Scientific Software Installation Testing Using a Plan-Do-Check-Act Methodology: A Case Study

Jorge Domínguez-Patiño*, Antonio Rodríguez Martínez2, Rosenberg Javier Romero2, Marco Cruz Chávez2, Martha Lilia Domínguez Patiño3 and Rosa María Melgoza Alemán3

1. Posgraduate School in Engineering and Applied Science, Autonomous University of Morelos State, Cuernavaca 62209, México
2. Research Centre of Engineering and Applied Science, Autonomous University of Morelos State, Cuernavaca 62209, México
3. FCQel (Faculty of Chemical Sciences and Engineering), Autonomous University of Morelos State, Cuernavaca 62209, México

Abstract: The scientific software installation testing has a main goal: Evaluate if the software meets its requirements and specifications. In this paper, the scientific software installation in six machines is evaluated. The software installation was tested using a PDCA (Plan-Do-Check-Act) approach in 3 machines and were compared with other 3 machines which were installed exclusively based in the installer experience. The software installed on the machines using a PDCA approach for testing, lead to the expected results. Scientific software installation should be tested during the installation and not as a final test. A methodology based on PDCA is recommended for testing scientific software.

Key words: Testing, PDCA, scientific software.

1. Introduction

Scientific software is widely used in science and engineering fields. Scientific software is mainly developed to better understand or make predictions about real world processes [1]. Kanewala U. [1], defines scientific software as software used for scientific purposes.

Due to the lack of systematic testing of scientific software, subtle faults can remain undetected [1].

The software testing process has two distinct goals:

(1) To demonstrate the developer and the customer that the software meets its requirements. For custom software, this means that there should be at least one test for every requirement in the user and systems requirements documents. For generic software products, it means that there should be tests for all of the systems features that will be incorporated in the product release.

(2) To discover faults or defects in the software where the behavior of the software is incorrect, undesirable or does not conform to its specification. Defect testing is concerned with rooting out all kinds of undesirable systems behavior, such as system crashes, unwanted interactions with other systems, incorrect computations and data corruption [2].

Software testing is commonly seen as a process of executing test cases that are carefully predesigned using test case design techniques (Baizer 1990 [3]; Kaner et al. 1999 [4]; Myers 1979 [5]). In this test case based approach, the goal is to document the required knowledge in the test case. The actual test execution, even if performed as a manual activity, is considered a mechanical task. During execution, the predefined test cases are run and their output results, are compared with the documented expected results [6].

The software acceptance testing process takes the software system as a whole (including documentation and other such objects) and runs a series of system test suites against it. The software acceptance testing process tests the system from the perspective of the

*Corresponding author: Jorge Domínguez Patiño, master in chemical engineer, research fields: life cycle assessment and validation. E-mail: jorge.dominguez@uaem.mx.
client, verifying that the system as a whole operates according to requirements [2].

According with GAMP 5® [7] testing fulfills objectives such as: identifying defects so they can be corrected or removed before operational use, preventing failures that might affect patient safety, product quality or data integrity, providing document evidence that the systems performs as specified, demonstrating the system meets its requirements, providing confidence that the system is fit for its intended use, providing a basis for user acceptance and meeting a key regulatory requirement.

Software testing is necessary to produce highly reliable systems [8].

Software testing entails running software products under known conditions with defined inputs and documented outcomes that can be compared to their predefined expectations. It takes time, the activity is difficult and imperfect. As such, it requires early planning in order to be efficient [9].

Finding defects before release is an important and costly software engineering activity that is typically achieved through software testing and reviews [10].

It is important to recognize that quality cannot be tested into products, quality should be built in or should be by design [11].

The goal of this paper is to present a testing installation methodology for scientific software. In this case the scientific software installed is SimaPro 8 software [12].

2. Methodology

The study was split into two stages:

(1) Installation of the software on three machines without using a testing methodology based solely on the experience of the installer responsible.

(2) Installation of the software on three machines using the PDCA (Plan-Do-Check-Act) approach.

Testing process as software development can also be divided into four components:

- **Plan**: Test plan should include: test scope, approach, resources, activities, schedule, to define which format to use (test script).
  - The test script is based in the information provided by the software supplier [13].
- **Do**: Run the test script.
- **Check**: Check the results (test results) of the execution (meets or does not meet the expected result).
- **Act**: According to the results generate the plan of CAPA (corrective actions and preventive actions) for tests that does not meet the expected results.

Fig. 1 shows a testing process flow, definitions of each step are below.

Test Plan: A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning, GAMP 5® [7].

---

**Fig. 1** Testing process flow.
Test Script: Test script should contain the details of the tests. The test script should be described in sufficient detail to enable consistent repetition of the test GAMP 5® [7].

Test script should contain general information, specific test and expected results in order to verify compliance with a specific requirement.

Test Execution: Execution of the test based on the test script indication and test plan as reference.

Test Results: A set of results obtained during the execution of the test script.

The information to be retained should include: passed tests, failed tests, test failure records, test reports and any supporting documentary evidence required by the tests, such as printouts, screen shots, notes, and pictures GAMP 5® [7].

Corrective Action: Action to eliminate the cause of a detected non-conformity or other undesirable situation [14].

3. Results

The installation of the software on machines (M-01, M-02 and M-03) without using a testing methodology based solely on the experience of the installer responsible obtained the results were shown in Table 1.

The installation of the software on the machines (M-04, M-05 and M-06) with use of a testing methodology PDCA obtained the results are shown in Table 1.

4. Discussion of Results

In Table 7 the results are shown for analysis.

Using the methodology PDCA for testing scientific software installation, could give as result that we can get evidences; meet the expected results; meet the user acceptance and reach the user confidence in the system, according to results in machines M-04, M-05 and M-06.

Using a testing methodology based solely on the experience of the installer responsible not always is enough to get the expected results as in the machines M-01, M-02 and M-03 where the results weren’t meeting.

5. Conclusions

Scientific software installation should not be taken
### Table 3  Specif test section.

<table>
<thead>
<tr>
<th>Step action</th>
<th>Expected result</th>
<th>Test result compliance</th>
<th>Tester (initiales)/ Date (mm/dd/yy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Download the commercial installation file from the supplier web page.</td>
<td>Commercial installation files from the supplier web page downloaded</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>2. Open commercial file dot exe</td>
<td>Commercial file dot exe opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>3. Execute commercial file to start the installation software by giving double click.</td>
<td>Wizard opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>4. Click next on the wizard, read the license agreement</td>
<td>License agreement screen opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>If agree with license agreement accept the agreement if not do accept the agreement and skip to section comments</td>
<td>License agreement readed and accepted</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>5. Enter registration name and code and click next</td>
<td>Available languages screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>6. Choose English and click next</td>
<td>Available units screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>7. Choose all and click next</td>
<td>Select destination location screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>8. Take note of the destination route and click next</td>
<td>Select start menu folder screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>9. Take note of the shortcut name and click next</td>
<td>Select additional tasks screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>10. Choose create a desktop icon and create a quick launch icon and next</td>
<td>Multi user screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>11. Click OK</td>
<td>Open security alert screen</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>12. Allow access to commercial software into the computer</td>
<td>Open databases screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>13. Click servers</td>
<td>Edit servers is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>14. Click scan</td>
<td>Servers are scanned</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>15. Double click on server scanned</td>
<td>Date base name are shown on open database screen</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>16. Choose a data base and double click</td>
<td>Select user screen is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>17. Choose user expert and double click</td>
<td>Data base selected is opened</td>
<td>Yes</td>
<td>JADP 05/29/14</td>
</tr>
</tbody>
</table>

### Table 4  Evidences test section.

<table>
<thead>
<tr>
<th>Step action</th>
<th>Result</th>
<th>Tester (initiales)/ Date (mm/dd/yy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Take note of the screenshots taken</td>
<td>18 screenshots were taken and annexed</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>2. Take note of the pictures taken</td>
<td>None</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>3. Take note of the failures presented</td>
<td>None</td>
<td>JADP 05/29/14</td>
</tr>
<tr>
<td>4. Take note of general comments</td>
<td>None</td>
<td>JADP 05/29/14</td>
</tr>
</tbody>
</table>
informally because even if it already had been tested by the supplier, risks of a bad installation is present mainly due to lack of experience in that particular software by who installs it.

With the PDAC methodology for testing software, we are certain that the software is properly tested and if we have problems later, the testing evidence could serve as a reference to find a promptly solution.

Testing with PDAC methodology ensures:
- Evaluation regarding software installation is conform to its expected result;
- Evaluation of correct versions of the program is placed into production;
- Testing evidences usefully for maintenance purpose.

Software installation should be tested during the installation phase and not at the final product.

References


Scientific Software Installation Testing Using a Plan-Do-Check-Act Methodology: A Case Study

Information and Software Technology 56: 1597-612.


