DOI: 10.36558/rsc.v11i2.7215

# InnoTrace: Tracing Requirements in Innovation and Software Processes

Breno Leonardo G. de M. Araújo
Federal institute of Education Science and Technology of Pará
brenolgma@gmail.com

Ricardo A. C. de Souza
Federal Rural University of Pernambuco - UFRPE
ricardo.souza@ufrpe.br

## ABSTRACT

Innovation-driven approaches prescribe principles, practices and tools that are used to develop innovative products or services. Introducing elements of these innovation-driven approaches in the context of software processes allows for aspects related to business, market and creativity to be considered in the design of an innovative software product. However, there are gaps between innovation processes and software processes. This work seeks to address the relationship between elements of innovation processes and elements of software processes through traceability. To this end, this work proposed an approach called InnoTrace that aims to enable tracking of innovation requirements towards system requirements and vice versa. The InnoTrace approach consists of: specifying signs to represent innovation requirements and system requirements; specifying a track to represent the relationship between signs through the notation provided by the SysML language; and traces that consist of following the track in the context of cases that demonstrate the application of software processes that incorporate innovation practices and techniques. The main contribution of this work is to provide a method appropriate for documenting traceability relations (cause and effect) between innovation and system requirements and thus mitigate the hiatus commonly observed between innovation and software processes.

## CCS Concepts

**• Software and its engineering→Software notations and tools→Development frameworks and environments.**

## Keywords

Requirement Traceability; Innovation Process; Software Process.

## 1. INTRODUCTION

In recent years, businesses have been working in environments that are both global and constantly changing. Software use has become a part of practically all business operations to attend to new opportunities and markets, turning itself into one of the main items for investment. According to the Gartner Group, just in 2021 the world software market grew 7.2%, totaling US$ 492.4 million against the US$ 459.2 million registered in 2020 [1].

The software development process is more complex than a few years ago. New roles, techniques, practices, and tools have been adopted to improve software quality. Software quality can be defined as “an effective software process applied in a manner that creates a useful product, providing measurable value for those who produce it and those who use it” [2].

The ISO 9001 Standard [3] states that user satisfaction is met by monitoring relevant information on whether or not the software has attended to the desired requirements. The software requirements, in turn, are activities that the software must perform, with its limitations and restrictions, besides the characteristics not directly linked to the functions performed by the software [4].

The software requirements can present problems in their definition, as when users are not sure of their necessities, or do not understand the capacities and limitations of the computer environment, or do not have full control of the problem; have difficulty in expressing demands; and omit information which they believe to be obvious [4].

Requirements Engineering seeks to resolve these problems from requirements development activities, such as elicitation, analysis, specification, and verification; and from requirements management activities, such as changes control, configuration control, traceability, and quality control [4].

According to Kotonya & Sommerville [5], requirements management has, as its finality, control: the requirements and their changes; the traceability between requirements; and the traceability between artifacts produced during the software development lifecycle.

According to Berg, Bishop, and Muthig [6], requirements traceability describes the relationship or dependency between artifacts to provide a better understanding of how the requirements were produced.

To develop successful market software, it becomes necessary to go beyond and “learn what the users real needs, and not what the users say they want or what we think they want” [7]. To this end, tools, practices, and techniques from innovation-driven approaches can be applied in the context of software development.

Amongst the main innovation-driven approaches are Business Model Generation [8], Blue Ocean Strategy [9] [10], and Design Thinking [11]. The Business Model Generation describes how a business creates, delivers, and captures value through a Business Model. The Blue Ocean Strategy seeks to exploit new market spaces and make the competition irrelevant. Design Thinking is centered on the human being and seeks to resolve problems creatively. These approaches provided tools, techniques, and practices that can be used for an innovative software development process [12] [13].

According to Beyhl [14], there exists a gap between innovative processes and conventional software development processes. The main problem is in the documentation produced in an innovation process, generally informal and analogical (post-it, prototypes on paper, etc.). The reasons behind a design decision, alternatives evaluation, and user feedback are often not considered in the delivery of the artifacts. In Design Thinking, for example, the delivery is ideas and prototypes. However, some relationships, alternatives, feedback, and rejected/unused requirements are generally ignored, causing a loss of important information used in making decisions for software design [14].

The traceability technique can be used to mitigate the gap between innovative processes and conventional software development processes. To this end, it is necessary to document the relationships between the requirements extracted from artifacts produced in the innovative processes and conventional software development processes.

Faced with this context, the research problem that drove this work consists of the following: how does one perform the tracing of requirements extracted from innovation processes toward the system requirements extracted from conventional software development processes (and vice-versa)?

As an alternative response to the research problem, we present a requirement tracing approach denominated InnoTrace. InnoTrace is concerned with (1) the relationships tracing between the requirements

extracted from the innovation processes and (2) relationships tracing between the requirements extracted from innovation processes and the requirements extracted from conventional software development processes.

Besides this introductory section, this work is structured into five more sections. Section II presents reference works for the development of the InnoTrace approach. Section III presents some of the main related works. Section IV presents concepts, signs and their relationships (tracks) inside the InnoTrace approach. Section V presents a case study through the demonstration of a trace oriented by the InnoTrace approach. Section VI presents the final considerations, including the main limitations and proposals for future work.

# 2. BACKGROUND

In this section the approaches which make up the theoretical framework for this study are detailed.

## 2.1 Innovation-driven Approaches

Innovation refers to new applications of knowledge, ideas, methods, and skills that can generate unique capabilities and give leverage to an organization's competitiveness [15]. Currently, many companies employ the innovation process to obtain competitive advantages in business. The process of innovation can be described as changes in the way an organization produces a product or service [16].

Diverse approaches exist which are utilized in the development of innovative solutions directed towards the market of manufactured products and traditional services for clients. Amongst the most noteworthy approaches which help to look for innovation are: Design Thinking, Blue Ocean Strategy, and Business Model Generation.

These approaches are references in this work since they provide tools, practices, and techniques to help in tasks execution for InnoTrace.

The Business Model Generation [8] approach provides a method to illustrate and clarify the business model of an organization in a way of communicating the general panorama and establishing a common language. The objective is to fill in, in a structured manner, the construction blocks defined in the Canvas Business Model tool. The building blocks of the Canvas Business Model are the following:

- Customer segments (CS) - aims to identify those who want to create value;
- Value Propositions (VP) - aims to identify which value is delivered to the customer segments. The value can be proposed to solve a problem or to serve a customer need;
- Channels (CH) - describe how an organization communicates with and reaches its customer segments to deliver its value proposition;
- Customer Relationships (CR) - aim to establish and maintain each customer segment;
- Revenue Streams (RS) - represent how revenues will be generated by each customer segment;
- Key Resources (KR) - describes the main features that are required to make the business model work;
- Key Activities (KA) - describes the most important actions that an organization must take to operate successfully;
- Key Partners (KP) - describes the main partners and providers that are necessary for the business model to work;
- Cost Structure (CS) - describes the most important costs inherent in the operation of the business model;

Blue Ocean Strategy focuses on discovering new market niches, offering clients something exclusive, not yet exploited in a determined segment. It produces, in this sense, the so-called value innovation which aligns innovation with immediate use, with competitive prices and gains in costs [10].

In the red oceans (the traditional Market), the sectorial frontiers are defined and accepted, and the competitive rules of the game are known. The organizations try to overcome their competitors in order to acquire a bigger slice of the existing demand. In the blue oceans the competition is irrelevant since the rules of the game have not yet been established.

Amongst the available tools for the approach to help in the identification of blue oceans are: Strategy Canvas, Four Actions Framework, and the ERRC Grid.

Strategy Canvas has as its purpose to capture the current situation of the market segment to understand what the competitors are investing in and identify the factors on which the competition is based. The value curve is the basic component of the strategy canvas. It is a graphic depiction of a company's relative performance across its industry's competing factors.

Figure 1 [9] presents a reference model for the Strategy Canvas. The horizontal axis defines the competing factors used in the comparison between the average of the approaches found in the market (industry value curve) and the proposed approach (blue ocean's strategic moves).

The proposed approach generally adds new attributes not met by the market. The vertical axis of the matrix identifies the offering level and factors (represented by dots) mapping the compared approaches on a scale of lo (low) to hi (high).

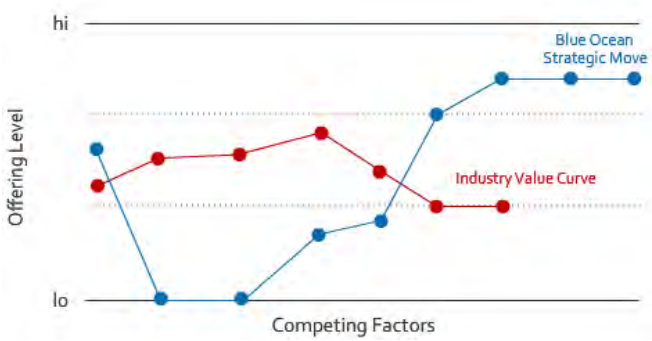


**Figure 1. Strategy Canvas**

Four Actions Frameworks has as its purpose to create a new value curve through the redefinition of the level of offering factors which dictate the competition in a way to differentiate the proposed solution of the competitors.

For this purpose, it is necessary to answer the following questions:

- Which are the factors - that the industry has long competed with - which should be eliminated?
- Which factors should be reduced well below the industry's standard?
- Which factors should be raised well above the industry's standard?
- Which factors should be created? The ones that the industry has never offered?

The Eliminate – Reduce – Raise - Create (ERRC) Grid compliments the Four Actions Framework. It pushes companies not only to ask the questions posed in the Four Actions Framework, but also to act on all four to create a new value curve, which is essential to unlocking a new blue ocean.

Design Thinking is "a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible

and what a viable business strategy can convert into customer value and market opportunity" [11].

Design Thinking is executed by putting together the following stages: Inspiration, in which the circumstances are identified (problem, opportunity, or both) motivating the search for solutions; Ideation, which is the process of generation, development and testing of ideas - and can lead to solutions; and Implementation, which is the path which leads the product/service to the market.

Design Thinking looks for innovation from restrictions imposed by the criteria of: practicability, that which is functionally possible in the near future; viability, that which will probably turn itself into part of a sustainable business model; and desirability, that which makes good sense for people.

Design Thinking provides a set of tools and techniques to help the process of innovation. Amongst them are:

- Divergence and convergence: diverging thinking creates options whilst the converging makes choices;
- Analysis and synthesis: the analysis is used to take apart complex problems, to the end of understanding them better; synthesis is the base of the creative process, being the collective act of joining the parts to create complete ideas;
- Attitudes of experimentation: conceding time, space and budget for creative teams to learn through error;
- Brainstorming: generating creativity across the free flow of ideas;
- Prototyping: producing any tangible thing to exploit, assess, and put forth as an idea;
- Storytelling: to elaborate scenarios which describe some future situation which shows potential, using words and images;
- Experience design: mapping stages through which the imaginary client can run from the start to the finish of an experience of using a product or a service.

## 2.2 Foundations for Traceability

The Requirements Traceability can be defined as "the capacity to describe and accompany the lifecycle of a requirement in both directions, from the origin (cause) and from the implementation passing through all related specifications" [17].

Gotel e Morris [18] present as foundations for requirements traceability the concepts sign, track and trace. Sign is "an identifying mark made by, or associated with, for a particular purpose". Track is "a pattern of signs". Trace consists of "follow a track, sign to sign". The premise is that there is "no ability to trace without a track and there is no ability to lay a track without making signs".

#### *2.2.1.1 Sign*

Sign here exists as "an identifying mark made by (or associated with for a particular purpose) an animate or inanimate object". For example, a sign may be an animal print or some disturbance in the environment (e.g., droppings, hair, etc.), an industry standard code, appropriate to different stages in the food chain, an industry standard code.

A number of factors are closely related to the concept of sign. The questions addressed are: What is the animate or inanimate entity of interest (signed entity)? What is it that marks the 'movement' of the signed entity (sign)? Who or what makes the sign (sign maker)? Is the sign associated with or directly made by the sign maker (mode of sign)? What carries or bears the sign (the medium of the sign)? What does an instance of a sign stand for (what the sign represents)? How long will a sign survive (permanence of sign)?

In the context of innovation processes, a sign can represent a Value Proposition, an Insight or an Idea. Value Proposition is the centerpiece of a business model and consists of a promise of delivery of specific benefits for a target public. Insight and Idea are derived from creative activities. Insight is an innovation opportunity identified in immersion in the problem. Idea is a proposed solution to attend to one or more insights.

In the context of software processes, a sign can represent a System Demand or a Functionality. System Demand is a capacity required for a software system. Functionality is an implementation to satisfy a system demand.

### *2.2.2 Track*

Track is "a pattern of signs created as these signs are generated". For example, a track may represent the movement of the animal in space and time, the replication or mutation of the pathogen, and the infection of a new host or the movement of an item of luggage in space and time.

As with sign, a number of factors are related to the concept of track. The questions addressed are: What activity causes a track to be made (cause of track generation)? Who or what creates the track (track maker)? What does an instance of a track stand for (what the track represents)? What shape does the track have (pattern of track)? How does the track begin and end (origin and destination of track)? How long will the track remain (permanence of track)?

The track is a metamodel which defines the conceptual relationships between signs such as, for example, those presented in Figure 2. In this example, an Insight can be derived from a Value Proposition; an Idea seeks to satisfy an Insight; a System Demand can be derived from an Idea; and a Functionality seeks to satisfy a System Demand.

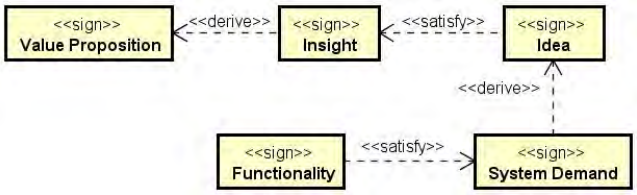


**Figure 2. Example of Track**

### *2.2.3 Trace*

A trace consists in following a track, i.e., the trace is the occurrence of the track. Trace means "to identify a track following its pattern sign by sign". Traceability is 'a quality of an entity allowing it to have a track, howsoever made, plus the availability of a means to follow that track'. For example, a trace may be used to identify and find an animal, to determine the potential, spread of a disease, to identify and establish the origin of a work of art, etc.

As with sign and track, a number of factors are related to the concept of trace. The questions addressed are: What is the intention behind the trace (trace objective)? Does the trace go one way or another (direction of trace)? Who makes it (trace maker)? Who uses it (trace user)? Who benefits from its creation and use (trace beneficiary)? By what actions does the trace come about (trace process)? What assists trace making (aids to tracing)? What happens if there is no sign where one is expected (impact of gaps in tracing)? How long will the trace remain (permanence of trace)?

The trace is an instance (occurrence) of the track. The trace consists of identifying the course of signs in a real context. For example, in a travel controlling system, a Value Proposition can be the interaction between travelers; an Insight can be the sharing of travelling experiences; an Idea would be the use of georeferencing to suggest interesting places for the travelers; a System Demand would be the sharing of information and experiences between system users; and a

Functionality can be the check-in and commentary of travelers in the visited places and establishments.

## 2.3 SysML

Language of Systems Modeling (SysML) is a general purpose visual modeling language for systems engineering applications [19]. SysML provides modeling mechanisms to represent requirements based on text and to relate them with other model elements. SysML specifies a diagram of specific purpose denominated Requirements Diagram which is used to represent requirements in a graphic format.

The Requirements Diagram permits the specification of the relationships between requirements which include hierarchy between requirements, requirements derivation, attendance (satisfaction of) requirements, requirements verification and requirements refinement.

Table 1 [20] presents the semantics of types of relationships between requirements and test cases which can be used in the Requirements Diagram.

**Table 1. Types of Requirements Relationships in SysML**

| ***Relationship*** | ***Description*** |
|---|---|
| *Nest* | *Specifying requirements hierarchies precludes reusing requirements in different contexts since a given model element can only exist in one namespace.* |
| *Copy* | *The relationship is a dependency between a supplier requirement and a client requirement that specifies that the text of the client requirement is a read-only copy of the text of the supplier requirement.* |
| *Derive* | *The relationship is a dependency between two requirements in which a client requirement can be derived from the supplier requirement. For example, a system requirement may be derived from a business need.* |
| *Satisfy* | *Relationship is a dependency between a requirement and a model element that fulfills the requirement.* |
| *Verify* | *Relationship is a dependency between a requirement and a test case or other model element that can determine whether or not a system fulfills the requirement.* |
| *Refine* | *The refine requirement relationship can be used to describe how a model element or set of elements can be used to further refine a requirement. For example, a use case diagram may be used to refine a text-based functional requirement.* |
| *Trace* | *The relationship provides a general-purpose relationship between a requirement and any other model element.* |
| *Dependency* | *The general relationship of dependency where one requirement is dependent on the other.* |

Figure 3 [21] presents an example of the elements and relationships of a SysML Requirement Diagram.

In accordance with Figure 3, a complex requirement (REQ_A1) can be decomposed into two sub-requirements (REQ_A1.1 e REQ_A1.2). The relationships of derivation (<<deriveReqt>>) and copy (<<copy>>) can only exist between requirements. The relationships trace (<<trace>>), refinement (<<refine>>), satisfaction (<<satisfy>>) and verification (<<verify>>) can exist between a requirement and whichever other element of the model.

The relationship of verification can only exist between a requirement and a behavioral element stereotyped as a test case (<<test case>>). A test case is a method for verifying whether or not a requirement is attended to.

The commentary notes, for example, to relate the problem (<< problem>>) and the reason (<<rationale>>), can be added as required to any element of the model in order to capture some features and decisions.

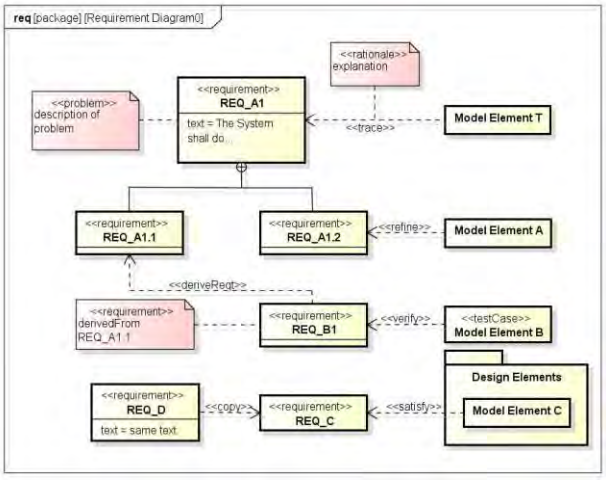


**Figure 3. Example of Requirements Diagram**

# 3. RELATED WORK

Beyhl et al. [14] investigated how the projects of Design Thinking are documented. At the end of each project the design thinkers present an idea, demonstrate a prototype and produce an informal document which describes the idea. However, the way by which the idea was reached, including reasons, alternatives and user feedback, and a list of attended to/ rejected requirements are not considered.

The final presentation and prototype are only enough to pass on a general vision of that which needs to be constructed. Software engineers need more detailed information to construct the software product. There is a gap between the Design Thinking approach and the software process [14].

Faced with this scenario, Beyhl et al. [14] describe how the concepts of sign, track and trace can be applied in Design thinking projects.

A sign in Design Thinking corresponds with the identification of a type of information in an artifact (signed entity). These signs are manually created by Design Thinkers (sign fabricators) and need to be annexed to artifacts (sign environs) to make it possible for software engineers to re-establish the context in which creation occurred. Each sign substitutes the artifact in itself for the representation of the artifact. The sign will exist until the moment it is destroyed (sign permanence).

A track is the movement of artifacts between differing contexts (cause of the track). Design Thinkers (track fabricators) capture the state of the work in various points in time, in this way presenting the contexts in which the artifacts are utilized (representation of the track). A track originates in the creation of an analogous artifact (a post-it, for example) and stops when the analogous artifacts no longer need to be captured by digital artifacts (for example, in a photo).

The track is the course that Design Thinkers construct during the innovation process (track objective). The primary users of the track are software engineers. Software engineers track down in the reverse direction (“backwards”) from the final presentation or prototypes to arrive at the reason or cause of a project decision. The secondary users are the Design Thinkers who use the track to reflect on iterations – meaning that the track permits Design Thinkers to understand the process and why some paths were taken, and not others.

The approach proposed by Beyhl et al. [14] applies the concepts of signs, track and trace to promote traceability between analogical artifacts produced in the Design Thinking process. The InnoTrace approach now seeks to apply the same mechanisms of traceability,

although to control the relationships between innovation requirements extracted from the artifacts produced with the help of some practices and techniques provided by the complementary approaches of Design Thinking, Blue Ocean Strategy and Business Model Generation.

According to Maiden et al. [22], the process of software development can incorporate problem-solving activities based on the CPS method (Creative Problem-solving*)* [23].

The CPS Method supports six stages of problem-solving: (1) finding objectives - seeks to establish objectives and goals relevant to a new problem; (2) fact finding - seeks to discover other types of knowledge needed during problem solving; (3) problem finding - encourages exploration of a problem by quickly expressing it in different ways in order to provide different insights into it; (4) idea finding - seeks to generate as many ideas as possible; (5) solution finding - taking action on selected ideas, and developing solution selection criteria; and (6) acceptance finding - considering real-world issues to overcome implementation issues.

Creativity techniques and software tools can be applied to fill the gap and find new ideas expressed as requirements [22]. Basic research is needed to develop new models of requirements engineering that incorporate idea finding. One role of the models is to define and recognize a creative requirement in the set of information artifacts generated during a requirements process. A second role is to describe the processes with which stakeholders can effectively generate creative requirements from idea finding.

The InnoTrace approach combines a set of practices and techniques provided by the Design Thinking, Blue Ocean Strategy and Business Model Generation approaches, being appropriate to give support to the generation of ideas, as well as the creation of artifacts (models) from which creativity requirements can be extracted.

Souza, Cysneiros Filho and Batista [13] present an heuristic approach for supporting innovation in requirement engineering. The proposed approach consists of the following tasks: (1) Identify Innovation Opportunities; (2) Evaluate Competitors; (3) Specify Competing Factors; (4) Ideation; (5) Design Business Model; (6) Develop Vision; and (7) Define Product Backlog.

Identify Innovation Opportunities consists of the observation and study of a market segment to identify opportunities for products and services that meet undeclared (latent) needs of people and turn them into possible demands.

In Evaluate Competitors, related solutions in the existing target market are evaluated to avoid creating something already existing. The evaluation is based on objective criteria such as reputation (i.e. user feedback) and popularity (i.e. number of downloads).

Specify Competing Factors is to define which competing factors should be eliminated, reduced and increased in relation to the industry standard; appropriate factors not offered by the industry should be created.

In Ideation, the team goes in search of insights and ideas to describe the new value curve in order to differentiate the proposed solution compared to the market average.

Design Business Model aims to produce an innovative business model. A business model describes "the rationale of how an organization creates, delivers and captures value" [8].

Develop Vision aims to produce a Vision. The Vision describes a view of the solution to be developed and it should reflect the stakeholders' needs, along with the features that are proposed to address those needs.

Define Product Backlog aims to elaborate Product Backlog. The product backlog is "an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product" [24].

The tasks of the approach by Souza, Cysneiros Filho and Batista [13] are executed with the help of tools provided by innovation approaches [8][10][11] used as a reference in the InnoTrace approach.

The InnoTrace approach provides means for tracing innovation requirements, which permit one to identify the reasons behind a decision about design, as well as the influence (effect) of requirements innovation in system requirements.

The InnoTrace approach is complementary to the approach proposed by Souza, Cysneiros Filho and Batista [13]. In this work we use the case study demonstrated in the approach proposed by Souza, Cysneiros Filho and Batista [13] for the specification of the corresponding trace (Section V).

# 4. THE INNOTRACE APPROACH

In this section, the definition and the concepts of the InnoTrace approach are the first to be presented. Subsequently, the signs which show the requirements extracted from innovation processes are specified. Finally, the track, which describes the relationships between the signs and which consists of a metamodel to drive the traces, is specified.

## 4.1 Definition

The InnoTrace approach is concerned with the tracing of requirements extracted from artifacts produced during an innovation process for the inception of software products.

The InnoTrace approach is based on the concepts of sign, track and trace. The sources for the extraction of signs are artifacts produced by techniques and tools provided by innovation-driven approaches: Design Thinking [11], Blue Ocean Strategy [10] and Business Model Generation [8].

The requirements extracted from an innovation process comprehend creativity requirements, business requirements, market requirements and user requirements. Such requirements can derive system requirements. The system requirements comprehend functional and non-functional requirements.

For the representation of elements and their relationships, defined in the InnoTrace approach, the Diagram for Requirements of the SysML [18] language was used.

The SysML language was chosen for the following reasons: it is a standard maintained by OMG [25]; it has the support of commercial modeling tools [26]; and supplies notation that can be used in the representation of signs and track, in agreement with what is proposed by the InnoTrace approach.

Figure 4 presents the high-level relationships between the types of requirements in the InnoTrace approach.

In the context of the innovation processes: a Business Requirement can be derived from a Market Requirement or from a User Requirement; a Creativity Requirement can satisfy a Business Requirement, a Market Requirement or a User Requirement.

In the context of the conventional software processes: a System Requirement generalizes a Functional Requirement or a Non-Functional Requirement.

The dependency between packages denotes that the requirements extracted from software processes can be derived from the requirements extracted from innovation processes.

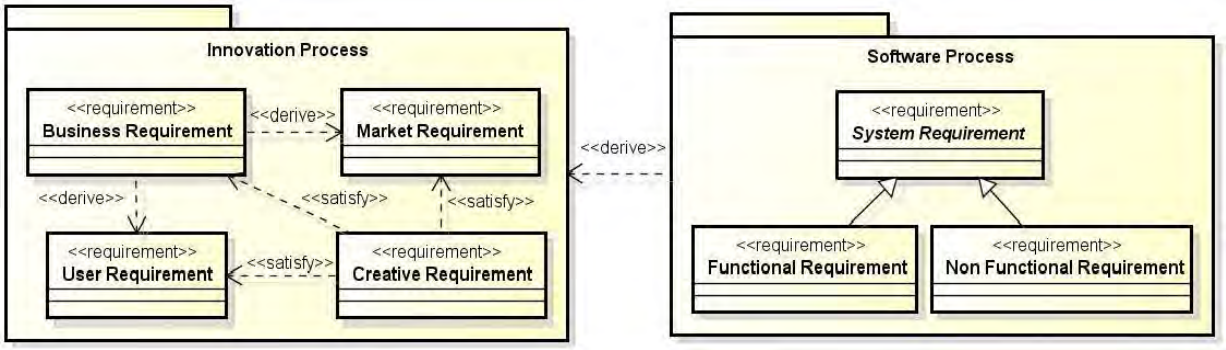


**Figure 4. Relationships between requirements inside InnoTrace**

Business Requirements are strategic requirements at a high level obtained from the organization's business model. A Business Model can be defined as a conceptual tool containing a set of objects, concepts and the relationships between them, with the objective of expressing the organization's business logic – what value is provided for the clients, how this is done and what the financial consequences are [27]. The concept of Business Model can help to improve engineering requirements [27].

Market Requirements are high-level strategic requirements obtained from the analysis of competing factors in the market segment where the business performs. The analysis of the competing factors, which direct already established competitors in a market segment, helps an organization to create and capture new demand, focus on an unexplored market space and define a differentiated value curve in relation to the status quo [10]. Differentiation today gains a position as one of the most powerful strategies in the business world, the one which gains the most from innovation [28].

The difference between Business Requirements and Market Requirements is that the first is centered on the internal perspective on the organization, while the second is centered on the external perspective of the organization in relation to competitors, i.e., how the organization is positioned in relation to the competition in the market segment where it performs.

Creativity Requirements are useful and original requirements extracted from artifacts produced in creative problem-solving activities [22]. Among the general activities involved in generating creative requirements are: understanding the problem, producing ideas and finding a solution. According to Maiden et al. [22], it becomes necessary to develop applied research in order to extend requirement processes with techniques in creative processes which give support to the generation of ideas.

User Requirements are operational requirements referring to necessities, demands and restrictions made explicit by users.

System Requirements are the basis for the construction of a software system. System Requirements are classified into Functional and Non-Functional Requirements.

Functional Requirements describe functionalities and demands that software systems make available to attend to users' needs. Non-Functional Requirements are the restrictions in services offered by the software system or attributes of a particular software quality, or of the software as a whole.

In relation to other requirement traceability approaches, the difference in the InnoTrace lies in the utilization of the Requirement Diagram from SysML in terms of the specification of the track (relationships between signs) and the description of the traces.

## 4.2 Signs of the InnoTrace approach

A sign represents a mark in an artifact which shows an innovation requirement. The sign is the basic unit for the movement of tracing innovation requirements toward system requirements. A sign can be extracted from one or more artifacts and an artifact can contain one or more signs.

The artifacts which contain the signs are produced from tools supplied by Business Model Generation (BMG), Blue Ocean Strategy (BOS), and Design Thinking (DT) approaches.

The Business Model Canvas tool, supplied by the BMG approach, is used for the specification of the Business Model which is the base artifact for the extraction of signs which describe Business requirements.

The Strategy Canvas and ERRC Grid tools, provided by the BOS approach, are used for the specification of the Value Curve which is the base artifact for the extraction of signs which describe Market requirements.

The DT approach, in turn, oversees a user-centered set of tools and techniques for the production of artifacts, such as Experience Design, Empathy Map, Affinity Diagram, Insight Cards, Ideas Menu, amongst others. From these artifacts are extracted the signs which represent the Creativity and User requirements.

In the context of this work, the signs which represent the System Requirements (Functional and Non-Functional) are derived from the requirements extracted from innovation processes. Table 2 describes the signs for each type of requirements and, for each sign, the respective identifier, purpose and source of extraction.

## 4.3 Track

Track is a line of signs. The track represents the possible relationships between the signs. The track consists of a pattern of signs which defines the movement of tracing requirements extracted from the process of innovation towards system requirements. Figure 5 presents the track specified by the InnoTrace approach through a Requirement Diagram of the SysML language.

**Table 2. Signs identified by InnoTrace**

| Requirement Type | Sign | Identifier | Purpose | Source |
|---|---|---|---|---|
| Business Requirement | Value Proposition | VP | Identifies the value submitted to clients, which can be the solution of a problem or satisfying a demand/ necessity. | Business Model |
| | Customer Segment | CS | Identifies to whom one desires to create/ deliver value. | |
| | Channel | CH | Identifies the means to reach and deliver value to clients. | |
| | Customer Relationship | CR | Identifies the means to establish and maintain clients. | |
| | Revenue Stream | R$ | Identifies the incomes generated by each client segment. | |
| | Key Activities | KA | Identifies the most important actions for the organization to operate successfully. | |
| | Key Resource | KR | Identifies the main assets required for the business to function. | |
| | Key Partner | KP | Identifies the main partners and providers which are necessary for the business. | |
| | Cost Structure | C$ | Identifies the most important costs inherent to the functioning of the business. | |
| Market Requirement | Competing Factor | CF | Identifies the main factors which (re)direct the competition in a market segment. | Value Curve |
| | Offering Level | AL | Identifies the level of attendance to each competitive factor in a relative scale between zero and ten. | |
| Creative Requirement | Premise | PR | Identifies an argument or premise generally obtained by an immersion in the context of the problem to be resolved. | Exploratory Research; Desk Research; Interviews; Conceptual Map; *Personas*; Empathy Map; Prototypes. |
| | Insight | IN | Identifies a latent demand. | Insight Cards; Affinity Diagram; Staging Scenarios; *Brainstorming*; Empathy Map; |
| | Idea | ID | Identifies a solution for one or more insights. | *Brainstorming*; Co-creation Workshop; Menu of Ideas; Prototypes. |
| User Requirement | User Demand | UD | Identifies a demand or necessity made explicit by users. | The same as Creative Requirement |
| | User Constraint | UC | Identifies a restriction for a solution imposed by users. | |
| Functional Requirement | Functionality | FU | Identifies a functionality to attend to a system demand. | Software Process |
| | System Demand | SD | Identifies a capacity or necessity required by the system. | |
| Non-Functional Requirement | System Constraint | SC | Identifies a constraint or attribute of software system quality. | Software Process |

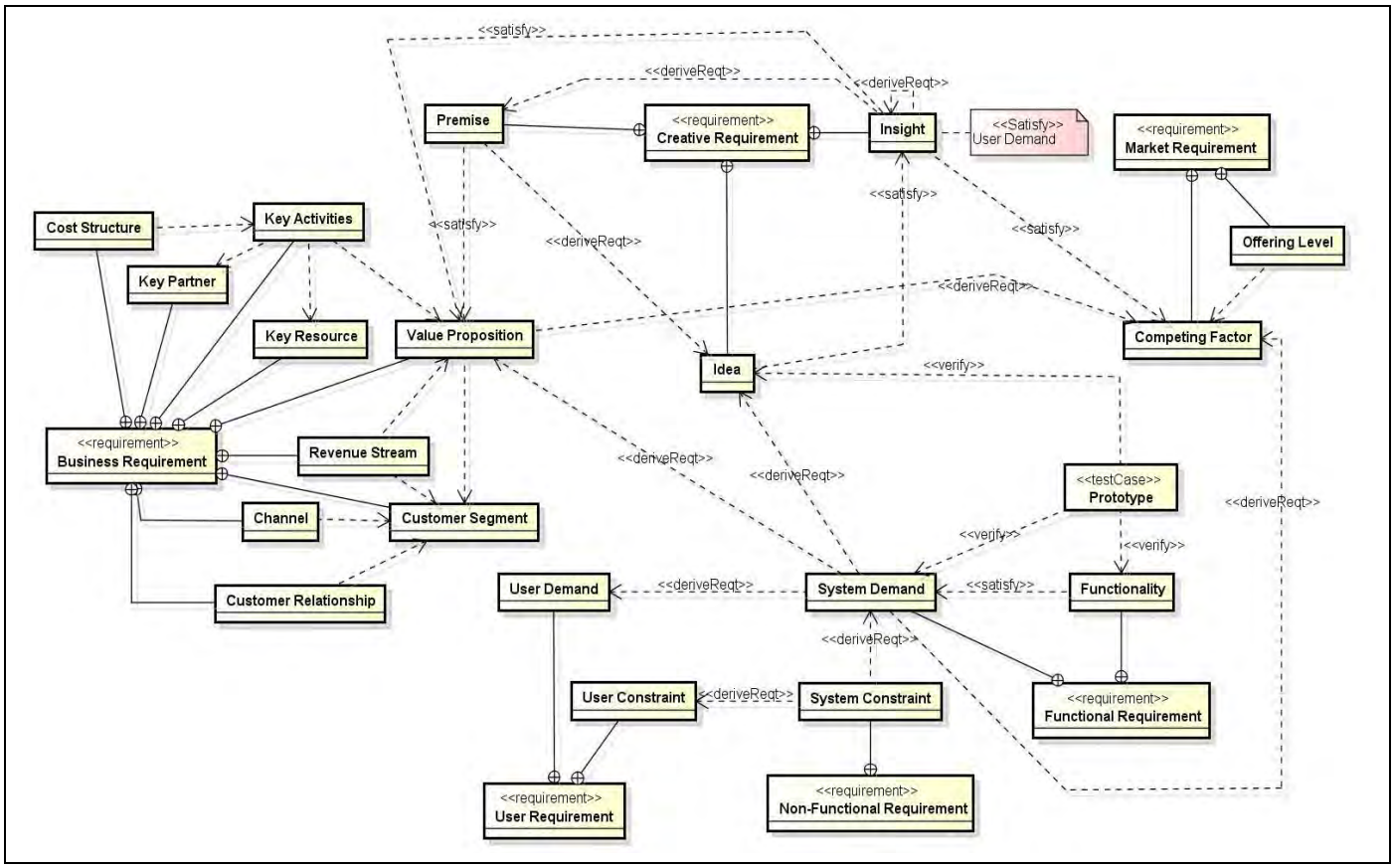


**Figure 5. Track for the InnoTrace approach**

The Track (Figure 5) also presents nesting between requirements and signals, which means that an innovation or system requirement can be described by a set of signs. Table 3 presents, for each relationship Nest between requirements and signs, the respective explanation.

**Table 3. Relationship Nest between requirements and signs**

| **Requirement** | **Sign** | **Explanation** |
|---|---|---|
| Business Requirement | Value Proposition | Delivery of value is the business objective. |
| | Customer Segment | Value is destined to a target public. |
| | Customer Relationship | Ways in which the client is reached and maintained. |
| | Channel | Ways that value is delivered to clients. |
| | Revenue Stream | The business should be viable and sustainable. |
| | Key Activities | For the business to work some fundamental activities are necessary. |
| | Key Resource | The activities demand essential resources. |
| | Key Partner | Supply chain for a business. |
| | Cost Structure | Any business involves costs. The costs should be less than the revenues. |
| Market Requirement | Competing Factor | Factor which directs competition in the market. |
| | Offering Level | Level which directs investment in a competing factor. |
| Creative Requirement | Premise | Premise associated with an idea. |
| | Idea | Creative solution for a problem. |
| | Insight | Relevant demand or problem which emerges during the creative process. |
| User Requirement | User Demand | Demand made explicit by the user. |
| | User Constraint | Restriction made explicit by the user. |
| Functional Requirement | Functionality | Function (service) hoped for in the system. |
| | System Demand | High-level requirement for the system. |
| Non-Functional Requirement | System Constraint | Quality or limitation of a demand of the system or of the system as a whole. |

The Track (Figure 5) defines further the relationships between the signs which serve to guide the trace of the requirements extracted from the innovation process towards system requirements, and vice-versa.

Each type of relationship possesses a purpose which permits one to identify the contribution presented by each link. Table 4 presents, for each relationship between signs, the type of relationship and respective explanation.

**Table 4. Relationship between Signs**

| Source Element | Target Element | Relationship | Explanation |
|---|---|---|---|
| Value Proposition | Customer Segment | Dependency | The value to be delivered is directed towards a target consumer. |
| Customer Relationship | Customer Segment | Dependency | The ways of reaching and maintaining clients are customized for each client segment. |
| Channel | Customer Segment | Dependency | The means for value delivery belong to each client segment. |
| Revenue Stream | Customer Segment | Dependency | The revenues obtained with value delivery are specific for each client segment. |
| Revenue Stream | Value Proposition | Dependency | The revenues obtained by value delivery. |
| Key Activities | Value Proposition | Dependency | The main activities are directed towards the delivery of the promised value. |
| Key Activities | Key Resource | Dependency | For the execution of activities appropriate resources are necessary. |
| Key Activities | Key Partner | Dependency | The effectiveness of activities depends on a nucleus of providers and partners. |
| Cost Structure | Key Activities | Dependency | The cost is inherent for each performed activity. |
| Offering Level | Competing Factor | Dependency | For each competing factor an appropriate level of assistance is required. |
| Value Proposition | Competing Factor | DeriveReqt | The value proposition can emerge from the evaluation of the competition. |
| Idea | Insight | Satisfy | An idea is a high-level proposition for attending to a latent demand. |
| Premise | Idea | DeriveReqt | A premise can be related to a specific proposal for problem resolution. |
| Insight | Premise | DeriveReqt | An insight can emerge from a defined premise in the context of the problem. |
| Insight | Value Proposition | DeriveReqt | An insight can emerge from immersion in the context of the problem for value delivery. |
| Insight | Insight | DeriveReqt | An insight can emerge from the contextualization of other insights. |
| Insight | Competing Factor | Satisfy | An insight can emerge from immersion in market analysis. |
| Premise | Value Proposition | Satisfy | A premise can be related to the delivery of a specific value. |
| Prototype | Idea | Verify | A test case can be used to validate an idea. |
| Insight | User Demand | Satisfy | An insight can emerge from a necessity made explicit by the user. |
| System Demand | User Demand | DeriveReqt | A system demand (functional requirement) can emerge from a demand made explicit by the user. |
| System Constraint | System Demand | DeriveReqt | A system restriction can be made specific by a system demand. |
| System Constraint | User Constraint | DeriveReqt | A system restriction can emerge from a restriction made explicit by the user. |
| System Demand | Idea | DeriveReqt | A system demand can emerge from an idea. |
| System Demand | Value Proposition | DeriveReqt | A system demand can arise from a value to be delivered. |
| System Demand | Competing Factor | DeriveReqt | A system demand can be specific to attend to a competing factor. |
| Prototype | System Demand | Verify | A test case can be used to validate a system demand. |
| Functionality | System Demand | Satisfy | A functionality seeks to attend to a system demand. |
| Prototype | Functionality | Verify | A test case can be used to demonstrate functionality. |

## 5. Case Study

Trace is an occurrence of the Track. Trace is directed by the pattern of signs defined by the Track. Tracing is the act of tracing, meaning the result of the Trace.

In the InnoTrace approach, the Trace follows the signs (Table 2) which represent the requirements extracted from innovation and software processes through the Track (Figure 5).

In terms of demonstrating the trace, following the track of the InnoTrace approach, we utilize as a base the case study presented in the approach proposed by Souza, Cysneiros Filho and Batista [13].

The direction of the tracing went backward with the objective of validating system requirements. In other words, for the system requirement evaluated, derived from innovation requirements extracted from innovation processes, the respective causes were identified at their source.

Tracing in the direction of system requirements (effect) towards requirements extracted from innovation process (cause), through the InnoTrace approach, will be presented according to the following steps:

Brief presentation of a study case demonstrated in the process for conceiving innovative software, following the approach by Souza, Cysneiros Filho and Batista [13];

a) Selection of a system requirement presented in case study for demonstration of tracing in the backward direction;
b) Specification of the trace for the selected system requirement, through a Requirements Diagram of the SysML language;
c) Specification of a Tracing Matrix for the selected system requirement;
d) Description of the source artifacts from which the trace signals were extracted.
e) Discussion on traceability demonstrated by the InnoTrace approach.

## 5.1 Case Study Presentation

The case study consisted of a solution conception to give support to the travel planning, through the approach proposed by Souza, Cysneiros Filho and Batista [13]. The case study was executed by a team of the innovation laboratory at Federal Rural University of Pernambuco, Brazil.

The first step was to evaluate some of the most popular solutions used for travel planning. To this end the tripit.com and worldmate.com applications were evaluated.

After assessment of the competition, competing factors on support for planning trips were identified. The Competing Factors that govern markets are the following: (A) Multiplatform - availability of the software product on web and mobile devices platforms; (B) Social Media Integration – used for posting of information on platforms such as Facebook and FourSquare; (C) Georeferencing - use of information from GPS to provide guide services; (D) Geographic Search – a means for searching based on geographic attributes; (E) Offline Content – storage of information on local devices; (F) Internationalization - support for different languages; (G) Multimedia Content - video, text, image; (H) Search/Assistance Services - support services such as hotel reservations and flights; and the search for suppliers and establishments related to the tourism business; and (I) Pricing – ways of registering charges: paid, free, and mixed.

Figure 6 shows the Strategy Canvas for evaluated software that assist in trip planning.

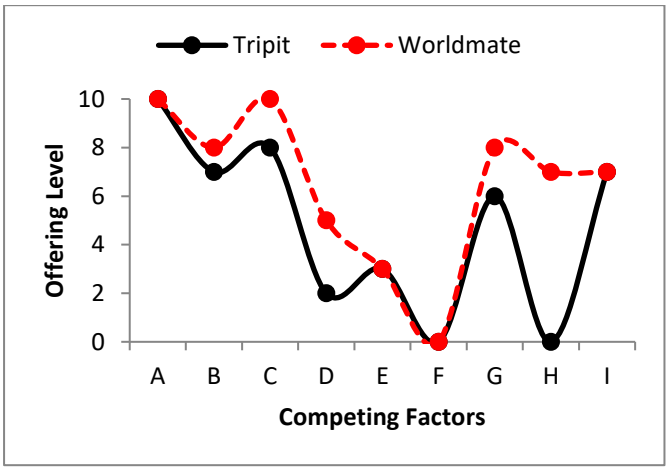


**Figure 6. Strategy Canvas of software for planning travel**

The next step was to specify a value curve for the software product that the proposed solutions differentiate with respect to the market, from the insights gained by the project team.

Thus, the Four Actions Framework was used to guide the project team to define the competing factors that should be eliminated and reduced, relative to the market average, and what value attributes not offered by the market should be created. The competing factors should be kept as they are currently on the market and are not considered in this model.

The ERRC grid of the software product proposed was then filled, as shown by Table 5.

The competing factors were created: (J) interaction between provider and tourist; and (K) personalized recommendations. The first is to enable the interaction between service providers and tourists as the events of the journey unfold, linking demand/supply. The second is to analyze personal information, preferences and affinities of tourists contained in social networks to define a profile and then customize the recommendation of products/services, programs and events.

**Table 5. ERRC Grid of Software Product Proposed**

| Eliminate | Raise |
|---|---|
| Search/assistance services (H) | Social media integration (B)<br>Geographic search (D)<br>Offline content (E)<br>Internationalization (F)<br>Pricing (I) |
| **Reduce** | **Create** |
| Multiplatform (A)<br>Multimedia content (G) | Interaction between provider and tourist (J)<br>Personalized recommendations (K) |

After that, a Strategy Canvas (Figure 7) was developed to enable a comparison between the value curve of the proposed software product and the average of the software products evaluated (market).

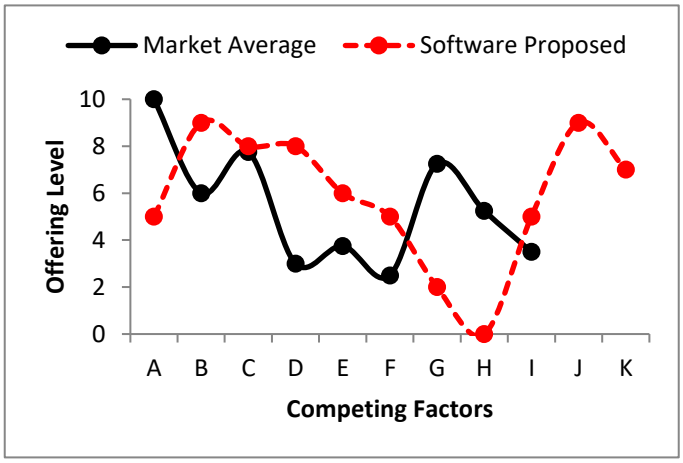


**Figure 7. Value Curve for the proposed software**

After the specification for the value curve, it was established that the proposed software product would offer the following competitive factors: Multiplatform - available only on the Web and Android mobile platform; Integration with social media - sharing experiences and travel plans; Geo-referencing - guidance through GPS; Geographic search - exploration geographical attributes to allow more targeted searches; Offline content – to allow use without total dependence on network data; Internationalization - adaptation to Portuguese, English and Spanish languages; Multimedia content – restriction of the display of images and videos; Search/assistance services - not providing any assistance for services such as booking hotels and flights; Pricing - using freemium/premium models; Interaction between provider and tourist – allowing participants to anticipate and provide services in accordance with the demands of tourists contained in the travel plan; Personalized recommendations – to suggest scheduling based on the user profile, defined from information extracted from social networks. Then the business model (Figure 8) was developed through the Business Model Canvas tool. Each block of the business model is described below.

Customer segments: the providers are citizens or companies that offer services for the tourist trip; tourists are individuals traveling for leisure or business, seeking offers and personalized recommendations.

Value Propositions: for the suppliers this would entail the discovery of demands early, or in real-time, for their businesses; for the tourist it would be peer interaction for sharing experiences, personalized recommendations, and the offer of products and services directed at their consumption profile.

Customer Relationships: for the supplier this would be a self-service system which presents demands updated in real time directed towards the profile of their business; for the tourist it would be a self-service system which presents offers updated in real time for the targeted consumer profile and community, making the exchange of information possible.

<table>
<tr><td rowspan="5">Key Partners<br><br>Marketing agencies<br><br>Government tourism agencies</td><td rowspan="3">Key Activities<br><br>Management meeting platform between supply and demand<br><br>Optimization recommendation engine</td><td>Value Propositions<br><br>Discover demands early in real-time</td><td>Customer Relationships<br><br>Self-service system with demands updated in real time</td><td rowspan="5">Customer segments<br><br>Provider<br><br>Tourist</td></tr>
<tr><td>Peer interaction<br><br>Personalized recommendations<br><br>Offer services and products targeted to profile</td><td>Self-service system offers updated in real time<br><br>Community</td></tr>
<tr><td rowspan="3">Key Resources<br><br>Reputation<br><br>Recommendation engine<br><br>IT system</td><td rowspan="3"></td><td rowspan="3">Channels<br><br>Web Application<br><br>Application for mobile devices</td></tr>
<tr><td colspan="2">Cost Structure<br><br>Human resources<br><br>IT system hosting</td><td colspan="2">Revenue Streams<br><br>Freemium<br><br>Free<br><br>Premium</td></tr>
</table>

**Figure 8. Business Model for proposed software**

Channels: for the supplier this would be a web-based application; for the client it would be an application for mobile devices.

Revenue Streams: for the provider there would be freemium (basic) and premium (advanced) versions; for the tourist they would always be free.

Key Activities: maintaining and managing the software platform to link supply and demand, optimizing the recommendation engine (algorithm) for offers targeted towards the tourist profile.

Key Resources: business reputation to attract and retain users; recommendation engine based on information collected from tourists on social networks; and from IT systems with the necessary infrastructure (software, people) to implement the business model.

Key Partners: marketing agencies responsible for the dissemination of the software products to attract customers; and government agencies responsible for promoting tourism for the disclosure of the software product through official channels.

Cost Structure: human resources necessary for the development, maintenance and administration of the business; hosting system for IT datacenter processing power.

Next a Vision of the system was produced from artifacts that were created at a previous stage to describe market analysis, the value curve and business modeling. Vision is described below.

"The software product should be a platform to support planning stages and performance in travel tourism, giving support to the business opportunities which occur in relation to these steps.

The niche market to be explored would be the interaction between tourists and small (off the map) suppliers (players); services such as rental of sports equipment and recreational vehicles; and lower cost hosting residences and hostel rooms, shared transportation, translators, independent tour guides, concierges, etc.

The software product should allow the sharing of experiences, information and recommendations among users. Furthermore, it should support the process of decision making as the events of the trip are conducted through personalized recommendations, proactively and not reactive to a question from the user, and according to the consumption profile and with context.

Finally, the software product should always seek to provide a good user experience, and be as simple as possible, should be mobile, developed in Java platform/ANDROID, based on geolocation and integrated with social media as well as following the freemium/premium pricing model so that most users use the application freely, funded by a minority with more specific needs for their business."

The product backlog (list of demands) was defined with the following items: (1) travel plan to schedule daily events planned; (2) checklist to control actions that must be performed before the trip (reservations, passport, vaccinations, etc.); (3) survival guide for registering local customs and other relevant information; (4) check-in to record the events held; (5) virtual tour guide to guide itinerary of tourist attractions; (6) assistance platform to connect supply and demand/convenience targeting the interaction between operators and travelers; and (7) integration with social media authentication and to allow access to user information from Facebook.

## 5.2 Selection of System Requirement for Tracing

The second step consists of demonstrating the causes of a system requirement according to the InnoTrace approach. To this end, we selected system requirement (4) "check in to record the events held" of the backlog product described in the previous step (A).

The selected system requirement permits tourists to check in to events in a variety of spots during a trip – for example, airports, restaurants, hotels, tourist attractions, etc.

## 5.3 Trace Specification

The third step consists of specifying a trace in accordance with the track of the InnoTrace approach. The direction of the trace is in the reverse direction, meaning system requirement in the direction of requisites extracted from innovation processes, to demonstrate the validation (cause) of the system requirement selected. Figure 9 presents system requirement trace (4) "check in to record the events held" through a Requirement Diagram, in accordance with the metamodel (Figure 5) specified by the InnoTrace approach.

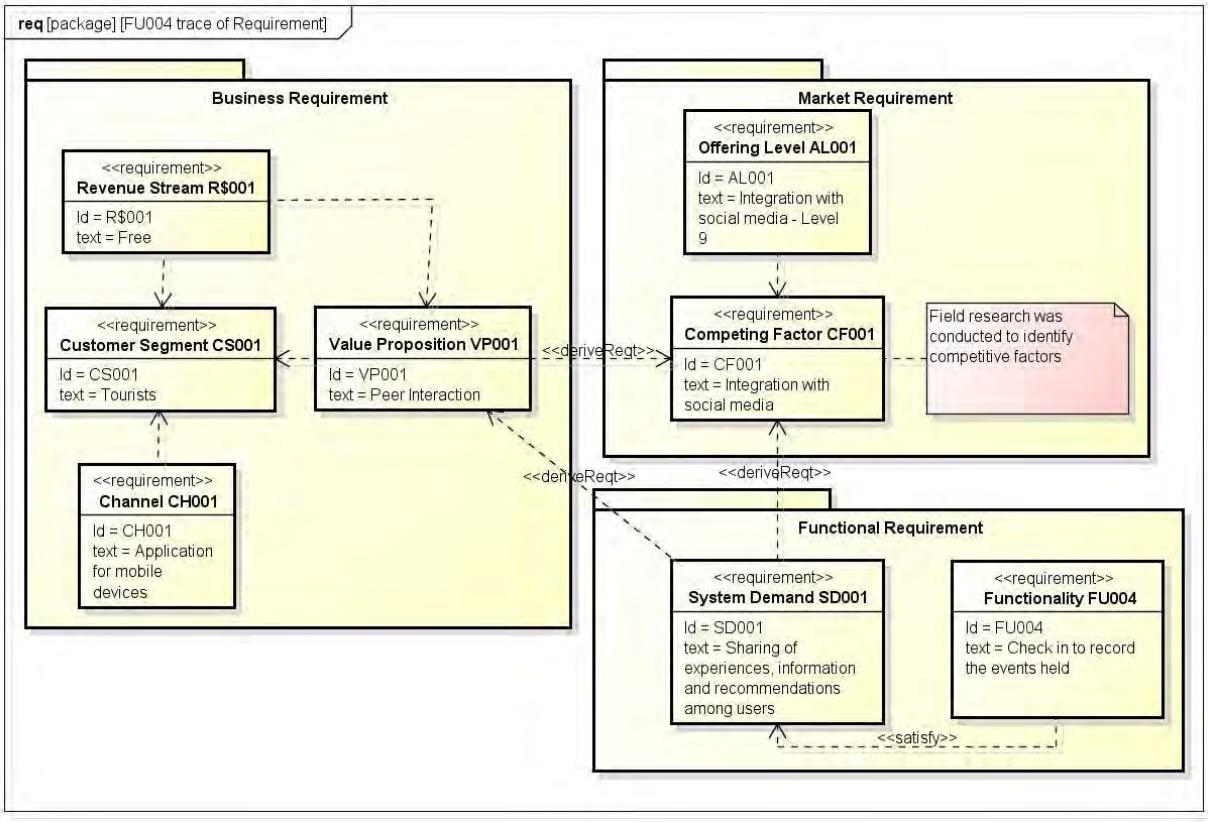


**Figure 9. Trace of the System Requirement**

## 5.4 Specification of the Tracing Matrix

The fourth step consists of specifying the tracing matrix for the selected system requirement.

Table 6 presents the tracing matrix for the system requirement (4) "check in to record the events held", with the objective of better explaining the existing relationships in the trace (Figure 9).

**Table 6. Tracing Matrix for the System Requirement**

| Source Element ID | Target Element ID | Relationship | Explanation |
|---|---|---|---|
| FU004 | SD001 | Satisfy | The functionality FU004 - "Check-in to record the events held" satisfies the system demand SD001 - "Sharing of experience, information and recommendations among users" |
| SD001 | CF001 | DeriveReqt | The system demand SD001 is derived from the competition factor CF001 - "Integration with social media" |
| SD001 | VP001 | DeriveReqt | The system demand SD001 is also derived from the value proposition VP001 - "Peer interaction" |
| AL001 | CF001 | Dependency | The offering level AL001 of the competition factor CF001 is 9 on a scale of 0 to 10 |
| VP001 | CF001 | DeriveReqt | The value proposition VP001 is derived from the competition factor CF001 |
| VP001 | CS001 | Dependency | The value proposition VP001 is specific for the customer segment CS001 - "Tourist" |
| CH001 | CS001 | Dependency | The channel CH001 - "Application for mobile devices" is appropriate for the use of the customer segment CS001 |
| R$001 | CS001 | Dependency | The return of investment R$001 - "Free" of the customer segment CS001 is non-monetary (reputation) |
| R$001 | VP001 | Dependency | The value proposition VP001 is freely delivered (R$001) to the customer segment CS001 |

## 5.5 Sources of the Signs

The next step of the demonstration case study of the InnoTrace approach consists of the identification of the source artifacts from which the exhibited signs were extracted in the trace of the system requirement.

Table 7 presents the types of requirements represented by signs and their respective identifiers and source artifacts for the trace of system requirement (4) "check in to record the events held" (Figure 9).

**Table 7. Artifact Source of the Signs**

| Requirement | Sign | ID | Source Artifact |
|---|---|---|---|
| Functional Requirement | Functionality | FU004 | Product Backlog |
| | System Demand | SD001 | Vision |
| Market Requirement | Offering Level | AL001 | Strategy Canvas (Value Curve) |
| | Competing Factor | CF001 | |
| Business Requirement | Value Proposition | VP001 | Business Model |
| | Customer Segment | CS001 | |
| | Channel | CH001 | |
| | Revenue Stream | R$001 | |

## 5.6 Discussion on Tracing

Tracing in the backward direction made possible the identification of the cause (origin) of the functionality FU004 - "Check in to record the events held ".

The use of the InnoTrace approach for the specification of tracing permits one to consider the following:

- The necessity for the system requirement was motivated by market and business requirements. The traditional software processes focus on user demands in the specification of system requirements;
- The market and business requirements were captured from artifacts produced through the specific purpose tools Strategy Canvas and Business Model Canvas, respectively. The traditional software processes use specific tools to capture only users' functional demands, such as user cases, user stories, vision, requirements document, amongst others;
- The internal relationships between signs of the market and business requirements permit understanding of competition and business strategies of the organization for the evaluated requirements. For the business requirements are presented: the value to be delivered, the target audience, the return of investment and the distribution channel. For the market requirement are presented: the competition factor and how the organization positions itself in relation to this factor (offering level) on a relative scale from 0 to 10;
- The relationships of the functional requirement with the market and business requirements permit understanding of the logic of how one of the functionalities of the software product contributes to the market and business objectives of the organization. In traditional software processes, generally the starting point for tracing are the functional requirements, thus there being difficulty in tracing motivations originating in necessities related to the market, business and creativity;
- The trace of the system requirement (Figure 9) presented through a Requirement Diagram of the SysML language. This visual mechanism facilitates the understanding of the tracing of a requirement. The Requirements Diagram is a more expressive graphic tool than a Traceability Matrix (Table 6), which is based on text.

# 6. CONCLUSION

The research problem engaged by this work was how to perform the tracing of requirements extracted from innovation processes towards system requirements and vice versa.

The main contribution of this work was to develop the InnoTrace approach which constitutes a proposed solution for the research problem formulated.

The InnoTrace approach is a method for tracing requirements in software development processes which use tools and techniques provided by the innovation-driven approaches: Design Thinking, Business Model Generation and Blue Ocean Strategy.

The InnoTrace approach is based on concepts of signs, track and trace defined by the requirements traceability framework proposed by Gotel and Morris [18].

The InnoTrace approach consists of the specification of signs which represent requirements extracted from innovation and software processes; the specification of a track (metamodel) which describes the relationships between signs; and the evaluation of traces from the specified track.

A sign is a mark extracted from an artifact which shows a requirement. The signs of the InnoTrace approach are extracted from artifacts produced by way of tools appropriate for specifying aspects related to: business, market, creative solving-problems, and user.

A track defines the possible relationships between the signs. The track of the InnoTrace approach presents a metamodel with the semantic relationships between the signs. The track is specified through the Diagram of Requisites of the SysML language [21].

A trace consists of following a trace sign by sign. For the evaluation of a trace, from the specified track in the InnoTrace approach, a case study was used [13] which incorporates tools provided by innovation-driven approaches referred to in this work.

Among the questions which can be more easily answered by the InnoTrace approach are: Which are the business requirements, market requirements and creativity requirements related to a system requirement? How does software functionality contribute to the business and market strategy of an organization? How do cause/effect relations of requirement traceability become graphically visible?

The additional contributions of this work can be enumerated as the following: (1) provide means for the treatment of aspects related to business modeling, market analysis, and creativity in the conception of software products; (2) establish relationships between artifacts of innovation processes and artifacts of software processes; (3) promote innovation as a primary aspect in software development; and (4) provide a means for validation and verification of system requirements derived from requirements extracted from innovation processes.

One of the limitations of this work is that the specification of signs, which represent requirements extracted from innovation processes, and of the track, which represents the relationships between signs, need to be better evaluated empirically as to correctness and completeness.

There are still more innovation approaches which provide tools and techniques which can be utilized in the context of software development, such as Lean Startup [7], Lean Thinking [29] and Disruptive Innovation [30].

Another limitation of this work is the absence of a software tool to give support to the InnoTrace approach, seeing that to maintain the relationships of tracing between signs and requirements manually is a tiring task and subject to errors. This limitation spurs us on to another opportunity for future work.

## 7. REFERENCES


[1] Gartner. 2020. *Gartner Says Worldwide IT Spending to Grow 4% in 2021*. Gartner. Available at: <https://www.gartner.com/en/newsroom/press-releases/2020-10-20-gartner-says-worldwide-it-spending-to-grow-4-percent-in-2021>. Accessed on February 15, 2021.

[2] Pressman, R. S. 2014. *Software Engineering: A Practitioner's Approach*. 8ª. ed. [S.l.]: McGraw-Hill Higher Education.

[3] ISO. 2008. ISO 9001:2008. *International Organization for Standardization*. Quality Management System [S.l.].

[4] Sommerville, I. 2015. *Software Engineering*. 10ª. ed. [S.l.]: Pearson.

[5] Kotonya, G. and Sommerville, I. 1998. *Requirements engineering: processes and techniques*. Chichester, England: John Wiley.

[6] Berg, K.; Bishop, J.; Muthig, D. 2005. Tracing software product line variability: from problem to solution space. In *Proceedings of the 2005 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on IT Research in Developing Counstries*, White River, South Africa, v. 150, p. 182-191, 20-22.

[7] Ries, E. 2011. *The Lean Startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. [S.l.]: Crown Business.

[8] Osterwalder, A.; Pigneur, Y. 2010. *Businesss Model Generation*. [S.l.]: John Wiley & Sons.

[9] BOS. 2021. *Blue Ocean Strategy*. Blue Ocean Strategy. Available at: <http://www.blueoceanstrategy.com/>. Accessed on January 16, 2021.

[10] Kim, C.; Mauborgne, R. 2005. *Blue Ocean Strategy: how to create uncontested market space and make competition irrelevant*. Rio de Janeiro: Harvard Business School Press.

[11] Brown, T. 2009. *Change by Design: how Design Thinking transforms organizations and inspires innovation*. [S.l.]: HaperCollins.

[12] Frye, U.; Inge, T. 2013. *The Integration of Design Thinking and Lean Software Development from the Perspective of Product Owners and Scrum Masters*. Master's Thesis: CHALMERS UNIVERSITY OF TECHNOLOGY. Göteborg, Sweden.

[13] Souza, R. A. C.; Cysneiros Filho, G. A. D. A.; Batista, G. H. C. 2015. A Heuristic Approach for Supporting Innovation in Requirements Engineering. In *18ª Workshop em Ingeniería de Requisitos (WER)*. Lima, Peru: [s.n.].

[14] Beyhl, T.; Berg, G.; Giese, H. 2013. *Why Innovation Processes Need to Support Traceability*. TEFSE. USA: San Francisco: [s.n.].

[15] Andersson, M.; Lindgren, R.; Henfridsson, O. Architectural knowledge in interorganizational IT Innovation. 2008. *Journal of Strategic Information Systems* 17 (1), p. 19–38.

[16] Koberg, C.; Detienne, D.; Heppard, K. 2003. An empirical test of environmental, organizational, and process factors affecting incremental and radical innovation. *Journal of High Technology Management Research* 14 (1), 21–45.

[17] Gotel, O.; Finkelstein, A. 1994. An analysis of the requirements traceability problem. In *Proceedings of the 1st International Conference on Requirements Engineering*. [S.l.]: [s.n.]. p. 94-101.

[18] Gotel, O. C. Z.; Morris, S. J. 2011. Out of the Labyrinth: Leveraging Other Disciplines for Requirements Traceability. In *19th International Requirements Engineering Conference*. Trento, Italy: IEEE. p. 121-130.

[19] SysML.org. 2015. *SysML Open Source Specification Project*. SysML.org. Available at: <http://sysml.org/>. Accessed May 25, 2015.

[20] Friedenthal, S.; Moore, A.; Steiner, R. 2014. *A Practical Guide to SysML: The Systems Modeling Language*. 3rd Ed. [S.l.]: Morgan Kaufmann. ISBN 0128008008, 9780128008003.

[21] OMG. 2006. *Introduction to SysML*. OMG.org. Available at: <http://www.omg.org/news/meetings/workshops/Real-time_WS_Final_Presentations_2008/Tutorials/00-T2_Moreland.pdf>. Accessed May 12, 2015.

[22] Maiden, N., Jones, S., Karlsen, I. K., Neill, R., Zachos, K. and Milne, A. 2010. A Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding. In *18th IEEE International Requirements Engineering Conference*. [S.l.]: [s.n.]. p. 57-66.

[23] Osborn, A. F. 1953. *Applied Imagination: Principles and Procedures of Creative Problem Solving*. New York: Charles Scribener's Sons.

[24] Schwaber, K.; Sutherland, J. 2013. *The SCRUM Guide: The Rules of the Game*. [S.l.].

[25] OMG. 2021. *Unified Modeling Language (UML)*. UML.ORG. Available at: <http://www.uml.org/>. Accessed on April 18, 2021.

[26] Astah. 2015. *Astah SysML Tutorial*. Astah. Available at: <http://astah.net/tutorials/sysml/requirement-diagram>. Accessed May 15, 2015.

[27] Osterwalder, A.; Pigneur, Y.; Tucci, C. L. 2005. Clarifying business models: Origins, present, and future of the concept. *Communications of the association for Information Systems*, v. 16, n. 1, p. 1-25.

[28] Neumeier, M. 2009. *The Designful Company: how to build a culture of nonstop innovation*. [S.l.]: New Riders.

[29] Womack, J. P.; Jones, D. T. 2013. *Lean Thinking: Banish Waste And Create Wealth In Your Corporation*. [S.l.]: Simon and Schuster.

[30] Christensen, C. 2013. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. [S.l.]: Harvard Business Press.