



I want to become a better Python software developer for medical image reconstruction and pulse sequence programming. I've heard about code reviews, but I don't know exactly what kinds of fixes or code suggestions they entail. I also am the only person on my team capable of developing image reconstruction algorithms and pulse sequences. I want to start using AI (Claude Opus or Sonnet or Windsurf) to propose changes to my code after a first working draft of a module so that my code is clean, readable, documented, more robust, less buggy, more performant, and more generalizable.

I've heard about "modes" for code reviews. For example, Correctness Mode would check for logic bugs, numerical issues, and array handling. Performance mode would check for bottlenecks on large 3D or 4D volumes, Test Gap Mode would check for missing edge cases and failure modes. Security Safety Mode would check for SAR limits, PHI/HIPAA compliance, and FDA compliance. API Design Mode would check for usability, naming, type hints, and docs. Maintainability Mode would check for tech debt, duplication, and complexity. Domain specific mode would check for MRI physics, k-space, coils, and sequences. You may agree or disagree on the inclusion of these modes or whether additional modes should be included in your final product.

I want you to do research on how to perform good code reviews for my type of work as a software developer, MRI pulse sequence engineer, and image reconstruction scientist. Then, I want you to output separate markdown files for each mode that would be used to prompt Claude or Windsurf to act as a code reviewer after I finish writing my code. Each code review should create for me a learning opportunity - it should identify the problems or potential problems in my code and propose a solution.

I already have an established code style that I would like to maintain across my codebase. The attached markdown file documents the code style. if you think the code style should be changed or adjusted, also output a new markdown file so that Claude or Windsurf AI can enforce consistent code style across all modules, functions, and files.

Perfect! I've created a comprehensive set of code review mode prompts tailored to your work as an MRI pulse sequence engineer and image reconstruction scientist. Here's what I've delivered:

▮ Code Review Mode Files Created

1. **01_correctness-mode.md** - Logic bugs, numerical issues, array handling, edge cases
2. **02_performance-mode.md** - Memory optimization, vectorization, bottlenecks for large 3D/4D volumes
3. **03_test-gap-mode.md** - Missing edge cases, failure modes, test coverage
4. **04_security-safety-mode.md** - SAR limits, HIPAA/PHI compliance, FDA validation, patient safety
5. **05_api-design-mode.md** - Usability, naming, type hints, documentation
6. **06_maintainability-mode.md** - Tech debt, duplication, complexity, refactoring
7. **07_code-style-mode.md** - Enforces YOUR documented style with medical imaging context
8. **08_mri-domain-mode.md** - MRI physics, k-space, coils, sequences, reconstruction algorithms

▮ How to Use These

For Claude or Windsurf:

```
I've finished my first working draft of [module name].  
Please review it using [mode name] from the attached prompt.
```

```
[Paste code here]
```

Recommended Review Sequence:

1. **First pass:** Correctness Mode (catch bugs early)
2. **Second pass:** MRI Domain Mode (verify physics)
3. **Third pass:** Security/Safety Mode (ensure patient safety)
4. **Polish:** API Design + Code Style modes
5. **Optimization:** Performance Mode (if needed)
6. **Final check:** Test Gap + Maintainability modes

▮ Key Features of These Prompts

Each mode provides:

- ✓ **Strengths identified** (positive reinforcement)
- ✗ **Issues with examples** (before/after code)
- ▮ **Domain-specific recommendations**

- **Learning opportunities** (explains WHY, not just WHAT)
- **Graded assessment** (A-F rating)
- **Estimated fix time**

▮ **Special MRI/Medical Focus**

The prompts understand:

- K-space operations and FFT conventions
- Coil sensitivity and parallel imaging
- Gradient timing and slew rates
- SAR calculations and patient safety
- HIPAA compliance for PHI
- FDA validation requirements
- Multi-dimensional medical image arrays

▮ **Your Code Style is Preserved**

The **Code Style Mode** enforces your existing standards from `documentation_style.md` including:

- Google-style docstrings
- Type hints
- NumPy arrays without explicit types
- Your specific formatting preferences

You can use these prompts independently or combine multiple modes in one review session. They're designed to make you a better developer by teaching you MRI-specific best practices while improving code quality.

Would you like me to create a "master review prompt" that runs multiple modes in sequence, or would you like any adjustments to the individual mode files?

✱

1. `documentation_style.md`
2. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3745282/>
3. https://radiology.queensu.ca/source/MR_Pulse_Sequences_What_Every_Radiologist_Wants_to_Know.pdf
4. <https://www.bradyknowsmedical.com/documents-download/technical-review-guideline-of-medical-device-software-registration/>
5. <https://github.com/jongcye/kspace.deeplearning.MRI>
6. <https://www.sciencedirect.com/science/article/abs/pii/S0730725X2100237X>
7. <https://carecloud.com/continuum/radiology-coding-guidelines-best-practices/>
8. <https://mriquestions.com/what-is-k-space.html>
9. <https://www.augmentcode.com/guides/python-code-review-checklist>

10. <https://graphite.com/blog/code-review-best-practices>
11. <https://www.ketryx.com/blog/fda-software-verification-vs-validation-whats-the-difference>
12. <https://pmc.ncbi.nlm.nih.gov/articles/PMC4561593/>
13. <https://redwerk.com/blog/python-code-review-checklist/>
14. <https://group107.com/blog/code-review-best-practices/>
15. <https://www.greenlight.guru/blog/software-validation>
16. https://www.linkedin.com/posts/qodana_heres-your-python-code-review-checklist-activity-7359350158372782080-iaq3
17. <https://jellyfish.co/library/developer-productivity/peer-code-review-best-practices/>
18. <https://www.ni.com/en/support/documentation/supplemental/08/fda-validation-of-medical-devices-with-national-instruments-hard.html>
19. <https://www.jetbrains.com/pages/static-code-analysis-guide/python-code-review-checklist>
20. <https://www.legitsecurity.com/aspm-knowledge-base/best-practices-for-code-review>
21. <https://www.fda.gov/media/73141/download>
22. https://www.reddit.com/r/learnpython/comments/1iii1ye/my_python_code_review_checklist/
23. <https://www.devicelab.com/blog/the-complete-guide-to-medical-device-software-design-and-development/>
24. <https://dev.to/marcopatino/i-analyzed-code-review-best-practices-for-a-year-this-is-what-i-learned-about>
25. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation>
26. <https://www.biac.duke.edu/research/safety/mri-safety-tutorial>
27. <https://www.dicomdirector.com/hipaa-standards-for-medical-imaging/>
28. <https://pubmed.ncbi.nlm.nih.gov/34861358/>
29. <https://www.aapm.org/meetings/02am/pdf/8356-48054.pdf>
30. <https://www.hipaajournal.com/hipaa-photography-rules/>
31. https://mr.research.columbia.edu/sites/default/files/content/Tong_Validating.pdf
32. <https://mriquestions.com/sar-limits.html>
33. <https://pmc.ncbi.nlm.nih.gov/articles/PMC12561909/>
34. https://openaccess.thecvf.com/content/ICCV2021W/LCI/papers/Ryu_K-Space_Refinement_in_Deep_Learning_MR_Reconstruction_via_Regularizing_Scan_ICCVW_2021_paper.pdf
35. <https://www.fda.gov/radiation-emitting-products/mri-magnetic-resonance-imaging/mri-information-industry>
36. <https://www.acr.org/Data-Science-and-Informatics/Informatics/PHI>
37. <https://www.fda.gov/media/74201/download>
38. <https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html>
39. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9011005/>
40. <https://www.pnas.org/doi/10.1073/pnas.2020516118>
41. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11376551/>
42. <https://www.nature.com/articles/s41598-024-72820-2>

43. <https://www.ismrm.org/24/pf/WE-20.htm>

44. <https://web.eecs.umich.edu/~weimerw/p/weimer-fse2020-bias.pdf>