A feature-based Modeling Strategy for Managing Stakeholders

Karam Ignaim\textsuperscript{1} and Joao Fernandes\textsuperscript{2}

\textsuperscript{1}Al-Balqa Applied University Prince Abdullah bin Ghazi Faculty of Information Technology
\textsuperscript{2}Universidade do Minho Departamento de Informatica

July 4, 2023

Abstract

Successful product development is mostly based on how well ideas, opinions, and contributions from stakeholders are taken into account. There are signs that the way a project manager involves stakeholders in the project is directly linked to how well software products are delivered. Engagement of stakeholders enables organizations to proactively consider the requirements and desires of anyone with a stake in the stakeholder management process. For the stakeholder management process to work, all stakeholders must be listed in a reference document, and their involvement in the process must be kept up-to-date. The main goal of this study is to find a solution to this problem by coming up with a feature-based strategy for capturing and documenting the right stakeholders during the stakeholder management process. The strategy also supports ongoing monitoring of the status and attributes of stakeholders during this process. To evaluate the strategy, we have conducted a case study on the stakeholder management process of Smart Home to assess the ease of use, usefulness, effectiveness, and efficiency of the proposed strategy. The results reveal that our strategy of documenting and keeping track of the stakeholders makes the stockholder management process more effective and efficient. The results have also shown that the proposed strategy is considered easy to learn and useful by the software engineers.
A feature-based Modeling Strategy for Managing Stakeholders

Karam Ignaim\textsuperscript{a,1}, João M. Fernandes\textsuperscript{b,2}

\textsuperscript{a}Department of Software Engineering, Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University (BAU), Jordan
\textsuperscript{b}Departamento de Informática, Universidade do Minho, Braga, Portugal

Abstract

Successful product development is mostly based on how well ideas, opinions, and contributions from stakeholders are taken into account. There are signs that the way a project manager involves stakeholders in the project is directly linked to how well software products are delivered. Engagement of stakeholders enables organizations to proactively consider the requirements and desires of anyone with a stake in the stakeholder management process. For the stakeholder management process to work, all stakeholders must be listed in a reference document, and their involvement in the process must be kept up-to-date.

The main goal of this study is to find a solution to this problem by coming up with a feature-based strategy for capturing and documenting the right stakeholders during the stakeholder management process. The strategy also supports ongoing monitoring of the status and attributes of stakeholders during this process. To evaluate the strategy, we have conducted a case study on the stakeholder management process of Smart Home to assess the ease of use, usefulness, effectiveness, and efficiency of the proposed strategy. The results reveal that our strategy of documenting and keeping track of the stakeholders makes the stockholder management process more effective and efficient. The results have also shown that the proposed strategy is considered easy to learn and useful by

\textsuperscript{*}Corresponding author
\textsuperscript{1}karam.ignaim@bau.edu.jo,
\textsuperscript{2}jmf@di.uminho.pt

\textit{Preprint submitted to Elsevier}
the software engineers.

Keywords: Stakeholder Management Process, Feature Models, Feature Model Refactoring.

1. Introduction

A system stakeholder is an individual or organization that has an impact on the system's requirements, either directly or indirectly. Stakeholders can include anybody with an interest in the system, including users, operators, developers, architects, consumers, and testers [1][2]. Stakeholder selection refers to the process of identifying appropriate stakeholders to elicit requirements for a certain product development process [3]. The selection of appropriate stakeholders for software engineering is a critical stage and a prerequisite for software requirement elicitation. When compared to other viable options for requirements, stakeholders are by far the most essential [3][4][5]. Typically, stakeholder elicitation often starts with suggestions from management or domain experts about who the relevant stakeholders are and what they want [1][6]. When working with constrained resources, it’s important to choose the most suitable stakeholders for requirement elicitation [7][8]. If stakeholders do not receive sufficient attention from the requirements engineer, they may be too critical of the product development process [9]. This implies that the different stakeholders involved in a development process must be mapped appropriately through user management and access rights administration. Studies have shown that improper stakeholder selection typically results in a poor requirement elicitation procedure. Such occurrences would afterwards have severe effects on the product development process, including expensive rework, timetable overruns, and poor software [3][10][3][11].

Stakeholder management is an iterative process that makes it easier for software engineers to identify, document, and keep track of stakeholders. According to research [12], further work needs to be done to bring novel insights into the stakeholder management process [13]. The main objective of this study is to ad-
dress this issue by proposing a feature-based model for capturing, documenting, and monitoring stakeholders in the stakeholder management process from the start of project initiation all the way through project closure. The findings of research also indicate that selecting stakeholders with appropriate characteristics—factors such as the stakeholder’s role, knowledge, interest, communication skills, and personality—has significant effects on the requirements elicitation phase [8] [9]. Thus, our strategy uses these factors to propose a strategy based on feature models that address stakeholder involvement in the stakeholder management process, which influences the appropriate stakeholder selection with respect to the requirement elicitation purpose. Our study does not use feature models to represent software product lines. Instead, we propose a strategy that exploits feature models to document stakeholders in a single location and supports their engagement in the product development through continuous monitoring. Further, it is worth mentioning that this study focuses on stakeholder capturing, documentation and monitoring rather than stockholder selection and analysis. Stakeholder analysis is a common elicitation technique that requires a thorough examination of the stakeholder, such as determining who they are and what their attitudes and interests are [14]. Prioritizing stakeholders requires taking into account the myriad ways in which they can contribute to product development process. Stakeholder prioritization is concerned with the varying importance of stakeholders [15]. The prioritization of stakeholders was found to regularly make use of a variety of various attributes in the associated studies in particular, type, interest, level of influence or impact, level of understanding, capacity for interpersonal involvement, and relationship. Few studies have examined the model or framework for stakeholder selection using stakeholder prioritization [16]; thus, we decided to include this issue in our own research.
2. Background

2.1. The Stakeholder Management Process

Figure 1 shows how the stakeholder management process works. To start, the software engineers must (1) **identify** the stakeholders of the project. They do not need to collect all of them, but they must identify the most important ones. There are numerous methods for accomplishing this, including so-called content analysis and brainstorming [3][5]. Following that, the software engineers must (2) **capture and document** the stakeholders in a reference document. Typically, this is accomplished using a document such as the Stakeholder Register or the Stakeholder Influence Matrix. Once software engineers have identified the stakeholders, the following step is to (3) **analyze** and classify them. This is accomplished by the software engineers utilizing a variety of metrics and techniques; the most frequent method involves determining each stakeholder’s level of power (also known as influence) and interest in the project. The primary goal of stakeholder management should be to (4) **engage** each stakeholder in order to maximize the project’s benefits. This phase of the process focuses on determining the optimal frequency and method of communication with each individual. Since stakeholder management is an ongoing, iterative process that continues throughout the project’s lifecycle, software engineers continue identifying, documenting, analyzing and classifying, and planning and implementing engagement from the beginning of project initiation until the end of project closure, which is referred to as (5) **monitor and update**.

2.2. Feature Model

Feature models are commonly used to represent the commonality and variability of a system in terms of mandatory, optional, and exclusive features, as well as propositional constraints over the features [17] [18] [19]. Engineers typically use feature models to represent features and their connections in a tree-like structure. Feature models provide various modeling notions for this purpose, such as mandatory and optional features, a feature hierarchy, feature groups,
and cross-tree constraints. Figure 2 shows a demo example for “Car” 3 feature model. There is always a single root feature present in every feature model (or “tree”). The “Car” is the name of the root node. The “Car” feature model supports “Engine” and “Brake” features, with “Regenerative” and “Drum” features available on an optional basis (i.e., brake types). The “Engine” is a common feature that is available as “Hybrid,” “Petrol,” or “Electric.” A feature is considered common (or mandatory) if it is present in all members of a family. In contrast, a feature that is not present in every member of a family is an optional

3We use double quotes when we refer to a feature of the feature model, such as “Car” in Figure 2.
Extended feature models offer much more information about the features through the use of feature attributes or parameters [23]. An attribute of a feature is any measurable property of the feature that can appear in complex cross-tree relations [23]. Using the attributed feature model, every feature may have one or more attributes with domain value type like Integer, Long, Double, String, or Boolean. Thus, attributes of a specific feature would be able to enhance the graphical feature model with such information. Figure 3 depicts a portion of the feature model in Figure 2 with attributed features and our unique notation, which was inspired by [24]. As certain features contain attributes or parameters, Figure 3 shows the simplified version of the attributed feature model for the “Car” feature model (presented in Figure 2 without attributes), [25]. The upper side of the picture depicts the feature model, while the lower side depicts the attribute “HEV” of the “Hybrid” feature, where possible values for the attribute are “mild-hybrid”, “full-hybrid”. The value of “HEV” attribute in this demo example is “fully-hybrid”.

3. Strategy

This section presents the strategy proposed in this research work. The strategy uses the attributed feature models to document and monitor, in a single location, the involvement of stakeholders in the stakeholder management process.

---

4HEV is an abbreviation for Hybrid Electric Vehicles.
This makes communication, collaboration, and decision-making in the development process easier and faster. The strategy also supports the updates in the stakeholder status for the purpose of requirement elicitation in the product development process.

3.1. Stakeholder Lists

A complementary method to identify stakeholders is maintaining checklists using, for example, tables and spreadsheets [1]. This permits the systematic and targeted engagement of pertinent stakeholders. Other research makes use of the stakeholder database, which is an integral part of achieving the objective of project and disseminating its findings[26]. If the list of stakeholders is updated too late or insufficiently, it is possible that essential system aspects may go undiscovered, that the objective of project are missed, or that considerable additional expenses are incurred to solve issues.

3.2. Stakeholders Documentation

The stakeholders best suited for requirement elicitation must be carefully chosen because of the limited resources available to interact with all of them [27]. Commonly, some details should be put, for example, in tables and spreadsheets to keep track of the people who have a stake in the development process. These
details include name, role, and additional personal and contact information; availability in time and space during the development of the project; relevance of the stakeholder; area and level of expertise of the stakeholder; and the project-related goals and interests of the stakeholder. All of these criteria cannot be met because projects must comply with strict deadlines and financial constraints. Therefore, in order to determine priorities, a stakeholder analysis is necessary [27]. Managing stakeholders also involves a constant flow of information through regular status updates and ongoing stakeholder involvement. This may help the requirements engineer turn people who were only negatively affected by the project into contributors who are well-integrated and jointly responsible stakeholders [1]. Thus, in this research, we decided to tackle this problem by using an evolvable artifact called an attributed feature model that makes it easier to document and keep track of how stakeholders are involved in the elicitation process.

3.3. Stakeholder Types

Stakeholders could be positive, negative, or neutral factors for the product’s development. Key stakeholders are those who have a considerable interest in and/or influence over your project (through money, resources, or power, for example) [28]. These are the people who require the most attention and commitment. The type of stakeholder may further be subdivided into four types: primary stakeholders, secondary stakeholders, external stakeholders, and extended stakeholders [29][27][13]. Definitely, primary stakeholders are more essential than secondary stakeholders because they are directly affected by development of the product. Thus, the primary stakeholders should be included due to their considerable impact on the product. If not, it is more difficult to build a product well aligned with the stakeholder’s expectations. The importance of primary stakeholders comes from their control, authority, and responsibility over the resources [29].

Secondary stakeholders are people who are indirectly impacted by the success of the product. Although they are less significant than primary stakeholders,
their interest in the product should be effectively managed. External stakeholders are those outside the product development team who have expectations for it. Often, it is essential to recognize the contributions they can make to the product. Often, external stakeholders are people who aid others in achieving their visions. Based on the above description, our strategy visually represents the stakeholders of a product as features, with the primary stakeholders depicted as mandatory features and the other types of stakeholders as optional features.

In the context of the present study, the strategy helps project managers and requirements analysts document the relevant stakeholders for requirements elicitation based on their characteristics during the stakeholder management process and keep regular monitoring of the stakeholders throughout the process.

The strategy has two uses for the attributed feature model:

1. The stakeholders feature model will serve as a repository to capture and document the identified stakeholders in a single location. This is typically performed via a document called a stakeholder feature model.

2. The stakeholder feature model serves as an evolvable artifact that helps to continuously monitor all the stakeholders currently identified in the feature model and update their statuses as appropriate.

In this work, we use an attributed feature model, in which each partner can fill in his/her desired stakeholders in a standard way

3.4. Information to Collect in the Stakeholder Feature Model

Based on the investigation of the recent studies concerning stakeholder identification to select appropriate stakeholders, we decided to include the following factors in the stakeholder feature model [8][9][27][3]:

1. Role of the stakeholder.

2. Area of the stakeholder.

3. Extent of expertise of the stakeholder.

4. Goals of the stakeholder.

5. Interests of the stakeholder regarding the project.
6. Personal and contact data of stakeholders, including Name, Telephone, and Email.

Table 1 shows the main attributes of each stakeholder that our strategy uses to build the different types of stakeholder feature models, including all types of stakeholders. The table also shows the relationship between the features of the feature model and each attribute of the stakeholder. Each row relates an attribute of the stakeholder to its corresponding feature design in the stakeholder feature model. In the stakeholder feature model, for example, the "Product Name" attribute will be shown as a "Root Feature," and the "Area" attribute title will be shown as a "Parent Feature." Further "Area" classifications will be presented as "Child Feature."
3.5. Design the Stakeholders Feature Model

Based on the factors that influence stakeholder identification (i.e., attributes of stakeholder), using our strategy, we decide to design the stakeholders feature model as shown in Figure 4. Our strategy visually represents the stakeholders of a product as features. The primary stakeholders depicted as mandatory features (i.e., stakeholder 1 of the stakeholder feature model presented in Figure 4) and the other types of stakeholders (i.e., secondary, external, and extended stakeholders) as optional features (i.e., stakeholder 2 of the stakeholder feature model presented in Figure 4). For each stakeholder feature model, the root feature shows the “Product Name”, the parent features hold the stakeholder information (i.e., factors/attributes) regarding the project, such as “Role”, “Area”, “Experience”, “Goals”, and “Interests”, as well as “Personal and Contact Data.” The child features of each parent feature show further information of the stakeholder’s attributes. The “Personal and Contact Data” feature makes use of feature attributes/parameters to represent the stakeholder’s “Name”, “Phone number”, and “Email address”, as well as its “Priority” and whether (s)he is a “Key” stakeholder. The starting point for stakeholder elicitation is often the suggestions of relevant stakeholders that are made by management or by domain experts, for example. On the basis of these suggestions, relevant stakeholders can be identified and then our strategy uses the attributed feature model to document and monitor their involvement in the product development process.

4. Running Example

To clarify our strategy, we present a running example. We use as case study the Health Care Mental Patient Management System (HCM-PMS). This system is used to keep records of patients who are receiving treatment for issues related to their mental health. Following the Stakeholder Management Process presented in subsection 2.1, we propose that the requirement engineer has to: (1) identify the stakeholders in the HCM-PMS. The following list identifies some of the individuals who could be considered stakeholders in the HCM-PMS.
• **Patients** whose information is stored in the database.

• **Doctors** who diagnose and treat patients.

• **Nurses** who organize doctor visits and carry out some treatments.

• **IT staff** who are responsible for system installation and maintenance

Using our strategy, the software engineer must (2) **capture and document** the stakeholders in a single document, namely the stakeholder feature model. Figure 5 depicts the attributed feature model of MHC-PMS. To build the stakeholder feature model, the software engineer needs to determine the priority of each stakeholder (on a scale from 1 to 3, where 1 is the highest priority, 2 is medium priority, and 3 is the lowest one) and then (3) **analyze and classify** them. The software engineer may use the stakeholder feature model (e.g., presented in Figure 5) to (4) **engage** each stakeholder in order to maximize the quality of the product development process. The engagement phase of the process focuses on determining the optimal frequency and method of communication with each individual. Using our strategy, the software engineer can use the information presented in the features and attributes of each stakeholder (e.g., the “Contact Information” feature and “Phone” attribute presented in the stakeholder feature model of Figure 5) to support the communication with stakeholders. As
mentioned earlier in this manuscript, stakeholder management is an ongoing, iterative process that continues throughout the product development process. Thus, software engineer can continue identifying, documenting, analyzing and classifying, and planning and implementing engagement from the beginning of the product development process until the end of the process. Following our strategy, the software engineer can use the stakeholder feature model for continuous (5) monitoring and updating of the stakeholders.

Ongoing monitoring of the stakeholders implies a continuous update of the stakeholder’s information. Thus, the software engineer may need to update the information stored in the stakeholder feature model (i.e., feature model refinement). For that, our strategy adopts the following operations on the feature model in order to refine the stakeholder feature model to reflect the new status and updated information of the stakeholders.

1. Feature model refactoring.
2. Insertion operator.

![Feature Model Refactoring](image)

**Feature Model Refactoring** In case a software engineer needs to perform a transformation to the stakeholder feature model by keeping or improving its configurability (e.g., adding a new role to a specific stakeholder), he/she can apply a set of sound refactorings for feature models presented in [30]. The sound refactorings guarantee configurability improvement. For example, a software engineer can extend the role of the doctor to include "View Consultation" by applying Refactoring 12 (add optional node) of the catalog and then applying...
Refactoring 2 (collapse optional and or) of the catalog on the stakeholder feature model presented in Figure 6. The stakeholder feature model after refinement is depicted in Figure 6. A red asterisk is placed close to the refactoring point. A complete explanation for the feature model refactoring is presented in [30], where the authors offer a list of sound feature model refactorings, which were verified by automatically examining the properties of the resulting feature models. The authors also provide a catalog of refactorings that is accompanied by examples to demonstrate their use.

**Insertion operator** In case a software engineer needs to perform a modification to the stakeholders feature model to capture a change in the stakeholder lists that includes adding a new stakeholder (i.e., installers) and his/her related information (attributes). This change (adding a new stakeholder) can be presented in the form of a sub-tree (i.e., sub-feature model). The software engineer can use the insertion operator proposed by [31] to insert this sub-feature model into the stakeholder feature model in a specific position (i.e., a target feature in the stakeholder feature model) that is specified by the requirement engineer. In the stakeholder feature model, for example, the stakeholder list can be expanded to include an Non-Governmental Organization (NGO) stakeholder. Figure 7 depicts the stakeholder feature model after inserting the NGO sub-feature model in a specific position (i.e., the HCM-PMSA root feature) of the stakeholder feature model.

![Stakeholder Feature Model](image)

Figure (6) The stakeholder feature model of the HCM-PMS after refinement.
5. Case Study Evaluation

A case study is one of the most common approaches for evaluating software engineering methods[32]. In this section, our proposed strategy is applied and evaluated in a practical case study; thus, we investigate the applicability of our strategy in a real-world stakeholder management process.

5.1. Case Study context

Home automation refers to the automatic control of household features, and appliances. In practical terms, it means that one can easily control the utilities and features of the home via the Internet to make life more secure and convenient and to spend less on household bills. Home automation projects usually span across various domains. In recent years, Smart Homes have become a popular subject of scientific and technological research and development. Typically, microprocessors are utilized to control everyday technological appliances. With the advent of home automation, these devices are connected to a network, enabling them to coordinate and perform complex tasks without human
intervention. Smart Homes boast intuitive user interfaces that make their functionality easy to access. The evolution of Smart Homes is influenced by diverse fields, such as web technology that permits remote access to home operations over the Internet.

5.2. Stakeholders in the Smart Home

The design, installation, and operation of a home automation system involve numerous stakeholders. It is essential to consider the interests of all stakeholders when designing, installing, and operating a home automation system to ensure that the system meets their needs and expectations. As shown in Figure 8, the different stakeholders and their interests are presented below:

1. Residents: The residents are the primary users of the Smart Home system. They have specific requirements for the system based on their lifestyle, habits, and preferences. They may want a system that is easy to use, provides security, energy efficiency, and convenience.

2. Building owner: The building owner has an interest in the Smart Home system that adds value to the property. They may want a system that is reliable, efficient, and cost-effective. They may also want a system that is easy to maintain and upgrade.

3. Caretaker: The caretaker is responsible for the administration and accounting of the residence. They have an interest in the Smart Home system that is easy to manage, monitor, and control. They may also want a system that is secure and provides reliable data for billing purposes.

4. Company managers: The managers of the company that develops and brings the Smart Home system to market have an interest in the features of the product, its functionality, and its marketability. They want a system that is innovative, efficient, and competitive in the market.

5. Developers: The developers of the Smart Home system have an interest in creating a system that meets the requirements of users while being technically feasible and cost-effective. They want to create a system that is scalable, secure, and easy to maintain.
6. Installers: The installers of the Smart Home system have an interest in installing the system correctly and efficiently. They want a system that is easy to install, configure, and integrate with other systems.

7. Maintenance personnel: The maintenance personnel of the Smart Home system have an interest in ensuring that the system is operating at peak performance. They want a system that is easy to diagnose, troubleshoot, and repair.

Figure (8) Stakeholders’ examples of the Smart Home domain [33].

5.3. Case Study Design and Planning

We performed a case study, according to the guidelines presented by Runeson et al. [34][35], to assess the ease of use and the usefulness of our proposed strategy. Besides this being its first evaluation, the obtained results have shown the strategy to be a promising solution. In order to evaluate our strategy, we need to plan the case study. First of all, we identified the objective. Thus, we stated the goal of the case study as follows: to analyze the proposed strategy for the purpose of evaluating it with regard to its ease of use and usefulness from the viewpoint of a set of software engineers in the context of a Smart Home project during the stakeholder management process. Secondly, we identified the case (context) of the study. The context of the study is the stakeholder management process during product development. The selected product is called Smart Home.
Table (2) The research questions of the case study.

<table>
<thead>
<tr>
<th>Q1</th>
<th>What is the impact of the strategy presented in this research on the performance of software engineers in the stakeholder management process?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2</td>
<td>What is the impact of the strategy presented in this research on usability of the software engineer in the stakeholder management process?</td>
</tr>
<tr>
<td>Q3</td>
<td>How does the ease of use and usefulness of the system change over time as software engineers become more familiar with the proposed strategy?</td>
</tr>
</tbody>
</table>

(see Subsection 2.1). In order to fulfill the objective of this study, the research questions that state what is needed to know are presented in Table 2.

Based on the design of the study, the study has one treatment, which is the strategy presented in this research. The study has two subjective dependent variables, which are ease of use and usefulness.

- **Ease of Use:** The degree to which a software engineer thinks that it will be easy to use our strategy. This variable is a subjective assessment of how hard the strategy is to learn and use.

- **Usefulness:** The degree to which a software engineer thinks that the proposed strategy will achieve its stated objectives. This variable represents a perception of the effectiveness of the strategy.

We performed a usability survey to examine both variables after putting our proposed strategy into a real-world setting. The survey consisted of questions regarding ease of use, ease of learning, fatigue, simple preference, and other questions that were relevant to the strategy that was being used. In accordance with the recommendations for conducting an effective survey that are detailed in [35], the NASA TLX (Task Load Index, Figure 9) was adjusted so that it better reflected our approach. In order to accomplish this, we rephrased questions 1, 2, and 3, replacing the word "task" to "strategy" wherever it appeared.

The NASA-TLX is a widely used subjective measure for evaluating the workload and task demand associated with a particular task or activity. The NASA-TLX uses six questions to assess different aspects of task demand, including physical demand, mental demand, temporal demand, performance, effort, and
Each question is rated on a scale of 1 to 7, where 1 represents the lowest task demand and 7 represents the highest, with the exception of the performance question, where 1 indicates the highest and 7 indicates the lowest. This allows the respondent to rate different aspects of the task demand according to their perceived level of difficulty or effort required. The overall TLX score is calculated as the sum of the six scores, providing a single score that represents the overall perceived workload and task demand associated with the task or activity. The NASA-TLX is widely used in a variety of contexts, including aviation, healthcare, and human factors research, to evaluate the subjective experience of task demand and workload (see Figure 9).
Our study has two hypotheses related to the ease of use and usefulness of the strategy. In the following, we formulate the null hypotheses (H0E and H0U) and the alternative hypotheses (H1E and H1U) to assess the two subjective dependent variables:

- **H0E**: The proposed strategy is difficult to use.
- **H1E**: The proposed strategy is easy to use.
- **H0U**: The proposed strategy is not useful.
- **H1U**: The proposed strategy is useful.

In the study, we also define a correct key solution for the tasks of the stakeholder management process of the Smart Home. The aim is to compare solutions of the software engineers with the key solution to analyze the degree to which the software engineers applied the strategy effectively and efficiently. Thus, six objective dependent variables were defined in Table 3.

In order to start the study, we prepared the case study design (see Table 4). This design was used within both sessions (the Software Engineering course and Pioneers Company). It took us 2 days for each group (the Software Engineering course and Pioneers Company), with a total of 3 hours and 30 minutes per day, to perform the case study for each session. On the first day, we performed a one-hour training session, which included a presentation of the proposed strategy and a practical exercise to apply the strategy. On the second day, we performed the case study. In the first 15 minutes, we asked the software engineers to answer a form with background information, and then they were asked to perform the 3 tasks of the strategy. After completing the tasks, the software engineers answered a survey about the strategy. The software engineers were chosen based on convenience, the software engineers of Group 1 (i.e., the Software Engineering course) were students taking the Software Engineering course at Al- Balqa’ Applied University (BAU, Jordan) and software engineers from Pioneers Company (Jordan). As case study instrumentation, several documents were created: slides for the training session, an explanation of the tasks,
Table (3) Objective dependent variables of the case study.

<table>
<thead>
<tr>
<th></th>
<th>Effectiveness-FM Design is calculated as the ratio of the number of correct designs of the stakeholder feature model identified by the software engineer to the total number of correct designs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Effectiveness-FM Re-Update is calculated as the ratio of the number of right updates to the stakeholder feature model that the software engineer performed (using feature model refactoring) to the total number of right updates to the stakeholder feature model.</td>
</tr>
<tr>
<td>3</td>
<td>Effectiveness-FM In Update is calculated as the ratio between the number of right updates performed by the software engineer in the stakeholder feature model (using the insertion operator) and the total number of right updates performed in the stakeholder feature model.</td>
</tr>
<tr>
<td>4</td>
<td>Effectiveness-FM In Update is calculated as the ratio between the number of right updates performed by the software engineer in the stakeholder feature model (using the insertion operator) and the total number of right updates performed in the stakeholder feature model.</td>
</tr>
<tr>
<td>5</td>
<td>Efficiency-FM Re-Update is calculated as the ratio between the number of correct updates to the stakeholder feature model that the software engineer performed (using feature model refactoring) and the total time spent on the feature model updates.</td>
</tr>
<tr>
<td>6</td>
<td>Efficiency-FM In Update is calculated as the ratio between the numbers of right updates in the stakeholder feature model (using the insertion operator) that the software engineer performed and the total time spent on the updates in the feature model.</td>
</tr>
</tbody>
</table>

and a questionnaire (i.e., the NASA TLX survey). These documents were used by the software engineers, who were questioned regarding their experiences in the field; the results revealed that each of them had prior experience in this context. As a result, we did not classify software engineers according to their experience with the stakeholder management process. The first session consisted of six undergraduate students from BAU. Six software developers from Pioneers Company participated in the second session. Thus, there were a total of 12 participants in the empirical investigation. For training on the strategy, we presented to the software engineers the running example explained in Section 4. The stakeholder feature model of the HCPMS was created. The feature model had 32 features. After the strategy training, the software engineers were asked to perform 3 tasks. Regarding **Task 1**, the software engineers were asked to build the stakeholder feature model for the Smart Home (see Section 5.2). In **Task 2**, the software engineers were asked to update the information of the stakeholders by applying feature model refactoring to the stakeholder feature model. Finally, in **Task 3**, the software engineers were asked to update the information of the stakeholders by applying an insertion operator over the same
Table (4) The case study design.

<table>
<thead>
<tr>
<th>First day</th>
<th>The strategy training</th>
<th>✓ Introduce the strategy – 30 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✓ Apply a practical example – 30 minutes</td>
</tr>
<tr>
<td>Second day</td>
<td>Collect software engineers’ background</td>
<td>✓ Apply the Background Questionnaire – 15 minutes</td>
</tr>
<tr>
<td></td>
<td>Conduct the case study</td>
<td>✓ Identify tasks by the researchers – 30 minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Perform tasks by the software engineers – one hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Check the tasks by the researchers – 30 minutes</td>
</tr>
<tr>
<td></td>
<td>Collect the software engineers’ feedback</td>
<td>✓ Apply NASA TLX survey – 15 minutes</td>
</tr>
</tbody>
</table>

Before starting the actual study, the participants were asked to complete a form with background information containing five questions that aimed to gather information about the experience of the software engineer in various areas related to the study, such as stakeholder management, feature models, and requirement elicitation. After the study was completed, the software engineers were asked to complete a survey called NASA TLX, which is a widely used tool to measure mental workload. This survey contained six questions about the opinions of software engineers about the strategy used in the study.

In this case study, we performed data collection before, during, and after the empirical study. Before the study, we collected data from the software engineer’s background. During the empirical study, the stakeholder management process was performed by designing and updating the stakeholder feature model in order to deal with the following tasks: 1) build the stakeholder feature model 2) update the information of the stakeholders by applying feature model refactoring to the stakeholder feature model; and 3) update the information of the stakeholders by applying an insertion operator over the same feature model. After the empirical study, we were able to collect data from the software engineers through the NASA TLX survey.

To perform the data collection of our proposed case study, we have used the following:

1. **Background Form:** It took us around 15 minutes, for each session (BAU
and Pioneers Company) to collect the information about the software engineer’s experience with stakeholder management processes, requirements engineering, and feature models. Most of the software engineers at BAU knew what the stakeholder management process and requirement engineering are, and they have, in general, less than 1 year of experience within them. Moreover, the majority of the software engineers at Pioneers Company knew what the stakeholder management process and requirement engineering are and have between 1 and 5 years of experience in them. Unfortunately, the software engineers from both groups have no knowledge of or experience with feature models. For that, we have offered them an intensive presentation session.

2. **Empirical Study:** It took us around 3 hours and 30 minutes, for each session (BAU and Pioneers Company), to collect the software engineer’s answers about the stakeholder management process of the Smart Home. The data was collected for each one of the following tasks:

- **Identify the tasks (what).** All the software engineers received 3 tasks to perform using the proposed strategy. The first task (Task 1) was to design the stakeholder feature model. The second task (Task 2) was to update stakeholder information using the feature model refactoring. The third task (Task 3) was to update stakeholder information using the insertion operator.

- **Design the stakeholder feature model of the Smart Home.** Once the stakeholder management process for the Smart Home is identified, the software engineers should design the stakeholder feature model for the Smart Home (Task 1). For that, we gave the software engineer Figure 8 and its related explanation (see subsection 5.2), and the participants should design the stakeholder feature model according to our proposed strategy. The software engineers should capture and document the stakeholders of the Smart Home in the stakeholder feature model.
• **Update the stakeholder feature model.** Last but not least, the participants should perform a transformation that makes a change to the stakeholder feature model by keeping or improving its configurability (e.g., adding new role to the “Owner” stakeholder like, the home owner shall be able to send a request about the door lock status via mobile phone). They should include new features to the stakeholder feature model (Task 2). The new feature (new role) should be associated with the stakeholder feature model using feature model refactoring. In the second update, the software engineer needs to perform a modification to the stakeholders feature model to capture a change in the stakeholder lists that includes adding a new stakeholder and his/her related information (Task 3). In the stakeholder feature model of the Smart Home, the stakeholder list can be expanded to include the “Installer”, “Technician”, and “Security” stakeholders using the insertion operator.

3. **Survey:** To collect the survey information, it took approximately 15 minutes per session (BAU and Pioneers Company). The NASA TLX survey results indicated that the software engineers perceived the strategy used in the study to have several steps that were dependent on feature models. This finding suggests that there may be a need for tool support to facilitate the use of the strategy. Additionally, while the feature model concept was reported as difficult to understand, the software engineers noted that the steps of the strategy were well explained and easy to follow, providing a systematic way to manage stakeholder engagement in product development.

After completing the data collection, we started the data analysis. We conducted both qualitative and quantitative analyses of the data. The first analysis was qualitative and focused on the objective variables, such as effectiveness and efficiency, observed during the execution of the tasks. This analysis aimed to identify any deficiencies in the strategy and improve the quality of the report.
in a qualitative manner.

The second analysis was quantitative and involved closed-ended questions that were filled out by the software engineers after the empirical study. This information was analyzed in a quantitative manner to determine the perceptions of software engineers regarding the strategy’s ease of use and usefulness and to determine their statistical significance. By conducting both qualitative and quantitative analyses, the study could obtain a more comprehensive understanding of the strategy’s effectiveness and the software engineers’ perceptions of it.

The results of the qualitative analysis conducted by software engineers in this research are shown in Table 5. Regarding Task 1 of the study, which involved identifying design scenarios and the effectiveness of the software engineers in this task was measured as the quotient of the number of correct design scenarios identified by the total number of design scenarios in the key solution (Effectiveness-FM-Design). The reported result is that the participants were able to correctly identify 77.7% of the total evolution scenarios in Task 1. To determine the effectiveness of Task 2 of the empirical study, we compared the correct updates in the stakeholder feature model, using feature model refactoring, with the ones in the key solution (Effectiveness-FM-Re-Update). The results show that this variable has a mean of 0.8055, meaning that the software engineers were able to correctly evolve 80.5% of the total requested updates (80.5% out of 100). For the purpose of assessing the effectiveness of Task 3, we compared the correct updates in the stakeholder feature model, using the insertion operator, with the updates written in the key solution (Effectiveness-FM-In-Update). The results show that this variable has a mean of 0.8611, meaning that the users were able to correctly update 86.1% of the total updates in the stakeholders feature model (86.1% out of 100).

In addition, we measured the time spent and estimated efficacy for each task. The results indicate that Task 1 was completed by the software engineers in approximately 24 minutes, with an Efficiency-FM-Design value of 0.19 out of 0.25 (number of correct design scenarios / time). For Task 2, the participants
took around 5 minutes to complete the task and achieved an Efficiency-FM-Re-Update value of 0.48 out of 0.60 (number of right updates of the stakeholder feature model using feature model refactoring / time). This value suggests that they were able to correctly update the stakeholder feature model using feature model refactoring almost about two-thirds of the time during the task. For Task 3, the participants took around 4 minutes to complete it and achieved an Efficiency-FM-In-Update value of 0.65 out of 1 (number of right updates of the stakeholder feature model using the insertion operator / time). This value indicates that they were able to correctly update the stakeholder feature model using the insertion operator about two-thirds of the time during the task.

Overall, these results suggest that the participants had a moderate level of proficiency in updating the stakeholder feature model using both feature model refactoring and insertion operator. However, further analysis may be needed to determine if the participants’ performance was affected by any other factors such as prior experience or training in updating the stakeholder feature model.

We repeated the analysis for the same tasks to estimate the efficiency of the software engineers while performing the tasks, but this time we asked the software engineers to perform the tasks using the stakeholder management process.

Table (5) The results of the objective dependent variables for both Mean and Standard Deviation.

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Mean</th>
<th>Max</th>
<th>Standard Deviation (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness-FM-Design</td>
<td>0.7777</td>
<td>1</td>
<td>0.1641</td>
</tr>
<tr>
<td>Effectiveness-FM-Re-Update</td>
<td>0.8055</td>
<td>1</td>
<td>0.1716</td>
</tr>
<tr>
<td>Effectiveness-FM-In-Update</td>
<td>0.8611</td>
<td>1</td>
<td>0.1716</td>
</tr>
</tbody>
</table>

Average = 0.8148
The results show that the software engineers took around 30 minutes to complete **Task 1**, with an **Efficiency-FM-Design** value of 0.16. The software engineers took around 7 minutes to complete **Task 2**, with an **Efficiency-FM-Re-Update** value of 0.30. Finally, the software engineers took around 8 minutes to complete **Task 3**, with an **Efficiency-FM-In-Update** value of 0.41. Figure 10 shows a chart that compares the proposed strategy with the normal strategy in terms of efficiency, while the software engineers perform the tasks related to the empirical study. In total, the comparison reveals that the software engineers from both groups (BAU and Pioneers Company) performed the tasks related to the proposed strategy more efficiently than using the normal strategy. The analysis presented in Table 5 and the comparison results depicted in Figure 10 offer answers to research question1 (Q1) and research question2 (Q2).

Figure 11 presents a case of task-performance measurement related to an urgent request to Pioneer Company to engage a new stakeholder in the software development process. For comparison purposes, we called the stakeholder management process presented in subsection 2.1 the normal strategy.  

---

3 For comparison purposes, we called the stakeholder management process presented in subsection 2.1 the normal strategy.
Figure (11) A case of task-performance measurement: (1) add a new stakeholder using the normal strategy, and (2) add a new stakeholder using the proposed approach. Task completion time for adding new stakeholder in the stakeholder management process.

product. We asked the software engineers to engage the stakeholder using both the normal and the proposed strategy, and then we computed the completion time of each software engineer. Task completion times for adding a new stakeholder during the test trials were recorded for more detailed postanalysis. Since task-performance measurement is only meaningful when compared to the normal strategy (reference case), two measurements were made between the normal strategy and the proposed strategy, and statistical analysis was then applied to derive meaningful and significant differences between the two measurements. It’s worth mentioning that for accuracy purposes, both types of experiment trials were run over a long period of time to assess the ease of learning, and task performance was measured over weeks to see how quickly the user recalls how to operate the strategy and produce higher performance. The result of the task performance assessment supports the answers to research questions 1 (Q1) and research question (Q2) and, at the same time, presents an answer to research question 3 (Q3).

The result presented in Figure 11 reveals that the performance of the software engineers improved for both strategies over the course of the trials. The software
engineers produced higher performance in Trial 2 compared to Trial 1, and they
produced the highest performance in Trial 3. The result gives an indicator of the
learnability of the strategy, which allows software engineers to quickly become
familiar with it and make good use of all its features and capabilities. The
result also indicates that the software engineers produced higher performance
when they used the proposed strategy over the normal strategy. This result also
supports the answer for question 3 (Q3)

Regarding the survey, we asked the software engineers to fill in the NASA
TLX questionnaire (Figure 9) on a scale from 1 (very low) to 7 (very high).
With respect to question 1 (Mental Demand), the software engineers gave 5
times 2, 5 times 3, and one time 4, where the best answer for this question is
1. The software engineers gave a response to question 2 (Physical Demands)
as follows: 5 times 2, 5 times 3, and one time 4, where the best answer for this
question is 1. For question 3 (Temporal Demand), the software engineers gave 4
times one, 6 times 2, and 2 times 3, where the best answer for this question is 1.
Concerning question 4 (Performance), the software engineers gave 3 times 5, 5
times 6, and 3 times 7, where the best answer for this question is 7. Regarding
question 5 (Effort), the software engineers gave 3 times one, 4 times 2, 4 times
3, and one time 4, where the best answer for this question is 1. Finally, the
software engineers gave 5 times one, 6 times 2, and one times 3 for question 6
(Frustration), where the best answer for this question is 1. As shown in Table 6,
the scores were all positive and closer to the best answers. The scores (6 times
3 and one time 4 of question 1) were due to some difficulties in adaptability
with feature model concepts from software engineers. The scores (5 times 3
and one time 4 of question 2) and (4 times 3 and one times 4 of question 5)
were due to clear demands from software engineers to use a tool that supports
our strategy. The scores on the remaining question (i.e., questions 3, 4, 5, and
6) gave an indicator that the software engineers produced high performance
using the proposed strategy. The scores give a good insight into whether the
proposed strategy is easy to use and useful. As a consequence, we reject both
null hypotheses and accept the alternative hypotheses (H1E and H1U), which
Table (6) The scores of the participants from the NASA TLX.

<table>
<thead>
<tr>
<th>Software engineer</th>
<th>Mental Demand (BA=1)</th>
<th>Physical Demand (BA=1)</th>
<th>Temporal Demand (BA=1)</th>
<th>Performance (BA=7)</th>
<th>Effort (BA=1)</th>
<th>Frustration (BA=1)</th>
<th>TLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>2.</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>3.</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>5.</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>6.</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>7.</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>8.</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>9.</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>10.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>11.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>12.</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
</tbody>
</table>

proves that the proposed strategy is perceived as easy to use and useful.

6. Threats to Validity

This section addresses the validity of the study, which indicates how reliable the findings are and how much they are unaffected by the researchers’ subjective viewpoint. In this study, we discussed validity not just in the analysis stage but also in all earlier stages of the study. In this section, validity is presented in three aspects: construct validity, internal validity, and external validity.

**Construct Validity:** This aspect of validity reflects the extent to which the operational measures studied accurately reflect what the researchers had in mind and what is being investigated in accordance with the research questions. For example, the fact that the interview questions of the case study are not interpreted in the same way by the researchers and the software engineers is a threat to construct validity. To avoid this threat to validity, in our study, we interviewed 12 software engineers that have experience with the context of this study, specifically the stakeholder management process and requirement engineering. To the best of our knowledge, these software engineers possess the ability and domain knowledge necessary to mitigate the threat further, and those who are unfamiliar with the field have already been excluded.
**Internal Validity:** This aspect of validity is of concern when investigating causal relationships. When the researcher is investigating whether using the proposed strategy affects the performance of the software engineers, there is a risk that the investigated factor is also affected by the software engineers’ intrinsic backgrounds or interests. A few provisions can be made to reduce such biases; since it was difficult to use a large number of subjects (e.g., more than 30 people), we used a seven-point scale to answer the questionnaire. While the same danger exists with respect to environmental biases, we mitigate this by choosing subjects from two different working environments (software engineers at BAU and Pioneers Company).

**External validity:** This aspect of validity examines the extent to which it is possible to generalize the findings of this research as well as the extent to that the findings are relevant to software engineers outside the case study. To mitigate this kind of validity, the researchers analyzed the applicability of the findings to other cases. For that, the researchers apply a random selection for the participant of case study from a fair sample population of software engineers that satisfies the conditions of research work. Moreover, the authors plan to evaluate the strategy in other software development domains and with larger participants’ population.

### 7. Related Work

Stakeholders are essential to software engineering and can be involved at any phase of product development process. For that reason, much research has been conducted to investigate this topic and provide methods and solutions to improve the efficiency of the stakeholders’ involvement in the product development process. The authors of [3] confirmed that improper stakeholder selection during the requirement elicitation phase can be the primary cause of software project failure rates that are extensively high. In order to tackle this problem, the authors developed a model for identifying appropriate stakeholders throughout the requirements elicitation process based on criteria such as stakeholder role,
knowledge, interest, and communication competence. The results of the statistical analysis revealed that all independent variables were statistically significant, with the exception of stakeholder interest. In addition, the results indicate that selecting stakeholders with the proper attributes has a substantial impact on the phase of requirement elicitation. The results also indicate that the phase of requirement elicitation has a substantial effect on requirement quality. Furthermore, this model assists project managers in determining which stakeholders should be chosen based on their characteristics throughout the phase of requirements elicitation. The result of this study is one of the studies considered by our research in order to construct the stakeholder feature model with suitable stakeholder attributes (referred to as characteristics in [3]).

Razali and Anwar [27] emphasized the significance of having the right stakeholders during the requirement engineering phase, as mistakes committed during this phase can result in project failure. Through a content analysis of the chosen literature, the elements of a successful stakeholder selection procedure were determined. The elements were developed as a procedural framework for the methodical selection of stakeholders that includes three steps: identifying, filtering, and prioritizing. The first phase classifies the project stakeholders according to their roles, stakeholder types, and project definition. The second phase assesses the mental ability of the stakeholders, meaning their knowledge and interest. In the last phase, stakeholders are selected based on their interpersonal abilities. The framework might serve as a guide for future research by highlighting the crucial features that require additional study. This study also influences our research about the types of stakeholders that might be considered by our strategy and the selection and ranking (priority) of stakeholders, which contributes to the development of the stakeholder feature model.

McManus [29] confirmed that successful software engineering projects result when stakeholders care about the issues and are aware that their ideas, opinions, and contributions are valued. The author also claimed that the effective execution of software engineering projects is directly tied to how the project manager includes stakeholders in the decision-making process throughout the
project’s various stages. Thus, the author analyzed and discussed the amount to which stakeholders participate in the project management process and the extent to which stakeholders impact it. The findings of this study indicate that the existing literature devotes considerable attention to identifying stakeholders and assessing the extent and quality of stakeholder participation but very little attention to assessing the costs and benefits of stakeholder participation or the effect of stakeholder participation on performance, outcomes, and sustainability.

Caputo [13] attempted to examine the stakeholder management system in the real estate industry, taking into consideration theories on negotiations and decision-making processes. They conduct a comprehensive evaluation of studies from real estate and management literature in order to organize what they already know about stakeholder management in real estate development projects. According to [13], an external stakeholder should do the following: (1) identify external stakeholders; (2) estimate external stakeholder needs and interests; (3) consider the potential impact these can have on project decisions; and (4) evaluate project implementation solutions while respecting stakeholders’ interests. The authors also concluded that stakeholder analysis is considered a dynamic process that influences the nature of the stakeholders’ impact. In addition, groupings of stakeholders and the nature of their effect may change substantially over time, necessitating the establishment of an iterative method for addressing this matter. The systematic approach to stakeholder management enables managers to organize and assess information on stakeholders and their influence on project decisions. Therefore, their impact also depends on the expressed demands of interested parties and the level of satisfaction they obtain, without compromising the project’s primary purpose. Finally, a comprehensive stakeholder management theoretical model was established.

Polonsky and Michael [36] examined how marketers may employ the stakeholder management process and the stakeholder matrix when developing marketing strategies. Although most marketing theories implicitly recognize the necessity of designing strategies that satisfy the demands of multiple groups or stakeholders, the author contends that present marketing theories do not
employ the stakeholder management approach when doing strategic marketing planning. Marketers should be able to design more effective marketing strategies by knowing and utilizing stakeholder theory. While management literature on stakeholder theory is rich, marketing material on this topic is scarce, thus, the author of [36] seeks to partially address this gap. Our strategy also supports the employment of stakeholder management processes when developing a software product by using an evolvable artifact called the stakeholder feature model.

Sutterfield et al. [37] indicated that stakeholder theory is a valuable framework for evaluating the behavioral aspects of the project management process, especially the complex project management process within the Department of Defense (DOD). The authors also believe that the agendas of diverse stakeholders within the organizational structure might pose obstacles to projects. When this occurs, a solid stakeholder management plan must be implemented to maximize the likelihood of project success. This is a case study of a failed Department of Defense initiative that was properly justified and urgently required. This case analysis explores the potential causes of the project’s failure by applying stakeholder theory as its theoretical foundation. The failure-learned project management lessons and a project stakeholder management strategy framework are offered to aid project managers in making better decisions and raise the likelihood of successful project management outcomes.

8. Conclusions and Future Work

The stakeholder management process aids in maintaining beneficial relationships with those who have the most influence over product development. In conclusion, effective communication with each individual can play a vital role in keeping people efficiently engaged in the development of the software product. In this paper, a strategy has been proposed to support the systematic stakeholder management process. The strategy is based on feature modeling for capturing, documenting, and monitoring the involvement of suitable stakeholders during the stakeholder management process. This involvement is based on the stake-
holders’ characteristics (attributes), such as the stakeholder’s Area, Experience, Role, Goal, and Interest. The strategy uses an evolvable artifact called the stakeholder feature model that stores and presents the potential stakeholders and their attributes for a specific software product. These attributes are identified based on a review of the literature on the effective selection of stakeholders. The attributes were conceptualized as features of the stakeholder feature model, and attributes of features are used to support the best methods of communication with the stakeholders. The evolvability of the stakeholder feature model is implemented using two basic operations defined for feature modeling, namely feature model refactorings and the insertion operator. To evaluate our strategy, we have conducted a case study in the Smart Home development domain. The results reveal that the strategy has a significant influence on the quality of the stakeholder management process.

There are still many aspects of our strategy that could be improved. For example, to build a stakeholder feature model to deal with the conflicts in stakeholders’ requirements. Furthermore, the strategy needs to be assessed through a tool in order to improve its accuracy and productivity.

Acknowledgment

We would like to thank the students of BAU and the development team of Poineers Company. The case study will not be complete without their continuous support and cooperation.

References


[2] I. Hristov, A. Appolloni, Stakeholders’ engagement in the business strategy as a key driver to increase companies’ performance: Evidence from man-


