J. Chem. Chem. Eng. 9 (2015) 227-232
doi: 10.17265/1934-7375/2015.03.010

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

Jorge Domínguez-Patiño[1*], Antonio Rodríguez Martínez[2], Rosenberg Javier Romero[2], Marco Cruz Chávez[2], Martha Lilia Domínguez Patiño[3] and Rosa María Melgoza Alemán[3]

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. FCQeI (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].

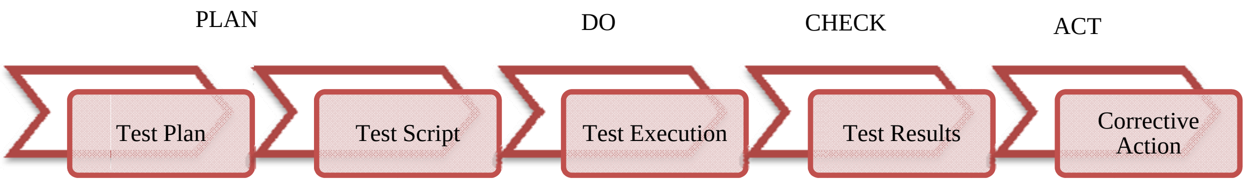

Fig. 1 Testing process flow.

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 1 Results obtained in the six machines evaluated.

Item	M-01	M-02	M-03	M-04	M-05	M-06
Licence agreement evidence	NO	NO	NO	YES	YES	YES
Selection lenguages evidence	NO	NO	NO	YES	YES	YES
Software location evidence	NO	NO	NO	YES	YES	YES
Server found	NO	NO	NO	YES	YES	YES
Data base found	NO	NO	NO	YES	YES	YES
User confidence	NO	NO	NO	YES	YES	YES
User acceptance	NO	NO	NO	YES	YES	YES
Software version loaded	DEMO	DEMO	DEMO	8	8	8

Test Script Results for: Machine M-04

Table 2 General information section.

Test title	Scientific software installation testing on client machine M-04
Test number	Client machine M-04
References	Commercial software installation plan
Test objective	Evalute the software installation in the client machine M-04, is according with the expected results in order to prevent failures
Machine code	100041529

Table 3 Specif test section.

Step	action	Expected result	Test result compliance Yes/No/Not Applied)	Tester (initiales)/ Date (mm/dd/yy)
1.	Download the commercial installation file from the supplier web page.	Commercial installation files from the supplier web page downloaded	Yes	*JADP* 05/29/14
2.	Open commercial file dot exe	Commercial file dot exe opened	Yes	*JADP* 05/29/14
3.	Execute commercial file to start the installation software by giving double click.	Wizard opened	Yes	*JADP* 05/29/14
4.	Click next on the wizard, read the license agreement	License agreement screen opened	Yes	*JADP* 05/29/14
5.	If agree with license agreement accept the agreement if not do not accept the agreement and skip to section comments	License agreement readed and accepted	Yes	*JADP* 05/29/14
6.	Enter registration name and code and click next	Available languages screen is opened	Yes	*JADP* 05/29/14
7.	Choose English and click next	Available units screen is opened	Yes	*JADP* 05/29/14
8.	Choose all and click next	Select destination location screen is opened	Yes	*JADP* 05/29/14
9.	Take note of the destination route and click next	Select start menu folder screen is opened	Yes	*JADP* 05/29/14
10.	Take note of the shortcut name and click next	Select additional tasks screen is opened	Yes	*JADP* 05/29/14
11.	Choose create a desktop icon and create a quick launch icon and next	Multi user screen is opened	Yes	*JADP* 05/29/14
12.	Click OK	Open security alert screen	Yes	*JADP* 05/29/14
13.	Allow access to commercial software into the computer	Open databes screen is opened	Yes	*JADP* 05/29/14
14.	Click servers	Edit servers is opened	Yes	*JADP* 05/29/14
15.	Click scan	Servers are scanned	Yes	*JADP* 05/29/14
16.	Double click on server scanned	Date base name are shown on open database screen	Yes	*JADP* 05/29/14
17.	Choose a data base and double click	Select user screen is opened	Yes	*JADP* 05/29/14
18.	Choose user expert and double click	Data base selected is opened	Yes	*JADP* 05/29/14

Table 4 Evidences test section.

Step action	Result	Tester (initiales)/ Date (mm/dd/yy)
1. Take note of the screenshots taken	18 screenshots were taken and annexed	*JADP* 05/29/14
2. Take note of the pictures taken	None	*JADP* 05/29/14
3. Take note of the failures presented	None	*JADP* 05/29/14
4. Take note of general comments	None	*JADP* 05/29/14

Table 5 Acceptance criteria section.

Acceptance criteria	The installation test results must be according with the expected result for each one of the specific testing
Result	All the installation test results were on compliance with the expected result

Table 6 Approval section.

Step	action	Responsible (name)	Responsible (initiales)/ Date (mm/dd/yy)
1.	Performer	*Jorge Domínguez*	*JADP* 05/29/14
2.	Reviewer	*Antonio Rodríguez*	*ARM* 05/29/14
3.	Approver	*Rosenberg Romero*	*RRD* 05/29/14

Table 7 Results for the six machines tested.

Machine	Licence agreement evidence	Selection languages evidence	Software location evidence	Server found	Data base found	User confidence	User acceptance	Software version loaded
01	NO	NO	NO	NO	NO	NO	NO	DEMO
02	NO	NO	NO	NO	NO	NO	NO	DEMO
03	NO	NO	NO	NO	NO	NO	NO	DEMO
04	YES	YES	YES	YES	YES	YES	YES	8
05	YES	YES	YES	YES	YES	YES	YES	8
06	YES	YES	YES	YES	YES	YES	YES	8

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

[1] Kanewala, U., and Bieman, J. M. 2014. “Testing Scientific Software: A Systematic Literature Review.” *Information and Software Technology* 56: 1219-32.

[2] Sommerville, I. 2004. *Software Engineering*. Seventh Edition. Boston: Addison Wesley.

[3] Beizer, B. 1990. *Software Testing Techniques*. New York: Van Nostrand Reinhold.

[4] Kaner, C., Falk J., and Nguyen, H. Q. 1999. *Testing Computer Software*. New York: Wiley.

[5] Myers, G. J. 1979. *The Art of Software Testing*. New York: Wiley.

[6] Juha, I., and Mika V. M. 2014. “Are Test Cases Needed? Replicated Comparison Between Exploratory and Test Case Based Software Testing.” *Empir Software Eng.* 19: 303-42.

[7] ISPE GAMP ® Good Practice Guidess. 2008. International Society for Pharmaceutical Engineering (ISPE), GAMP 5, A Risk-Based Approach to Compliant GxP Computerized Systems.

[8] Siegel, S. 1996. *Object Oriented Software Testing*. United States of America: John Wiley & Sons.

[9] General Principles of Software Validation; Final Guidance for Industry and FDA Staff. U.S. Department Of Health and Human Services, Food and Drug Administration, Center for Devices and Radiological Health and Center for Biologics Evaluation and Research. January 11, 2002.

[10] Mika, V. M., and Juha, I. D. 2014. “How are Software Defects Found? The Role of Implicit Defect Detection, Individual Responsibility, Documents, and Knowledge.”

Information and Software Technology 56: 1597-612.

[11] ICH Pharmaceutical Development Q8 (R2). 2009. International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH).

[12] Software: SimaPro 8.0.3. 2014. Classroom Multi user.

[13] Vincent, C., Michiel O., and Chris, G. 2014. Pré SimaPro, SimaPro 8.0 Installation Manual.

[14] ICH Pharmaceutical Quality System Q10. 2008. International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH).