

A new model for the selection of web development frameworks: application to PHP frameworks

Khaoula Benmoussa¹, Majida Laaziri², Samira Khouli³, Mohamed Larbi Kerkeb⁴, Abir El Yamami⁵

^{1,2,3}Information System Engineering Research Group, National School of Applied Sciences,
Abdelmalek Essaadi University, Morocco

⁴Information System Engineering Research Group, Faculty of Sciences, Abdelmalek Essaadi University, Morocco

⁵Laboratory Signals, Distributed Systems and Artificial Intelligence, ENSET Mohammedia,
Hassan II University, Morocco

Article Info

Article history:

Received Feb 10, 2018

Revised Aug 9, 2018

Accepted Aug 28, 2018

Keywords:

CodeIgniter

Laravel

PHP frameworks

PHP frameworks evaluation

Symphony

Zend

ABSTRACT

The use of a framework is often essential for medium and large scale developments, but is also of interest for small developments. PHP has evolved as the scripting language the most chosen by developers, which has generated an explosion of PHP frameworks. There is a big debate about what the best PHP frameworks are, because the simple fact is that not all frameworks are built for everyone. Indeed, not all frameworks meet the same needs, and several frameworks can be used together in certain situations. Choosing the right framework, however, can sometimes be difficult. In order to make the selection process easier, we propose a pragmatic and complete model to compare and evaluate the main PHP frameworks. This model is based on a set of comparison criteria based on the Intrinsic durability, industrialized solution, technical adaptability, strategy, technical architecture, and Speed criteria. Results show that the values of these criteria allow developers to easily and properly choose the framework that best meets their needs

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Benmoussa Khaoula,

Information System Engineering Research Group,

National School of Applied Sciences, Abdelmalek Essaâdi University,

Mhannech II, B.P 2121 Tetuan, Morocco.

Email: benmoussakhaoula24@gmail.com

1. INTRODUCTION

In recent years, the PHP programming language has evolved as the scripting language of choice by most developers; there has been an explosion of upcoming PHP frameworks. These frameworks for PHP

the design and IT would like to choose PHP frameworks, especially because they offer developers the ability to create more complex, secure and complete web applications faster than ever before. Thanks to their convenient file structure, they make it possible to organize the files and the code of the developer and to increase its productivity. PHP frameworks are available in all kinds of shapes and sizes and have targeted developers with different levels of experience, application needs, hosting capabilities and development time.

The authors [1] declare that the PHP framework facilitates Web programming and makes it better organized in several ways. First, PHP frameworks increase the productivity of programming because writing a piece of code that usually takes hours and takes hundreds of lines of code can be done in minutes using built-in functions of the framework. On the other hand, a PHP framework usually comes with a support team, documentation or important support forums where users can quickly get answers. The authors [2] emphasize the importance of using PHP frameworks; they propose a modular workspace, consisting of a set of libraries,

tools and conventions that allow focusing on the business logic of the 'application. It also provides a framework for the application, usually based on the use of the Model View Controller (MVC) paradigm. This ensures a separation between: the data model that is in charge of access and requests to the database, the "view", which is a description of the presentation of the data (eg: generation of HTML, XML, JSON, etc.), and the controller, which implements the business logic by retrieving the model data to make them available to view [3]. The authors [4], [5] argue that PHP frameworks have a major security advantage since its users become long-term testers. If a user finds a security issue, they can notify the infrastructure developer website so that the development team can fix it [6]. Thus the authors [7] declare that PHP frameworks have features that satisfy most web programmers depending on the development style they want. Actually, there are PHP frameworks with all kinds of features such as security, robustness, ease of use.

However, choosing a PHP framework introduces many requirements and complexity, as some development habits will have to be set aside when it is adopted. Thus, although it is supposed to improve the productivity of the user's developments, there is always a phase during which the cost is greater than the gain: this choice must be carefully considered.

To this end, our contribution in this article is to propose a new complete model, which gathers a set of comparison criteria dedicated to PHP frameworks, in order to facilitate the developer's choice of his suitable framework, since there are no methods comparison and comprehensive assessments. The reminder of this paper is as follow: section 2 presents a literature review about the subject, then we present through section 3 our proposed model dedicated to the evaluation of web development frameworks. Section 4 presents a discussion of the obtained results, to sum up with a conclusion in section 5.

2. LITERATURE REVIEW

Choosing a framework for a given project can be quite complicated when we do not know all the modern PHP frameworks that are most used, and more particularly their differences. Beyond the technical considerations when choosing a framework, it is essential to assess the risks presented by each of the selected solutions. Since the framework must eventually form the basis of the majority of developments made for the company, it is important that its durability be ensured: its development must be sufficiently dynamic, in order to limit the risks of abandonment or closure of the code. The QSOS method (Qualification and Selection of Open Source Software) has been proposed to evaluate these risks.

The QSOS method (Qualification and Selection of Opensource Software) was initiated by Raphaël Semeteys and developed by Atos Origin [8]. Was designed to evaluate Free Software and OpenSource objectively, the objective was to design and make available: an evaluation methodology, tools related to it and especially a repository of assessments carried out. Who says referential, says obligation to define a set of common criteria for each functional area and to evaluate each free software with regard to these. The proposed approach also includes a notion of weighting to adapt the evaluation grid and the rating. Obtained by the software to its particular context [9] [10]. QSOS is an iterative process divided into four stages. Here are these different steps:

- a. Define: the repository data (types of licenses, types of communities, functional coverage grids by domain).
- b. Evaluate : software according to three main axes : functional coverage, risks from the point of view of the user company, risks from the point of view of the service provider (expertise, training, support). Each axis consists of a number criteria. For example, the risk axis undertaken includes: intrinsic sustainability, integration, technical adaptability, level of industrialization and project strategy. These criteria are themselves composed of sub-criteria.
- c. Qualify: the specific context of a company (or a user) by weighting the previous criteria.
- d. Select: and compare software that meets the needs.

To better understand this process, Figure 1 illustrates QSOS methodology [9]. There is another method of evaluation that only concerns PHP frameworks. This method was created by the company SMILE [11], [12]. This company has created a book about frameworks, then it used comparison criteria to evaluate them:

PHP Version, Architecture Model, ORM, Templates, Cache, Friendly URLs, Validation Forms, Ajax, Extensions, Code Generation, Internationalization I18n, Recommendation. What is important is the ease of use of each framework, as well as the tools it offers. The more tools a framework offers for development, the faster the implementation phase of an application is. If we take for example the criterion "Automated unit tests", the framework offering this tool avoids the application developer having to spend time testing his entire application. When a report is generated automatically, it will be faster to know where the application's errors are. We also need a criterion concerning the learning of the framework.

The QSOS method is characterized by the evaluation of any open source software, is not dedicated specifically to the PHP framework. On the other hand, the method of criterion of the frameworks of the company SMILE concerns only the evaluation of PHP frameworks, but it does not use sufficient and global criteria of comparisons. For that aim, we propose a pragmatic and complete model to compare and evaluate the main PHP frameworks. We have focused on the main criteria for the selection of a framework that can differentiate a good from a less good framework, and we provide a quick summary of the reasons that push to use or not a framework in general, depends on the needs of the user.

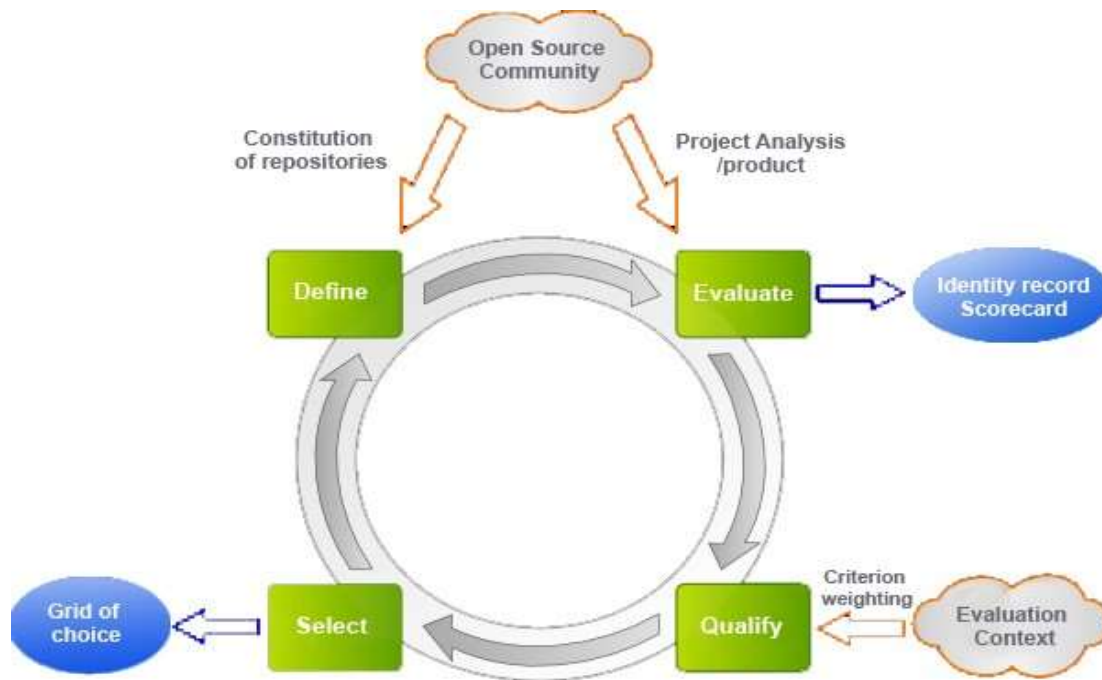


Figure 1. QSOS process

3. PROPOSED MODEL

In order to provide a holistic approach for the comparison of web development frameworks to be used by IT project developers, we provide a new model that combines six dimensions: Intrinsic durability, the industrialized solutions proposed by each framework, technical adaptability, the strategy, the technical architecture and finally, the speed, efficiency and economy of resources.

We present below our proposed model, for more understanding, we have selected 4 PHP frameworks, that are most used in web development these last years (Laravel, Symfony, Zend and Codeigniter). For each criterion, the scoring rule is:

- a. 0 if the feature is not covered.
- b. 1 if the feature is partially covered.
- c. 2 if the feature is fully covered.

3.1. Intrinsic durability

Beyond the technical considerations that come when choosing a framework, it is essential to evaluate the risks presented by each of the solutions chosen. Thus, it is important that its durability be ensured: its development must be sufficiently dynamic, in order to limit the risks of abandonment or closure of the code. Table 1 presents the intrinsic durability comparison of the selected PHP frameworks.

Table 1. Intrinsic Durability Comparison

Intrinsic durability	Laravel [13]	Symfony[14]	Zend [15]	Code Igniter[16]
<i>Maturity</i>				
Age	1 release date : 2011	1 release date : December 2005	1 release date : 2007	1 release date : 2006
Stability	2 stable version : 5.4.27	2 stable version 3.3.9	2 stable version 3.0	2 stable version 3.1.6
History, known issues	2	2 a bug history is available on the site [13]–[16]	2	2
Probability or provenance of a FORK	1 Laravel is Symfony's Fork	1 Symfony is a FORK of Mojavi2	2	0
<i>Adoption</i>				
Popularity	2	2	2	2
References	2 Many references (Amazon, etc)	2 Many references (Yahoo!, etc.). cf. http://symfonians.net/applications/	2 Many references (IBM, Nokia, etc.). cf. http://framework.zend.com/community/applications/	1 some references whose Emmaus.uk
Community of contributors	2 the community is active on dedicated forums, blogs and Google Groups	2 the community is important and active on the dedicated forums, blogs and Google Groups	2 the community is important and active on dedicated forums and blogs	2 the community is active on the dedicated forums
Publication	1 publication of English books « larevel 5.x cookbook » and « larevel essentials » etc.	1 publication of English books « The Definitive Guide to Symfony » and « Develop your website with Framework symfony 3 » etc.	1 parution de livres en anglais « Architect's Guide to Programming with Zend Framework » and « Zend Framework in action » etc.	1 publication of a book in English « professionnel codigniter » and « Codeigniter for Rapid Php Application Development » etc.
<i>Direction of developments</i>				
Leading team	2 Laravel management team	2 Fabien Potencier of Sensio Labs	2 management team from Zend Technologies	2 EllisLab management team
Direction mode	2 a person	1 a person	2 group of people	1 a person
<i>Activity</i>				
Number of developers, identification, turnover	2 a dozen people	2 fifty people	2 fifty people	2 Ten people
Activity around bugs	2 bug tracking is available via github laravel	2 responsive forum and bug tracking available via a symfony trac	2 tracking of available bugs through a dedicated wiki	2 bug tracking accessible via the page BugTracker
Activities around features	2 a section of the forum is devoted to the demand for new features	2 a section of the forum is devoted to the demand for new features	2 an IssueTracker page is dedicated to the demand for new features	2 a section of the forum is devoted to the demand for new features
Activities on releases	2	2	2	2
<i>Independence of developments</i>				
Independence of developments	1	1	1	1

3.2. Industrialized solution

Several constants must be taken into account before starting an industrialization developments based on web development frameworks: the mode of operation of the development, infrastructure and maintenance teams, the current and future functional requirements, the constraints of the hosting platform. We present through Table 2, a comparison of the selected frameworks based on industrialized solution criterions.

Table 2. Industrialized Solution Compariso

Industrialized Solution	Laravel	Symfony	Zend Framework	Code Igniter
<i>Services</i>				
Training	2 Fabien Le corre and Steven Sil are for example providers offering training laravel	2 Anaska, Sqli or Clever Age are for example providers offering Symfony training	2 Zend and Anaska for example, are providers offering training at Zend Framework	2 Jérôme Megel is for example offers a codeigniter training
Support	2 several existing offers forums, mailing lists and Google Groups, tutorials etc.	2 several existing offers (Clever Age, Sensio, etc.), forums, mailing lists and Google Groups., tutorials	1 Zend Technologies offers support for this framework via its network	1 provides support for this framework via forums, tutorials
<i>Documentation</i>				
Documentation	2	2	2	2
<i>Quality method</i>				
Quality assurance	2	2	2	1
Tools	2	2	2	1
<i>Packaging</i>				
Source	2	2	1	1
Debian	2	2	2	2
FreeBSD	1	1	1	1
HP-UX	1	1	1	1
Mac OS X	1	1	1	1
Mandriva	1	1	1	1
NetBSD	1	1	1	1
OpenBSD	1	1	1	1
RedHat/Fedora	1	1	1	1
Solaris	1	1	1	1
SuSE	1	1	1	1
Windows	1	1	1	1
<i>Operability</i>				
Ease of use, ergonomics	2	2	2	2
Administration / Supervision	2	2	2	2

3.3. Technical Adaptability

The selected frameworks should be compared according to the technical adaptability characteristics. Table 3 depicts the results of the comparison.

Table 3. Technical Adabtability Comparison

Technical Adaptability	Laravel	Symfony	Zend Framework	Code Igniter
<i>Modularity</i>				
Modularité	2	2 plugins	2 extensions	2
<i>Derivative works</i>				
Technical ease of modification of the existing code	2	2	1	2
Easy code extension	2 ability to use components, templates and plugins additional	2 possibility to use additional plugins	2 possibility to use extensions	2 possibility to add plugins

3.4. Strategy

The strategic dimension is very important in the selection of information systems development frameworks. Thus, we present trough Table 4, the comparison between the selected PHP frameworks based on the strategic features.

Table 4. Strategic Comparison

Strategy	Laravel	Symfony	Zend Framework	Code Igniter
<i>Licence</i>				
Permissiveness	2	2	2	2
	Licence MIT	Licence MIT	Licence BSD	Licence MIT
<i>Rights holders</i>				
Roadmap	1	1	1	1
<i>Sponsor</i>				
Sponsor	1	1	1	1
	Sole sponsor: Taylor Otwell.	Sole sponsor : Sensio Labs	Sole sponsor : Zend technologies	Sole sponsor : EllisLabs
<i>Strategic independence</i>				
Strategic independence	1	1	1	1

3.5. Technical Architecture

We present through Table 5 the technical architecture criteria for the assessment of the frameworks.

Table 5. Comparison of the Technical Architecture

	Laravel	Symfony	Zend	Codeigniter
PHP version	5	5	5	4/5
Documentation	2	3	1	2
Community	3	3	3	3
Tutorials	3	3	3	2
Automated unit tests	yes	yes	yes	yes
Rights management	yes	yes	yes	yes
Architecture model	MVC	MVC	MVC	MVC
Integrated ORM	Eloquent	Doctrine or Propel	Doctrine	Doctrine
Template	blade	PHP or smarty	PHP	PHP
Size of tool library	3	3	3	2
AJAX	yes	yes	No	No
code generator	yes	yes	No	No

3.6. Speed Comparison

The selection of web development framework is largely geared towards the execution speed of the framework. As such, the alternatives should be compared according to their speed to estimate the speed of development of the project to the use of a framework. High speed allows developers to save time developing applications.

4. RESULTS AND DISCUSSIONS

Selecting a Web development framework is not an easy task, since this choice involves quite a variety of issues: internal design of the framework, ease of use, technical qualities, future evolution, etc. But, beyond the simple qualities of the frameworks compared in this paper, several constants must be taken into account before starting to develop by one of these frameworks: the mode of operation, development teams, infrastructure and maintenance, the current and future functional requirements, and the constraints of the hosting platform. In any case, the choice of a development framework can only improve the quality of an existing which, without procedures, conventions or standards, is rarely well designed. It remains to determine what the framework that best suits your needs and your context is.

This section aims to analyze the different frameworks cited in this paper. It should be noted that the information gathered through the analysis is mainly statistical. We present through Table 6 the representation in tabular form, expressed as a percentage of the maximum achievable score, the distribution of the notes of these four frameworks according to the main categories analyzed. Then we provide a representation in the form of a histogram graph through Figure 2:

Table 6. Summary Table of these Main Categories of the QSOS Method

	Laravel	Symfony	Zend	Codeigniter
Perenity	83,33%	83,33%	86,66%	76,66%
Industrialised solution	73,68%	73,68%	68,42%	63,15%
Adaptability	100%	100%	83,33%	100%
Strategy	70%	70%	60%	60%

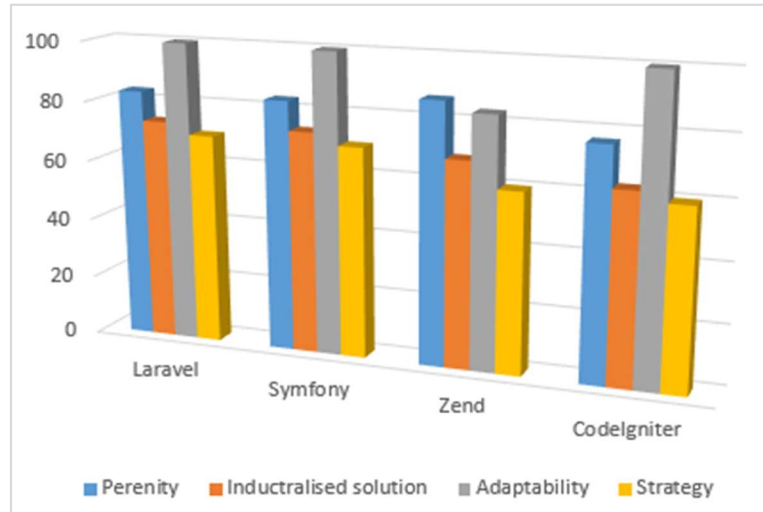


Figure 2. Histogram' summary chart

The purpose of Table 7 is to be able to define a selection of the most interesting frameworks with which we will make an example application to really see the differences in practice. Here is the comparison chart:

Table 7. Comparative of the Selected Frameworks [12]-[16],[17]

	Laravel	Symfony	Zend	Codeigniter
PHP version	5	5	5	4/5
Documentation	2	3	1	2
Community	3	3	3	3
Tutorials	3	3	3	2
Automated unit tests	Yes	yes	yes	yes
Rights management	Yes	yes	yes	yes
Architecture model	MVC	MVC	MVC	MVC
Integrated ORM	Eloquent	Doctrine or Propel	Doctrine	Doctrine
Template	blade	PHP or smarty	PHP	PHP
Size of tool library	3	3	3	2
AJAX	Yes	yes	No	No
code generator	Yes	yes	No	No

We present the Bar graph representation according to code generator, AJAX, MVC architecture model, rights management, automated unit testing criteria as shown in Figure 3.

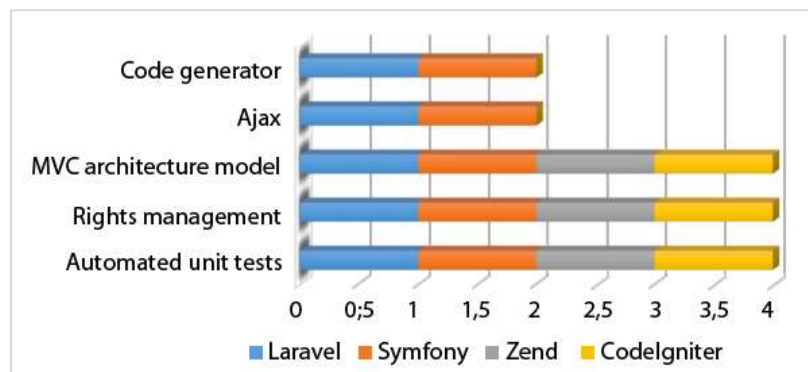


Figure 3. The comparison between framework tools

Figure 4 presents the curved graphic representation according to the criteria: documentation, community, tutorials, tool library size.

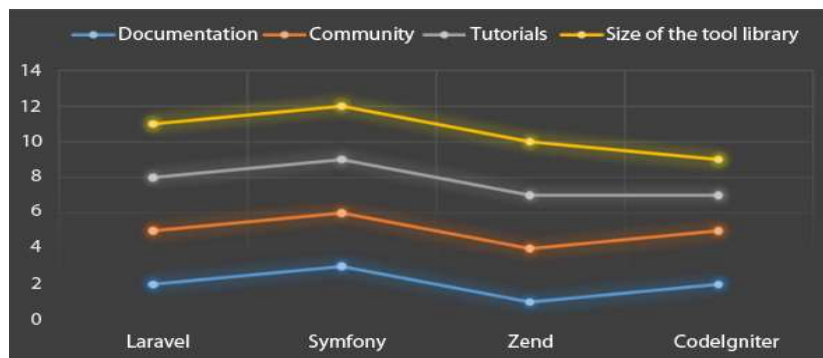


Figure 4. Comparison between frameworks with different criteria

These graphs allow to have an analysis of the selected frameworks. We notice that most frameworks have quite similar results. The next step consists of evaluating the frameworks according to speed criteria, efficiency and optimization of resources. For that aim, tests were done with these four frameworks on the same computer and taking the same actions. Apache Benchmark has been defined as a test tool. The computer on which the tests were performed has the following specifications: Intel Core i5, 8 GB DDR3, SATA3 SSD, Ubuntu 14.04 LTS, Apache 2.4, PHP 5.5. Figure 5 presents the number of queries that these compared frameworks could handle per second [12], [18]:

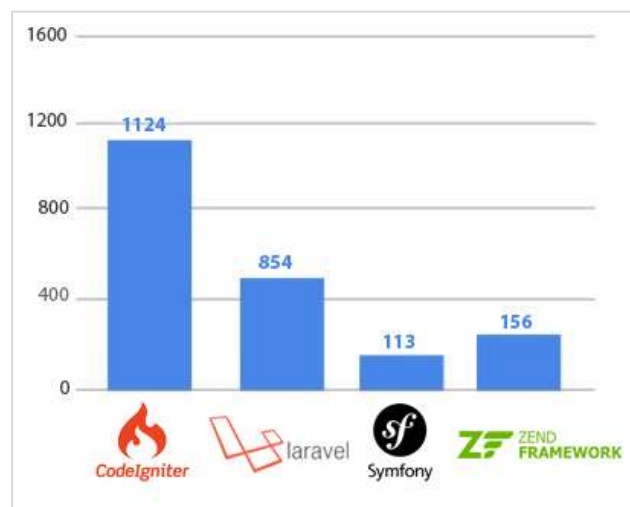


Figure 5. Comparison between the frameworks by the criterion of speed, efficiency and economy of resources

Results show that CodeIgniter framework is slow and does not comply with current requirements. Thus, this tool is not worth learning at this stage.

Laravel, Zend and Symfony are currently in compliance with modern standards and requirements. They are in demand and have huge features. For beginners, it will be easier to start with Laravel, rather than Symfony or Zend.

Laravel can be used for the rapid development of large-scale applications, even with less experienced resources, but if it is a complex level application development, it would be preferably for developers to choose Symfony, but they need to be more experienced to develop with this framework.

Zend's development team includes PHP kernel developers and makes sure the framework is robust, complete and stable. On the other hand, Zend is one of the most difficult settings to learn and has a very steep learning curve. Further, the configuration options are huge and simply increase the complexity.

It has for a very long time evolved very little: the community could hardly contribute to the framework, therefore version 2 had to correct that. Furthermore, it is very restrictive and also requires a long learning time.

Finally, CodeIgniter framework is « light »: it leaves a lot of flexibility to the developer. This can be a weak point as a strong point. His grip is extremely fast (a few hours). And the community is becoming more available. In addition, we can find on Github dozens of libraries to handle all the classic needs of a developer. It is ideal for simple projects and short.

5. CONCLUSION

In this paper, we have proposed a new model for the full assessment of IT project development frameworks to be selected by developers. For this, we have proposed a set of criteria to be used in the selection phase, based on six dimensions: the intrinsic durability, industrialized solution, technical adaptability, strategy, technical architecture, and Speed criteria.

For more understanding of our proposed model, we have applied our model, to set of 4 PHP frameworks that are the most used in these last years, and we provided a detailed analysis of this comparison. In perspective, the proposed model can be used as an AHP (Analytical Hierarchy Process) [19] hierarchy to calculate the weight of each criteria depending on the decision makers.

REFERENCES

- [1] N. Prokofyeva and V. Boltunova, "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia Comput. Sci.*, vol. 104, pp. 51–56, 2017.
- [2] B. Ninassi and S. Panay, "Les frameworks de développement web : comment enrichir rapidement et efficacement le système d'information de votre établissement," pp. 1–8, 2011.
- [3] S. Subramaniam, S.-C. Haw, and P. K. Hoong, "Bridging XML and Relational Databases: An Effective Mapping Scheme based on Persistent Labeling," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 2, no. 2, pp. 239–246, 2011.
- [4] R. Das and D. Prasad Saikia, "Comparison of Procedural PHP with Codeigniter and Laravel Framework," *Int. J. Curr. Trends Eng. Res. Sci. J. Impact Factor*, vol. 2, no. 6, pp. 42–48, 2016.
- [5] D. R. Lakshmi and S. S. Mallika, "A Review on Web Application Testing and its Current Research Directions," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 4, p. 2132, 2017.
- [6] V. Narayana, P. Premchand, and A. Govardhan, "Performance and Comparative Analysis of the Two Contrary Approaches for Detecting Near Duplicate Web Documents in Web Crawling," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 2, no. 6, pp. 819–830, 2012.
- [7] F. Sierra, J. Acosta, and J. Ariza, "Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web," *Rev. Investig. y Desarro. en TIC*, vol. 4, no. 2, pp. 1–13, 2013.
- [8] R. Semetey, "Method for Qualification and Selection of Open Source Software," *Technology innovation management review*, no. May 2008. Talent First Network, 2008.
- [9] V. Blazevic, "Technology & Innovation Management," *Open Source Bus. Resour.*, no. May 2008, 2012.
- [10] Umm-E-Laila, A. Zahoor, K. Mehboob, and S. Natha, "Comparison of open source maturity models," in *Procedia Computer Science*, 2017, vol. 111, pp. 348–354.
- [11] N. Richeton and C. Version, "Frameworks PHP," *Consultant*, pp. 1–78, 2007.
- [12] R. F. Olanrewaju, T. Islam, and N. Ali, "An empirical study of the evolution of PHP MVC framework," in *Lecture Notes in Electrical Engineering*, 2015, vol. 315, pp. 399–410.
- [13] "Laravel - The PHP Framework For Web Artisans." [Online]. Available: <https://laravel.com/>. [Accessed: 03-Feb-2018].
- [14] "Symfony, High Performance PHP Framework for Web Development." [Online]. Available: <https://symfony.com/>. [Accessed: 03-Feb-2018].
- [15] "Home - Zend Framework." [Online]. Available: <https://framework.zend.com/>. [Accessed: 03-Feb-2018].
- [16] "CodeIgniter Web Framework." [Online]. Available: <https://codeigniter.com/>. [Accessed: 03-Feb-2018].
- [17] S. T. AMIT and DR. S. E. YEDEY, "Study of Codeigniter Technology," *Int. J. Res. Comput. Inf. Technol.*, vol. 1, no. 2, pp. 1–3, 2016.
- [18] V. Petrosyan, "A Guide to Popular PHP Frameworks for Beginners | SEJ," 2016. [Online]. Available: <https://www.searchenginejournal.com/guide-popular-php-frameworks-beginners/180922/>. [Accessed: 03-Feb-2018].
- [19] S. Roy, P. K. Pattnaik, and R. Mall, "A cognitive approach for evaluating the usability of storage as a service in cloud computing environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 2, pp. 759–769, 2016.