Test Smell Research

Welcome! We are a group of researchers in Software Engineering investigating how developers perceive and prioritize test smells in Python-based software systems. This study is part of a broader effort to evaluate how developer perceptions align with metric-based prioritization approaches.

Your expert insights are invaluable to us and will help align academic research with real-world practices.

The survey takes approximately 15 minutes. Your participation is voluntary and anonymous. We do not collect any personal data beyond your responses. All data is stored securely and used solely for academic research purposes. You may exit or opt out at any time.

For questions, contact: [Generic Contact Info]

* Indicates required question

Section 2: Demographics

Purpose: Collect contextual data about participants' roles and experience to analyze how these factors influence test smell perceptions.

1. 01. What is your primary role in software development? (Select all that apply) *

Check all that apply.

- [ ] Software Developer/Engineer
- [ ] Test Engineer
- [ ] QA Engineer
- [ ] Technical Lead/Manager
- [ ] Researcher/Academic
- [ ] Other: _____

2. 02. How long have you worked (in any way) in Software Development / Coding / Programming? *

Mark only one oval.

- ◯ Never
- ◯ Less than 1 year
- ◯ 1 year or more but less than 2 years
- ◯ 2 years or more but less than 5 years
- ◯ 5 years or more but less than 10 years
- ◯ 10 years or more but less than 15 years
- ◯ 15 years or more but less than 20 years
- ◯ Over 20 years

3. 03. How long have you been writing unit test cases (in any way)? *

Mark only one oval.

- ◯ Never
- ◯ Less than 1 year
- ◯ 1 year or more but less than 2 years
- ◯ 2 years or more but less than 5 years
- ◯ 5 years or more but less than 10 years
- ◯ 10 years or more but less than 15 years
- ◯ 15 years or more but less than 20 years
- ◯ Over 20 years

Section 3: Definition-and-Familiarity-of-Test-Smell

Test Smell: The test code, just like production code, is subject to sub-optimal design and bad programming practices, also known as test smells.

To explore a comprehensive list of test smells and their characteristics, please refer to:
🔗 **The Open Catalog of Test Smells** — https://test-smell-catalog.readthedocs.io/en/latest/

4. 05. Are you familiar with the concept of "test smells"? *

Mark only one oval.

- ◯ Yes, I'm very familiar with test smells
- ◯ Yes, I've heard of them but am not very familiar
- ◯ No, I'm not familiar with test smells
- ◯ Other: _____

5. 06. Do you pay attention to test smells when designing/writing the test cases? *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Other: _____

6. 07. If **no**, what could be the reasons why you don't pay attention to test smells? (Select all that apply)

Check all that apply.

- ☐ Lack of awareness about test smells and their impact
- ☐ Time constraints, Resource constraints and delivery pressure
- ☐ Management not prioritizing test quality
- ☐ No enforcement of test quality in code reviews
- ☐ Refactoring test smells requires too much time or resources
- ☐ Difficulty remembering numerous test smell types
- ☐ Other: _____

7. 08. If you had access to a data-driven **test smell prioritization list** that ranks smells based on their impact, would it help you address test smells ? *

Mark only one oval.

- ◯ Yes, definitely
- ◯ Probably yes
- ◯ Not sure
- ◯ Probably not
- ◯ No, definitely not

Section 3 – Practical Insights and Prioritization of Test Smells

For each test smell below, please:

1. Read the definition
2. Rate how important you believe it is to refactor this smell (1-5 scale)
3. Indicate how frequently you encounter this smell in practice

8. **01. Assertion Roulette** *

Definition: A test case contains more than one assertion statement without an explanation/message.

Mark only one oval.

- ◯ 1 - Not important (No need to refactor)
- ◯ 2 - Slightly important
- ◯ 3 - Moderately important
- ◯ 4 - Important
- ◯ 5 - Very important (Critical to refactor)

9. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

10. **02. Conditional Test Logic** *

Definition: A test case contains one or more control statements (i.e., if, for, while).

Mark only one oval.

- ◯ 1 - Not important (No need to refactor)
- ◯ 2 - Slightly important
- ◯ 3 - Moderately important
- ◯ 4 - Important
- ◯ 5 - Very important (Critical to refactor)

11. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

12. **03. Constructor Initialization** *

Definition: A test suite contains a constructor declaration (an __init__ method).

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

13. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

14. 04. **Duplicate Assert** *

Definition: A test case contains more than one assertion statement with the same parameters.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

15. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

16. 05. **Empty Test** *

Definition: A test case does not contain a single executable statement.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

17. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

18. **06. Exception Handling** *

Definition: A test case contains either the try/except statement or the raise statement.

Mark only one oval.

- ◯ 1 - Not important (No need to refactor)
- ◯ 2 - Slightly important
- ◯ 3 - Moderately important
- ◯ 4 - Important
- ◯ 5 - Very important (Critical to refactor)

19. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

20. 07. **General Fixture** *

Definition: Not all fields instantiated within the setUp() method of a test suite are utilized by all test cases in this test suite.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

21. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

22. 8. **Lack of Cohesion of Test Cases** *

Definition: The mean of the pairwise cosine similarities between test cases in a test suite ≤ 0.4.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

23. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

24. 9. **Magic Number Test** *

Definition: A test case contains an assertion statement that contains a numeric literal as an argument.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

25. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

26. 10. **Obscure In-Line Setup** *

Definition: A test case contains ten or more local variables declarations.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

27. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

28. 11. **Redundant Assertion** *

Definition: A test case contains an assertion statement in which (1) the expected and actual parameters of equality are the same, e.g., assertEqual(X, X) or (2) the assertion of truth is carried out on the unchangeable object, e.g., assertTrue(True).

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

29. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

30. 12. **Redundant Print** *

Definition: A test case invokes the print() function.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

31. Have you ever encountered this test smell in your work? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

32. 13. **Sleepy Test** *

Definition: A test case invokes the time.sleep() function with no comment.

Mark only one oval.

- 1 - Not important (No need to refactor)
- 2 - Slightly important
- 3 - Moderately important
- 4 - Important
- 5 - Very important (Critical to refactor)

33. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

34. 14. **Suboptimal Assert** *

Definition: A test case uses a general assertion function (e.g., assertTrue, assertFalse, assertEqual) instead of a more specific unittest assertion function (e.g., assertIn, assertNotEqual, assertIs), reducing the clarity of the test's intent and the informativeness of failure messages

Mark only one oval.

- ◯ 1 - Not important (No need to refactor)
- ◯ 2 - Slightly important
- ◯ 3 - Moderately important
- ◯ 4 - Important
- ◯ 5 - Very important (Critical to refactor)

35. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

36. 15. **Test Maverick** *

Definition: A test suite contains at least one test case that does not use a single field from the SetUp() method.

Mark only one oval.

- ◯ 1 - Not important (No need to refactor)
- ◯ 2 - Slightly important
- ◯ 3 - Moderately important
- ◯ 4 - Important
- ◯ 5 - Very important (Critical to refactor)

37. Have you ever encountered this test smell in your work? *

Mark only one oval.

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

Section 5: Final Thoughts

Thank You!

Thank you for participating in our research on test smell prioritization. Your responses will help improve our understanding of how developers perceive and prioritize test smells, and will contribute to the development of better tools and methodologies for test code maintenance.

For questions or feedback about this research, please contact:

[Generic Contact Info]

38. Do you have any additional comments or insights about ***test smell prioritization*** that weren't covered in this survey?