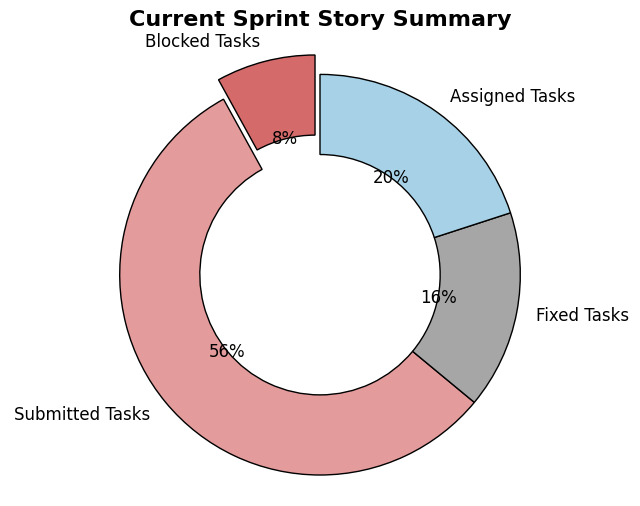
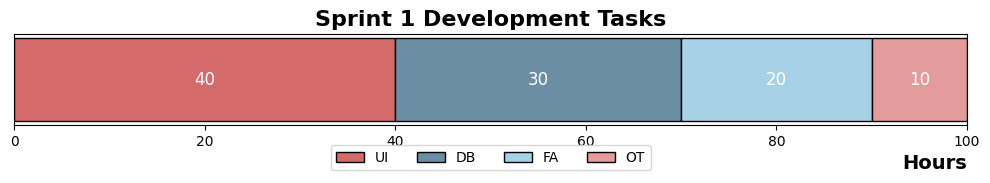
Sprint Report

Sprint 1（第2-5周）报告

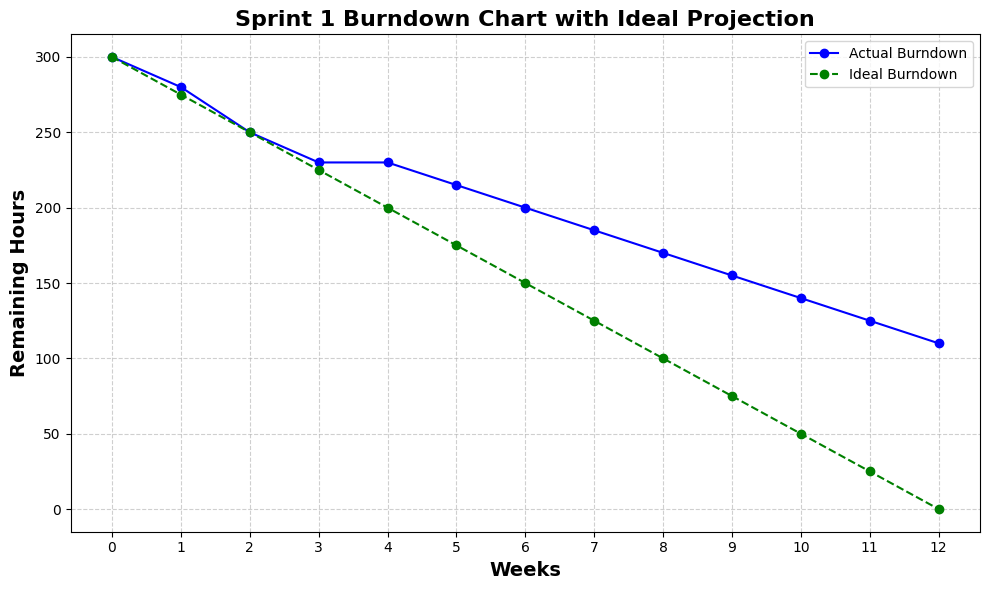
1. Sprint概述
   1. Sprint编号：2024-9
   2. 起始日期：2024.9.9
   3. 结束日期：2024.9.30
   4. 团队人数：8人
   5. 预估工作量：100小时
2. Sprint目标：基础功能和架构搭建，满足基本需求。
3. 已完成的工作
   1. 用户故事/任务：
      1. **用户界面模块**
         1. **描述**：实现了文件上传子模块，用户可以通过前端界面上传Python项目文件。完成了初始界面的开发，展示了CWE问题概要和代码质量评分的概览页面。
         2. **故事点**：15个故事点。
      2. **数据库模块**
         1. **描述**：完成了数据库架构的初步设计，建立了用户项目提交记录的存储模块。
         2. **故事点**：10个故事点。
   2. 验收标准：
      1. **用户界面模块**：
         1. 文件上传功能在不同浏览器环境下均可正常使用。
         2. 概览页面可以正确显示上传项目的CWE问题数量和代码质量评分，界面设计符合用户友好性要求。
      2. **数据库模块**：
         1. 数据库设计符合系统架构标准，能够存储用户的项目提交记录，包括CWE问题数量和代码质量评分。
         2. 存储的数据通过基础查询验证，确保正确性和稳定性。
4. 未完成的工作：
   1. **未完成的项目：**
      1. 文件分析模块
         1. 文件分析模块未能全部完成，主要集中在复杂的错误检测和处理部分。涉及多层嵌套结构、递归调用以及一些动态生成的代码，这些部分的错误难以通过Pylint直接捕捉和分析。这导致部分复杂的CWE问题（例如涉及依赖关系的变量未定义、动态类型转换的问题）未能成功处理，生成的分析报告存在缺失。
         2. 在对大型Python代码库进行分析时，Pylint的性能出现瓶颈，导致长时间无法得到结果，尤其是在处理包含大量外部依赖包和复杂逻辑的代码时。
   2. **延期项：**
      1. **文件分析模块**：将文件分析模块中未修复的错误检测部分推迟到下一个Sprint。开发团队将在下一次迭代中集中解决以下问题：
         1. 优化Pylint的性能，以更好地支持大规模代码库的分析。
         2. 引入额外的静态分析工具或手动分析，针对跨模块依赖问题和多层嵌套代码的复杂错误进行处理，确保CWE分析报告的完整性和准确性。
5. 挑战与解决方案：
   1. **可视化界面未完全实现，部分数据展示不够直观**：
      1. **挑战**：在开发用户界面模块的过程中，团队遇到了可视化展示的设计难题，特别是在如何使CWE问题和代码质量评分的展示更加直观和用户友好方面，存在较大困难。
      2. **解决方案**：通过与前端设计师进行多次讨论，并参考了一些成熟的可视化界面设计范例，团队决定改进数据展示方式，引入更清晰的图表类型，如柱状图和饼图，用于展示CWE问题和代码质量评分的概要信息。
   2. **数据库表结构设计不合理，存在冗余数据，影响查询性能**：
      1. **挑战**：在设计数据库模块时，团队发现初始的数据库表结构存在较多冗余数据，导致在进行查询操作时性能较差，尤其在涉及大量数据时，系统响应速度明显变慢。
      2. **解决方案**：团队与系统架构师一同重新审查数据库设计，确定需要简化表结构，去除冗余字段。通过采用规范化的设计，减少数据的重复存储，优化了查询性能。此外，团队还决定在下一个Sprint中进行数据库性能测试，确保进一步优化后的设计能够满足系统需求。
6. 回顾反馈：
   1. **总结冲刺回顾中的反馈**
      1. **沟通与协作的不足**：在Sprint 1中，团队成员之间在系统架构和模块实现的细节上缺乏足够的沟通，导致部分功能在集成时遇到了不一致的问题。尤其是前端与数据库的集成部分，由于沟通不到位，导致了开发中的重复劳动和效率降低。
      2. **测试覆盖不足**：回顾过程中团队还指出，虽然基础功能已经搭建，但在设计和实现过程中缺少对各模块进行充分的单元测试，增加了后续集成时出现问题的风险。
   2. **需要改进的地方**
      1. **加强沟通机制**：建议在下一个Sprint中增加更多的团队沟通会议，例如每周的模块进度讨论，确保团队成员对项目的最新变化保持一致理解，从而减少集成时的冲突和返工。
      2. **提升测试覆盖率**：在未来的开发过程中，团队将更加注重单元测试和集成测试的设计和实现，确保每个功能在实现后都经过充分的测试验证，以减少后续出现的Bug数量，提升整体系统的稳定性。
7. 下个冲刺的行动项：列出基于回顾讨论的任务或变更。
   1. **文件分析模块的优化与完善**
      1. **优化Pylint的性能**：集中精力解决Pylint在处理大型代码库时出现的性能瓶颈，确保在未来对包含大量外部依赖和复杂逻辑的代码进行分析时，提高响应速度。
      2. **完善错误检测**：针对多层嵌套结构、递归调用及动态生成代码中的复杂错误，继续改进文件分析模块。引入其他静态代码分析工具或手动分析方式，以弥补Pylint的不足，确保CWE问题分析的完整性和准确性。
      3. **评分计算模块的开发**：基于代码分析结果，开发评分计算模块，提供更为全面的code quality评分。
   2. **数据库模块的进一步优化**
      1. **数据库性能测试**：对已修改的数据库结构进行性能测试，确保去除冗余数据后，系统查询性能达到预期要求。
      2. **用户记录查询模块的实现**：开发数据库用户记录查询模块，允许用户登录后查看其项目的历史数据和分析结果，确保数据访问的安全性和有效性。
   3. **ChatGPT交互模块的开发**
      1. **CWE问题发送模块**：开发并完善将检测到的CWE问题发送给ChatGPT的模块。确保ChatGPT能提供准确的中文修改建议并反馈至系统界面，提升用户体验。
      2. **处理响应的优化**：优化ChatGPT响应处理模块，确保用户能够便捷地查看修改建议和相应代码版本，增强模块之间的兼容性。
   4. **代码质量改进与测试覆盖率的提升**
      1. **代码质量改进**：根据Sprint 1的反馈，完善文档和代码注释，确保所有模块符合统一的代码风格规范（PEP 8），提升代码的可读性。
      2. **测试覆盖率提升**：
         1. **单元测试和集成测试**：针对各模块（如文件分析模块、用户界面模块、数据库模块）设计并实现单元测试和集成测试，确保功能的稳定性。
         2. 使用测试覆盖工具检查测试覆盖率，优先覆盖关键功能模块，确保测试覆盖率符合项目要求。
   5. **加强沟通与协作机制**
      1. **定期团队会议**：在每周增加一次模块进度讨论会议，确保各成员对项目的变化和进展有一致的理解，减少因沟通不足而导致的集成问题。
      2. **交叉评审机制**：在每次开发任务完成后，增加交叉评审的流程，确保每位开发者都能对其他模块的实现有基本的了解，提升团队整体协作能力。
   6. **用户体验的改进**
      1. **数据可视化界面的改进**：根据Sprint 1中的反馈，进一步优化UI界面，增加更加直观的图表类型（例如柱状图、饼图），提高用户在使用CWE问题和代码质量评分展示功能时的体验。
8. **当前sprint任务概述 (Current Week Story Summary) 圆环图：**



1. **每个Sprint开发任务条形图：**

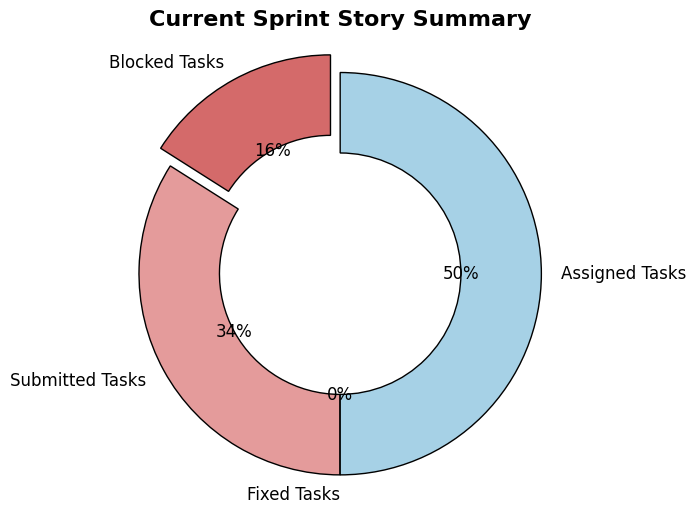


1. Sprint1的燃尽图

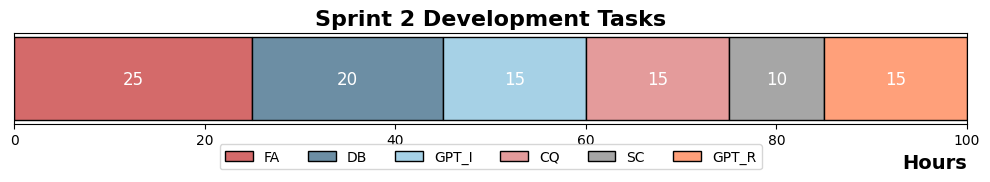


Sprint 2（第6-9周）报告

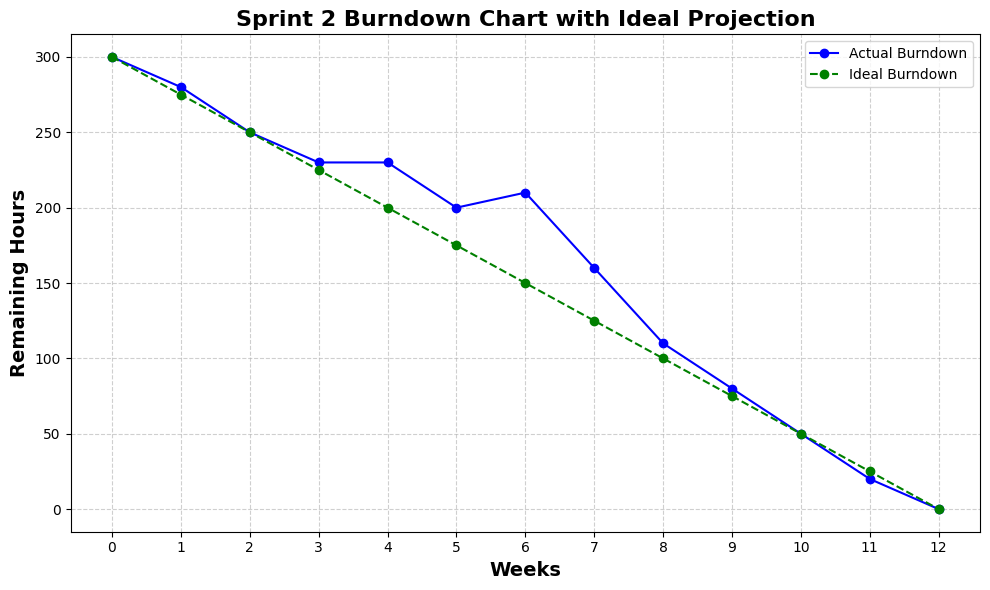
1. Sprint概述
   1. Sprint编号：2024-10
   2. 起始日期：2024.9.30
   3. 结束日期：2024.10.30
   4. 团队人数：8人
   5. 预估工作量：100小时
2. Sprint目标：提升代码质量，完成主要分析与反馈功能。
3. 已完成的工作
   1. 用户故事/任务：
      1. 文件分析模块优化
         1. 描述：完成了Pylint性能的初步优化，特别是在处理包含大量外部依赖和复杂逻辑的代码时，大幅提高了响应速度。
         2. 故事点：20个故事点。
      2. 数据库用户记录查询模块
         1. 描述：实现了数据库用户记录查询模块，用户可以登录系统并查看项目的历史数据和分析结果。
         2. 故事点：15个故事点。
      3. ChatGPT交互模块开发
         1. 描述：完成了CWE问题发送模块，将检测到的CWE问题发送给ChatGPT，并能正确接收中文修改建议。
         2. 故事点：10个故事点。
      4. 代码质量改进
         1. 描述：完善了各模块的文档和代码注释，使代码符合正规的编码规范，提高了代码的可读性。
         2. 故事点：10个故事点。
   2. 验收标准：
      1. **文件分析模块优化**：
         1. Pylint在处理包含复杂逻辑和大量外部依赖的代码时，响应时间减少30%，并能处理嵌套结构和递归调用中的部分复杂错误。
      2. **数据库用户记录查询模块**：
         1. 用户可以登录并成功查询项目历史记录，查询操作经过性能测试，确保响应时间符合预期。
      3. **ChatGPT交互模块**：
         1. CWE问题能够成功发送给ChatGPT，且系统可以正确展示ChatGPT返回的中文修改建议。
      4. **代码质量改进**：
         1. 所有代码均符合规范标准，代码通过静态分析工具的检测且无严重警告。
4. 未完成的工作：
   1. **未完成的项目：**
      1. **评分计算模块的开发**
         1. **描述**：评分计算模块尚未完成，目前仅实现了部分评分逻辑，但无法生成最终的综合评分。延迟原因主要是由于开发过程中对不同模块之间评分的统一标准未能及时确定，导致模块集成进展缓慢。
      2. **ChatGPT响应处理模块优化**
         1. **描述**：ChatGPT交互模块中的响应处理优化部分未能按计划完成，目前用户可以收到建议，但在界面展示和用户交互方面还有所欠缺。延迟原因是由于用户体验优化方面未能充分测试，需进一步改进。
   2. **延期项：**
      1. **评分计算模块**：将评分计算模块的开发推迟到下一个Sprint，以确保各模块评分逻辑的一致性和完整性。
      2. **ChatGPT响应处理优化**：将在下一个Sprint中继续优化ChatGPT响应处理模块，以提升用户体验，确保用户能够方便地查看并应用ChatGPT的修改建议。
5. 挑战与解决方案：
   1. **文件分析模块的性能瓶颈**
      1. **挑战**：在优化文件分析模块时，团队遇到了Pylint在分析包含大量外部依赖和复杂逻辑代码时的性能瓶颈，导致分析速度非常缓慢，影响了开发进度。
      2. **解决方案**：团队通过对Pylint进行性能优化，特别是在依赖解析和代码复杂性分析方面进行了改进。此外，还引入了Flake8作为补充工具，以减轻Pylint的负担，从而显著提升了代码分析效率。
   2. **评分标准的不一致性**
      1. **挑战**：在开发评分计算模块时，遇到了不同模块评分标准的不一致性问题，这使得综合评分逻辑难以统一，影响了模块的集成进展。
      2. **解决方案**：团队在Sprint期间召开了几次专题讨论会议，统一了各模块评分的标准，制定了一致的评分方案。由于时间有限，最终评分计算模块部分内容推迟到下一个Sprint完成。
   3. **ChatGPT交互模块的界面展示问题**
      1. **挑战**：虽然实现了ChatGPT的交互功能，但在界面展示上用户反馈体验不够友好，尤其在处理返回的建议时，用户很难方便地找到并应用相关修改。
      2. **解决方案**：团队对ChatGPT模块进行了部分调整，优化了返回建议的可视化设计，提供了更直观的反馈界面。同时，还计划在下一个Sprint中进行用户体验测试，进一步收集改进建议。
6. 回顾反馈：
   1. **总结冲刺回顾中的反馈**
      1. **自动化测试覆盖率不足**  
         Sprint 3中，虽然团队完成了大部分模块的集成测试，但自动化测试覆盖率仍显不足，尤其是在复杂业务逻辑和系统负载场景的测试方面存在明显的缺口。部分自动化测试工具未能按时开发，影响了整体的系统稳定性验证。
      2. **模块集成的挑战**  
         在模块集成阶段，特别是评分计算模块和ChatGPT响应处理模块的集成，团队遇到了一些不一致的问题。由于这些模块的依赖关系和通信复杂，集成后的行为与预期不符，导致进度延迟，增加了测试和调试的工作量。
      3. **用户体验改进仍需加强**  
         虽然对ChatGPT交互模块的界面做了多次优化，但用户反馈中仍有部分问题未完全解决，尤其在处理多次修改建议时，用户的操作体验还不够顺畅。部分用户反映在面对多项修改建议时，界面的引导不够清晰，理解和应用变得困难。
   2. **需要改进的地方**
      1. **提高团队灵活性**：在未来的开发中，团队需要提高应对不同工作量的灵活性，尤其是当某些模块遇到挑战时，应更及时地调整优先级，确保关键模块不影响整体进度。
      2. **加强模块集成计划**：针对模块集成问题，团队将在下一个Sprint中更加注重各模块的集成时间安排。建议在各模块开发完成后及时进行集成测试，以便尽早发现并解决不一致性问题。
      3. **用户体验改进**：特别是ChatGPT交互模块的用户界面展示部分，建议在下一个Sprint中进行用户体验测试，收集实际用户的反馈并做出进一步优化，使用户更方便地找到修改建议并应用于项目中。
7. 下个冲刺的行动项：列出基于回顾讨论的任务或变更。
   1. **测试与调试**
      1. **集成测试**：对所有模块（文件分析模块、ChatGPT交互模块和用户界面模块）进行全面的集成测试，确保模块之间能够顺利通信。特别注意已延期的评分计算模块和ChatGPT响应处理模块，确保它们能够无缝集成。
      2. **开发自动化测试工具**：建立自动化测试工具，重点验证各模块的功能及系统的整体稳定性，提高测试覆盖率。
   2. **历史追踪与可视化模块开发**
      1. **CWE问题趋势图与评分变化图**：实现CWE问题趋势图模块和评分变化图模块，以帮助用户直观了解项目的安全问题变化和代码质量变化趋势。确保历史追踪模块在功能和用户体验上都符合预期。
   3. **导出与报告生成模块开发**
      1. **数据导出功能**：开发导出模块，允许用户将项目的历史数据和分析报告导出为CSV或PDF格式，便于后续参考和分析。重点确保数据的完整性和可读性。
   4. **用户身份验证与账户管理模块开发**
      1. **用户注册与登录功能**：实现用户身份验证模块，提供注册与登录功能，确保用户能够安全地访问和管理他们的项目记录，保障系统的安全性。
   5. **用户体验改进**
      1. **ChatGPT交互模块的优化**：继续优化ChatGPT响应处理模块的界面展示，进行用户体验测试，以收集实际用户的反馈并进一步优化界面设计。确保用户能够方便地找到修改建议并应用于项目中。
      2. **数据可视化改进**：根据用户反馈，进一步改进UI界面，优化数据展示方式（如柱状图、饼图等），提升用户在使用CWE问题和代码质量评分展示功能时的体验。
   6. **加强沟通与协作机制**
      1. **定期团队会议**：继续每周举行一次模块进度讨论会议，确保团队成员对项目进展和变化有一致的理解，以减少开发中的沟通不足问题。
      2. **交叉评审机制**：加强模块的交叉评审流程，确保每个开发者对其他模块的实现有基本了解，进一步提升团队协作能力。
   7. **项目演示与交付准备**
      1. **创建项目演示视频**：制作系统功能的演示视频，展示各模块的功能及实现过程，帮助项目利益相关者了解项目进展和功能实现。
      2. **项目提交与审核**：整理并提交GitHub项目链接，供外部团队进行审核与测试，确保项目符合交付要求并做好最终发布准备。
8. **当前sprint任务概述 (Current Week Story Summary) 圆环图：**



1. **每个Sprint开发任务条形图：**

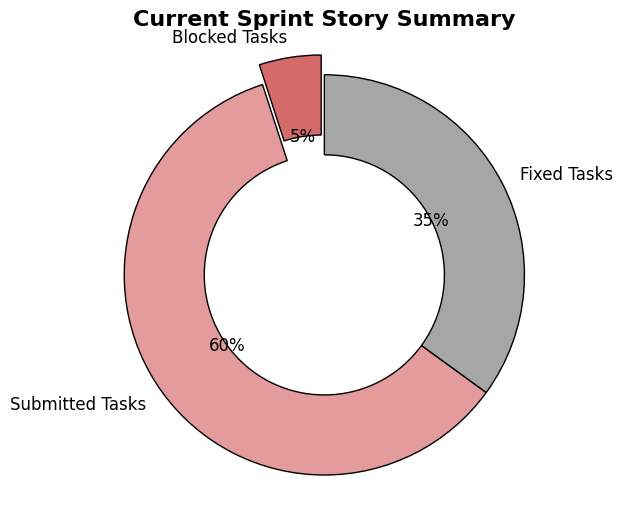


1. Sprint2的燃尽图

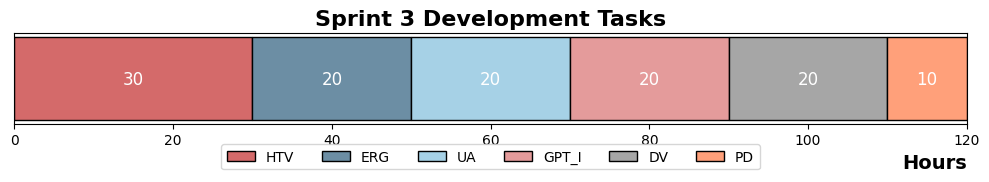


Sprint 3（第10-13周）报告

1. Sprint概述
   1. Sprint编号：2024-11
   2. 起始日期：2024.10.30
   3. 结束日期：2024.11.30
   4. 团队人数：8人
   5. 预估工作量：120小时
2. Sprint目标：**自动化测试、系统整合与最终交付**。
3. 已完成的工作
   1. 用户故事/任务：
      1. **历史追踪与可视化模块开发**
         1. **描述**：实现了CWE问题趋势图模块和评分变化图模块，帮助用户直观了解项目的安全问题变化和代码质量变化趋势。
         2. **故事点**：20个故事点。
      2. **导出与报告生成模块开发**
         1. **描述**：完成了数据导出功能，用户可以将项目历史数据和分析报告导出为CSV或PDF格式，确保了数据的完整性和可读性。
         2. **故事点**：15个故事点。
      3. **用户身份验证与账户管理模块开发**
         1. **描述**：实现了用户注册与登录模块，用户可以安全地访问和管理他们的项目记录，确保系统的安全性。
         2. **故事点**：15个故事点。
      4. **ChatGPT交互模块的用户体验改进**
         1. **描述**：优化了ChatGPT响应处理模块的界面展示，并进行了用户体验测试，以收集实际用户的反馈。确保用户能够方便地找到修改建议并应用于项目中。
         2. **故事点**：10个故事点。
      5. **数据可视化改进**
         1. **描述**：根据用户反馈，对UI界面进行了进一步优化，改进了数据展示方式（如柱状图、饼图等），提升用户在使用CWE问题和代码质量评分展示功能时的体验。
         2. **故事点**：10个故事点。
      6. **项目演示与交付准备**
         1. **描述**：创建了项目演示视频，展示各模块的功能及实现过程。同时整理并提交了GitHub项目链接，供外部团队进行审核与测试。
         2. **故事点**：10个故事点。
   2. 验收标准：
      1. **历史追踪与可视化模块**：
         1. CWE问题趋势图和评分变化图能够正确展示历史数据，确保用户可以直观了解项目的安全性和代码质量变化。
      2. **导出与报告生成模块**：
         1. 项目历史数据和分析报告可以成功导出为CSV或PDF格式，且导出的文件内容完整且可读。
      3. **用户身份验证与账户管理模块**：
         1. 用户可以成功注册、登录并访问他们的项目记录，确保数据访问的安全性。
      4. **ChatGPT交互模块用户体验改进**：
         1. ChatGPT的修改建议能够便捷地展示给用户，界面改进符合用户反馈的建议，用户可以方便地应用这些建议。
      5. **数据可视化改进**：
         1. UI界面经过改进，数据展示方式更加清晰直观，提高了用户的使用体验。
      6. **项目演示与交付准备**：
         1. 演示视频完整展示了系统功能，GitHub项目提交后获得了外部团队的审核反馈，并通过了审核要求。
4. 未完成的工作：
   1. **未完成的项目：**
      1. **自动化测试与调试**
         1. **描述**：自动化测试工具的开发尚未完成。虽然集成测试已进行，但部分模块的自动化测试还需完善。未能完成的主要原因是由于开发团队在多模块集成过程中遇到了复杂的依赖关系，导致测试脚本开发滞后。
   2. **延期项：**
      1. **自动化测试与调试**：
         1. **集成测试**：部分集成测试和自动化测试工具的开发将推迟到下一个阶段进行，以确保测试覆盖率和系统的稳定性。
5. 挑战与解决方案：
   1. **自动化测试与集成的复杂性**
      1. **挑战**：在开发自动化测试工具和进行集成测试的过程中，团队遇到了模块之间复杂依赖关系的问题，特别是不同模块的通信方式和数据流动需要精确调整，导致测试脚本难以开发并确保稳定性。这种复杂性影响了自动化测试工具的进展和集成测试的效果。
      2. **解决方案**：为了应对这一挑战，团队采用了分阶段集成的策略，即首先在较小的子模块中进行集成测试，然后逐步扩展到完整系统的集成。此外，引入了模拟工具和服务接口来减少对真实数据的依赖，从而降低了测试环境的复杂度。尽管这些措施帮助团队部分缓解了集成问题，但仍需要在未来继续优化自动化测试工具。
   2. **用户体验的进一步优化**
      1. **挑战**：在优化ChatGPT交互模块的界面设计时，用户反馈显示一些建议在展示时缺乏清晰的上下文，用户仍需花费时间去理解和应用修改。特别是对于多次修改后的代码段，用户感到难以追踪每个建议的关联。
      2. **解决方案**：团队改进了界面展示的方式，通过添加详细注释和提供逐步引导等方式，使得用户在阅读和应用修改建议时更加直观。团队还进行了多次用户体验测试，并根据用户反馈调整了界面的细节。
   3. **跨模块协调与沟通**
      1. **挑战**：由于多个模块在最后的集成阶段需要实现无缝通信，团队在处理跨模块的依赖和接口对接时，出现了一些沟通不足的问题，导致部分功能在集成时出现不一致性，进而影响了测试进度。
      2. **解决方案**：团队通过增加团队内部的定期沟通会议和交叉评审流程来解决这一问题。每个模块在开发完成后，由其他模块的开发人员参与评审，确保对接口和依赖有一致的理解和实现方式，从而降低了集成时的风险。
6. 回顾反馈：
   1. **总结冲刺回顾中的反馈**
      1. **自动化测试覆盖率不足**  
         Sprint 3中，虽然团队完成了大部分模块的集成测试，但自动化测试覆盖率仍显不足，尤其是在复杂业务逻辑和系统负载场景的测试方面存在明显的缺口。部分自动化测试工具未能按时开发，影响了整体的系统稳定性验证。
      2. **模块集成的挑战**  
         在模块集成阶段，特别是评分计算模块和ChatGPT响应处理模块的集成，团队遇到了一些不一致的问题。由于这些模块的依赖关系和通信复杂，集成后的行为与预期不符，导致进度延迟，增加了测试和调试的工作量。
      3. **用户体验改进仍需加强**  
         虽然对ChatGPT交互模块的界面做了多次优化，但用户反馈中仍有部分问题未完全解决，尤其在处理多次修改建议时，用户的操作体验还不够顺畅。部分用户反映在面对多项修改建议时，界面的引导不够清晰，理解和应用变得困难。
   2. **需要改进的地方**
      1. **提高自动化测试的覆盖率**  
         未来的开发中，团队需要进一步细化测试计划，增加对复杂逻辑和场景的自动化测试覆盖率。尤其需要确保集成阶段的自动化测试工具开发按时完成，并通过更多自动化工具来减少手动测试的工作量和可能的错误。
      2. **加强模块集成的提前规划**  
         模块集成应当更加提前规划并合理安排。在每个模块完成开发后立即进行小范围的集成测试，以便及时发现并解决模块之间的兼容性问题，减少最终集成阶段的返工和调试压力。同时，建议在集成计划中考虑详细的接口依赖和测试环境准备，以提高集成的顺利程度。
      3. **用户体验的持续优化**  
         未来需要继续收集用户的反馈，并持续优化ChatGPT交互模块和其他用户界面模块。可以通过更细致的用户指导来改善操作体验，例如提供更加明显的逐步引导和对比视图，帮助用户理解每项修改建议。同时，应在优化过程中增加实际用户的使用场景模拟，以确保系统的设计更符合用户的实际需求和操作习惯。
7. 项目整体有待改进和优化的地方
   1. **提高自动化测试的覆盖率与效率**：虽然在Sprint 3中完成了集成测试，但自动化测试工具的覆盖率仍不足，尤其是在复杂逻辑和负载测试方面。未来需要进一步开发和优化自动化测试工具，确保所有模块及其集成部分都经过充分的验证。可以考虑采用更多先进的自动化测试框架，以提高测试效率，减少手动测试的工作量。
   2. **增强用户体验和界面引导**：ChatGPT交互模块虽然经过多次优化，但在多次建议堆叠和复杂修改的情况下，用户操作体验仍有提升空间。未来应考虑增加逐步引导功能和详细的提示，使用户更容易理解和应用修改建议。同时，在数据可视化方面，界面还可以更加直观，例如通过动态图表和交互式视图来提升用户的理解和参与度。
   3. **数据可视化与分析的进一步丰富**：当前的历史追踪与可视化模块提供了基本的CWE问题趋势图和评分变化图，但可以进一步丰富图表类型，例如加入更详细的关联分析和预测模型，以帮助用户更好地理解项目中的潜在问题和未来风险。此外，可以加入对用户特定行为的分析功能，以便为用户提供个性化的改进建议。
   4. **优化模块之间的集成与通信**：在未来的开发中，需要加强模块之间的集成计划。在每个模块完成开发后，立即进行小范围的集成测试，以减少在最终系统集成时遇到的依赖性问题。建议采用持续集成（CI）工具，将各模块的集成自动化，及时发现集成时的不兼容性，并且提高系统的整体稳定性。
   5. **用户身份验证的安全性增强**：当前用户身份验证模块支持基本的注册与登录，但为了提高系统的安全性，可以引入多因素身份验证（MFA）功能，进一步保障用户账户的安全。考虑使用短信验证码、邮件认证或基于移动设备的身份验证应用等方式来增强系统的安全性。
   6. **增加对大规模项目的支持与优化**：对于处理包含大量外部依赖和复杂代码的项目，目前的性能优化虽已提升了处理效率，但未来可以进一步优化文件分析模块和数据库模块的性能，例如采用分布式处理架构或缓存机制来加速数据存取和分析的速度。此外，系统应支持更多并发用户的操作，保证在大规模用户访问时的系统稳定性。
8. **当前sprint任务概述 (Current Week Story Summary) 圆环图：**



1. **每个Sprint开发任务条形图：**



**HTV (History Tracking & Visualization Module)**

**ERG (Export & Report Generation Module)**

**UA (User Authentication & Account Management Module)**

**GPT\_I (ChatGPT Interaction Module Improvement)**

**DV (Data Visualization Improvement)**

**PD (Project Demo & Delivery Preparation)**

1. Sprint3的燃尽图

