Google Gemini Deep Research

Given the Software Engineering issues 1 to 10 in the following repository, analyze each of them and give a thorough description of what is required of each, also give all the requirements for each in point format. Finally, give a score out of 10 for the complexity of each and an estimated number of minutes to solve each:

https://github.com/openai/SWELancer-Benchmark/tree/main/issues

Analysis of Initial Software Engineering Issues in the SWELancer-Benchmark Repository

This report provides a detailed analysis of the first ten software engineering issues identified within the openai/SWELancer-Benchmark repository, as requested. Each issue has been thoroughly examined to understand the reported problem, delineate the necessary requirements for resolution, assess its complexity, and estimate the time needed for a solution.

The initial observation from the provided information indicates a potential underlying problem with the repository's issue tracking system. Specifically, issue #1, as mentioned in the provided data, points to an inaccessible website [1]. While the user query focuses on issues 1 through 10, the outline also references issue #62, which similarly reports an inaccessible website [2]. This recurring inaccessibility suggests a need to investigate the platform or process used for hosting and managing these issue pages to ensure reliable access for users seeking information or reporting problems.

Analysis of Issue 1: "Could you explain issue_introduction.patch?"

This issue indicates a user's need for clarification regarding the purpose and structure of the issue_introduction.patch file. The user's question implies that the documentation or existing context does not adequately explain this component of the benchmark, leading to confusion about its role and format.

Requirements:

- Develop comprehensive documentation that clearly explains the issue_introduction.patch file.

- Specify the expected format and content of this file, including any specific syntax or conventions.
- Articulate the role and purpose of this file within the overall benchmarking process.
- Include illustrative examples of valid issue_introduction.patch files to aid user understanding.

Complexity Score: 3/10 (Primarily a documentation task).

Estimated Time to Solve: 60 minutes.

The fact that a user is seeking basic information about a file named issue_introduction.patch suggests that this file is fundamental to interacting with the benchmark. Addressing this through clear documentation will likely improve the user experience and reduce the number of similar queries in the future.

Analysis of Issue 2: "Can the model see the issue's GitHub discussion thread?"

This issue raises a question about the scope of information accessible to the language model being evaluated by the benchmark. The user is specifically asking whether the model can access and utilize the discussion thread associated with a GitHub issue. This inquiry suggests an interest in understanding the context provided to the model during the benchmarking process.

Requirements:

- Clearly define and document the information sources that the benchmarked model has access to.
- If the intention is for the model to access GitHub discussion threads, the benchmark infrastructure must be configured to facilitate this.
- If the model is not intended to access discussion threads, this limitation should be explicitly stated in the benchmark's documentation.
- Consider the potential impact of including or excluding discussion thread context on the evaluation of the model's software engineering capabilities.

Complexity Score: 5/10 (Depends on the intended functionality and potential modifications to the benchmark setup).

Estimated Time to Solve: 90 minutes.

Understanding the data provided to the model is crucial for interpreting the benchmark results. Clarifying whether discussion threads are part of the model's context will help users understand the evaluation criteria and the capabilities being assessed.

Analysis of Issue 3: "Invalid issue_introduction.patch

files"

This issue reports a functional problem where users are encountering invalid issue_introduction.patch files. This directly relates to the previous issue regarding the explanation of this file and suggests potential problems with either the creation of these files or the validation process within the benchmark [3].

Requirements:

- Investigate the root cause of the invalid issue_introduction.patch file errors.
- Implement robust validation mechanisms to ensure the integrity and correctness of these files.
- Provide informative error messages to users when an invalid file is detected, guiding them on how to rectify the issue.
- Review and update the documentation for issue_introduction.patch files to ensure accuracy and completeness, as highlighted in the analysis of Issue 1.

Complexity Score: 7/10 (Involves debugging and implementing validation logic).

Estimated Time to Solve: 180 minutes.

Encountering invalid input files can be a significant barrier to using the benchmark. Addressing this issue by improving validation and providing clear guidance will enhance the user experience and the reliability of the benchmark.

Analysis of Issue 4: "Can anyone find the manager tasks?"

This issue indicates a user's difficulty in locating "manager tasks" within the SWELancer-Benchmark. This suggests a lack of clarity regarding what these tasks are, where they are located within the repository or benchmark setup, or whether they are currently available in the open-source version [4]. The README file encourages users to open issues for problems or questions [4], which is what the user has done here.

Requirements:

- Clearly define what constitutes "manager tasks" within the context of the SWELancer-Benchmark.
- If these tasks are part of the benchmark, provide explicit instructions on how to access or identify them within the repository or execution environment.
- If "manager tasks" are not currently available in the open-source version or are exclusive to an internal setup, this should be clearly stated in the documentation to manage user expectations.
- Evaluate whether these tasks should be included in the open-source version and, if so, plan for their integration and documentation.

Complexity Score: 4/10 (Primarily requires clarification and documentation).

Estimated Time to Solve: 75 minutes.

The term "manager tasks" might originate from associated research papers or internal documentation. Ensuring consistency between these sources and the open-source repository is important for user understanding and engagement.

Analysis of Issue 5: "uv run python run_swelancer.py error"

This issue reports an error encountered when attempting to execute the main benchmark script, run_swelancer.py, using the uv tool [3]. This indicates a problem with the execution environment setup or potentially with the script itself, preventing users from running the benchmark.

Requirements:

- Reproduce the reported error in a controlled environment to understand the specific circumstances under which it occurs.
- Debug the run_swelancer.py script to identify the root cause of the error.
- Develop a solution, which might involve modifying the script, updating the environment setup instructions, or addressing dependencies.
- Ensure the benchmark documentation provides clear and accurate instructions on the required environment setup and the correct way to execute the run_swelancer.py script, including the use of the uv tool.

Complexity Score: 8/10 (Debugging runtime errors can be complex).

Estimated Time to Solve: 240 minutes.

An error in the primary execution script is a critical issue that directly impacts the usability of the benchmark. Resolving this will likely unblock many users and is essential for wider adoption.

Analysis of Issue 6: "Inconsistent Code Output Format Requirements in Prompts"

This issue highlights a potential flaw in the design of the benchmark prompts. The user reports inconsistencies in the expected format of the code output across different prompts. This can lead to confusion for the model being evaluated and potentially result in inaccurate scoring or assessment of its capabilities.

Requirements:

- Conduct a thorough review of all prompts used within the SWELancer-Benchmark.
- Identify and document the specific code output format requirements for each prompt.
- Standardize the output format requirements across prompts where it is logically feasible and beneficial for the evaluation.
- Ensure that the expected output format is clearly communicated within the prompt itself (if

possible) and is also documented in the benchmark's guidelines or documentation.

Complexity Score: 6/10 (Requires careful review and potential modification of multiple prompts).

Estimated Time to Solve: 150 minutes.

Consistency in the expected output format is crucial for a fair and reliable benchmark. Addressing this issue will improve the quality of the evaluation and reduce potential misinterpretations of the results.

Analysis of Issue 7: "Fail to run user-tool in docker"

This issue indicates a problem with running a specific "user-tool" within the provided Docker environment. This suggests that the benchmark relies on this tool for some part of the evaluation process, and users are encountering difficulties in its execution within the Docker container. This is related to other Docker-related issues, indicating a potential area for improvement in the benchmark's containerization [3].

Requirements:

- Identify the "user-tool" in question and understand its intended functionality within the benchmark.
- Investigate the reasons why the user-tool is failing to run within the Docker environment. This could involve examining configuration issues, missing dependencies within the Docker image, or incorrect usage instructions.
- Update the Docker setup (Dockerfile, configurations) or the benchmark documentation to ensure the user-tool can be executed successfully within the container.
- Provide clear and concise instructions on how to use the user-tool within the Docker environment, including any necessary setup or command-line arguments.

Complexity Score: 7/10 (Debugging Docker-related issues and understanding the user-tool's requirements can be complex).

Estimated Time to Solve: 200 minutes.

Docker is often used to provide a consistent and reproducible environment. Ensuring that all necessary components, including user-tools, function correctly within the Docker container is essential for a positive user experience.

Analysis of Issue 8: "Has Anyone Successfully Run the SWE-Lancer Benchmark?"

This issue is a broad question that reflects a significant concern about the overall usability and accessibility of the SWELancer-Benchmark [3]. The fact that a user is asking this question publicly suggests that they, and potentially others, are facing substantial challenges in setting up and running the benchmark successfully. This could be a culmination of the issues already

discussed, such as setup problems, execution errors, and unclear documentation.

Requirements:

- Address the underlying issues that are preventing users from successfully running the benchmark, as identified in the analyses of other issues (e.g., Issues 5, 7, 9, and 10).
- Improve the overall user experience by providing comprehensive and easy-to-follow documentation, including detailed setup instructions, troubleshooting guides for common problems, and illustrative examples.
- Consider establishing a community forum or a dedicated communication channel where users can ask questions, share their experiences, and receive support.
- Explore the possibility of creating a quick-start guide or a simplified setup process to lower the barrier to entry for new users.

Complexity Score: 9/10 (This is a meta-issue that requires addressing multiple underlying problems and focusing on overall usability).

Estimated Time to Solve: Ongoing (Improving usability is a continuous effort).

This issue highlights a critical need to focus on the user experience. If users are struggling to even run the benchmark, it limits its value and potential impact. Addressing the root causes and providing better support and documentation are crucial.

Analysis of Issue 9: "Docker Run Failed"

This issue reports a general failure when attempting to run the SWELancer-Benchmark using Docker [3]. Similar to Issue 7, this indicates a problem with the Docker setup or the process of running the Docker image. Without specific error messages, it is difficult to pinpoint the exact cause, but it suggests a fundamental issue with the containerization of the benchmark.

Requirements:

- Investigate the potential causes of the Docker run failure. This would ideally involve examining the specific error messages or logs encountered by the user.
- Ensure that the Dockerfile is correctly configured and that the Docker image builds without any errors.
- Verify that all necessary dependencies, configurations, and environment variables are correctly set up within the Docker image.
- Provide clear and detailed instructions in the documentation on how to run the Docker image, including any prerequisites, required commands, or environment configurations.
- Consider providing example docker run commands or a docker-compose.yml file for easier setup.

Complexity Score: 7/10 (Debugging Docker run failures can vary in complexity depending on the specific error).

Estimated Time to Solve: 180 minutes.

A failing Docker setup significantly hinders the accessibility and reproducibility of the

benchmark. Ensuring a smooth Docker experience is vital for users who prefer or require containerized environments.

Analysis of Issue 10: "Docker network error when running uv run python run_swelancer.py"

This issue reports a specific type of error – a Docker network error – occurring when running the main benchmark script using uv within the Docker environment [3]. This suggests a problem with how the Docker container is configured to access network resources, which might be required by the run_swelancer.py script, the uv tool, or some underlying dependencies.

Requirements:

- Determine if the run_swelancer.py script or the uv tool requires network access when run within the Docker container.
- Review the Docker network configuration to ensure it allows the necessary network connections. This might involve checking port mappings, network modes, or any custom network configurations.
- Investigate whether any firewall settings on the host machine are interfering with the Docker container's network access.
- Document any specific network requirements or configurations that are necessary to run the benchmark successfully within the Docker environment.

Complexity Score: 8/10 (Network-related issues in Docker can be challenging to diagnose and resolve).

Estimated Time to Solve: 210 minutes.

Network errors within Docker can be particularly frustrating to debug. Clearly understanding the network dependencies of the benchmark and providing the correct Docker network configurations are crucial for resolving this issue.

Summary of Complexity and Estimated Time

The following table summarizes the complexity score and estimated time to solve for each of the ten issues analyzed:

Issue Number	Title	Complexity Score (out of 10)	Estimated Time to Solve (minutes)
1	Could you explain	3	60

	issue_introduction.patch?		
2	Can the model see the issue's GitHub discussion thread?	5	90
3	Invalid issue_introduction.patch files	7	180
4	Can anyone find the manager tasks?	4	75
5	uv run python run_swelancer.py error	8	240
6	Inconsistent Code Output Format Requirements in Prompts	6	150
7	Fail to run user-tool in docker	7	200
8	Has Anyone Successfully Run the SWE-Lancer Benchmark?	9	Ongoing
9	Docker Run Failed	7	180
10	Docker network error when running uv run python run_swelancer.py	8	210

This summary highlights that a significant portion of the initial user-reported issues are related to setting up and running the benchmark, particularly within a Docker environment. Issues 5, 7, 9, and 10, which all concern execution or Docker, have higher complexity scores and estimated resolution times compared to documentation-related queries like Issue 1 and 4. Issue 8 stands

out as a critical indicator of broader usability challenges.

Conclusions

The analysis of the first ten issues in the SWELancer-Benchmark repository reveals several key areas requiring attention. A recurring theme is the difficulty users are experiencing in setting up and running the benchmark, especially within the provided Docker environment. Issues related to Docker failures, network errors within Docker, and general execution errors suggest a need to thoroughly review and improve the containerization strategy and the overall execution process.

Furthermore, the questions raised about the issue_introduction.patch file and the "manager tasks" indicate gaps in the current documentation, leading to user confusion. Addressing these with clear and comprehensive documentation is crucial for improving the user experience and enabling users to effectively interact with the benchmark.

The report of inconsistent code output format requirements in prompts points to a potential issue in the benchmark's design that could affect the accuracy and fairness of the evaluation. Standardizing these requirements and clearly communicating them to the model and the users is essential for the benchmark's integrity.

Finally, the overarching question of whether anyone has successfully run the benchmark underscores the need for a holistic approach to improving usability. This includes not only fixing the identified bugs and improving documentation but also potentially providing better support channels for users. Addressing these initial hurdles will be critical for the successful adoption and utilization of the SWELancer-Benchmark by the wider community.

Works cited

1. accessed January 1, 1970, https://github.com/openai/SWELancer-Benchmark/tree/main/issues/1
2. accessed January 1, 1970, https://github.com/openai/SWELancer-Benchmark/issues/62
3. Issues · openai/SWELancer-Benchmark · GitHub, accessed March 18, 2025, https://github.com/openai/SWELancer-Benchmark/issues
4. openai/SWELancer-Benchmark: This repo contains the ... - GitHub, accessed March 18, 2025, https://github.com/openai/SWELancer-Benchmark