Methodology to define an integration process between frameworks SCRUM, Django REST framework and Vue.js, implemented for software development, from quality management approach and agility

J G Arévalo¹,², L Viecco¹ and L Arévalo¹
¹University of La Guajira, Riohacha, La Guajira – Colombia.
²ENAPROT, Investigation Group of the University of La Guajira.

E-mail: jarevalo@uniguajira.edu.co

Abstract. At present, the industry has great challenges for the construction and maintenance of software, in terms of quality, costs and time, for this it is significant to have a methodological framework, which allows the possibility of having effective and practical methods, applying programming techniques from an agile approach, which guarantees the construction of quality code from the beginning of software development. This study analyzes the integration between frameworks SCRUM [1], Django REST framework [2] and Vue.js [3], which are commonly implemented in isolation for the execution of software development projects, and components are also incorporated of the IEEE 830 standard defined for the specification of software requirements [4], and of the ISO 25000 standard for software quality assessment [5]. Likewise, the characteristics of each study framework within the software architecture are analyzed, establishing of traceability between them and obtaining as a result a methodology for the definition of a process map; where each framework is integrated and it is described the procedure for structuring the planning, development and delivery phases. Thus providing a guide for the application of programming techniques from the concept of agility and quality management, from the process of integration of the frameworks analyzed in this study.

1. Introduction
There is a tendency for the use of agile methodologies for software development projects; because they are included in frameworks where different layers of the architecture of a project are structured, planned and controlled significantly reducing the execution time of the same. Currently, a large number of technologies and frameworks are available, however, in a relative way they are directed to medium-scale implementations down. In this study, the analysis of various frameworks is presented, to define the architecture of an integrated process, having a practical guide for the development of scalable software projects, based on their use.

In the software architecture the different levels for the development of a system are grouped, from the design, data storage, views for the validations and data processing, security aspects, quality, among others. Beyond the programming and definition of data schemas, a guide or methodology is required to keep track of the activities that are carried out throughout the architectural process. Several studies have presented cases focused on the integration of work frameworks, however on the one hand they focus on the permanent search to ensure software quality [6] [7] [8]; and on the other hand they are...
grouped into having agile tools [9] [10]. The referenced research’s guides this study to obtain the methodology to define the process of integrating SCRUM frameworks, Django REST framework and Vue.js for software development from the agility and quality management approach.

2. Theoretical bases

The following sections present related concepts, used as theoretical bases in this study, including: Software architecture components (section 2.1), Development languages and tools (section 2.2) and frameworks (section 2.3).

2.1. Software architecture components

2.1.1. Software architecture. It is part of the fundamentals of software engineering and is also a broad term, so it is necessary to specify its definition for this particular study, this definition refers to the stages of software development throughout the execution of a project, where each main component is included, its behavior within the system, the iterations between the components and the coordination in search of reaching each objective of the system. It is important to clearly have every stage of the development of the architecture, because in the search to guarantee the quality of the product, in some stages the standards recognized by the Institute of Electrical and Electronic Engineers are implemented and developed - IEEE, world association of engineers dedicated to Standardization in technical areas; and by the International Organization for Standardization (ISO), dedicated to the creation of international standards; In addition to establishing the traceability of each frame for each architectural stage, in the application of the agile manifesto.

The Standard ISO/IEC/IEEE 42010: 2011 [11], addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions. A conceptual model of architecture description is established. The required contents of an architecture description are specified. Architecture viewpoints, architecture frameworks and architecture description languages are introduced for codifying conventions and common practices of architecture description.

2.1.2. Integrated Development Environment (IDE). Corresponds to the logical components or applications, which, depending on their use, allow, on the one hand, the definition of the database schemas and, on the other, the software programming. The IDE integrates tools, libraries or components that allow to speed up the development process. An alternative not to use the IDE and that is used with great frequency are the text editors, which, like integrated environments, allow the creation of digital files in the formats of the different programming languages. They are used for the development of server-side code and for user-side interfaces.

2.1.3. Hardware. Corresponds to computer equipment or input or output devices, used to perform activities related to data processing and software development.

2.1.4. Database. It is the set of structures or schemes, defined logically for data storage. The definition of these schemes, for the most part, must respond to the data generated from the definition of the functional requirements. These schemes must also allow the consultation of data for proper processing and have the required reports. There are several managers that allow you to manage the content of the databases and are defined as database management systems DBMS. On the other hand, the database engines are responsible for providing the protection of the schemes, as well as allowing the storage and processing of the data.

Finally, it is important to indicate that the schemas can be defined in different structures, the most currently implemented are the Relational Database - BDR, BDOR Relational Object Database and Non-Relational Database - BD NoSQL.
2.1.5. **Software Requirements Specification.** The description of the requirements, is made up of a list of functionalities that the software to develop must comply with, also includes the use cases of the iterations that the user would have in the software. The IEEE 830 standard [4], has a structure for the specification of software requirement, where it contains the requirements at a level of detail sufficient to allow designers to design a system that meets the needs of the client and allow the team plan and perform tests that meet or not the established requirements.

2.1.6. **Life cycle of a project.** It is significant to have clarity about the life cycle for the execution of the projects, depending on the type of project they can have several stages, however, in the vast majority for software management projects, the stages are constituted as the specific guide of the fundamentals for project management [12]: start, planning, execution, monitoring-control and closure; Thus, in this way they have been established in various methodology for software development, however, difficulties have arisen in relation to the current challenges of the industry for the construction and maintenance of software, in terms of quality, costs and time.

However, for its part SCRUM encompasses the life cycle stages of a software development project in: Product Backlog, Sprint Planning, Sprint Backlog, Sprint, Sprint Burndown, Release or Product Burndown, Impediments Backlog, Product and Retrospective Increase, throughout these stages the requirements are specified, each work day is established, the list of tasks is made, each task is executed with a time limit, the progress is monitored, the rhythm of delivery of functionalities tested, deliveries are made to the final customer and the process is retrospectively and how the team works, including techniques, practices and development tools.

2.2. **Development languages and tools**

2.2.1. **Structured Query Language – SQL.** It is a language designed for the administration of transactional database management systems. Based on the use of SQL statements, statures or schemas are defined and grouped into data definition language (DDL); records management in relation to the basic actions to create, read, update and delete records known by the acronym CRUD, are grouped into data manipulation language (DML); to manage and control access to data, the data control language (DCL) statements are used; and to control the processing of transactions in the database the transaction control sentences (TCL) are used.

2.2.2. **Python.** It is an interpreted programming language, with an intuitive syntax that favors readable code. It is a multi-paradigm programming language, since it supports object orientation, console programming, web programming, and mobile programming and commonly used for the analysis of large volumes of data and artificial intelligence.

2.2.3. **JavaScript.** Like Python, it is an interpreted programming language, dialect of the ECMA script standard. It is defined as object oriented, prototype based, imperative, weakly covered and dynamic.

2.3. **Frameworks**

2.3.1. **Methodology of development software.** It is a framework used to structure, plan and control the development process in information systems.

2.3.2. **Scrum.** It is a very complete agile framework for project development. Scrum is an adaptable, iterative, fast, flexible and efficient framework that is designed to deliver value to the client throughout the project development. The primary objective is to meet the needs of the client through an environment of transparency in communication, collective responsibility and continuous progress [1].
2.3.3. Django rest framework. It is a powerful and flexible toolkit for creating web API [2]. Some of the reasons that define in its documentation to use this REST framework are: i) The web browsing API is a great usability gain for developers, ii) Authentication policies that include packages for OAuth1a and OAuth2, iii) Serialization compatible with ORM and non-ORM data sources, iv) Customizable to the end: only use regular function-based views if you don't need the most powerful functions, v) Used and trusted by internationally recognized companies, such as Mozilla, Red Hat, Heroku and Eventbrite.

2.3.4. Vue.js. It is defined as the progressive framework for JavaScript [3]. Integrates HTML, CSS and JavaScript, reference languages used in the development of client-side views. Similarly, an ecosystem that can be adopted incrementally and that can be scaled between a library and a full-featured framework is defined as versatile.

3. Methodology for defining the integrated process map

3.1. Framework features
From the analysis of the characteristics of each study framework, they can be classified within the architecture of a software development project. The SCRUM structure is selected for the project execution process. The processes implemented in SCRUM are structured in time boxes as one of the principles of the framework, and that allows to effectively manage the planning and execution of a project, where you can have several iterations throughout in order to increase the product in the deliveries to the end user, these iterations are divided into the planning, development and delivery phases.

Django REST framework, is made up of a powerful and flexible toolkit to create web APIs, therefore it is used for the development of server-side software or Backend, which is the part that connects to the database and the server. For its part, Vue.js is an open source JavaScript framework for creating single-page user interfaces and applications (SPA), its implementation will be on the client side or Frontend, which is the part of a website that interacts with the users. Table 1 shows the characteristics of each framework.

| Items | Framework       | Phase Scrum | Features                                                                 |
|-------|-----------------|-------------|--------------------------------------------------------------------------|
| 1     | Django REST     | Development | • Powerful and flexible toolkit for building Web APIs.                     |
|       | Framework       |             | • Serialization that supports both ORM and non-ORM data sources.          |
|       |                 |             | • Customizable all the way down - just use regular function-based views if you don't need the more powerful features. |
|       |                 |             | • JSON Web Token Authentication, Hawk HTTP Authentication and HTTP Signature Authentication. |
|       |                 |             | • Djoser and Django-rest-knox.                                           |
| 2     | Vue.js          | Development | • Build APIs for it frontend                                             |
|       |                 |             | • Is incrementally adoptable, scalable between a library and a full-featured framework. |
|       |                 |             | • Integrates HTML, CSS and JavaScript.                                    |
|       |                 |             | • Minimal Optimization Efforts.                                          |
3.2. Selection of standards

As described above, is selected the structure SCRUM for the planning and execution of a project is maintained, in relation to this, then in Table 2, the criteria for the incorporation of standards in the process of integration of the frameworks are presented. The selection of standards and integrate them into the process, is done in search of providing a guide for the definition of requirements and the evaluation of the software product, however, requirements can be defined with another method that allows to standardize the process, but it is recommended the continuation of the guide. The selected standards apply in different phases of software development.

On the one hand the selected criteria of the IEEE 830 allow to clearly specify the software requirements to have an overview of the product perspective, to define the restrictions that are imposed on the development, the assumptions and dependence that represent the factors that, if change, can affect the requirements specify, finally the functional and external requirements can be defined as design and performance, it is significant to highlight the standard is integrated in the planning phase. On the other hand, the selection of the ISO 25000 standard, allows to establish the quality requirements and the evaluation guide for independent developers, acquirers and evaluators, the standard is integrated into the delivery phase of the software product.

| Items | Standard | Phase Scrum | Selection Criteria |
|-------|----------|-------------|--------------------|
| 1     | IEEE 830 | Planning    | • Overview of the software product perspective.<br>• Restrictions that are imposed on development.<br>• Assumptions and dependence.<br>• Definition of specific requirements at the level of functionalities.<br>• Definition of external requirements at the level of design and performance. |
| 2     | ISO 25000| Delivery    | • Quality requirements.<br>• Evaluation guide for developers, acquirers and independent evaluators. |

3.3. Methodological integration

At this point, once the characteristics of the frameworks and the criteria for selecting the standards to be incorporated have been described, the process of their integration can be graphically represented, where the traceability of the frameworks is observed in the stages of Planning, development and delivery. In Figure 1, the defined process map can be observed, it should be noted that the structure of SCRUM is maintained, and the other study frameworks and / or standards are integrated as complementary mechanisms for the development of the execution phases of the project. During the execution of a project, several iterations can be presented, in order to increase the product for the end user.
The integrated model focuses on the unification of the planning, development and delivery phases, where the methodological components of SCRUM, Django REST framework and Vue.js are identified, as well as the complement of the IEE 830 and ISO 25000 standards. A brief description of the stages of the representative model of the integration process follows.

3.3.1. Planning. In this phase the iteration of a project begins, from the proportion of requirements of the Stakeholder to the Product Owner, this turn, in the Product Backlog in establishes and communicates to the Scrum Master the prioritized business requirements. For their part, the Team Scrum and the Scrum Master, are responsible for the planning of each sprint, supported by the specifications of the IEEE 830 standard, obtaining as a result the Sprint Backlog, where the general vision of the product perspective is defined, the restrictions that are imposed on the development, the specific requirements at the level of functionalities, design and performance, and also the criteria of acceptance of the product. For the use of the IEEE 830 standard, the following sections are established.

3.3.1.1. Model overview
- Functionalities: This subsection will show a summary, broadly speaking, of the functions of the future system.
- User characteristics: This subsection describes the general characteristics of the users of the product, including educational level, experience and technical experience.

3.3.1.2. Requirements
- External interface requirements. Requirements that affect the user interface, interface with other systems (hardware and software) and communications interfaces will be described.
- Functional requirements. This subsection must specify all those actions (functions) that the software must carry out. In table 3, the form to specify the functional requirement is structured.
### Table 3. Functional requirements

| Identifier | Description |
|------------|-------------|
| FN-0X      |             |

#### Dates Entry: Process: Exit:

#### Conditions Initials: Finals:

- Design requirements. The design requirements are listed, in table 4 a simple structure is available for this.

### Table 4. Design requirements

| Requisite | Description   |
|-----------|---------------|
| DI-01     | short description |
| DI-02     | short description |
| DI-0N     | short description |

3.3.2. Development. In this phase, each of the activities that were planned for each Sprint is developed, direct responsibility of the Team Scrum, where the deliverables are developed incrementally in each Sprint. For which it is necessary that on the backend side implement Django rest framework, and its three fundamental components corresponding to routers, views and serializers; i) the routers are the tools that allow to define the URLs of the API, arranging in an orderly manner a set of URLs for the methods based on HTTP such as POST, GET, PUT, among others; ii) the views are extensions of the Django class-views, which allow you to easily render an html in response, in a Json, XML or other data structure that interests us to return our API; and iii) the serializers allow to define in detail how the responses that our API will return will be and how we will process the content of the requests that come to us.

To install Django Rest framework, you need to install Python, from the command console you enter "virtualenv -p python env", and then pip your dependency handler, induced in console "pip install". The project is created with the command “django-admin startproject” + “project name”. Finally create an application (module) with the command “python manage.py startapp” + “app name”.

In each application (in terms of modularity) created in Django, the data schemas that are subsequently migrated to a database engine are established. SQL allows data management in several engines such as the database Oracle, PostgreSQL, MySQL, MariDB, among others. For a practical case, it is recommended to use one of the most common engines today and, which is compatible with Oracle, we refer to MySQL, for which it is necessary to implement the connector whit the command “pip install mysql-connector-python”; and general settings such as database name such as user, password, engine and auto-commit. Once the schemes are defined, migrations to MySQL can be performed. The files views, serializer and router are configured and in this way, the definition of the migrated database schemas in the respective engine, and the workable Api Web, would be used to consolidate the data in Json or other formats allowed by Django REST framework.

For the development of the frontend, the use of vue.js is required, implementing the npm your dependency manager, in console enter the command “npm install”, equally it is implemented the Vue CLI tool that allows the creation of user interfaces is installed with command “npm install -g @vue/cli”, from components where HTML, CSS and JavaScript are integrated, in addition to building and sharing reusable solutions for needs common. These interfaces will consume the processed data of the API built with Django rest framework, for which it is necessary to use Axios which is an HTTP
client based on promises it is installed with the console command "npm install axios". Beyond consuming and displaying API data, it also communicates with the functions of the server to execute the main actions of CRUD, after configuring the access for it.

In this same phase of development, in order to monitor the progress of each sprint, daily meetings of 15 minutes of obligatory nature are held to discuss the progress and inconveniences that arise, these meetings correspond to the Daily Standup, where the Team Scrum and the Scrum Master. From now on, several Vue components can be implemented to improve the design of the interfaces, such as bootstrap-Vue, which allows the implementation of templates developed with bootstrap for the vue.js components, another recommended library is sweetalert for the visualization of the notifications or alerts on the interfaces, among other libraries that will allow a better experience in terms of use by the end user.

3.3.3. Delivery. At the end of each Sprint, the application of product quality tests begins, defined in the ISO 25000 standard, where quality requirements are evaluated, for verification and incremental delivery of the product to the Product Owner and in cases where relevant Stakeholders are required. Throughout this phase, at the end of each Sprint, the functional requirements, design, performance, and the acceptance criteria of the product are also verified, in addition the progress and the rhythm of delivery the functionalities tested are monitored and the retrospective of the process and the way in which the team works, including techniques, practices and development tools.

The Quality Measurement Division corresponds to ISO/IEC 2502n, these standards include a reference model of product quality measurement, definitions of quality measures (internal, external and in use) and practical guidelines for their application. Table 5 shows the divisions of the selected standards for the methodology.

| Items | Division ISO / IEC | Description |
|-------|-------------------|-------------|
| 1     | 25022             | Measurement of quality in use: specifically defines the metrics to measure the quality in use of the product. |
| 2     | 25023             | Measurement of system and software product quality: specifically defines the metrics for measuring the quality of products and software systems. |
| 3     | 25024             | Measurement of data quality: specifically defines the metrics for measuring data quality. |

4. Evaluation test and resulted.

The methodological guidelines presented solve a problem of integration of frameworks that act in different layers of the software development architecture. The description of each integrated component automates the process of selecting and analyzing the characteristics of the frameworks, as well as their integration, and establishes the procedure to structure, plan and control the software development process. To evaluate the suitability of the proposed methodology to integrate the frameworks, a panel of experts in the area of engineering, architecture and software development was convened, consisting of teachers assigned to the systems engineering undergraduate of the University of La Guajira, academic program It has a high quality accreditation certificate from the Ministry of National Education, a component that provides reliability in the evaluation by the experts convened, and is also a national reference program.
To the panel convened, they were presented with the methodological proposal and also the evaluation criteria, defined in Table 6. Evaluation criteria of the methodological guideline. Subsequently, in Table 7. Results obtained in the evaluation, the consolidation of the evaluation carried out by each invited expert can be observed, in relation to each evaluation criterion.

| Table 6. Evaluation criteria of the methodological guideline |
|-------------------------------------------------------------|
| Items | Evaluation Criteria (EC) |
|-------|--------------------------|
| 1     | The descriptive level of the components of the methodological guidelines is adequate. |
| 2     | It considers the use of methodological guidelines for integration with the frameworks of study work viable. |
| 3     | An adequate solution is represented, for the integration of frameworks for backend and frontend development. |
| 4     | The technologies implemented represent modern or innovative technological tools. |
| 5     | The basic foundations presented are in accordance with the technologies implemented. |

| Table 7. Results obtained in the evaluation |
|--------------------------------------------|
| Items | Results |
|-------|---------|
| Developer Experts (DE) | Score by evaluation criteria (EC) |
| | EC # 1 | EC # 2 | EC # 3 | EC # 4 | EC # 5 |
| DE # 1 | 4.5 | 4.6 | 5.0 | 4.2 | 4.8 |
| DE # 2 | 4.7 | 4.4 | 4.6 | 4.9 | 4.8 |
| DE # 3 | 4.5 | 4.3 | 4.5 | 4.4 | 4.9 |
| DE # 4 | 4.6 | 4.4 | 4.8 | 4.4 | 4.5 |
| DE # 5 | 4.8 | 5.0 | 4.4 | 4.6 | 5.0 |
| AVERAGE | 4.6 | 4.5 | 4.7 | 4.5 | 4.8 |

The results of the evaluation reflect that, the descriptive level of the components that integrate the proposed methodological guidelines is adequate, in addition the use of the defined process with the frameworks of study is considered viable, also, and it allows to show that the process represents a solution convenient, for the integration of innovative technologies implemented in software development. For the rest, in this study the analysis of the characteristics of the theoretical foundations, which allowed to determine the process specifications of each study framework, within the software development architecture is left in this study.

5. Conclusion
In this study, developing effectively methodology to define an integration process between the frameworks implemented for software development, from the agility and quality management approach. In the process, the criteria for selecting the implemented standards and the characteristics or components of each framework for its integration were defined for each stage of the software architecture. Finally, a process map is presented that leads to the result of the integration of the frameworks SCRUM, Django REST framework and Vue.js.
The integration of SCRUM, Django REST framework and Vue.js, has a guide to positively impact the planning, development and delivery of software, providing for each iteration incrementally a product of greater value for the client. The above from the use of the efficient tool that supports the process with greater precision because it allows through the definition of three phases to manage and control each activity of the project; where IEEE standard are integrated to define specifications that are required, the use of framework for development significantly reducing the times of the same and applying methodologies for the agility of the project, but still does not guarantee that the product is of quality; for which they are complemented with ISO standards, which allows to assess compliance with the parameters that are stable to measure the quality of the software product.

References

[1] Platzi. 2017 , Qué es SCRUM y los roles en SCRUM. [En línea]. Available: https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/?utm_source=google&utm_medium=paid&utm_campaign=intention&utm_content=&adgroup=71299386940&gclid=EAIaIQobChMI8IgJTt_OP4wIVx4SzCh0H3giVEAAAYAiAAEgIZefD_BwE.

[2] Django REST framework 2011. [En línea]. Available: https://www.django-rest-framework.org/.

[3] E. You, 2014 VUEJS The Progressive JavaScript Framework, [En línea]. Available: https://vuejs.org/.

[4] Mendez G 2008 Especificacion de Requisitos segun el estandar IEEE 830. [En línea]. Available: https://www.fdi.ucm.es/Profesor/gmendez/docs/is0809/IEEE830.pdf

[5] ISO - Organización Internacional de Normalización, Portal ISO/IEC 25000, 2019. [En línea]. Available: https://iso25000.com/index.php/normas-iso-25000.

[6] Holguien Barrera J A 2015 Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000, Ventana Informatica, nº 32, pp. 151-164.

[7] Herrera Uribe E and Valencia Ayala L E 2007 Del manifiesto ágil sus valores y principios, Scientia et Technica, vol. XIII, nº 34, pp. 381-385.

[8] Mercado Ramos V H, Zapata J and Fernandez Ceballos Y 2015, Tools and good practices for software quality insurance under skilful methodologies, Revista de investigacion desarrollo e innovacion: RIDI, vol. 6, nº 1, pp. 73-83.

[9] Sihuay M, Dávila A and Pessoa M, Factores en la Adopcion de Métodos Ágiles en el Proceso de Desarrollo de Software: Revisión Sistemática de la Literatura.

[10] Vivas L, Cambarieri M, Garcia Martinez N, Petroff M and Muñoz Abbate H 2014 Un Marco de Trabajo para la Integración de Arquitecturas de Software con Metodologías Ágiles de Desarrollo, ResearchGate, pp. 1288-1297.

[11] International Organization for Standardization 2019, “IEEE 42010-2011 - ISO/IEC/IEEE Systems and software engineering -- Architecture description,” 31 10 2011. [Online]. Available: https://www.iso.org/standard/50508.html

[12] PMBOK® 2017 Guide, Guide to the Project Management Body of Knowledge, Pennsylvania: Project Management Institute.

Acknowledgments
The authors thank the help of colleagues from the research group ENAPROT, engineering faculty and to the research Center of the University of La Guajira, and the organizing members of EXPOTECNOLOGÍA 2019 of the Antonio de Arévalo Technological Foundation.