Grok 3 Deep Search

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

Key Points

- It seems likely that the first 10 issues in the SWELancer-Benchmark GitHub repository are software engineering tasks, based on the analysis of available GitHub issues.
- The tasks vary from fixing build errors to code cleanup and dataset management, with complexity ranging from 2 to 6 out of 10.
- Estimated time to solve each task ranges from 10 minutes to 1 hour, depending on the task's nature.
- Some issues, like server specs and feature requests, may not be typical coding tasks, adding complexity to the analysis.

Analysis of Issues

Below is a detailed breakdown of each issue from 1 to 10, including what is required, requirements in point format, complexity score, and estimated time to solve.

Issue 1: Server Specs and Auto Deploy Config
- **Description**: Determine server specifications for scaling and check for auto deploy configuration.
- **Requirements**:
  - Research server requirements for the benchmark.
  - Determine hardware and software specs needed.
  - Check for existing auto deploy config; suggest or implement if absent.
- **Complexity Score**: 3/10
- **Estimated Time**: 30 minutes

Issue 2: Fixing Dockerfile Build Error on x86
- **Description**: Resolve a gem installation error in Dockerfile on x86 architecture.
- **Requirements**:
  - Identify the gem installation error.

  - Find missing libraries or headers.
  - Modify Dockerfile to include necessary packages.
- **Complexity Score**: 6/10
- **Estimated Time**: 1 hour

Issue 3: Fixing Issue 2
- **Description**: Implement the fix for the Dockerfile build error from Issue 2.
- **Requirements**: Same as Issue 2.
- **Complexity Score**: 6/10
- **Estimated Time**: 1 hour

Issue 4: Feature Request for AI Earnings Project
- **Description**: Evaluate a proposal for AI to earn money at a constant speed.
- **Requirements**:
  - Assess proposal feasibility and relevance.
  - Provide feedback or decide next steps.
- **Complexity Score**: 2/10
- **Estimated Time**: 15 minutes

Issue 5: Fixing Build Error on Mac M1-Max
- **Description**: Resolve a GPG keyserver error during RVM installation on Mac M1-Max.
- **Requirements**:
  - Investigate the GPG keyserver error.
  - Ensure keyserver accessibility or find alternative.
  - Modify Dockerfile or script to fix the issue.
- **Complexity Score**: 5/10
- **Estimated Time**: 45 minutes

Issue 6: Code Cleanup in run_swelancer.py
- **Description**: Perform minor cleanup, removing unused imports in run_swelancer.py.
- **Requirements**:
  - Review run_swelancer.py.
  - Remove unused imports and clean code.
  - Ensure functionality remains intact.
- **Complexity Score**: 3/10
- **Estimated Time**: 20 minutes

Issue 7: Releasing Dataset on Hugging Face
- **Description**: Host SWE-Lancer Diamond dataset on Hugging Face for visibility.
- **Requirements**:
  - Prepare dataset in suitable format.
  - Upload to Hugging Face.
  - Update documentation or links.
- **Complexity Score**: 4/10

- **Estimated Time**: 30 minutes

Issue 8: Linking Actual Upwork Tasks
- **Description**: Suggest attaching actual Upwork tasks to benchmark tasks.
- **Requirements**:
  - Evaluate feasibility of linking tasks.
  - Decide implementation method.
  - Update dataset or documentation.
- **Complexity Score**: 2/10
- **Estimated Time**: 10 minutes

Issue 9: Fixing Setup Instructions
- **Description**: Correct issues in setup instructions, including timeouts and typos.
- **Requirements**:
  - Identify and correct typos.
  - Adjust timeouts or configurations.
  - Ensure consistent container names.
- **Complexity Score**: 5/10
- **Estimated Time**: 45 minutes

Issue 10: Minor Fix in run_swelancer.py
- **Description**: Modify run_swelancer.py to pass taskset from args if provided.
- **Requirements**:
  - Modify run_swelancer.py for taskset argument.
  - Test changes for expected functionality.
- **Complexity Score**: 3/10
- **Estimated Time**: 20 minutes

Comprehensive Analysis Report

This report provides a detailed examination of the first 10 issues in the SWELancer-Benchmark GitHub repository, identified as software engineering tasks. The analysis includes a thorough description of each issue, requirements in point format, complexity scores out of 10, and estimated times to solve, based on the available GitHub issue descriptions. The current date is Tuesday, March 18, 2025, and all analysis is conducted within this context.

Background and Context

The SWELancer-Benchmark repository, located at [SWELancer-Benchmark GitHub](https://github.com/openai/SWELancer-Benchmark), contains the dataset and code for the paper "SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?" published in 2025. The paper introduces a benchmark of over 1,400

freelance software engineering tasks from Upwork, valued at $1 million USD, aimed at evaluating large language models (LLMs) in real-world settings. The repository includes branches like `main` for the latest version and `paper` for the version at paper release, with contacts available at samuelgm@openai.com and michele@openai.com for questions or contributions.

The analysis initially faced challenges in locating the exact task descriptions, as the "issues" directory in the repository appeared empty, and the dataset's public evaluation split, SWE-Lancer Diamond, was not directly accessible in file form. After exploring, it was determined that the first 10 GitHub issues (issues 1 through 10) likely represent the software engineering tasks in question, based on their descriptions and relevance to software engineering activities.

Detailed Analysis of Each Issue

Below is a table summarizing the issues, followed by detailed descriptions for each:

Issue Number	Description Summary	Complexity Score	Estimated Time
1	Server specs and auto deploy config question	3/10	30 minutes
2	Fix Dockerfile build error on x86	6/10	1 hour
3	Fix for Issue 2	6/10	1 hour
4	Feature request for AI earnings project	2/10	15 minutes
5	Fix build error on Mac M1-Max	5/10	45 minutes
6	Code cleanup in run_swelancer.py	3/10	20 minutes
7	Release dataset on Hugging Face	4/10	30 minutes
8	Suggestion to link actual Upwork tasks	2/10	10 minutes
9	Fix setup instructions issues	5/10	45 minutes
10	Minor fix in run_swelancer.py for taskset args	3/10	20 minutes

Issue 1: Server Specs and Auto Deploy Config
- **Description**: This issue, found at [issue 1](https://github.com/openai/SWELancer-Benchmark/issues/1), is a question asking for server specifications to run the benchmark at scale and whether there is an auto deploy configuration. It is more of an information-gathering task rather than a coding task.
- **Requirements**:
  - Research the server requirements for running the benchmark.
  - Determine necessary hardware (e.g., CPU, RAM, storage) and software specifications (e.g., OS, dependencies).
  - Check if an auto deploy configuration exists within the repository, such as scripts or CI/CD pipelines.
  - If no auto deploy config exists, suggest or implement a basic configuration, potentially using tools like GitHub Actions or Docker Compose.

- **Complexity Score**: 3/10, as it involves research and possibly light implementation, but no deep coding is required.
- **Estimated Time**: 30 minutes, given the need for research and documentation review.

Issue 2: Fixing Dockerfile Build Error on x86
- **Description**: This issue, at [issue 2](https://github.com/openai/SWELancer-Benchmark/issues/2), reports a build failure on x86 architecture during gem installation (pusher-fake and eventmachine), with errors indicating missing libraries or headers. The resolution involved adding "clang" to the Dockerfile_x86, as seen in the commit history.
- **Requirements**:
  - Identify the specific error, which is an ERROR: Failed to build gem native extension for pusher-fake, likely due to missing libraries.
  - Determine which libraries or headers (e.g., development headers for Ruby gems) are missing, possibly by checking mkmf.log.
  - Modify the Dockerfile_x86 to include necessary packages, such as adding "clang" to the apt-get install command, as done in the resolution.
- **Complexity Score**: 6/10, requiring knowledge of Dockerfile, gem installation, and debugging build errors.
- **Estimated Time**: 1 hour, considering the need to debug and test the build process.

Issue 3: Fixing Issue 2
- **Description**: This issue, at [issue 3](https://github.com/openai/SWELancer-Benchmark/issues/3), is explicitly about fixing the problem from Issue 2, which is the Dockerfile build error on x86.
- **Requirements**: Identical to Issue 2, as it involves implementing the same fix.
- **Complexity Score**: 6/10, same as Issue 2, due to the technical nature of Dockerfile modifications.
- **Estimated Time**: 1 hour, reflecting the effort needed for implementation and testing.

Issue 4: Feature Request for AI Earnings Project
- **Description**: This issue, at [issue 4](https://github.com/openai/SWELancer-Benchmark/issues/4), is a proposal by James Brown for a project called Cybergod, aiming to train AI to earn money at a constant speed through five phases, including agent pre-training and reinforcement learning. It includes links to GitHub and a paper, suggesting a managerial or strategic task rather than a coding one.
- **Requirements**:
  - Evaluate the proposal's feasibility within the context of SWELancer-Benchmark, considering its focus on freelance tasks.
  - Determine relevance to current project goals, possibly by reviewing the paper at [Cybergod Paper](https://james4ever0.github.io/Cybergod__God_is_in_your_computer.html) and GitHub at [Cybergod Project](https://github.com/James4Ever0/agi_computer_control).
  - Provide feedback or decide on next steps, such as whether to integrate or reject the idea, potentially via email to samuelgm@openai.com or michele@openai.com.

- **Complexity Score**: 2/10, as it is more about evaluation and decision-making, not technical implementation.
- **Estimated Time**: 15 minutes, given the need for a quick review and response.

Issue 5: Fixing Build Error on Mac M1-Max
- **Description**: This issue, at [issue 5](https://github.com/openai/SWELancer-Benchmark/issues/5), reports a build failure on Mac M1-Max during the RVM installation step, with a GPG keyserver error ("gpg: keyserver receive failed: End of file"). It suggests a Docker build issue specific to the architecture.
- **Requirements**:
  - Investigate the GPG keyserver error, which occurs at step [12/42] in the build process, possibly due to network issues or keyserver unavailability.
  - Ensure the keyserver (hkp://keyserver.ubuntu.com) is accessible, or find an alternative method, such as using a different keyserver or caching keys.
  - Modify the Dockerfile or build script (e.g., Dockerfile) to resolve the issue, potentially by adding a fallback or adjusting the command.
- **Complexity Score**: 5/10, requiring understanding of GPG, Docker, and Mac-specific build environments.
- **Estimated Time**: 45 minutes, considering debugging and testing on the M1-Max.

Issue 6: Code Cleanup in run_swelancer.py
- **Description**: This issue, at [commit URL](https://github.com/openai/SWELancer-Benchmark/commit/03fc70f08e0a3925ba6cb6b84d6055527a566793) and [pull request](https://github.com/openai/SWELancer-Benchmark/pull/6), involves minor cleanup in run_swelancer.py, such as removing unused imports and other small improvements, noted as the benchmark working great so far.
- **Requirements**:
  - Review the file run_swelancer.py, located in the repository root, to identify unused imports and other minor issues.
  - Remove unused imports and clean up code, ensuring readability and maintainability.
  - Test the script to ensure no functionality is broken, possibly by running it with `uv run python run_swelancer.py` as per the README.
- **Complexity Score**: 3/10, a straightforward code maintenance task requiring basic Python knowledge.
- **Estimated Time**: 20 minutes, given the minor nature of the changes.

Issue 7: Releasing Dataset on Hugging Face
- **Description**: This issue, at [issue 7](https://github.com/openai/SWELancer-Benchmark/issues/7), is a request from NielsRogge to host the SWE-Lancer Diamond dataset on Hugging Face for increased visibility, with benefits like easy loading via `from datasets import load_dataset` and dataset viewer support.
- **Requirements**:

  - Prepare the SWE-Lancer Diamond dataset, which is part of the public evaluation split mentioned in the paper at [ArXiv Paper](https://arxiv.org/abs/2502.12115), in a format suitable for Hugging Face, possibly JSON or Parquet.
  - Upload the dataset to Hugging Face using their documentation at [Hugging Face Docs](https://huggingface.co/docs/datasets/loading), ensuring compliance with their guidelines.
  - Update the repository documentation or links, such as the README.md, to reflect the new location, potentially citing the paper as suggested.
- **Complexity Score**: 4/10, requiring familiarity with Hugging Face dataset upload process and dataset preparation.
- **Estimated Time**: 30 minutes, considering the structured nature of the task.

Issue 8: Linking Actual Upwork Tasks
- **Description**: This issue, at [issue 8](https://github.com/openai/SWELancer-Benchmark/issues/8), is a suggestion to attach or cite the actual Upwork tasks corresponding to the benchmark tasks, enhancing traceability.
- **Requirements**:
  - Evaluate the feasibility of linking the benchmark tasks to their original Upwork postings, considering data privacy and availability.
  - Decide on an implementation method, such as adding URLs or identifiers in the dataset, possibly in a metadata file.
  - Update the dataset or documentation, such as the README or dataset files, to include these links, ensuring compliance with Upwork's terms.
- **Complexity Score**: 2/10, primarily a documentation and decision-making task.
- **Estimated Time**: 10 minutes, given the simplicity of evaluation and planning.

Issue 9: Fixing Setup Instructions
- **Description**: This issue, at [pull request 9](https://github.com/openai/SWELancer-Benchmark/pull/9), addresses errors in reproducing setup instructions, including default timeouts too low for NPM install, a typo in venv creation, container naming inconsistencies, and dotenv module issues.
- **Requirements**:
  - Identify and correct the typo in the README, which led to creating a new venv instead of using .venv, as per the repository setup instructions.
  - Adjust default timeouts in alcatraz for tasks, ensuring containers complete NPM install, possibly by increasing timeout values in configuration files.
  - Ensure container names are consistent, fixing the expectation of `swelancer` instead of `swelancer_x86` in code, potentially in run_swelancer.py.
  - Resolve the dotenv module issue by ensuring `uv run python` is used instead of `python`, as it solves the issue on Linux environments.
- **Complexity Score**: 5/10, requiring understanding of setup processes, debugging, and documentation updates.
- **Estimated Time**: 45 minutes, considering the multiple aspects to fix and test.

Issue 10: Minor Fix in run_swelancer.py

- **Description**: This issue, at [pull request 10](https://github.com/openai/SWELancer-Benchmark/pull/10), involves a minor fix to pass the example taskset to the runner from the args if provided to run_swelancer.py, enhancing flexibility.
- **Requirements**:
  - Modify run_swelancer.py, located in the root directory, to accept and pass the taskset argument correctly when provided via command-line args.
  - Test the changes to ensure the runner handles the taskset as expected, possibly by running with sample inputs and verifying output.
- **Complexity Score**: 3/10, a simple code modification task requiring basic Python and argument parsing knowledge.
- **Estimated Time**: 20 minutes, given the minor nature and testing needs.

Observations and Unexpected Findings

The analysis revealed that not all issues (e.g., Issues 1, 4, 8) are typical software engineering tasks, with some being questions, proposals, or suggestions. This unexpected finding suggests that the repository's GitHub issues include a mix of technical tasks and managerial or informational requests, which may not align with the benchmark's focus on freelance software engineering tasks from Upwork. This mix adds complexity to the analysis, as the benchmark paper indicates over 1,400 tasks, yet the GitHub issues analyzed here are fewer and of varying nature.

Conclusion

The analysis provides a comprehensive view of the first 10 issues, treating them as software engineering tasks with varying complexity and time estimates. The complexity scores range from 2 to 6, reflecting the diversity in task types, from simple documentation updates to more technical Dockerfile fixes. Estimated times range from 10 minutes to 1 hour, aligning with the task's scope. This report serves as a detailed guide for understanding and prioritizing these tasks within the SWELancer-Benchmark project.

Key Citations

- [SWELancer-Benchmark GitHub Repository](https://github.com/openai/SWELancer-Benchmark)
- [SWE-Lancer Paper ArXiv](https://arxiv.org/abs/2502.12115)
- [Cybergod Project GitHub](https://github.com/James4Ever0/agi_computer_control)
- [Cybergod Paper Website](https://james4ever0.github.io/Cybergod__God_is_in_your_computer.html)
- [Hugging Face Datasets Documentation](https://huggingface.co/docs/datasets/loading)