INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 9(6):721-729, December, 2014.

Framework for Automated Reporting in EU funded Projects

A. Mihăilă, D. Benţa, L. Rusu

Alin Mihăilă*, Lucia Rusu
Faculty of Economics and Business Administration
Babes-Bolyai University of Cluj-Napoca
alin.mihaila@econ.ubbcluj.ro, lucia.rusu@econ.ubbcluj.ro
*Coresponding author: alin.mihaila@econ.ubbcluj.ro

Dan Benţa
Faculty of Law and Economics
Agora University of Oradea
dan.benta@univagora.ro

Abstract: Reporting processes in EU funded projects involve a huge amount of financial operations. This time consuming procedure can be easily automated as it involves repetitive operations. The novelty of the paper consists in the presentation of AutoFiState software application for automatic financial data capture in projects financed through ESF (European Social Fund) in Romania. The AutoFiSate application was developed on the basis of results obtained in the fields of automated testing and scripting languages. The use of the AutoFiState software application leads to maximum effectiveness in how ESF funds are used by reducing the time needed to draw up the financial reports and the related labor costs. Using scripting languages to develop such reporting-support programs we can improve reporting and save employees effort and time.
Keywords: automatic financial data capture, automated testing, scripting languages

1 Introduction

It is widely known that Romania has a very low degree of EU funds absorption. There are many reasons for the failure to absorb the EU's so-called structural funds and it is not the objective of this paper to discuss them. We will like to address just one of the problems and to suggest a possible solution in order to have projects with greater capacity of absorption. From our own experience with these kinds of projects, one of the main problems in EU funded projects conducted by the universities is the reporting procedure of financial evidence when the requests for reimbursement are made. This is a time consuming activity that needs high cognitive skills such attention to small details. In general, in a reporting period, several EU funded projects are handled by a single person, the financial official from projects' beneficiary. Moreover, financial evidence and reporting should be done by people with great expertise and most of the time these types of tasks cannot be transferred to another team member. We had to experience all this problems and that is why we were interested in this field and finally we managed to develop a software solution to divide reporting tasks in order to facilitate the work of financial officials in beneficiary offices.

Test automation [1], [2], [3], [4] is a significant area of investment and the market awareness of highly automated testing is very high. As some white papers present [5], [6] *you can get the most benefit out of your automated testing efforts by automating: repetitive tests that run for multiple builds, tests that are highly subject to human error, tests that require multiple data sets, frequently-used functionality that introduces high risk conditions, tests that are impossible to perform manually, tests that run on several different hardware or software platforms and*

configurations, tests that take a lot of effort and time when doing manual testing. Moreover, authors [6] describe best practices for script automation or for cross-site scripting attacks [7].

As reporting operations are repetitive tasks, using scripting languages to develop reporting-programs proved to be a great solution for many other businesses, inclusive for our own projects. In a market research report made in 2013 by Musier & Javed [8], automated testing delivers business benefits in multiple areas for most business companies. More than 4 out of 5 firms using test automation identified business benefits of test automation in multiple areas (86%), with most respondents identifying 3 to 6 different areas of benefits. The 5-top ranked areas of value were: 1) greater staff efficiency and time savings; 2) Early identification of defects before business users are impacted; 3) Higher quality in business processes and the software that supports them; 4) Greater accuracy in catching more defects; 5) Faster deployment of innovation and new features for business users.

The authors also mention that testing is increasingly seen as an essential competency for most of the global companies. More than that, harvesting economic benefits will continue to drive the industry's shift toward highly automated testing and away from manual approaches, as companies continue to push for higher quality execution and greater business agility at lower cost.

Beside the aspects mentioned, an important aspect for any work field is the dynamics of the group. This concept is an essential factor in long term projects as it can affect the project performance. Because some staff is short time employed and the same task may be allocated to several employees during the implementation period, aspects of group dynamics become even more important [9].

The reason tackled in this paper is the fulfilment of financial statement demanded when applications for reimbursement (refund) are submitted. The current legislation requires that, for each project implemented by a public institution, each application for reimbursement (refund) should be submitted no later than three months after the first payment is made [10]. Drawing up a funding application is a complex and time-consuming activity. In big organizations, and not only, where ESF-funded projects are implemented, the job of financial officials who draw up funding applications is very difficult and demanding. To support them during the stage of financial recording, the present paper proposes the automation of the process of data input in the reporting system of the managing authority in Romania for the Sectorial Operational Programme Human Resources Development. The reporting system used in Romania by the Managing Authority for the Sectorial Operational Programme Human Resources Development is an online software application called Action Web. Consequently, the automation of the process of data input into the reporting system represents the automation of the process of filling in a web form which, in turn, is similar to automated testing that is appropriate for [5] tests that are highly subject to human error and/or tests that take a lot of effort and time when performed manually. The main advantage of automatizing the financial reporting operations lies in greater staff efficiency and time saving [8].

The novelty of the paper consists in the presentation of AutoFiState software application for automatic financial data capture in projects financed through ESF in Romania. To develop the AutoFiState software application for the automatic data input into the Action Web to prepare the financial statement, use was made of results in the fields of automated testing [1], [2], [3], [4], [5], [6], [8] and scripting languages [6], [7], [11]. The structure of the present paper is as follows: Section 2-Problem statement; Section 3-Presentation of the solution to the problem and of the AutoFiState software application; Section 4-Presentation of tests and results; Section 5-Conclusions.

Within this framework and taking into consideration that a project can be very difficult to manage without proper instruments, we present our solution for data validation and reporting

process in order to avoid human reporting mistakes and to reduce data input time in EU projects. We also consider that our solution helps with the problems that can arise with the short time employment, helps to improve the dynamic of employees and to build trust between partners in EU funded projects.

2 Problem statement

In the case of ESF-funded projects implemented by public institutions, the lead partner must submit, no later than three months after the first payment is made during the reimbursement period, an application for reimbursement consisting of several documents among which is the financial statement. The financial statement contains data about all the expenditures made by all partners in the project during the reimbursement (refund) period. The data obtained from the financial officials of all partners are then entered by the project coordinator into the Action Web software which automatically generates the financial statement in PDF file format (Figure 1). For each type of expenditure made in the project (human resources, participants, other costs, indirect expenses), data must be entered for the documents certifying the commitment of expenditures (date of issue, number of documents, supplier, tax identification number, description, annual budget, budget line - expenditure category, payer, currency, VAT-inclusive price, VAT-exclusive price, VAT-exclusive project costs, VAT) and for the documents certifying the payment was made (date of payment, number of payment document, currency, VAT-inclusive price, VAT-exclusive price).

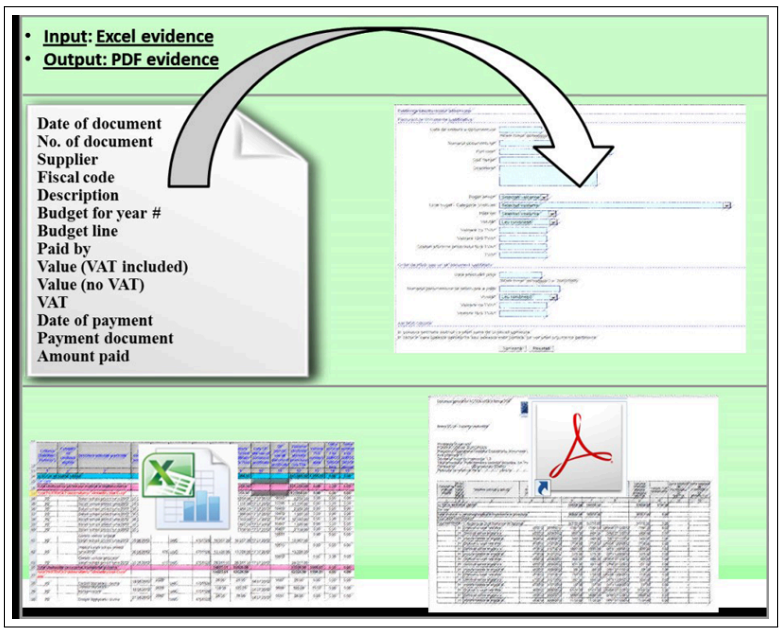

Figure 1: Input and output for reporting

Given that:

• data input into the Action Web application is a time-consuming operation,

• the data to be entered into Action Web follow the same structure/format for all types of expenditures, and

• the Action Web application does not feature the automatic data capture from a file of a particular format/structure, the opportunity was opened up to develop a software application entitled AutoFiState that will be used to automatically enter data into Action Web.

By using the AutoFiState software application:

• the time needed to enter data into Action Web is reduced, and

• the time needed to identify and correct errors made when entering data into Action Web is also reduced.

As a direct consequence thereof, the labor costs for drawing up each financial statement are reduced.

3 Software Design and Implementation

The preparation of a financial statement using the AutoFiState software application is accomplished in three stages (Figure 2):

• The preparatory stage during which the financial officials of all partners in the project prepare the financial data related to all expenditures made by partners and the project coordinator centralizes financial data collected from the financial officials of the partners;

• The AutoFiState stage during which the project coordinator automatically enters the centralized financial data into the Action Web application;

• The final stage during which the financial official of the lead partner (beneficiary) generates the financial statement for the entire project and prepares it to be approved by the legal representative of the project.

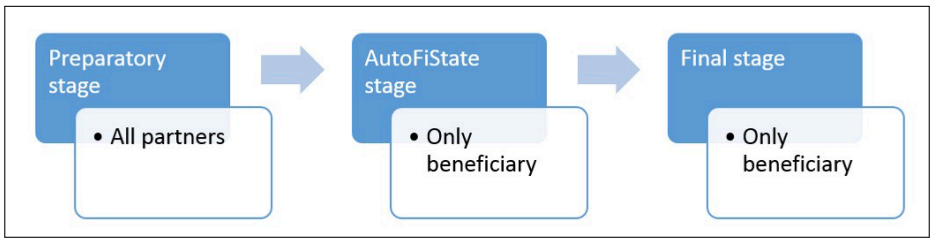

Figure 2: Financial statement generation phases

All mentioned steps involve more than one employee and the beneficiary financial officials' job is to check the whole evidences and costs asking for supporting documents. In our case, we have developed and tested this solution, for the first time, in an EU funded project with five partners (our university as beneficiary and four other partners). We are describing in detail the performed operations in a reporting period.

a) During **the preparatory stage** (Figure 3), the financial officials of each partner draw up their own financial statements by entering the data about each category of project expenditures into an Excel template file. Some data must be entered into the Excel template in the format required by the Action Web application. For instance, calendar dates must be entered in the "DD/MM/YYYY" format and the numeric data should not use any separator except for the dot (".") character. The financial official of the beneficiary receives the financial statements from each partner and checks the eligibility of all project expenditures on the basis of supporting documents. After the eligibility of expenditures for all partners has been checked, the financial official of the beneficiary forwards the Excel templates containing the related financial statements to an operator who automatically enters the data into Action Web using the AutoFiState application.

b) During **the AutoFiState stage** (Figure 4), an operator of the beneficiary checks whether the data in the Excel template file containing the financial statements of each partner comply with the formats of the Action Web application. If some data are entered not according to the required format, the operator processes the data so that they may comply with the format required by the Action Web application. After the data in all Excel template files have been found to comply

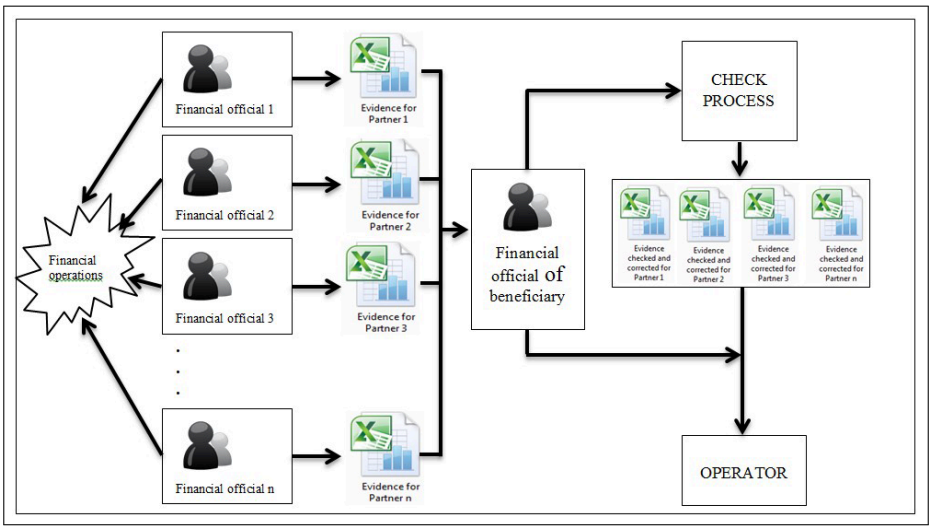

Figure 3: The preparatory stage

with the format required by the Action Web application, they then are transferred to a central Excel file. All fields of the central Excel file are set at identical height and width because the AutoFiState script uses the screen coordinates to copy data from the central Excel file into the Action Web application. While the operator inputs all the records, beneficiary financial official can perform other mandatory operations and prepare the requested for the reimbursement (for instance, documents for the external auditor).

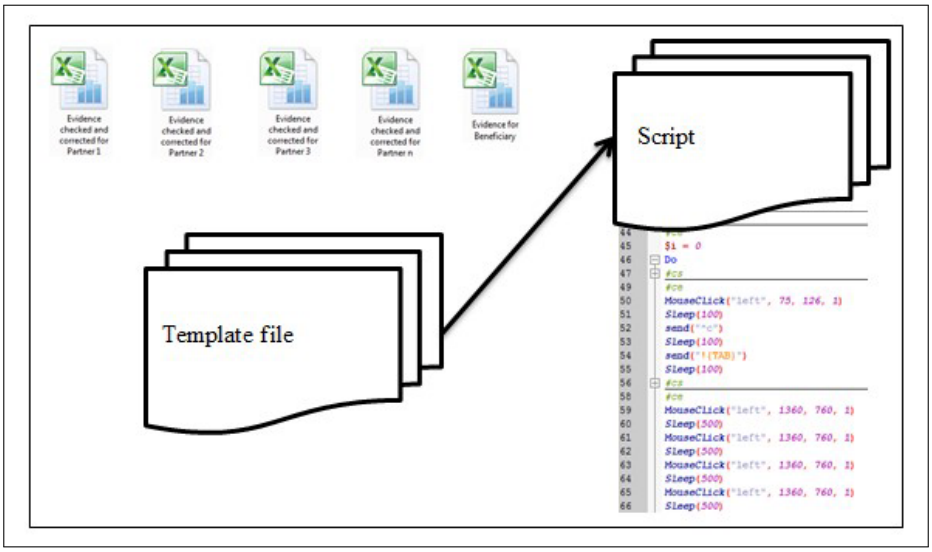

Figure 4: The AutoFiState stage

The AutoFiState application reads the central Excel file row by row so that at a given moment a single row is active or current. Initially, the current row is represented by the first row of the central Excel file. The AutoFiState application copies/captures the data from the current row into the Action Web software. The data from the current row are copied cell by cell, from the left to the right of the row. After having copied the data from the current row, the AutoFiState application waits until the data are saved on the server of the managing authority and sets the next row of the central Excel file as the current row. The new current row is set by clicking on the

vertical scroll bar so that the next row of the central Excel file becomes the first row displayed on the screen. This process is repeated until all rows of the central Excel file are copied into the online application of the managing authority. The AutoFiState application should be run under supervision because either the computer or the Action Web application or both are sometimes slow in responding. In this case, the AutoFiState application will be closed and then reopened from the record that failed. Closing and reopening the AutoFiState application is made using the hotkeys/predefined keys from AutoIT scripts.

To develop and use the AutoFiState application, the following software applications were used:

• Microsoft Office Excel, 2010 version, for data preparation by the financial officials of all partners to be entered into the Action Web software;

• Mozilla Firefox, version 15.0.1, to access the Action Web application;

• Auto IT scripting package, version v3.3.8.1, to develop and use effectively the AutoFiState application (during the development stage, Window Info Cursor Position Tool proved extremely helpful).

In order to properly use the AutoFiState application, several prerequisites/conditions must be met:

• Only Microsoft Excel file and reporting platform should be opened - for our test we used Office 2010 package;

• Use Mozilla Firefox to open reporting platform because Full Screen result differs when IE or Google Chrome browsers are used;

• Reporting platform should run in Full Screen (clicking coordinates were defined in an Full Screen operation);

• Minimize the Ribbon should be activated in Microsoft Excel 2010;

• Auto-hide the taskbar option should be checked in Taskbar and Start Menu Properties.

Those restrictions assure a proper script running and reporting according to Excel file records. With this pre-settings made, the script runs.

Main operations performed by reporting script (Figure 5) are coordinated by hotkeys in AutoIT scripts. A hotkey press for such scripts calls a user function that may pauses or interrupts current AutoIT function. Most used and defined hotkeys in AutoIT scripts are for pause: HotKeySet("PAUSE", "TogglePause") and for terminate: HotKeySet("ESC", "Terminate"). In our case, we have used "ESC" and terminate function to stop script running (Figure 5a). In case of processing delays or server slow response, the script can be stopped and restarted from the last selected record. A message window with some information is defined, then, the script reporting phase is initialized (from 0 to n-1, where n is the number of records from Excel file). Further, the script runs and each record from Excel file is inputted in the reporting web form (Figure 5b). After last record, an increased waiting time (sleep time) is defined in order to secure the web server response time and loading reporting form to enter the next record (Figure 5c). The application crosses financial evidence file in row order. Then, all specific operations for the first row are applicable to the next one. To assure this, after finishing a row, the next row should be repositioned. For this to happen, the application clicks once the roll down bar from Excel so that second row became first displayed row. This process repeats until the last record is inputted in the reporting web form.

c) The automatic financial data capture from the central Excel file into the Action Web application is followed by the final stage (Figure 6) during which the financial official of the beneficiary generates the financial statement from the online application of the managing authority and submits it for approval by the legal representative. Before report generation, incomes must be inputted. Later, along with other documents, the reimbursement (refund) request is send to authorities.

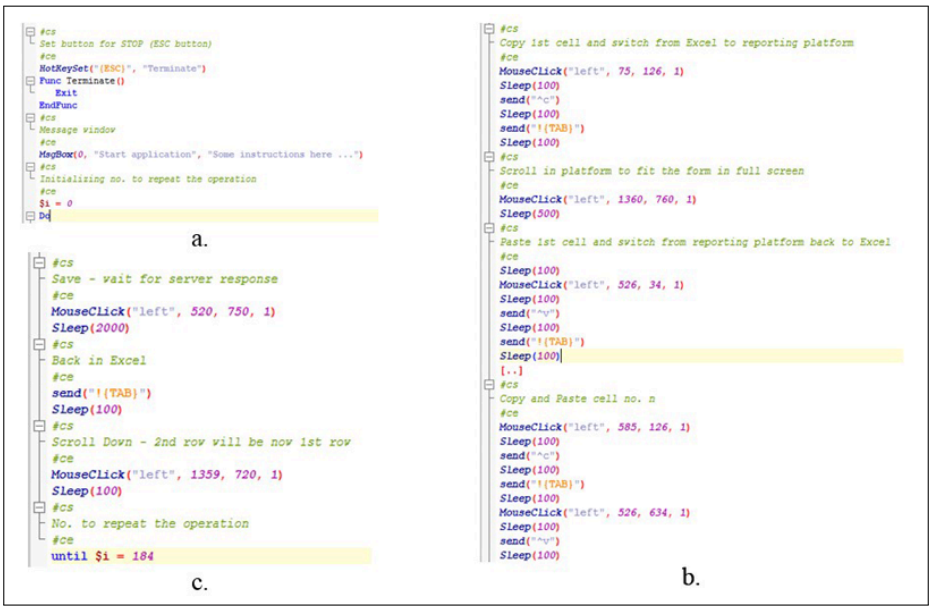

Figure 5: AutoIT script: a. Safety settings; b. Automated reporting; c. Save and repeat.

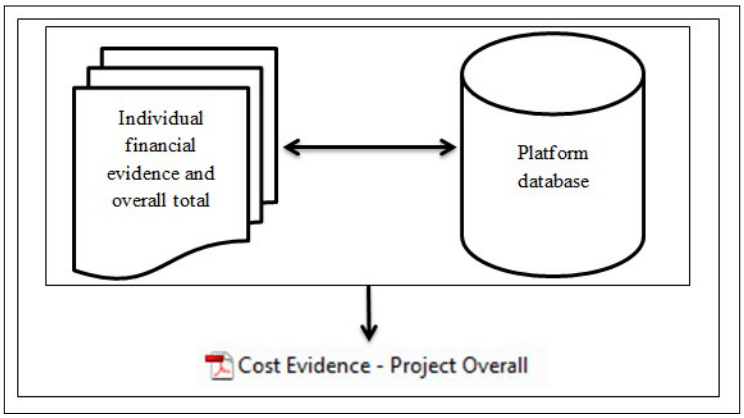

Figure 6: The final stage

4 Tests Results

The AutoFiState application has been used with four ESF projects financed through the Sectorial Operational Programme Human Resources Development:

Only the AutoFiState software application was used to enter the data because, for each project, the amounts of data were relatively large and the available time needed to draw up a funding application and prepare a financial statement was not sufficient as regards entering the data into the Action Web application both manually and automatically.

Leaving aside the time needed to discover possible mistakes due to manual data entry and correct them, it has been noticed that 142 hours have been saved by using the AutoFiState application. Since the gross hourly salary of a financial official amounts to 150 lei, then the saving amounts to 21300 lei.

Table 1: Data test

Project ID	No. of partners	No. of months	Budget (in million Euros)	Status
1	5	36	5	Closed
2	2	29	4,5	Ongoing
3	2	36	3,5	Closed
4	9	36	5	Closed

Table 2: Test results

Project ID	Manual time* (in hours)	AutoFiState time (in hours)	Manual error	AutoFiState error	No. of records
1	111	40	$\geq 0\%$	0%	4767
2	30	11	$\geq 0\%$	0%	1301
3	23	8	$\geq 0\%$	0%	978
4	57	20	$\geq 0\%$	0%	2435

5 Conclusions

Automated reporting shortened our reporting period and improve the quality of reporting tasks. The current version we developed is good enough to make the reporting procedure of all stored records in an efficient and clear manner.

There are various testing and automation software solutions that may improve such repetitive reporting tasks. The implemented solution constitutes a solid foundation for our future work.

As web browsers update is regular made, sometimes defined coordinated doesn't perfect match and should be redefined. A solution to resolve this problem may be implemented. This solution have two major advantages: saved financial official time with more than 50% and offers dynamic collaborative framework for projects' partners. It can be applied to all EU funded projects that need this reporting procedure and can also be applied successfully in national research projects (ANCS, CNCSIS etc.).

Reporting procedures in EU funded projects needs careful planning and design work as a huge amount of time and effort is involved. Using this reporting-support application, a full range of requests is considered. The software enhances the reporting process by increasing data validation and efficiency, removing reporting complexity and lowering costs. It can be adapted and customized for different reporting procedures in projects when repetitive data input operations are needed.

The use of the AutoFiState software application leads to maximum effectiveness in how ESF funds are used by reducing the time needed to draw up the financial reports and the related labor costs. The AutoFiState software application can be improved to the point where it no longer depends on the regular updates of the web browser. The AutoFiState application can be used for all ESF-funded projects in Romania, whose financial reports are forwarded to the managing authority via the Action Web application. The AutoFiState software can also be slightly adjusted so that it can be used successfully in other EU-funded or government-funded projects.

Acknowledgement

The work presented has been founded by the research Grant FP7 AAL NITICS and was financed by IT Center for Science and Technology, Romania, as a partner in this program.

Bibliography

[1] Jureczko, M. (2008). The Level of Agility in the testing process in a large scale financial software project. Software engineering techniques in progress, Oficyna Wydawnicza Politechniki Wroclawskiej, 139-152.

[2] Li, K., & Wu, M. (2004). Available GUI Testing Tools vs. Proposed Tool. *Effective GUI Test Automation: Developing an Automated GUI Testing Tool*. SYBEX.

[3] Myers, G. J., Badgett, T., Thomas, T. M., & Sandler, C. (2004). *The Psychology and Economics of Program Testing. The Art of Software Testing-Second Edition*, John Wiley & Sons, ISBN 0-471-46912-2..

[4] Dustin, E. (2003). Automated Testing Tools. *Effective Software Testing: 50 Specific Ways to Improve Your Testing*. Pearson Education, ISBN 0-201-79429-2, 137-154.

[5] Juniper Networks (2008), Increasing Network Availability with Automated Scripting, Available on-line http://support.neoteris.com/solutions/literature/white_papers/200252.pdf

[6] SmartBear Software (2013), 6 Tips to Getting Started with Automated Testing, White Paper, Available on-line http://smartbear.com/SmartBear/media/pdfs/6_Tips_for_Automated_Test.pdf

[7] Van Acker, S., Nikiforakis, N., Desmet, L., Joosen, W., & Piessens, F. (2012). FlashOver: Automated Discovery of Cross-site Scripting Vulnerabilities in Rich Internet Applications, ASIACCS '12, May 2-4, 2012, Seoul, Korea, ACM.

[8] Musier, R., Javed, S. (2013). 2013 Trends in Automated Testing For Enterprise Systems, pp. 3-4, Available on-line http://www.worksoft.com/files/resources/Worksoft-Research-Report-2013-Trends-in-Automated-Testing.pdf

[9] Levi, D. (2011). *Group Dynamics for Teams*. Sage Publications, 3rd Edition, ISBN: 9781412977623

[10] Romanian Government Emergency Ordinance no 120 from December 23, 2010, Art. 5^1.

[11] AutoIT, http://www.autoitscript.com/site/autoit