Factors with Negative Influence on Software Testing Practice in Spain: A Survey

Luis Fernández-Sanz[1], M. Teresa Villalba[2], José Ramón Hilera[1], and Raquel Lacuesta[3]

[1] Depto. De C. de la Computación, Universidad de Alcalá, Ctra. Madrid-Barcelona Km 33,600, Alcalá de Henares, 28871, Madrid, Spain
[2] Depto. de Sistemas Informáticos, Universidad Europea de Madrid, C/Tajo s/n, Villaviciosa de Odón, 28670, Madrid, Spain
[3] Dept. of Comp. and Syst. Eng., Univ. de Zaragoza, C. Escolar s/n, 44003 Teruel, Spain
luis.fernandezs@uah.es, maite.villalba@uem.es, jose.hilera@uah.es, lacuesta@unizar.es

Abstract. Software testing is the commonest technique for software quality assurance. It is present in every development project and concentrates a large percentage of effort, there are still not many studies which address the real practice of individuals and organizations. Anyway, practitioners usually agree with the idea that software testing efficiency and effectiveness in their organizations might be improved. Two previous studies in Spain have revealed implemented testing practices in organizations and individual performance of software professionals when designing test cases should be improved. This paper presents the results of a survey designed to know if 23 factors determined by a panel of experts in 2007 may explain this situation of testing practice. Data collected reveal that none of the factors is clearly rejected as a negative influence for testing although some of them are not generally accepted. Exploratory statistical analysis reveals relations between certain pairs of items as well as a new grouping in factors.

Keywords: Software testing, survey, influence factors.

1 Introduction

Software testing is the commonest techniques for verification and validation in development projects. Every project includes a specific phase for testing and debugging. According to different statistical studies of effort distribution throughout the life cycle [1][2][3], this phase usually requires around a large percentage, around one-third (ranging from 30 to 35%), of the total effort of the project.

Different studies have tried to analyze real practice but in many cases empirical works are focused on analyzing or demonstrating the benefits of specific methods or approaches to testing. As stated in [4], there is a need of real practice empirically-based data not vested by such purpose but aimed at providing more light on this area. This type of studies is rare although, as can be seen in the following sections, there are interesting contributions. In order to gain knowledge in this area, a series of studies centered

R.V. O'Connor et al. (Eds.): EuroSPI 2009, CCIS 42, pp. 1–12, 2009.

on software testing practices in Spain were launched in 1999 by L. Fernandez-Sanz. It began with a survey on testing practices in organizations (see Section 2) which finally collected information from 210 software professionals. After analyzing results which reveal a weak situation for organizations and as suggested by respondents, a specific study on 72 individual practitioners' performance in test case design (see Section 3) was carried out to control if professionals might get good results despite poor organizational environment. This study concluded that individual performance was also weak so as a final step a survey was launched to discover the underlying causes. This paper is focused on this final stage presented in Section 4 although a brief presentation of the two first studies is included in Sections 2 and 3. Finally, section 5 discusses results and conclusions as well as future works.

2 Analysis of Testing Practices in Organizations

In order to know something more about which the real testing practices of software organizations in Spain are, a study was carried out by the Software Quality Group of ATI (www.ati.es), the main computing professionals association in Spain, the national body of CEPIS (www.cepis.org), the Council of European Professional Informatics Societies. This study (partly published in [5]) collected, during the period 1999-2007 information, from 210 IT professionals engaged in software development projects in Spain corresponding to almost all the activity sectors as well as many different positions (see table 1). Data were collected using anonymous questionnaires during specific events (like training courses, both in-company and open access, and QA events) as well as exploiting direct relations with IT professionals in companies.

Table 1. Respondents in the study of testing practices in organizations

Sector	%	Position	%
Finance	14.3%	Tester	16.2%
Consultancy	12.8%	Analyst	12.8%
Telco/IT	10.4%	Project manager	11.1%
Energy/industry	5.2%	Manager	9.4%
Transportation/Airlines	4.3%	Software engineer	8.5%
Defense	4.3%	QA specialist	5.9%
Government	3.8%	Programmer	5.9%
Tourism	3.8%	IT director	5.1%
Health	2.3%	Others	25.1%
Others	38.8%		

Although different process models (such as CMMi[1] [6][7],TMM [8], TMMI [9][10], TPI [11] y TMap [12][13].) are applicable to testing and include specific practices, only recently [14] description for testing process improvement have been analyzed in a rigorous way. Conclusions of this study reflect that there is not a complete and well described set of practices in those models so a quick method to collect

[1] As stated in the areas of Product Integration, Validation and Verification.

information from a wide range of organizations is not available for a survey. Obviously, data from CMMI evaluation or similar activities would give information on real testing practice. However, it is difficult to access to details of such evaluation processes, only a small percentage of organizations have been evaluated according to this model and SEI public information does not include details of each process area.

Knowing the limitation, we decided to use as reference the list of best practices for software testing of Quality Assurance Institute (www.qaiusa.org): one of advantage is that QAI carried out several surveys using this list from 1994 to 1999 in the USA so this reference of 20 practices was refined with their results and experience. In fact, published surveys do not focus their attention in specific process models but in customized list of questions covering from detailed techniques to organizational topics. In the case of [15][16] items covered from general testing approach (independent testing, etc.), budget or standards to specific methods and possible barriers to adoption of specific practices. Other studies [17] were more focused on detailed methods (e.g. use of notations like UML, structured diagrams, etc. for documentation) and specific data on percentage of effort devoted to testing even related to project size; in the case of [18] (also based on contacts of a network of practitioners and researchers) it was focused on extremely detailed aspects of software unit testing although some conclusions might be common to general testing practices.

Table 2. Summary of results from survey on testing practices implemented in organizations

QAI practice	Implem
1. Identified responsibility for testing processes in the organization?	28.57%
2. Is there and is used a standard for test plans?	23.33%
3. Is there and is used a standard for unit testing?	18.10%
4. Is there and is used a standard for test reports?	27.14%
5. Testing planning and execution process parallel to the whole development process?	28.57%
6. Check if software specifications are correct?	39.05%
7. Besides being correctly implemented. check if customer expectations are fulfilled?	48.57%
8. Testing staff check if development documents are complete and correct?	21.43%
9. Testing staff report defects to developers (and not to managers)?	41.43%
10. Testing staff identifies business risks before developing test plan?	11.43%
11. Are there measurable objectives for each tested system?	14.76%
12. Testing objectives are clearly linked to business risks?	14.29%
13. Are detected defects recorded, reported and used to improve development and testing processes?	28.10%
14. Has testing staff defined defect expectations according to paste experience?	17.62%
15. Is there a testing processes improvement process?	18.10%
16. Defects are identified with a unique code?	20.95%
17. Does the organization record, report and used defect data to asses test effectiveness?	17.62%
18. Are metrics used for planning and evaluating testing processes?	9.05%
19. Are there specific training processes for the testing staff?	17.62%
20. Do testing tools represent a significant element of testing process?	12.50%

Organizations			Implem
No. Practices from 0 to 4	10.95%	No. Practices from13 to 16	19.05%
No. Practices from 5 to 8	31.90%	No. Practices from17 to 20	13.33%
No. Practices from 9 to 12	24.29%		

Results updated to 2007 from the survey [5] are summarized in Table 2: each practice is described with a short description. Although QAI suggests that organizations would be classified in a scale of five levels according to the number of practices implemented, we think this is scheme is not rigorous enough although results are shown below as an indicator of aggregated number of practices per respondent. The survey also included information on two additional items:

- Specialized training in software testing: only 30.61% of respondents had attended such training. Similar formal [16] and informal surveys referred to testing training revealed slightly higher percentage of those with training).
- Relationship between training and answers to questions on testing foundations: number of people with specific training who passed the questions is twice the number of those with no training at all.

Looking at the result in table 2 with general low percentages in all items, it is clear there is still a wide margin of improvement for software testing practices: poor testing practices are not exclusive of Spain as can be seen in [14][15][16]. Trying to go further, we wanted to investigate if individual performance of testers would be good despite the weak organizational practices so we devised a specific study (Section 3).

3 Analysis of Individual Performance in Test Case Design

To check if software professionals were good at designing functional test cases, a small size case study (4 use cases) was created to control such activity with selected IT professionals contacted in seminars and events. The problem to be solved was the design of test cases for a basic DVD list management application where several defects were injected. A website with the following features was created:

- Access to the natural-language specification for the application
- Collection of anonymous data of the participants: position, sector, experience, etc.
- Interactive recording of test cases with options to create cases, to "execute" (by simulation) showing the list of stored DVDs after it, to determine if a defect is detected and to review the list of "executed" cases.
- Recording of time devoted by each participant.
- Presentation of a list of suggested correct test cases (to assure full coverage of the application) and recording of priority of each test case suggested by each participant according to his/her vision of the program objectives.

Table 3. Participants in the study of individual test case design

Sector	%	Position	%
Consultancy and IT	36.1%	Researcher	28%
Education	12.4%	Tester	27%
Internet	16.2%	Project manager	17%
Energy/industry	9.8%	Software engineer	13%
Finance	5.5%	Programmer	9%
Government	5.5%	Systems analyst	6%
Transportation/Airlines	1.5%		
Others	29.2%		

This experience had also a secondary objective as a check of the acceptance of test generation based on UML activity diagrams: "correct" solution presented at the end of the experience was generated using this method. The sample of 71 IT professionals (discarding unreliable tryouts and incomplete data) who participated in the first wave is shown in Table 3. Average development experience of respondents is 5.6 years and average time devoted to the experience was 27 minutes.

Results were presented at [18] and [19] and they can be summarized as follows:

- Only 1 participant covered more than 75% of the options of the program. 70.4% of participants did not reach the 50% of coverage of functional options, 13% did not detect any of the 4 injected defects and 40.8% detected at least 3. As an average, 50% of defects were detected and participants claim detection of 8 not real defects.
- As an average, around 50% of the cases designed by a tester were oriented to test program options previously controlled in similar cases executed by him/her. This was especially intense in test cases oriented to enter data in the program (e.g. insert new DVD data) rather than when deleting or modifying records.
- On one hand, among the 10 most executed test cases, there was only one of the ten most important ones according to participants' own rank of priority. On the other hand, among the 10 least executed cases, there were 3 of the most important ones.
- As additional information, it was also shown that practitioners considered a trade-off to invest in detailed UML models like Activity Diagrams for software specifications in order to gain productivity and effectiveness in test case generation using the AQUABUS method and its associated Eclipse plug-in [19].

These results reveal a weak situation and an opportunity for improving both effectiveness and efficiency through a more systematic design of cases. Nor organizational practices neither individual abilities of developers offer good results for productivity and quality in software so our next logical step was the investigation of possible causes of this situation: a detailed study was launched in 2007.

4 Survey on Factors Which Influence Testing Practice

Although some information on which is the state of practice in software testing is available, it is really difficult to find analysis on which can be the causes of the situation. In general, it is possible to locate articles (e.g. [20] [21]) based on subjective personal analysis of experts analyzing or explaining the contributing factors that impede efficient and effective application of software testing best practices. However, it is difficult to find works based on evidences, quantitative data or, at least, analysis of experiences (e.g. [22] or [23]) although specific surveys (e.g. [15]) have included questions on which are some of the barriers for better performance in software testing. Another interesting approach is the use of ethnographic methods to capture and analyze the work of software developers in projects [4] as they allow realizing a distance between theory and practice exists in real testing practices.

In our case, analyzing the results in organizations (Section 2) and the ones of individual performance in test case design (Section 3), we decided to investigate the possible causes of such situation. As part of the research network REPRIS (focused on software testing in software engineering and funded by the Spanish Ministry of

Science), we exploited the opportunity of promoting a debate with a panel of experts from industry (9 specialists) and academia (16 researchers) during a workshop hold in Zaragoza (Spain) in 2007. After an intensive session and refining and consolidating conclusions (reviewed by participants), 23 factors arose as possible causes of problems in software testing real practice. Although it was an interesting result, we decided to check if software professionals in Spain really confirm such factors were applicable to their professional environment. A questionnaire was created (see Table 5) where respondents had to indicate if they consider each item an effective factor of influence; they also ranked influence in a three-level scale: total/partial/ none.

Table 4. Factors of influence as items of questionnaire with ID for factorial analysis

Id	Factor
Q1.1.	When delays or finance problems appear, it is usual to shorten quality and testing effort.
Q1.2.	It is not strange a QA position disappears transferring people to software development roles
Q2.1	Testing is not creative: it is something annoying and not attractive which you have to do. Even it is negative (looks for defects) and destructive (goes against developers' work)
Q2.2	It is an area without good opportunities of career development or promotion.
Q2.3	Career development in testing does not guarantee the same salary or conditions as in other professional careers in software development (even you may expect worse conditions).
Q2.4	It is not usually recognized this work on testing, it is not usually accounted to be paid by customers, it is usually an internal service with no direct relationships with customers, etc.
Q2.5	Low level testers do not require a university degree, maybe only a basic professional education so this tend to project an image of not attractive professional career
Q3.1	Many IT university graduates have not attended specific training on testing
Q3.2	Many IT professionals have not also received specific training on testing
Q3.3	Courses on software testing are not usual in company training programs for software professionals (more focus on technology, new versions of products or in software development methods)
Q3.4	Testing is not a hot topic in universities: many teachers mention it and encourage students to do but few of them understand the correct philosophy and techniques of testing.
Q3.5	Specific testing training tend to focus on unit/detailed testing while functional/system testing is addressed as a marginal topic
Q3.6	Low importance or absence of specific training/qualification in testing (materials, certifications, etc.)
Q4.1	Junior tend to focus on programming and code: reject to work in other activities like testing
Q4.2	Many managers did not attend good training on software testing so they do not appreciate its interest or potential for efficiency and quality
Q5.1	People tend to execute testing in an uncontrolled manner until the total expenditure of resources in the belief that if we test a lot, in the end, we will cover or control all the system
Q5.2	It is not usual to plan and design efficient cases with minimum cost or to link tests to priorities or risks; there is not control on incurred risks depending on tests, no control of evidences, etc.
Q5.3	Test design usually means a rework of what analysts did not completed or documented because testing is totally dependent on a good requirements specification
Q5.4	Software test phase is located at the end of the project suffering shortened schedule due to delays of the previous development phases and the impossibility of postpone delivery to customer
Q5.5	Relationship between software models and testing is not exploited, specially for test design: "testing is something we do at the end once we have code"
Q5.6	It is not usual to design tests once we have a specification (although it is possible to do it in parallel with analysis): in fact, they document knowledge on functionality and requirements
Q6.1	Market is not mature enough so certain software quality problems are not sufficiently penalized
Q6.2	Best business is possible when customer pays maintenance of defects delivered with the developed software (getting money for repairing defects one has created)

Table 5. Respondents in the survey of factors of influence on software testing

Sector	%	Position	%
Government	22.9%	Project manager	21.8%
Telco/IT	21.8%	Tester	14.6%
Consultancy	14.5%	Manager	14.6%
Finance/insurance	9.8%	Programmer	12.5%
Defense	5.2%	QA specialist	9.4%
Tourism	5.2%	Systems analyst	7.3%
Health	3.1%	Others	20.6%
Transportation/Airlines	3.1%		
Others	14.4%		

Again, through direct contact with software professionals in events and training courses, we got a varied sample of 127 practitioners to collect opinion on the proposed list of factors of influence (see Table 4). The following sections will present both the general descriptive results and the detailed statistical data analysis.

4.1 Descriptive Data

As a first step, a simple descriptive analysis of data is done. Percentages of respondents who chose each option (i.e., confirmation of factor as a fact in professional settings and each level of possible influence) are presented in Table 6.

Table 6. Results of survey on factor of influence on testing

		Rank of influence					Rank of influence		
Factor	**Confirm.**	**Total**	**Partial**	**None**	**Factor**	**Confirm.**	**Total**	**Partial**	**None**
Q3.1	96,1%	40,94%	39,37%	19,69%	Q5.2	74,0%	40,94%	33,07%	25,98%
Q3.2	93,7%	39,37%	40,16%	20,47%	Q2.4	71,7%	23,62%	37,80%	38,58%
Q5.4	92,1%	61,42%	22,05%	16,54%	Q6.1	70,1%	31,50%	40,94%	27,56%
Q1.1	90,6%	61,42%	25,98%	12,60%	Q2.5	66,9%	25,98%	37,01%	37,01%
Q3.3	90,6%	37,01%	44,09%	18,90%	Q3.5	63,8%	30,71%	33,86%	35,43%
Q3.4	85,8%	30,71%	44,09%	25,20%	Q5.1	58,3%	29,13%	33,86%	37,01%
Q4.2	85,8%	48,03%	30,71%	21,26%	Q6.2	57,5%	31,50%	28,35%	40,16%
Q5.5	85,0%	48,82%	26,77%	24,41%	Q2.3	54,3%	14,96%	35,43%	49,61%
Q5.3	80,3%	40,16%	33,07%	26,77%	Q1.2	48,0%	30,71%	25,98%	43,31%
Q3.6	78,0%	33,86%	44,09%	22,05%	Q2.1	48,0%	25,20%	35,43%	39,37%
Q5.6	78,0%	30,71%	45,67%	23,62%	Q2.2	41,7%	18,90%	35,43%	45,67%
Q4.1	77,2%	29,92%	41,73%	28,35%					

As can be seen in Table 6, only three factors are not confirmed by at least 50% of respondents: Q1.2 (unstability of QA positions), Q2.1 (testing is not attractive) and Q2.2 (poor career development). However, in our opinion, they should not be rejected because there are a significant percentage of respondents supporting the idea. Another

group of factors (Q 53.6, Q 5.6, Q 4.1, Q 5.2, Q 2.4, Q 6.1, Q 2.5, Q 3.5, Q 5.1, Q 6.2, Q 2.3) have a greater proportion of support although an important percentage of people are not convinced of their presence in professional environments. And finally, a group of factors (Q 3.1, Q 3.2, Q 5.4, Q 1.1, Q 3.3, Q 3.4, Q 4.2, Q 5.5, Q 5.3) have a confirmation percentage above 80% so they can be considered as real facts in the software development world. As an additional action, for some of the respondents (33), we collected information about testing training. Combining with the first survey (Section 2), a global 36% of professionals have attended specific testing training.

4.2 Detailed Analysis of Results

A first objective is the validation of the questionnaire used to collect data analyzing correlations between pairs of questions. First of all, reliability of the scale should be tested through the Cronbach´s alpha coefficient. This measure helps to verify if the questions are related between them, i.e., all the questions measure the same concept. Then factorial analysis will be applied to verify which concept measures each group of questions. This allows determining the structure of the scale [24].

4.2.1 Previous Analysis

Exploratory analysis enables the detection of possible errors during data collection as well as the checking of feasibility of factorial analysis. Subsequently we examine descriptive statistics (means, standard deviation, median, minimum and maximum and absolute and relative frequencies) of all the variables in the study. Moreover, box plots can help to determine data entry errors and the coefficient of variation can be used to check the homogeneity of data. The correlation matrix gives information about factorial analysis applicability: correlations higher than 0.30, significance levels and determinants close to 0 shows there are correlated variables [25].

SPSS 16.0.1 and LISREL 8.80 statistical programs have been used to analyze the collected data. A first visual inspection of correlation matrix showed us that there was an essential number of correlations higher than 0.30; consequently, we concluded that there were interrelated variables [25]. Moreover, as almost all significance levels are close to zero, we had to reject the null hypothesis and concluded there was linear relationship between the variables. The determinant is near zero too (9,06E-005): it confirms these variables are highly correlated so factorial analysis is applicable.

4.2.2 Reliability Analysis

The first validation is the reliability analysis of the used factors. The reliability is the degree in which the observed variable measures the real value and is free of error. The reliability analysis includes the examination of corrected item-to-total correlations to find out if each factor measures the same issue than the rest of the factors. We eliminate specific items to improve reliability alpha coefficient. In this case, questions 1.1 (Q1.1) and 3.5 (Q3.5) have been eliminated. The final Cronbach´s alpha coefficient value after these eliminations is 0.908 demonstrating high consistency and reliability of the model[2].

[2] The conventional minimum for Cronbach's alpha coefficient is established in 0.7 [26].

Q1.1 had provoked some comments during data collection because some people were reluctant to recognize this practice. Comparing with descriptive analysis (section 4.1) Q1.1 was confirmed as a real fact (90.6%) and its influence on testing is high enough (only the 12,60% rank influence as none). Although it is an important factor, it is not related to the scale here presented: maybe this question should be better formulated. Anyway more research is needed to verify the importance of this factor.

Q3.5 experienced problems due to people without specific testing training. 63.8% of the respondents considered it as a real fact but the influence on testing is not clear because the data are very similar for the three categories.

4.2.3 Exploratory Factorial Analysis (EFA)

EFA enables to identify the underlying structure of the relations obtaining a predictive validation of the model. To investigate acceptability of factorial analysis results the

Table 7. EFA Results for the model

	EFA Loadings (after varimax rotation)[a]				
Factor	C1	C2	C3	C4	C5
Q1.1.					,673
Q1.2.			,382	,590	
Q2.1				,812	
Q2.2				,602	,328
Q2.3		,367		,302	,572
Q2.4				,715	
Q2.5		,850			
Q3.1		,824			
Q3.2		,728			
Q3.3	,446	,368	,347		
Q3.4	,487		,471		
Q3.5	,513		,372		
Q3.6	,562		,325		
Q4.1	,448				,565
Q4.2			,430		,576
Q5.1			,646		
Q5.2			,701		
Q5.3			,685		
Q5.4	,376		,408		
Q5.5	,658			,353	
Q5.6	,820				
Q6.1					,673
Q6.2			,382	,590	
Cronbach´ alpha					0.908
Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy					0.853
Bartlett's Test of Sphericity (Approx. Chi-Square)					1099,9(df = 210) [b]
Correlation Matrix Determinant					9,06E-005

Note: EFA=Exploratory Factor Analysis, loadings < 0.32 not shown;
a.Total variance extracted by the five factors = 60,995%, b. p<0.001

Kaiser-Meyer-Oklin (KMO) index[3] and the Bartlett's Test of Sphericity are checked. Then, principal components as extraction method with Varimax (with Kaiser normalization) as rotation method and the breaks-in-eigenvalues criterion [25] is used to decide the initial number of factors to keep. Factor loadings equal to or greater than 0.5 are considered strong [25]. Items with low loadings on all components (the cut-off value for loadings was 0.32 [26]) are eliminated too. Table 7 shows the KMO and Bartlett´s Test and the extracted components with their loadings. KMO was clear with value greater than 0.80 and Bartlett´s Test indicates a meaningful relationship among the variables. The extracted components have been labeled as follows:

Q3.4 Q3.6 Q4.1 Q4.2 Q6.1 Q6.2 = C1 (Market and attitude toward testing)
Q3.1 Q3.2 Q3.3 = C2 (Education and training)
Q5.3 Q5.4 Q5.5 Q5.6= C3 (Integration with development)
Q2.1 Q2.2 Q2.3 Q2.5 = C4 (Career)
Q1.2 Q2.4 Q5.1 Q5.2 = C5 (Attractiveness)

Note that EFA has extracted one factor less than the initial set of the questionnaire: initial grouping was done by the expert in charge of coordination based on his own experience and was not object of debate but it was confirmed by experts.

4.2.4 Confirmatory Factorial Analysis

The predictive validation obtained after applying EFA in previous section should be confirmed to obtain the final model. Confirmatory Factor Analysis (CFA) through Structural Equation Model (SEM) and Maximum Likelihood (ML) estimation method is used in order to assess the validity of the model. Several indicators were used to assess model fit in order to compare the alternative models such as the Root Square Error of Approximation (RMSEA), Comparative Fit Index (CFI), the Normed Fit Index (NFI), the Non-Normed Fit Index (NNFI) and the Relative Fit Index (RFI).

Table 8. Goodness of Fit indicators for the model

	Suggested cut-off Values	Factor´s questionarie
χ^2 (df)		189,036 (179)
S- χ^2	>1, <2	1.05
RMSEA	< 0.08	0.0021
CFI	> 0.9	0.997
NFI	> 0.9	0.954
NNFI	> 0.9	0.997
RFI	> 0.9	0.946

Table 8 shows minimum recommended values for good fit [28] as well as the calculated values for the model. All the indicators exceed the minimum recommended values for good fit providing evidence of discriminate validity.

5 Conclusions

One of the usual shortcomings of the area of software engineering is the lack of trustable data about which is the state of practice in general and more specifically in

[3] KMO: above 0.5, it should be accepted, 0.7-0.8, good value; above 0.8, meritorious [27].

software testing. Data from industry tend to be collected and processed in an informal and even slanted way so they are not trustable as an accurate view of reality; academia usually experience many problems to access software professionals and organizations to get information. The above presented studies are a contribution to overcome the mentioned absence of information.

In general, besides the need of improvement of organizational and individual practices, many of the 23 explicative factors have been confirmed by a varied and significant sample of 127 software professionals so there is now a guideline for improving software testing conditions. One of the most evident barriers is the lack of training and expertise, something consistent with other surveys [14] although market maturity and career issues are also considered very important factors. It is remarkable the traditional divorce between the development deliverables and test case design methods, something also detected in the data from individual practices (section 2).

We are now working to launch this survey across Europe with the help of CEPIS to check if the factors are common or if local differences arise. To support this effort, we intend to use results of the factorial analysis of the questionnaires for the grouping of items as well as for establishing the final model of factors to be applied. Anyway the model would be useful also for other researchers who may collect data in this area.

Acknowledgments

This study was supported by the projects TIN2007-67843-C06-01 and TIN2007-30391-E partially funded by the Spanish Ministry of Science and Innovation.

References

1. Jones, C.: Estimating software costs. McGraw-Hill, New York (1998)
2. Grindal, M., Offutt, J., Mellin, J.: On the Testing Maturity of Software Producing Organiza-tions: Detailed Data. Technical Report ISE-TR-06-03, Department of Information and Software Engineering, George Mason University (2006)
3. McGarry, F., Pajerski, R., Page G., Waligora, S., Basili V., Zelkowitz, M.: Software Process Improvement in the NASA Software Engineering Laboratory, Technical Report, CMU/SEI-94-TR-22, SEI Carnegie-Mellon University (1994)
4. Martin, D., Rooksby, J., Rouncefield, M., Sommerville, I.: 'Good' Organisational Reasons for 'Bad' Software Testing: An Ethnographic Study of Testing in a Small Software Company. In: Proc. of 29th Int. Conf. on Soft. Engin., pp. 602–611 (2007)
5. Fernandez-Sanz, L.: Un sondeo sobre la práctica actual de pruebas de software en España. REICIS 2, 43–54 (2005)
6. SEI: CMMi® for Development. SEI-Carnegie Mellon University (2006)
7. Paulk, M., Weber, C., Curtis, B., Chrisis, M.: The Capability Maturity Model. Addison-Wesley, Reading (1995)
8. van Veenendaal, E.: Test Maturity Model Integration (TMMi) Versión 1.0. TMMI Foundation (2008), http://www.tmmifoundation.org
9. Burnstein, I.: Practical Software Testing. Springer, Heidelberg (2002)
10. VanVeenendaal, E.: Guidelines for Testing Maturity. STEN IV, 1–10 (2006)

11. Koomen, T.: Test process improvement: a practical step-by-step guide to structured testing. Addison Wesley, Reading (1999)
12. Pol, M., Teunissen, R., van Veenendaal, E.: Software Testing. A guide to the TMap Approach. Addison-Wesley, Reading (2002)
13. Koomen, T., van der Aalst, L., Broekman, B., Vroon, M.: TMap Next for result-driven testing. UTN Publishers (2006)
14. Sanz, A., Saldaña, J., García, J., Gaitero, D.: TestPAI: A proposal for a testing process area integrated with CMMI. In: European Systems and Software Process Improvement and Inno-vation (EUROSPI 2008), pp. 3–5 (2008)
15. Ng, S.P., Murnane, T., Reed, K., Grant, D., Chen, Y.: A Preliminary Survey on Software Testing Practices in Australia. In: ASWEC, Proceedings of the 2004 Australian Software Engineering Conference, pp. 116–125 (2004)
16. Geras, A.M., Smith, M.R., Miller, J.: A survey of software testing practices in Alberta. Canadian J. of Electrical and Computer Engineering 29, 183–191 (2004)
17. Groves, L., Nickson, R., Reeves, G., Reeves, S., Utting, M.: A Survey of Software Practices in the New Zealand Software Industry. In: Proceedings of the 2000 Australian Software Engi-neering Conference, pp. 189–101 (2000)
18. Runeson, P.: A Survey of Unit Testing Practices. IEEE Softw. 23, 22–29 (2006)
19. Lara, P., Fernández-Sanz, L.: Un experimento sobre hábitos de pruebas artesanales de software: Resultados y Conclusiones. In: Taller sobre Pruebas en Ingeniería del Software PRIS 2007, pp. 23–30 (2007)
20. Lara, P., Fernández-Sanz, L.: Test Case Generation, UML and Eclipse. Dr.Dobbs Journal 33, 49–52 (2008)
21. Whittaker, J.A.: What Is Software Testing? And Why Is It So Hard? IEEE Software 17, 70–79 (2000)
22. Glass, R.L., Collard, R., Bertolino, A., Bach, J., Kaner, C.: Software Testing and Industry Needs 23, 55–57 (2006)
23. García, A., de Amescua, M.V., Sanz, A.: Ten Factors that Impede Improvement of Verification and Validation Processes in Software Intensive Organizations. Software Process Improvement and Practice 13, 335–343 (2008)
24. Taipale, O., Karhu, K., Smolander, K.: Observing Software Testing Practice from the Viewpoint of Organizations and Knowledge Management. In: First Intern. Symp. on Empirical Software Engineering and Measurement, pp. 21–30 (2007)
25. Hair, J.F., Tatham, R.L., Anderson, R., Black, W.: Multivariate Data Analysis. Prentice Hall, Englewood Cliffs (1998)
26. Tabachnick, B.G., Fidell, L.S.: Using Multivariate Statistics. Pearson, London (2006)
27. Nunnally, J., Bernstein, I.: Psychometric Theory. McGraw-Hill, New York (1994)
28. Hu, L., Bentler, P.M.: Cutoff: Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria vs new alternatives. Structural Equation Modeling 6, 1–55 (1999)