A Unified Feature Model for Scrum Artifacts from a Literature and Practice Perspective

Luciano A. Garcia[1], **Edson OliveiraJr**[1], **Gislaine Camila L. Leal**[1], **Marcelo Morandini**[2]

[1]Informatics Department, State University of Maringá (UEM)
Maringá, Brazil.
[2]School of Arts, Sciences and Humanities,University of São Paulo (USP),
São Paulo, Brazil.

lucianogarcia11@hotmail.com, edson@din.uem.br, gclleal@uem.br, m.morandini@usp.br

Abstract. *Scrum has become one of the most popular Agile methods. Among its main elements are its artifacts. These artifacts are related to the requirements required for the software and how they will be worked on during a Scrum interaction called Sprint. Given the importance of artifacts in the Scrum structure, evidence of the adaptations of these artifacts was collected with the aid of a systematic mapping study and a survey literature with practitioners of the method. Later, we systematized the evidence of adaptations found and built models of features in order to register them and enable users of the methods to have a broader understanding of the features that Scrum artifacts can assume.*

1. Introduction

There has always been a search for productivity and quality in software development, which is particularly evident in the agile manifesto[1] [Schwaber and Sutherland 2017], as opposed to traditional software development processes oriented to documentation that until then were the most accepted. Therefore, methods that were already known became popular, among them Scrum.

Scrum is an interactive and incremental approach that replaces the phases of the traditional software development process with the delivery of a high-value suite, which provides an early return of successes and errors in the development of the respective software [Schwaber and Sutherland 2017]. According to the authors of Scrum [Schwaber and Sutherland 2017], the main components of Scrum are the roles, events, artifacts and rules that unite and predict the interaction among them.

In this paper, we show the Scrum artifacts. Scrum artifacts are related to the requirements needed for the software to be developed, which and how these requirements will be developed in Scrum interactions and finally a functional version of these requirements.

In view of the importance of artifacts for Scrum and, consequently, for software development, we seek evidence of adoption and adaptation in the literature and in the experience of practitioners in the use of Scrum, thus we try to answer the following research question "***How have been Scrum artifacts adapted throughout actual software***

[1]https://agilemanifesto.org

development projects?". To this end, we use information about the artifacts that are part of a Systematic Mapping Study (SMS) of Scrum practices and complement it with information obtained through a Survey with Scrum practitioners. To organize and relate the evidence found for the Scrum artifacts in the SMS and the Survey, we used features models [Czarnecki et al. 2005].

2. Background and Related Work

In this section we present the main theoretical concepts to support our proposal.

2.1. The Scrum Framework and Artifacts

According to the authors of Scrum in [Schwaber and Sutherland 2017], Scrum aims to develop, deliver and maintain complex products. It is defined as a framework in which people approach complex and adaptive problems in a productive and creative way to deliver products with the highest possible value.

Scrum uses an iterative and incremental approach to improve predictability and risk control. Scrum is made up of Scrum teams linked to roles, events, artifacts and rules. Each component has a specific purpose and is essential for the use and success of Scrum. Although there are many elements involved in the Scrum dynamics, we emphasize the Scrum artifacts in this paper.

The Scrum artifacts are designed to maximize the transparency of information, thus everyone has the same understanding of what is actually done. Therefore, Scrum is composed of three main artifacts, namely: Product Backlog, Sprint Backlog and Increment.

The **Product Backlog (PB)** is an ordered list of everything needed in the product, in the case of software that will be developed. It is the only source of requirements and software changes. The PO is responsible for the PB, including, updating and ordering its content. The PB and its items must be visible to all stakeholders, assimilating the pillar of transparency preached by Scrum.

The Sprint Backlog (SB) consists of a subset of PB items that were selected by the Dev. Team to be developed, taking into account their priority and the Dev. Team's development capacity. This activity should take place at the Sprint Planning event. Seeking to comply with the Scrum transparency principle, the SB should always be visible to stakeholders, identifying which of its items are ready, then in progress, and which have not yet started. The SB is a source of work to be performed by the Dev. Team in a sprint cycle. As the sprint is being executed, new tasks can be identified to complete the PB items that were selected for the SB, and they must be included in the SB. It is also necessary to add at least one improvement item identified in the Sprint Retrospective event, in order to have a continuous improvement of the process.

The **Increment** is the result of the PB items that were selected for Sprint and that have become a functional version of the product to be delivered. They were inspected in the Sprint Review and released by the PO.

2.2. Feature Modeling

According to [Czarnecki et al. 2005] feature is a system property that is relevant to some stakeholders (customers, analysts, architects, developers, system administrators, etc.) and is used to capture similarities or differences between systems in a family. The features are organized in diagrams, in the shape of a tree, where the root represents a concept (such as a software system). The diagrams added with descriptions of resources, relationships, priorities, stakeholders, etc., form what is defined as feature models. A feature model is a relationship among a parent feature and its daughter features [Czarnecki et al. 2005].

A feature model is composed of some basic elements, which are: feature diagram, composition rules and relational analysis. In addition, the feature models follow the Czarnecki-Eisenecker notation [Czarnecki and Eisenecker 1999] and the FeatureIDE Tool[2], to elaborate the feature models of this paper.

Figure 1 presents the main elements involved in a feature model and which will be detailed below.

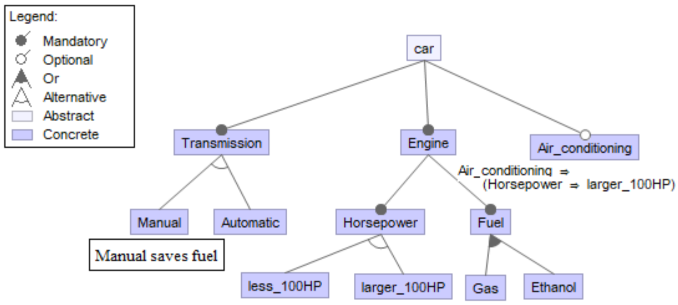

Figure 1. Example of a feature model based on the [Kang et al. 1990] notation

Feature Diagram. In general, the features are organized in the feature diagrams in the form of trees [Sochos et al. 2004] as shown in Figure 2. The features are represented by the tree nodes described in the feature diagram, and in this hierarchy child features can be classified as: *mandatory* - the daughter characteristic must be selected; *optional* - the child feature may or may not be selected; alternative (OR) - at least one of the child features must be selected; and Exclusive OR (XOR) - only one of the child features must be selected.

A feature can also be defined as a concept, and in this case called abstract, this can be seen in Figure 1, for the node where the *car* is written and the others observed in the figure are concrete features.

Composition Rules. They define the relationship between features that cannot be expressed in the features diagram, indicating which combinations of features are valid. In Figure 1, a composition rule is required under the feature *Ar_conditioning* that requires the car to have an engine with a power greater than 1000 in order to support the air conditioning.

Relational Analysis. It is a recommendation specifying when a particular feature should or should not be selected. In the example contained in Figure 1 for the definition of a car, below the *manual* feature there is information that recommends that the choice of a car with the manual transmission tends to be more economical in terms of fuel.

[2] `http://www.featureide.com`

2.3. Related Work

To the best of our knowledge and based on a non-systematic search there is no related work aimed at modeling the adaptations of Scrum artifacts using feature models.

However, Diebold et al. [Diebold et al. 2015] present how practitioners have adapted Scrum in 10 German software projects. They claim such adaptations occur in the Sprint length, events, team size, and requirements engineering. Practitioners also varied the roles, effort estimations and quality assurance. Certain adaptations come from a previous hierarchical non-agile organization, thus many of them are for good reasons. We corroborate a significant part of the results regarding artifacts adaptations.

3. Adaptations for Scrum Artifacts

To find the evidence regarding the adaptations of the Scrum artifacts in the literature and in the experience of the practitioners, we used clippings from a Systematic Mapping Study (SMS) and also part of a Survey conducted with Scrum practitioners. In the next topics we show the main information related to the SMS and the Survey regarding the Scrum artifacts.

3.1. Adaptations from the Literature

The literature reports different studies on the use of Scrum in the most varied domains and situations. We sought to carry out a Systematic Mapping Study (MSL) by primary studies that revealed elements or features of Scrum adjusted when an organization decides to adopt it.

SMS planning followed the recommendations of Petersen et al. [Petersen et al. 2015]. The SMS had a wide search without date restriction in 5 electronic databases and a manual search in 11 journals and 17 conferences with a date restriction from 07/2007 to 07/2017, in other words, 10 years. Through this process, 281 primary studies were related, in which the inclusion and exclusion criteria defined in [Garcia 2019] were applied, and which resulted in 50 studies selected for the extraction of information. More information about SMS is available at [Garcia 2019] and at `https://doi.org/10.5281/zenodo.3357803`.

Our SMS focuses on the Scrum artifacts, thus the research questions defined are: **RQ1** - What were the Scrum artifacts adapted?; **RQ2** - Do the artifacts follow the recommendations in the Scrum guide?; and **RQ3** - Which techniques were used to prioritize and organize and estimate items in Scrum artifacts?

All studies answering the survey are shown in Table 1.

Table 1. Studies with adaptations in Scrum artifacts.

Study ID	PB	SB	INC
S2, S4, S13, S14, S28, S32, S36, S37, S38, S41, S43, S44, S50, S52, S58	X		
S5, S6, S15	X	X	X
S9, S11, S12, S26, S29, S33, S48, S49	X	X	
Total	**26**	**11**	**03**

Table 1, which answers the research question **RQ1**, shows that of the 50 selected studies, 26 presented information about Product Backlog (PB), 11 about Sprint Backlog (SB) and only 3 about Increment (INC).

Regarding the question **RQ2**, it can be seen that the recommendations of the Scrum guide were only partially followed in the studies. This happened in 19 of the 26 studies, including the following: S4, S5, S6, S9, S11, S13, S12, S14, S15, S28, S29, S33, S36, S37, S38, S41, S44, S48, S50.

To answer the **RQ3** question, only studies that partially met the Scrum recommendations were considered, since the others do not. However, not all studies in partial compliance brought the necessary information to answer this research question. With regard to the **INC** artifact, which is shown in Table 1, the 3 related studies only mentioned the artifact and did not bring any relevant information to MSL. We then analyzed only the other two artifacts **PB** and **SB** for this research question, that are in Table 2.

With regard to PB the information found in the studies was systematized in Table 2. In it one can see a column called Feature ID, which represents an identifier for the information / feature found in the study. The identifier has the following notation: ***XXxnX***. Where: ***XX*** represents the artifact (PB, SB); ***x*** what kind of feature (e - how to estimate, p - how to prioritize and r - how to represent); ***n*** is a sequential number for the feature; ***X*** identifies the origin of the feature (M-SMS, S-Survey)

Table 2. SMS Information for PB and SB.

Study ID	Information	Feature ID	Study ID	Information	Feature ID
Product Backlog			**Sprint Backlog**		
S4	Prioritize: PO and Dev Team worked together Representation: User Stories	PBp1M PBr1M	S6	Selection: Items selected for SB by Dev. Team	SBs1M
S5, S28	Estimate: Planning Poker	PBe1M	S11	Status: SB had the tasks framed in the following status: Not Started, In Progress, Completed, Blocked	SBt1M SBt2M SBt3M SBt4M
S6	Prioritize:Screening meeting	PBp2M	S33	Progress:features to identify task progress: PB identifier to which the task belongs, Completion time	SBp1M SBp2M
S9	Prioritize: PO with SM and Dev. Team help Representation: User Stories	PBp3M PBr2M			
S11	Prioritize:PO didn't prioritize Estimate: SM and Dev Team	PBp4M PBe2M			
S12	Prioritize:Only the PO Representation: User Stories	PBp5M PBr3M			
S13, S14	Prioritize: PO and Dev Team worked together	PBp6M			
S29	Prioritize: Only the PO Estimate: PO and Dev Team Representation: User Stories	PBp7M PBe3M PBr4M			
S33	Estimate:Experienced professionals	PBe4M			
S36, S44	Estimate:Planning Poker Representation: User Stories	PBe5M PBr5M			
S37	Estimate: Planning Poker Representation:Use case	PBe6M PBr6M			
S38	Representation: User Stories	PBr7M			
S41	Estimate:Story Points and Value Points Representation: User Stories	PBe7M PBr8M			
S48	Prioritize: LOEs visual representation of software priorities of the customer and what they want in the final state.	PBp8M			
S50	Prioritize:Refinery requirements.	PBp9M			

Looking at Table 2 , the information regarding the **RQ3** question was systematized in the who prioritized, how was estimated and how was represented PB.

Still responding to **RQ3**, now with respect to SB, the 11 studies presented in Table 1 were analyzed, but only 3 studies returned information capable of indicating adaptations for this artifact (see Table 2).

As can be seen in Table 2, the information was systematized for the SB as follows: (i) who selected the items of the PB to be worked on in the SB; (ii) which status the tasks within the SB assumed during its development and (iii) how to monitor the progress of the tasks. The Feature ID column in Table 2, also followed the notation defined for the features in Table 2 (***XXxnX***), with a small difference for ***x***, where it assumed the following values: s -selection, t- state, p - progress.

With the information listed in Tables 1 and 2, the feature model illustrated in Figure 2 was elaborated.

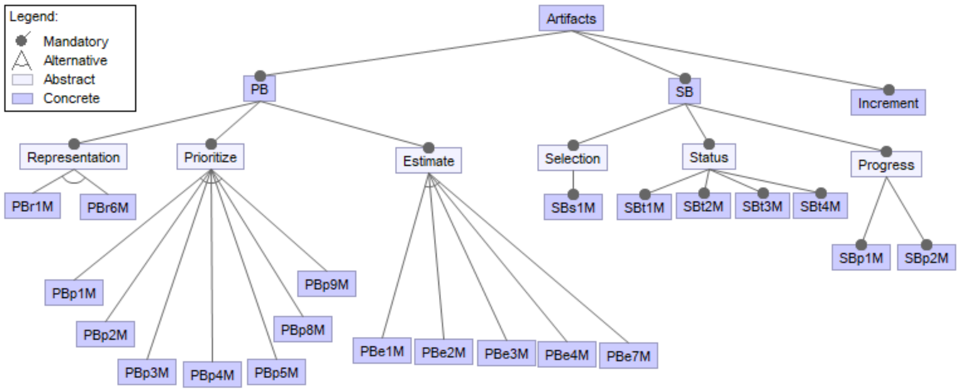

Figure 2. Feature Model for Scrum Artifacts based on the SMS

In the elaboration of the feature model in Figure 2 , some features were removed because they represent the same information for the parent features: Representation, Prioritize and Estimate. For the PB representation form, only the *PBr1M* and *PBr6M* features were considered. The others were the same as the *PBr1M* feature. For the way of prioritizing PB, features *PBp6M* and *PBp7M* were dispensed because they are equal to features *PBp1M* and *PBp5M* respectively. Regarding the way to estimate the PB, the *PBe5M* and *PBe6M* features were discarded because they are the same as the *PBe1M* feature. Regarding the SB, no treatment needed to be done.

3.2. Adaptations from the Practitioners Survey

The evidence of adaptations presented for the Scrum artifacts, in relation to the practitioners, is part of a broader Survey that was conducted, which includes the following steps: Planning, Pilot Test, Data Collection and Analysis of Results.

The survey in question had an international scope in relation to practitioners. A total of 14 respondents were obtained, who demonstrated having a profile with good knowledge, when asked how much experience they had with the use of Scrum in software development. Of these respondents, 21.4% reported having more than 6 years of experience with Scrum, 35.7% between 3 and 6 years of experience, 35.7% between 1 and 3 years of experience and only 7.1% reported having less 1 year of experience with Scrum in software development.

The information found with the SMS helped us to elaborate the research instrument (questionnaire) that was applied to the respondents. More information about the Survey conducted can be verified in [Garcia 2019], we will focus here in relation to the data obtained from the respondents for the Scrum artifacts, due to space limitations.

The answers to the questions investigated in the Survey regarding the Scrum artifacts are presented here in a grouped form, regardless of the percentage, as it is enough once mentioned to characterize it as being used, and also due to space limitations. The following questions for the PB were investigated: **RQ1** - What form of representation is used in PB?; **RQ2** - How was PB prioritized?; **RQ3** - What form of estimate is used for BP?; **RQ4** - What is the way to measure work progress in PB?; and **RQ5** - What software was used for the management of PB?.

Table 3 presents the systematized responses obtained by the Survey for the PB. It used the same notation used to identify features in SMS ***(XXxnX)***. The differences are in the ***x***, where *g* has been added for *Progress* and *w* for *Software*. Also the ***X*** that has now assumed the *S* to represent the origin of the information as a Survey.

For SB in the Survey, the following questions were investigated: **RQ6** - Who did select the items for the SB?; **RQ7** - In what states were the tasks classified?; **RQ8** -How was the progress of tasks in the SB monitored?; and **RQ9** - What softwares were used to manage the SB?.

Table 3. Survey information for PB and SB.

RQ	Information	Feature	RQ	Information	Feature	RQ	Information	Feature
Product Backlog			**Sprint Backlog**			**Increment**		
RQ1	Representation: User History	PBr1S	RQ6	Selection: PO	SBs1S	RQ10	Test: Test Driven Development (TDD)	INCe2S
RQ1	Representation: Use Cases	PBr2S	RQ6	Selection: PO and Dev.Team	SBs2S	RQ10	Test:Integration Test	INCe1S
RQ2	Prioritize: PO and Dev. Team	PBp1S	RQ6	Selection: Dev.Team	SBs3S	RQ10	Test:Ad-hoc (defined in Ticket Jira)	INCe3S
RQ2	Prioritize:PO alone	PBp2S	RQ6	Selection: SM	SBs4S	RQ10	Test:Unit Testing	INCe4S
RQ2	Prioritize: PO and SM	PBp3S	RQ7	Task - Status: Concluded, In progress, Blocked, New, Open, Classifying, Draft, Verified, To do, To test Testing	SBt1S SBt2S SBt3S SBt4S SBt5S SBt6S SBt7S SBt8S SBt9S SBt10S SBt11S	RQ10	Test:User Note	INCe5S
RQ3	Estimate:Planning Poker	PBe1S	RQ8	Task - Progress:Monitoring task board	SBg1S	RQ10	Test:Manual with some automation	INCe6S
RQ3	Estimate:Story Points and Value Points	PBe2S	RQ8	Task - Progress:Burndown Sprint Chart	SBg2S	RQ10	Test:Team Demonstration	INCe7S
RQ3	Estimate:Story Points	PBe3S	RQ8	Task - Progress:Weekly report	SBg3S	RQ10	Test:User Acceptance Test	INCe8S
RQ3	Estimate: Story Points and Value Points with Planning Poker	PBe4S	RQ8	Task - Progress: Board Jira	SBg4S	RQ11	Version:Used field in Jira and Jira Releases	INCv1S
RQ4	Progress: Performed in Sprint Planning or pre-sizing	PBg1S	RQ8	Task - Progress: Visual Board	SBg5S	RQ11	Version:ServiceNow Update Sets	INCv2S
RQ4	Progress: Risk and Impact Techniques	PBg2S	RQ9	Software:ServiceNow	SBw1S	RQ11	Version:From Visual Studio	INCv3S
RQ4	Progress: Burndown Release Chart	PBg3S	RQ9	Software:Readmine	SBw2S	RQ11	Version:Crashlytics	INCv4S
RQ5	Software: Azure Devops	PBw1S	RQ9	Software:Jira	SBw3S	RQ11	Version:Version Control by itself XX.YY.ZZZ	INCv5S
RQ5	Software: TFS	PBw2S	RQ9	Software: Azure Devops	SBw4S	RQ11	Version:GitHub	INCv6S
RQ5	Software: Trello	PBw3S	RQ9	Software: RTC	SBw5S	RQ12	Quality:Sprint Review	INCq1S
RQ5	Software: Jira	PBw4S	RQ9	Software: Microsoft Team Foundation Server	SBw6S	RQ12	Quality:Evaluated by the PO	INCq2S
RQ5	Software: ServiceNow	PBw5S	RQ9	Software: Version One	SBw7S	RQ12	Quality:Acceptance test by PO	INCq3S
RQ5	Software: Product Roadmap	PBw6S				RQ12	Quality:By the QC testing team	INCq4S
RQ5	Software: RTC	PBw7S				RQ12	Quality:Functional and unit tests	INCq5S
RQ5	Software: Microsoft Foundation Server	PBw8S				RQ12	Quality:Search with users	INCq6S
RQ5	Software: MS	PBw9S				RQ12	Quality:QA tested and verified problems	INCq7S
RQ5	Software: Version One	PBw10S				RQ12	Quality:By defect density	INCq8S
RQ5	Software: Rally	PBw11S				RQ12	Quality:Definition of Done, static code, others	INCq9S

Regarding the Increment (INC), the following questions were investigated: **RQ10** - What testing techniques are used for the INC?; **RQ11** - Which Version control is used for the INC?; and **RQ12** - How was the quality of the INC assessed?

Table 3 systematizes the answers found for the questions investigated for the INC. The notation used to represent the features for the INC follows the same idea as the one used for PB and SB, with some differences, which are: ***XXX*** to represent the INC artifact, ***x*** that assumes (e -test, v-version and q -quality).

Based on Table 3, the characteristics model was elaborated with the information from the Survey for Scrum artifacts. The model is shown in Figure 3.

3.3. Threats to Validity

Our main threats to this study are: small samples in the SMS and the survey, which might jeopardizes statistical significance; lack of quality evaluation of the primary studies of the SMS; and the interpretation of the adaptations might be biased in a certain way due to the expected results from the researchers point of view.

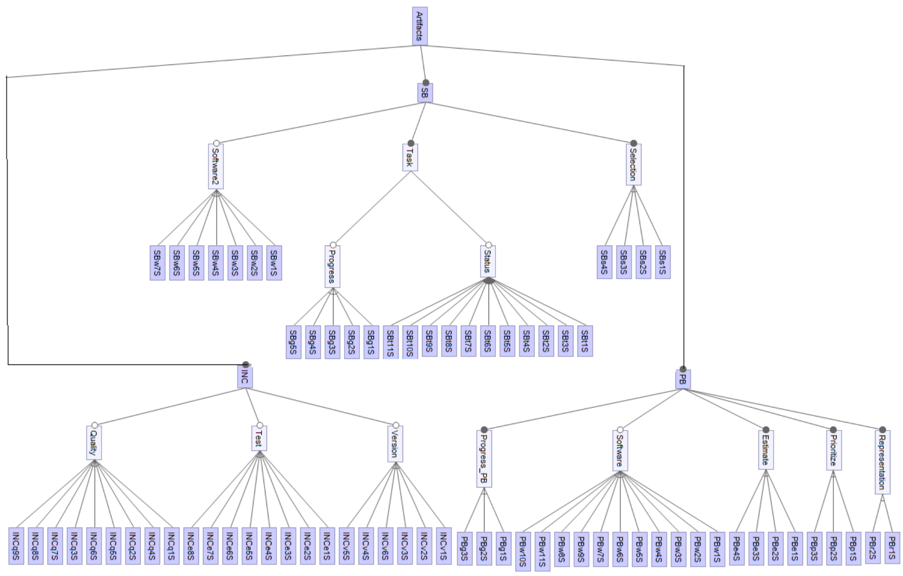

Figure 3. Feature Model for Scrum Artifacts based on the Survey

4. Feature Unification Process

In this section we describe the process of unifying the models of features obtained for the SMS and the Survey.

4.1. Scrum Guide Compliance Check

Product Backlog (PB). The information regarding the PB in the SMS and that appears in Table 2 was compared with the Scrum guide and no inconsistencies were found in them. Regarding the PB data in the Survey, which are shown in Table 3, they were compared with the Scrum guide and there were also no inconsistencies.

Sprint Backlog (PB). Regarding the SMS, the information found for the SB and shown in Table 2, there was no inconsistency with the Scrum guide. For the SB information in the Survey and shown in Table 3, we found inconsistencies in the following features for the *Selection* of items for SB , in relation to the Scrum guide:*SBs1S*, *SBs2S* and *SBs4S*. The Scrum guide states that it is the responsibility of the Dev. Team to select the items to compose the SB. Soon, inconsistent features will be eliminated for the unified feature model.

Increment (INC). In the SMS there was no return of information for the Increment so there are no inconsistencies regarding the Scrum guide to be verified. Regarding the Survey, although more information was returned to the INC (see Table 3), no inconsistencies were found in relation to the Scrum guide.

4.2. Elimination of Redundant Features

We eliminated the features common in the SMS and in the Survey for artifacts. We wanted to maintain the features found in the Survey, for cases of repeated/similar features, with no prejudice to information.

Product Backlog (PB). Observing Tables 2 and 3, the following features were found to be the same for PB: *PBr1M = PBr1S, PBr6M = PBr2S, PBp1M = PBp1S,* *PBp5M = PBp2S, PBe1M = PBe1S* and *PBe7M = PBe2S*.Therefore, in the unified features model, only the Survey's original features will be present, which are: *PBr1S, PBr2S, PBp1S, PBp2S, PBe1S* and *PBe2S*.

Sprint Backlog (SB). Looking at Tables 2 and 3, it was found that the following features for the SB are the same: *SBs1M=SBs3S, SBt2M=SBt2S, SBt3M=SBt1S* and *SBt4M=SBt3S*. The features that will be part of the unified model will be those that have the origin of the Survey, that is, those that end with the letter *S*.

Increment (INC). With regard to the Increment artifact, repeated features were not identified since the SMS did not return information for these artifacts, so those that were returned by the Survey were assumed.

After the consistency of the information from the SMS and the Survey was made with the Scrum guide and verification of the redundant features in both models, in this section we show the resulting model with the remaining features. This can be seen in Figure 4.

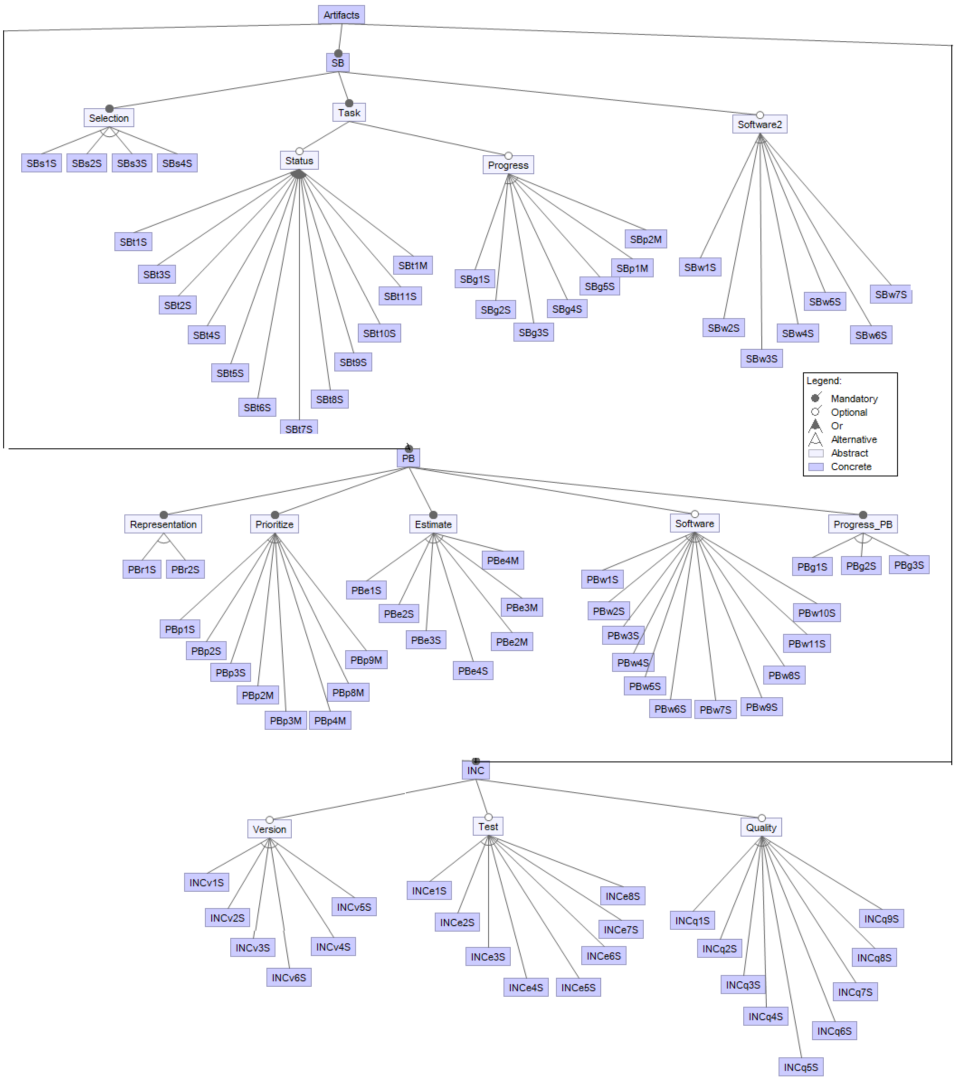

Figure 4. A Feature Model for Scrum Artifacts Adaptations

5. Conclusion

In the information sets we analyzed for the SMS and the Survey, we were able to build models of feature that would meet the recommendations of the Scrum guide in relation to its artifacts. We observed that the analyzed literature did not bring significant information about the Increment artifact, which can be demonstrated in a certain way were a shortage on the subject. But practitioners, when provoked through the Survey, provided interesting information about this artifact such as: tools used to manage PB and SB, version control for Increment, etc.

Through the feature models elaborated in this work, we store and systematize the knowledge about the adaptations of the Scrum artifacts found. Feature models facilitate the identification of relationships between features and also show which are mandatory in the use of artifacts. We do not intend in this work that the unified feature models for Scrum artifacts provide all possible solutions for the use of this component, through its derivation, but that it allows users of the method to have it as a starting point to create their own model or who knows how to improve it. Although we checked the conformity of the features found with the Scrum guide, we did not carry out a practical validation of the proposed models, which appears as an opportunity for future work and which we believe to be important.

References

Czarnecki, K. and Eisenecker, U. W. (1999). Components and generative programming. In *ACM SIGSOFT Software Engineering Notes*, volume 24, pages 2–19. Springer-Verlag.

Czarnecki, K., Helsen, S., and Eisenecker, U. (2005). Formalizing cardinality-based feature models and their specialization. *Software process: Improvement and practice*, 10(1):7–29.

Diebold, P., Ostberg, J.-P., Wagner, S., and Zendler, U. (2015). What do practitioners vary in using scrum? In Lassenius, C., Dingsøyr, T., and Paasivaara, M., editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 40–51. Springer International Publishing.

Garcia, L. (2019). Adaptations of the scrum framework for software development projects. Master's thesis, Universidade Estadual de Maringá.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. (1990). Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.

Schwaber, K. and Sutherland, J. (2017). *The Definitive Guide to SCRUM: The rules of the Game*. Scrum.org. https://www.mitchlacey.com/resources/official-scrum-guide-current-and-past-versions.

Sochos, P., Philippow, I., and Riebisch, M. (2004). Feature-oriented development of software product lines: mapping feature models to the architecture. In *Net. ObjectDays*, pages 138–152. Springer.