methods at home and abroad are compared and analyzed, and the software-based measurement model is related theoretically. This article explores the importance of software scale measurement to software development. The measurement methods of commonly used software models are summarized and classified, the advantages and disadvantages of their use are compared and analyzed, and a reasonable experimental design is proposed for the software scale measurement component of the virtual experimental teaching platform of software engineering economics. Through detailed analysis and classification of IFPUG and NESMA methods, five types of user functions and application methods will be studied, and IFPUG and NESMA methods will be used in the entire experimental design. At the same time, combined with workload evaluation and software cost calculation methods, the experimental model is summarized, so that students participating in the experiment can understand and familiarize themselves with commonly used software scale measurement methods and calculation processes, and deepen their grasp of course knowledge.

Key words: virtual simulation experiment, software scale measurement, software engineering economics

目 录

| | | |
|-------|-------------------------|----|
| 1 | 引 言 | 1 |
| 1.1 | 选题背景 | 1 |
| 1.2 | 虚拟仿真实验教学平台应用现状以及当前面临的问题 | 1 |
| 1.2.1 | 虚拟仿真实验教学平台应用现状 | 1 |
| 1.2.2 | 虚拟仿真实验教学平台当前面临的问题 | 3 |
| 1.3 | 国内外软件规模度量应用现状 | 3 |
| 1.4 | 论文研究内容 | 3 |
| 1.4.1 | 研究目标 | 3 |
| 1.4.2 | 研究内容 | 4 |
| 1.4.3 | 主要工作 | 4 |
| 1.5 | 论文结构 | 4 |
| 2 | 文献综述 | 6 |
| 2.1 | 软件规模度量理论 | 6 |
| 2.1.1 | 软件规模度量的定义 | 6 |
| 2.1.2 | 软件规模度量的特性 | 6 |
| 2.1.3 | 软件规模度量能够解决的问题 | 6 |
| 2.1.4 | 软件规模度量带来的新问题 | 7 |
| 2.2 | 常用软件规模度量方法 | 7 |
| 2.2.1 | IFPUG 功能点分析法 | 7 |
| 2.2.2 | NESMA 功能点方法 | 7 |
| 2.2.3 | Mark II 功能点方法 | 8 |
| 2.2.4 | 代码行方法 (LOC) | 8 |
| 2.2.5 | 德尔菲方法 | 8 |
| 2.2.6 | COCOMO II 模型 | 8 |
| 2.3 | 软件规模度量方法的比较 | 9 |
| 2.4 | 本章小结 | 10 |
| 3 | 需求分析 | 11 |
| 3.1 | 功能性需求 | 11 |
| 3.1.1 | 用例图 | 11 |
| 3.1.2 | 用例规约 | 11 |
| 3.2 | 非功能性需求 | 15 |
| 3.2.1 | 数据精确度 | 15 |
| 3.2.2 | 可使用性及兼容性 | 15 |
| 3.2.3 | 时间特性 | 15 |
| 3.2.4 | 可维护性 | 15 |
| 3.2.5 | 安全性 | 15 |
| 3.3 | 需求验证 | 16 |
| 3.3.1 | 一致性 | 16 |
| 3.3.2 | 完整性 | 16 |
| 3.3.3 | 现实性 | 16 |
| 3.3.4 | 有效性 | 16 |
| 3.4 | 本章小结 | 16 |
| 4 | 设计解决方案 | 17 |

| | | |
|-------|---------------------|----|
| 4.1 | IFPUG 软件规模度量方法 | 17 |
| 4.1.1 | 概述 | 17 |
| 4.1.2 | IFPUG 法工作流程 | 17 |
| 4.1.3 | 用户功能类型 | 18 |
| 4.1.4 | 确定系统边界及范围 | 18 |
| 4.1.5 | 计算数据功能点及其复杂度 | 19 |
| 4.1.6 | 计算事务功能点及其复杂度 | 19 |
| 4.1.7 | 计算初始功能点 | 20 |
| 4.1.8 | 计算软件规模 | 20 |
| 4.1.9 | 实验设计 | 21 |
| 4.2 | NESAM 软件规模度量方法 | 21 |
| 4.2.1 | 概述 | 21 |
| 4.2.2 | NESMA 方法工作流程 | 22 |
| 4.2.3 | 估算方法 | 22 |
| 4.2.4 | 实验设计 | 23 |
| 4.3 | 工作量估算 | 24 |
| 4.3.1 | 工作量估算方法 | 24 |
| 4.3.2 | 实验设计 | 24 |
| 4.4 | 软件开发成本估算 | 24 |
| 4.5 | 界面规约 | 25 |
| 4.5.1 | 学生相关界面 | 25 |
| 4.5.2 | 助教相关界面 | 25 |
| 4.7 | 数据库设计 | 26 |
| 4.7.1 | student 表 | 26 |
| 4.7.2 | teacher 表 | 26 |
| 4.7.3 | score 表 | 26 |
| 4.7.4 | experiment 表 | 27 |
| 4.7.5 | question 表 | 27 |
| 4.7.6 | experiment_manage 表 | 27 |
| 4.8 | 本章小结 | 27 |
| 5 | 系统的实现 | 28 |
| 5.1 | 系统软件结构 | 28 |
| 5.1.1 | 系统架构 | 28 |
| 5.1.2 | 体系结构 | 28 |
| 5.2 | 虚拟仿真实验方案的实现 | 29 |
| 5.2.1 | 模拟的项目基本信息介绍 | 29 |
| 5.2.2 | 应得出的软件规模度量结果 | 31 |
| 5.3 | 代码实现 | 32 |
| 5.3.1 | 用户登录 | 32 |
| 5.3.2 | 实验流程 | 32 |
| 5.3.3 | 实验/用户管理 | 34 |
| 5.4 | 本章小结 | 34 |
| 6 | 系统的测试 | 35 |
| 6.1 | 测试环境 | 35 |
| 6.2 | 功能测试 | 35 |
| 6.3 | 性能测试 | 36 |
| 6.4 | 可用性测试 | 36 |
| 6.5 | 本章小结 | 36 |

| | |
|--------------|----|
| 7 总结与展望..... | 37 |
| 7.1 总结..... | 37 |
| 7.2 展望..... | 37 |
| 参考文献..... | 38 |
| 谢 辞..... | 39 |

装

订

线

1 引言

本章简单介绍了目前虚拟仿真实验教学平台面临的问题及软件规模度量的国内外现状和研究意义，章末阐述本文的研究内容特色与结构。

1.1 选题背景

根据教育部《教育信息化十年发展规划(2011-2020 年)》，2013 年我国启动了国家级虚拟仿真实验教学中心建设工作，将虚拟仿真技术引入到本科实验教学当中^[1]。发展规划为各大高校建设虚拟仿真实验教学平台指明了方向，突出强调要建设优质的数字教育资源，达成能够发布高质量的网络课程和高精准度的实验系的目的。近年来，虚拟仿真实验教学也由此逐步成为了高等教育信息化建设和实验教学示范中心建设的重要组成部分^[2]。

在建立软件工程管理与经济虚拟仿真实验平台的过程中，软件规模度量实验是其重要组成部分之一。软件规模度量是指对软件规模、软件项目工作量、软件生产率、软件项目开发成本、软件质量、软件发布日期等进行量化，使复杂的软件过程能够通过数字的描述使相关人员进行正确理解和管理^[3]。软件度量需要同时满足项目管理、组织和用户的需求。在软件规模度量的过程中，项目经理能够根据软件测量的数据，合理地部署和配置相关资源，有效地监控项目的进度和执行情况，确定软件产品是否满足质量要求；同时，根据测量的数据，组织可以清楚地了解开发效率和质量的整体水平，以便更好地进行产品组合，确定资金的投入，规划、管理或验证软件开发活动；最后，根据实测数据，用户可以准确地确定投入资金、合理的项目交付期限和提交项目的质量。因此，对软件规模度量的研究具有十分重要的社会意义和应用意义。

在软件度量学科中，软件规模的度量是其他软件度量任务的基础和关键。在软件生产过程中，因为客户不理解其中涉及到的许多复杂步骤，所以会与开发人员对项目的“规模”存在认知矛盾。开发商在项目初期并不知道项目的规模，客户在项目开发之初就需要知道项目的规模，才能掌握开发成本等问题，所以开发商在这个时候一定要对客户做出一定程度的保证。对于开发商来说，控制项目，确保项目能够按照预期的计划和要求进行，显得尤为重要。软件规模的度量为软件工作量、成本和进度提供了初始输入数据。因此，准确的软件项目规模估算直接影响到项目的后续工作。作为软件工程专业的学生，研究软件规模度量方法，并通过虚拟仿真实验准确应用，显得尤为重要。

在本次毕业设计中，应通过对软件规模度量理论以及现有方法的深入学习、对比，总结出软件规模度量组件实验模型，使其应用于软件工程管理与经济虚拟仿真实验平台中。

1.2 虚拟仿真实验教学平台应用现状以及当前面临的问题

1.2.1 虚拟仿真实验教学平台应用现状

教育与科学技术的发展密切相关，人们利用计算机技术和网络技术，从理念到实现网上实验教学只有短短十几年的时间。国内外许多著名大学已经很早的开展了虚拟仿真实验教学平台的研究

究和建设工 作，对构建在线实验教学系统的技术也相对成熟，并且取得了较为丰硕成果。

下表总结了国内外部分虚拟仿真实验教学平台应用现状及优缺点对比：

表 1.1 虚仿平台应用现状对比如例列表

| 高校虚仿平台名称 | 简要介绍 | 优点 | 缺陷 |
|------------------------------|---|--|---|
| 麻省理工学院 Web Lab 远程实验室 | 提供用于进行微电子学和电路设计课程的实验教学，允许学生在本人计算机上设计并修改电路模型。 | 学生可以通过手机远程控制实验；用 Java 激活 Web 浏览器来使用在远程实验室里的昂贵的测试设备获取测试数据，验证设计。 | 实验操作真实感不强，难以展现真实的实验环境，缺乏沉浸感。 |
| VLAB | 美国俄勒冈大学物理系主办的物理实验网站，包含天体物理、能量与环境、力学、热学等方面的几十种虚拟实验。 | 在真实实验前预演操作，提升试验成功率，弥补场地设备短缺问题。 | 实验内容多而复杂，使学生缺少明确目标。 |
| 清华大学 材料科学与工程 实验教学中心 | 创新性地在本科教学中开展高危、高消耗、极端条件下材料制备、分析及加工的虚拟仿真实验教学。 | 沉浸式 3D 立体显示系统、专业 3D 计算机，并且可以在多专业、多学科、多学院之间进行资源共享，最大化发挥实验室效用。 | 虚拟实验绝大多数针对某门课程进行设计，同一学科内没有形成系列化的虚拟实验课程体系。 |
| 山西财经大学 虚拟仿真实验项目 教学平台 | 深入推进信息技术与实验教学的深度融合、不断加强实验教学优质资源建设与应用。 | 界面美观大方，实验流程完善。 | 现有虚仿实验的共享仅局限于较小的地域范围和较窄的专业领域，开放性和实用性均不足。 |
| 农林植物学虚拟仿 真实验教学中心 | 华南农业大学对信息化实验教学资源进行整合，升级网络教学平台，有效保障本校植物学实验网络课程教学需求。 | 功能较为完善，发挥了学科优势，丰富了实验教学内容。 | 师生使用率不高，自由与互动性差 界面老旧，实验平台开放共享性不足。 |
| 大连理工大学 软件工程虚拟仿真 实验教学中心 | 中心信息化设备和资源齐全，软件工程虚拟仿真实验教学高成本、高消耗、综合训练，形成横跨基础技能训练、综合能力培养、学科前沿探索和的软件工程实践教学体系。 | 实现了课程和软件资源共享、信息发布、数据分析等功能，基于案例的教学，面向多学科、多年级、多层次人才进行培养。 | 师生使用率不高，缺少有效评价；网页建设和宣传力度不足 |

1.2.2 虚拟仿真实验教学平台当前面临的问题

通过对比以上部分具有相对代表性的虚拟仿真实验教学平台，可以发现现有软件工程专业虚拟实验教学存在问题，结如下：

- （1）虚拟实验只针对本校单一课程设计，同一学科缺乏标准的虚拟实验教学项目；
- （2）现有虚拟实验教学项目共享范围小（仅供本课程使用），开放性和实用性有待提高；
- （3）部分高校开发的虚拟实验项目在实验操作中真实感较弱，难以展现真实的、高精度的实验环境，使学生缺乏沉浸感；
- （4）部分实验平台界面老旧、开放共享性不足，师生使用率不高，自由与互动性差，缺少有效课程评价。

1.3 国内外软件规模度量应用现状

我国国内从事软件规模度量的研究人员较少，国内培训和课程也较少。一项对香港 50 家商业公司进行调查的案例研究结果显示，79%的企业进行了实际的软件项目评估，21%的企业没有进行科学的软件项目评估。在我国，代码行法被广泛应用于软件规模的度量，但代码行法在应用中存在一些固有的缺陷。例如，在大规模软件开发的早期阶段，由于早期尚未产生一定量的代码行，所以在各方面的度量都不是很准确。此外，中国现在有大量的外包业务，对于业务供需双方来说，迫切需要一些标准化测量的模型和工具组件。而功能点法是在确定软件需求后，对软件规模进行度量的一种方法。从需求说明计算出功能点比计算出代码行更容易，而且可以使用功能点作为基础，计算出以代码行表示的规模度量值^[4]。

在日本、印度和韩国，功能点法的应用非常普遍。因此，这些国家的外包业务发展非常迅速。为了与世界接轨，国内一些大型企业和外包公司也对软件规模的度量进行了改进。另外，软件规模度量也是满足 CMM 4 级或 5 级要求不可缺少的一部分，因此有必要采用功能点法。然而，国内很多企业仍然采用代码行法，因为功能点法的使用需要相关培训和专家的配合，这是很多企业不愿克服的障碍。然而，随着科学技术的不断发展，国内行业协会也在积极推进软件规模度量的标准化和通用化，并陆续翻译出版了许多国外书籍，使功能点法在我国的应用迅速发展。

1.4 论文研究内容

1.4.1 研究目标

研究国内外软件规模度量标准及行业应用案例，设计一个较为完整的软件规模度量虚拟仿真教学实验。实验标准采用 IFPUG 软件规模度量标准和 NESMA 软件规模度量标准，对目标软件开发项目做软件规模的功能点测量，获得实验结果，验证实验结果并得出实验结论。实验目的是使得参与实验的学生能够理解和熟悉常用的软件规模度量的方法与计算过程，加深课程知识点的掌握。

1.4.2 研究内容

（1）学习软件规模度量理论以及现有方法，包括 IFPUG 功能点方法、NESMA 功能点方法、Mark II 功能点方法、代码行方法(LOC)、德尔菲方法(Delphi)、COCOMO II 模型等，并对比分析其优劣。

（2）参照国家标准，设计和构造本课题软件规模度量组件模型。根据，目标项目需求，采用多种软件规模度量方法，测量目标项目软件规模度量参数，经交叉验证得到软件功能点规模数据；

（3）设计和开发软件规模度量实验项目组件，并完成组件的测试、调试和部署。

1.4.3 主要工作

（1）研究现有主流的软件规模度量模型和方法，了解其规模度量原理及实际使用方式，并且结合实际项目进行相关调整，找到最适合的逻辑；

（2）参照国家软件开发推荐标准，同时根据大学实验教学规律及特点并结合软件工程专业培养方案及课程教学大纲，设计构造本课题软件规模度量组件模型。根据实际项目需求，以及不同软件规模度量方法，探索给出适用于本项目的软件规模度量组件模型和参数；

（3）设计、开发软件度量实验项目组件，并完成相应的测试、集成等工作；

（4）撰写毕业论文。

1.5 论文结构

通过对各种软件估算方法的研究，本文将基于功能点方法，将其应用到虚拟仿真实验教学平台中去，并对其做出一定的改进，使学生在使用虚拟仿真实验教学平台进行软件规模度量实验时能更好地理解其原理，以达到良好的实践效果。

毕业设计论文主要内容安排如下：

第一章：项目背景简介。调研目前虚拟仿真实验教学平台的短板及问题，结合项目背景阐述软件规模度量的意义和任务，阐明本项研究的意义。

第二章：文献综述。详细介绍软件规模度量理论以及现有方法的估算规则和使用步骤，包括 IFPUG 功能点方法、NESMA 功能点方法、Mark II 功能点方法、代码行方法(LOC)、德尔菲方法(Delphi)、COCOMO II 模型等，并对比分析其优劣。

第三章：需求分析。从虚拟仿真实验的问题域出发，凝练本项研究拟解决的问题。确定本课题需求，包括功能需求、性能需求、可靠性和可用性需求、安全性需求等等，并对提出的需求进行验证。

第四章：解决方案设计。参照国家标准及行业最佳实践，构造解决本课题问题的解空间，即软件规模度量模型框架。根据该框架，完成系统构件设计。

第五章：系统实现(构建)。根据实际项目需求，以及不同软件规模度量方法，作为虚拟仿真实验平台的实验项目之一，探索给出适用于本项目的软件规模度量组件模型和参数，并根据组件

的概要设计以及详细设计，完成组件的开发编码工作。

第六章：系统的验证。测试，以及测试后的相应修订、回归测试、调试和部署。

第七章：总结与展望。基于本文工作的阶段性总结，并且提出有待继续研究的课题。

装

订

线

2 文献综述

本章详细介绍了软件规模度量理论的相关特性、目标应用场景和可能带来的新问题；同时聚焦于几种常用的软件规模度量方法，包括 IFPUG 功能点方法、NESMA 功能点方法、Mark II 功能点方法、代码行方法(LOC)、德尔菲方法(Delphi)、COCOMO II 模型等，并对比分析其优劣。

2.1 软件规模度量理论

2.1.1 软件规模度量的定义

规模估算通常估算软件有“多大”。这是通过统计特征、功能点、代码行或对象的数目并赋予适当的权引，从而得到一个较为具体的数字来代表软件的规模^[5]。

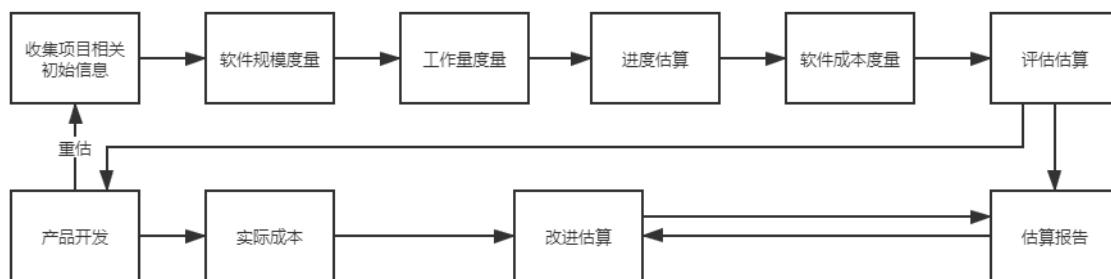


图 2.1 软件开发过程中软件规模度量的一般流程

2.1.2 软件规模度量的特性

（1）软件规模度量能够量化软件的大小：通过定量、定性等度量方法，能够实现对软件规模的大小，进而计算软件项目的研发周期和研发成本，为软件预算编制、招投标、价格谈判、项目管理、提高开发效率等提供帮助。

（2）软件规模度量不是软件开发固有的活动：软件开发的固有活动包括需求分析、设计、编码、测试等，并不包括软件规模度量，度量的成果不会作为产品或服务的一部分交付客户。

（3）软件规模度量需要有成本：软件规模度量需要投入开发团队、项目管理人员、质量保证人员、企业管理人员的时间，以及工具和基础设施的投入。

2.1.3 软件规模度量能够解决的问题

（1）在制定项目预算、确定招投标价格方案等活动中，由于缺乏具有科学性以及统一性的软件规模度量标准，导致制定的项目预算或招投标价格不合理，进而导致预算浪费或预算不足，或恶意投标、低价投标、流标等。

（2）在项目实施过程中，由于缺乏科学的成本控制依据，项目进度往往滞后，成本会远超初始预算。

（3）在企业推行工程化管理或精益管理过程中，由于缺乏软件开发效率、软件交付能力等模型数据，很难为企业制定合适的项目实施计划，进而无法动态跟踪成本、工作量、进度等情况。

（4）在企业管理中，很难对企业的开发能力、工作效率、软件缺陷数量进行量化考核。

2.1.4 软件规模度量带来的新问题

（1）增加项目成本或第三方评估费用：无论由甲方或乙方或是第三方对软件功能规模进行度量，都会带来人力成本的消耗，从而增加项目研发成本。

（2）容易引导不正常的利益驱动：如果度量结果和绩效等直接挂钩后，容易带来不正常的利益驱动。比如，某企业根据度量软件代码行数来计算员工的生产效率和工作量，或是甲方根据代码行数来计算乙方的研发费用，都必然出现的结果就是，带来了大量冗余代码、无用代码等。

（3）可能影响软件的使用体验或软件质量：软件研发企业（乙方）在追求软件功能点规模的时候，可能会牺牲软件的用户体验，进而影响软件的质量。比如，某企业采用功能点方法度量软件，原则上软件功能规模与开发实现方式无关，但在实际工作中，程序开发人员可能会为了多计功能点的利益，损失软件的用户体验或软件质量，换取较大的软件功能规模。

总而言之，软件度量是软件造价、量化管理、精益管理的必由之路，但应尽可能降低带来的不利影响，持续研究和改进软件度量方法，建立健全各类标准体系和操作指南。作为软件工程专业的学生，软件规模度量将会伴随学习工作生涯，由此，研修、学习、模拟，进行软件规模度量虚拟仿真实验教学是一项非常重要的举措，同时也会为提升学生的全面大局观打下良好的基础。

2.2 常用软件规模度量方法

2.2.1 IFPUG 功能点分析法

1979 年，IBM 公司的研究员 Allan Albrecht 开发了一种软件规模度量算法，即功能点分析法。1984 年，国际性功能点法研究组织(International Function Point Users Group, 简称 IFPUG) 成立，继承了 Albrecht 的研究成果，同时首次发布了功能点计算方法的相关手册。根据 IFPUG 方法模型，软件系统的功能由“逻辑数据文件”和“事务处理功能”组成。前者包括 ILF 和 EIF 的基本功能，后者包括 EI、EO 和 EQ 的基本功能，用户首先根据规则识别这五个基本功能，然后确定功能复杂度权重因子、每个功能的类型和权重因子，据此计算得出相关功能点数。具体方法流程将在在后文详细分析。

2.2.2 NESMA 功能点方法

NESMA（Netherland Software Measurement Association，荷兰软件度量协会）的功能点标准经由 IFPUG 标准演化而来，但在一些规则细节上与 IFPUG 不完全相同。NESMA 功能点估算方法于 2003 年升级为 ISO 标准，与 IFPUG 方法相比更侧重在项目前期进行估算。另外，针对 IFPUG 方法分析过程复杂、计算工作量大的缺点，NESMA 方法实现了快速计算。具体方法流程将在在后文详细分析。

2.2.3 Mark II 功能点方法

Mark II 功能点法起源于 Charles Symons 在 1991 年创作的一本书《Software Sizing and Estimating: MK II Function Point Analysis》。1988 年, Charles Symons 认为 IFPUG 在文件处理中有更多的主观因素, 所以他提出改进 IFPUG 功能点分析法, 最后形成了 IFPUG 功能点分析法的变种 Mark II 功能点分析法。Mark II 功能点法减少了功能点对文件处理的主观性, 保证了整个系统测量和部分总测量的一致性。同时, Symons 还提出修改 GSC 的基本特征值, 将 14 个特征值增加到 19 个特征值, 从而使得不同的应用领域有不同的权重因子。Mark II 功能点法适用于任何逻辑事务可以被确定的软件类型。

2.2.4 代码行方法 (LOC)

代码行 (Lines of Code, LOC) 是软件开发人员最早的软件规模度量方法之一。代码行是指可执行源代码行的个数, 包括可交付工作控制语言语句、数据定义、数据类型声明、等价声明、输入输出格式描述等等, 一个代码行的值和每个人每月的平均代码行数可以反映一个软件组织的生产能力, 组织可以审核历史数据来计算单行代码的值。在早期的系统开发中, 开发者大部分时间是在写代码, 而且代码也是开发过程中最直观的产出, 只需要在项目结束时计算代码行数, 就很容易确认估算是否准确^[6]。

2.2.5 德尔菲方法

德尔菲法, 是以古希腊城市德尔菲(Delphi) 命名的规定程序专家调查法^[7]。德尔菲法又称专家法, 是一种专家评估技术, 常常在没有历史数据的情况下评定项目。德尔菲法是由组织者就拟定的问题设计调查表, 通过函件分别向选定的专家组成员征询调查^[8]。专家组成员按照规定程序通过组织者反馈材料匿名交流意见。通过几轮咨询和反馈, 不断迭代, 专家意见逐渐统一, 最终得出具有统计意义的集体判断结果。

2.2.6 COCOMO II 模型

COCOMO (Constructive Cost Model, 构造性成本模型) 是一种精确的、易于使用的基于模型的成本估算方法; 而 COCOMO II 模型是对经典 COCOMO 模型的彻底更新, 反映了现代软件过程与构造方法^[9]。COCOMO II 模型事实上是一个估算模型的层次体系, 在基本了解软件功能后, 利用功能点技术求出项目的功能点, 进而计算源代码的行数来估算软件工作量和进度, 度量软件工作量和进度。在软件工程各阶段均可使用 COCOMO II 模型进行规模的估算。

2.3 软件规模度量方法的比较

上文总结了几种常见的软件规模度量方法的起源以及基本的使用方法、应用场景等等。注意到软件规模度量方法总体可以分为两大类，包含上文未提到的其他方法，具体如下图所示：

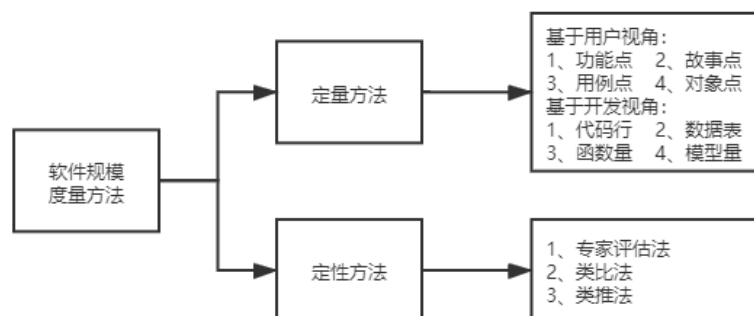


图 2.2 软件规模度量方法的分类

其中，用户视角指的是使用用户的语言对用户业务需求进行的一种正规描述。用户视角是一个业务功能的描述、是用户容易理解和识别的、可以用来计算功能点，也可以在物理形式上变化（例如：事务处理目录、建议书、需求文档、外部规格说明、详细规格说明、用户手册）。用户视角就是只关注用户需要的功能，不关程序员通过什么技术、工具、架构来实现用户的功能。开发视角指的是使用程序员的语言对用户业务需求进行的一种描述。开发视角：是一个业务开发过程的描述、是用户是难以理解的、很难用来计算功能点，但也可以在物理形式上变化的（例如：数据库关系表、业务逻辑图等）。对开发者而言，往往不自主的想到软件的架构、编程语言、数据库表、数据逻辑、页面展示等过程，关心怎么实现功能。习惯于从数据层、业务逻辑层、展现层等多个维度来估算软件的工作量。

除了由专家参与的德尔菲法之外，上文提到的 IFPUG、NESMA、Mark II 都属于功能点方法，COCOMO II 模型也与功能点、代码行息息相关，所以对于本文想要建立的软件工程管理与经济虚拟仿真实验平台来说，代码行方法和功能点方法是比较实用并适合具体研究的两种方法。

通过对各类算法的分析，下表将对对比总结代码行方法和功能点方法的使用特点以及优缺点，具体如下表所示：

表 2.1 代码行和功能点方法优缺点对比列表

| 软件规模 度量方法 | 优点 | 缺点 |
|--------------|---|---|
| 代码行 方法 | 1、估算准确，对于一个已经完成的项目来说，代码行的估算具有较强的稳定性，多次估算的结果一致性好。 2、代码行方法估算具有较强的客观性，不会因人而异。 | 1、实际代码行必须在项目开发出来后才能确定，但是软件规模度量是在项目开始阶段就急需的。在项目早期无法精确的使用代码行方法来估算项目规模。 2、代码行与开发语言密切相关，难以在各开发语言之间通过代码行的差异比较规模大小。 3、不同语言之间的代码行换算困难。 4、现代开发工具会自动生成大量代码，这些代码如何换算成代码行度量，现在没有标准。 |
| 功能点 方法 | 1、功能点方法是用户需求驱动的，所以在项目早期就可以对软件规模进行度量。 2、功能点方法的度量结果独立于具体技术实现，因此可以在不同的系统之间使用统一标准进行比较。 | 1、参数给定主观性较强，对同一个项目的估算可能因人而异。 2、参数值的给定区间过于简单，估算误差可能较大。 |

比较可知，代码行方法更注重在事后估算，功能点方法则更加侧重软件开发期间提前的估算。虚拟仿真实验教学平台将会详细研究功能点方法。

2.4 本章小结

本章详细介绍了软件规模度量理论的三大特性、在企业管理中能够解决的问题和可能带来的新问题；同时对比分析了几种常用的软件规模度量方法，包括 IFPUG 功能点方法、NESMA 功能点方法、Mark II 功能点方法、代码行方法(LOC)、德尔菲方法(Delphi)、COCOMO II 模型等，总结了它们之间的联系和区别，提出了本次虚拟仿真实验之软件规模度量组件需要使用的主要方法。

3 需求分析

从问题域出发，凝练出本课题的需求。确定本课题的综合需求，包括功能需求、性能需求、可靠性和可用性需求、接口需求等。

3.1 功能性需求

3.1.1 用例图

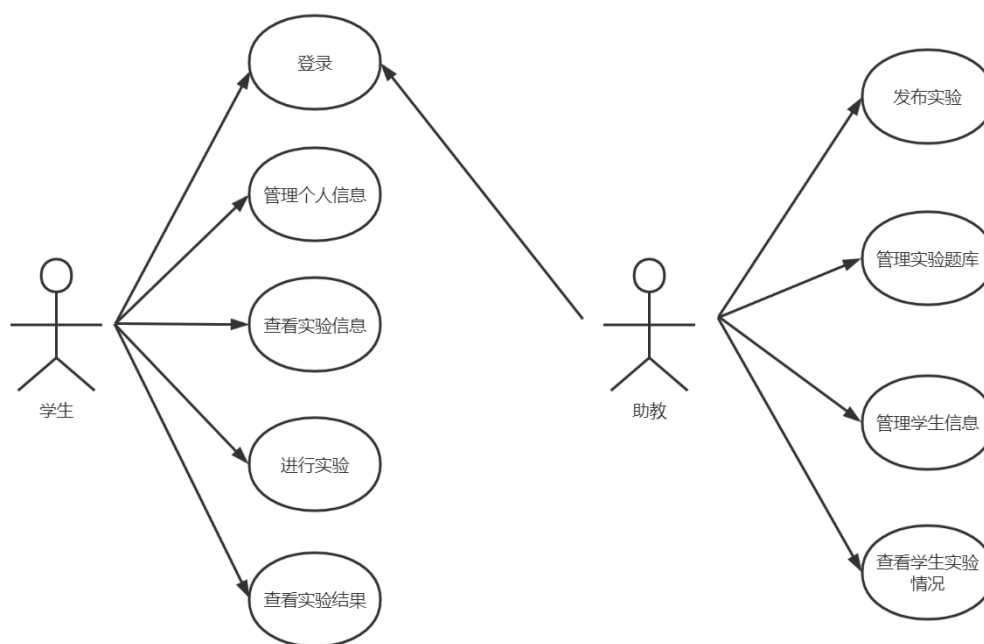


图 2.2 软件规模度量组件用例图

3.1.2 用例规约

| | |
|----------|---|
| Use Case | Uc01 登录 |
| 主要参与者 | 助教，学生 |
| 前置条件 | 用户正常打开虚拟仿真实验教学平台，进入登录界面。 |
| 基本事件流 | 1.使用者点击登录； 2.使用者输入学号和密码并确定； 3.系统查找数据库，确定该用户是否在数据库中。 |
| 后置条件 | 若用户存在数据库中则登陆成功进入主页面；若用户不存在，则系统提示使用者：该用户不存在。 |

| | |
|----|---|
| 扩展 | <p>a.使用者输入的学号与密码不匹配：</p> <p>a1.系统提示使用者：该号码与姓名不匹配；</p> <p>a2.重新进入登录界面，提示使用者重新输入学号及密码。</p> <p>b.使用者要求更换学号登录：</p> <p>b1.使用者点击返回，返回登录界面，重新输入。</p> |
|----|---|

| | |
|----------|--|
| Use Case | Uc02 管理个人信息 |
| 主要参与者 | 学生 |
| 前置条件 | 用户登录系统成功。 |
| 基本事件流 | <p>1.使用者选择管理个人信息；</p> <p>2.进入个人信息管理界面，选择修改用户信息；</p> <p>3.输入新的信息并确认。</p> |
| 后置条件 | 系统更新用户数据库。 |
| 扩展 | <p>a.使用者要求修改密码：</p> <p>a1.使用者进入信息管理界面后，选择密码；</p> <p>a2.系统提示输入原密码，使用者输入并确认；</p> <p>a3.若原密码输入正确，则进入密码修改界面；使用者输入新密码，再次输入新密码，并确定修改完成。系统核查两次密码，如果一致，系统更新用户数据库；如果不一致，系统提示使用者：两次密码输入不一致；</p> <p>a4.若原密码输入不正确，则系统提示使用者密码输入不正确。</p> <p>b.使用者要求取消管理个人信息：</p> <p>b1.使用者确认取消；</p> <p>b2.系统退出个人信息管理界面，用户信息维持原样。</p> |

| | |
|----------|---|
| Use Case | Uc03 查看实验信息 |
| 主要参与者 | 学生 |
| 前置条件 | 用户登录系统成功。 |
| 基本事件流 | <p>1.用户进入实验界面；</p> <p>2.系统调出数据库相关实验的基本信息和拓展信息。</p> <p>3.系统将信息通过实验界面显示给用户。</p> |
| 后置条件 | 用户浏览实验信息操作成功。 |

| | |
|----|--|
| 扩展 | a.用户要求取消浏览实验操作： a1.用户选择取消浏览实验； a2.系统返回主界面。 |
|----|--|

| | |
|----------|--|
| Use Case | Uc04 进行实验 |
| 主要参与者 | 学生 |
| 前置条件 | 用户登录系统成功。 |
| 基本事件流 | 1.用户进入实验操作界面； 2.按照实验流程，填写不同方法下的实验结果。 3.实验完成后提交，显示实验成绩。 |
| 后置条件 | 用户进行实验成功。 |
| 扩展 | a.用户要求取消实验操作： a1.用户选择取消实验； a2.系统返回主界面。 |

| | |
|----------|--|
| Use Case | Uc05 查看实验结果 |
| 主要参与者 | 学生 |
| 前置条件 | 用户登录系统成功。 |
| 基本事件流 | 1.用户进入实验结果界面； 2.系统显示已参与过的实验成绩。 |
| 后置条件 | 用户查看实验结果成功。 |
| 扩展 | a.用户要求取消查看实验结果操作： a1.用户选择取消查看实验结果； a2.系统返回主界面。 |

| | |
|----------|-----------------------------|
| Use Case | Uc06 发布实验 |
| 主要参与者 | 助教 |
| 前置条件 | 助教成功登录账号。 |
| 基本事件流 | 1.助教选择发布新实验； 2.助教填写实验信息； |

| | |
|------|--------------------------|
| | 3.助教点击提交; |
| 后置条件 | 系统将实验信息存入数据库, 界面展示新发布实验。 |
| 扩展 | a.助教选择取消发布。 |

| | |
|----------|--|
| Use Case | Uc07 管理实验题库 |
| 主要参与者 | 助教 |
| 前置条件 | 助教成功登录账号。 |
| 基本事件流 | 1.助教选择对实验题库操作; 2.助教填写实验题库信息; 3.助教点击提交; |
| 后置条件 | 系统将实验信息存入数据库, 界面展示新发布实验题库。 |
| 扩展 | a.助教选择取消发布。 |

| | |
|----------|--|
| Use Case | Uc08 管理学生信息 |
| 主要参与者 | 助教 |
| 前置条件 | 助教成功登陆账号。 |
| 基本事件流 | 1.助教进入学生信息管理界面。 2.助教选择要进行的操作。 3.助教进行操作。 4.助教选择提交。 5.系统进行审核。 |
| 后置条件 | 系统更改数据库中用户信息。 |
| 扩展 | a.助教取消操作并返回。 b.助教选择查看用户信息。 c.助教选择更改用户信息。 d.助教选择删除用户信息。 e.助教选择增加用户信息。 |

| | |
|----------|---|
| Use Case | Uc09 查看学生实验情况 |
| 主要参与者 | 助教 |
| 前置条件 | 助教成功登陆账号。 |
| 基本事件流 | 1.助教进入学生实验成绩界面。 2.助教选择要进行的操作。 3.助教进行操作。 |

| | |
|------|------------------------|
| | 4.助教选择提交。 5.系统进行审核。 |
| 后置条件 | 助教查看学生实验成绩成功。 |
| 扩展 | 无。 |

3.2 非功能性需求

3.2.1 数据精确度

软件工程经济学虚拟仿真实验教学平台在进行实验的过程中，填写录入系统的实验数据信息必须保证经过数据格式和现实逻辑的校验，并且数据显示的格式必须符合现实认知，同时必要数据的精确程度不会过于复杂或者是过于简化，而是符合现实常识和逻辑需要。

3.2.2 可使用性及兼容性

软件工程经济学虚拟仿真实验教学平台需要保证在标准测试合格后才发布上线供学生使用。除了未发现的意外错误，软件工程经济学虚拟仿真实验教学平台不应该在业务流程中出错，也不应该出现与承诺功能不相符的实际服务。此外，软件工程经济学虚拟仿真实验教学平台的网页端需要能够在任意主流浏览器使用、显示和运行，网页界面简洁,数据各模块数据展示应简洁清晰,实现良好的数据可视化。

3.2.3 时间特性

软件工程经济学虚拟仿真实验教学平台需要保证作为主要业务逻辑的实验操作功能的响应时间控制在 1 秒以内，最长不超过 3 秒。其他次要的业务逻辑也能保证响应时间最长不超过 2 秒。即使在一定数量的用户同时请求的情况下，系统依然能够维持上述时间指标。

3.2.4 可维护性

软件工程经济学虚拟仿真实验教学平台需要保证系统的各项运行指标能被实时监控和记录，使运维人员能及时预防错误的发生，同时在系统出错时能有迹可循。同时，系统在开发的过程中需要满足行业中优秀的实践原则，例如书写代码需要提供必要的注释以保证可读性，降低模块之间的依赖以满足低耦合特性等等。

3.2.5 安全性

软件工程经济学虚拟仿真实验教学平台需要保证学生和助教在登录或注册的过程中使用高安全性的加密协议进行数据传输，防止报文被截获破解。同时需要保证数据库中的密码不保存明文而是加密后的密文，使得用户密码不易被轻易获得，以期在数据意外泄露时也同样能保障用户

的账号安全。

该系统应防止学生和助教改变权限以外的任何有效信息，严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作，防止数据篡改或信息泄露，从而确保数据的机密性和完整性。

此外，出于安全原因，只有助教可以更改部分账号信息（譬如增删改查部分信息缺漏的学生个人信息），但不包括学生已完成的实验的基本信息。

3.3 需求验证

3.3.1 一致性

对于用自然语言书写的需求分析，除了靠人工技术审查验证软件系统规格说明书的正确性之外，目前还没有其他更好的“测试方法”。

组内成员分别扮演不容的利益相关者包括老师、助教和学生等等，分别从不同角度检查自身需求是否得到满足，自身需求是否与其他利益相关者的需求存在冲突。在进行多次评审和调整之后，得到了最终的满足一致性的需求规约。

3.3.2 完整性

对已有的软件工程管理与经济虚拟仿真实验平台以及软件规模度量模型进行了梳理，并根据不同成员扮演的不同角色所提需求，整理需求规约，确保需求规约中的需求能够满足各方面用户的实际需求。

3.3.3 现实性

针对现有各软件工程管理与经济虚拟仿真实验平台的实现进行分析，了解一般虚拟仿真实验平台的设计实现方案，以及相关实验模块的实现流程，从底层逻辑了解其原理和运作流程，从而验证本系统的现实性。

3.3.4 有效性

为了验证系统的有效性，即能够确实解决用户面对的问题，查阅相关资料并查看与本系统设计相似的其他虚拟仿真实验平台的运行情况，同时了解软件规模度量组件的使用情况。

根据多个虚拟仿真实验平台的投入运行情况及用户的对软件规模度量方法使用情况，来验证本系统的有效性。

3.4 本章小结

本章详细分析了软件工程经济学虚拟仿真实验教学平台之软件规模度量组件的功能性需求和非功能性需求，对于功能性需求，研究总结出项目中存在的用例，并将其制作成总体用例图，对于每一个重要用例都进行详细的分析，绘制了时序图；对于非功能性需求，我用软件工程语言表达、总结出来，为之后的设计实现阶段奠定良好的基础。

4 设计解决方案

软件开发早期阶段，首要工作就是对软件功能点估算，再到对工作量估算、成本估算，这将为项目的计划、决策及资源分配提供必要的支持。功能点估算方法是对程序规模的综合度量，可以用于项目早期阶段的规模估算。从需求说明计算出功能点比计算出代码行更容易，而且可以使用功能点作为基础，计算出以代码行表示的规模度量值。因此，软件工程经济学虚拟仿真实验平台拟用 IFPUG 功能点度量方法和 NESMA 功能点度量方法为基础设计实验，让学生深入了解通过功能点法度量软件规模的流程。

4.1 IFPUG 软件规模度量方法

4.1.1 概述

IFPUG 法是将软件系统按组件进行分类，并对每类组件以 IFPUG 定义的功能点作为度量单位进行计算，得到反应整个软件系统规模的功能点数量。

4.1.2 IFPUG 法工作流程

IFPUG 法工作流程由 5 个步骤组成：第一步确定系统边界及范围，第二步计算数据功能点及其复杂度，第三步计算事务功能点及其复杂度，第四步计算初始功能点，第五步计算软件规模，具体如下图所示：

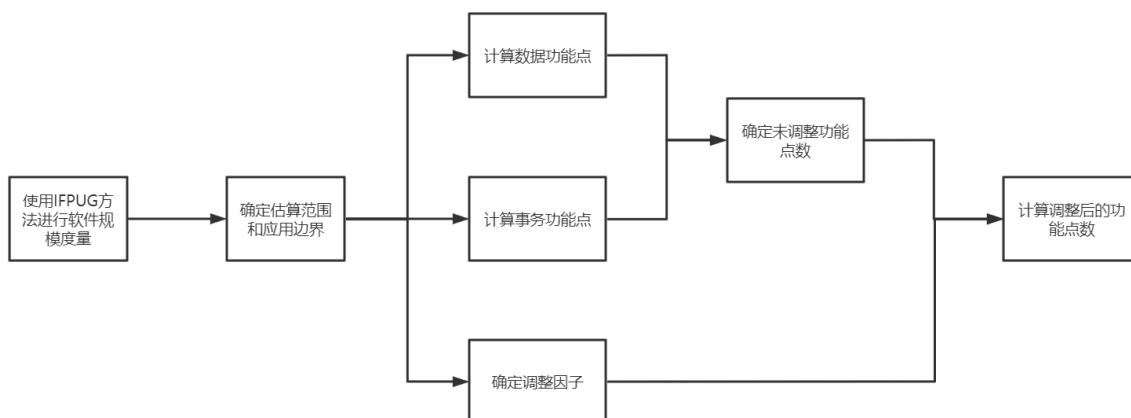


图 4.1 IFPUG 法工作流程

4.2.3 用户功能类型

软件项目管理中的功能点估计法将功能点分为 5 类，具体如下表所示：

表 4.1 功能点分类

| 名称 | 描述 |
|---|--|
| 内部逻辑文件 ILF: Internal Logical File | 内部逻辑文件 ILF 指的是在应用程序的边界内所维护的、用户可以识别的逻辑相关数据组或者控制信息。ILF 的主要目的是保存由被计数的应用程序的一个或多个基本处理所维护的数据。 |
| 外部接口文件 EIF: External Interface File | 外部接口文件 EIF 指的是被度量应用所引用，同时在另一个应用边界内被维护的，用户可以识别的、逻辑相关的数据组或控制信息。EIF 的主要目的是保存由被计数的应用程序边界内的一个或多个基本引用的数据，即一个应用的 EIF 必定是另一个应用的 ILF。 |
| 外部查询 EQ: External Inquiry | 外部查询 EQ 指的是将数据抑或控制信息发送至应用程序边界外部的一个基本处理过程。EQ 的主要目的是通过提取应用的内部逻辑文件或外部接口文件的数据或控制信息，向用户呈现信息。其中，处理逻辑既不包含各类数学公式以及计算，也不创建新衍生的数据。处理期间既不会维护内部逻辑文件 ILF，也不会改变系统应用的行为。 |
| 外部输出 EO: External Output | 外部输出 EO 指的是将数据抑或控制信息发送至应用程序边界外部的一个基本处理过程，但与 EQ 相对比，EO 同时包含额外的处理过程。EO 的主要目的是通过检索应用的内部逻辑文件或外部接口文件的数据或控制信息，并对其进行逻辑处理，向用户呈现信息。其中，处理逻辑必须包含大于等于一个数学公式或是计算过程，或者能够创建新衍生的数据。一个外部输出 EO 也可以维护一个或多个内部逻辑文件 ILF，同时也有可能改变系统应用的行为。 |
| 外部输入 EI: External Input | 外部输入 EI 指的是数据抑或控制信息由应用程序边界外部向应用程序边界内传输的一个基本处理过程。EI 的主要目的是维护一个或多个内部逻辑文件 ILF 或是改变系统行为。增、删、改是都是典型的 EI。 |

其中：（1）数据功能性需求指标：ILF、EIF；（2）事务功能性需求指标：EI、EO、EQ。

4.1.4 确定系统边界及范围

功能点计算中，边界的划分和确定能够正确度量软件系统规模的基石，通常认为估算范围是指需要度量的软件，程序的边界是指拟测量程序与其他外部程序或用户之间的边界。边界的划分对估算的结果影响很大，理论上划分的越细，功能规模就越大。边界的划分一定要从用户角度出发，不能从开发视角进行划分。要确定划分后的软件或子系统一般能独立运行。一般来说，规模庞大的软件系统，可以按子系统划分边界。

4.1.5 计算数据功能点及其复杂度

数据功能是指供更新、引用和检索而储存的可用的逻辑数据，分为内部逻辑文件和外部接口文件。每个内部逻辑文件和外部接口文件都必须根据相关的数据元素类型（DET）和记录元素类型（RET），被赋予一个功能复杂度。

其中，数据元素类型（Data Element Types）是指在一个 ILF 或 EIF 内，用户可认知的、唯一的、非重复的字段。记录元素类型（Record Element Types）是指在一个 ILF 或 EIF 内，用户可认知的数据元素子集。

复杂度分为高、中、低三种等级，具体情况见下表：

表 4.2 ILF 和 EIF 复杂度表

| 记录元素类型（REF） | 数据元素类型（DET） | | |
|-------------|-------------|-------|-----|
| | 1-19 | 20-50 | >50 |
| 1 | 低 | 低 | 中 |
| 2-5 | 低 | 中 | 高 |
| >5 | 中 | 高 | 高 |

4.1.6 计算事务功能点及其复杂度

事务功能主要是指外部输入、外部输出、外部查询、完成更新、检索和输出等操作。每种功能都有其自身的基于特定复杂度的功能点权重值。外部输入（EI）、外部输出（EO）和外部查询（EQ）的复杂度系数见下表：

表 4.3 EI 复杂度表

| 引用元素类型（FTR） | 数据元素类型（DET） | | |
|-------------|-------------|------|-----|
| | 1-4 | 5-15 | >15 |
| 0-1 | 低 | 低 | 中 |
| 2 | 低 | 中 | 高 |
| >2 | 中 | 高 | 高 |

表 4.4 EO 和 EQ 的复杂度表

| 引用元素类型（FTR） | 数据元素类型（DET） | | |
|-------------|-------------|------|-----|
| | 1-5 | 6-19 | >19 |
| 0-1 | 低 | 低 | 中 |
| 2-3 | 低 | 中 | 高 |
| >3 | 中 | 高 | 高 |

4.1.7 计算初始功能点

依据数据功能及其复杂度、和事务功能及其复杂度，将每项功能的复杂度级别乘以相应的复杂度权指数，再将各类功能相加，得到系统的未调整功能点数值 UFP: Unadjusted functional point count。初始功能点反映了软件功能需求的规模，并未考虑软件非需求对软件规模的影响。

组件复杂度与功能点对应关系如下表：

表 4.5 组件复杂度等级与功能点数的对应关系表

| 复杂度级别 | 低 | 中 | 高 |
|--------|----|----|----|
| 功能类型 | 权重 | 权重 | 权重 |
| 外部接口文件 | 5 | 7 | 10 |
| 内部逻辑文件 | 7 | 10 | 15 |
| 外部输入 | 3 | 4 | 6 |
| 外部输出 | 4 | 5 | 7 |
| 外部查询 | 3 | 4 | 6 |

由上表确定根据不同的复杂度而定的 5 个部分的加权因子 W，以及应用中每个部分的数量 X，计算得出最终的未调整功能点数值。具体计算公式如下：

$$UFP = \sum \sum W \times X \quad (4.1)$$

4.1.8 计算软件规模

在 IFPUG 引入 14 个通用系统特征调整参数，其比重占未调整功能点的±35%，来对初始功能点进行调整。调整后得到系统功能点数，计算公式如下：

$$TDI = \sum DI \quad (4.2)$$

$$VAF = TDI \times 0.01 + 0.65 \quad (4.3)$$

由（4.1）（4.2）（4.3）式得到功能点数：

$$FP = UFP \times VAF \quad (4.4)$$

其中，DI 为影响程度：Degree of Influence，

TDI 为系统特征影响度之和：Total Degree of Influence，

VAF 为调整因子：Value Adjustment Factor，

FP 为功能点数：Functional Point。

这 14 项通用系统特征调整参数分别是：

表 4.6 通用系统特征 GSC

| 通用系统特征 GSC | 影响程度 DI | | | | | |
|-------------------------------------|---------|---|---|---|---|---|
| 数据通信(Data Communication) | 0 | 1 | 2 | 3 | 4 | 5 |
| 分布数据处理(Distributed Data Processing) | 0 | 1 | 2 | 3 | 4 | 5 |
| 性能(Performance) | 0 | 1 | 2 | 3 | 4 | 5 |
| 重度使用配置(Heavily used Configuration) | 0 | 1 | 2 | 3 | 4 | 5 |
| 事务率(Transaction Rate) | 0 | 1 | 2 | 3 | 4 | 5 |
| 在线数据输入(Online Data Entry) | 0 | 1 | 2 | 3 | 4 | 5 |
| 终端用户效率(End-User Efficiency) | 0 | 1 | 2 | 3 | 4 | 5 |
| 在线更新(Online Update) | 0 | 1 | 2 | 3 | 4 | 5 |
| 复杂处理(Complex Processing) | 0 | 1 | 2 | 3 | 4 | 5 |
| 可重用性(Reusability) | 0 | 1 | 2 | 3 | 4 | 5 |
| 安装简易性(Installation Ease) | 0 | 1 | 2 | 3 | 4 | 5 |
| 易操作性(Operational Ease) | 0 | 1 | 2 | 3 | 4 | 5 |
| 多站点(Multiple Sites) | 0 | 1 | 2 | 3 | 4 | 5 |
| 便于更改(Facilitate Change) | 0 | 1 | 2 | 3 | 4 | 5 |

4.1.9 实验设计

虚拟仿真实验教学平台中的软件规模度量实验拟设计一个模拟的软件开发流程，让学生使用上述 IFPUG 方法流程进行软件规模度量，体会度量的过程。实验给出模拟项目的具体信息和软件规模度量相关的补充条件，学生据此计算出目标系统的未调整功能点数值、调整因子，以及调整后功能点数值。

4.2 NESAM 软件规模度量方法

4.2.1 概述

IFPUG 方法在使用时对需求完整程度有一定的要求、对业务逻辑也有要求比较精准的要求，无法完全满足软件项目早期的管理要求。由此，荷兰软件度量协会通过对功能点分析方法的不断深入研究，对 IFPUG 方法进行发展和细化，推出了 NESMA 功能点分析方法。

4.2.2 NESMA 方法工作流程

NESMA 功能点分析法与 IFPUG 在度量步骤上有一定的相似之处，具体流程如下图所示：

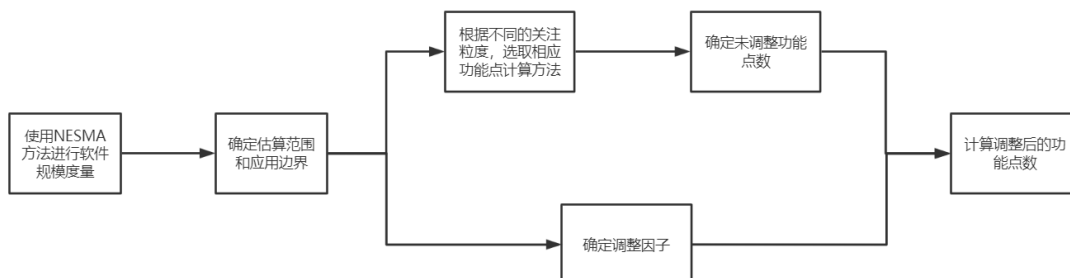


图 4.2 NESMA 法工作流程

4.2.3 估算方法

NESMA 功能点标准提供了三种类型的功能点估算方法，分别为指示功能点分析法、估算功能点分析法，以及详细功能点分析法。一般情况下，使用前需判断需求文档的详细程度，来明确要采用的功能点估算方法。

不同估算方法的关注粒度不同，具体如下表所示：

表 4.7 不同 NESMA 功能点估算方法关注粒度表

| 功能点分析方法 | 关注粒度 | | |
|---------|------|------|-----|
| | 逻辑文件 | 事务功能 | 复杂度 |
| 指示功能点方法 | √ | | |
| 估算功能点方法 | √ | √ | |
| 详细功能点方法 | √ | √ | √ |

（1）指示功能点方法：

指示功能点方法也被称为预估功能点方法，是指在估算软件功能规模时，只估算数据功能（ILF、EIF），不估算事务功能（EI、EO、EQ）。

具体计算公式如下：

$$UFP = 35 \times ILF + 15 \times EIF \quad (4.5)$$

（2）估算功能点方法

估算功能点方法也称高级功能点方法，是指确定每个功能类型（ILF、EIF、EI、EO、EQ），使用固定的复杂难度计算原始功能点数。

按照 2018 年发布的《NESMA 功能规模度量方法》(ISO/IEC24570)，数据功能采用低复杂度，事务功能采用中等复杂度。

具体计算公式如下：

$$UFP = 7 \times ILF + 5 \times EIF + 4 \times EI + 5 \times EO + 4 \times EQ \quad (4.6)$$

（3）详细功能点方法

在确定每个功能类型（ILF、EIF、EI、EO、EQ）的基础上，依据其复杂度，确定每个功能类型的取值。引入数据元素类型 DET、记录元素类型 RET 和引用文件类型 FTR 三个参数确定复杂度，根据上述三个指标的复杂度确定每个功能类型的具体取值，分别计算后求和。

具体计算公式如下：

$$UFP = \sum ILF + \sum EIF + \sum EI + \sum EO + \sum EQ \quad (4.7)$$

对比三种估算方法的应用场景：

表 4.8 代码行和功能点方法优缺点对比列表

| NESMA 估算方法 | 简要分析 | 具体应用场景 |
|---------------|---|--|
| 指示功能点方法 | 指示功能点分析方法主要用于项目早期。此时软件项目需求精度不够，无法确定具体的事务功能，指示功能点分析方法在该情况下得到应用。只需通过需求将系统所涉及到的数据功能识别出来即可。 | 在项目预算或招投标阶段：采用预估功能点方法能够快速判断项目所需完成的规模、工作量、工期和成本，从而决定组织是否能够支撑或接受该项目。 |
| 估算功能点方法 | 估算功能点分析方法在进行软件规模度量时需识别出所有事务功能和数据功能，并采用各功能类型的标准值来确定应用系统的未调整的功能规模。 | 在合同谈判阶段：甲方可以采用估算功能点方法，对乙方所提供的软件功能点数量进行估算并支付合同进度款项。 |
| 详细功能点方法 | 采用详细功能点计算法，软件系统规模按照其复杂度评估的结果以及复杂度取值表进行计算。 | 在项目验收阶段：甲方可以采用详细功能点方法，对乙方所提供的软件的最终功能点数量进行验收并支付合同款项。 |

4.2.4 实验设计

本实验中关于使用 NESMA 方法对软件规模进行度量，由于实验预先给出了模拟项目的相关信息，因此学生可以选择使用恰当的功能点估算方法进行软件规模度量，得出未调整功能点数、调整因子以及调整后的功能点数。

4.3 工作量估算

4.3.1 工作量估算方法

工作量估算方法有方程法、类比法和类推法。

- (1) 方程法：基于基准数据建立参数模型，输入各项参数，估算规模数值。
- (2) 类比法：比对待估算项目的部分属性与类似的一组基准数据，估算规模数值。
- (3) 类推法：将待估算项目的部分属性与高度类似的一个或几个已完成项目的数据进行对比，进行适当调整后确定规模数值。

4.3.2 实验设计

本实验拟采用方程法，以得到一个较为具体的估算结果设计实验。

- (1) 实验将假设，基于基准数据建立的回归方程如下式所示：

$$UE = C \times S^a \quad (4.8)$$

上式中：UE 未调整工作量，单位是人时(p · h) ；

C 生产率调整因子,单位是人时/每功能点(p · h/FP) ；

S 软件规模，单位是功能点(FP)；

a 基于基准数据计算得出的软件规模调整系数。

- (2) 假设根据相关性分析和经验确定，调整后工作量公式如下式所示：

$$AE = UE \times A \times IL \times L \times T \quad (4.9)$$

上式中：AE 调整后工作量，单位是人时(p · h) ；

A 应用领域调整因子，取值范围 0.8~1.2；

IL 软件完整性级别，取值范围 1.0~1.8；

L 开发语言调整因子，取值范围 0.8~1.2；

T 最大团队规模因子，取值范围 0.8~1.2。

由此可以通过公式计算出实验模拟系统的一个确切的工作量范围。

4.4 软件开发成本估算

软件开发成本计算可以按照软件成本组成元素，分别计算直接人力成本 DHC、直接非人力成本 DNC、间接人力成本 IHC 和间接非人力成本 INC，然后求和计算软件开发成本 SDC。

也可以依据工作量估算结果和平均人力成本费率，直接计算出直接人力成本和间接成本的总和，再加上直接非人力成本，得到软件开发成本。

同时还可以采用第三种软件开发成本估算方法，即，依据规模估算结果 S(功能点)和规模综合单价 P(元/每功能点)，直接计算出直接人力成本和间接成本的总和，再加上直接非人力成本 DNC，得到软件开发成本。

计算公式如下：

$$SDC = P \times S + DNC \quad (4.10)$$

其中，规模综合单价，依据软件行业协会 2017 年发布，它的分布类似正态分布。在估算时，通常 P50 是最有可能值，P25 和 P75 是估算上下限，在特殊情况（项目成本、工期、质量约束极其严格）下，也可采用 P10 和 P90 的值估算上下限。

具体数值如下表所示：

表 4.9 功能点耗时率

| | P10 | P25 | P50 | P75 | P90 |
|----|------|------|------|-------|-------|
| 人时 | 1.79 | 3.58 | 7.16 | 12.99 | 17.81 |

第三种软件开发成本估算方法也是本实验拟采用的方法。

4.5 界面规约

4.5.1 学生相关界面

登录界面：该界面用于用户输入用户名和密码匹配登录。

虚仿实验教学平台实验主界面：该界面用于展示全部的软件工程经济学虚拟仿真实验，并完成有关学生和实验信息的交互，响应用户的请求，并调用对应的模块。

软件规模度量实验主界面：该界面可用于查看软件规模度量实验的各项信息，以及包含进入实验接口。

软件规模度量实验详情界面：该界面用于展示当前虚仿实验的具体信息，包括展示实验目的、实验原理、实验内容和具体操作等等，还可提供展示相关软件规模度量方法的拓展知识。

软件规模度量实验操作界面：

- （1）模拟项目的项目简介；
- （2）阶段一：功能点、规模；
- （3）阶段二：人时、工作量；
- （4）阶段三：报价、不同岗位的劳动生产率价格，确定软件研发成本并提交实验结果；
- （5）实验结果展示界面

个人中心界面：该界面用于展示用户个人信息，包括对本人正在参与实验的进展情况、查看已参加的虚仿实验成绩等。

4.5.2 助教相关界面

实验信息管理界面：该界面用于助教管理学生的信息，包括增删改查以及封禁有违规操作的用户等等。同时用于助教发布实验、发布实验题库、结束实验、查看学生成绩。

4.7 数据库设计

4.7.1 student 表

| 字段名 | 数据类型 | 字段说明 |
|-------------|-------------|---------------|
| studentId | int(7) | 学号 |
| studentName | varchar(20) | 姓名 |
| grade | varchar(4) | 年级 |
| major | varchar(20) | 专业 |
| clazz | varchar(10) | 班级 |
| institute | varchar(30) | 学院 |
| tel | varchar(11) | 电话号码 |
| email | varchar(30) | 电子邮件 |
| pwd | varchar(20) | 密码 |
| sex | varchar(2) | 性别 |
| role | varchar(1) | 角色（1 助教，2 学生） |

4.7.2 teacher 表

| 字段名 | 数据类型 | 字段说明 |
|-------------|-------------|---------------|
| teacherId | int(7) | 助教学号 |
| teacherName | varchar(20) | 姓名 |
| institute | varchar(30) | 学院 |
| tel | varchar(11) | 电话号码 |
| email | varchar(30) | 电子邮件 |
| pwd | varchar(20) | 密码 |
| sex | varchar(2) | 性别 |
| role | varchar(1) | 角色（1 助教，2 学生） |

4.7.3 score 表

| 字段名 | 数据类型 | 字段说明 |
|----------------|-------------|------|
| scoreId | int(9) | 成绩编号 |
| experimentCode | int(9) | 实验编号 |
| studentId | int(7) | 学号 |
| subject | varchar(20) | 实验名称 |
| score | int(5) | 实验成绩 |
| answerdate | varchar(10) | 答题日期 |

4.7.4 experiment 表

| 字段名 | 数据类型 | 字段说明 |
|----------------|--------------|--------|
| experimentCode | int(9) | 实验编号 |
| description | varchar(50) | 实验介绍 |
| source | varchar(20) | 实验名称 |
| examId | int(9) | 个人实验编号 |
| examDate | varchar(10) | 实验日期 |
| totalTime | int(3) | 实验时长 |
| grade | varchar(10) | 年级 |
| term | varchar(10) | 学期 |
| major | varchar(20) | 专业 |
| institute | varchar(20) | 学院 |
| totalScore | int(4) | 实验得分 |
| type | varchar(255) | 实验类型 |
| tips | varchar(255) | 实验须知 |

4.7.5 question 表

| 字段名 | 数据类型 | 字段说明 |
|------------|--------------|--------|
| questionId | int(9) | 实验问题编号 |
| subject | varchar(20) | 实验名称 |
| question | varchar(255) | 问题内容 |
| answer | varchar(255) | 问题答案 |
| analysis | varchar(255) | 问题分析 |
| score | int(2) | 问题分数 |
| section | varchar(20) | 问题章节 |

4.7.6 experiment_manage 表

| 字段名 | 数据类型 | 字段说明 |
|------------|--------|--------|
| examId | int(9) | 个人实验编号 |
| questionId | int(9) | 实验问题编号 |

4.8 本章小结

本章详细设计了软件工程经济学虚拟仿真实验平台使用 IFPUG 功能点度量方法和 NESMA 功能点度量方法为基础进行实验的基本流程，同时，也进行了系统设计界面设计，以及数据库设计，为系统的实现打下了基础。

5 系统的实现

从根据实际项目需求, 以及不同软件规模度量方法, 作为虚拟仿真实验平台的实验项目之一, 探索给出适用于本项目的软件规模度量组件模型和参数, 并根据组件的概要设计以及详细设计, 完成组件的开发编码工作。

5.1 系统软件结构

5.1.1 系统架构

按照调用和返回体系结构, 得到主程序/子程序系统架构图如下:

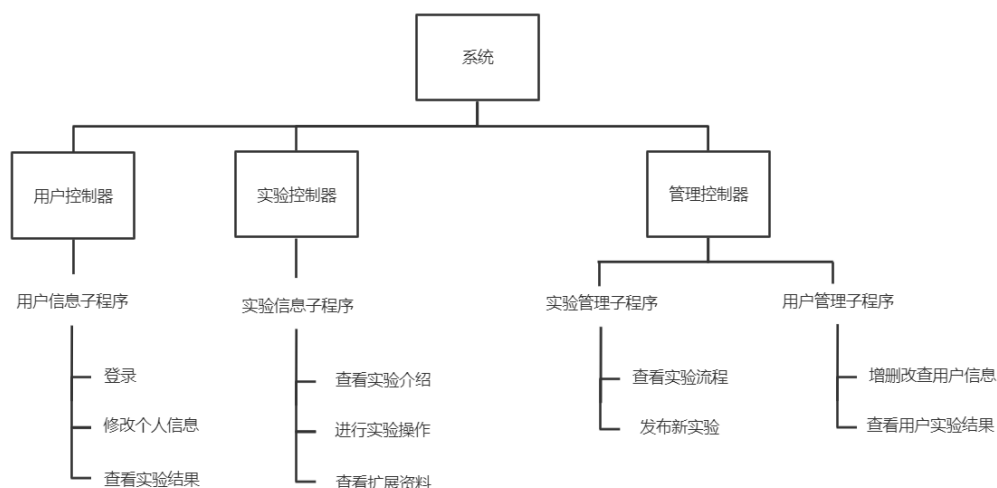


图 5.1 系统架构图

5.1.2 体系结构

系统采用了如下的体系结构:

表现层主要处理对用户的请求接受以及数据的返回, 为客户端提供应用程序的访问。

服务层提供了 API, 让调用者可以使用接口暴露的方法。

业务层把数据层的操作进行整合, 被划分为多个子系统。

数据层是提供访问位于持久化容器中数据的功能, 完成对数据文件的操作。

下图为具体的体系结构:

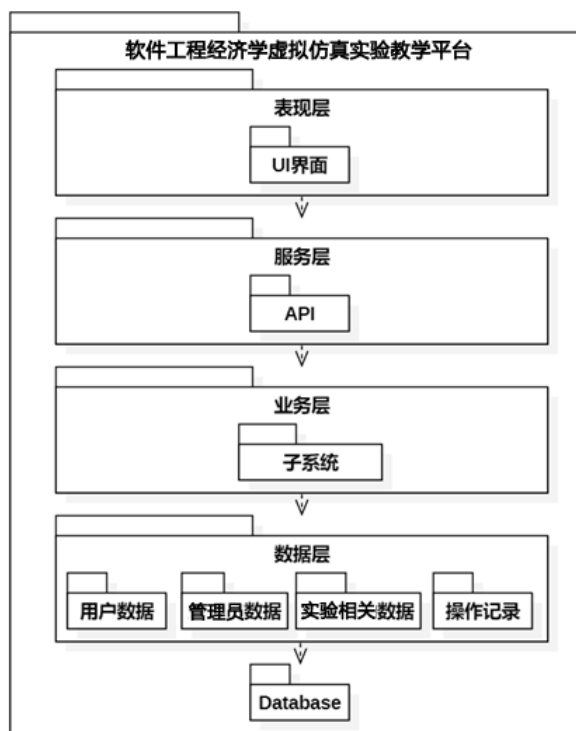


图 5.2 体系结构图

5.2 虚拟仿真实验方案的实现

5.2.1 模拟的项目基本信息介绍

学生将要以实验室仪器设备管理系统为基础，完成对其的软件规模度量。

模拟的实验室仪器设备管理系统描述如下：

实验仪器设备管理系统的主要功能包括管理人员信息、管理仪器设备信息、管理设备借出、归还信息和查询仪器设备信息等。

以下是软件规模度量提前已知的条件：

(1)

表 5.1 组件复杂度等级与功能点数对应关系表

| 复杂度级别 | 低 | | 中 | | 高 | |
|--------|----|----|----|----|----|----|
| | 权重 | 个数 | 权重 | 个数 | 权重 | 个数 |
| 外部接口文件 | 5 | 3 | 7 | 7 | 10 | 1 |
| 内部逻辑文件 | 7 | 2 | 10 | 10 | 15 | 1 |
| 外部输入 | 3 | 3 | 4 | 4 | 6 | 1 |
| 外部输出 | 4 | 2 | 5 | 5 | 7 | 1 |
| 外部查询 | 3 | 2 | 4 | 2 | 6 | 1 |

表 5.2 系统功能点表

| 序号 | 功能点类型 | 基本过程 | 复杂度 |
|----|--------|-----------|-----|
| 1 | 外部输入 | 仪器设备续借 | 低 |
| 2 | 外部输入 | 借用仪器设备 | 一般 |
| 3 | 外部输入 | 添加和修改仪器设备 | 一般 |
| 4 | 外部输入 | 修改规则 | 高 |
| 5 | 外部输出 | 归还仪器设备提醒 | 低 |
| 6 | 外部输出 | 用户信息统计 | 一般 |
| 7 | 外部输出 | 用户借还详细信息 | 高 |
| 8 | 外部查询 | 查看规则信息 | 低 |
| 9 | 外部查询 | 查看用户信息 | 低 |
| 10 | 外部查询 | 查询仪器设备信息 | 一般 |
| 11 | 外部查询 | 查询用户预约信息 | 高 |
| 12 | 外部接口文件 | 管理员信息文件 | 低 |
| 13 | 外部接口文件 | 用户信息文件 | 一般 |
| 14 | 外部接口文件 | 实验室员工信息文件 | 高 |
| 15 | 内部逻辑文件 | 规则信息文件 | 低 |
| 16 | 内部逻辑文件 | 权限信息文件 | 一般 |
| 17 | 内部逻辑文件 | 仪器设备信息文件 | 高 |

表 5.3 影响度表

| 通用系统特征 GSC | 影响程度 DI |
|-------------------------------------|---------|
| 数据通信(Data Communication) | 4 |
| 分布数据处理(Distributed Data Processing) | 4 |
| 性能(Performance) | 4 |
| 重度使用配置(Heavily used Configuration) | 3 |
| 事务率(Transaction Rate) | 4 |
| 在线数据输入(Online Data Entry) | 5 |
| 终端用户效率(End-User Efficiency) | 5 |
| 在线更新(Online Update) | 3 |
| 复杂处理(Complex Processing) | 1 |
| 可重用性(Reusability) | 1 |
| 安装简易性(Installation Ease) | 1 |
| 易操作性(Operational Ease) | 4 |
| 多站点(Multiple Sites) | 3 |
| 便于更改(Facilitate Change) | 3 |

（2）基于基准数据建立的回归方程为： $UE = C \times S^a$

根据相关性分析和经验，调整后工作量公式为： $AE = UE \times A \times IL \times L \times T$

参考基准数据的功能点耗时率 25 百分位数、50 百分位数和 75 百分位数，C（生产率调整因子）取值分别是 $8p \cdot h/FP$ 、 $10p \cdot h/FP$ 、 $14p \cdot h/FP$ ，软件规模调整系数取值 0.9。

同时，根据经验选择各项调整因子如下：A 应用领域调整因子，取值为 1；IL 软件完整性级别，取值为 1；L 开发语言调整因子，取值为 0.8；T 最大团队规模因子，取值为 1.1。

（3）不包括直接非人力成本的人月基准单价，2.88 万元/人月。非直接人力成本，假定为 3 万元。按每月 176 小时工作时间计算（22 天×8 小时）

5.2.2 应得出的软件规模度量结果

（1）IFPUG 方法：

系统未调整的功能点数： $UFC=32+59+23+25+20=159$

调整因子 $VAF = TDI \times 0.01 + 0.65 = 1.1$

功能点数 $FP = UFP \times VAF = 174.9$

未调整工作量 $UE \in [4.74, 8.3]$

未调整工作量最有可能值=5.93

计算人月最可能数： $AE=S \times P50/176$

计算人月上限： $AE=S \times P75/176$

计算人月下限： $AE=S \times P25/176$

调整后工作量 $AE \in [4.17, 7.3]$

调整后工作量最有可能值=5.22

开发成本预算下限=15.01

开发成本预算上限=24.02

开发成本预算中值=18.03

（2）NESMA 方法：

指示功能点法 系统未调整的功能点数 $UFP=35 \times ILF + 15 \times EIF=385$

估算功能点法 系统未调整的功能点数 $UFC=7 \times ILF + 5 \times EIF + 4 \times EI + 5 \times EO + 4 \times EQ=234$

详细功能点法 系统未调整的功能点数 UFC 与 IFPUG 方法相同

根据详细功能点法：

未调整工作量 $UE \in [4.74, 8.3]$

未调整工作量最有可能值=5.93

调整后工作量 $AE \in [4.17, 7.3]$

调整后工作量最有可能值=5.22

开发成本预算下限=15.01

开发成本预算上限=24.02

开发成本预算中值=18.03

5.3 代码实现

5.3.1 用户登录

学生和助教均通过输入用户名和密码进行登录。学生登录需要助教提前将学生信息上传至数据库，学生首次登录系统通过助教设定的初始密码进行登录。用户名（学号）和密码匹配则成功登录，若用户名密码不匹配/账号或密码为空等情况则会提示错误信息。

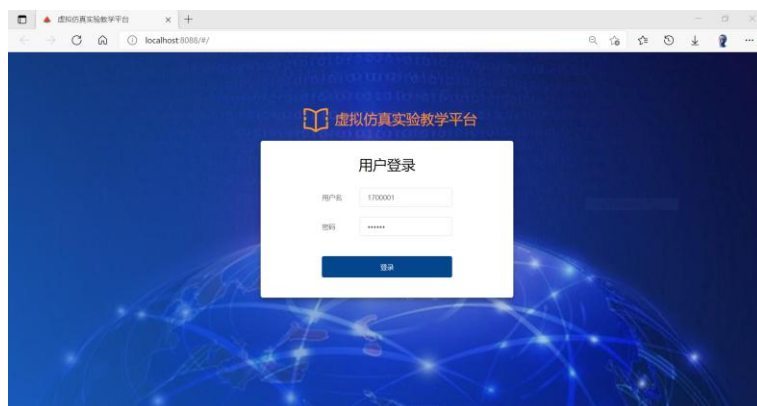


图 5.3 登录界面截图

5.3.2 实验流程

学生登录后进入主界面，主界面展示虚仿平台现有的实验模块，学生选择相应实验。



图 5.4 虚仿平台主界面截图

进入实验界面，可以学习相关的实验基本信息。



图 5.5 实验介绍主界面截图

点击开始实验。



图 5.6 实验进行界面截图

实验完成，显示结果。

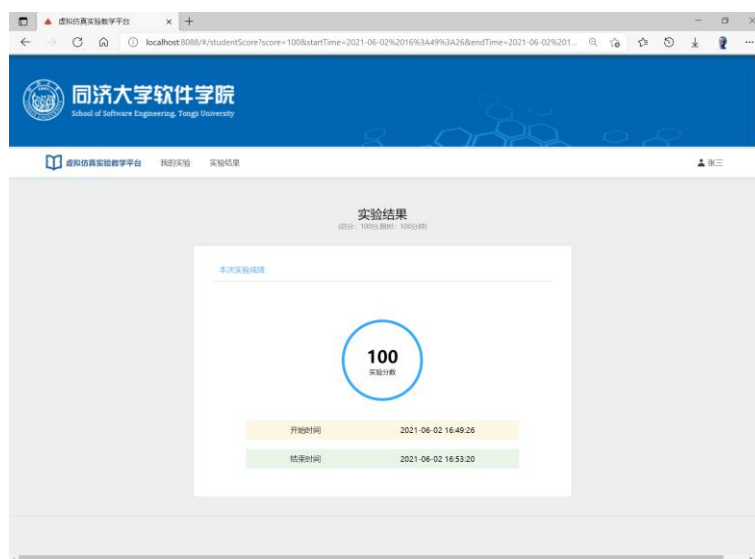


图 5.7 实验结果界面截图

查看实验结果：

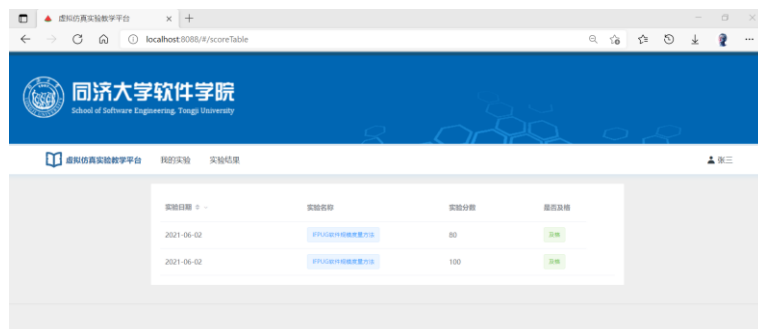


图 5.8 实验成绩界面截图

5.3.3 实验/用户管理

助教登录成功后可以管理现有实验、发布新实验、管理实验题库，也可以查看学生实验的情况、管理学生的信息。

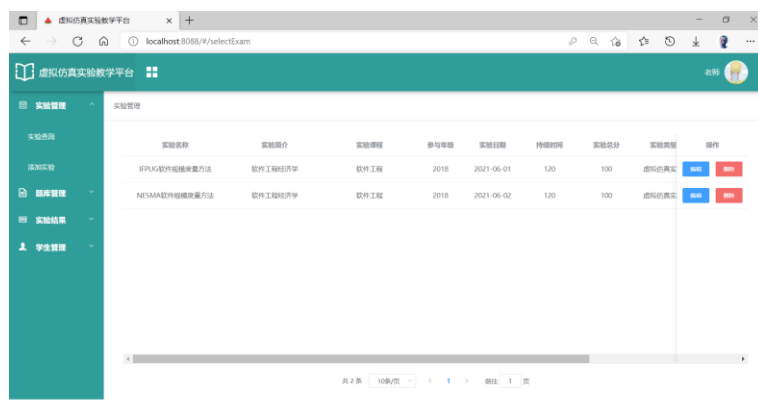


图 5.9 系统后台界面截图

5.4 本章小结

本章以一个具体的实例，实现了软件规模度量组件的模型和详细的实验设计，以及相应的开发编码工作，并展示了系统运行的相应界面。

6 系统的测试

在软件投入运行前，对系统的功能和性能进行测试。通过系统测试寻找程序错误，同时与用户需求相比较，发现系统功能与用户需求之间不相符或有矛盾、歧义的地方，进而对系统进行纠错，不断完善，以保证软件的质量、准确性和可靠性，最终满足设计要求。

6.1 测试环境

操作系统：Windows 10

浏览器：Chrome

测试工具：Postman

6.2 功能测试

| 编号 | 用例 | 前置条件 | 输入 | 预期结果 | 实际结果 |
|----|----------|-----------------|--|----------|-------|
| 1 | 登录 | 数据库中存在用户信息 | studentID={StudentID} pwd={pwd} | 请求成功 | 与期望一致 |
| 2 | 获取实验列表 | 学生登录成功 | examcode={examcode} | 返回实验参数 | 与期望一致 |
| 3 | 实验答题 | 学生登录成功 | answer={answer} | 返回答题参数 | 与期望一致 |
| 4 | 查看实验结果 | 学生登录成功，且用户已完成实验 | studentID={StudentID} | 返回实验结果 | 与期望一致 |
| 5 | 修改密码 | 学生登录成功 | studentID={StudentID} pwd={pwd} | 修改成功 | 与期望一致 |
| 6 | 查询实验 | 助教登录成功 | examId={ examId } | 返回实验参数 | 与期望一致 |
| 7 | 增加实验 | 助教登录成功 | experimentCode={experimentCode} description={description} source={source} examId={examId} examDate={examDate} totalTime={totalTime} grade={grade} term={term} major={major} totalScore={totalScore} type={type} tips={tips} | 添加成功 | 与期望一致 |
| 8 | 增加实验题库 | 助教登录成功 | questionId={questionId} subject={subject} question={question} answer={answer} analysis={analysis} score={score} section={section} | 添加成功 | 与期望一致 |
| 9 | 查看学生实验情况 | 助教登录成功 | studentID={StudentID} score={score} | 返回学生实验结果 | 与期望一致 |
| 10 | 添加学生 | 助教登录成功 | studentId={studentId} studentName={studentName} | 添加成功 | 与期望一致 |

| | | | | | |
|--|--|--|--|--|--|
| | | | grade={grade} major={major} clazz={clazz} institute={institute} tel={tel} email={email} pwd={pwd} sex={sex} role={2} | | |
|--|--|--|--|--|--|

6.3 性能测试

性能是系统运行的一个关键指标，如果软件性能达不到用户的需求，可能会导致系统崩溃。系统性能主要从响应速度和负载能力进行测试。响应速度就是用户向系统发出请求，系统响应请求所耗费的时间。负载能力就是系统所能承受的并发用户数，即同时登陆系统进行操作的用户数量。

（1）响应速度测试

实验操作功能能够在 1 秒左右完成，其他功能基本都在 0.5 秒以内响应。

（2）负载能力测试

经过测试，并发用户分别为 10、20、30、40、50，系统功能的响应速度变化不大，大部分功能都能在 1 秒以内完成。

6.4 可用性测试

可用性测试是用户对产品进行典型性操作，验证系统可用性的测试，有助于提高交互的友好程度，本系统从可行性、有效性、安全性、通用性、易学性、易记性六个维度的可用性目标出发，对项目进行了可用性评估，最终得出本项目具有较好的可用性。

（1）可行性：用户可以完成系统正常的功能。

（2）有效性：实验操作功能及其他相关功能均能够在 1 秒内完成。

（3）通用性：搜索实验时，用户被提示搜索出的实验数目。

（4）安全性：首次登录后用户会被提示修改登录密码。

（5）易学性：用户会被提示完成登录等必要操作；用户可以按照提示输入搜索关键词；用户通过图标了解系统提供的服务以及实验流程。

（6）易记性：个人中心导航操作方便，便于记忆。

6.5 本章小结

本章对软件工程经济学虚拟仿真实验平台之软件规模度量组件进行了功能测试、性能测试以及可用性测试，以保证软件的质量、准确性和可靠性，最终满足设计要求。

7 总结与展望

7.1 总结

本文以软件工程经济学虚拟实验教学平台的开发为背景，设计并实现了其中的一个虚拟仿真实验模块——软件规模度量。本文完成的具体工作如下：

（1）完成软件规模度量模型的相关理论研究。基于现有国内外软件规模度量方法现状和优缺点进行对比分析，同时系统且深入的学习了软件规模度量的相关知识，较为全面地分析了软件规模度量之于软件开发的重要意义。

（2）完成软件规模度量常用方法的归纳整理，以及相关使用中的优缺点对比分析，在针对现有软件规模度量现状进行汇总后，提出对于软件工程经济学虚拟实验教学平台之软件规模度量组件合理的实验设计。

（3）完成利用 IFPUG 功能点度量方法及 NESMA 功能点度量方法设计完整实验流程。首先，对 IFPUG 方法和 NESMA 方法的工作流程进行详细分析与整理，对五种用户功能类型以及应用手段进行研究，同时结合作量估算以及软件成本计算方法，归纳出实验模型，使之后使用的学生能通过本模型详细了解软件规模度量流程。

（4）完成软件规模度量组件相应的编码开发以及测试流程，使其符合用户需求和相关规定。

7.2 展望

作为项目管理关键因素之一，软件规模度量近年来逐渐受到学者的广泛关注和研究，但其实就企业应用现状来讲，重视程度和应用方法还存在很大的局限性。现有软件研发体系并没有形成一套完备的项目规模度量流程，仍未达到可以合理高效地使用成熟的软件规模度量技术提高项目成本估算的准确性以更好地辅助决策的目的。其次，目前软件规模度量方法使用还有一定局限性，仍然过度依赖人员主观判断。现有估算大多利用经验值，并没有准确的基准或者规范可参照，专家对于软件规模度量专业知识的掌握程度等都将对估算结果产生较大影响。所以，软件规模度量在未来发展上仍有很大的空间。

不过目前广泛使用的软件规模度量方法不失为一种很好的解决方案，可以有效地进行粗略的成本估算，并且通过目前众多项目的检验。但是随着对于软件规模度量技术的深入研究，如何更好的利用成熟的模型进行估算，更加合理的在项目之初估算和规划项目，提高项目的成功率，降低因估算误差产生的项目延期或失败比例，应该是重点去研究的。

学生在还未进入企业，在校学习时，通过虚拟仿真实验了解、模拟进行软件规模度量，可以为之在企业具体工作时，在结合真实项目后能更快的结合理论与实际熟练掌握利用并发展软件规模度量技术，更高效的利用软件规模度量实验项目管理。

参考文献

- [1] 王森, 李平. 2014 年国家级虚拟仿真实验教学中心分析[J]. 实验室研究与探索, 2016, 35(04):82-86.
- [2] 崔瑾, 成丹, 鲁燕舞, 等. “虚实互补”的农业生物学实验教学体系的构建与实践[J]. 高校实验室科学技术, 2019(1):1-4.
- [3] 胡云龙. 软件规模度量方法介绍[J]. 计算机时代, 2006(07):17-18+21.
- [4] 宁静峰, 童旅杨. 软件项目功能点估算[J]. 长春工业大学学报(自然科学版), 2014, 35(03):309-314.
- [5] 张云帆. 软件规模度量方法比对研究[D]. 同济大学软件学院, 2009.
- [6] 敖金霞, 王媛媛. 基于整合 LOC 和 FPA 的软件规模度量研究[J]. China's Forgn Trade, 2012(6):429-429.
- [7] Saaty T.L., Vargas L.G.. Estimating technological coefficients by the analytic hierarchy process[J]. Pergamon, 1979, 13(6).
- [8] 田军, 张朋柱, 王刊良, 等. 基于德尔菲法的专家意见集成模型研究[J]. 系统工程理论与实践, 2004, 24(1):57-62.
- [9] 勃姆. 软件成本估算 COCOMO II 模型方法[M]. 机械工业出版社, 2005.

谢 辞

时光易逝，韶华难追，总是无言。

在打下这一行字的时候，我正坐在图书馆二楼角落，稍稍抬头，左前方的窗外一枝绿叶郁郁葱葱，随风轻轻晃动，让人平添了几分惆怅。

大学的四年时间过得很快，快到仿佛昨天我才带着憧憬的脚步踏入校门；大学的时间过得也很慢，用老套的话说，那带着数不清的欢笑和泪水的种种时刻还历历在目，每一分每一秒都隽永如昨日。可能是因为从小到大都顺风顺水没有经历过很大的挫折，我没有那些特别坚持自律、自强不息的品格，仅仅靠自己的一点小聪明，还有父母的费心培养、老师的关照支持，侥幸至今。我生怕自己不是美玉良材而不敢细细雕琢，又有几分相信自己是明珠而不想碌碌无为，然后每天用没心没肺的模样掩饰自己循环往复的矛盾纠结。但我自始至终也是一个幸运的人。

向我的导师黄杰致谢。明师之恩，诚为过于天地。没有黄杰老师不厌其烦的指导，我无论如何也不可能顺利地完成毕业设计，完成这篇毕业论文。黄老师一次次谆谆不倦的开会指导，提供我们所需的各种指导和帮助，在不断地督促论文进度的同时也关心我们的就业工作，提出了很多针对性的建议。金玉良言，盛感谢意。

向我的班主任沈莹老师和大学期间所有帮助、教授过我的老师们致谢。工作细致、认真负责、关心生活、指导学习，很多说着简单，但落到实处却很难。同时，日常学业、以及论文完成过程中，也要感谢我的同学们，在我冒昧向他们表达困惑的时候尽心解答我提出的问题，为我提供帮助。同门之谊，弗敢相忘。

向我的父母和家人致谢。哀哀父母，生我劬劳。他们一向是疼宠甚至娇惯我的，包容我的任性，安慰我的焦虑。前不久又收到家里寄来的几箱水果，我忙打电话回去询问，我妈说正好是吃芒果的季节，还有她朋友开的农家乐，摘得有新鲜水果给我寄来吃，离家在外要好好照顾自己，均衡营养。我笑着说上海又不是没有水果店，转身却有些沉默。仿佛隐隐约约体会到朱自清在《背影》里构建出的那一点神韵。这种神韵既属于朱自清的父亲，又好像属于我们每一个人的父母。初高中同学常常在作文里写突然发现了父亲的驼背、母亲的白发，而意识到时光匆匆父母渐老。初觉千篇一律，现却恍然。我不愿称他们是伟大的父母，他们是我平凡的爸爸妈妈，如诗经所云，他们抚我、畜我、长我、育我、顾我、复我、出入腹我，父母之恩，实无以言表。惟愿岁月不催，笑口常开。

向我的朋友们致谢。高山流水，人之相知，贵在知心。我们在嘉定全家二楼狭窄的小桌上一起为项目通宵；我们在宿舍挑灯夜战为课程作业奋斗；我们在课堂上手肘碰手肘认真听讲；我们在课余时间闲聊、约饭、追剧、游玩、交谈，我们互相鼓励、互相倾诉，嬉笑怒骂，种种皆难以忘怀。我不是在最好的时光遇见了你们，而是遇见了你们，我才有了这段最好的时光。愿岁并谢，与长友兮。

太阳就要落山了，但她不吝惜分我一丝余晖。此刻，扑进了一阕蓬松的阳光，洒在我面前一直未曾合上的电脑屏幕上，还有一丛在我指尖跳跃，追逐到了文章的末尾。

最后，最后了。感谢，感谢。再见，再见。