



## Collaborative Approach for Feature Models in Software Product Lines

### Enfoque colaborativo para modelos de características en líneas de productos de software

 Jazmín Gómez<sup>1</sup>;  
  Pablo H. Ruiz<sup>2</sup>;  
 Vanessa Agredo Delgado<sup>3</sup>;  
 Marta Cecilia Camacho<sup>4</sup>

<sup>1</sup>Corporación Universitaria Comfacaucua, Popayán-Colombia,  
[jgomez@unicomfacaucua.edu.co](mailto:jgomez@unicomfacaucua.edu.co)

<sup>2</sup>Universidad Nacional Abierta y a Distancia, Popayán-Colombia,  
[pablo.ruiz@unad.edu.co](mailto:pablo.ruiz@unad.edu.co)

<sup>3</sup>Universidad del Cauca, Corporación Universitaria Comfacaucua,  
Popayán-Colombia,  
[vanessaagredo@unicauca.edu.co](mailto:vanessaagredo@unicauca.edu.co)

<sup>4</sup>Institución Universitaria Colegio Mayor del Cauca, Popayán-Colombia,  
[cecamacho@unimayor.edu.co](mailto:cecamacho@unimayor.edu.co)

---

#### How to cite / Cómo citar

J. Gómez, P. H. Ruiz, V. Agredo Delgado, and M. C. Camacho,  
“Collaborative Approach for Feature Models in Software Product Lines,”  
*Tecnológicas*, vol. 27, no. 60, e3001, 2024.  
<https://doi.org/10.22430/22565337.3001>

---

**Abstract**

One of the strategies that help to software reuse are the Software Product Lines (SPL), which are a set of products developed from common and variable features that meet specific needs of a domain. In this sense, feature models are a key tool to manage common features, variability, and customization of the line products; however, their definition is a complex task that requires the participation of a multidisciplinary team. Therefore, to achieve their definition, it is crucial to establish clear guidelines for communication and collaboration among stakeholders. The lack of effective collaboration may result in a poor definition of the model since it is a fundamental component for the construction of an SPL. This paper aims to present CINDERELLA, a collaborative approach to define feature models in SPLs, and to show its initial evaluation. Evaluation was carried out by defining an experiment in an academic environment. The experiment revealed that the students had a positive perception of CINDERELLA, highlighting its usefulness and completeness, although the clarity of its instructions needs to be improved. CINDERELLA is perceived as a user-friendly, useful, and complete approach to define feature models, because of its consistency and organization. However, its description needs to be improved and additional experiments in real contexts are required to confirm its applicability and effectiveness.

**Keywords**

Software product lines, collaborative work, features model, SPL scope definition, reuse of software.

**Resumen**

Una de las estrategias de ayuda a la reutilización de software son las Líneas de Productos de Software (LPS), las cuales son un conjunto de productos desarrollados a partir de características comunes y variables que satisfacen necesidades específicas de un dominio. En este sentido, los modelos de características son una herramienta clave para gestionar características comunes, variabilidad y personalización de los productos de la línea; sin embargo, su definición es una tarea compleja que requiere la participación de un equipo multidisciplinario. Por lo tanto, para lograr su definición, es crucial establecer directrices claras de comunicación y colaboración entre actores involucrados. La falta de colaboración efectiva puede resultar en una definición deficiente del modelo, debido a que es un componente fundamental para la construcción de una LPS. Este artículo tuvo como objetivo presentar CINDERELLA, un enfoque colaborativo para definir modelos de características en LPS, y mostrar su evaluación inicial. La evaluación se hizo mediante la definición de un experimento en un entorno académico. El experimento reveló que los estudiantes tuvieron una percepción positiva de CINDERELLA, destacando su utilidad e integridad, aunque se necesita mejorar la claridad de sus instrucciones. CINDERELLA es percibido como un enfoque fácil de usar, útil y completo para definir modelos de características, gracias a su coherencia y organización. Sin embargo, se requiere mejorar su descripción y realizar experimentos adicionales en contextos reales para confirmar su aplicabilidad y efectividad.

**Palabras clave**

Líneas de productos de software, trabajo colaborativo, modelo de características, definición del alcance en LPS, reutilización de software.

## 1. INTRODUCTION

Software engineering practitioners reuse existing software elements in new developments instead of building them from scratch. In this sense, one of the existing strategies that helps in software reuse is Software Product Lines (SPLs), which focus on creating families of related products from a common set of software assets. Rather than developing each product individually, SPLs leverage the similarities between all products to optimize the development process and reduce production time and cost [1], [2]; they identify variability between products by defining mandatory, common, and optional features. These features refer to a functionality or perceptible behavior of a software product for the end user. Considering this, during SPL development, these features specify and convey the commonalities and differences between products [3]. They also guide the structure, reusability, and adaptability in the stages called domain engineering and application engineering, which are part of the SPL creation cycle [4]. In this sense, during the domain engineering stage, teams must define feature models, which are fundamental in SPL development and management. These models represent the variabilities and relationships between different features that products may have. Teams represent features as nodes, and they represent inclusion, exclusion, or dependency between these features as arcs between the nodes. However, defining feature models is complex [5], [6]. Teams face the first complexity of visualizing a set of products instead of a single product [2]. Since several people with different ideas and knowledge are involved, teams need to properly understand and interpret various opinions on the model's needs that should emerge from the modeling team [7]. The team involved in defining the scope of a product line must identify, classify, associate, and negotiate the features of a set of products. Scoping a complete set of products requires participation from several business units with their own objectives and points of view, which presents a challenge [7]. To address this, during the identification of characteristics, teams need to abstract and understand the domain knowledge of various experts, as well as seek information from different sources such as books, user manuals, design documents, and source programs [8]. They also seek to understand the scope of the domain, the intended use of domain products, and the various external conditions and common domain vocabularies.

Researchers focus on the automatic analysis of the feature model because this task is prone to errors due to the considerable number of characteristics and relationships that must be considered. They have identified the difficulty in manually creating large-scale feature models but have not considered the implications or need for diverse participants in this modeling, even though they have deemed this factor important in specifying the scope of the line, an activity that includes feature modeling [6], [7], [9], [10]. To define the feature model, teams must combine all these aspects, making the activity complex and requiring adequate collaboration between people who use techniques to manage and elicit sources of information. If these aspects are not well-related, they can lead to a poorly defined model and render it useless [9]. Therefore, establishing clear guidelines for communication and collaboration between the different actors involved is crucial for a successful model definition. This paper presents CINDERELLA (Spanish acronym for **definiCIÓN de moDElos de caRacterísticas en SPL based on coLLaborative work**) as an approach for defining feature models based on collaborative work. This paper extends the work originally presented [10], but it differs by showing in more detail the elements of the CINDERELLA definition as an approach to defining feature models that relies on collaborative work. CINDERELLA defines elements such as workflows, tasks, collaborative patterns associated with Thinklets or Gamestorming, roles, and input and output artifacts to facilitate its application. Additionally, it presents the initial evaluation of CINDERELLA, conducted in an academic environment, which revealed

that users find it easy to use due to the coherence and organization of its elements. Users also find it complete, as it provides all necessary elements for defining feature models, and useful because the resulting model can be used in various stages of the SPL development process.

We organize this paper as follows: In Section 2, we briefly describe the related work. In Section 3, we describe the methodology used for the execution of the research. In Section 4, we describe part of the CINDERELLA approach. In Section 5, we show the validation of CINDERELLA, and finally, in Section 6, we present the conclusions.

## **2. RELATED WORK**

### **2.1 Background of feature model**

In [11] first introduced the Feature Model as part of Feature-Oriented Domain Analysis (FODA), a method that identifies salient or distinctive characteristics of software systems resulting from domain analysis. In [12] was proposed to integrate the feature modeling of feature-oriented domain analysis with the processes and work products of reuse-driven software engineering business (RSEB). Their proposal provides an effective reuse-oriented model as a "catalog" that links use cases, variation points, reusable components, and configured applications.

Researchers widely use the features model to characterize and configure a SPL [13]. Several proposals aim to facilitate constructing the feature model, such as the semi-automatic approach to build feature models based on grouping requirements [14] and the algorithmic and parameterizable approach to computing a logical and appropriate hierarchy of features [13]. Other proposals test the models and identify possible errors [15] and suggest automated analysis of feature models [16]. Some works indicate aspects to consider when creating relationships and modeling features, such as restrictions, dependencies, and incompatibilities [17], [18]. Additionally, researchers have proposed tools for creating feature models, such as the tool using web forms [19], the reasoning and configuration system for feature models called S.P.L.O.T [20], the automated approach to building feature models from product descriptions available in repositories [21], and the specification approach of feature models using automatically generated user interfaces and encoding the models in a relational database [22].

Researchers use a wide variety of terms to refer to the process of building or defining a feature model [13]. The most widespread technique for modeling features, originally presented in FODA, uses a graphical notation similar to a tree to show the hierarchical organization of features. The tree's root represents the entire SPL node, and the other nodes represent different types of features [11]. Some proposals aim to facilitate defining feature models by automating feature extraction [23].

When creating the features model, stakeholders select and deselect features, corresponding to a set of decisions they must make. Typically, several stakeholders participate, including business leads, IT leads, and project managers, who base their decisions on their knowledge and experience. However, some decisions about which features to include, their types, or relationships made by these stakeholders can create conflicts between product configurations and business needs [24].

Most of the approaches proposed in the literature focus on searching for techniques or tools that facilitate defining a feature model, recognizing the complexity involved and the

importance of specifying an SPL. However, the literature reviewed does not define or materialize an approach that interacts with stakeholders or is based on collaborative work.

## 2.2 Collaborative work in *SPL*'s

Some proposals have considered incorporating collaborative work into practices or methods of engineering software product lines. In [25], researchers propose a support approach for coordinating teamwork decision making in the context of product configuration. This approach is based on configuring feature models, highlighting that misconfigurations can lead to the production of invalid product specifications. Regarding collaborative work in defining the scope of SPLs, [26], [27] propose a collaborative process for scope definition in SPLs, aiming to balance agility with the intrinsic needs of this activity within the SPL approach. Additionally, an exploratory study in [28], [29] identifies issues related to collaborative work in scoping from a practical perspective. Another collaborative approach to scoping combines scoping practices with collaboration patterns and ThinkLets, aiming for effective participation of required roles in the scoping activity. An alternative approach presented in [30], aims to establish a Collaborative Software Product Line Engineering Lab, accessible to all SPL communities to support their development and adoption in both industry and academia.

These works present various perspectives on feature models and approaches using collaborative work within the SPL framework. They offer different views and strategies on creating, manipulating, and managing feature models and integrating collaborative engineering in SPL contexts. However, none of these approaches include or are based on collaborative work specifically to build feature models. Therefore, none utilize elements of collaborative engineering that specifically support feature model creation, which is the primary contribution of this paper.

## 3. METHODOLOGY

This work followed the multi-cycle action research methodology, specifically adopting the approach to guide the use of action research in distributed research projects [31]. The research cycle included problem diagnosis, action intervention, and reflective learning.

- Problem diagnosis: During this phase, systematic mapping was planned and executed to identify characteristics of various approaches in creating SPL feature models. A characterization of these approaches supported the definition of CINDERELLA. Literature review revealed challenges requiring integration of knowledge and experiences of different individuals without adequate collaboration techniques and evaluation of contributions.
- Action intervention: The basic structure of elements comprising CINDERELLA was defined and incrementally incorporated during this phase. Collaborative work patterns suitable for SPL contexts were identified and integrated based on expert evaluation. Three experts evaluated CINDERELLA during formulation: an expert in product lines, an expert in collaborative work, and a researcher with experience in software development projects in southwestern Colombia. They assessed the utility of CINDERELLA tasks in defining product lines and suggested enhancements, such as integrating tools for asynchronous collaboration and analyzing feature and model correctness.

- **Reflective learning:** An experiment was conducted to evaluate CINDERELLA's usefulness, ease of use, and completeness in defining feature models from the participants' perspective. The evaluation also assessed the level of collaboration achieved by the working group.

## 4. CINDERELLA

This section presents a proposal for an approach to define feature models in SPL based on collaborative work called CINDERELLA.

CINDERELLA is an approach that applies collaboration patterns to most tasks necessary for creating the SPL feature model. These patterns aim to enhance the contribution of participating roles, thereby ensuring the feature model's completeness, correctness, and utility. The approach systematically guides the definition of feature models using a coherent flow of tasks, roles, and artifacts. This approach establishes a structured basis that fosters communication and collaboration among the different roles that are part of the approach. CINDERELLA targets small or medium-sized companies aiming to adopt software product line approaches, starting from an existing product to expand into a product set for an identified market niche.

Feature modeling is a crucial practice in product line development, involving the identification, classification, and association of product set characteristics. Addressing the complexity of variability in emerging application domains, as well as limitations, potential errors, and the need for model analysis, requires a collaborative approach involving diverse knowledge and participants. This approach proposes methods to enable various contributors to define features and their relationships, which is essential for SPL development.

### 4.1 CINDERELLA a collaborative approach

CINDERELLA consists of eleven tasks detailed to guide their execution and achieve the objectives set by the assigned roles. For each task, appropriate collaboration patterns have been identified based on the teamwork required and the objective of each group activity [32]. The tasks comprise steps or actions adapted using ThinkLets or Gamestorming to enhance collaboration among roles and guide task development. ThinkLets are reusable units for defining predictable and repeatable teamwork tasks [33], while Gamestorming proposes game-based techniques to foster interactive and creative activities among team members [34]. CINDERELLA was developed using the Collaborative Engineering approach to design [35]. Table 1 illustrates the relationships between tasks necessary for creating a feature model and the collaborative elements integrated into the proposed approach. Table 2 outlines the input and output artifacts for each task specified by the approach.

The feature model definition team comprises the following roles: Project Leader (PL), Domain Expert (DE), Software Architect (SA), Marketing Expert (ME), Collaborative Work Advisor (CA), Potential Customer (PC), Sales Person (SP), SPL Expert (SE), Technical Expert (TE), Domain Analyst (DA), and Company Product Manager (CP), each bringing necessary knowledge and experience. CINDERELLA specifies for each role whether their participation is mandatory or optional. Table 3 illustrates the roles' participation in each task proposed by CINDERELLA, indicating whether their involvement is mandatory, optional, or not applicable if a role does not participate in a particular task.

**Table 1.** Relationships between tasks and collaborative elements. Source: own elaboration.

No.	Task	Collaborative pattern	ThinkLets/ Gamestorming
1	Contextualize Concepts	Clarify	Visual Glossary
2	Identify the SPL Domain	Reduce	Dot Voting
3	Disclose Existing Products	Not Applicable	Not Applicable
4	Explore Similar Products	Not Applicable	Not Applicable
5	Propose Features	Generate	FreeBrainstorm
6	Analyze Features	Reduce	GarlicSqueezer
7	Evaluate Features	Reduce	StrawPoll
8	Define the variability of the characteristics	Reduce	Dot Voting
9	Formalize the FM	Not Applicable	Not Applicable
10	Validate the FM	Reduce	Dot Voting
11	Socialize the FM	Not Applicable	Not Applicable

**Table 2.** Cinderella Tasks, Artifacts, and Roles. Source: own elaboration.

No.	Task	Input artifacts	Output artifacts
1	Contextualize Concepts	SPL and FM documentation	Time of attendance
2	Identify the SPL Domain	Business Goals	SPL Domain
3	Disclose Existing Products	Base Product Documents Base products	Preliminary feature list
4	Explore Similar Products	Similar External Product Documents Similar external products	Preliminary feature list
5	Propose Features	Preliminary feature list	Feature List
6	Analyze Features	Feature List	Feature List Analyzed
7	Evaluate Features	Feature List Analyzed	List of evaluated features
8	Define the variability of the characteristics	List of evaluated features	Sketch of the FM diagram
9	Formalize the FM	FM Diagram Sketch	Final FM diagram
10	Validate the FM	Final FM diagram	Validated FM diagram
11	Socialize the FM	Validated FM diagram	FM delivery certificate

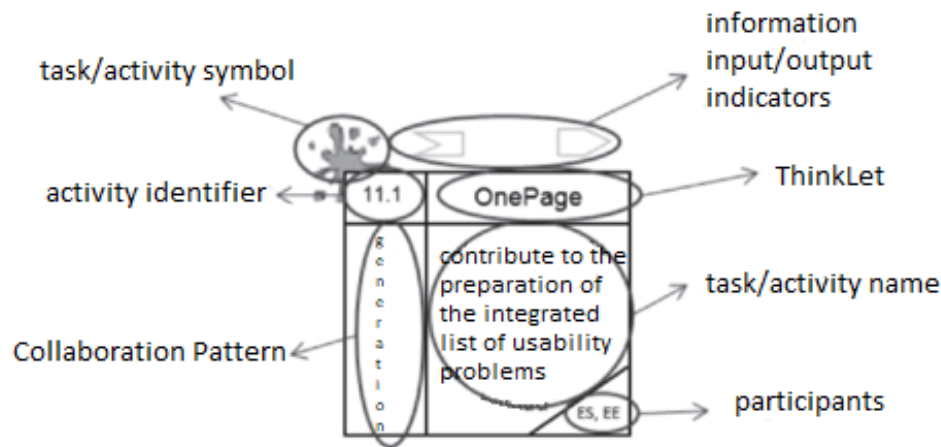
**Table 3.** Mandatory or Optional Roles for each task. Source: own elaboration.

Task	LP	DE	AS	EM	AC	CP	SP	ES	ET	CM	DA
Contextualize Concepts	M	M	M	M	M	M	M	M	NA	M	NA
Identify the SPL Domain	M	O	M	M	M	M	O	O	O	M	NA
Disclose Existing Products	M	M	M	M	O	O	O	O	O	M	M
Explore Similar Products	M	M	M	M	O	O	O	O	O	M	M
Propose Features	M	M	M	M	M	O	O	O	O	O	NA
Analyze Features	M	M	M	M	O	O	O	O	O	O	O
Evaluate Features	M	M	M	M	O	O	O	O	O	O	O
Define the variability of the characteristics	M	M	M	O	O	O	O	O	O	O	O
Formalize the FM	M	NA	NA	NA	NA	NA	NA	NA	NA	NA	M
Validate the FM	M	M	M	M	O	O	O	NA	NA	O	NA
Socialize the FM	M	M	M	M	M	M	M	M	M	M	M

## 4.2 CINDERELLA graphic specification

The specification of CINDERELLA utilized the extension of HAMSTERS (Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems). This notation allows graphical definition of relationships and representation of information between tasks and their participants (roles). The extended notation includes representation of collaborative elements involved in the tasks, such as participating roles, collaboration patterns, associated ThinkLets, and task steps [36].

Each task's description is depicted using the HAMSTERS extension. The model is represented by a figure or graphic, where the task is symbolized by a rectangle composed of five fields: the task identifier in the upper left, the associated ThinkLets or Gamestorming in the upper right, the main collaboration pattern on the left, the task name in the largest field, and the acronym of participants or mandatory roles in the lower right triangle, formed by the first letters of each role's name (see Figure 1).

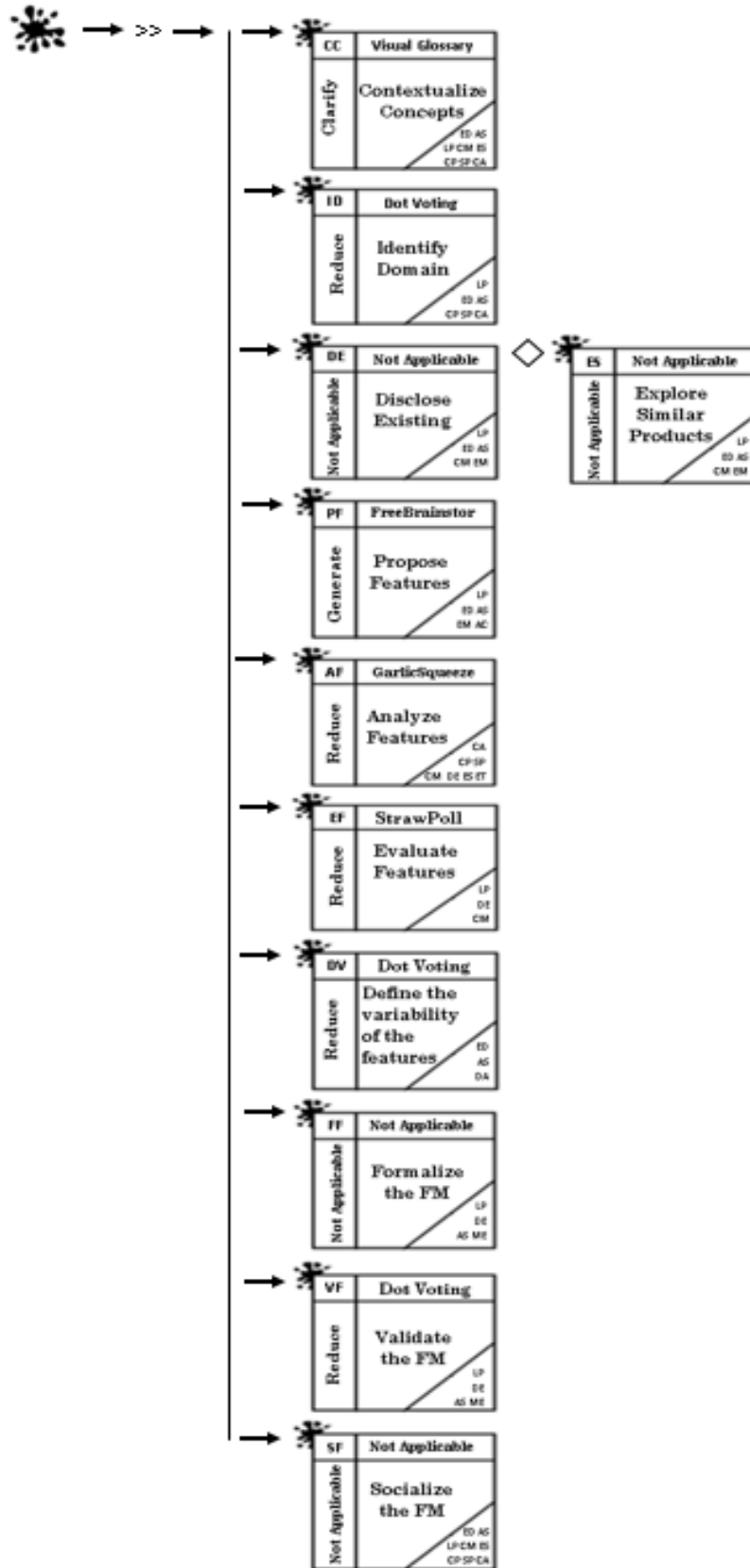


**Figure 1.** Collaborative task description used in CINDERELLA. Source: [36].

CINDERELLA comprises eleven tasks: Contextualize concepts, Identify the SPL domain, disclose existing products, explore similar products, propose features, analyze features, evaluate features, define feature variability, Formalize the feature model, Validate the feature model, and finally Socialize the feature model. CINDERELLA outlines a workflow to simplify following and guide the sequence of required work. Figure 2 illustrates a comprehensive view of the proposal and its workflow.

The description of each task specifies the steps required for completion. The HAMSTERS extension incorporates graphical elements to represent the type of step to be performed. This notation indicates whether the step involves collaboration and specifies the type of collaboration. Table 4 illustrates the graphical notation and types of steps.



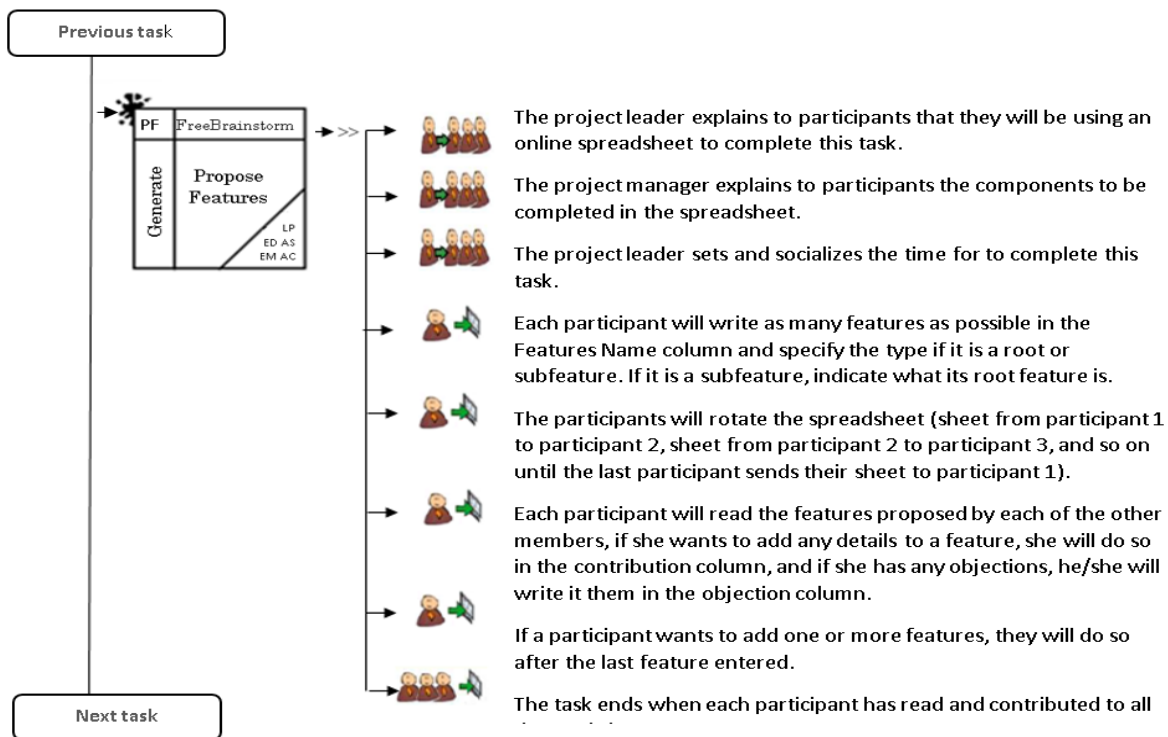


**Figure 2.** Task Flow of CINDERELLA. Source: own elaboration.

**Table 4.** Graphical representation of the steps to be performed in a task. Source: adapted from [36].

No.	Symbol	Step type
1		Cognitive analysis (non-collaborative)
2		Cognitive analysis (collaborative)
3		Share information
4		Analysis or decision making
5		Input data to system (No collaborative)
6		Input data to system (collaborative)

This paper presents one of the tasks from CINDERELLA. The fifth task in our proposal is called "Propose features." Figure 3 presents the model for this task, and Table 5 corresponds to the task description table.

**Figure 3.** The description model of task proposes features. Source: own elaboration.

**Table 5.** The description table of task Propose features. Source: own elaboration.

Task	Propose features
Description	The objective of this task is to identify the features that will comprise the product line through a brainstorming session, where each team member proposes a maximum number of features, considering similar products and the SPL domain.
Mandatory Roles	Project Leader Domain Expert Software Architect Marketing Expert Collaborative Work Consultant Business Administrator
Optional Roles	SPL Expert Potential Customer Sales personnel technical expert
Collaborative Pattern	Generate
ThinkLet/ Gamestorming	FreeBrainstorm
Input artifacts	Preliminary feature list
Output artifacts	Feature List
Steps	<ol style="list-style-type: none"> <li>1. The project leader informs participants that they will use an online spreadsheet for this task.</li> <li>2. The project manager explains the components to be completed in the spreadsheet.</li> <li>3. The project leader schedules and communicates the deadline for completing this task.</li> <li>4. Each participant writes as many features as possible in the "Feature Name" column and specifies whether it is a root or subfeature. If it is a subfeature, they indicate its root feature.</li> <li>5. Participants rotate the spreadsheet (Participant 1's sheet goes to Participant 2, Participant 2's sheet goes to Participant 3, and so on until the last participant sends their sheet to Participant 1).</li> <li>6. Each participant reviews the features proposed by other members. If they wish to add details to a feature, they do so in the "Contribution" column. If they have objections, they write them in the "Objection" column.</li> <li>7. If a participant wants to add one or more features after the last feature entered, they do so accordingly.</li> <li>8. The task concludes once each participant has reviewed and contributed to all spreadsheets</li> </ol>
Rules	No member may eliminate features proposed by others.

## 5. VALIDATION OF THE CINDERELLA PROPOSAL

To validate CINDERELLA, we applied this approach in an educational context to evaluate the usefulness, ease of use, and completeness of defining feature models in SPLs from the perspective of the experimental participants.

### 5.1 Experiment context

The experiment involved 15 students from the Software Engineering II course of the VI semester of the Systems Engineering program at Corporación Universitaria Comfacaucá (UNICOMFACUCA), located in Popayán, Colombia. Specifically, the students actively participated in the experiment, which provided a controlled execution environment for CINDERELLA. The experiment focused on defining a feature model for the project titled

“System for Monitoring and Management of Traffic and Irregularities in the Streets of the City of Popayan”. This project aimed to develop a solution for identifying, analyzing, and managing vehicular traffic and irregularities in the city's road infrastructure. The system intended to utilize real-time monitoring technologies, data analysis, and reporting tools to enhance urban mobility and road safety.

The Table 6 presents the profiles of the students who participated in the academic experiment. This group represents a spectrum of expertise in different fields of software engineering, providing an overview of the participants' capabilities.

**Table 6.** Students' profiles. Source: own elaboration.

Student	Age	Skills
1	20	Programming in Java and Python, SQL databases
2	21	Programming in Python, data analysis, machine learning
3	19	Project management with Scrum, software process
4	24	Network configuration, web application security
5	20	Android application development, UX/UI design with Figma
6	26	Programming in C/C++, IoT platforms like Arduino and Raspberry Pi
7	22	SQL database administration
8	23	Software architecture design, design patterns, UML modeling
9	25	Development in Unity, programming in C#
10	21	Services in AWS and Azure, Docker and Kubernetes
11	22	Requirements analysis and documentation, elicitation techniques
12	23	Programming in JavaScript and Node.js,
13	24	Robot programming, reuse of software in robots
14	22	Usability testing, user experience research, HCI methodologies
15	18	Cryptographic, programming languages

## 5.2 Experiment context

Table 7 summarizes the activities planned for the development of the experiment, along with the estimated time for their execution and the tools to support their development.

**Table 7.** Summary of experiment activities. Source: own elaboration.

Experimentation Activities	Duration	Support instruments
1. Socialize and contextualize the academic experiment.	20 minutes	Power Point presentation of the Introduction to the academic experiment and conceptual elements.
2. Present the CINDERELLA approach: task flow to create or define SPL Feature Model, based on Collaborative work.	30 minutes	Power Point presentation and document with the description of the approach.
3. Apply the proposed approach.	150 minutes	Guidance document, online spreadsheets, diagram of tasks to execute the approach.
4. Resolve questionnaire	15 minutes	Survey.
<b>Total time:</b> 3 hours 35 minutes		

### 5.3 Hypotheses

To evaluate the ease of use, usefulness, and completeness of CINDERELLA from the perspective of the group of UNICOMFACAUCA students, we evaluated the following hypotheses (see Table 8).

**Table 8.** Hypotheses of the experiment. Source: own elaboration.

Hypotheses	Variables
H.1.1 Users understand the instructions and guidelines of the approach.	Ease of use
H.1.2 Users understand the supporting examples provided by the approach.	
H.2.1 Users perceive that the approach provides the necessary information to guide its application.	
H.3.1 Users perceive that the approach is useful in the process of defining SPL Feature Models.	Utility
H.3.2 Users perceive the approach as organized and consistent.	
H.4.1 Users perceive that the elements of the approach are necessary and sufficient for the definition of a feature model.	Completeness

### 5.4 Execution

The following provides a general description of how the tasks for validating CINDERELLA were conducted.

Activity 1: Initially, participants were introduced to and familiarized with the experiment. They were also shown and instructed on how to carry out the activities planned for the experiment.

Activity 2: A comprehensive presentation of CINDERELLA was delivered to explain its structure, guide its application using supporting materials, and clarify key concepts used in its description.

Activity 3: Participants in the experiment applied CINDERELLA by following the guide for defining SPL feature models. As part of this activity, participants completed the artifacts defined by CINDERELLA for each task.

Activity 4: In the final activity, students completed a survey that included questions about the usability, usefulness, and completeness of the approach.

### 5.5 Qualitative analysis and results

Qualitative analysis involved studying the survey responses using the Likert scale, a measurement method for assessing attitudes and gauging agreement levels with a set of statements.

The scale used for survey responses is structured as follows.

- Value 1: Strongly disagree.
- Value 2: Disagree.
- Value 3: Neither agree nor disagree.
- Value 4: Agree.
- Value 5: Strongly agree.

Based on the hypotheses outlined in Table 8, the following null hypotheses were formulated:

- H.1.1,  $\pi_1 \leq 60\%$ , where  $\pi_1$  represents the percentage evaluating the ease of understanding of CINDERELLA instructions and guidelines.
- H.1.2,  $\pi_2 \leq 60\%$ , where  $\pi_2$  represents the percentage assessing the ease of understanding of CINDERELLA examples.
- H.2.1,  $\pi_3 \leq 60\%$ , where  $\pi_3$  represents the percentage evaluating whether CINDERELLA contains the necessary information for its application.
- H.3.1,  $\pi_4 \leq 60\%$ , where  $\pi_4$  represents the percentage assessing the perceived usefulness of CINDERELLA in defining an SPL Feature Model.
- H.3.2  $\pi_5 \leq 60\%$ , where  $\pi_5$  represents the percentage assessing the perception that CINDERELLA is organized and consistent.
- H.4.1,  $\pi_7 \leq 60\%$ , where  $\pi_7$  where  $\pi_7$  represents the percentage evaluating the completeness of CINDERELLA elements.

The hypotheses were validated using the results obtained from the survey conducted among the participants of the experiment. The detailed results are presented in Tables 9, 10, and 11.

**Table 9.** Results on perceived ease of use. Source: own elaboration.

Hypothesis	Questions	Strongly agree (%)	Agree (%)	Neither agree nor disagree (%)	Disagree (%)	Strongly disagree (%)	Validation
H1.1	Question 1	16.7	66.7	8.3	8.3	0.0	Rejected
	Question 3	8.3	41.7	33.3	16.7	0.0	
H1.2	Question 2	25.0	66.7	0.0	8.3	0.0	Accepted
	Question 5	8.3	58.3	33.3	0.0	0.0	
H2.1	Question 4	16.7	58.3	16.7	8.3	0.0	Accepted
	Question 6	8.3	58.3	16.7	16.7	0.0	

For the ease-of-use variable, the percentage of students' perception regarding the combined percentages of "agree" and "strongly agree," based on questions 1, 2, 3, 4, 5, and 6, is 83.4 %, 91.7 %, 50.0 %, 75.0 %, 66.6 %, and 66.6 %, respectively. It was determined that:

- H1.1 can be rejected; hence, it can be concluded that the students did not fully understand the instructions and guidelines of CINDERELLA.
- H1.2 can be accepted; thus, it can be concluded that the examples provided by CINDERELLA are easy to understand.
- H2.1 can be accepted; thus, it can be concluded that the supporting information provided by CINDERELLA is sufficient for its application.

**Table 10.** Results on perceived utility. Source: own elaboration.

Hypothesis	Questions	Strongly agree (%)	Agree (%)	Neither agree nor disagree (%)	Disagree (%)	Strongly disagree (%)	Validation
H3.1	Question 1	25.0	75.0	0.0	0.0	0.0	Accepted
	Question 4	16.7	83.3	0.0	0.0	0.0	
H3.2	Question 2	8.3	91.7	0.0	0.0	0.0	Accepted
	Question 3	16.7	75.0	8.3	0.0	0.0	

For the utility variable, the percentage of students' perception regarding the combined percentages of "agree" and "strongly agree," based on questions 1, 2, 3, and 4, is 100 %, 100 %, 100 %, 91.7 %, and 100 %, respectively. It was determined that:

- H3.1 can be accepted; thus, it can be concluded that the tasks in CINDERELLA are useful for defining SPL feature models.
- H3.2 can be accepted; thus, it can be concluded that CINDERELLA is an organized and coherent approach.

**Table 11.** Results on Perceived Completeness. Source: own elaboration.

Hypothesis	Questions	Strongly agree (%)	Agree (%)	Neither agree nor disagree (%)	Disagree (%)	Strongly disagree (%)	Validation
H4.1	Question 1	0.0	75.0	25.0	0.0	0.0	Accepted
	Question 2	8.3	66.7	25.0	0.0	0.0	
	Question 3	0.0	66.7	33.3	0.0	0.0	
	Question 4	8.3	66.7	25.0	0.0	0.0	
	Question 5	0.0	50.0	50.0	0.0	0.0	
	Question 6	8.3	66.7	16.7	0.0	8.3	
	Question 7	8.3	66.7	25.0	0.0	0.0	
	Question 8	8.3	33.3	33.3	16.7	0.0	

For the completeness variable, the percentage of students' perception, regarding the combined percentages of "agree" and "strongly agree," based on questions 1, 2, 3, 4, 5, 6, 7, and 8, is 75.0 %, 75.0 %, 66.7 %, 75.0 %, 50.0 %, 75.0 %, 75.0 %, 75.0 %, and 41.6 %, respectively. It was determined that:

- H4.1 can be accepted, as 6 out of 8 questions obtained values higher than 60 %. Therefore, it can be concluded that all elements (artifacts, tasks, examples) of CINDERELLA are complete.

## 5.6 Discussion

The experiment followed the planned activities closely, with only minor deviations in time allocation. This allowed the students to fully participate in each activity of the process. The final activity involved collecting information through surveys using the Likert scale, providing students' perceptions of CINDERELLA. Responses regarding ease of use showed mixed results. While the students easily understood the practical examples provided by CINDERELLA (H1.2 accepted), the instructions and guidelines were not entirely clear to all participants (H1.1 rejected). This suggests that, although the practical examples were effective, the instructional content could benefit from greater simplicity or clarity to improve comprehension. The students positively rated the usefulness of the approach. They accepted

the hypotheses about its usefulness for defining Feature Models (H3.1) and its organization and coherence (H3.2). These results highlight the practical value and systematic nature of the approach from the participants' perspective. The students accepted the CINDERELLA completeness hypothesis (H4.1), with most questions receiving good scores. This indicates that they found the elements of CINDERELLA both necessary and sufficient, affirming the comprehensiveness of the approach.

Qualitative analysis of the survey responses shows that students generally have a positive perception of the CINDERELLA approach. High scores for usefulness and completeness highlight its effectiveness in defining Feature Models in SPL. However, results related to the ease of understanding the instructions indicate the need to improve the clarity of the instructional materials. These findings demonstrate the value of CINDERELLA in academic settings. To obtain more complete validation, further experimentation of the approach in real contexts and/or companies is needed with the participation the experts in SPL. These additional experiments will allow verification of CINDERELLA's applicability and effectiveness in practical and business situations, providing valuable data for refinement and adaptation to various settings.

Improving instructional clarity will likely enhance overall usability, ensuring that future users can fully benefit from CINDERELLA's strengths. This experiment not only validates the core concepts of CINDERELLA but also offers practical ideas for its continuous improvement.

## 6. CONCLUSIONS

This paper presents CINDERELLA, a collaborative approach proposed for defining feature models in SPLs, which outlines a workflow consisting of detailed collaborative tasks. The proposal integrates collaborative patterns, ThinkLets, and steps with collaborative techniques to encourage diverse and productive participation from each participant, facilitating task completion and feature model development.

Based on the evaluation results, it can be concluded that CINDERELLA is easy to use. Regarding usefulness, the elements comprising CINDERELLA are coherent and organized, making it perceived as a useful approach for defining feature models in SPLs. The elements defining CINDERELLA were considered sufficient and necessary, thus the approach is perceived as complete. Improvement of the instructions and guidelines based on evaluation feedback is emphasized. These results emphasize the value of CINDERELLA in academic settings, yet underscore the need for further experimentation in real-world contexts. Such additional validations will confirm the applicability and effectiveness of CINDERELLA in practical and commercial situations, providing valuable data for refining and adapting it to diverse environments.

CINDERELLA contributes to defining models in SPLs through collaborative work, enabling participants to collaborate by sharing knowledge and experience, providing, classifying, and evaluating information through collaborative tasks, and contributing to feature and feature model proposals. As a collaboratively designed approach, CINDERELLA enhances the potential for obtaining more comprehensive and useful feature models. Future work includes publishing the proposal for broader availability and developing a tool to facilitate distributed collaborative work.

The research group aims to develop a collaborative and automated feature model analysis tool. This tool, along with the proposed approach, will be presented to companies and development groups for evaluation in case studies or workshops, assessing its usefulness,



ease of use, and potential adoption in their projects. Furthermore, it is intended to develop a collaborative and automated feature model analysis tool. This tool, along with the proposed approach, will be presented to companies and development groups for evaluation in case studies or workshops, assessing its usefulness, ease of use, and potential adoption in their projects.

## 7. ACKNOWLEDGMENT

Thanks to Corporación Universitaria Comfacaucá – Unicomfacaucá for providing its facilities for their help in conducting the research.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## AUTHORS' CONTRIBUTION

Jazmin Gómez: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, writing – original draft, Writing – review & editing.

Pablo H. Ruiz: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Writing – original draft.

Vanessa Agredo Delgado: Conceptualization, Formal analysis, Investigation, Methodology, Software, Supervision, writing – original draft, Writing – review & editing.

Marta Cecilia Camacho: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Supervision, writing – original draft, Writing – review & editing.

## 8. REFERENCES

- [1] F. J. García-Peñalvo, “Capítulo 7. Ingeniería del Software,” in *Ciencia de la Computación e Ingeniería Artificial*, Salamanca, España: Departamento de Informática y Automática. Universidad de Salamanca, 2018, pp. 277-388.  
<https://darielingsilva.wordpress.com/wp-content/uploads/2020/11/1.-ingenieria-software.pdf>
- [2] L. M. Northrop *et al.*, “A Framework For Software Product Line Practice, Versión 5.0,” 2012.  
<https://insights.sei.cmu.edu/library/a-framework-for-software-product-line-practice-version-50/>
- [3] J. Bosch, G. Florijn, D. Greefhorst, J. Kuusela, J. H. Obbink, and K. Pohl, “Variability issues in software product lines,” in *Lecture Notes in Computer Science*, F. Van der Linden, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 13–21. [https://doi.org/10.1007/3-540-47833-7\\_3](https://doi.org/10.1007/3-540-47833-7_3)
- [4] T. Berger *et al.*, “What is a feature?: A qualitative study of features in industrial software product lines,” in *Proceedings of the 19th International Conference on Software Product Line*, New York, NY, USA, 2015, pp. 16-25. <https://doi.org/10.1145/2791060.2791108>
- [5] T. Von Der Maßen, and H. Lichter, “Deficiencies in feature models,” in *Variability Management for Product Derivation – Towards Tool Support: a one-day workshop to be held in conjunction with SPLC 2004 – 3rd Software Product Line Conference*, Boston, Massachusetts, 2004, p. 21. <http://www.soberit.hut.fi/SPLC-WS/AcceptedPapers/Massen.pdf>
- [6] D. Benavides, S. Segura, and A. Ruiz-Cortés, “Automated analysis of feature models 20 years later: A literature review,” *Information Systems*, vol. 35, no. 6, pp. 615-636, Sep. 2010.  
<https://doi.org/10.1016/j.is.2010.01.001>

- [7] P. Hofman, T. Stenzel, T. Pholey, M. Kircher, and A. Bermann, "Domain specific feature modeling for software product lines," in *SPLC '12: Proceedings of the 16th International Software Product Line Conference - Volume I*, Salvador, Brazil, 2012, pp. 229-238. <https://doi.org/10.1145/2362536.2362568>
- [8] S. Apel, D. Batory, C. Kästner, and G. Saake, "A development Process for Feature-Oriented Product Lines," In *Feature-Oriented Software Product Line*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 17-44. <https://doi.org/10.1007/978-3-642-37521-7>
- [9] K. Lee, K. C. Kang, and J. Lee, "Concepts and guidelines of feature modeling for product line software engineering," in *Lecture Notes in Computer Science*, C. Gacek, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 62-77. [https://doi.org/10.1007/3-540-46020-9\\_5](https://doi.org/10.1007/3-540-46020-9_5)
- [10] J. Gómez, P. H. Ruiz, V. Agredo Delgado, and M. C. Camacho, "Approach for feature models definition in software product lines based on collaborative work," in *Communications in Computer and Information Science*, P. H. Ruiz, V. Agredo Delgado, and A. Mon, Eds., Cham: Springer Nature Switzerland, 2024, pp. 77-92. [https://doi.org/10.1007/978-3-031-57982-0\\_7](https://doi.org/10.1007/978-3-031-57982-0_7)
- [11] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Carnegie Mellon University, Software Engineering Institute's, Technical Report CMU/SEI-90-TR-021, Nov. 1990. <https://insights.sei.cmu.edu/library/feature-oriented-domain-analysis-foda-feasibility-study/>
- [12] M. L. Griss, J. Favaro, and M. d'Alessandro, "Integrating feature modeling with the RSEB," in *Proceedings. Fifth International Conference on Software Reuse (Cat. No.98TB100203)*, Victoria, BC, Canada, 1998, pp. 76-85. <https://doi.org/10.1109/ICSR.1998.685732>
- [13] G. Bécan, R. Behjati, A. Gotlieb, and M. Acher, "Synthesis of attributed feature models from product descriptions," in *Proceedings of the 19th International Conference on Software Product Line*, New York, NY, USA, 2015, pp. 1-10. <https://doi.org/10.1145/2791060.2791068>
- [14] K. Chen, W. Zhang, H. Zhao, and H. Mei, "An approach to constructing feature models based on requirements clustering," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, Paris, France, 2005, pp. 31-40. <https://doi.org/10.1109/RE.2005.9>
- [15] P. Arcaini, A. Gargantini, and E. Riccobene, "Optimizing the automatic test generation by SAT and SMT solving for Boolean expressions," in *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Lawrence, KS, USA, 2011, pp. 388-391. <https://doi.org/10.1109/ASE.2011.6100079>
- [16] J. A. Galindo, D. Benavides, P. Trinidad, A. M. Gutiérrez-Fernández, and A. Ruiz-Cortés, "Automated analysis of featuremodels: Quo vadis?," *Computing*, vol. 101, no. 5, pp. 387-433, May. 2019. <https://doi.org/10.1007/s00607-018-0646-1>
- [17] K. Czarnecki, S. Helsen, and U. Eisenecker, "Staged configuration using feature models," in *Lecture Notes in Computer Science*, R. L. Nord, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 266-283. [https://doi.org/10.1007/978-3-540-28630-1\\_17](https://doi.org/10.1007/978-3-540-28630-1_17)
- [18] D. Benavides, P. Trinidad, and A. Ruiz-Cortés, "Automated reasoning on feature models," in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, O. Pastor, and Falcão e Cunha, Eds., Cham: Springer International Publishing, 2005, pp. 491-503. [https://doi.org/10.1007/11431855\\_34](https://doi.org/10.1007/11431855_34)
- [19] H. Shatnawi, and H. C. Cunningham, "Automated analysis and construction of feature models in a relational database using web forms," in *Proceedings of the 2020 ACM Southeast Conference*, New York, NY, USA 2020, pp. 233-238. <https://doi.org/10.1145/3374135.3385312>
- [20] M. Mendonca, M. Branco, and D. Cowan, "S.P.L.O.T: Software product lines online tools," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, New York, NY, USA, 2009, pp. 761-762. <https://doi.org/10.1145/1639950.1640002>
- [21] J. M. Davril, E. Delfosse, N. Hariri, M. Acher, J. Cleland-Huang, and P. Heymans, "Feature Model Extraction from Large Collections of Informal Product Descriptions," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, New York, NY, USA pp. 290-300, 2013. <https://doi.org/10.1145/2491411.2491455>
- [22] H. Shatnawi, and H. C. Cunningham, "Mapping SPL feature models to a relational database," en *Proceedings of the SouthEast Conference*, New York, NY, USA, 2017, pp. 42-49. <https://doi.org/10.1145/3077286.3077325>
- [23] R. Gacitúa, S. Sepúlveda, and R. Mazo, "Building feature models: A framework for comparing and classifying proposals," in *2016 XLII Latin American Computing Conference (CLEI)*, Valparaiso, Chile, 2016, pp. 1-12. <https://doi.org/10.1109/CLEI.2016.7833336>
- [24] L. Ochoa, O. González-Rojas, and T. Thüm, "Using decision rules for solving conflicts in extended feature models," in *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, New York, NY, USA, 2015, pp.149-160. <https://doi.org/10.1145/2814251.2814263>

- [25] M. Mendonça, T. T. Bartolomei, and D. Cowan, "Decision-making coordination in collaborative product configuration," in *Proceedings of the 2008 ACM symposium on Applied computing*, New York, NY, USA, 2008, pp. 108-113. <https://doi.org/10.1145/1363686.1363715>
- [26] M. A. Noor, P. Grünbacher, and R. O. Briggs, "A collaborative approach for product line scoping: a case study in collaboration engineering," in *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*, New York, NY, USA 2007, pp. 216-223. <https://dl.acm.org/doi/10.5555/1332044.1332079>
- [27] M. A. Noor, R. Rabiser, and P. Grünbacher, "Agile product line planning: A collaborative approach and a case study," *Journal of Systems and Software*, vol. 81, no. 6, pp. 868-882, 2008. <https://doi.org/10.1016/j.jss.2007.10.028>
- [28] M. C. Camacho Ojeda, F. Á. Rodriguez, and C. A. Collazos, "Identifying collaborative aspects during software product lines scoping," in *Proceedings of the 23rd International Systems and Software Product Line Conference - Volume B*, Paris, France 2019, pp. 98-105. <https://doi.org/10.1145/3307630.3342420>
- [29] M. C. Camacho, F. Álvarez, C. A. Collazos, P. Leger, J. A. Hurtado, and J. D. Bermúdez, "A Collaborative Method for Scoping Software Product Lines: A Case Study in a Small Software Company," *Applied Sciences*, vol. 11, no. 15, p. 6820, Jul. 2021. <https://doi.org/10.3390/app11156820>
- [30] D. M. Weiss, and D. Li, "Workshop on Collaborative Software Product Line Engineering," in *2012 IEEE 25th Conference on Software Engineering Education and Training*, Nanjing, China, 2012, pp. 77-78. <https://doi.org/10.1109/CSEET.2012.28>
- [31] F. J. Pino, M. Piattini, and G. Horta-Travassos, "Managing and developing distributed research projects in software engineering by means of action engineering by means of action-research," *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 68, pp. 61-74, Feb-Sep. 2013. <https://www.redalyc.org/pdf/430/43029811006.pdf>
- [32] G. J. de Vreede, R. O. Briggs, and A. P. Massey, "Collaboration Engineering: Foundations and Opportunities: Editorial to the Special Issue on the Journal of the Association of Information Systems," *Journal of the Association for Information systems*, vol. 10, no. 3, pp. 121-137, Mar. 2009. <https://doi.org/10.17705/1jais.00191>
- [33] R. Briggs, G. Kolfshoten, G. J. de Vreede, and D. Douglas "Defining Key Concepts for Collaboration Engineering," in *Americas Conference on Information Systems (AMCIS)*, New York, NY, USA, 2006, pp. 121-128. <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1565&context=amcis2006>
- [34] D. Gray, S. Brown, and J. Macanufo, *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers*, Newton, Massachusetts, USA: O'Reilly Media, 2010. [https://books.google.com.co/books/about/Gamestorming.html?hl=es&id=-xnEDNPxwYC&redir\\_esc=y](https://books.google.com.co/books/about/Gamestorming.html?hl=es&id=-xnEDNPxwYC&redir_esc=y)
- [35] G. L. Kolfshoten, and G. J. de Vreede, "The collaboration engineering approach for designing collaboration processes," in *Lecture Notes in Computer Science*, J. M. Haake, S. F. Ochoa, A. Cechich, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 95-110. [https://doi.org/10.1007/978-3-540-74812-0\\_8](https://doi.org/10.1007/978-3-540-74812-0_8)
- [36] A. Solano, T. Granollers, and C. A. Collazos, "Modelado de Procesos Colaborativos Extendiendo Elementos de la Notación HAMSTERS," *Revista Colombiana de Computación*, vol. 16, no 2, pp. 144-161, Aug. 2016. <https://revistas.unab.edu.co/index.php/rcc/article/view/2555/2190>