**软件需求规约**

<"Testland"在线评测平台>

软件需求规约

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <20/6/2023> | <1.0> | <撰写软件需求规约> | <张奕涵/胡彤/全雨乐/杨菡雪> |
| <25/6/2023> | <1.1> | <根据修改意见微调> | <张奕涵/胡彤/全雨乐/杨菡雪> |
|  |  |  |  |
|  |  |  |  |

软件需求规约 (简化版)

1. **简介**

**1.1 目的**

本软件需求规格说明（SRS）的目的是定义和规范一个功能强大的在线代码测评平台的开发，提供对于该平台的全面描述，包括其外部行为、非功能性需求、设计约束以及其他因素，为开发团队提供清晰的指导，以确保开发出符合用户期望和预期的在线代码测评平台。

**1.2 定义、首字母缩写词和缩略语**

|  |  |  |
| --- | --- | --- |
| 序号 | 术语或缩写 | 定义 |
| 1 | Testland | 上海交通大学电子信息与电气工程学院软件工程专业《互联网产品设计与开发》课程第二组四位同学开发的代码在线评测平台。 |
| 2 | 用户 | 所有经过注册的平台使用者的统称，以用户名进行识别。 |
| 3 | 系统管理员 | 上传题目及测试用例以供用户查看及作答的角色 |
| 4 | 评测 | 对程序代码进行编译和执行，并与正确输出进行比较，以检查代码正确性的过程。 |
| 5 | 测试用例 | 系统管理员在后台给出的一组或多组输入、操作或条件，用于验证用户代码的功能、性能或其他方面的正确性。对于给定的功能或需求，测试用例包括输入数据、预期输出以及预期用时。测试用例旨在覆盖不同的情况和边界条件，以检测用户代码的正确性。 |
| 6 | 排行榜 | 对于用户总答题情况（答对题数、正确率、速度等）进行排序形成的用户序列，对所有合法用户进行展示。 |

**1.3 参考资料**

暂无

2. **整体说明**

**2.1 产品总体效果**

“Testland”在线代码测评平台旨在帮助用户改进编程技能、评估代码质量并促进学习和知识分享。用户可以在本平台浏览题库，使用平台支持的多种语言，针对选定题目编写代码并提交，得到实时评估与反馈，并通过排行榜获得激励。

**2.2 产品功能**

1. 提供在线代码测评环境：创建一个能够提供在线代码编写、执行和测试的平台，用户可以在该平台上提交自己的代码，并得到即时的评估和反馈。
2. 多语言支持：支持多种编程语言，以满足不同用户的需求。用户可以选择他们熟悉的编程语言进行编写和测试。
3. 实时评估和反馈：在用户提交代码后，快速进行代码编译、执行和评估，并向用户提供实时的反馈和评分。用户可以根据评估结果进行代码改进和调试。
4. 用户管理和权限控制：提供用户管理和权限控制功能，允许用户注册、登录和查看他们的代码项目。管理员可以发布与管理平台上的题目，发布测试用例等。
5. 可视化和统计功能：提供可视化的界面和统计功能，让用户能够查看他们的代码评估结果、进度和历史记录，以便进行分析和改进。

**2.3 用户特征**

目标用户群体一般具有一定的编程基础，以初学型和熟练型为主，对于平台题库质量、数量要求较高，对于多种编程语言支持的要求较高，对于平台评测实时性要求较高。

**2.4 约束**

**预算**

项目经费两千元，用于华为云平台的开发与部署

**人员**

四人团队开发

**设备**

* 操作系统：不同操作系统（如Windows、MacOS、Linux）可能对开发环境和部署环境有不同的要求和限制。需要确保我们的代码测评平台能够在不同操作系统上正常运行和提供一致的用户体验。
* 浏览器兼容性：不同的浏览器（如Chrome、Firefox、Safari、Edge等）在支持HTML、CSS和JavaScript方面可能存在差异。需要确保我们的代码测评平台在主流浏览器上都能正常运行，并提供一致的功能和用户界面。
* 设备分辨率：用户可能使用不同分辨率的设备来访问您的代码测评平台。需要确保平台能够适应不同的屏幕尺寸和分辨率，以提供良好的用户体验。
* 网络连接：代码测评平台可能需要与远程代码执行环境进行交互，因此对网络连接的要求较高。需要确保平台在不稳定或低速网络环境下仍能正常运行，并提供适当的错误处理和提示。
* 计算资源：在进行代码编译、执行和评估时，可能需要消耗大量的计算资源。需要确保能够有效管理和分配计算资源，以避免性能问题或系统崩溃。
* 安全性：在线代码测评平台需要处理用户提交的代码，因此安全性是一个重要考虑因素。需要采取适当的安全措施，防止恶意代码执行、数据泄露或其他安全威胁。

**时间**

开发可用时间有限，需在2023年7月14日之前完成初代软件的发布，并在2023年9月10日前完成软件开发。

**2.5 假设与依赖关系**

**假设**

* 用户具有基本的计算机操作知识和编程基础。
* 用户能够稳定地连接到互联网，并且具有足够的带宽来上传和下载代码。
* 用户的设备（计算机、手机等）具备足够的性能和资源来访问和使用在线代码测评平台。
* 在线代码测评平台所使用的第三方组件和服务（如编程语言解释器、编译器、代码托管平台等）能够稳定运行并提供所需的功能和接口。

**依赖关系**

* 依赖于适当的开发工具和环境，如集成开发环境（IDE）、编译器、调试器等。
* 依赖于合适的操作系统和服务器环境来部署和运行在线代码测评平台。
* 依赖于云服务提供商（如华为云）提供的计算和存储资源来支持平台的运行和扩展。
* 依赖于网络连接和通信基础设施，以确保用户能够访问平台并与之交互。
* 依赖于第三方组件和服务的稳定性和可靠性，如代码版本控制系统、用户认证服务等。
* 依赖于用户提供合法、有效的代码，以进行评估和执行。

**2.6 需求子集**

1. 功能子集：

* 提供在线代码测评环境
* 多语言支持
* 实时评估和反馈
* 用户管理和权限控制
* 可视化和统计功能

1. 性能子集：

* 快速代码编译和执行
* 高并发支持
* 良好的响应时间

1. 安全子集：

* 用户身份验证和访问控制
* 防止恶意代码执行和攻击
* 数据保护和隐私保密

1. 用户界面子集：

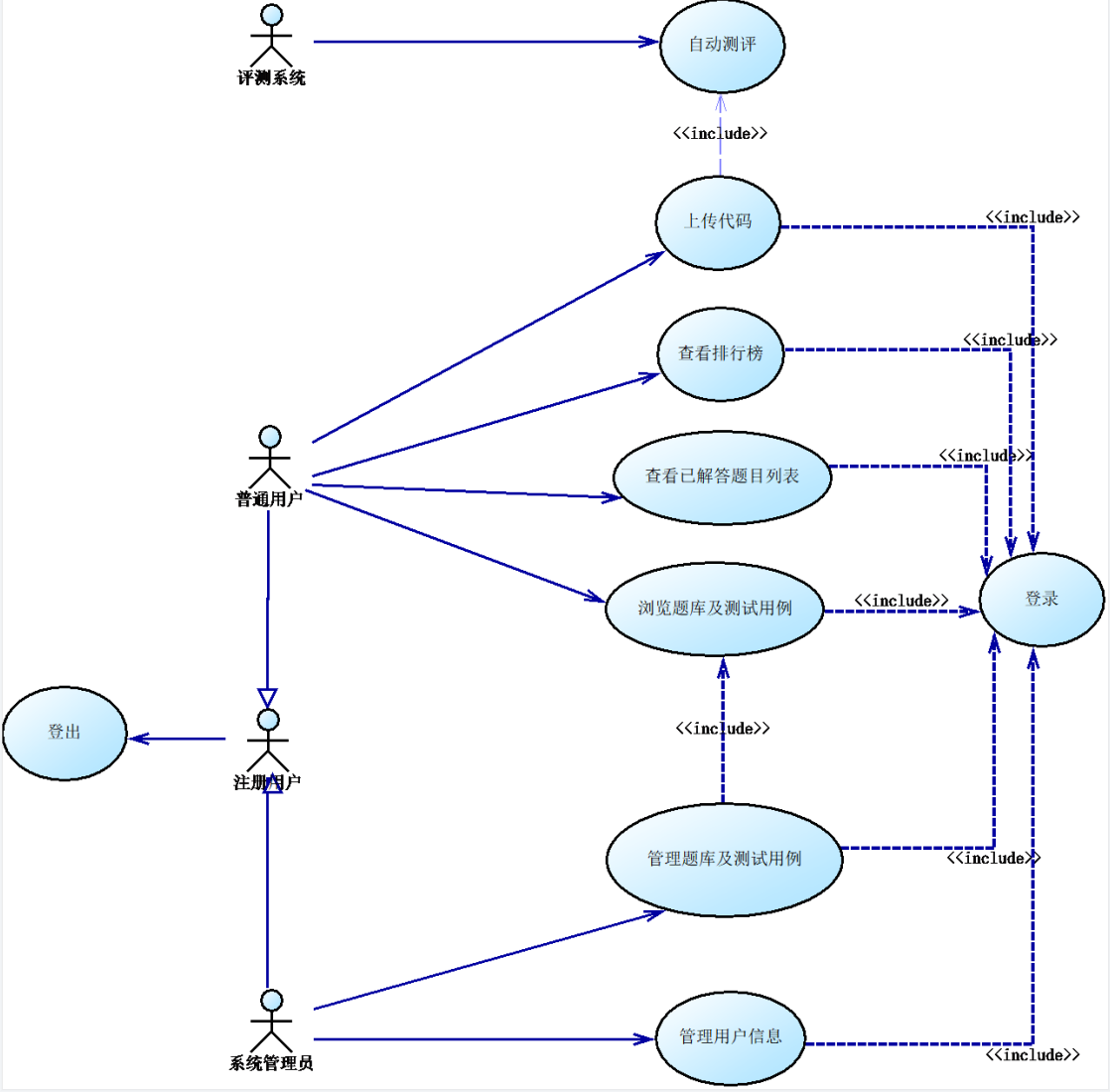
* 直观易用的界面设计
* 响应式布局，适应不同设备和屏幕尺寸
* 浏览器兼容性

1. 可靠性子集：

* 异常处理和错误提示
* 恢复能力，防止系统崩溃和数据丢失
* 数据备份和恢复机制

3. **功能需求**

**<Use case 图>**



Actor：

1. 注册用户：已注册用户，经登录可以使用"Testland"平台。
2. 普通用户：平台的一般用户，可以操作平台并进行普通用户权限下的操作，区别于系统管理员。
3. 系统管理员：特殊身份用户（非普通用户），可以管理平台上的内容，包括题库与测试用例。
4. 评测系统：负责自动测评的系统，会进行自动编译及基于测试用例的测评。

usecase：

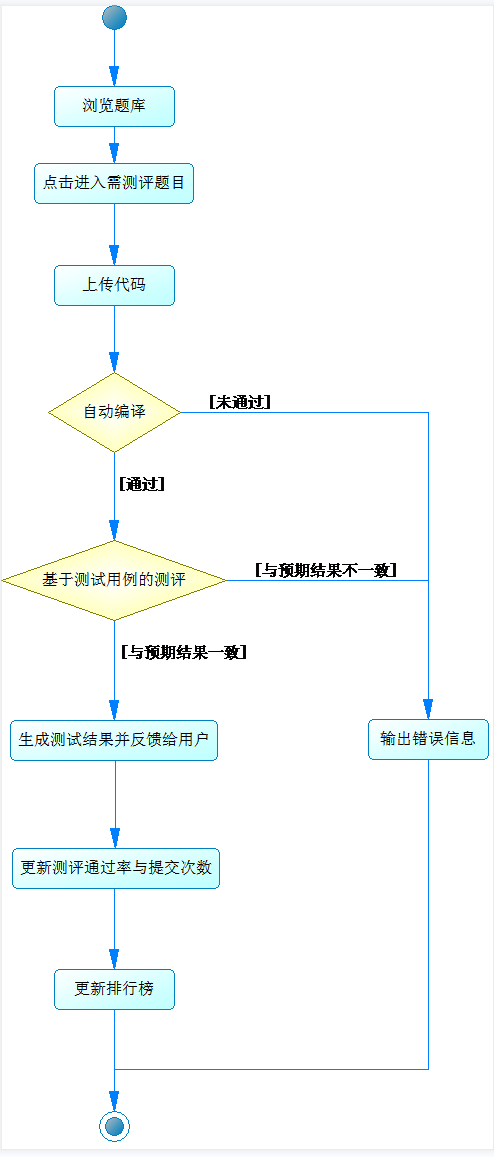
1. 登录：已注册用户进入平台进行操作的途径，用于识别身份，且区分系统管理员与普通用户模式。
2. 登出：已登录用户退出账号的操作。
3. 浏览题库：普通用户与管理员均可查看平台上的题库。
4. 管理题库及测试用例：系统管理员可以管理题目及对应测试用例，包含了浏览题库操作。
5. 上传代码：普通用户可选择不同编程语言，将答案上传到对应题库下，用于自动测评。
6. 查看已解答题目列表：普通用户可以查看自己已经解答的题目列表，包括题目名称、通过率、提交次数等。
7. 查看排行榜：普通用户可以查看平台根据用户的题目解答情况（综合考虑解题数、题目难度、一次提交成功率等）进行排名后生成的排行榜。
8. 管理用户信息：系统管理员可以禁用或解禁用户。
9. 自动测评：评测系统进行的自动编译和基于测试用例的自动测试。

**<Use case1 规约>**

**用户上传代码及评测系统自动测评：**

详细过程：

普通用户浏览题库，点击可进入需要测评的题目，可按需选择不同编程语言，进行代码上传，平台会对上传的代码进行自动编译，若通过则进行基于测试用例的测评，两环节中有不成功或与预期不符的情况，会输出错误错误信息并直接结束，若基于测试用例的测评也通过，则生成测试结果并反馈给用户，更新提交次数与通过率，然后根据解题数、题目难度、一次提交成功率等信息更新用户排行榜。

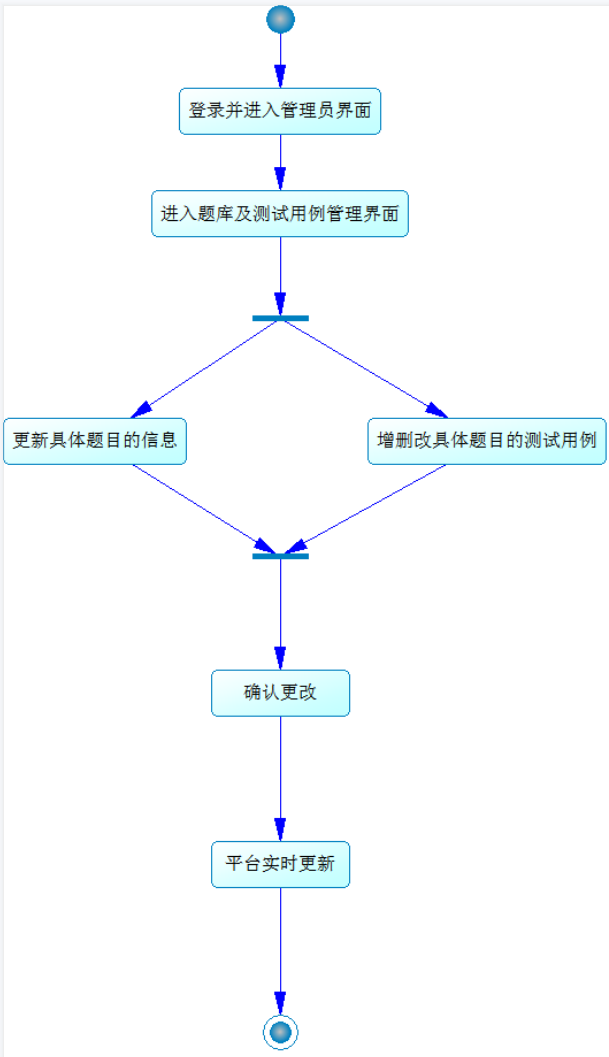


**<Use case2 规约>**

**系统管理员管理题库及测试用例：**

详细过程：

首先，管理员需登录进入管理员界面，进入题库与测试用例管理界面，在此页面更新具体题目信息或增删改题目的测试用例，此时需确认修改，使修改后的题库实时更新在平台上。



4. **非功能需求**

**4.1 易用性**

* 培训时间：普通用户最大培训时间不得超过30分钟，高级用户最大培训时间不得超过15分钟。
* 可评测任务次数：每个典型任务至少需要进行5次评测，以确保系统的稳定性和易用性。
* 符合易用性标准：系统交互界面应符合IBM的CUA标准和Microsoft的GUI标准等公认易用性标准，页面布局、交互方式、字体颜色等方面需具有统一性和美观性，用户调查结果的平均满意度不得低于85%。

**4.2 可靠性**

* 可用性：系统应该能保证24小时不间断运行，一年内该系统平均可用时间至少达到99.5%。
* 平均故障间隔时间 (MTBF)：MTBF 的目标时间最好达到 5,000 小时或更多，并不断优化以提高系统的可靠性。
* 平均修复时间 (MTTR)：系统在发生故障后，平均修复时间目标应该控制在 2 小时内，减少影响用户的停机时间。
* 精确度：系统在输出结果时，应该具备用户要求的精度和精确度，如代码评测失败时能反馈用户失败原因。
* 最高错误或缺陷率：每千行代码错误数应该小于 1 - 5 个，具体的错误率还取决于代码复杂性、程序员技能水平等因素；每个功能点错误数应该小于 1 个或更少，这也取决于功能点的复杂性和规模。
* 错误或缺陷率：不出现严重错误，如用户上传评测代码数据丢失；大错误每个功能点不多于0.5个；小错误每个功能点不多于1个。

**4.3 性能**

* 对事务的响应时间：平均响应时间应该在 1-2 秒之内，最长响应时间应该控制在 5 秒以内。
* 吞吐量：每秒处理的事务数需要达到上百个或者更多。
* 容量：系统应支持1000个并发用户。
* 降级模式：当流量激增时，可以通过缩短响应时间或者关闭一些不必要的功能来降低系统负载。
* 资源利用情况：需要对内存和磁盘使用情况进行监控和调整，以保证其在合理的范围内，并避免出现资源不足的情况导致系统崩溃。

**4.4 兼容性**

客户端浏览器兼容性要求：

* Google Chrome：114及以上版本。
* Mozilla Firefox：114及以上版本。
* Apple Safari：170及以上版本。
* Microsoft Edge：最新版本及之前的几个主要版本。

服务器端兼容性要求：

* 操作系统：Linux Ubuntu 18.04
* 数据库：MySQL、MongoDB。

**4.5 可支持性**

* 编码标准：制定统一的命名规范、代码结构规范等以确保开发人员的代码风格统一，本次开发中代码应符合Google代码规范。
* 命名约定：为变量、函数、类等元素制定明确的命名约定，采用有意义的名称来描述变量的作用以增加代码的可读性和可维护性。
* 类库：尽量采用社区流行、稳定性高的类库，在使用第三方类库时，需要确保其版权和授权合法有效，并通过评估确认其安全性和稳定性。
* 维护访问权：对系统进行访问控制，并限制非授权人员的访问权限。
* 维护实用程序：使用日志管理工具、数据库监控工具等实用程序和工具，以便于系统的维护和管理工作。

**4.6 设计约束**

* 软件语言：采用Java语言进行开发。
* 软件流程需求：采用敏捷开发方法，按照迭代循环的方式进行软件开发。
* 开发工具的指定用途：在开发过程中，需要使用 Git 进行版本控制，使用Maven进行项目构建和依赖管理，使用IntelliJ IDEA等IDE工具进行开发和调试。
* 架构及设计约束：采用Spring框架进行应用程序的开发和集成，使用MySQL数据库存储数据，前端采用JavaScript等技术进行开发。
* 购买的构件：需符合已选架构和设计标准，通过严格的评估审查后方可采购和应用于系统。
* 类库：尽量采用社区流行、稳定性高的类库，在使用第三方类库时，需要确保其版权和授权合法有效，并通过评估确认其安全性和稳定性。

5. **其它产品需求**

**5.1 联机用户文档和联机帮助的需求**

1. 用户指南：

* 提供系统的概述和功能说明，包括如何注册、登录、退出等基本操作。
* 详细说明系统中支持的各种功能和模块的用法和操作步骤，例如查看题目列表、选择题目、提交代码等。
* 提供示例和截图，以便用户更好地理解和遵循指南。

1. 帮助文档：

* 提供针对常见问题和疑问的解答，包括系统使用中的常见错误和故障排除方法。
* 解释系统的术语和概念，帮助用户理解系统的工作原理和背后的逻辑。
* 提供用户界面的说明和导航指引，以便用户可以快速找到所需的功能和信息。
* 可以提供搜索功能，方便用户根据关键词查找相关内容。

**5.2 接口需求**

**5.2.1 用户界面**

1. 登录界面：登录界面包含用户名和密码输入框，以及登录按钮。用户输入正确的用户名和密码后，点击登录按钮进行身份验证。
2. 注册界面：注册界面包含用户名、密码、确认密码等字段的输入框，以及注册按钮。用户填写完整的注册信息后，点击注册按钮进行账户注册。
3. 题目列表界面：题目列表界面展示系统中支持的题目列表，包括题目名称、难度、标签等信息。用户可以通过点击题目进入题目详情页面。
4. 题目详情界面：题目详情界面显示选定题目的详细信息，包括题目描述、示例输入输出、输入限制、输出限制等。用户可以选择编程语言，并输入代码解答问题，然后提交代码进行评测。
5. 评测结果界面：评测结果界面展示用户提交代码的评测结果，包括编译信息、运行时间、内存占用、输出信息等。用户可以查看代码的评测详情。
6. 我的解答界面：我的解答界面显示用户已经解答的题目列表，包括题目名称、通过率、提交次数等信息。用户可以点击题目进入评测结果界面查看详细的评测信息。
7. 排行榜界面：排行榜界面展示用户的排名信息，根据用户的题目解答情况（综合考虑解题数、题目难度、一次提交成功率等）进行排名。
8. 管理员界面：管理员界面用于管理题目及其测试用例，管理员可以添加、编辑、删除题目和对应的测试用例。此界面仅对管理员可见。

**5.2.2 软件接口**

1. 用户接口：

* 描述：用户通过Web浏览器或IDE插件与代码测评平台进行交互。
* 预期行为：用户可以注册、登录、查看题目列表、选择题目、提交代码、查看评测结果等。

1. 第三方API接口：

* 描述：代码测评平台可能需要与第三方服务集成，例如发送邮件通知、进行支付等功能。
* 预期行为：通过使用第三方API接口，代码测评平台可以实现与第三方服务的交互和功能扩展。

1. 外部系统接口：

* 描述：代码测评平台可能需要与其他子系统进行交互，这些子系统可能是购入的构件或来自其他应用程序的构件。
* 预期行为：通过与外部系统接口进行交互，代码测评平台能够获取或提供所需的数据和功能。

**5.2.3 通信接口**

1. 用户终端（Web浏览器或IDE插件）：

* 描述：与用户终端进行通信，接收用户的请求并提供相应的响应。
* 通信方式：HTTP协议或WebSocket等网络通信协议。

1. 数据库服务器：

* 描述：与数据库服务器进行通信，对用户信息、题目列表、用户解答记录、评测结果等数据进行读写操作。
* 通信方式：一般使用数据库相关的API或协议，如MySQL的TCP/IP协议、MongoDB的RESTful API等。

1. 第三方服务：

* 描述：与第三方服务进行通信，例如发送邮件通知、进行支付等功能。
* 通信方式：使用第三方服务提供的API进行数据传输，通常采用HTTP协议或其他适当的协议。

**5.3 适用的标准**

1. 安全标准：

* ISO/IEC 27001:2013 - 信息安全管理系统标准，确保用户数据的机密性、完整性和可用性。
* OWASP Top 10 - 针对Web应用程序的安全风险管理的指南。

1. 编程语言标准：

* 根据支持的编程语言，参考各个语言的官方文档和规范，例如C标准（ISO/IEC 9899）、C++标准（ISO/IEC 14882）、Java语言规范等。

1. Web开发标准：

* HTML5 - 最新的HTML标准，定义了Web页面的结构和内容。
* CSS3 - 最新的CSS标准，用于美化和布局Web页面。
* JavaScript标准 - ECMAScript规范是JavaScript的官方标准。

1. 用户体验标准：

* ISO 9241：人因工程学界的一系列标准，关注人机界面设计和用户体验。

1. 互操作性标准：

* RESTful API设计和实现的最佳实践，确保代码测评平台与其他系统之间的数据交换和集成。

1. 国际化标准：

* Unicode - 字符编码标准，支持多种语言和字符集。

1. 操作系统相容性标准：

* POSIX标准 - UNIX及类UNIX操作系统的接口标准。
* Windows API - 用于与Windows操作系统进行交互和开发的应用程序接口。

1. 质量标准：

* ISO/IEC 9126：软件质量特性和度量的国际标准，包括功能性、可靠性、可用性、效率、可维护性和可移植性等方面。