

A Montana Tech Method Software Development Standard

MTM Simple Program Development Sheet Inspection Process

Version 1.8
Feb. 11, 2019

A. Frank Ackerman

Software Engineering
Montana Tech of the University of Montana

<i>Version</i>	<i>Date</i>	<i>Author</i>	<i>Comment</i>
1.0	05/20/10	Frank Ackerman	First version
1.1	04/09/11	Frank Ackerman	Improvements for Senior Seminar Talk
1.2	07/28/11	Frank Ackerman	Generalize for multiple inspections
1.3	09/19/11	Frank Ackerman	Clarify re-work considerations
1.4	02/06/14	Celia Schahczenski	Modify for AbOut inspections, ESOF 326, Spring 2014
1.5	01/18/18	Celia Schahczenski	Modified for AbOut inspections, ESOF 326, Spring 2018
1.6	02/10/18	Celia Schahczenski	Minor updates for AbOut inspections, ESOF 326, Spring 2018
1.7	01/23/19	Celia Schahczenski	Modified for AbOut inspections, ESOF 326, Spring 2019
1.8	02/11/19	Celia Schahczenski	Updated to improve use of GitLab and Outlook Calendar Meetings

1. Introduction

This document describes the inspection process to be used to inspect AbOut items.

2. Entry Conditions

1. One or more inspections are required to close an issue. For each issue inspections of the relevant portions of the SRS, SDD, test suite, database and code (php, html, css, mysql, javascript) are required. Multiple items, for the same issue, may be inspected at the same time.
2. The author(s) of the inspection item(s) is the person who has made changes to the item. The author informs the scrum master that an inspection is needed by making a merge request and assigning the request to the scrum master. The merge request is to contain all changes relevant to the issue and no other changes. The author estimates how much time an inspector will need to look over the item and how much time will be needed for the inspection meeting. These are added to the merge request as:

Estimated prep time for each inspector:

Estimated inspection time:

3. The scrum master selects a moderator and the inspectors. The scrum master asks the moderator to facilitate and oversee the inspection, and tells who the inspectors will be. (In Spring 2019 all non-authors, non-moderators are likely to be inspectors.) The scrum master adds an estimation of the total time needed for the inspection via:

`/estimate n`

(Example: `/estimate 1d3h30m5s`. GitLab looks for these specifications when they are the only text in a comment.)

4. The moderator talks to the author and determines if adequate attention has been given to the SRS, SDD, test suite, database and code. If the moderator determines items are ready to be inspected the moderator sets a date and time for the inspection, creates an Outlook Calendar Meeting (OCM) for it, and sends an invitation to the author(s), the inspectors and the product owner. The OCM includes the issue number, issue title and a link to the merge request associated with the issue. This OCM must go out at least 24 hours in advance, but hopefully much earlier.
5. Inspectors carefully review all changes present in the merge request, placing comments, located at the relevant locations, into the merge request. Inspectors should make note of parts of the material he or she did not fully understand, he or she suspects may be defective, poorly written statements, code which does not satisfy the coding standards, and any items he or she think are actual defects.

Each inspector should be aware of how much time he or she spent on preparation and record that in the merge request (/spend, similar to /estimate above).

NOTE: In order for each meeting to be effective, it is a general rule to limit items in each meeting so that only 150-200 lines of code/documentation are being review per 1 hour.

3. Inspection Process

a. For every inspection there are three participant roles:

i. Moderator

The Moderator is in charge of the inspection meeting. The Moderator needs to make sure that all defects the team discovers are clearly recorded in the merge request. The Moderator is the inspection facilitator, and does not necessarily review the inspection items.

ii. Author(s)

The Author is the person who is responsible for the item(s) to be inspected. The Author is not an inspector. The item(s) inspected already contain the Author's changes.

The Author's inspection grade does not depend on the number of defects identified, but on quality in which the defects are addressed and the timeliness and completeness of the materials provided for the inspection.

iii. Inspector

Any member of the inspection team that is not the moderator or an author is designated simply as an Inspector. Inspectors are responsible for: (1) preparing for the meeting by making note of potential defects, poorly written statements, code which does not satisfy the coding standards, and (2) actively participating in the meeting by pointing out each occurrence of the above.

3.1 Inspection meeting

The inspection Moderator conducts the inspection meeting. The first item of business is to decide if the team is sufficiently well prepared to proceed with the inspection. The moderator looks at the preparation "spend" time on the merge request. If the team is not sufficiently well prepared the meeting should be rescheduled.

Author begins by describing the relevant issue(s) and shows the pertinent user interface, if applicable. The Author then walks through the merge request comments on all inspection item(s). Comments on the merge request may be resolved in the meeting, or may constitute a defect that the author needs to resolve at a later time.

An inspection meeting should be narrowly focused on finding defects in work products. All inspectors are responsible for keeping this focus and should keep in mind that it is the work products that are being inspected, not the Author. Avoid personalizing the work product, address the work product items, not the Author. For example, don't say, "You wrote here ..." but rather something like "The code here is saying ...". Do not try to correct any of the defects found. Problem solving is fun, but an inspection meeting is not the place for it; speak to the Author after the meeting if you think you have an idea that he or she would benefit from.

Since the purpose of an inspection meeting is to find defects in the Author's work products it's often easy to create interpersonal tensions. All Inspectors should go out of their way to avoid this and the Moderator should step in and address any interpersonal problems that arise.

3.2 Determination if re-inspection is necessary

The inspection team determines whether or not a re-inspection is necessary. Usually the defect corrections necessary can safely be checked when the Moderator verifies the Author's rework (see step below) so a re-inspection is not necessary. In some cases, however, (1) the Author may not have clearly understood the problem, (2) the requirements may have changed between the time the Author was assigned the problem and the time of the inspection or (3) significant parts of the artifact being inspected were missing. In these cases a re-inspection may be necessary. Also, if any of the defects found during the meeting appear to involve a substantial amount of re-work or complex rework a re-inspection should be held.

3.3 Author addresses defects

The Author should fix and comment on each item not resolved during the inspection. Usually this will just be the comment "fixed" written into the merge request. In some cases, however, the Author may decide that what he or she thought was a defect during the inspection meeting is not in fact a defect. The reasons for this should be noted.

3.4 The Moderator verifies the Author's rework

When the author has fixed all of the code defects, he/she informs the moderator. The Moderator verifies that all defects are resolved and closes the merge request, removing the relevant branch.

4. Exit Condition

The process described above is complete.