International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Hybrid Method to Develop Software based on Agile and Traditional Methodologies: Requirements, Analysis and Design Phases

Ing. Ayizdeth Fuentes Zarate[1], Ing. Miguel Ángel Muñoz Pozos[2], Dr. René Santaolaya Salgado[3], Dr. José Juan Hernández Mora[4]

[1]Tecnológico Nacional de México/Apizaco, Carrera Apizaco-Tzompantepec. esquina con Av. Instituto Tecnológico s/n,
ConurbadoApizaco-Tzompantepec, Tlaxcala, México C.P. 90300
M13370774[at]apizaco.tecnm.mx

[2]Tecnológico Nacional de México/Apizaco, Carrera Apizaco-Tzompantepec. esquina con Av. Instituto Tecnológico s/n,
ConurbadoApizaco-Tzompantepec, Tlaxcala, México C.P. 90300
M13370809[at]apizaco.tecnm.mx

[3]Tecnológico Nacional de México/CENIDET, Interior Internado Palmira S/N, Palmira, 62490 Cuernavaca, Morelos, Mexico
rene.ss[at]cenidet.tecnm.mx

[4]Tecnológico Nacional de México/Apizaco, Carrera Apizaco-Tzompantepec. esquina con Av. Instituto Tecnológico s/n,
ConurbadoApizaco-Tzompantepec, Tlaxcala, México C.P. 90300
juan.hm[at]apizaco.tecnm.mx

Abstract: ***In this article, a hybrid method is proposed for the needs, analysis and design phases, in software development both traditional methodologies and agile methodologies are involved. In the requirements survey and analysis phase, the delphi method is applied and due to the flow of the stages it resembles the waterfall methodology, during the design it is proposed to use a user-oriented design and a nomenclature standard to define the base of data, at the end of the previous phases obtain a list of priorities that is used in the agile methodology of SCRUM, obtaining as a result a hybrid method (Cascade / Scrum). The main objective of this hybrid method is to generate documentation that makes the coding phase easier for programmers.***

Keywords: Agile method, requirements phase, analysis phase, design phase

1. Introduction

Currently software development has become essential to automate various activities that are carried out in daily life, due to this, the demand to build software products has been increasing exponentially. This led to the development of various techniques and methods that allow this process to be carried out more quickly and with quality. But with the passage of time it has been discovered that it is difficult to completely follow any methodology, because the projects and work teams are very different, as in the article *"Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond"*[1], does a study with various participants who use different development approaches and they combine during practice, showing that most use a traditional model as a framework, connecting it with various agile practices. 83.9% of the participants affirm that the development approach arises from experience, 52.2% that the company defines a standard approach, more than a quarter affirms that the development approaches are selected on a more individual basis and adapted during projects in response to given situations.

The work *"Hybrid Software Development Approaches in Practice: A European Perspective"*[2], conducts an interesting study where they survey different companies that applied hybrid development approaches which arose from the evolution of carrying out different work practices since those established in their workplace did not cover 100% the needs for software development. The foregoing confirms that the needs of each company are different from each other, so relying on a single process for software development is insufficient to meet the needs of the projects.

One of the primary stages in this process is the gathering of requirements, based on *"Application of a hybrid process software requirements management"*[3], software requirements management is essential to achieve expected levels of productivity and quality in software development. This stage must be given more than 50 percent of the estimated time for a project, coupled with this in the "*Impact of Requirements Elicitation Processes on Success of Information System Development Projects*"[4]states that it is much more expensive to correct an error at the coding stage than to correct in the early stages.

For this reason, a hybrid method between agile and traditional methodologies is proposed in the requirements, analysis and design stages, with the aim of generating documentation that will allow the maintenance of the software product to be faster and more efficient in the future. The programmers who perform such activity.

2. Hybrid Method

Agile methodologies provide various benefits to software development, such as: incremental project deliveries, quick releases, flexible to changes in different stages of software development. However, one of its main limitations is the lack of documentation throughout this process, which makes maintenance and updates difficult.

The hybrid method proposed in this article consists of combining the advantages offered by agile methodologies, in this case SCRUM is used: iterations with sprint, priority list and extreme programming methodology (XP) that during coding do pair programming. It also takes advantage of traditional methodologies such as technical documentation. The Delphi method is applied in the requirements survey, analysis and definition of a coding standard.

In the Figure 1shows a diagram that outlines the proposed hybrid method.

It is important to mention that this hybrid method is divided into two jobs, in this article only the first phases for software development will be addressed, which are: requirements survey, analysis, design. The coding, testing and launch phases can be found in the article *"Hybrid Method for Software Development Based On Agile Methodologies in the Coding and Testing Phases"*[5],

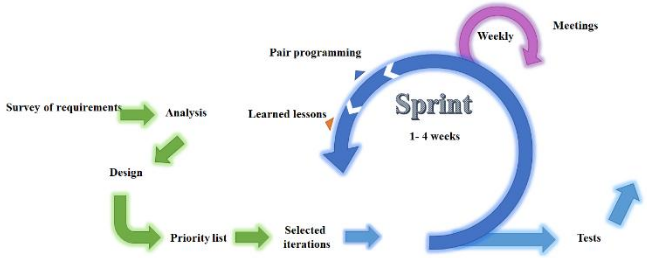


Figure 1: Hybrid method sketch

3. Requirements phase

The requirements survey is one of the most important phases of software development, it is in charge of compiling the client's needs. Within the proposed method, it is defined to carry out the following cycle as observed in the Figure 2these iterations based on the delphi method. In a comparative way, it is mentioned that the delphi method is an iterative process with the participation of people considered experts in order to obtain a consensus according to a common issue or problem, in this case the people who will be involved in the iterative process are the owners of the products and all the people involved in the process in order to obtain more detailed information from different points of view.

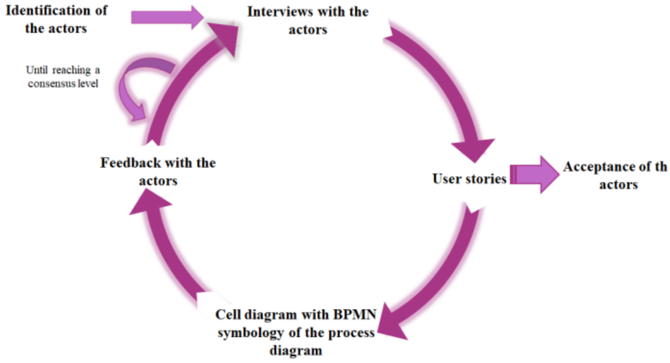

Identification of the actors
Interviews with the actors
Until reaching a consensus level
Feedback with the actors
User stories
Acceptance of th actors
Cell diagram with BPMN symbology of the process diagram

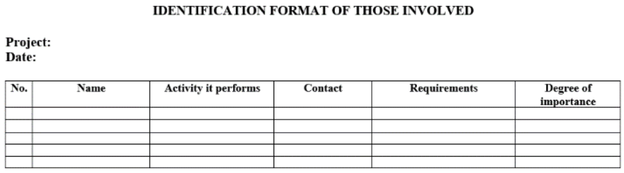
Figure 2: Survey of requirements

The following is a detailed explanation of the application of each of the steps necessary to perform the requirements survey. It should be clarified that as illustrated in Figure 2, the application of the delphi method consists of performing 2 to 3 cycles from the interview to feedback with the actors in order to collect the greatest amount of information in each round and obtain specific details.

1) Identify the sources of the requirements

SWEBOK [6] mentions that in the area of knowledge requirements the requirements have many sources in typical software, and it is important that all potential sources are identified and evaluated for their impact.

Therefore, it is proposed to carry out this identification of the sources in a table to describe the activities carried out by each actor involved in the process, the proposed format is illustrated in the Figure 3.



IDENTIFICATION FORMAT OF THOSE INVOLVED

Project:
Date:

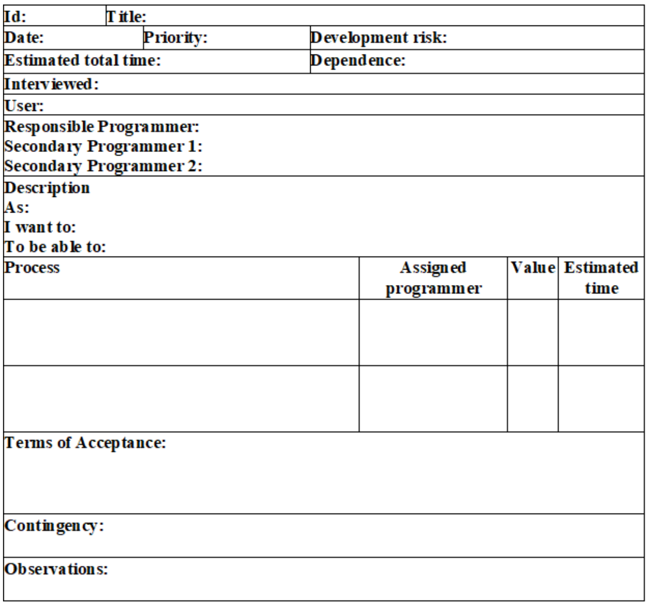
No.	Name	Activity it performs	Contact	Requirements	Degree of importance

Figure 3: Stakeholder identification form

The format consists of briefly and broadly describing the activities carried out by those involved, according to this assigning a degree of importance (high, medium and low) during the process.

2) User stories

In this method it is proposed to make the user stories with a suggested format see the Figure 4.The format was designed with the objective of collecting as much information as possible, adding features that later worked for planning.



Id:	Title:		
Date:	Priority:	Development risk:	
Estimated total time:		Dependence:	
Interviewed:			
User:			
Responsible Programmer: Secondary Programmer 1: Secondary Programmer 2:			
Description As: I want to: To be able to:			
Process	Assigned programmer	Value	Estimated time
Terms of Acceptance:			
Contingency:			
Observations:			

Figure 4: User story format

After completing the user stories, an extremely important step is carried out, which is the realization of the box diagram with BPMN symbology, this allows to visualize the flow of information and the people involved in a general way.

3) Feedback with stakeholders

The delphi method has various uses, in this case it is applied to the requirements survey stage with the purpose of obtaining more detailed customer needs.

Step-by-step method application:

a) *Identification of the problem or issue*: it is briefly explained what the current problem is.
b) *Preparation of the questionnaire:* a questionnaire related to the project to be developed is prepared.
c) *Define the panel of experts or participants in the survey:* for the application of the delphi method in raising requirements, the experts are the people involved in the project who are our source for obtaining the requirements.
d) *Distribute the questionnaire:*this step is covered when conducting the interviews.
e) *Analyze results:* for the analysis of the results, the user stories and the cell diagram are made.
f) *Delivery of analysis to experts:* it is briefly explained to those involved what was obtained from the interviews and if their needs were expressed in the best way.
g) *Second analysis*: new more detailed questionnaires are carried out with questions that resolve doubts about the first application of the interview, to carry out a second analysis and identify more specific details of the requirements and if necessary modify the user stories and the diagram of cells with BPMN symbology.

Carry out the process again from step two to four, with the objective that in the new round, more specific aspects that were omitted in the first interaction with the actors are considered. This process is repeated until a level of consensus is reached. Due to the times the process of trying several times cannot be very long.

4) Acceptance of the actors

In the acceptance by the actors, it is explained in detail how is the process that is currently used to solve some activity and identifying the possible needs that may arise during the use of the software product; If everyone agrees to continue with the next phase, it is reflected in a document where the person in charge of the project and the owner of the process sign with their consent. This step is extremely important since in the future if small changes can be made, but the total change of the project will not be possible since that is why the document is signed where the deliverables are reflected.

4. Analysis Phase

Requirements analysis is the most complex phase during the software life cycle as it is in charge of translating user requirements into operational characteristics of the software. These are usually directed so that programmers can translate them into software.

It is proposed to carry out the following process shown in the Figure 5. At this stage, the delphi method is implemented with the aim of performing various iterations that allow the entire analysis to be carried out in more detail and that when the programmers in charge of the software development are building software, it is easy to understand the process.

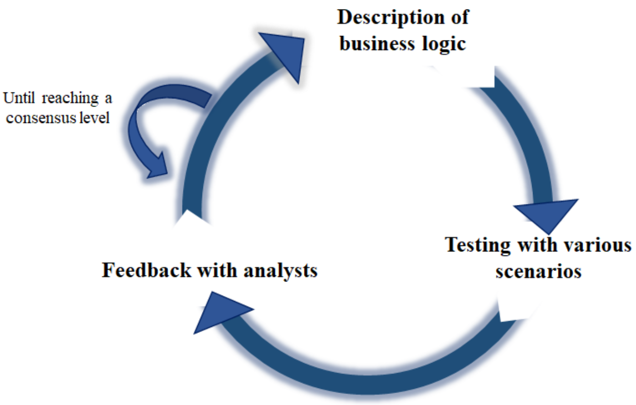
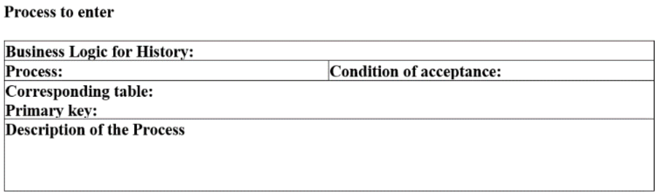

Description of business logic
Until reaching a consensus level
Testing with various scenarios
Feedback with analysts

Figure 5: Requirements analysis

1) Description of business logic

It is in charge of defining the rules that are applied in the real world to instructions so that a software system can understand how it is going to treat information. A format is proposed (Figure 6) that allows the business logic to be clearly organized, it is important to mention that for each user story there must be a business logic.



Process to enter

Business Logic for History:	
Process:	**Condition of acceptance:**
Corresponding table: **Primary key:**	
Description of the Process	

Figure 6: Business logic format

2) Testing with various scenarios

These tests are carried out using the box diagram obtained in the requirements survey and the business logic.

3) Feedback with analysts

On several occasions, carrying out a first analysis is not enough because in principle all the small details are not considered, therefore, carry out steps two and three as a team with the different analysts so that all aspects that were not initially covered are covered. were observed and if the details described in the business logic are easy to understand for a programmer who is not related to the project.

5. Design Phase

In the design process it is of utmost importance to have as much contact as possible with the client and especially with the end user, since this will provide us with better information about the design.

In the proposed model, the design process is shown in the Figure 7.

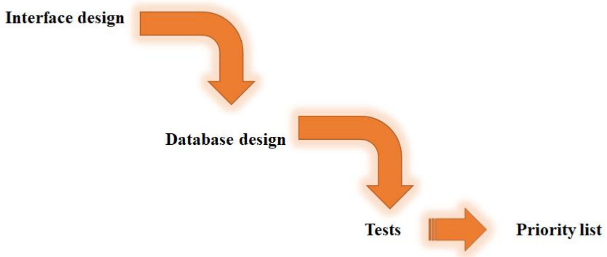


Figure 7: Design

1) Interface design

The interface design is focused on the user whose purpose is to create software products that meet specific needs of end users, obtaining the greatest satisfaction and the best possible user experience with the minimum effort on their part. This process can be done in various ways such as:

- Freehand sketches
- Mockup
- Prototype (Showing the navigation but without the final operation)

2) Database design

The design of the database is largely derived from business logic and is done using a relational diagram and data dictionary. The design must be based on the defined standard to properly name databases, tables, and fields. The following standard is suggested:

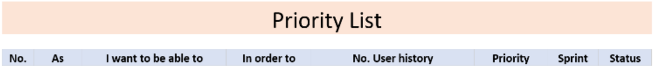
- The name of the database is capitalized.
- The names of tables use camel notation. In camel notation (so called because they look like the humps of a camel) the names of each word start with a capital letter and the rest are written in lowercase. No periods or hyphens are used to separate the words.
- Field names start with lowercase and later use camel notation.
- The names of databases, tables and fields must be meaningful. Write comments to each field.
- Normalize the database.

3) Testing

As it has a user-centered design, the tests to verify the correct operation of the software product are carried out with the end user and according to each user story.

4) Priority list

The priority list allows the deliverables to be in the form of iterations, the requirements are classified by different priorities, using the MoSCoW method and indicating in a column the sprint number to which it belongs. To do the above, the format of the *Figure 8*



Priority List

No.	As	I want to be able to	In order to	No. User history	Priority	Sprint	Status

Figure 8: Format for priority list

6. Application of the proposed hybrid method

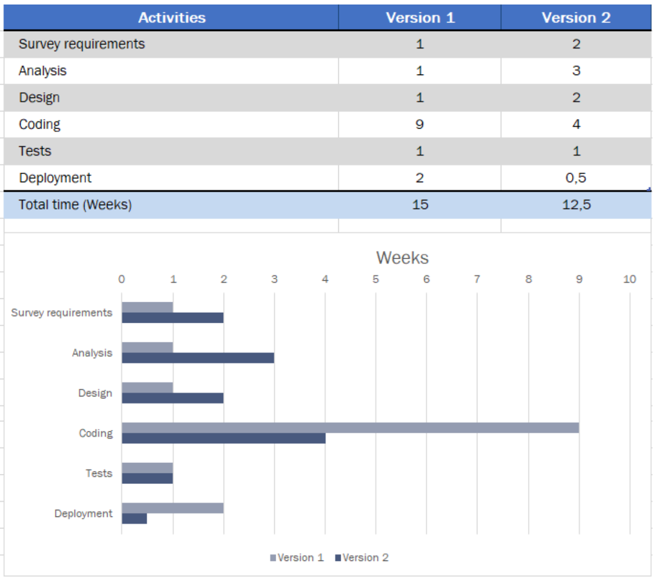
This hybrid method was applied during the migration of an information system called "Teaching Evaluation" for the Technological Institute of Apizaco, and excellent results were obtained.

Teacher evaluation was carried out in two versions, in version 1 methodologies and / or methods that supported software development were not used, in version 2 the proposed hybrid method was used, to clearly visualize the differences that existed during development software, a comparison is shown in the Table 1, this includes the stages of requirements collection, analysis and design.

Table 1: Hybrid method application comparison

Teacher evaluation		
	Version 1	Version 2 Applying hybrid method
Identification of actors	✓	✓
Interview with actors	✓	✓
User stories		✓
Cell diagram		✓
Delphi method application for feedback with actors		✓
Business logic		✓
Delphi method application for feedback with analysts		✓
User-oriented interface design		✓
Database design	✓	✓
Design tests		✓
Generate priority list		✓

Figure 9 shows the approximate execution time of each of the phases of software development. As can be seen, the coding time decreased considerably since more time was dedicated in the first phases and the documentation originating from them allows the software construction process to be faster and more understandable for programmers.



Activities	Version 1	Version 2
Survey requirements	1	2
Analysis	1	3
Design	1	2
Coding	9	4
Tests	1	1
Deployment	2	0,5
Total time (Weeks)	15	12,5

Figure 9: Comparison of software development phases

7. Conclusions

The hybrid method proposed for the requirements, analysis and design phase is proposed with the intention of covering one of the limitations of agile methodologies, which is the lack of documentation. The case study called Teacher Evaluation shows a practical example of the application of the hybrid method in a web-oriented project with a controller view model. The documentation obtained allows the programmer to be clear about the objectives and goals of the code.

Finally, we can conclude that the hybrid method reduced programming errors, which in turn optimized the coding time and, in the end, higher quality software was obtained with documentation that allows easier maintenance and / or changes.

References

[1] Kuhrmann, Marco &Diebold, Philipp & Münch, Jürgen & Tell, Paolo &Garousi, Vahid&Felderer, Michael &Trektere, Kitija& Mccaffery, Fergal&Linssen, Oliver &Hanser, Eckhart & Prause, Christian. (2017). Software and system development in practice: Waterfall, Scrum, and beyond. 10.1145/3084100.3084104.

[2] M. Kuhrmann et al., "Hybrid Software Development Approaches in Practice: A European Perspective," in IEEE Software, vol. 36, no. 4, pp. 20-31, July-Aug. 2019, doi: 10.1109/MS.2018.110161245.

[3] D. Alves da Silva, E. C. de Oliveira, E. Dias Canedo and H. F. Martins, "Application of a hybridprocess software requirements management," 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Las Palmas, 2016, pp. 1-7, doi: 10.1109/CISTI.2016.7521442.

[4] Bormane, Līga & Gržibovska, Jūlija & Bērziša, Solvita & Grabis, Janis. (2016). Impact of Requirements Elicitation Processes on Success of Information System Development Projects. Information Technology and Management Science. 19. 10.1515/itms-2016-0012.

[5] Ing. Miguel Angel Munoz Pozos, Ing. Ayizdeth Fuentes Zarate, Dr. Rene Santaolaya Salgado, Dr Jose Juan Hernandez Mora, "Hybrid Method for Software Development Based On Agile Methodologies in the Coding and Testing Phases", International Journal of ScienceandResearch (IJSR), https://www.ijsr.net/search_index_results_paperid.php?id=SR201119084738, Volume 9 Issue 11, November 2020, 1085 – 1089

[6] RE Fairley, P. B. (2014). Guide to the Software Engineering Body of Knowledge, Versión 3.0. IEEE Computer Society. http://www.swebok.org

Author Profile

Ayizdeth Fuentes Zarate received the B.I. degree in Engineering in information and communication technologies from Technological Institute of Apizaco in 2018. During 2016-2019, she stayed in Computer Center from the same institute to develop Web-based information systems to use of students, teachers and administrative staff. She now study Master's degree in Computer Systems.

Miguel Angel Muñoz Pozos received the degree of Eng. In Information and Communication Technologies from Tecnologico de Apizaco in 2018, during 2013-2018 he belonged to the degree in Information and Communication Technologies, from August 2018 to date he is In the division of postgraduate studies, he completed a master's degree in computer systems.

René Santaolaya Salgadois professor of Technological Research and Develop National Center (CENIDET), Cuernavaca, Morelos, México. His research interest includes software process model, software architecture, software reuse and web services. René Santaolaya received a PhD in Computer Science (Software Engineering) from National Polytechnic Institute, Computing Research Center. He is a senior member of the IEEE. He has been developer, analyst and project leader in the Academic Body of Software Engineer.

José Juan Hernández Mora has a Doctor of Teaching Excellence degree from the University of Los Angeles, 2019. He also has a degree in Computer engineer from the Universidad Autónoma de Tlaxcala, from 1994. Master's in computer science at the National Center for Research and Technological Development of the TecNM, 2003. Research professor at the Tecnológico de Apizaco del TecNM. Teacher of the Master of computer systems of the TecNM / Instituto Tecnológico de Apizaco.