Labels:

1. non_SATD
    a. Definition*: comment or textual artifact that does not contain any explicit or implicit self-admission of technical shortcuts, suboptimal design, workaround, or intentional violations of best practices.*
2. code/design_debt
    a. Definition*: comments that refer to sub-optimal, temporary, or expedient implementation choices that degrade code quality or design structure.*
    b. Examples:
        i. *"This is a hacky solution, should refactor later."*
        ii. *"Quick workaround, not the best design."*
        iii. *"This method is too complex, needs to be broken up."*
3. documentation_debt
    a. Definition*: comments where the author expresses that there is no proper documentation supporting that part of the program.*

b. Examples:
    i. *"**FIXME** This function needs documentation" - [from Columba] "*
    ii. *// TODO Document the reason for this" - [from Apache Jmeter]*

4. requirement_debt
    a. Definition*: comments express incompleteness of the method, class or program.*
    b. Examples:
        i. *"/TODO no methods yet for getClassname" - [from Apache Ant]*
        ii. *"//TODO no method for newInstance using a reverse-classloader" - [from Apache Ant]*
        iii. *"TODO: The copy function is not yet * completely implemented - so we will * have some exceptions here and there.*/" - [from ArgoUml]*
5. test_debt
    a. Definition: *comments are the ones that express the need for implementation or improvement of the current tests.*
    b. Examples:
        i. *"// TODO - need a lot more tests" - [from Apache Jmeter]*
        "//TODO enable some proper tests!!" - [from Apache Jmeter]
6. scientific_debt
    a. Definition*: comments express the accumulation of suboptimal scientific practices, assumptions, and inaccuracies within scientific software that potentially compromise the validity, accuracy, and reliability of scientific results.*
    b. Examples:
        i. *"We are going to share neff between the neutrinos equally. In detail, this is not correct, but it is a standard assumption because properly calculating it is (a) complicated (b) depends on the details of the massive neutrinos (e.g., their weak interactions, which could be unusual if one is considering sterile neutrinos)." - [From Astropy]*
        ii. *"We assume here that new ice arrives at the surface with the same temperature as the surface. TODO: Make sure this assumption is consistent with energy conservation for coupled simulations." - [From CESM]*
        iii. *"This frame is defined as a velocity of 220 km/s in the direction of l=270, b=0. The rotation velocity is defined in: Kerr and Lynden-Bell 1986, Review of galactic constants. NOTE: should this be l=90 or 270? (WCS paper says 90)." - [From Astropy]*
        iv. *"This does not work for large molecules that span ¿ half of the box!" - [From ROOT]*
        v. *"Since the energy and not forces are interpolated, the net force might not be exactly zero. This can be solved by also interpolating F, but that comes at a cost. A better hack is to remove the net force every step, but that must*

be done at a higher level since this routine doesn't see all atoms if running in parallel. Don't know how important it is? EL 990726." - [From GROMACS]

1. Possible Indicators of each debt type:
    a. Code/Design Debt:
        i. Automated Static Analysis (ASA) Issues
        ii. Code Metrics
        iii. Code outside of standards
        iv. Duplicated code
        v. Multithread correctness
        vi. Slow Algorithm
        vii. Brain Method
            1. A method that knows too much and controls too much of the program's flow.
        viii. Code Metrics
        ix. Code Smells
        x. Data Class
            1. Classes that only contain fields and getter/setter methods, with little or no behavior.
        xi. Data clumps
            1. Groupings of data that often appear together and should probably be encapsulated in their own class.
        xii. Dispersed Coupling
            1. A module/class that is dependent on many unrelated parts of the system.
        xiii. God class (or large class)
        xiv. Grime
        xv. Intensive Coupling
        xvi. Issues in the software design
        xvii. Refused Parent Bequest
            1. When a subclass does not use inherited methods or attributes from its parent.
        xviii. Schizophrenic Class
            1. A class that exhibits multiple, unrelated responsibilities or behaviors.
        xix. Structural Analysis
    b. Documentation Debt:
        i. Documentation does not exist
        ii. Incomplete Design Specification

    iii. Incomplete Documentation
    iv. Insufficient comments in code
    v. Outdated Documentation
    vi. Test Documentation
c. Test Debt:
    i. Incomplete Tests
    ii. Low coverage
    iii. Lack of tests
d. Requirement Debt:
    i. Requirement Backlog List
    ii. Comments discussing incompleteness
e. Scientific Debt:
    i. Translation Challenges
    ii. Assumptions
    iii. New Scientific Findings
    iv. Missing Edge Cases
    v. Computational Accuracy