

时光清浅日程软件开发计划

文档编号：DOC002727

文档类型：项目文档

作者：20215898 习婉婷（组长），20215925 吉廉嘉，20215932 吕希彤，20215845 孙欣欣

版本：01

日期：2024/4/6

批准信息：

角色	签字	日期
研发负责人	习婉婷	2024/4/6
项目经理	杨金柱	2024/4/7

		版本： 01
		页码： 1 / 14

版本信息：

版本	作者	日期	变更描述
01	习婉婷	2024.4.6	新建

注释：

- 1. 使用本文档前请确认是最新的版本。
- 2. 本文档的所有页都应该保持为同一版本。

		版本： 01
		页码： 2 / 14

1. 项目介绍

1.1 项目总括

随着生活节奏的加快和人们对时间管理的需求增加，日程管理成为了越来越多人生活中的重要组成部分。传统的纸质日历已经不能满足人们对灵活性和便捷性的需求，因此我们决定开发一款全新的日程软件，以满足现代用户对日程管理、提高效率的需求。

时光清浅日程软件(Schedule Management Software)是自主研发的用于轻松管理用户日程、提高用户效率及专注度的软件产品。我们的目标用户包括白领人士、自由职业者、学生、家庭主妇等一切有有效地管理工作和个人生活需求的人群。软件提供了用户注册与登录、添加、查看、编辑、删除和管理日程安排等基本功能。系统还支持通过云盘进行多设备间的日程同步、数据备份与恢复等保障数据一致性与安全性的功能。除此之外，我们在市场已有日程管理软件基础上，创新性地为软件添加专注模式等其他功能，帮助用户更高效地完成日程。

具体功能分析与介绍如下一章节所示。

1.2 功能分析与介绍

我们拥有一系列日历与日程管理软件的基本功能，详细介绍如下：

1. 日程创建功能：

用户可以轻松创建各种类型的日程安排，如会议、紧急任务、家庭聚会等，只需输入相关信息，如主题、地点、参与者等。同时，在日程创建时，用户需要选择具体日期和时间，以便为日程安排设定具体的开始和结束时间；用户也可以为日程安排设置重复周期，如每天、每周、每月等，以便管理定期任务。

2. 日程编辑功能：

用户可以随时修改已创建的日程安排，包括任务的日期、时间、重复周期等，以满足实际需求。

3. 日程删除功能：

用户可以删除不再需要的日程安排，以保持日程表的整洁和高效。

4. 重要事件提醒设置：

用户可以为日程安排设置提醒方式，如通知提醒、闹钟提醒等，确保不错过任何重要事件。

5. 任务优先级设置：

用户可以为日程安排设置优先级，以便更好地安排工作和生活，确保优先完成重要任务。

		版本： 01
		页码： 3 / 14

6. 视图切换功能:

本软件主要设置两种视图模式。用户可以查看日历视图，以便更方便地查看和管理日程安排；用户也可以查看任务清单，以便快速了解即将进行的任务。

7. 导出日程:

用户可以将自己创建好的日程以文件的形式转发，方便分享或者留存。

8. 番茄工作法:

本软件提供番茄工作法（Pomodoro Technique）的功能，帮助用户更好地管理工作和提高工作效率。用户可以设定工作时间和休息时间，应用会自动提醒用户进行工作或休息。

9. 习惯养成计划:

支持用户创建和跟踪个人习惯的养成情况。用户可以设定每日、每周或每月的目标，应用会提醒用户完成任务，并记录习惯养成的进度。

10. 数据分析功能:

提供时间统计功能，记录用户不同任务或活动的时间利用情况。用户可以查看时间分布图和统计饼状图，了解自己的时间利用情况。除此之外，用户可以查看自身的习惯养成情况，同时本软件将向用户提供时间管理建议和优化方案。

11. 数据同步功能:

用户可以通过云盘将日程列表与手机、电脑等多个设备同步，以便随时随地查看和管理日程安排，保证数据一致性。

12. 数据备份与恢复功能:

用户可以手动备份日程数据至本地或云端存储，确保数据的安全性和可靠性。提供数据恢复功能，用户可以在需要从备份文件中恢复日程数据。

除上述基本功能外，拟开发软件拥有许多创新功能，详细介绍如下:

1. “专小果”功能树:

我们设计的软件“时光清浅”，可以通过种植虚拟的小树来帮助用户专注于工作或学习。当用户开始专注于一项任务时，他们可以种植一棵小树，并设定一个计时器。在这段时间内，用户需要保持应用界面处于前台，不能离开应用去做其他事情。如果用户成功保持专注一小时，即可获得一次浇水或者施肥的机会。在用户的浇水和施肥下，小树会逐渐生长成一棵大树，如果用户离开应用，种植的树木将被砍倒。

当小树长成大树后，可以获得一次邮寄水果的机会。我们的水果均来自于美丽中国的偏远地区。在用户专注收获满满的同时，不仅有好吃的水果，也更加助力偏远地区的水果经济。

		版本: 01
		页码: 4 / 14

2. 健康与生物钟整合：

根据用户的生物钟和健康需求，优化日程安排。例如，根据用户的睡眠质量，调整早晨的起床时间和一天的活动安排。

3. 多人自习室功能：

在网络发达，线上的工作和学习也是必不可少的。为此本软件提供了多人在线同时完成日程的机会。创建自习室的用户将成为室长，室长最多可以设置五个管理员，管理员和室长可以控制自习室里的用户的人数和状态，包括但不限于将用户从自习室中剔除、邀请用户加入、邀请用户连麦、视频和将用户闭麦、关视频等。每个用户每周最多可创建五次自习室模式，每次自习室里最少需要有 2 名用户，最多 50 名用户。

如有更多创建需求，可以根据不同的需要购买 VIP 或 SVIP。VIP 用户创建自习室模式不限次数，人数上限提高至 500 人。SVIP 用户在不限次数的时候，将同时在线人数提高至 2000 人。每月为普通用户提供两次 VIP 体验卡，需要观看 5 次赞助商的广告，每次看完五个视频立即生效，每张体验卡的有效时间为 24 小时。

1.3 软件本期开发生命周期（过程模型）

1. 软件开发前期将遵循快速原型模型进行开发，以明确用户基本需求，流程详见图 1。

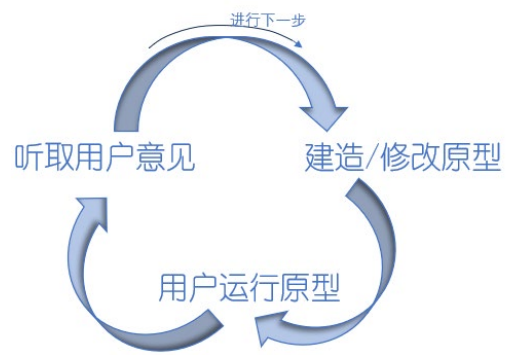


图 1 前期开发模型（快速原型模型）

选择此模型进行前期开发的主要原因分析如下：

首先，本开发团队经验不足，在开发初期开发者不确定所选用算法的有效性以及交互界面的形式，因此开发人员可以通过建立快速原型验证算法的正确性并确定 UI 设计。此外，开发人员可以通过建立原型系统快速积攒经验，降低在后续设计、编码和测试等环节发生错误的可能性。

其次，此日程软件所涉需求繁杂，若客户不能准确、全面地描述其需求，可以在软件开发初期建立一个简化的样品程序以期与客户进行有效沟通。当原型系统已经通过客户运行测试，应当据此产生规格说明文档正确全面地描述用户需求。因此，在开发时间紧张的情况下，在后续开发

		版本： 01
		页码： 5 / 14

过程中不会因为发现规格说明文档的错误而进行较大的返工，同时也方便更加明确地规划实际开发过程中使用增量模型时的每次增量。

2. 规格说明文档完成后，在实际开发过程中将遵循**增量模型**进行开发，以满足全部系统需求，流程详见图 2。

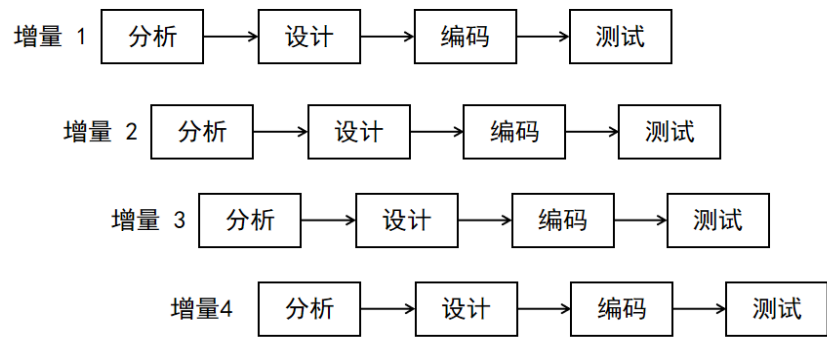


图 2 开发模型（增量模型）

选择此模型进行开发的原因分析如下：

首先，在开发团队经验不足的情况下，一次性将满足所有需求的产品成功开发并通过测试提交给客户难度较大且风险性较高。因此，可以选用增量模型分批开发不同功能模块以满足客户全部需求，分批地逐步向客户提交产品，同时防范“一步到位”的技术风险。

其次，本项目所开发的产品功能复杂，逐步增加产品功能可以使用户有较充裕的时间学习和适应新产品，从而减少一个全新的软件可能给客户组织带来的巨大冲击。

在使用上述两种模型完成系统基本开发后，还应进行如下两步：

- 软件批准发行
- 软件后期维护

其中，软件发行版本根据开发情况不唯一，软件后期需要对整个开发流程进行维护。

1.4 项目度量

1. 项目规模

项目规模分为码量与人月两个方面的估计。

码量：依据 PM（项目经理）的估计，一个包含用户认证、日程创建与编辑、提醒设置、日历视图、数据同步等核心功能的基本日程管理 APP，代码量可能在数千行到一万行左右。然而，如

		版本： 01
		页码： 6 / 14








果要开发一个功能更加完善、界面更加精美、用户体验更加出色的日程管理 APP，代码量可能会显著增加。特别是当涉及到创新功能，如专注果树、小游戏、生物钟整合以及多人自习室等时，这些功能需要额外的代码来实现。因此，我们预估最终的码量在三万行左右。

人月：本项目预计开发四个月，项目人员 5 名。据此估计人月=5 人×4 月=20 人月。

2. 项目成本

项目成本分为人力成本和其他资源成本。人力成本主要包括开发人员薪资以及奖金等，其他资源成本主要包括笔记本电脑、高性能服务器、路由器等硬件设备和项目管理成本以及其他风险成本。

2. 软件开发项目成员（人员分工）

Role	Personnel	Responsibility
开发工程师	习婉婷，吉廉嘉，吕希彤，孙欣欣	 开发完成软件代码  完成软件的单元开发，集成开发及系统开发  完成软件的发布文档
项目软件负责人	习婉婷	 完成各类软件开发过程中的文档  参加软件设计缺陷评审  确保软件进度的按时完成  批准软件的结构设计  批准软件测试报告  批准软件发布
项目经理 (PM) *	杨金柱老师	 批准软件开发设计计划  确保软件进度的按时完成  召开软件设计缺陷评审
质量负责人 (QA)	吕希彤	 参加软件设计缺陷评审  参与软件设计开发计划，测试计划和测试报告的评审
测试工程师	习婉婷，吉廉嘉，吕希彤，孙欣欣	 按照验证计划要求，按时完成测试  完成相应的测试报告

		版本： 01
		页码： 7 / 14

3. 软件运行机制

- 软件部门负责人将每半个月举行软件项目会，沟通项目进展情况，及后续项目部署。
- 软件工程师将参与项目缺陷评审，讨论解决软件相关问题。
- 软件开发设计计划会不定期的更新，以反映软件开发进度。

4. 软件开发语言与工具

4.1 开发语言

本日程管理 APP 包含十余项基本功能和四项特色功能，这将为用户带来丰富的使用体验、便捷的工作和生活方式、跨平台使用的便利、数据安全和隐私保护以及市场竞争优势等多方面的优势。然而，功能的增加也会提高操作的复杂性，增加 APP 的内存占用和 CPU 使用率，导致性能下降，影响 APP 的响应速度和稳定性，降低用户满意度。因此需要在功能丰富性和易用性之间找到平衡。为此，本开发团队选择 C++作为本软件开发项目的主要开发语言。

使用 C++语言开发的主要优势详细分析如下：

1. 性能优势：

C++是一种高效的编程语言，能生成快速且紧凑的代码。在开发需要高性能和低延迟的应用程序时，C++在编译优化、内存管理、数据结构和算法等方面的优势能够显著提升程序的效率和性能。对于日程管理 APP 来说，特别是当需要处理大量数据或进行复杂的计算时，C++的这些性能优势能够确保应用流畅运行，提供良好的用户体验。

2. 代码组织清晰，易于修改、维护：

C++支持面向对象编程，这使得代码的组织更加清晰、易于理解和维护。在日程管理 APP 中，可以通过封装、继承和多态等特性，更有效地组织和管理代码，实现更复杂的功能。

3. 跨平台性：

本应用需要支持跨平台使用，如同时在 Android 和 iOS 上运行。C++具有良好的跨平台性，可以在不同的操作系统和硬件上运行。这使得日程管理 APP 能够更容易地实现跨平台开发，满足不同用户的需求。

4. 丰富的库简化开发过程：

C++拥有丰富的标准库和大量的第三方库，这些库提供了许多常用的数据结构和算法，以及网络、文件操作、图形渲染等功能。使用这些库可以大大简化开发过程，提高开发效率。

		版本： 01
		页码： 8 / 14

5. 社区支持和资源丰富：

C++拥有庞大的开发者社区和丰富的资源，包括教程、文档、论坛和开源项目等。在本开发团队项目经验、代码经验都不足的情况下，这为我们提供了强大的支持，可以从中学习到新的技术和最佳实践，遇到问题时也能够迅速找到解决方案。

4.2 技术重点难点分析

日程管理 APP 的技术重点和难点详细分析如下。

技术重点，主要包括以下三方面。

1. 用户界面设计：

C++本身并不直接支持图形用户界面(GUI)的开发。通常需要使用额外的库或框架，如 Qt 等来实现用户界面。这要求开发者熟悉所选框架的 API 和设计模式，以便创建直观、易用的用户界面。

2. 数据一致性、完整性、安全性：

时光清浅日程软件需要保存用户的数据，以便在关闭和重新打开应用时能够恢复数据。同时，需要确保用户可以手动备份日程数据至本地或云端存储，以便在不同设备间实现同步。这通常涉及到文件操作或数据库的使用，开发者尤其需要注意维持数据的一致性、完整性、安全性。

3. 多线程和异步处理机制：

为提高应用程序的响应速度和用户体验，可能需要在后台处理一些耗时的任务，如数据同步、提醒服务等。这涉及到多线程和异步编程的技术，需要开发者熟悉事件处理机制，同时对多线程编程有深入的了解。

技术难点，主要包括以下四个方面。

1. 跨平台兼容地开发应用程序：

时光清浅日程软件设计在不同的操作系统或平台上运行，所以需要充分考虑跨平台兼容性的问题。可以使用跨平台的库和框架，并进行充分的测试和调试，这将是一个极大的难点。除此之外，本团队从未有过开发应用程序的经验，也未掌握开发应用程序的技术，这将是更大的技术难点。

但是，软件发布时可以考虑在不同时间发布支持不同操作系统的应用程序。因此，可以逐步开发，逐步发布，同时在逐步开发的过程中，开发人员可以逐步掌握不熟悉的开发技术。

2. 数据管理和同步机制：

		版本： 01
		页码： 9 / 14

时光清浅日程软件设计主要利用云端存储进行设备间同步、数据备份与恢复等功能。这需要管理大量数据，在开发过程中，开发人员尤其需要注意维持数据的一致性、完整性、安全性。除此之外，云盘同步等利用云端存储的技术开发人员还未掌握。

3. 错误处理机制：

用户在使用应用程序过程中，可能会遇到各种预料之外的情况，如文件读写失败、网络连接中断等。开发者需要设计合理的错误处理机制，以确保应用能够在遇到错误时给出明确的提示，并尽可能地恢复到正常状态。

4. 性能优化：

对于功能繁杂、需要处理大量数据或频繁进行复杂计算的时光清浅日程软件来说，性能优化是一个重要的挑战。开发者需要通过算法优化、减少不必要的计算和数据传输等手段来提高性能。

4.3 软件开发工具

Index	Tools	Version	Description or Reference
1.	Visual Studio Code	1.73.0 (system setup)	系统应用程序开发环境
2.	Qt	5.15 LTS	系统 UI 原型设计工具
3.	VSS	Version 8.0	系统开发版本控制管理工具
4.	StarUML	5.0	系统分析、设计和建模等 UML 语言开发工具。
5.	Office Visio 2013	Professional 64	系统分析、架构、流程设计等可视化开发工具。
6.	PostgreSQL	14	数据库维护管理工具
7.	Mantis	1.2.11	缺陷跟踪管理工具
8.	Microsoft Project	2016	产品开发周期管理工具

5. 软件开发周期（进度计划）

本项目开发周期为 2024.03—2024.07，开发人员将分阶段完成整个软件模块的研发，具体进度计划详见图 3。

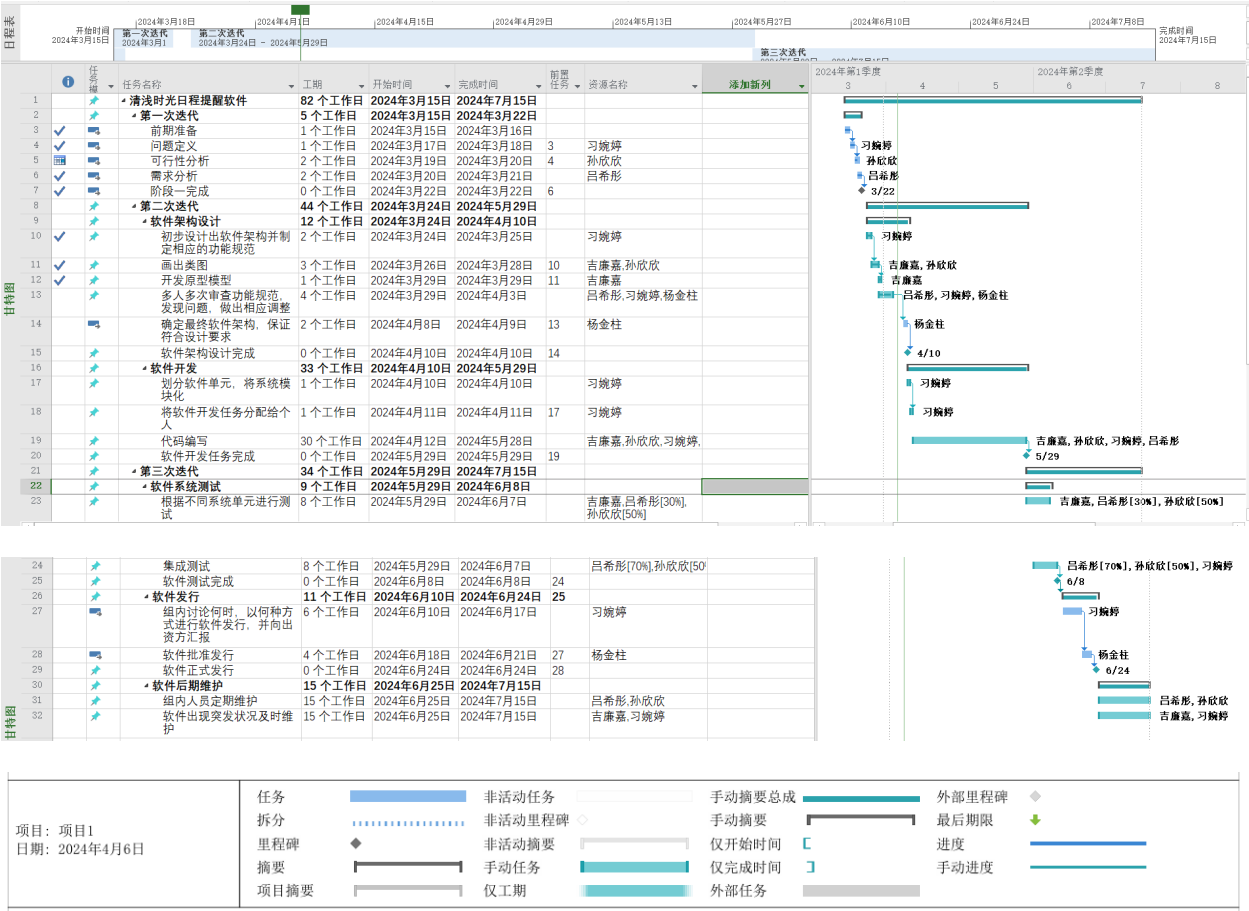


图 3 软件开发进度计划（甘特图）

以上进度计划可能在实施过程中按需调整，下面对进度计划的制定简要说明理由：

出于现实考虑在甘特图中建立了一定的依赖关系，例如代码编写之前必须先完成任务划分，做好人员合理分工才能进行。除此之外，在甘特图中标注出了关键信息，标注出重要的里程碑事件，大多在完成一个里程碑事件之后才能进行下一步开发，推进项目整体进度。

此进度计划中还包含了实际进度信息，比较实际进度与计划进度的差异。在甘特图中，实际进度较计划进度落后了一些，推测原因可能是清明假期导致的人员工作效率减少。

在资源分配上，由于项目组人员紧缺，导致许多成员必须同时多线程工作，可能导致出现任务延迟、资源冲突等问题，后期团队成员可以根据甘特图进行调整，如调整任务顺序、优化资源配置、解决潜在风险等。

		版本： 01
		页码： 11 / 14

6. 项目要点交付物及时间点

对应上节中的软件开发进度计划，下表给出项目要点交付物及具体的交付时间点。

项目要点	文档	Owner	时间点
软件开发计划	☑	项目软件负责人	2024. 3. 15
软件需求分析	☑	质量负责人	2024. 3. 21
软件风险管理计划	☑	项目软件负责人	2024. 3. 22
软件概要设计	☑	项目软件负责人	2024. 3. 24
软件详细设计	☑	开发工程师	2024. 3. 26
软件测试计划	☑	测试工程师	2024. 5. 29
软件配置计划	☑	开发工程师	2024. 5. 29
测试用例	☑	测试工程师	2024. 5. 29
软件测试总结	☑	测试工程师	2024. 6. 8
用户手册	☑	开发工程师	2024. 6. 10
对外发布	☑	项目软件负责人	2024. 6. 10
项目总结报告	☑	项目软件负责人	2024. 6. 25

7. 软件追溯性要求

7.1 系统要求和软件要求间的追溯性

系统要求是从客户需求中导出的高级需求，描述了系统的功能、性能、安全性等方面的要求，在快速原型模型构建完成并通过客户测试后基本确定。而软件要求是从系统要求中进一步细化得到的，描述了软件系统的详细功能、接口、性能、设计约束等方面的要求。系统要求中对软件的需求全部可在软件需求中追溯到，以确保软件开发过程中的需求完整性和一致性。

7.2 软件测试和软件需求间的追溯性

软件测试是软件开发过程中用来验证软件系统是否符合需求，并发现潜在的缺陷和问题的重要步骤。为确保软件需求中的功能都已成功实现并运行稳定，在追溯列表中，所有的软件需求都将追溯到至少一条测试用例。

例如，在用户注册与登录过程中，至少需要四种测试用例。测试用例 1：输入有效的用户名和密码，测试用户能够成功注册。测试用例 2：输入重复的用户名或两次验证不一致的密码，测试用户无法成功注册。测试用例 3：输入无效用户名，提示用户用户名不正确或进行注册。测试用例 4：输入无效密码，无法成功登录。根据示例，测试人员需设计合理、完备的测试用例，将所有的软件需求都在软件测试中进行验证。

		版本： 01
		页码： 12 / 14

7.2 风险控制 and 软件实施间的追溯性

风险控制是识别、分析、评估和应对软件开发过程中的各种潜在风险的过程。风险分析中需求相关的风险控制要求都包括于需求分析中，在需求变更等风险来临时，查阅需求分析文档中的应对策略，以协调开发人员做出快速、合理的应对。除此之外，在软件开发的具体实施过程中，需要为用户培训，强调变更管理的重要性，让用户了解如何提交变更请求以及变更对系统的影响。

风险分析中软件相关的风险控制要求都包括于软件要求中，在部署计划中考虑技术风险，提前规划技术验证和测试阶段，确保在部署前解决技术问题。

8. 软件配置管理

8.1 软件代码保存

为实现有效的软件配置管理，通常会使用一些配置管理工具。本项目的 软件代码使用 GIT 进行软件配置管理。

8.2 配置更改

任何软件单元配置的更改，都将通过该文件的升版反映。

在配置更改时，开发人员尤其需要注意确定项目中的配置项，并为每个配置项分配唯一的标识符；管理配置项的版本，确保对软件工件的修改得到有效控制。除此之外，开发工程师需要提供定期的配置状态报告，包括配置项的版本信息、变更请求的状态、配置问题和解决方案等；并组织项目经理、项目软件负责人、质量保证员进行合理评审。

8.3 软件版本标示信息

软件版本标示遵守软件命名规则，如时光清浅 1.0，时光清浅 2.0 等，以便通过版本号或其他信息快速识别软件的版本和状态。

8.4 现成软件管理

软件中不包含现成软件，因此不必对其进行特别的管理。

8.5 软件风险分析及缺陷管理

软件开发过程中所面临的风险，主要包括以下三个方面。

1. 技术风险：主要考虑三方面。

- 🚩 平台兼容性问题：不同操作系统和设备的兼容性可能导致软件在某些平台上运行不稳定或功能受限。

		版本： 01
		页码： 13 / 14

- ✚ 数据安全性：用户隐私数据泄露、数据丢失或被盗用的风险。
- ✚ 网络连接问题：依赖于网络连接的功能，如同步云端数据、更新日历等，可能受到网络不稳定性的影响。

2. 项目管理风险：主要考虑两方面。

- ✚ 需求变更：由于客户需求的频繁变更，可能导致项目进度延误和额外成本。
- ✚ 人力资源不足、人员离职或变动：考虑到开发团队内部成员较少，一人身兼数职，需尽快招到合适的员工。除此之外，也要防范人员离职或变动等导致的人员更加紧缺。

3. 竞争和市场风险：主要考虑两方面。

- ✚ 竞争激烈：市场上存在其他日程管理软件，竞争压力大，需要不断改进和创新以保持竞争优势。
- ✚ 用户接受度：用户对新功能的接受程度可能低于预期，影响软件的市场表现。

软件缺陷管理遵守文档要求。测试工程师在测试过程中发现的所有软件缺陷都将形成测试报告。项目软件负责人和相应的软件开发工程师参与项目缺陷评审，以期跟踪解决软件缺陷。

		版本： 01
		页码： 14 / 14