**Walk-through Comments Form**

Product To Review: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Doc. Date and Rev: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Product Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time to Review: \_\_\_\_\_\_\_\_\_\_\_\_ hr

Reviewer Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time to Rework: \_\_\_\_\_\_\_\_\_\_\_\_ hr

Date of Review: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Total Time: \_\_\_\_\_\_\_\_\_\_\_\_hr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Location in Product (line no, method) | Description of Issue and Comments (describe defect, improvement, or alt. implementation) | Disposition | Remarks (author) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Disposition: Inc = incorporate, N.I. = Not incorporated (state reason), Mod = Incorporate with modifications, Other

Software Walk-through Guidance

|  |  |
| --- | --- |
| Purpose | Find defects in code.  Suggest possible improvements and alternatives.  Share knowledge regarding software implementation and its function. |
| Preparation | 1. Author selects code to review and selects one or more reviewers.  2. One reviewer is designated as *scribe* to record items on walk-through form.  3. Select a quiet environment where walk-through can be done comfortably.  4. Agree on a time for walk-through. Allocate sufficient time. Typical rate is 100 - 200 lines per hour. |
| Procedure | 1. Author explains purpose of code being reviewed. Explain context or use case the code is used in.  2. Author answers any questions about the purpose.  Logging Phase:  3. Author leads reviewers through the code line-by-line. Reviewers should study each line and ask questions until they understand it. Make suggestions.  4. Scribe records comments, suggestions, defects found by reviewers.  Discussion Phase:  5. After the entire software product has been reviewed, review and discuss any issues that remain.  Sometimes a reviewer may change his mind about an issue after seeing the entire code.  Conclusion:  6. Record the total time spent on review.  7. Record defects in defect tracking system.  Rework & Follow-up:  8. Author should review all comments and either implement or not implement with explanation. |
| Suggestions | Make the code easy to read. Consider using more than one monitor.  Make sure all reviewers are reviewing same version of code. |
| Form Instructions | Document Name - location and name of the file in VCS. Specify branch of VCS where document is.  Date and Rev - identify document date and rev. For Git, use the commit hash prefix, e.g 43a7bc3. This is shown in document "history" on Github or Bitbucket |