

A Taxonomy for Software Testing Projects

Jose Calvo-Manzano Villalón, Gonzalo Cuevas
Agustin, Tomás San Feliu Gilabert

Department Languages and Informatics Systems and
Software Engineering
Polytechnic University of Madrid, UPM
Madrid, Spain

jcalvo@fi.upm.es, gcuevas@fi.upm.es, tsanfe@fi.upm.es

José de Jesús Jiménez Puello

Department of Computer Science
University of Panama, UP
Panama, Panama
jjimpue@hotmail.com

Abstract — This article establishes taxonomy for software testing projects, allowing the development team or testing personnel to identify the tests to which the project must be subjected for validation. The taxonomy is focused on identifying software projects according to their technology. To establish the taxonomy, a development method comprised of 5 phases was applied. The developed taxonomy is comprised of 9 categories and 27 subcategories and was validated by a group of information technology (IT) managers and professionals in the field of IT through the use of a survey. The results obtained from the survey are subjected to the Mann-Whitney U test, which indicates that the taxonomy is validated. The taxonomy can be implemented in development organizations with or without a testing team that provides a classification for technology projects.

I. INTRODUCTION

The development of information technologies provides a picture of the size and complexity of software [7]. Given that the complexity of software is an inherent property and not causal [2], this leads to the existence of projects developed under one technology or projects that integrate several technologies.

Every project must verify and validate its software [12]. Verification and validation are a review process, with analysis and tests used along the entire life cycle of the software to guarantee the production of high-quality software [11]. Through a series of tests, such as functional and non-functional tests, it is determined whether the software complies with the user's specifications and requirements. As a result, software and systems tests are essential due to factors that increase risk, such as complexity, heterogeneity, and technology variability [5].

In the case of tests that validate a software project, the first step is to identify the tests that must be applied to the project according to the technology being implemented. Because in many cases projects are different from each other due to the different characteristics they may have, it may be difficult to select such technologies, as indicated by [3].

Another important aspect is that testing tools are not generally oriented towards competence in their application

[1514]. In addition, there is a gap in the area of testing tool selection for projects [1413], which makes it necessary to have a software project taxonomy that allows for characterizing the tests to which a project must be subjected during the development cycle of the software.

By identifying the technology involved in a specific project, one can establish the software tests according to the project. This article proposes establishing a software project taxonomy that allows grouping projects based on common tests for a software project.

The article is divided into the following sections: Section 1 provides a brief introduction to the problem. In Section 2 the work related to the problem is presented. In section 3 taxonomy software project proposal is presented. Finally, Section 4 shows the main conclusions.

II. RELATED WORK

Glass and Vessey [6] performs a review of taxonomies and determines what is necessary to establish one taxonomy for applications and another taxonomy based on developmental methods, observing that both should complement each other. Forward and Lethbridge [4] establishes software taxonomy, with a focus on defining the type of software from a descriptive perspective of the unambiguous identification of software, to facilitate its localization within a specific application domain. International Data Corporation [8] presents a software taxonomy that takes as reference the worldwide software market, that is, applications that are sold to end users, for development and implementation. Kuitunen, Kontio, Mäkelä and Jokinen [10] presents a software classification based on the North American Product Classification System (NACPS), which classifies software into systems software and applications software, with the purpose of establishing a classification for software industry products.

The taxonomies described above are focused on classifying software developed for a specific field of application or competency, ignoring the technology used for its development, which is important for focusing testing efforts at the level of functional and non-functional requirements.

III. TAXONOMY OF SOFTWARE PROJECTS

The software project taxonomy that we propose to develop be focus on establishing projects based on a specific application or specialty. Instead, it is established based on the technology employed for the development and implementation of the project. Because there are countless software applications, establishing the characterization of tests and testing tools for a specific project depends on the technology being implemented. Therefore, it is necessary for testing teams to identify which tests and tools to which the software project will be subjected.

A. Importance of taxonomy

The taxonomy as classification tool, lets organize within a domain or area of knowledge, different terms that integrated, providing a defined structure that provides a mechanism to identify, assign, and perform actions on a domain, clearly, providing solutions and generating knowledge

The project taxonomy proposed should contribute the following benefits:

- It will help characterize the testing resources for the project.
- It will identify the testing tools associated with the project.
- It will reduce the time and cost of the testing team for the project.
- It will increase the effectiveness of the test because it uses widely validated resources.
- It will allow for the clear identification of projects according to technology.
- It will incorporate previous knowledge of the difficulties and virtues of the tests for the project.
- It will facilitate the identification of the development of new testing tools.
- It will allow the management of testing group knowledge.

The software project taxonomy to be developed will allow testing personnel to have a clear identification of projects, facilitating the relationships between projects and thus allowing the determination of testing approaches that are quicker and more efficient, thereby reducing time, costs, and subsequent issues. In addition, it will allow the identification of the absence of testing tools for a given technology.

B. Methodology for the construction of taxonomies

For the construction of taxonomy the method developed by [1] which defines 5 phases was applied:

- Planning.
- Identification and extraction of information.
- Design and construction of taxonomy
- Testing and Validation

- Deployment of taxonomy.

C. Applying the Method

The phases of the method applied to taxonomy development stages are described below:

Planning:

The area of knowledge is limited to the software testing area, focusing on the topic of classification, categorization, or taxonomy of software projects or software applications according to their technology. The objective is to determine the different technologies that are used software projects. The searching period is established, as well as terms or keywords for the search and sources of information used to perform the search.

Identification and Extraction of Information:

Extraction of the terms that define the categories and subcategories of the taxonomy is based on the WebPages of software development organizations, software magazines, databases, and testing tools. The terms used are the following: **“software development company” and “software testing tool”**. In the case of testing tools, adding those projects that refer to software testing for a particular technology to the results was **also considered. In the case of “software development company”**, the different projects carried out by development organizations are extracted. For digital magazines, the term **“software testing tool” is used, and when the articles refers to** software testing for a project, the testing is applied to specialized databases. In addition, the search of software testing tools is conducted on Google to determine the testing tools on which project technology is focused.

To TesT Magazine 223 items were reviewed and selected 5, Testingexperience, reviewed 416 articles and selected 34. Software Test Professionals, reviewed 613 articles and selected 22, Methods & Tools: reviewed 211 and selected 6 Testing Circus, was revised 432 and selected 20. The Testing Planet, was reviewed and selected 316 5. For the databases we have: IEEE Xplore: revised 1,239 and selected 63, ACM digital library, 52 reviewed and selected 12. Springer Link 117 results are reviewed and selected 15. A total of 182 articles accepted between journals and databases that refer to test a project.

These results obtained were organized based on different projects in relation to the search source. The software development organizations have 141 software projects in their entirety. It was found that the desktop projects represent a 6.38% (9) Client / Server 1.42% (2) Rich Client 0.71% (1), portals / websites 2.84% (4), traditional Web 18.44% (26), Web 2.0 2.84% (4), mobile 28.37% (40), Big data 2.13% (3) Cloud: 11.35% (16), embedded 3.55% (5), databases 2.84% (4), SaaS 1.42% (2) Legacy / migration 0.71% (1), mission Critical: 1.42% (2) server: 0.71% (1), middleware, 0.71% (1), ETL, 0.71% (1), BI, 3.55% (5) Data Warehouse 3.55% (5), SOA 3.55% (5), Web services, 1.42% (2), SAP 1.42% (2). The projects that develop the organizations according to the results obtained are: Web, mobile and cloud.

The Tables I,II,III,IV shows the results based on the different searches for software development projects, including

the results for testing tools, business projects, and the results obtained from magazines and databases, which represent a total of 360 projects. Based on the results of the review, the projects that currently display an above-average frequency for software testing are independent applications, Web, mobile, cloud, embedded, databases, Service-oriented *architecture* (SOA), and Web services (see Table V). It is worth highlighting that client/server projects are treated in the taxonomy as clients, critical systems are similarly labeled as critical mission, and *Extract, transform, load* (ETL) tests are considered part of Business Intelligence projects (BI). Agents are included, although the results did not yield any test project agents, but, yes, another technology projects mentioned as agents to test other technologies. Inclusive, were not found business process management (BPM) projects.

TABLE I. SOFTWARE PROJECTS

Source	Desktop	Client /Server	Client Rich	Portal/ Web Sites	Web	Web 2.0	RIA
Software testing company	9	2	1	4	26	4	0
Software testing tool	3	1	0	1	9	0	4
IEEE Explorer	0	0	0	0	16	0	2
Testing Planet	0	0	0	0	0	0	0
Testing circus	0	0	0	1	3	0	0
Methods and Tool	0	0	0	1	1	0	0
Springer Link	0	0	0	1	0	0	0
ACM library	0	0	0	0	0	0	0
Software test professional	0	0	0	0	7	3	0
Testing experience	0	0	0	1	2	0	0
Test magazine	0	0	0	0	0	0	0
Total	12	3	1	9	64	7	6

TABLE II. SOFTWARE PROJECTS

Source	Mobile	Big data	Cloud	Embedded	Data Bases	SaaS	Legacy
Software testing company	40	3	16	5	4	2	1
Software testing tool	9	0	3	1	0	0	0
Testing Planet	3	2	0	0	0	0	0
Testing circus	8	0	4	0	0	1	0
Methods and Tool	4	0	0	0	0	0	0
Software test professional	3	0	2	0	2	0	0

Testing experience	11	1	8	1	2	0	1
Test magazine	4	0	0	0	0	0	0
IEEE Explorer	6	0	1	7	7	2	2
Springer Link	2	0	1	3	1	0	0
ACM library	0	0	0	9	0	0	0
Total	20	6	35	26	16	5	4

TABLE III. SOFTWARE PROJECTS

Source	Server	Middleware	Etl	BI	DW	SOA	Web Services
Software testing company	1	1	1	5	5	7	3
Software testing tool	0	0	0	0	0	2	2
Testing Planet	0	0	0	0	0	0	0
Testing circus	0	0	0	0	2	0	0
Methods and Tool	0	0	0	0	0	0	0
Software test professional	0	0	0	0	0	2	3
Testing experience	0	0	0	0	0	4	3
Test magazine	0	0	0	0	0	0	0
IEEE Explorer	0	0	0	0	0	2	9
Springer Link	0	0	0	0	0	2	1
ACM library	0	0	0	0	0	0	0
Total	1	1	1	5	7	19	21

TABLE IV. SOFTWARE PROJECTS

Source	Net. Prot.	Com. Prot.	Time Control	Peer to Peer	Grid	Critical Systems	Agents
Software testing company	0	0	0	0	2	2	0
Software testing tool	0	0	0	0	0	0	0
Testing Planet	0	0	0	0	0	0	0
Testing circus	0	0	0	0	1	0	0
Methods and Tool	0	0	0	0	0	0	0
Software test professional	0	0	0	0	0	0	0
Testing experience	0	0	0	0	0	0	0
Test magazine	1	0	0	0	0	0	0
IEEE Explorer	2	2	1	1	0	1	2
Springer Link	1	1	1	0	0	1	0

ACM library	1	2	●	●	0	0	●	●
Total	4	5	2	1	3	1	5	●

TABLE V. FREQUENCY OF SOFTWARE PROJECTS

Nº	Project	Frequency	Average
1	Independent / Desktop	12	0.03
2	Client / Server	3	0.01
3	Rich Client	1	0.00
4	Portals / Web sites	9	0.03
5	Web	64	0.18
6	Web 2.0	7	0.02
7	RIA	6	0.02
8	Mobile	90	0.25
9	Big Data	6	0.02
10	Cloud	35	0.10
11	Embedded	26	0.07
12	Database	16	0.04
13	SaaS	5	0.01
14	Legacy / Migration	4	0.01
15	Server	1	0.00
16	Middleware	1	0.00
17	ETL	1	0.00
18	BI	5	0.01
19	Data Warehouse	7	0.02
20	SOA	19	0.05
21	Web services	21	0.06
22	Network Protocol	4	0.01
23	Communication Protocol	5	0.01
24	Real-time system	2	0.01
25	Peer to Peer	1	0.00
26	SAP	3	0.01
27	Grid	1	0.00
28	Critical Systems	5	0.01
29	Agents	0	0.00

In the case of testing tools, a total of 35 tools are identified. The relationship between the types of projects and tools developed to test those projects place independent projects at 8.57% (3), client/server at 2.86% (1), Web applications at 25.71% (9), enriched Web applications at 11.43% (4), mobile at 25.17% (9), cloud at 8.57% (3), embedded at 2.87% (1), SOA at 5.71% (2), and Web services at 5.71% (2). According to the results obtained, the projects that present the greatest number are traditional Web, mobile, and enriched Web applications.

Design and construction of taxonomy:

Based on the extraction of information on software projects, the systematic review method of [9] is applied. The review encompasses the period from 1990 to June 2014. Based on the results obtained, it is established that each category should have a relationship with the function of the software project and that project subcategories are related to the category based on their functionality.

As a result of the design and elaboration of the taxonomy, the taxonomy categories are identified. Each category has a description that frames it within a specific project that uses the technology implemented for its development. In addition, the subcategories are established as derived from the categories. The following categories and subcategories are obtained as a result of the classification process.

- Desktop or independent applications: Projects that are destined for the development of applications that are installed and executed individually or independently in the computer or that request services or information from a server.
- Web: Applications in distributed environments that use the browser installed in the computer.
- Mobile: Applications developed for mobile devices.
- Services: Applications that offer services as applications, such as a service structure between applications or system integration, which provide services or functionality through their interfaces.
- System migration: Applications that are inherited and that will be updated or integrated with other technologies or services.
- Processes: Focuses on business flow processes, allowing the development and delivery of a product or service.
- Time control: Based on the constant monitoring of time intervals for the development of processes.
- Storage: Applications that are related to data storage structures.
- Protocols: Refer to the development of network or communication (telecommunications) protocols.

After identifying the categories of projects, the associated elements are established and grouped into subcategories. The Table VI establishes the relationship between categories and subcategories as a result of the classification.

TABLE VI. PROJECT CATEGORY AND SUBCATEGORY

Category	Sub category
Desktop	Standalone applications (Mono users)
	Client applications: Client / server, Rich Client
Web	Web pages or websites
	Traditional web
	Rich Internet Applications (RIA)
	Web 2.0
Mobile	Native mobile
	Mobile Web
Services	Server Applications
	SOA
	Web Services
	Applications as services (SAAS)
	Cloud
	Grid
	P2P
	Agents
Middleware	
Process	SAP
Time control	Real Time Systems
	Critical Systems
	Embedded
System migration	Legacy
Storage	Data Bases
	Data Warehouse/BI
	Big Data
Protocols	Network Protocol

Communication Protocol

9. Protocols	Network Protocol	23	9
	Communication Protocol	20	11

Testing and validation of the taxonomy:

To determine the validity of the taxonomy, a study was conducted that consists of a descriptive investigation (survey-based descriptive research) of non-experimental design of the self-report type. The technique used was the personal survey. The instrument used to gather the data was a questionnaire that was designed for this purpose Table VII. It contained the categories and subcategories of the taxonomy and questions to which the answers were limited to either Yes or No. The survey was administered to information and communications technology (ICT) professionals and managers in the Republic of Panamá to determine their perceptions of the taxonomy and the projects established in the taxonomy. The population surveyed totaled 46, and the sample obtained was (n=37).

Once the surveys were completed, the results were tabulated using Microsoft Excel 2010 software. The Statistica 7 software was used to process and analyze the data tabulated in Excel. The Mann-Whitney U test, a parametric test applied to two independent samples (Yes or No answers), was used to analyze the data. The table VIII shows the results obtained from the test. The analysis of the results (Fig. 1) indicates that there is a difference between the answers (U=14, p<0.05). The Box-Whiskers box suggests that more people answered Yes. Therefore, there is a significant difference, which demonstrates that the participants approve the taxonomy for their projects.

TABLE VII. SURVEY RESULTS

	Project	Yes	No
1. Desktop	Standalone applications	11	22
	Client/Server application	31	2
	Client / server application	29	5
2. Web	Web pages or websites	33	1
	Traditional Web	37	0
	Rich Internet Applications	33	0
3. Mobile	Web 2.0	31	0
	Native mobile	19	14
	Mobile Web	33	2
4. Services	Server Applications	29	5
	SOA	28	3
	Web Services	31	4
	Applications as services	29	3
	Cloud	30	5
	Grid	20	10
	P2P	20	10
5. Process	Agents	18	9
	Middleware	20	10
	SAP	30	2
6 Time Control	Real Time Systems	29	3
	Critical Systems	28	4
	Embedded	20	11
7. System migration	Legacy	27	6
8. Storage	Data Base	29	6
	Data Warehouse/BI	36	0
	Big Data	16	9

TABLE VIII. TEST RESULTS "MANN-WHITNEY"

Sample	Degree of freedom	U	p-level	P<0.05
N=37	A = .05	14.00000	0.000000	**

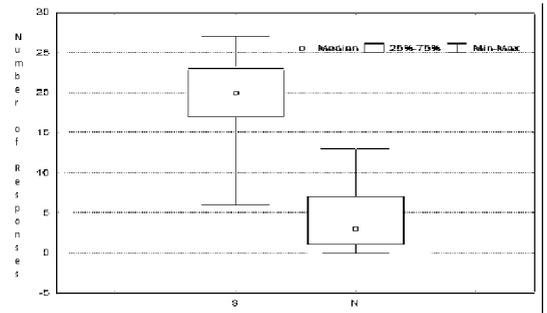


Figure 1. Approval of category and sub categories of software projects

Deployment of taxonomy:

The taxonomy will be deployed in software development companies to organize their projects for the validation and testing of software

IV. CONCLUSIONS

To establish the taxonomy, a taxonomy construction, definition, and development method was used. Through this method, the search for and identification of key concepts was performed to structure and build the software project taxonomy to allow structuring the categories and subcategories of the taxonomy based on the technology to be implemented in the software project.

As a result of the review, taxonomy was established with 9 categories and 27 subcategories of software projects. The most noteworthy projects are independents, Web, mobile, cloud, embedded, databases, SOA, and Web services.

The results indicate that the tools that present the greatest development are traditional Web, mobile, and enriched Web applications. The other projects present low percentages or the absence of tools at a commercial level, which may be due to their status as experimental projects or because, at a commercial level, they do not have a high demand; therefore, the tools are more specific for certain projects.

Through the taxonomy, several lines of research can be established, such as the characterization of software tests for each project and the establishment of testing templates that identify testing requirements, specific tests, techniques, and tools oriented towards a project, among other things. In addition, the taxonomy may organize testing policies and strategies that are in line with projects, and it may organize the application of specific tests, which have been previously applied, tested, and validated through previous projects, with a single project.

REFERENCES

- [1] S. Dayona, J. Calvo-Manzano, A. Gonzalo Cuevas y T. San "Critical success factors taxonomy for software process deployment", Spanish Journal of Innovation, Quality and Software Engineering, Vol. 6, No. 3, 2010.
- [2] F. Brooks, "No Silver Bullet Essence and Accidents of Software Engineering," *Journal Computer*, vol.20, no.4, pp.10-19, 1987.
- [3] A. Dias-Neto, and G. Horta Travassos, "Model-based testing approaches selection for software projects", *Information and Software Technology*, Vol. 51, pp. 1487-1504, 2009.
- [4] A. Forward, and T. Lethbridge, "A taxonomy of software types to facilitate search and evidence-based software engineering", In Proceedings conference of the center for advanced studies on collaborative research, ACM, pp. 4:179-14:191, 2008.
- [5] M. Garcia, and M. Elcuera, "Model for Training Specialists Software Testing Systems", Proceedings of Workshop on Software Engineering and Databases, Vol. 1, pp. 31-36, 2007.
- [6] R. Glass, and I. Vessey, "Contemporary Application-Domain Taxonomies". *IEEE Softw*, Vol. 12, pp. 63-76, 1995.
- [7] M. Harrold, "Testing: A Roadmap", In *Proceedings of the Conference on The Future of Software Engineering (ICSE '00)*. ACM, pp. 61-72, 2000.
- [8] IDC (International Data Corporation), "Software taxonomy", 2010. <http://www.idc.com/2010st/index.html>
- [9] B. Kitchenham, "Procedures for Performing Systematic Review", Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia, 2004.
- [10] H. Kuitunen, J. Kontio, M. Mäkelä, and J. Jokinen, "Software Product Industry Classification" Final Report of the capistus research project. Software Business Lab, Helsinki University of Technology, 2005. http://www.swbusiness.fi/attachments/software_classification_explanation.pdf
- [11] L. Li, and A. Walker, "Verification and Validation of the Seal Software Assessment Tool," *International Software Engineering Standards Symposium*, pp.15, 1997.
- [12] P. Tomar, and N. Gill, "Verification & Validation of Components with New X Component-Based Model", 2nd International Conference on Software Technology and Engineering, Vol. 2, pp. V2-366, V2-371,
- [13] S. Uspenskiy, "A survey and classification of software testing", Lappeenranta University of Technology Department of Information Technology Master of Science Thesis, 2010.
- [14] A. Yague, and J. Garbajosa, "Applying the Knowledge Stored in Systems Models to Derive Validation Tools and Environments", *Cognitive Informatics*, 6th IEEE International Conference on, pp.391,400, 2007.