SOURCE CODE REVIEW TEST REPORT

FOR

CLEARBALLOT APPLICATION

Prepared by: _____

Jack Cobb, Laboratory Director

September 2, 2014

Pro V&V, Inc.
700 Boulevard South, Suite 102
Huntsville, AL 35803
256-713-1111

TABLE OF CONTENTS

1.0 INTRODUCTION ……...1

1.1 References ……..…..1
1.2 Background …….……1

2.0 TEST ITEMS ……..1

2.1 Software ……..1
2.2 Test Support Equipment/Materials …….....2

3.0 TEST PROCESS……...……..…2

3.1 General Information……...…..2
3.2 Test Specifications……...…….....2
3.3 Test Procedures ……..3

4.0 TEST FINDINGS……..4

5.0 RECOMMENDATIONS ……..……...4

ATTACHMENT A TEST PROCEDURES ……..….5

ATTACHMENT B DIGITAL SIGNATURES……..….9

1.0 INTRODUCTION

Clear Ballot Group contracted with Pro V&V, Inc. to perform a source code review and a trusted build for the ClearBallot application. A source code review was performed to confirm compliance of the source code submitted by Clear Ballot Group for the ClearBallot application to the Election Assistance Commission (EAC) 2005 Voluntary Voting Systems Guidelines (VVSG). After the completion of the source code review a Trusted Build was performed per the EAC Testing and Certification Manual version 1.0. This Test Report documents the procedures followed, the results obtained, and the conclusions drawn from that examination.

1.1 References

The documents listed below were used in the development of this Test Report.

- United States Election Assistance Commission (EAC) 2005 Voluntary Voting Systems Guidelines (VVSG)
- EAC Decision on Request for Interpretation (RFI) 2010-02
- PEP 8 – Style Guide for Python Code (legacy.python.org/dev/peps/pep-0008/)
- Pro V&V. Inc. Source Code Review Test Plan for Examination of the ClearBallot Application, dated June 14, 2014

1.2 Background

The Clear Ballot Group submitted source code documented in Attachment B of this document for review of the ClearBallot application to the 2005 VVSG. Clear Ballot Group requested Pro V & V review the ClearBallot source code to the 2005 VVSG with all applicable EAC RFI's. Pro V&V performed an initial source code review and provided feedback to Clear Ballot Group. Pro V&V then performed a full review following the procedures described in the Pro V&V Test Plan. After all anomalies were resolved, Pro V&V performed a Trusted Build.

2.0 TEST ITEMS

The following sections document the source code submitted and the software used to perform the source code review.

2.1 Software

This subsection contains a listing of the software and firmware being evaluated.

Table 2.1 Software/Firmware

Component Name	Version	Unique Identifier (digital signature or hash value)
NY-ProVV-Source-Review-1.0.5.zip	1.0.5	7b1e6de4893b13a8142056489916532315170691 (SHA-1)

The PEP8 coding style is mature enough that static code analysis tools have been developed to enforce the requirements set forth by the documented coding style. Clear Ballot Group selected PYLint as a static source code analysis tool. An analysis was performed and PYLint was found to be acceptable as a static source code analyzer to enforce the PEP8 coding style. In addition to the PEP8 coding style, the source code for the ClearBallot application was subjected to the following VVSG requirements:

- Volume I Section 5.2.2
- Volume I Section 5.2.3 a
- Volume I Section 5.2.3 b
- Volume I Section 5.2.3 c
- Volume I Section 5.2.3 e
- Volume I Section 5.2.3 f
- Volume I Section 5.2.4
- Volume I Section 5.2.7 a

3.3 Test Procedures

The test procedures used for this source code review were VSTL-TP-200, VSTL-TP-210, and VSTL-TP 250 which are presented in Attachment A of this document.

The strategy for evaluating the ClearBallot application source code for compliance to the 2005 EAC VVSG was to review the PYLint configuration file (pylintrc) submitted by Clear Ballot against the PEP8 code style to ensure the false positive suppression does not violate any PEP8 standard or any VVSG standard. After the review, the PYLint static source code analysis tool was loaded with the Clear Ballot Group's configuration and the tool was then executed against the ClearBallot application source code. The report produced from this scan was then analyzed and any discrepancies noted were documented and resolved.

Following the static source code analysis, a manual source code review was performed to ensure the ClearBallot application source code was compliant with the following 2005 EAC VVSG requirements:

- Volume I Section 5.2.2
- Volume I Section 5.2.3 a
- Volume I Section 5.2.3 b
- Volume I Section 5.2.3 c
- Volume I Section 5.2.3 e
- Volume I Section 5.2.3 f
- Volume I Section 5.2.4
- Volume I Section 5.2.7 a

Any discrepancies discovered were documented and submitted to the Clear Ballot Group for resolution.

4.0 TEST FINDINGS

This section contains a detailed description of the results obtained.

VSTL-TP-200: Source Code Inspection

The objective of this test case was to perform a formal review of a customer submitted source code to specific requirements. The requirements were published standards for PEP8 and the 2005 VVSG.

Summary Findings: During execution of this test procedure, it was verified that the submitted source code met the requirements of the review. The review revealed that the source code was in compliance with the required standards and that the source code was written within the parameters of its design.

VSTL-TP-210: Compliance Inspection

The Compliance Inspection consisted of comparing customer submitted source code to specific criteria. The specific criteria were PEP8 coding standard and the 2005 VVSG.

Summary Findings: During execution of this test procedure, it was verified that the submitted source code met the requirements. The review revealed that the source code was in compliance with the required standards and that the source code was written within the parameters of its design.

VSTL-TP-250: Build Inspection

The Build Inspection consisted of compiling the ClearBallot source code into executable software and third party products into an installation media using the Clear Ballot Build procedures. This was performed using the procedure for a Trusted Build.

Summary Findings: Pro V&V created the following products during the Windows Builds for the ClearBallot source code:

- BallotDeleteBox.exe – SHA1 - 22751a6642155853ef7ea1c2d02e9a9b50a0d201
- BallotTabulator.exe – SHA1 - 275c7c6be35980e93566edd16b66b89f191447e0
- GetScannerInfo.exe – SHA1 - 9b7995e81e5d422149584d7b83f57bbc7334792a
- SetupScanner.exe – SHA1 - e43c9130e3219d671f4803b6ad35c9c5ba6bcb85
- Twain.pyd – SHA1 - c860ab78012a26074bd34cd58f86579d62e43b20

The products were combined with all required third party products during the Linux Build into the following installation media:

- Clearballot.iso – SHA1 - 4f7bf3b847e0157b58b56945f8b89ad2df693f79

5.0 RECOMMENDATIONS

Based on the review performed and the results obtained, the ClearBallot Application described in this report demonstrated successful implementation of all identified requirements submitted for evaluation.

ATTACHMENT A

TEST PROCEDURES

	VSTL TEST PROCEDURE: VSTL-TP-210
Title: Compliance Inspection	**Requirement:** 2005 EAC VVSG, Volume I & II, Section 5
Author: Jack Cobb	**Origination Date:** 1/31/2012
Procedure Description: Procedure to be followed during the performance of a Compliance Inspection	

Overview: A Compliance Inspection is a specific type of source code inspection. It consists of comparing customer submitted source code (Software Under Test – SUT) to specific criteria. The specific criteria could be, but not limited to, federal, state, or customer supplied standards. The purpose of this inspection is to examine the supplied source to determine if the source code is in compliance with the standard it is being compared to.

Procedure:

1. Receive and catalog the SUT per the VSTL CM Plan "Handling of Test Items and Test Data".
2. Determine if the inspection will be a manual line-by-line inspection or an automated inspection.
3. If the inspection is an automated inspection, ensure the software test tools is installed and validated according to VSTL Program Manual "Equipment". Once the test environment is properly configured, place the SUT into the test environment.
4. Execute the automated tool.
5. If the inspection is a manual line-by-line inspection, a determination needs to be made whether the inspection is an initial inspection or a comparison of previously inspected source code.
6. If the inspection is an initial inspection, then a qualified Pro V&V employee will inspect every line of the SUT and validate that it meets the standard or requirement it is being compared to.
7. If the inspection is an inspection of previously inspected source code, then a qualified Pro V&V employee will retrieve the original SUT and compare the new submission to determine the changed area of the SUT. A qualified Pro V&V employee will inspect every changed or modified line of the SUT and validate it meets the standard or requirement it is being compared to.
8. After the Compliance Inspection of SUT has been completed, all issues are recorded and reported.
9. The customer may be allowed to remediate the issues and resubmit the SUT.