### **By comparing the outputs of SonarQube and JDeodorant, you can highlight the key differences between their detection techniques and use that as the basis for your manual labeling. How to Use This in Your Approach**

You are correct that the difference between these two tools is precisely what you need to build a defensible ground truth. Here's a systematic way to execute your plan:

1. **Run Both Tools**: Scan the JUnit 5 repository with both SonarQube and JDeodorant.
2. **Compare and Contrast the Outputs**: You will get two lists of issues. The lists will likely be very different.
   * **SonarQube's List**: This will contain issues based on its predefined, simple rules. For instance, it might flag a 150-line method as a "Long Method" regardless of its content.
   * **JDeodorant's List**: This will contain issues based on its structural analysis. For instance, it might flag a "Long Method" because of high complexity and dependencies on other classes, even if it's not a huge number of lines.
3. **Define Your Ground Truth with Context**: Your job as the researcher is to act as the "human-in-the-loop" and create a ground truth that is more accurate than either tool alone.
   * **Cases of Agreement**: If both tools flag the same issue, it is highly likely to be a true positive, strengthening your confidence in its label.
   * **Cases of Disagreement**: This is the most critical part of your research. When SonarQube flags an issue but JDeodorant does not, you will investigate the code. You might find that the issue is a false positive due to a specific coding pattern that SonarQube's static rule doesn't understand. Conversely, if JDeodorant flags an issue that SonarQube misses, you will investigate the code to determine if it's a true negative for SonarQube, which your ML model can then learn to detect.