SEMAT Essence Game: A Fun Way to Learn about the SEMAT Initiative*

El juego de la Esencia de SEMAT: una Forma divertida de aprender sobre la iniciativa SEMAT

Carlos Mario Zapata Jaramillo [a]
Universidad Nacional de Colombia, Colombia
ORCID: https://orcid.org/0000-0002-0628-4097

Grissa Vianney Maturana Gonzáles
Universidad Nacional de Colombia, Colombia
ORCID: https://orcid.org/0000-0001-7719-2010

Johnathan Mauricio Calle Gallego
Universidad Nacional de Colombia, Colombia
ORCID: https://orcid.org/0000-0002-7297-1362

Abstract:

Objective: We aim to propose the SEMAT Essence game, a game-based simulation of the big picture related to the SEMAT Essence kernel. SEMAT (Software Engineering Method and Theory) is an initiative for refounding software engineering based on a solid theory, proven principles, and best practices. SEMAT has the Essence kernel of universal elements for any software engineering endeavor. The SEMAT Essence has been taught using workshops and some games, although such strategies only cover some topics related to the kernel.
Materials and methods: We use the SEMAT Essence kernel as a starting point, and we select a game as our game template. We then define the metaphor linked to our field of study and design the game by defining the equivalences to the game elements.
Results and discussion: We show the results of practicing the game with a set of students from four universities by summarizing the answers to a survey.
Conclusions: The results are promising because practitioners have reported some usage of the SEMAT Essence kernel elements after the game, and they are satisfied with the game's fun factor, realism, and current version.
Keywords: SEMAT, essence kernel, educational game, software engineering.

Resumen:

Objetivo: se propone el juego de la Esencia de SEMAT, una simulación basada en juegos de la vista panorámica relacionada con el núcleo de la Esencia de SEMAT. SEMAT (*Software Engineering Method and Theory*) es una iniciativa para volver a fundar la ingeniería de software con base en una teoría sólida, principios probados y buenas prácticas. SEMAT tiene un núcleo (denominado "Esencia") de elementos universales a cualquier esfuerzo de ingeniería de software. La Esencia de SEMAT se viene enseñando por medio de talleres y algunos juegos, pero estas estrategias sólo cubren algunos temas relacionados con el núcleo. Materiales y métodos: se usa el núcleo de la Esencia de SEMAT como punto de partida y se selecciona un juego que sirva de plantilla para el juego propuesto. Luego, se define la metáfora ligada con el campo de estudio y se diseña el juego mediante la definición de equivalencias con los elementos del juego. Resultados y discusión: también, se muestran los resultados de la práctica del juego con un conjunto de estudiantes de cuatro universidades mediante el resumen de las respuestas a una encuesta. Conclusiones: los resultados son prometedores, ya que los practicantes del juego reportan algún uso de los elementos del núcleo de la Esencia de SEMAT después del juego y se muestran satisfechos con el factor de diversión, el realismo y la versión actual del juego.
Palabras clave: SEMAT, núcleo de la esencia, juego educativo, ingeniería de software.

Introduction

Some problems have led to an initiative for refounding software engineering: the prevalence of "fashioned" methods, the gap between academia and industry, and the lack of theoretical basis. SEMAT (Software

Author notes

[a] Correspondence author: cmzapata@unal.edu.co

Engineering Method and Theory) is intended to solve such problems [1]. Software engineering theory has been developed by SEMAT in the Essence kernel with elements belonging to any software engineering endeavor. Alphas (the things we always work with), activity spaces (the things we always do), and competencies (the skills we need) are common to software endeavors regardless of the method we use [1].

The Essence kernel is intended to bridge the gap between academia and industry, because we can either teach software engineering students the foundations of any software engineering endeavor or train software engineering practitioners with the same ideas. Some courses have been designed to teach the Essence kernel using workshops [1, 2] and some games [2, 3, 4, 5, 6, 7, 8, 9, 10]. However, neither workshops nor games are suitable for teaching the generality of the Essence kernel.

As a way to solve the aforementioned problems, in this paper, we propose the SEMAT Essence game, an adaptation of the game Life® [11]. In the SEMAT game, we adopt the entire set of elements of the Essence kernel in a spider-web-chart board for simulating the health and progress of a software engineering endeavor. We also play the game with students from four universities with different backgrounds, and we report the findings.

The remainder of this paper is organized as follows: in the methods and materials section, we present a theoretical framework, a state-of-the-art review, and our game proposal; in the results section, we discuss the findings; and finally, in the conclusions section, we summarize the findings and discuss future work.

Materials and methods.

SEMAT Essence kernel

SEMAT is an initiative for refounding software engineering based on a solid theory, proven principles and best practices [1]. The SEMAT Essence [12] is currently an Object Management Group (OMG) standard comprising a kernel of elements present in any software engineering endeavor and a visual language for representing them. Some of these elements and their relationships are depicted in Figure 1. Definitions and additional information of such elements are presented in Table 1.
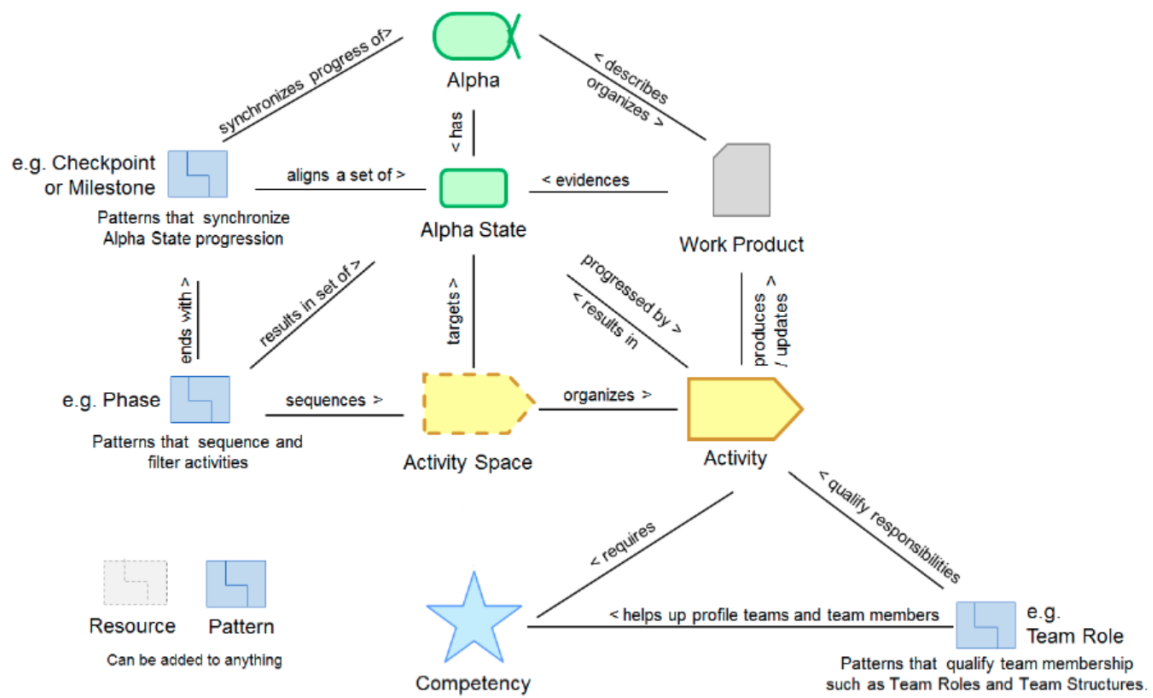
FIGURE 1.
Main elements of the SEMAT Essence kernel and their relationships
Source: OMG, 2018. [12]

TABLE 1.
Elements of the SEMAT Essence kernel

| Element	Symbol	Additional Information
Area of concern	Green (customer), yellow (solution), or blue (endeavor)	Elements can be divided into a collection of main areas of concern for paying special attention to. All elements fall into at most one of these.
Alpha		An essential element that is relevant to assess the progress and health of the endeavor—*i.e., opportunity* and *stakeholders* in *customer*; *requirements* and *software system* in *solution*; and *work, way of working*, and *team* in *endeavor*.
Alpha state		The progress and health of an alpha represented in a checklist. Alphas have limited states, *e.g.,* the o*pportunity* alpha has *identified, solution needed, value established, viable, addressed,* and *benefit accrued*.
Activity space		A placeholder for something to be done in the software engineering endeavor. 15 activity spaces are defined in the standard,
Activity		Define one or more kinds of work items and give guidance how to perform these. The standard has no predefined activities.
Competency		Encompasses the abilities, capabilities, attainments, knowledge, and skills necessary to do a certain kind of work. Six competencies are defined in the standard: *stakeholder representation, analysis, development, testing, leadership,* and *management*.
Work		An artifact of value and relevance for a software engineering endeavor. The standard has no predefined work products.
Product		
Pattern		A description of a structure in a practice. The standard has no predefined patterns, but three stereotypes are commonly used: *<role>, <phase>,* and *<milestone>*.
Practice		A repeatable approach to doing something with a specific objective in mind.
Method		The composition of a kernel and a set of practices to fulfill a specific purpose.

Source: Authors based on OMG, 2018 [12].

State-of-the-art review: SEMAT Essence kernel teaching with workshops and games

Alpha cards can be used in workshops as walkthroughs and pokers [1]. In a walkthrough, all the alpha cards are exposed on a table ordered by alpha in rows and state in columns, while the team decides which state each alpha is. In poker, each team member has a set of alpha cards and decides on the state of every alpha by showing the card he or she believes best represents the state of the alpha. Workshops are redefined in the shape of poker and chasing the state [10]. Jacobson et al. [10] propose two additional workshops with alpha cards: objective go and checkpoint construction.

Zapata and Jacobson [2] designed a first course in software engineering methods and theory. Workshops are the usual way to teach the SEMAT Essence language, and they also include the following games: the

software system alpha game and the requirements alpha game; SemCards and the SEMAT board-crossing game for learning terminology; and answering some questions related to the kernel elements.

MetricC [3] is a card game for practicing the way to use completion criteria to finish the work in activity spaces inside a practice, including some of the main elements of the SEMAT Essence kernel, such as practices, activity spaces, activities, alpha state cards, and patterns.

Ng [4] proposes a mix of workshop and card games for understanding practices in terms of alpha states in a similar way to the walkthrough [1], and he simulates some tasks. He uses the state of the alphas to guide the game.

Puzzler [5, 8] is a trivia-based video game for memorizing the main elements of the SEMAT Essence kernel, including alphas, activity spaces, and competencies, for remembering the names, relationships, and drawings related to such elements. Other trivia games, such as Kahoot!, and some workshop activities are included in this proposal [5, 8].

The SEMAT game [6] is a board game based on the representation of a PMBOK (Product Management Body of Knowledge; [13]), including alphas, states, activities, work products, and competencies. Alpha states are reached when work products are collected on the board, and the main goal of the game is acquiring all the work products for completing the practice.

Alphaspot [7] is a virtual game with activities for learning the names of the alphas and their states with 41 levels.

Kemmel et al. [9] developed and conducted a collaborative board game for teaching the meaning of the alphas in the context of a simulated project with some other elements, such as methods, tasks, and skills. The authors use nonstandard names for some elements.

We summarize the state of the art related to the SEMAT Essence kernel games and workshops in Table 2. For each approach, we present the SEMAT Essence kernel elements, including the classification and the purpose.

TABLE 2.
Summary of the state-of-the-art review

Approach (Authors)	SEMAT Essence kernel elements included	Classification	Purpose
Poker [1, 10]	Alphas, states, areas of concern	Workshop	Simulation
Walkthrough/Chase the state [1, 10]	Alphas, states, areas of concern	Workshop	Simulation
Workshops [2]	Alphas, states, areas of concern	Workshop	Simulation
The software system alpha [2]	Alphas, states, areas of concern	Game	Simulation
The requirements alpha [2]	Alphas, states, areas of concern	Game	Simulation
SemCards [2]	Alphas, activity spaces, competencies, areas of concern	Game	Concept memorization
SEMAT board-crossing [2]	Not established	Trivia Game	Concept memorization
MetricC [3]	Practices, alphas, states, activities, activity spaces, patterns	Game	Simulation
Puzzler and Kahoot! [5, 8]	Alphas, activity spaces, competencies	Trivia Game	Concept memorization
SEMAT game [6]	Alphas, states, activities, work products, competencies	Game	Simulation
Alphaspot [7]	Alphas, states	Video game	Concept memorization
Board game [9]	Alphas, states	Collaborative game	Simulation
Objective go [10]	Alphas, states, areas of concern	Workshop	Simulation
Checkpoint Construction [10]	Alphas, states, areas of concern	Workshop	Simulation

Source: Authors' own creation.

We infer that games are used for simulating some of the elements of the SEMAT Essence kernel with two purposes: simulating the actual environment of a software engineering endeavor and memorizing the concepts linked to the SEMAT initiative. None of the summarized approaches is intended to simulate the big picture related to the SEMAT Essence kernel using most of the elements depicted and defined in Figure 1 and Table 1.

The SEMAT Essence game

"The essence of metaphor is understanding and experiencing one kind of thing or experience in terms of another" [14]. We usually conceive games as metaphorical experiences, since we use some elements—*e.g.,*

boards, dice, cards, tokens, etc.—and give them some sense in certain domains. Our approach to the big picture related to the SEMAT Essence kernel starts by comparing the main idea of the standard with any other domain for adapting the game's metaphor. For example, the software system—the main result of the standard —has a lifecycle similar to human beings. Additionally, human beings have endeavors similar to software systems. For such reasons, we look for a game representing a human life. Fortunately, we have Life® [11].

Life® "simulates a person's travels through a fictional 'life' from birth to retirement, encountering jobs, marriage, kids, and other assorted life happenings along the way" [11]. We can adapt the metaphor since the software system development lifecycle also has a starting point—the *architecture selected* state—and a final point—the *retired* state. Our proposed game aims to encourage the players to practice the way the SEMAT Essence kernel elements are used in a simulated environment where alphas progress from one initial state to a final state. Additionally, the game is intended to strengthen the interpretation of the SEMAT Essence big picture by comprehending the elements involved in the kernel, specifically *patterns, methods, areas of concern, alphas, states, activities, work products, competencies, practices*, and simulated situations related to the interaction among them. Our game can be played by any person interested in learning about the SEMAT Essence standard, regardless of his or her field of knowledge.

Life® has a car with stick people (blue or pink according to the genre) that represent people. We select the symbol of the *pattern* (see Table 1) with the *<role>* stereotype for representing team members in our game. Each team member has a color according to the related *area of concern*. Additionally, each team member is numbered *one* or *two* to identify which team member is advancing during the game.

The Life® board represents the path a person should follow to advance through life; our equivalent board is presented as a hexagon representing the *method* element in the SEMAT Essence standard (see Table 1), as shown in Figure 2. The Life® board has two paths related to career or college. In the same way, our board has been divided into six equal parts, each representing an alpha—i.e., *stakeholder, opportunity, work, team, requirements, and software system.* Such alphas are grouped into the three *areas of concerns* differentiated by color. *Ways of working* were omitted from the board since they have a hexagon. The board has six different levels (*states*) represented by concentric hexagons. We omit the last *state* of each alpha—except the *team* alpha —since it has five states.
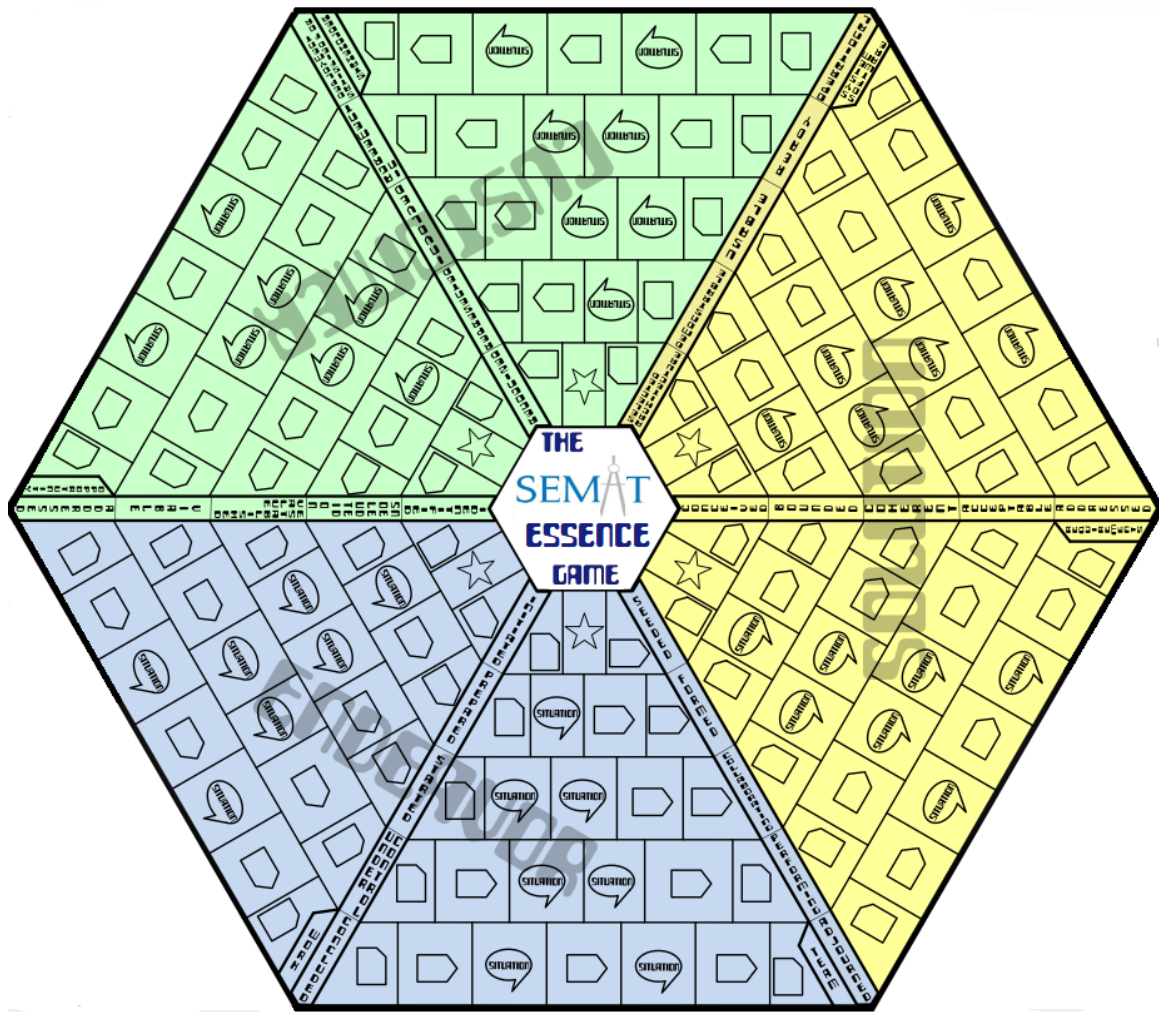
FIGURE 2.
The SEMAT Essence game board
Source: Authors' own creation.

Life® spots represent the paths to advance in life. Following the same metaphor, our *alpha states* (levels in the hexagon) are divided into spots with some elements needed to achieve any *practice*. Figure 3 illustrates the elements to be attained by the players during their project life cycle. Three SEMAT Essence kernel elements are included: *work product, activity*, and *competency. Competencies* can only be obtained in the first state, while the other elements can be developed at any state. We add a nonstandard element called the situation, as we explain later. Some spots in Life® are detailed by using cards. We use the same metaphor since we detail the elements of our game using definition cards. "A definition card is visualized as a solid-outline rectangle in landscape format containing a mix of symbols and textual syntax related to the element (...) The following is a basic anatomy although variations are allowed" (see Figure 4) [12].
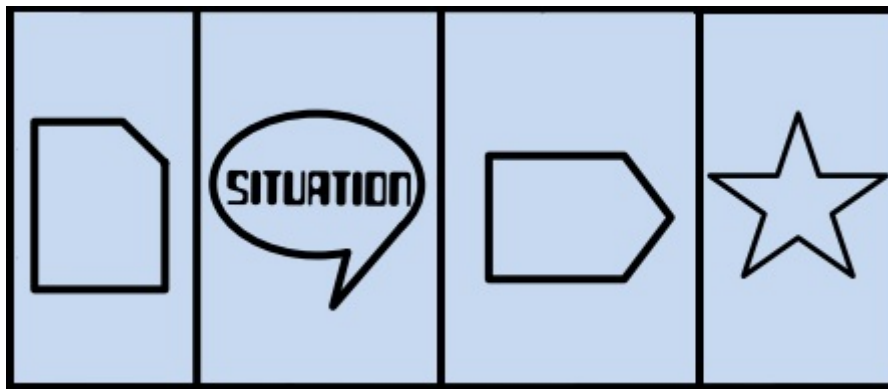
FIGURE 3.
SEMAT Essence elements per state in the SEMAT Essence game
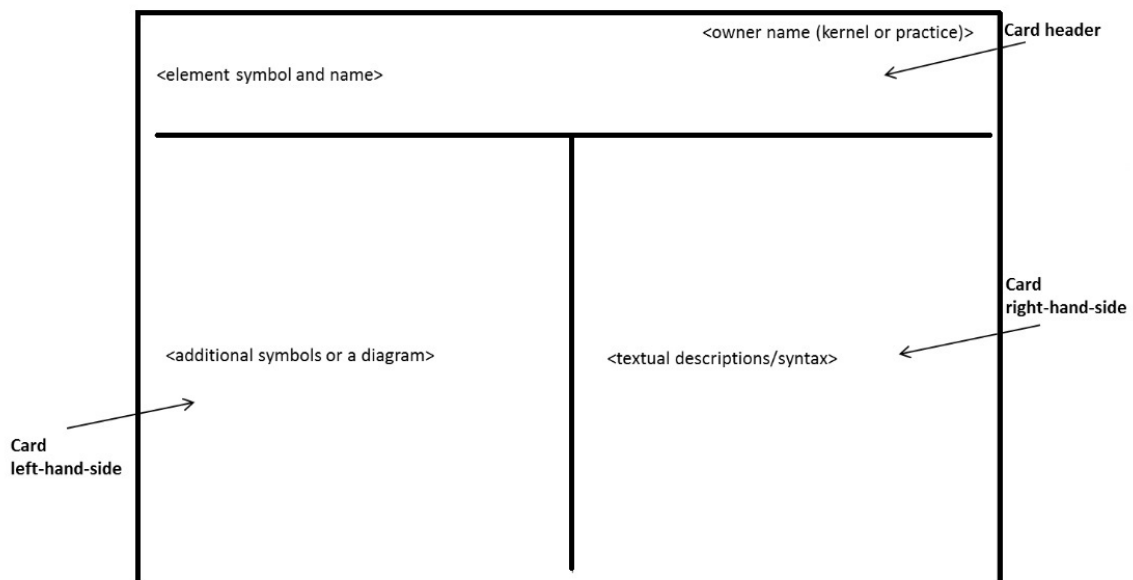Source: Authors' own creation.

FIGURE 4.
Basic definition card anatomy to visualize an element
Source: OMG, 2018. [12]

Competency cards contain the name of the *competency*, a definition, and an explanation of how it helps a team in real life. Finally, all the cards have a different color according to their *area of concern*. We define the following *competency cards: stakeholder representation, business mastering, and communication* (the latter are outside the SEMAT Essence standard, but we follow a similar structure, since we need three competencies per *area of concern*), belonging to *customers*; *analysis, development*, and *testing* (depicted in Figure 5) from *solution*; and *leadership*, *management*, and *project orientation* (the latter outside the SEMAT Essence kernel) from *endeavor*. Note that *competencies* directly belong to the kernel. The left-hand side indicates the five levels of competency defined in the standard, and the right-hand side includes the definitions provided in the standard; both levels and definitions are just informative inside the game. This card is obtained by the teams when they reach the symbol of the *competency* in the first *state* of an *alpha*.
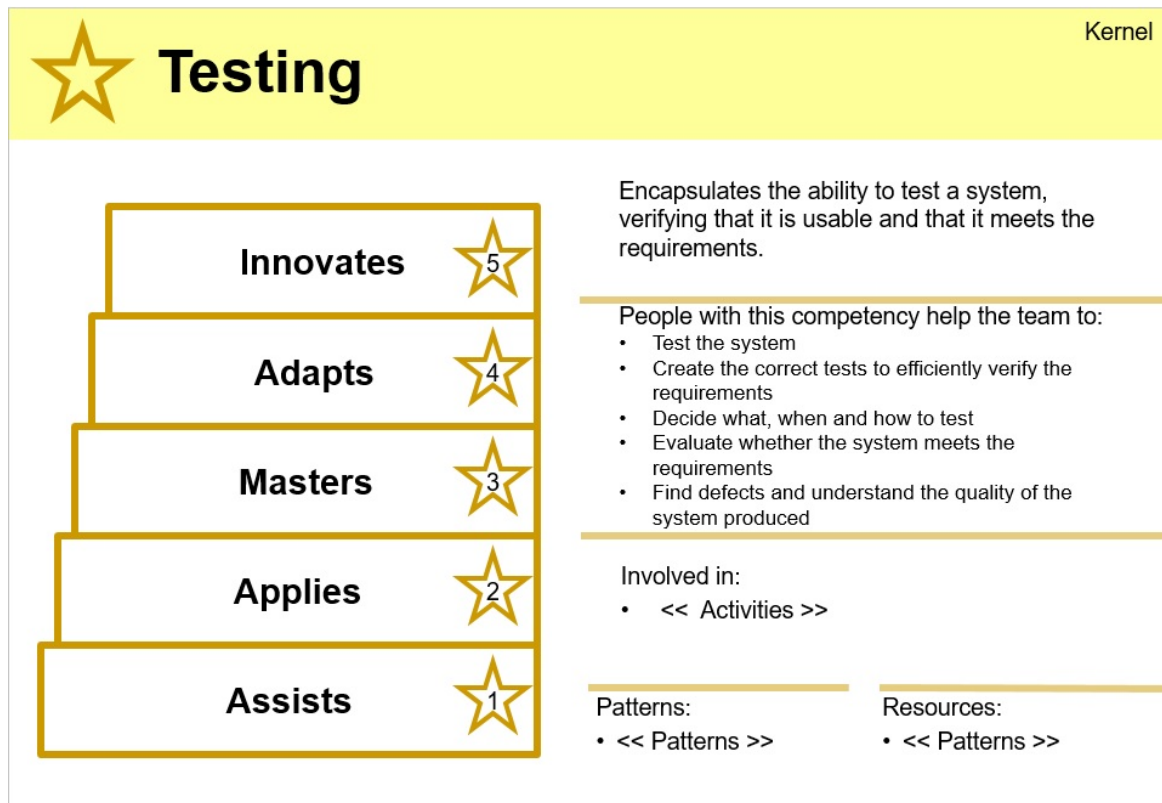
FIGURE 5.
Graphic representation of the competency card called *testing*
Source: Authors' own creation.

Activity cards include the name of the *activity* and the name of the *practice* in the header. The left-hand side indicates the *alpha* related to the *activity*, the *competency* needed to perform the *activity*, the *state* of the *alpha* after the *activity* is performed, and the resulting *work product*. The right-hand side includes a brief description of the *activity*. We define one activity per *alpha* per *state*. Practices are named after actual activities performed in each *area of concern*. We also illustrate the *activity* called *verify consistency* in Figure 6. This card is obtained by the teams when they reach the symbol of the *activity* in any *state*.
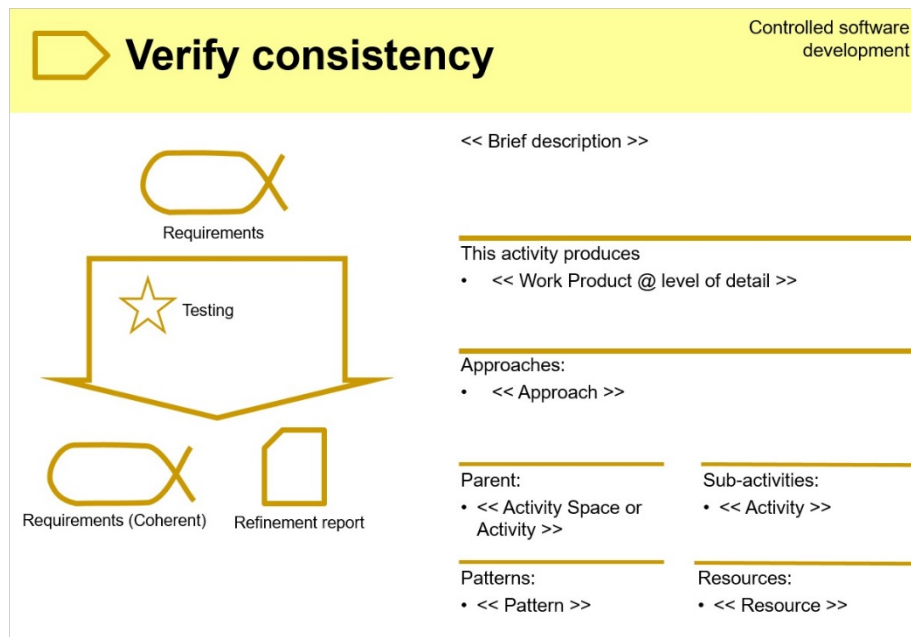
FIGURE 6.
Graphic representation of the activity card called *verify consistency*
Source: Authors' own creation.

Work product cards contain a header with the name and the related practice, a left-hand side with the *levels of detail* [12], and a right-hand side with the basic information about the *work product*. We also define a list of realistic *work products*, as illustrated with the *refinement report* in Figure 7. This card is obtained by the teams when they reach the symbol of the *work product* in any *state*.
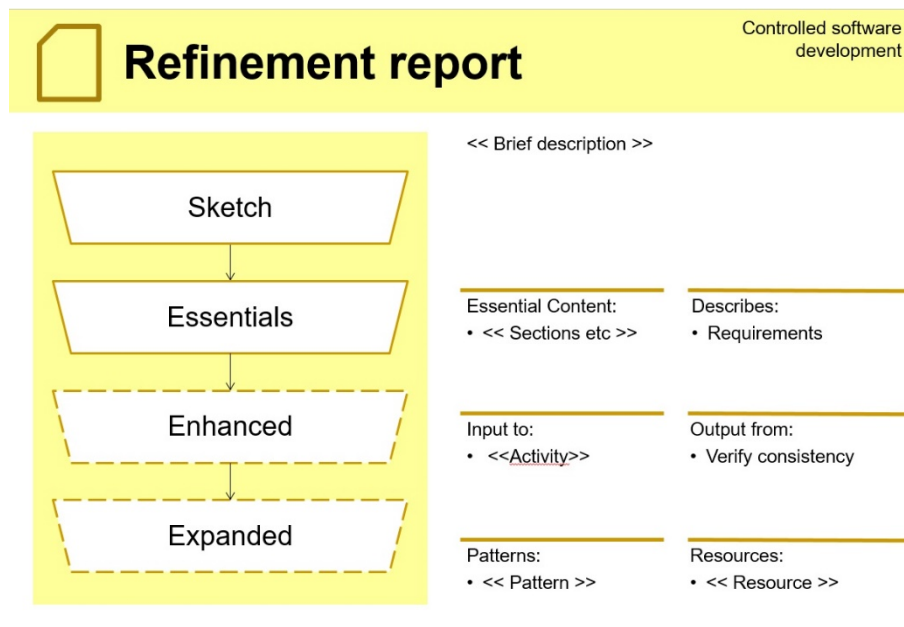

FIGURE 7.
Graphic representation of the work product card called the *refinement report*
Source: Authors' own creation.

Competencies, activities, and *work products* are integrated into a new kind of card for representing *practices*, containing a header with the name of the *practice* related to the kernel; a left-hand side with a graphical

representation by following the specification format of each symbol and the guidelines defined in the standard [12]; and a right-hand side with the *state* to be reached by completing the practice. This graphical representation includes the elements the teams must perform (*activities*), elaborate (*work products*), and obtain (*competencies*) to fulfill a practice. We define a set of realistic *practices* following the guidelines recommended for the name [15] and consistently related to the *areas of concern*. The entire list of practices is ordered by *area of concern*, including the *activities, work products*, and *competencies* are summarized in Tables 3, 4, and 5. We illustrate the *practice* card in Figure 8. This card is obtained by the teams when they obtain all of the elements needed to completely represent the *practice*, including *competencies*, *activities*, and *work products*.

TABLE 3.
Practices defined for the *customer* area of concern with relationships

Practice	Activity	Competency	Work product	Alpha	State
Market share increasing	Identify stakeholders	Stakeholder representation	Stakeholder list	Stakeholders	Recognized
	Survey the market	Business mastering	Mission statement	Opportunity	Identified
Collaborative stakeholder agreement	Select stakeholder representative	Stakeholder representation	Cause-and-effect diagram	Stakeholders	Represented
	Model the problem	Communication	Collaboration approach	Opportunity	Solution needed
Critical success factor definition	Develop stakeholder commitment	Communication	Stakeholder engagement assessment matrix	Stakeholders	Involved
	Analyze opportunity	Stakeholder representation	Critical success factors	Opportunity	Value established
Structured feasibility study	Sign the contract	Business mastering	Final contract	Stakeholders	In agreement
	Evaluate feasibility	Business mastering	Feasibility study	Opportunity	Viable
Complementary closure review	Close the endeavor	Communication	Closing protocol	Stakeholders	Satisfied in use
	Summarize expenditures	Business mastering	Financial Report	Opportunity	Addressed

Source: Authors' own creation.

TABLE 4.
Practices defined for the *solution* area of concern with relationships

Practice	Activity	Competency	Work product	Alpha	State
Progressive requirements evaluation	Apply standards	Analysis	SRS cards	Requirements	Conceived
	Define architecture	Development	Initial architecture	Software system	Architecture selected
Detailed scope definition	Translate requirements	Analysis	Conceptual schema	Requirements	Bounded
	Build prototype	Development	Alpha prototype	Software System	Demonstrable
Controlled software development	Verify consistency	Testing	Refinement report	Requirements	Coherent
	Build the software system	Development	Source code	Software system	Usable
Practical software validation	Animate the specification	Development	Usage scenarios	Requirements	Acceptable
	Execute test plan	Development and testing	Acceptance tests	Software system	Ready
Purposeful software delivery	Close the contract	Analysis	Final agreement	Requirements	Addressed
	Release the software system	Development	Final release	Software system	Operational

Source: Authors' own creation.

TABLE 5.
Practices defined for the *endeavor* area of concern with relationships

| Practice	Activity	Competency	Work product	Alpha	State
Systematic responsibility assignment	Analyze activities	Management	Work breakdown structure	Work	Initiated
	Meet the team	Project orientation	Responsibility matrix	Team	Seeded
Detailed work planning	Define constraints	Management	Governance policies	Work	Prepared
	Assign resources	Leadership	Work schedule	Team	Formed
Iterative progress evaluation	Write project chart	Project orientation	Project chart	Work	Started
	Make report	Management	Activity report	Team	Collaborating
Task internal auditing	Measure the work	Project orientation	Metric report	Work	Under control
	Assess commitment	Management	Commitment review	Team	Performing
Project final review	Close the endeavor	Management	Closure report	Work	Concluded
	Reassign team	Leadership	Project delivery	Team	Adjourned

Source: Authors' own creation.
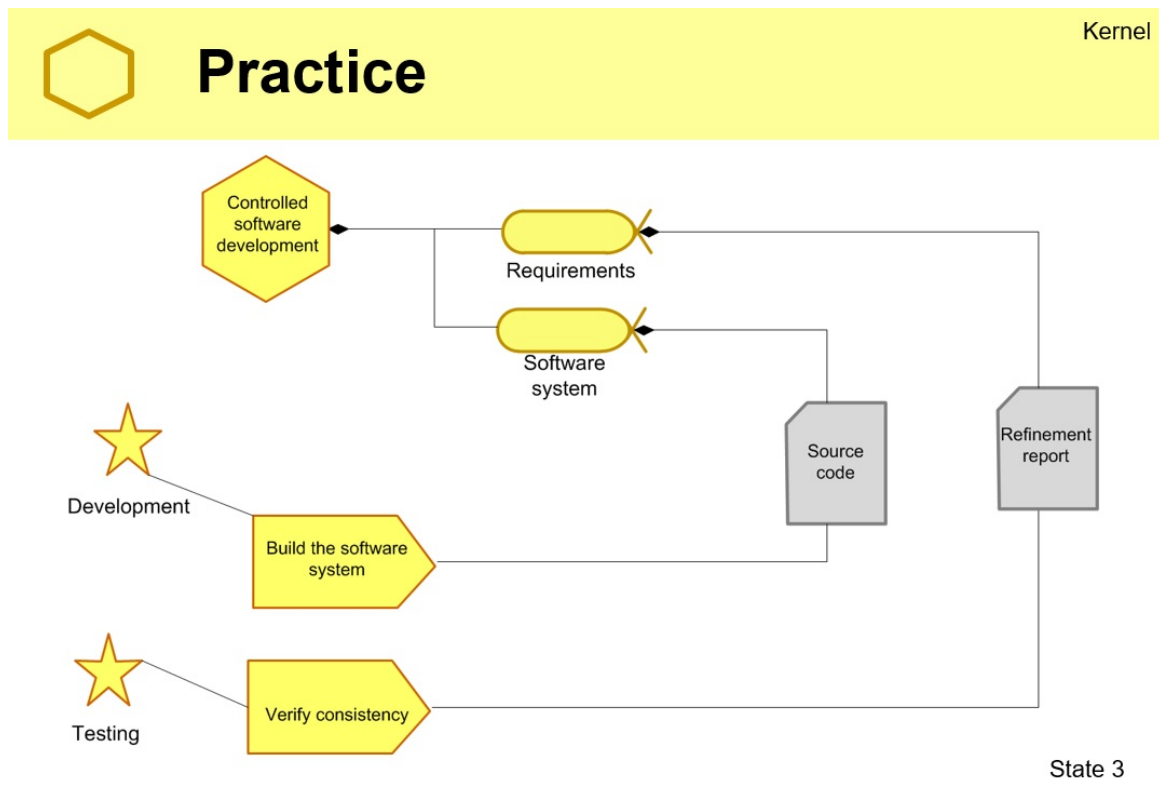
FIGURE 8.
Graphic representation of the practice card called *controlled software development*
Source: Authors' own creation.

Situation cards represent any situation in the software engineering endeavor at some time. *The situation* card has the following structure: a header, with the title *situation* related to the kernel; a left-hand side with two emoticons (positive and negative) linked to two kernel elements; and a right-hand side with the description of three situations (good, neutral, and bad). We illustrate a *situation* card in Figure 9.
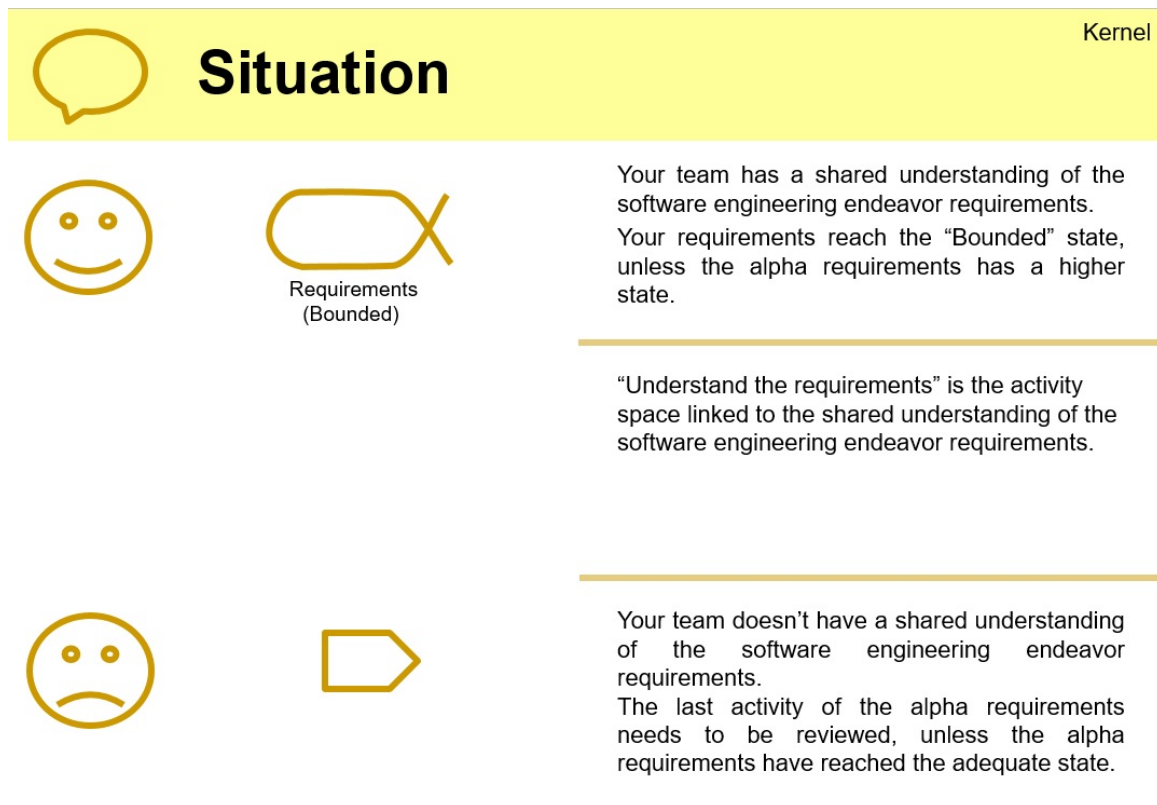
FIGURE 9.
Graphic representation of a *situation* card
Source: Authors' own creation.

The game starts with six team members located in the first *state* of an alpha at the corresponding *area of concern*. Team members try to obtain *competencies*, execute *activities*, and develop *work products* related to the *practice* with the first *state* of the *alphas* inside their *area of concern* (two team members are in charge of *customers*, another two are in charge of *solutions*, and the final two are in charge of *endeavors*). Team members move on the board by rolling one die and obtaining elements for completing the practices; initially, team members move clockwise in the level until they reach the boundary of the *area of concern*—after that, they start to move counterclockwise until they reach the other boundary. Be advised that (i) we avoid *situations* in state 1 and (ii) *competencies* can be obtained only in state 1, but we can keep it until the end of the game. Once all the elements are complete, they obtain a state check (see Figure 10) to cover such a *state*, and they can move to the next level (*state*) to start obtaining *work products* and *activities*. Once the team members reach a *situation*, they receive a *situation* card, and they have to roll the die once again to determine the kind of *situation* they face: one/two lead to the bad situation; three/four lead to the neutral situation; and five/six lead to the good situation. Team members act according to the sentence by giving/removing elements.

FIGURE 10.
Graphic representation of a state check
Source: Authors' own creation.

Team members win the game by advancing the most in the *alpha states* of their *area of concern*. The board itself (see Figure 2) can serve as a spider web chart for determining the winners. A possible final state of the game is shown in Figure 11, with the advance (area inside the red lines) larger for the *customer*, followed by the *solution*, and finally the *endeavor*. In the case of a tie, we evaluate the elements belonging to the incomplete *practice* card.
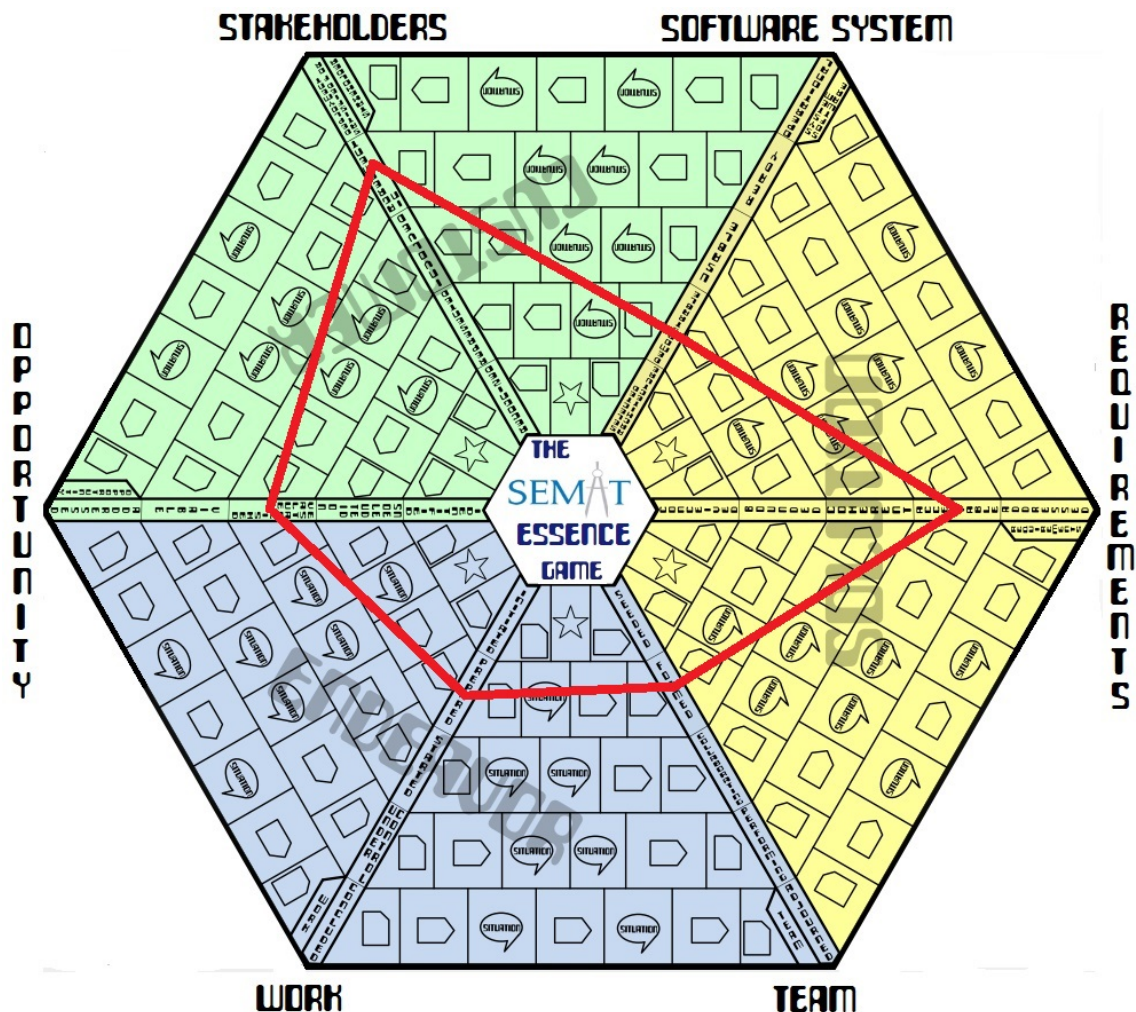

FIGURE 11.
Possible final state of the SEMAT Essence game
Source: Authors' own creation.

Results

We practiced our game with 33 software engineering students from four different universities. The sample is heterogeneous in the year of studies. We applied the following survey to evaluate the experience: (i) How close to reality is the game? (ii) How fun is the game? (iii) How simple is the game to play? (iv) What did you learn from the game? (v) What is the best strategy to win the game? (vi) What would you do to change the game.

Related to the first question (Figure 12), all the practitioners believe the game is close to reality in a certain way, since 58% of them answered *close/very close* and the rest express some feature of the game that is close to reality. Practitioners face *situations* related to an actual software engineering endeavor most of the time during the game. Additionally, they recognize elements of the kernel, such as requirements, software systems, teams, and stakeholders.
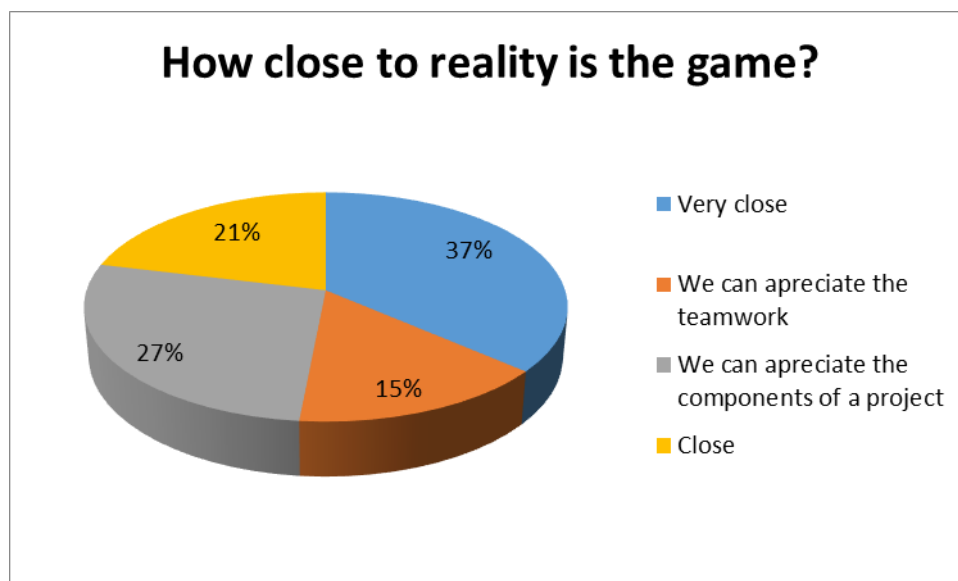

FIGURE 12.
Answers to the first question of the survey
Source: Authors' own creation.

Related to the second question (Figure 13), the results are impressive since all the practitioners recognize a level of fun ranging from *good* to *excellent*. Emotions flow in the game from the beginning, and practitioners keep attention throughout the game. A game close to reality that is fun to play is a wonderful resource to use in the classroom.
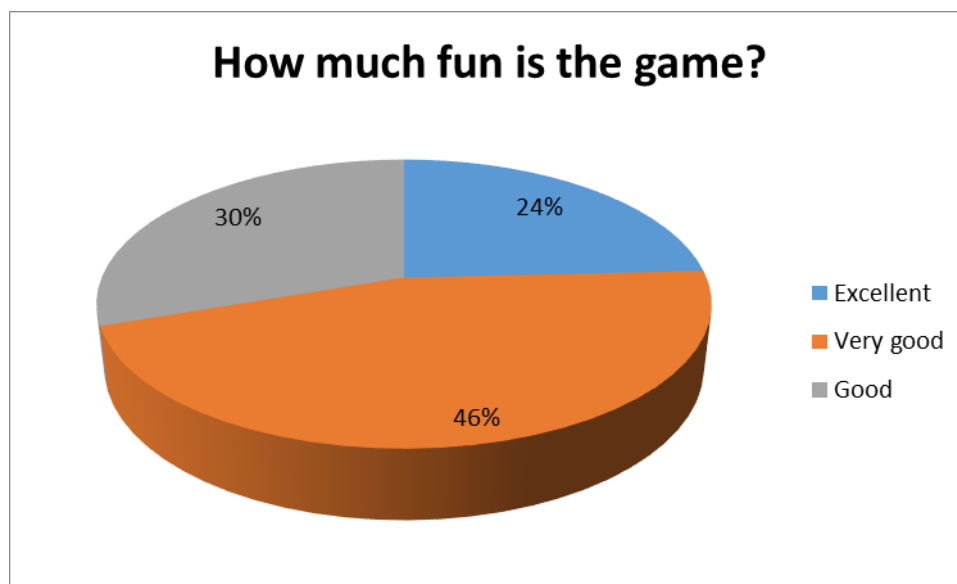
FIGURE 13.
Answers to the second question of the survey
Source: Authors' own creation.

Related to the third question (Figure 14), 72% of the practitioners recognize some complexity in the game. This result is expected due to the size of the board, the number of cards we use, the number of elements from the kernel to be represented, and the time needed to set up the game.
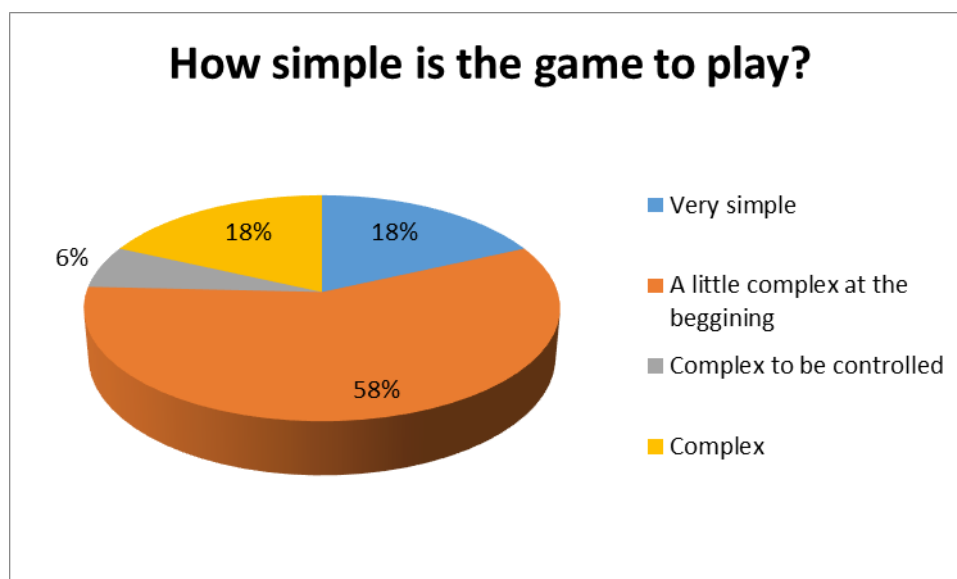

FIGURE 14.
Answers to the third question of the survey
Source: Authors' own creation.

Related to the fourth question (Figure 15), we use the metaphor related to the kernel during the game to reinforce learning about the elements of the standard. For example, we say "advance in the states of your alphas" instead of "throw the die." This is why 37% of the practitioners say they learn about the structure and components of the standard. Other practitioners (36%) use terms such as work, team, activity, and state in their answers, revealing some knowledge about the elements.
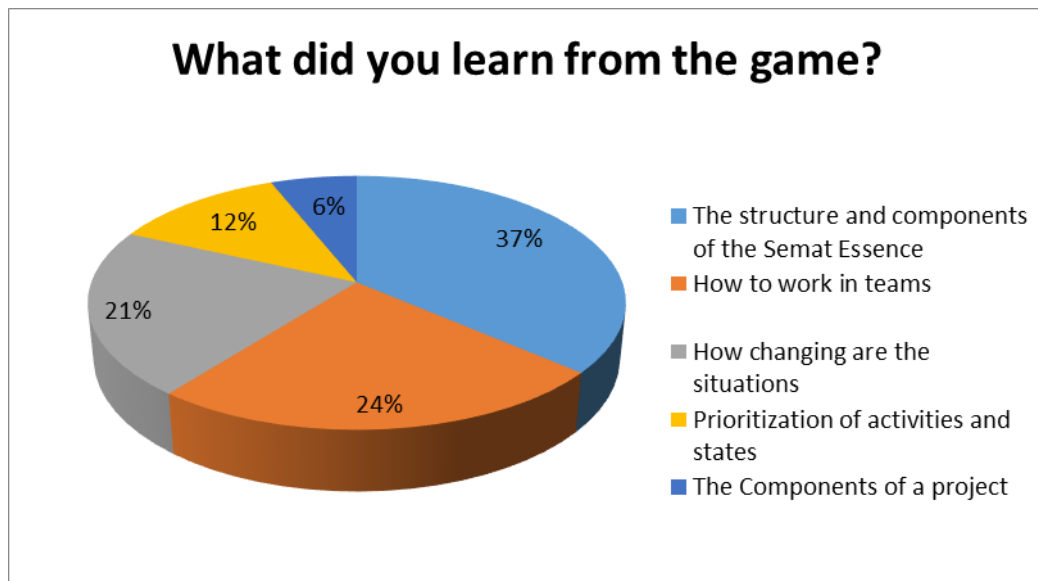
FIGURE 15.
Answers to the fourth question of the survey
Source: Authors' own creation.

Twenty-one percent of the practitioners talked about situations. Even though *situations* are nonstandard elements, we use them to provide realism in the game. The situations a team faces in an actual scenario are learning experiences of the game.

Related to the fifth question (Figure 16), most practitioners (49%) talk about adequate teamwork. Even though our game is a competition among teams with responsibilities related to isolated areas of concern, our board resembles the same method with practices to be followed in the three areas of concern. Accordingly, teams compete by applying the same method in the same endeavor. Forty-eight percent of the participants talked about the rules of the game and the importance of awareness of alphas and states in a coordinated way. Three percent of the answers are related to luck as a strategy; of course, we need to introduce elements such as dice in the mechanics of the game, but we are satisfied that few participants believe luck is important to win the game.
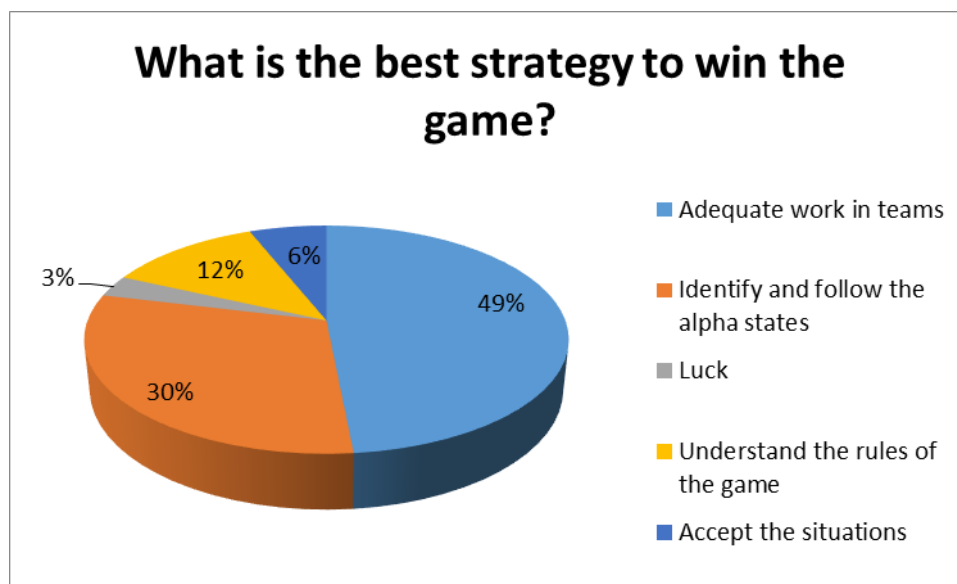
FIGURE 16.
Answers to the fifth question of the survey
Source: Authors' own creation.

Related to the sixth question (Figure 17), 37% of the participants like the game as it is. Fifty-four percent of the practitioners talk about the game mechanics and its complexity, i.e., the graphical control related to the use of the cards, the number of initial instructions, and the structure of the cards. Complexity arises from the desire to design a complete game with all the possible elements from the standard; we need to improve this issue, but keeping in mind that we are teaching the usage of most of the standard elements in a simulated environment. Nine percent of the practitioners suggested an increase in the level of competition. Our game is collaborative, but we can foster competition by playing a different version, with a team working in all of the areas of concern or by playing against each other.
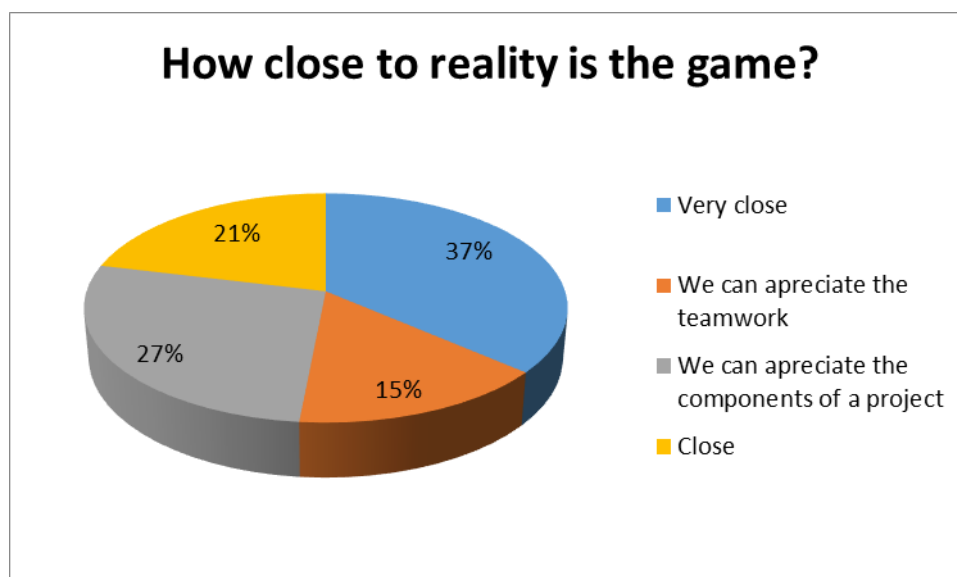

FIGURE 17.
Answers to the sixth question of the survey
Source: Authors' own creation.

Conclusions

Games and workshops are teaching strategies used for remembering and simulating the main elements of the SEMAT Essence standard. Such strategies are commonly related to a subset of elements—usually three to five—of the SEMAT Essence standard, leaving aside the big picture of the software engineering endeavors represented by using such elements. For this reason, this paper proposed the SEMAT Essence game based on Life®. We use a metaphor based on a human life to adapt the main elements of the game and finding some equivalences with software engineering endeavors. The players represent team members with *<role>* -stereotyped *patterns*; the hexagon-shaped board resembles *methods*; triangles represent the *alphas* grouped into *areas of concern*; concentric hexagons represent *alpha states*; and finally, *practices, activities, competencies*, and *work products* are represented by *standard cards*. We validated our game by playing with 33 students from four universities. They answered a survey related to the game's realism, fun factor, complexity, lessons learned, and areas of improvement. The results are very promising in almost all the topics, except in the complexity of the game, since we tried to incorporate most elements to represent the big picture of the standard. Some lines of future work can be defined from our game:

- Defining one version of the game for promoting competition. Even though the game has some competition, the current version is related to the same team trying to advance in all of the alpha states of the same software engineering endeavor.
- Practicing the game in an industrial environment. It is a difficult task since the current version lasts two hours or more, and managers may be unwilling to allow employees to devote this much time to a game.
- Creating a virtual board in a collaborative tool to spread the game around the world.
- Studying emotions during game practice. We are very positive about how good the game is for practitioners; thus, if we can deduce the way emotions flow during the game, we can replicate this experience in other games by incorporating some features in game design.

References

[1] I. Jacobson, P.-W. Ng, P. McMahon, I. Spence, and S. Lidman, *The Essence of Software Engineering. Applying the SEMAT Kernel*. Crawfordsville: Addison-Wesley Professional, 2013.

[2] C. M. Zapata and I. Jacobson, "A first course in software engineering methods and theory," Dyna, vol. 81, no.183, 2014, pp. 231–241.

[3] C. M. Zapata, G. Maturana, and L. Castro, "Tutorial sobre la iniciativa SEMAT y el juego MetricC," 8th Congreso Colombiano de Computación, Armenia, 2013.

[4] P.-W. Ng, "Software process improvement and gaming using Essence: an industrial experience," Journal of Industrial and Intelligent Information, vol. 2, no. 1, 2014, pp. 45–50.

[5] J. Pieper, "Discovering the essence of software engineering: an integrated game-based approach based on the SEMAT Essence specification," IEEE Global Engineering Education Conference (EDUCON), Tallinn, 2015, pp. 939–947.

[6] C. M. Zapata, M. D. Rojas, R. Arango, and L. Jiménez, "SEMAT game: applying a project management practice," Developments in Business Simulation and Experiential Learning, no. 42, 2015, pp. 133–143.

[7] E. Jiménez-Hernández, H. Oktaba, F. Díaz-Barriga, M. Piattini, A. Revillagigedo-Tulais, and S. Flores-Zarco, "Methodology to construct educational video games in software engineering," 4th International Conference in Software Engineering Research and Innovation (CONISOFT), Puebla, 2016, pp. 110–114.

[8] J. Pieper, O. Lueth, M. Goedicke, and P. Forbrig, "A case study of software engineering methods education supported by digital game-based learning: applying the SEMAT Essence kernel in games and course projects," IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 1689–1699.

[9] K. Kemmel, J. Risku, A. Evensen, P. Abrahamsson, A. Dahl, L. Grytten, A. Jedryszek, P. Rostrup, P., and A. Nguyen-Duc, "Gamifying the escape from the engineering method prison," IEEE International Conference on Engineering, Technology and Innovatiion (ICE/ITMC), Stuttgart, 2018.

[10] I. Jacobson, H. Lawson, P.-W. Ng, P. McMahon, and M. Goedicke, "The essentials of modern software engineering: free the practices from the method prisons!," Milton Keynes: ACM Books, 2019.

[11] B. Rome and Ch. Hussey, "Games' most wanted ™: the top 10 book of players, pawns, and power-ups," Quicksilver Drive: Potomac Books, 2013.

[12] Object Management Group, "Essence—Kernel and Language for Software Engineering Methods, version 1.2," 2018, Retrieved from https://www.omg.org/spec/Essence/About-Essence/.

[13] Project Management Institute, "A guide to the Project Management Body of Knowledge (PMBOK Guide)—Fifth Edition," Project Management Institute, Inc: Newtown Square, 2013.

[14] G. Lakoff and M. Johnson, "Conceptual metaphor in everyday language," Journal of Philosophy, vol. 77, no. 8, 1980, pp. 453–486.

[15] A. Barón, Modelo para la definición unificada de la práctica como constructo teórico en ingeniería de software, Ph.D. Thesis, Departamento de Ciencias de la Computación y de la Decisión, Universidad Nacional de Colombia, Medellín, 2019.

Notes

* Research article.