**Appendix A**

**Sample Process for Software Inspection**

Process Documentation Template:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Title</td>
<td>Defines process name, such as Software Inspection Process</td>
</tr>
<tr>
<td>Process #</td>
<td>CSP-ddd</td>
</tr>
<tr>
<td>Date</td>
<td>January 20, 1998</td>
</tr>
<tr>
<td>Created/Modified By</td>
<td>Software process group (SPG).</td>
</tr>
<tr>
<td>Definitions</td>
<td>Defines all the special concepts and roles in the process</td>
</tr>
<tr>
<td>Rationale</td>
<td>A software inspection is a method to identify and document defects and</td>
</tr>
<tr>
<td></td>
<td>problems found during the inspection Process.</td>
</tr>
</tbody>
</table>

**Definitions:**

A) **Defects**

A defect is an error in a software product including documentation, requirements models, designs, code, test plans/procedures/results, user documentation, .. etc. A defect prevents the software from meeting its requirements or meeting standards. Defects are classified according to severity, type, mode, phase generated, and phase found.

B) **Severity:**

- **Critical.** A major defect has no known work around, and will cause the system to fail in meeting a requirement.
- **Major.** Causes a malfunction or unexpected result, and would lead to an incorrect response or misinterpretation of the information by the user.
- **Minor.** It is undesirable but would not cause a malfunction or unexpected result, and is undesirable but would not be detected by the user.

C) **Type:**

- **Requirements Defects:**
  - Requirement Wrong (RW): An error exists in the requirements that are included.
  - Requirements Missing (RM): The requirements are incomplete.
  - More Detail (MD): The requirements are ambiguous as written. Need enough detail so that there’s only one correct interpretation of the information present.
  - Interface (IN): An error exists in an interface defined between objects.
  - Standards (ST): The specification do not meet the standards.

Design:
Logic (LO): An error exists in the logic.
More Detail (MD): The design is ambiguous as written. Need enough detail so that there’s only one correct interpretation of the information present.
Standards (ST): The code and ADL, including headers, does not perform to required specifications.
Performance (PE): If the object is time-critical, alternate designs may be discussed to optimize performance.
Higher Level Design (HL): The preliminary design does not correctly implement the requirements.
Requirements (RQ): A requirements error was detected at the detailed design inspection.
Interface (IN): An error exists in an interface defined between objects.
Precession (PR): The precession of a data entity is incorrect.
Test Case (TC): An error was found in an object test case.

D) Mode:
- Missing. Information that is specified in the requirements or standards, but is not present.
- Extra. Information that is not specified in the requirements or standards, but is present.
- Wrong. Information that is specified in the requirements or standards, but is incorrect.
E) Phase Generated:
Phase in which the defects was introduced. For defects found in inspections, this the current phase.

F) Phase Found:
Phase in which the defect was found. For defects found in specifications, this is the current phase.

G) Additional Information Needed Per Defect:
- Source SDR (If Any) {Not applicable for Defects found in inspections}
- Date Defect Found
- Date Defect Corrected
- Document or Code in which defect found
- Mhrs to Correct Defect.

H) Inspected Items:
Software inspection process is suitable for all software products such as:
- User manuals.
- Software development Plan.
- Software requirements specification.
- Software design document.
- Code.
- Test procedures.
- Test plans.

Software Inspection Process:
The typical phases of an inspection are as follows:

Planning.
After a work package is completed, it is scheduled for inspection. Specific roles are assigned for the inspection team, and the materials, such as listings, requirements specifications, and design documents, are distributed to the team members.

Overview.
A presentation is made to the team members to familiarize them with the work package to be inspected. If the team is already familiar with the package, then no overview meeting is necessary.

Individual Inspection.
Each inspector reads the material looking for and noting errors. As mentioned below, different inspectors may concentrate on different aspects of the same work package to speed the process.

Inspection Meeting.
This is a moderated meeting in which the reader presents the work package, and individual inspectors look for and note errors. The meeting is limited to two hours at the most, and defects are noted without proposing resolutions.

Rework.
The author of the work package fixes the listed defects.

Follow-up.
The corrections are checked and, if necessary, another inspection is scheduled.

<table>
<thead>
<tr>
<th>Roles &amp; How</th>
<th>The participants on a software inspection and a general description of their responsibilities are as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderator.</td>
<td>The person who ensures the readiness of the product for inspection (clean compiles, conforms to style standards, and so forth) distributed the inspection materials, schedules, and runs the inspection meeting. In the inspection meeting, the moderator makes sure that team members remain focused on finding and recording defects.</td>
</tr>
<tr>
<td>Reader.</td>
<td>During the inspection meeting, the reader presents the work and in some cases paraphrases the text. This should not be the author/developer but should be a key inspector.</td>
</tr>
<tr>
<td>Inspector.</td>
<td>The people who read the text of the work package both before and during the meeting, looking for defects. The number of inspectors should be minimized so that the whole inspection meeting has no more than six to nine people.</td>
</tr>
<tr>
<td>Author</td>
<td>The person who has created the work package. The author may not be the reader or the moderator. Answers questions during inspection. Fixes the problems found.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>When</th>
<th>Whenever needed and scheduled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regulations</td>
<td>Inspections should not be combined.</td>
</tr>
<tr>
<td></td>
<td>Inspections should not exceed the recommended number of attendees.</td>
</tr>
<tr>
<td></td>
<td>Inspections should not exceed two hours meeting time.</td>
</tr>
<tr>
<td></td>
<td>Any exceptions must be documented in the Project Development Plan (PDP) for that software project.</td>
</tr>
<tr>
<td></td>
<td>(... other regulations).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checklists</th>
<th>Note: Must be prepared by the author.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms</td>
<td>A sample form is included in Software Requirement Specifications Inspection Sheet.</td>
</tr>
<tr>
<td>Notes</td>
<td>Lists problems encountered and participants’ suggestions</td>
</tr>
</tbody>
</table>