### 需求分析报告

### 项目目标

1. 自动化Rust项目的构建和测试流程，确保代码质量和稳定性。
2. 支持多平台构建，包括Windows、macOS和Linux。
3. 在代码推送到特定分支（modification）时触发自动化流程。
4. 提供手动触发自动化流程的能力。

### 功能需求

1. **代码检出**：从GitHub仓库检出代码，支持大文件存储（LFS）。
2. **依赖缓存**：缓存Rust项目的依赖，加快后续构建速度。
3. **环境变量设置**：设置环境变量，如CARGO\_TERM\_COLOR，以优化构建输出。
4. **版本信息提取**：从Git历史中提取版本信息，用于版本标记。
5. **工具链设置**：安装并使用Rust的nightly版本进行构建和测试。
6. **版本号设置**：在Cargo.toml中设置基于Git版本信息的版本号。
7. **构建**：对项目进行构建，生成可执行文件。
8. **重命名输出**：根据Git版本信息重命名构建输出，以便区分。
9. **上传构建产物**：将构建好的可执行文件上传为GitHub Actions的工件。

### 性能需求

1. **构建速度**：通过依赖缓存机制减少构建时间，提高开发效率。
2. **多平台支持**：确保在Windows、macOS和Linux平台上均能顺利构建和测试。

### 可维护性需求

1. **代码和流程的可读性**：通过清晰的注释和结构化的配置，确保其他开发者能够容易理解和维护这个自动化流程。
2. **易于配置**：通过参数化和环境变量，使得调整构建流程（如更换Rust工具链版本）变得简单。
3. **扩展性**：设计时考虑未来可能的需求变化，如添加新的构建任务或支持更多的操作系统平台。