

Hermeneutical Elicitation of Requirements: A Technical Perspective to Improve the Conception of the Software Requirements

Wagner Varalda, Ítalo Santiago Vega

Program of Intelligence Technologies and Digital Design, Pontifical Catholic University of São Paulo, São Paulo, Brazil

Email: wagnervaralda@uol.com.br, italo@pucsp.br

How to cite this paper: Varalda, W. and Vega, Í.S. (2018) Hermeneutical Elicitation of Requirements: A Technical Perspective to Improve the Conception of the Software Requirements. *Journal of Computer and Communications*, 6, 132-149.

<https://doi.org/10.4236/jcc.2018.69010>

Received: August 6, 2018

Accepted: September 25, 2018

Published: September 28, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In order to develop quality software that meets the original needs of its users, it is necessary to perform the Requirements Engineering, so that the software context to be developed is identified, examined and specified properly. However, there is a problem that is increasingly in debate: the difficulty in understanding and establishing the purpose of the software to be developed, as pointed out by important researches in the area, such as the Chaos Report, which indicates that only 29% of software projects are successful, and the Software Engineering Institute, which points out software requirements as a critical factor for the success of software engineering and that deficiencies in this dimension are the main causes of software project failures. This article presents a proposal to address this problem through the use of the Hermeneutical Elicitation of Requirements, which is the conceptual adequacy of some hermeneutical methods in a technical approach that assists the requirements engineer to conceive better of the software requirements. In this way, the software engineer will be better able to develop the software to better meet the needs of its end users and sponsors.

Keywords

Software Engineering, Requirements Engineering, Elicitation of Requirements, Hermeneutics, Dasein

1. Introduction

Software Engineering aims to develop quality software that meets the original needs of its users and is produced according to the established deadlines and costs. In order to establish what the software must do, it is necessary to perform the Requirement Engineering activity so that its context is identified, examined

and properly specified. The other activities for software development have a strong dependence on this activity. However, an important problem is occurring more frequently: the difficulty in understanding and describe the needs of the business to be met and satisfied by the software to be developed, as well as the business opportunities to be enjoyed with their support. Several researches in the areas point out and warn about this problem [1]-[9].

The Requirements Elicitation is the initial activity of Requirements Engineering and aims to understand the problems that the software must solve and the business opportunities that can be enjoyed with its support. This should be done through an effective and consistent communication between the requirements engineer and the people of the business areas interested in software development, to witch establish the scope of the project, where it will contain the description of the business needs that must be met and satisfied by the software to be developed, the views of each person of the business areas, in relation from to the problems, impacts, solutions, opportunities and business rules that will define and restrict the structural and behavioral aspects of the software, the operating environment in which the software will run, the organizational environment, describing the business processes, cultures and policies of the organization for which the software will be developed, and the domain of the application, where the ontological approach is described, which will conceptually base the software requirements and the formulation of the objective of its project [3] [4] [5].

The Hermeneutical Elicitation of Requirements is a proposal whose purpose is to ensure that the problems that the software must solve and the business opportunities that can be enjoyed with its support are understood and interpreted in a reliable way to the reality for which it will act and that is faithfully produced the scope of your project, containing the correct view of the software to be developed. This is possible because of the appropriateness of some hermeneutical methods derived from the hermeneutic philosophy established by Martin Heidegger (1889-1976), more specifically those related to Dasein, where it has as central reflection the understanding of the sense of being, concept from where they originated the essential elements to the composition of the Hermeneutical Elicitation of Requirements [10] [11] [12].

Similar works that are closer to the proposal presented in this article use Peirceana Semiotics as theoretical foundation. Several works have been developed applying its triad of elements that help in the understanding of signs. These researches generally meet the problems related to human-machine interaction and machine learning [13]. There are also the SBVR standard, which defines the vocabulary and rules for documenting the semantics of business vocabularies, business facts, and business rules; as well as a schema for to do interchange of business vocabularies and business rules among organizations and between software tools [14].

However, problems related to the elicitation of software requirements are not addressed by these surveys. Such problems can be better solved by the applica-

tion of hermeneutics, since they refer to questions involving understanding and interpretation. This article presents the revision and updating of the proposal presented in the article *Hermeneutical Engineering of Requirements* [15], elaborated by its own authors, to align its concepts to the project *Hermeneutical Engineering of Software*, which is under development.

2. Hermeneutic Philosophy by Martin Heidegger

2.1. Introduction

The main objects of study of Hermeneutics are understanding and interpretation. It deals with the definition of processes that decipher the meanings of things, to assure the interpreter who applies them to identify, analyze and truly know the laws, structure, behavior and context about the things he seeks to purchase knowledge. However, there is no single or unified hermeneutic methodology that is capable of being applied in all situations. That is why, more specialized hermeneutical methods have emerged (and continue to emerge), which are more assertive and systematic in their fields of action, such as, for example, theological hermeneutics, philosophical hermeneutics, legal hermeneutics and clinical hermeneutics [16].

In the case of philosophical hermeneutics, his present state of art is the result of various contributions made by various thinkers and philosophers over time, with highlights to Friedrich Schleiermacher (1768-1834), Wilhelm Dilthey (1833-1911), Martin Heidegger (1889-1976), Hans-Georg Gadamer (1900-2002) and Paul Ricoeur (1913-2005), but not limited to these. With this, philosophical hermeneutics has reached and established a reflective thought that goes to meet the true interpretation contextualized understanding of what one intends to know [16].

With Martin Heidegger, hermeneutical thinking turns to the questions of the human being. In his work “Being and Time”, launched in 1927, the essence of his philosophy lies in the understanding of the sense of to be, which develops through the central theme *Dasein* [17].

2.2. Dasein

Dasein is a German term which, for Heidegger, is the essence that constitutes the mode of being of man, in an ontic and ontological way, enabling him to be what he is. Here the term “ontological” indicates the various possibilities of understanding that man can have of himself, of others and of all other things. The term “ontic” refers to the possibility that man has to understand himself in a more direct and immediate way, in order to meet his roots [10] [11].

Heidegger does not refer to each specific man as *Dasein*, but as *Ente*, which is the concept that singles out and personalizes each man. But this *Ente* is not something already ready, already defined. In reality, it is always in motion and in constant transformation, for it always comes to encounter new possibilities of molding its character and its existence. And this occurs from his birth until the

moment of his death, in the various relationships that he makes. Therefore, understanding is also understanding these relationships [12] [16].

That is why the understanding of Dasein is based on the understanding of three other fundamental concepts: “being-in-the-world”, “being-with-others”, and “being-to-death”, as illustrated by the triad shown in **Figure 1**.

2.3. “Being-in-the-World”

According to Heidegger, the world is not a space or a place in which Dasein is there, but rather a set of relations that give meaning to its existence, which influences and determines its behavior and all its ways of being and acting. This world, which is being contextualized from ancient times to the present, which is the historical world marked by temporality, also helps to contextualize Dasein, giving it meaning and shaping its existence and experience [10] [11].

Therefore, the understanding and interpretation of Dasein also occurs through the understanding and interpretation of the Dasein-World relationship. The openness in this relationship offers to Dasein the possibility of meeting people and all other things around them and thus discovering new relationships and finding new ways to overcome their limits, and with this, remodel and re-structure its existence. To this constant discovery and appropriation of new possibilities that are at any moment unveiled to Dasein, Heidegger gave the name of Ereignis [12].

While “being-in-the-world”, the Ente relates to other people and other things through the relationship of use and handling, that is, these other people and these other things are seen as instruments available to the Ente to do your tasks and your projects. In turn, these instruments occupied by the Ente relate to each other and refer to each other to form the world of Dasein [18] [19].

Figure 2 illustrates the triad that represents the concept of Ereignis and **Figure 3** illustrates the concept of “being-in-the-world”.

2.4. “Being-with-Others”

This concept indicates the relationship of coexistence. The openness that exists in the Dasein-World relationship allows the Entes to find a mutual, transcendental encounter that does not necessarily occur because of their physical proximity but

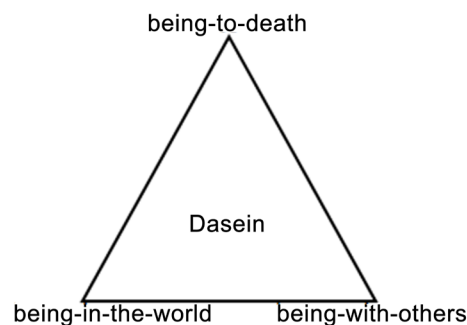


Figure 1. Triad of Dasein.

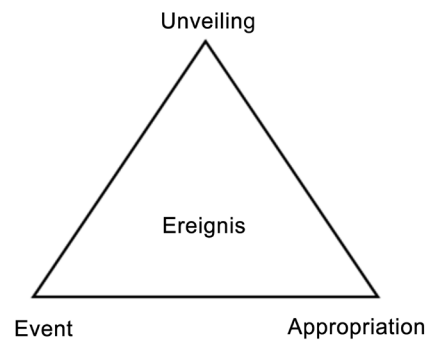


Figure 2. Concept of Ereignis.

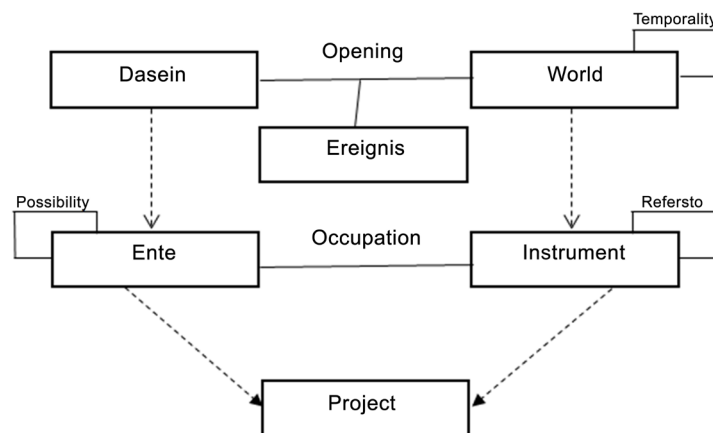


Figure 3. Concept “being-in-the-world”.

because of the harmony, intimacy and familiarity that exists between them. The historical world, marked and contextualized by temporality, is no longer the current, present and everyday world, where occasional encounters occur at all times. This world, now, is the circumstantial world, cohabited by these harmonic, intimate and familiar Entes [10] [11].

The other is no longer that Ente simply given as something that is available to be used and handled for the realization of the projects. The relationship, now, is no longer a relation of occupation, but of concern, where each between shares their life projects with each other and, together, carries out their individual projects in a collaborative way. Each Ente, finally, knows its importance and also the importance of the other, and so everything that does and produces for itself is also made and produced for the other, because it has usefulness, importance and meaning for both [12].

This other, called by Heidegger of Mitsein contributes to the composition of the ontological character of Dasein and gives meaning to its existence. Therefore, the understanding of being requires the understanding of this relationship guided by the attitudes of the Entes that occur through the coexistence that exists between them and their mutual presence in the circumstantial world [18] [20].

Figure 4 illustrates the concept of “being-with-others”.

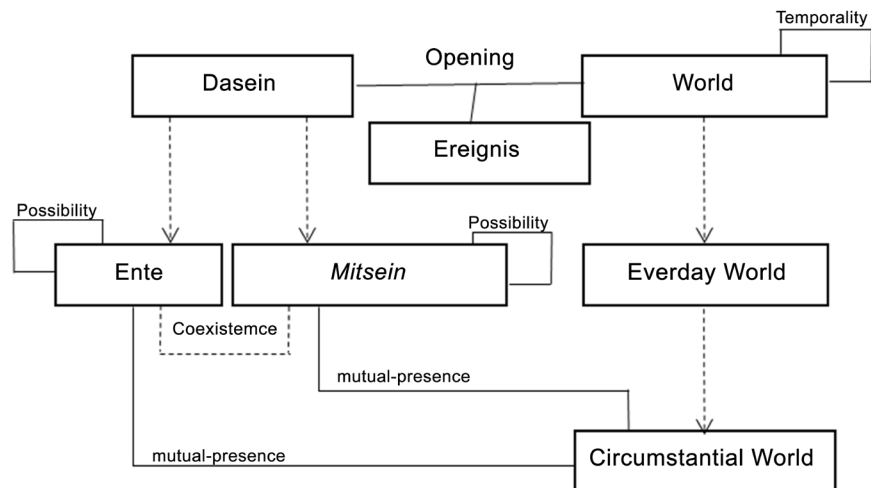


Figure 4. Concept “being-with-others”.

2.5. “Being-to-Death”

Concept indicates the existential limit of the being and allows him to be understood in its totality. Each Ente experiences, in its own way, the death that occurs with the other and this makes him understand his situation of finitude. Death, then, functions as the conditional parameter to the existence of Dasein and gives meaning to his life project, showing him that one day he will no longer be a “being-in-the-world” nor a “being-with-others” [10] [11].

It is because of this vision of existential limit that Dasein seeks to give meaning to the phenomena of life, such as the phenomenon of existence, that of death and that of language. Existence arouses curiosity, death creates anguish, and language allows him to understand the essence of being, through communicability [12].

Because of curiosity, Dasein seeks and discovers new possibilities that allow them to overcome challenges and to better understand their essence. Because of the anguish, Dasein attains the awareness that it is necessary to transcend the ephemeral feelings that give rise to them anguish and meet their authentic existence, according to the possibilities it projects and realizes for itself. Because of communicability, everything can be understood and interpreted. This is where the message of the meanings of things is conveyed [18] [20].

Figure 5 illustrates the concept “being-to-death”.

3. Hermeneutical Elicitation of Requirements

3.1. Introduction

At its core, the Hermeneutical Elicitation of Requirements is a technical approach that assists the requirements engineer to understand and interpret the original needs to be met by the software that will be developed, enabling the requirements of this software to be correctly identified, examined, and specified. This is possible because of the assurance that the problems that the software will

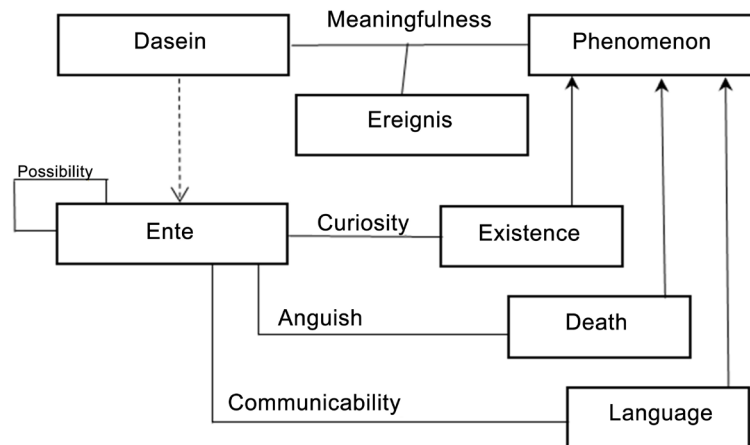


Figure 5. Concept “being-to-death”.

solve and the business opportunities that can be enjoyed with its support are understood and interpreted reliably to the reality for which it will act.

In applying Hermeneutical Elicitation of Requirements, the ontological sense of the application domain to which the software will be developed is truly understood and interpreted. With this, it is possible to correctly establish the scope of the project, which will contain the exact vision of the software to be developed, so that it reaches the best level of quality, reaching satisfactorily, and even exceeding, the expectations of its users.

Just as specialized hermeneutical methods emerged in specific areas (examples: theological hermeneutics, philosophical hermeneutics, legal hermeneutics and clinical hermeneutics), saw himself the need and the opportunity to develop a specialized hermeneutics in Software Engineering (a project which is in under development). The Hermeneutical Elicitation of Requirements is an integral part of this project, which has already been completed and is presented in this article.

3.2. Conceptual Adequacy

Some concepts originating from Heidegger’s hermeneutical philosophy base the composition of the Hermeneutical Elicitation of Requirements. These concepts were specifically adapted to act in the understanding and interpretation of the problems that the software must solve and of the business opportunities that can be enjoyed with its support, and, from there, support and guide the requirements engineer in the production of project scope, to ensure that it contains the correct view of the software being developed, which will describe the business needs that must be met by this software, the views of each person of the business areas interested, in relation from to the problems, impacts, solutions, opportunities and business rules that will define and restrict the structural and behavioral aspects of the software, the operating environment in which the software will run, the organizational environment, describing the business processes, cultures and policies of the organization for which the software will be developed, and the domain of the application, where the ontological approach is described,

which will conceptualizes the base the software requirements and the formulation of the objective of its project.

As with the Dasein concept, similarly the Hermeneutical Elicitation of Requirements offers mechanisms that allow the understanding and interpretation of the essence of the context for which the software will be developed. Both ontic and ontologically, that is, it enables the direct and immediate understanding and interpretation of the roots of this context, as well as the understanding and interpretation of everything that coexists and relates to it and in what way this coexistence and these related functions occur.

The central study object of the Hermeneutical Elicitation of Requirement is the Situational Difference, which turns out to be the event that triggers an inappropriate behavior, and/or non-standard, and/or not in accordance with the planning. It is from the Situational Differences that begin the processes of understanding and interpreting the problems to be solved and the business opportunities to be enjoyed.

For this, the triad of the Hermeneutical Elicitation of Requirements is constituted by the key concepts “Situational Difference Identification”, “Situational Difference Examination” and “Hermeneutical Specification of Requirement”, as shown in **Figure 6**.

Figure 7 illustrates the Event triad.

3.3. Situational Difference Identification

This concept refers to the gaze focused on a business community and/or a business

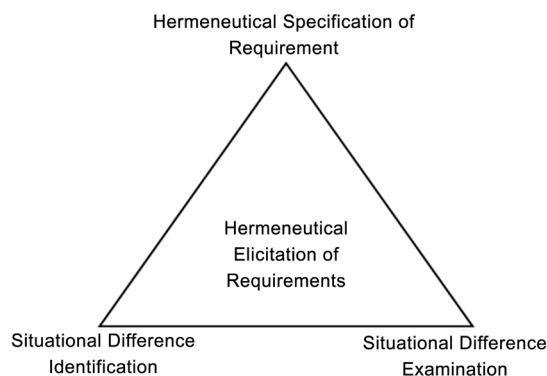


Figure 6. Triad of the hermeneutical elicitation of requirements.

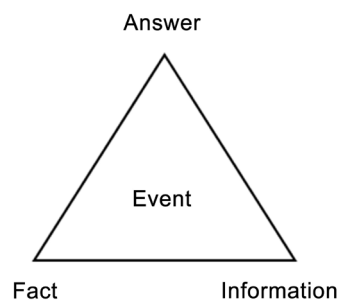


Figure 7. Triad of event.

area with the purpose of understanding and interpreting their structures and dynamics in a contextualized way so as to identify and thoroughly investigate their situational differences.

The situational difference depends on the events that occur between it and the business community and/or business area to which it is associated. Therefore, their identification is made through the analysis of these events. A situational difference cannot be analyzed separately from its area of business and/or business area, because its contextualization depends on the relationships that occur among each other. There is no situational difference without business area and/or business community and there is also no business area and/or business community with no situational difference.

A business community and/or business area is formed by individuals and/or legal entities, who are interested in the development of the software. It is these people who must be identified by the requirements engineer so that the organizational context for which the software will be developed is understood. These people interact with each other in a collaborative and/or contributory way to accomplish their tasks. Therefore, all these people should be consulted by the requirements engineer, for a better and more efficient understanding and interpretation of this organizational context. A situational difference manifests itself as problems and/or opportunities, which are perceived in different ways by each of the stakeholders. This is why each stakeholder can have a differentiated point of view in relation to these problems and/or opportunities. Therefore, in order to better understand the context of problems and/or opportunities, the requirements engineer must understand all stakeholders' points of view.

Through analysis of business plans and/or business processes, it is also possible to identify situational differences and, if such artifacts do not exist or are out of date, it is suggested that they be elaborated so that they are described and mapped in details of the relationships between stakeholders and problems/opportunities, and thereby record the procedures performed to deal with these problems and/or opportunities, as also to describe how to solve them, mitigate them and/or reproduce them according to the possibilities and benefits known.

When applying the Situational Difference Identification, the requirements engineer becomes better acquainted with the application domain and thus becomes more able to specify the software requirements, as he appropriates more widely and deeply from each of his elements explained here.

The Dasein concept: being-in-the-world (Section 2.3) grounds conceptually the composition of Situational Difference Identification.

Figure 8 illustrates the elements that make up the Situational Difference Identification. It is suggested to compare it with **Figure 3**, which illustrates the Dasein concept: being-in-the-world.

3.4. Situational Difference Examination

This concept refers to the gaze focused on the environment in which the situational

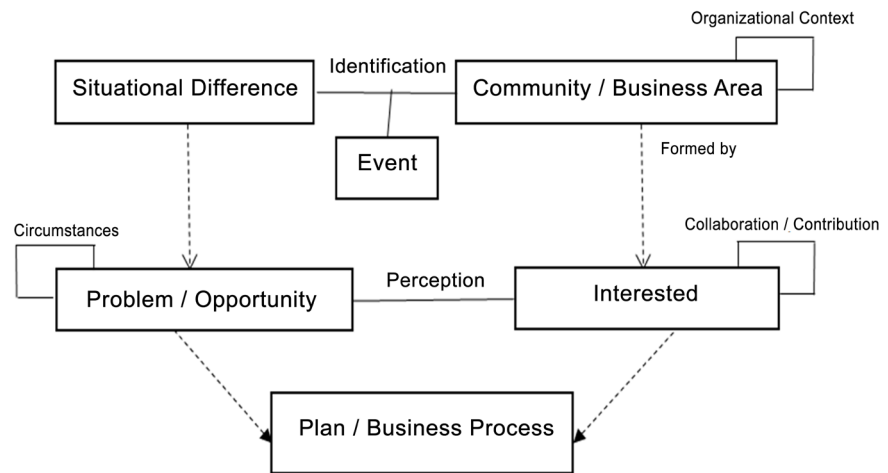


Figure 8. Situational difference identification.

difference is inserted. This is necessary in order for its ontological character to be understood and interpreted. This environment involves the culture that governs all the attitudes that take place in the organizational context that is related to the situational difference.

This culture establishes the rules, policies, goals, strategies, processes, practices, information, standards and control mechanisms that shape the behaviors of this environment, in which is performed the activities and tasks responsible by production of the utensils (or inputs). Each utensil (or input) refers to other utensils (or inputs) and are co-present in the environment to form their instrumental whole.

This environment is influenced by its global environment, which causes changes and adaptations in its culture. This global environment is formed by factors external to the environment, but which relate to it in some way. These external factors are of the most varied types, can be, but not be limited to: customers, competitors, suppliers, regulators, economics, demography, technology, sociocultural environment, and foreign laws and policies.

The global environment is also influenced by external factors that cause in him changes and adaptations. In the case, these external factors belong to the nature of the business, from which originated and determined the root purpose of the business area and/or business community. For example, an automobile industry and a footwear industry, both are of the same nature: industry, so their global environments are impacted by the industrial system. A kindergarten and a university, both are of the same nature: education, so their global environments are impacted by the educational system. Therefore, the understanding of the scenario of the business area and/or business community also depends on the understanding of its nature.

When applying the Situational Difference Exam, the requirements engineer raises his/her maturity level in relation to the application domain and is better able to specify the software requirements, as his/her understanding and inter-

pretation of the elements explained herein increases.

The Dasein concept: being-with-others (Section 2.4) grounds conceptually the composition of Situational Difference Examination.

Figure 9 illustrates the elements that make up the Situational Difference Examination. It is suggested to compare it with **Figure 4**, which illustrates the Dasein concept: being-with-others.

3.5. Hermeneutical Specification of Requirement

This concept refers to the focus on defining the scope of the software project to be developed. The existential limit of this scope was understood and interpreted in its totality through the applications of the Identification of Situational Difference and the Examination of Situational Difference. At this point, the conception of software requirements can (and should) be specified.

Through interactions between the requirements engineer and the stakeholders of the Community/Business Area, which occur through communication based on the context of the situational difference, the original needs (together with the respective expectations of stakeholders) are stated, the specification acceptable is produced and the software requirements is specified.

This communication based on the context of the situational difference functions as a conditional parameter to the understanding and interpretation of the elements described in the Situational Difference Identification and the Situational Difference Exam (respectively, **Figure 8** and **Figure 9**) and gives meaning to the software development project. The requirements engineer and the stakeholders of the Community/Business Area should be guided by these guidelines.

The original needs (along with stakeholder expectations) state the original needs to be met by the software, identified with the stakeholders about their perceptions and their points of view regarding the problems and/or opportunities, where the guidelines and suggestions on how to proceed in relation to each situation should be recorded, highlighting the impacts and the indicated solutions.

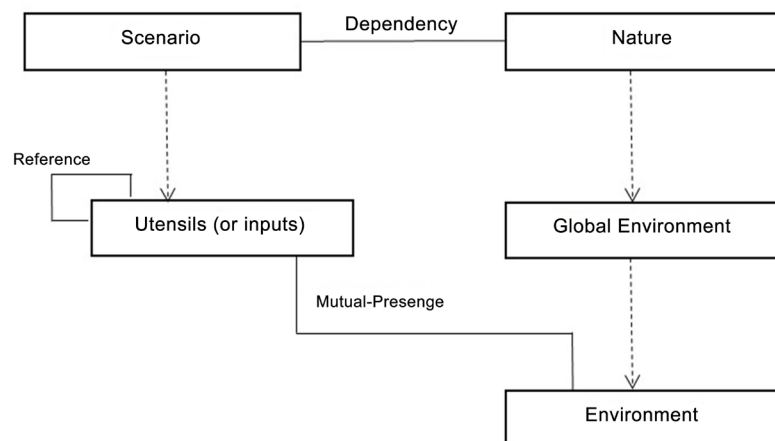


Figure 9. Situational difference examination.

The acceptable specification is produced from the original needs (and respective expectations). It contains the general scope of the software to be developed, which contains: a description of the business needs that must be met and satisfied by the software being developed, the views of each person in the business areas concerned, the problems, impacts, solutions and opportunities, business rules that will define and restrict the structural and behavioral aspects of the software, the operating environment in which the software will be run, the organizational environment, describing the organization's business processes, which software will be developed, and the domain of the application, where the ontological approach is described, which will conceptually base the software requirements and the formulation of the objective of its project.

The software requirements are specified according to the general scope of the software, produced by the acceptable specification, and contains: the descriptions of the tasks to be performed by the software, the information to be processed by the software, the rules that the software should obey, and the interactions the software will make with its external environment. With that, it is established what the software to be developed should do and how it should behave. The evolution of this artifact takes place during the accomplishment of the other activities of the Requirements Engineering, subsequent to Elicitation of Requirements.

While, on the one hand, the requirements engineer produces these artifacts (original needs, stakeholder expectations, acceptable specification and specification of software requirements), on the other hand, stakeholders of the Community/Business Area validate and approve them.

When applying the Hermeneutical Specification of Requirements, the requirements engineer specifies best the software requirements to be developed, containing clear, complete, sufficient, correct and compatible information with the application domain that originated and generated its existence, having the endorsement and approval of the interested in the development of this software.

The Dasein concept: being-to-death (Section 2.5) grounds conceptually the composition of Hermeneutical Specification of Requirements.

Figure 10 illustrates the elements that make up the Hermeneutical Specification of Requirements. It is suggested to compare it with **Figure 5**, which illustrates the Dasein concept: being-to-death.

4. Application of Hermeneutical Elicitation of Requirements

Is possible apply the Hermeneutical Elicitation of Requirements in any software development project, regardless of your application domain and degree of complexity. As an example, its application is presented in a software development project for the "Memory Game".

The decision to choose the "Game of Memory" was taken by the fact that it is a game well known and easy to understand. Thus, the application of the Hermeneutical Elicitation of Requirements is presented more directly, without the need for large explanations about the application domain.

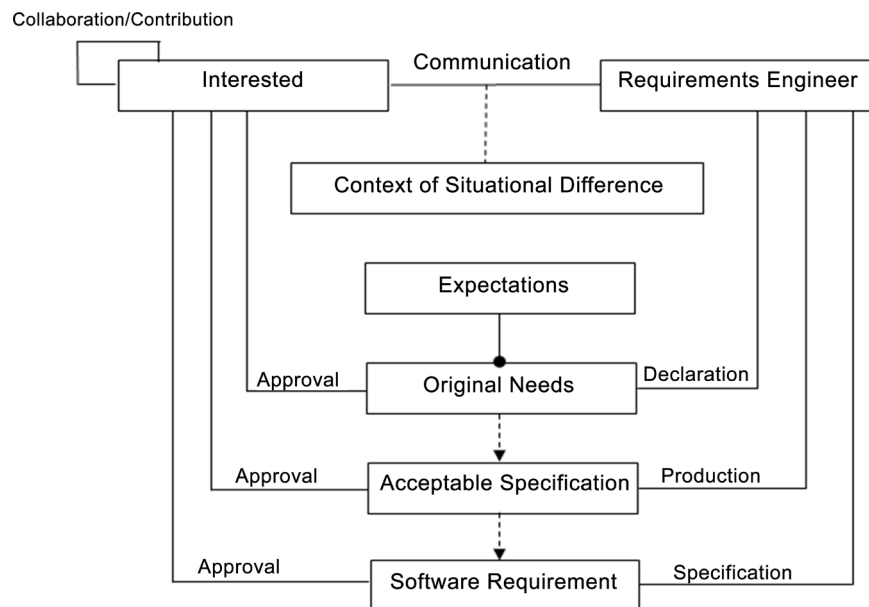


Figure 10. Hermeneutical specification of requirements.

The “Game of Memory”, in its essence, is a game of pieces (usually cards) that contains varied figures on one of its faces. Each figure is repeated into two cards. To start the game, shuffle the pieces and distribute them with the faces of the figures facedown, so that the players do not see them. Each player, one at a time, unlocks two pieces, so that all other players also see them and know which figures have been revealed by these pieces. If the figures revealed by these pieces are the same, the player who took them off collects them for themselves and continues to play. If the figures of these pieces are different, the player who revealed them again gets them in their respective places and the time to play is by another player, who will repeat the process in order to also find pieces with equal pairs of figures. This cycle repeats until there are no more pieces to be untapped. The player who manages to collect plus equal pieces wins the game [21].

Given this brief description of the “Game of Memory”, the software “Electronic Memory Game” should automate this scenario, taking into account four particularities: 1) The game should be played by only one player. 2) The player can choose difficulty levels (determined by number of pieces, ranging from 10 to 40). 3) The player may stop a match to continue playing at another time. 4) The player may start a new match at any time.

Table 1 shows the result of applying the Situational Difference Identification.

Table 2 shows the result of applying the Situational Difference Examination.

Table 3 shows the result of applying the Hermeneutical Specification of Requirements.

Table 4 shows the result of applying the Hermeneutical Specification of Requirements (Software Requirements).

Table 1. Result of applying the situational difference identification.

Situational Difference Identification	
Situational Difference	Electronic Memory Game
Community/Business Area	Memory Game Players
Problem/Opportunity	Promote entertainment among those involved.
Interested	At each match, two to four Memory Game Players.
Plan/Business Process	<p>The “Game of Memory”, in its essence, is a game of pieces (usually cards) that contains varied figures on one of its faces. Each figure is repeated into two cards. To start the game, shuffle the pieces and distribute them with the faces of the figures facedown, so that the players do not see them. Each player, one at a time, unlocks two pieces, so that all other players also see them and know which figures have been revealed by these pieces. If the figures revealed by these pieces are the same, the player who took them off collects them for themselves and continues to play. If the figures of these pieces are different, the player who revealed them again gets them in their respective places and the time to play is by another player, who will repeat the process in order to also find pieces with equal pairs of figures. This cycle repeats until there are no more pieces to be untapped. The player who manages to collect plus equal pieces wins the game.</p>

Table 2. Result of applying the situational difference examination.

Situational Difference Examination	
Scenario	Electronic Memory Game
Nature	The Memory Game is a game of the traditional genre.
Global Environment	Pieces containing varying figures on one of their faces. Each figure is repeated into two cards.
Environment	Virtual.
Utensils (or inputs)	<ul style="list-style-type: none"> • One Memory Game Player; • One Virtual Memory Game Player; • One Board; • Forty pieces with varied figures on one of their faces (each figure is repeated in two cards); • A Scoreboard to indicate the number of wins for each player.

Table 3. Result of applying the hermeneutical specification of requirements.

Hermeneutical Specification of Requirements	
Original Needs	To develop a virtual game of the type Memory Game, of the traditional genre, with the intention of promoting entertainment to the players of Game of Memory.
Expectations	Achieve playing memory game in virtual environments.
Acceptable Specification (description)	<p>The Memory Game Player will play Memory Game matches with the Virtual Memory Game Player. The Memory Game Player may choose to be “Player One” (the first player to make the move) or “Player Two” (the second player to make the move). The score will mark the number of times won by “Player One” and “Player Two.” These numbers will start with zero as soon as the Memory Game is loaded.</p>

Continued

Acceptable Specification (dynamics)	<p>1) When starting the game, the board will have all the pieces shuffled and distributed with their faces facing down.</p> <p>2) "Player One" performs the play by untacking two pieces, one at a time, in their respective places on the board.</p> <p>3) After "Player One" perform your move:</p> <p>a) If there are two pieces with equal figures, such pieces are collected for "Player One" and the score of this player is increased by one more point. If there are more pieces to be untapped, the turn to play is "Player One". Otherwise, the game is finish.</p> <p>b) If it is not two pieces with equal figures, such pieces are untapped again in their respective places and the time of play is of "Player Two".</p> <p>4) After "Player Two" makes his move:</p> <p>a) If there are two pieces with equal figures, such pieces are collected for "Player Two" and the score of this player is increased by one more point. If there are more pieces to be untapped, the time to play is "Player Two". Otherwise, the game is finish.</p> <p>b) If it is not two pieces with equal figures, such pieces are untapped again in their respective places and the time of play is of "Player One".</p>
--	--

Table 4. Result of applying the hermeneutical specification of requirements (Software Requirements).

	Hermeneutical Specification of Requirements Software Requirements (SR)
SR-1: Choose Player	The Memory Game Player chooses to play as "Player One" or as "Player Two." This choice cannot be made with the match in progress. The scoreboard is influenced by that choice, <i>i.e.</i> the score of the "Player One" is replaced by the score of the "Player Two" and vice versa.
SR-2: Restart Scoreboard	The Memory Game Player chooses to restart the scoreboard. This action causes "Player One" and "Player Two" scores to be zeroed. This action can be taken at any time of the match and the Memory Game Player must confirm this action before the scoreboard is restarted.
SR-3: Start Game	The Memory Game Player chooses to start a new game. This action may be taken at any time and the Memory Game Player must confirm this action. When starting the new match, all pieces will be shuffled and randomly distributed with their faces facing down. The results of the scores will remain the same.
SR-4: Reveal First Card	"Player One" or "Player Two", depending on the turn to play, analyzes the board and reveal one of its pieces.
SR-5: Reveal Second Card	The player who revealed the first piece analyzes the board and revealed another one of its pieces. If the figure of this piece that was untapped is equal to the figure of the first untapped piece, these two pieces are collected for the player who revealed them over and their respective score is increased by one more point. If the figure of this piece that was revealed does not equal the figure of the first revealed piece, these two pieces are returned in their respective places of the board.
SR-6: Save Scoreboard	The Memory Game Player chooses to save the scoreboard and, after confirming this action, the results of players One and Two are recorded for later retrieval. This action can be triggered at any time.
SR-7: Recober Scoreboard	The Memory Game Player chooses to retrieve the scoreboard and, after confirming this action, the last saved results of the One and Two players are recovered, replacing the results presented so far on the scoreboard. This action can be triggered at any time.

Continued

SR-8: Choose Level
of Difficulty

The Memory Game Player selects a level of difficulty for the game (by default, this level is preset to “1”). The level of difficulty is determined by the number of pieces, having “Level 1” ten pieces, “Level 2” twenty pieces, “Level 3” thirty pieces and “Level 4” forty pieces. This action can be triggered at any time.

5. Conclusions

In order to develop quality software, one must first understand and establish for what purpose it will be developed. However, this has been a major problem presented by software engineers: precisely, the difficulty in understanding and interpreting the problems to be solved by this software and the business opportunities that can be enjoyed with their support.

The Requirements Elicitation is the first activity of Requirements Engineering and has the purpose of conception the requirements of the software to be developed. For this, it is necessary that the problems that the software must solve and the business opportunities that can be enjoyed with their support are understood and interpreted in a reliable way to the reality and context for which this software will be developed.

The Hermeneutical Elicitation of Requirements assists the requirements engineer in this endeavor by providing a technical approach based on hermeneutical methods that ensure that the problems that the software must solve and the business opportunities that can be enjoyed with its support are understood and interpreted correctly and that it is produced the scope of your project, containing the correct view of the software to be developed. This is possible because of the appropriateness of some hermeneutical methods derived from the hermeneutic philosophy established by Martin Heidegger (1889-1976), more specifically those concerning Dasein, where it has as central reflection the understanding of the sense of being, concept from which they originated the essential elements to the composition of the Hermeneutical Elicitation of Requirements.

The Dasein triad, composed of the concepts “being-in-the-world”, “being-with-others”, and “being-to-death”, was suitably adapted to the triad of Hermeneutical Elicitation of Requirements, composed of the concepts “Situational Difference Identification”, “Situational Difference Examination” and “Hermeneutical Specification Requirement”. In this way, it was ensured that the essence, both ontic and ontological, of the meaning of software requirements is understood and interpreted correctly, enabling the exact and ideal conception of the software requirements to be developed.

The application of the Hermeneutical Elicitation of Requirements was presented as an example in a software development project for the “Game of Memory”. With this, it was possible to verify the practicality and simplicity of applying it, besides its independence with the methods, processes and tools adopted for the project.

Parallel works to the work presented in this article are related to the use of

semiotic computation, but the problems related to the Elicitation of Software Requirements are not emphasized by these studies. Therefore, it is opportune to develop a project that aims to improve the elicitation of software requirements, since the level of complexity of the software is increasing, as well as the domain of the application, and, for this reason, the difficulty of understanding them is increasing. The Hermeneutical Elicitation of Requirements is the technical approach that aims to help the requirements engineer better understand and interpret the application domain and thereby improve the conception of software requirements.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Project Smart (2014) The Standish Group Report—Chaos Report. <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
- [2] Chaos (1994, 1995, 2011, 2012, 2013, 2014, 2015) The Standish Group Report.
- [3] Sommerville, I. (2015) Software Engineering. 10th Edition, Pearson, London.
- [4] Pressman, R.S. and Maxim, B.R. (2016) Software Engineering: A Practitioner's Approach. 8th Edition, AMGH, Porto Alegre.
- [5] Northrop, L. (2006) Ultra-Large-Scale Systems: The Software Challenge of the Future. Software Engineering Institute, Pittsburg, PA.
- [6] International Conference on Software Engineering. <http://www.icse-conferences.org/>
- [7] Software Engineering Institute. <https://www.sei.cmu.edu/>
- [8] Medeiros, E. (2004) Desenvolvendo Software com UML 2.0: Definitivo. Pearson Makron Books, São Paulo.
- [9] Lawrence, S.P. (2004) Software Engineering: Theory and Practice. 2nd Edition, Prentice Hall, São Paulo.
- [10] Heidegger, M. (2005) Being and Time—Part I. 15th Edition, Vozes, Petrópolis, RJ.
- [11] Heidegger, M. (2005) Being and Time—Part II. 13rd Edition, Vozes, Petrópolis, RJ.
- [12] Stein, E. (2000) Coleção “Os Pensadores”: Heidegger—Vida e Obra. Editora Nova Cultural, São Paulo, SP.
- [13] Andersen, P.B. (1990) A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems (Cambridge Series on Human-Computer Interaction). Cambridge University Press, Cambridge, 416 p.
- [14] Object Management Group (OMG) (2017) Semantics of Business Vocabulary and Business Rules. Version 1.4. <http://www.omg.org/spec/SBVR/1.4/PDF>
- [15] Varalda, W. and Vega, Í.S. (2017) Hermeneutical Engineering of Requirements. *Journal of Computer and Communications*, 5, 7-16. <https://doi.org/10.4236/jcc.2017.52002>
- [16] Portocarrero, M.L. and da Silva, F. (2010) Hermenêutica Filosófica. Dissertation. Universidade de Coimbra, Coimbra.
- [17] Coelho, A. (2012) “Ser e Tempo”, de Heidegger (II): O Dasein (Ser-Aí).

<http://aquitemfilosofiasim.blogspot.com.br/2012/02/ser-e-tempo-de-heidegger-ii-o-dasein.html>

- [18] da Silva Roberto, L. (2009) Os modos de ser do “Dasein” a partir da analítica existencial heideggeriana. <http://pensamentoextemporaneo.com.br/?p=489>
- [19] Samaridi, I. (2011) O Ser-no-Mundo e suas possibilidades existenciais num contexto atual. Anais do Congresso de Fenomenologia da região Centro-Oeste. Caderno de Textos—IV Congresso de Fenomenologia da região Centro-Oeste—19-21 de Setembro de.
- [20] de Oliveira L.E.C. (2010) O Ser-Com como compartilhamento da verdade do Ser-Aí. Vol. 3, Revista Saberes, Natal, RN.
- [21] Web College (2017-2018) How to Learn More with the Memory Game? <https://www.colegioweb.com.br/educador/como-aprender-mais-com-o-jogo-da-memoria.html>