A Survey on Software Release Planning Models

David Ameller, Carles Farré(✉), Xavier Franch, and Guillem Rufian

Universitat Politècnica de Catalunya, Barcelona, Spain
{dameller, farre, franch}@essi.upc.edu,
guillemonl3@gmail.com

Abstract. Software release planning (SRP) is the problem of selecting which features or requirements will be included in the next release or releases. It is a crucial step in software development, which happens to be extremely complex given the need to reconcile multiple decision making criteria, (e.g., business value, effort and cost), while considering several constraints (e.g., feature precedencies, resource availability). For this reason, several SRP models have been proposed in the literature. The objective of this paper is to provide an updated review of SRP approaches reported in the literature.

Keywords: Software release planning · Next release problem · Resource optimization · Software evolution · Literature survey · State of the art

1 Introduction

According to Lehman's First Law of software evolution [6], software must be continually adapted or it becomes progressively less satisfactory to be used in its environment. Software changes are usually not implemented all together but incrementally [5]. Major enhancements are planned and incorporated, together with other minor changes, in each new release or upgrade. According to Sommerville [12], the main activities in an evolution loop are feedback, impact analysis, release planning, system implementation, and system release.

Software Release Planning (SRP) is the problem of finding the best combination of features to implement in a sequence of releases. SRP seeks to maximize business value and stakeholder satisfaction without neglecting the constraints imposed by the availability of adequate resources and the existence of dependencies between features, among other constraints [11]. There are several factors that make SRP a computationally complex problem: the number of features and their interdependencies; the number of stakeholders involved, their different levels of priority, and their conflicting interests; the variety of variables to be considered (e.g., business value, effort, cost); and the uncertainty and incompleteness of the available information [10].

Given its importance, many approaches to SRP have been proposed. Svahnberg et al. [14] presented a comprehensive survey of SRP models formulated until 2008. These models were analysed under different perspectives (inputs considered, industrial application, etc.). Since then, other proposals have been formulated.

The goal of this paper is to update the results of the survey [14] by considering these recent approaches to SRP. For attaining this goal, we have searched, analysed and

P. Abrahamsson et al. (Eds.): PROFES 2016, LNCS 10027, pp. 48–65, 2016.
DOI: 10.1007/978-3-319-49094-6_4

discussed the different SRP models that have been proposed in the scientific literature related to this question by performing a snowballing-based literature review.

Following the four perspectives for describing a goal in the Goal-Question-Metric (GQM) approach [2], the goal of our study is defined as follows:

- Purpose: find and characterize
- Issue: the proposed models in the academic literature since 2009
- Object: for software release planning
- Viewpoint: from the viewpoint of project managers and software developers.

The rest of the paper is organized as follows. In Sect. 2, we introduce the research method followed in the paper, with special attention to the snowballing procedure and the research questions. In Sect. 3, we extract the results of the surveyed papers in relation to the research questions. In Sect. 4, we analyse the results and provide the most relevant observations from it. Finally, in Sect. 5 we present the conclusions and future work.

2 Research Method

As stated above, we built on top of the knowledge gained in the Systematic Literature Review (SLR) conducted by Svahnberg et al. [14] that surveyed the SRP approaches proposed until 2008. We proceeded according to the following steps:

- Our research questions were based on those in [14].
- Given its high citation count and prominent venue of publication, we assumed that any sound primary study on SRP models published in the period 2011–2015 has cited [14]. Therefore, we based our search of primary studies on forward snowballing from this paper.
- For the period 2009–2010, we checked the references appearing in the papers found in the forward snowballing. In addition, this reference analysis (backward snowballing) was used also to check any further relevant work in the period 2011–2015.
- Finally, we conducted an expert consultation step and added a few extra references that were considered relevant.

The rest of the section presents in detail the research method applied under these general guidelines.

2.1 Research Questions

In this study, we kept two of the research questions (RQs) from the SLR presented in [14] while discarding the other two that were more specific and related to specific interests of the authors. We also decided to decompose the two remaining RQs into sub-RQs to structure the data collection and analysis.

The most fundamental RQ of our study is: *What SRP models have been presented since 2009? (RQ1).* We decompose it into four sub-questions:

- *RQ1.1. What are the main motivations for the models?*
- *RQ1.2. What are the inputs processed by the models?*
- *RQ1.3. What are the outputs generated by the models?*
- *RQ1.4. What are the algorithms or techniques applied by the models?*

We also want to know *to what extent have the SRP models surveyed in RQ1 been validated? (RQ2).* We have decomposed this RQ into:

- *RQ2.1. Are the models supported by tools?*
- *RQ2.2. How has been industry involved in the models?*
- *RQ2.3. What are the major threats identified on the models?*

2.2 Selection of Studies

As mentioned above, we combined forward and backward snowballing using [14] as starting point plus an additional final check with experts. Snowballing refers to using the reference list of a given paper (backward snowballing) or the citations to the paper (forward snowballing) to identify additional literature [16].

We defined the following inclusion criteria to select the relevant studies:

1. The paper is a full research paper published in any of: JCR-indexed journal belonging to Q1–Q3 quartiles; proceedings of one of the following main software engineering or software evolution conferences: ICSE, ESEC/FSE, ESEM, HICSS, ICSM, ICSME, CSMR; or any JCR-indexed journal, CORE A or B conference/workshop proceedings if at least one of the authors has an industry affiliation. In addition, we included PROFES because it is the most recurrent venue in the papers found by [14], together with IEEE Software.
2. The paper describes an SRP model.

The rationale about the restricted selection of venues in the first criterion is that the most relevant studies should be found published in the most renowned journals and conferences. The reason for which the selection of these venues is relaxed for studies with authors having industrial affiliation is that we want to find as many as possible proposals where a full-scale industrial validation of the proposed SRP model is conducted. We think that, in line with the conclusions and recommendations given in [14], industrial validation is a key issue when assessing an SRP model.

The process was implemented in three different iterations as depicted in Fig. 1. For the *first iteration*, we used both Scopus and Google Scholar to find the references that cite directly [14]. This search was performed on 2 October 2015 and we retrieved 56 references from Scopus, while Google Scholar provided us 101. All the references returned by Scopus, except two, were also in the set provided by Google Scholar. Most of the 47 Google Scholar references not included in Scopus were excluded later on (see application of inclusion criteria below), but a few of them were considered. These ones were references to papers published in major journals as “online first”, without being assigned yet to a concrete journal issue.

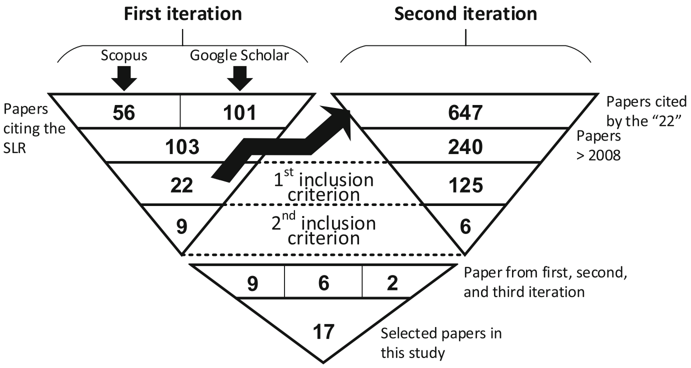

Fig. 1. Papers selected in each iteration.

From the grand total of 103 studies, 81 studies were excluded by the first inclusion criterion. The titles and abstracts of the remaining 22 studies were read to apply the second inclusion criterion. In the cases where that information was not sufficient to make a decision, the full text was also considered [16]. This resulted in the selection of 9 relevant papers listed in Table 1 (see the References section for full details).

Table 1. Papers selected from the forward snowballing iteration.

Ref.	Year	Paper title
M1	2012	A hybrid release planning method and its empirical justification
M2	2011	Quantitative release planning in extreme programming
M3	2013	Multi-sprint planning and smooth replanning: An optimization model
M4	2012	Solving the large scale next release problem with a backbone-based multilevel algorithm
M5	2013	Analyzing an industrial strategic release planning process - a case study at Roche diagnostics
M6	2013	Continuous release planning in a large-scale scrum development organization at Ericsson
M7	2015	Software requirements prioritization and selection using linguistic tools and constraint solvers — a controlled experiment
M8	2014	Industrial evaluation of the impact of quality-driven release planning
M9	2014	Theme-based product release planning: an analytical approach

For the *second iteration*, we used the 22 papers from the first iteration (i.e., the selected papers before applying the second inclusion criterion) and retrieved all their references. We retrieved 647 references overall, without taking into account any inclusion criteria. These references included papers, books, manifestos and programs. For the second iteration we used the same inclusion criteria, with the exception that from the 647 references we only considered those that were published after 2008. Then for the remaining 240 papers we applied the first inclusion criterion. Finally, the titles and abstracts of the 125 remaining papers were read to apply the second inclusion

Table 2. Papers selected from the backward snowballing iteration.

Ref.	Year	Paper title
M10	2009	Software project planning for robustness and completion time in the presence of uncertainty using multi-objective search based software engineering
M11	2010	Rigorous support for flexible planning of product release - a stakeholder centric approach and its initial evaluation
M12	2011	Conceptual scheduling model and optimized release scheduling for agile environments
M13	2015	Differential evolution with Pareto tournament for the multi-objective next release problem
M14	2010	An integrated approach for requirement selection and scheduling in software release planning
M15	2014	Bi-objective genetic search for release planning in support of themes

criterion. This resulted in the selection of 6 relevant papers listed in Table 2 (see the Reference section for full details).

For the *third iteration*, we asked for additional references to experts participating in the SUPERSEDE H2020 European project (www.supersede.eu). We received 5 proposals of papers which were considered relevant in the context of this project. From these 5 references, we discarded 3 of them: Ruhe's book because it was not providing new models but a compilation of knowledge [10], a paper by Durillo et al. [4] (see justification below) and a tool demo report since it was only 4 LNCS pages long [1]. The two remaining papers are listed in Table 3.

As a result, we selected 17 papers from this period starting in 2009 until date. This number is quite aligned with the total number of papers surveyed in [14], a total of 28 papers found in the period 1997–2008.

Table 3. Papers selected after recommendation in the SUPERSEDE project.

Ref.	Year	Paper title
M16	2016	Risk-aware multi-stakeholder next release planning using multi-objective optimization
M17	2010	Analytical product release planning

It is worth to mention that the second inclusion criterion left out some papers that are relevant to the SRP field but are not directly related to the goals of this research. We highlight four categories of papers that have not been included:

- Papers that present some experimentation (through benchmarks) on advanced algorithms that are applied to SRP (e.g., Durillo et al. [4] performing a sensitivity analysis of three genetic algorithms or Luna et al. [9] which surveys different metaheuristics for solving a multi-objective based formulation of the problem).

- Papers that assess the quality of SRP techniques or practices (e.g. Lindgren et al. [8] proposing a capability maturity model for release planning or Didar-al-Alam et al. [3] identifying release readiness improvement factors).
- Papers that focus on activities that may or must take place during SRP but do not focus on this particular context (e.g., Lehtola et al. [7] analysing requirements prioritization).
- Papers that add a small delta to previous publications (e.g., Szõke [15] adding assignment of features to distributed teams which does not change significantly the model proposed by the same author in another journal paper selected in our study).

2.3 Data Extraction and Analysis

For the data extraction we used a set of variables associated to each research question and the metadata used for the selection criteria (see Table 4). For the analysis we extracted data and categories as a basis for our results, and we revised the relevant papers for each particular topic to provide valuable information.

Table 4. Data extraction criteria

Metadata		Article title
		Authors' name
		Journal/conference
		Retrieval search query
		Date of publication
RQ1	Context	What is the main motivation for the model
	Models	What are the inputs that the model processes
		What are the algorithms/techniques used for computation
		What are the outputs that the model produces
RQ2	Tool	Tool availability
	Study	Study from academia or industry
		Description of threats to validity
	Adoption	Model proposed in literature or in industry
		Model validated in academy or in industry
		Model adopted in academy or in industry

2.4 Threats to Validity

Construct Validity. The selection of primary studies followed a strict protocol; however, the use of snowballing has some inherent, well-known limitations. The most important threat is that snowballing narrows the search scope to the referenced papers, therefore some papers may be left out. We consider that this threat was mitigated by the fact that we used an SLR published in a main software engineering journal as departing paper because such SLRs are normally cited by many researchers. A second mitigation action was to include a third iteration based on experts' opinion.

Internal Validity. Each paper was analysed in depth by one researcher of this study; however, it is known that it is easy to have different views or interpretations on the same paper depending, e.g., on the research background or past experiences in similar studies. Therefore, papers were checked by a second researcher when doubts arose.

External Validity. As in most literature reviews, this study does not aim to generalize results because there is no statistical basis to claim that the selected papers are a representative sample of the population (i.e., all the papers ever published about SRP). Therefore, any claim made in this study is limited to the set of studied papers. Moreover, the study only covers the works published in the literature, any model made available by other means (e.g., commercial tools) has not been considered.

Conclusion Validity. We defined a precise protocol of the steps to be followed, however as in any other literature review, we relied on the result of search engines which may offer different results in the future. Therefore, a replication of this study could lead to different selection of primary studies, and thus to different results.

3 Results

This section summarizes the results of the analysis of the 17 selected SRP references in relation to the RQs formulated in Sect. 2.2.

3.1 RQ1. *What SRP Models Have Been Presented Since 2009?*

Eleven [M2–M4, M6, M7, M10, M12–M16] out of the 17 papers that we have examined propose new models. The 6 remaining models [M1, M5, M8, M9, M11, M17] are extensions of the EVOLVE II model (being itself an extension of EVOLVE) and its implementation as a commercial tool, ReleasePlanner. Given this fact, we analyse separately the two families in the RQ. They are summarized in Tables 5 and 6 whose contents will be detailed in the rest of this subsection. Table 6 focuses on the customization that the different approaches propose on EVOLVE II.

Table 5. Summary of responses to RQ1: new SRP models

Ref	Motivation	Input factors	Output	Technique
M2	Planning in agile	RD • EC • SiF • VF	Next release • User stories • 3 priority levels	Nested knapsack problem, solved with branch and bound
M3	Planning in agile	RD • EC • VF • RF	Multi-release • User stories	Generalized knapsack problem, solved with branch and cut

(*continued*)

Table 5. (*continued*)

Ref	Motivation	Input factors	Output	Technique
M4	Scalability	RD • B&CC • SiF	Next release • Requirements	Backbone-based multilevel algorithms
M6	Scalability	B&CC • VF • RF	Next release • Features	Human assessment
M7	Scalability	SiF	Next release • Requirements • Prioritized list	Satisfiability modulo theory supported by NLP and AHP
M10	Robustness and completion time	RD • RC	Tasks assigned to developers	Genetic algorithm
M12	Planning in agile	RD • RC • EC • TiC • SiF	Multi-release • Features	0-1 multiple knapsack problem, solved with branch and bound
M13	Development cost and customer satisfaction	B&CC • SiF	Next release • Requirements	Evolutionary algorithm
M14	Time schedule +diversity of precedences	RD • RC • TiC	Next release • Requirements • Time schedule	Knapsack problem +resource-constrained scheduling problem, solved with branch and bound
M15	Grouping features by themes	RD • RC • EC • SiF • VF	Next release • Features	Evolutionary algorithm
M16	Risk assessment and stakeholder prioritization	B&CC • SiF	Next release • Requirements	Satisfiability modulo theory

Table 6. Summary of responses to RQ1: EVOLVE II-based models

Ref	Motivation	Input factors	Output	Technique
M1	Dealing with complex constraints	RD • TeC • RC • SiF	Multi-release • Features	ReleasePlanner+Constraint Programming Solver
M5	Scalability	RD • RC • SiF	Multi-release • Features	ReleasePlanner

(*continued*)

Table 6. (*continued*)

Ref	Motivation	Input factors	Output	Technique
M8	Quality aspects	RD • QC • RC • SiF	Multi-release • Features	ReleasePlanner
M9	Grouping features by themes	RD • RC • SiF	Multi-release • Features	Graph clustering +ReleasePlanner
M11	Active stakeholder involvement	RC • SiF	Multi-release • Features	ReleasePlanner +Weighting-Based techniques
M17	Introducing data analysis	RD • B&CC • SiF	Multi-release • Features	Crowdsourcing+Data Analysis+ReleasePlanner

RQ1.1. *What are the main motivations for the models?*

Among the 11 new models, we have identified a first group [M2, M3, M12] whose main concern is to address SRP in agile contexts, characterized by the uncertainty related to working with estimations (e.g., project velocity) and the flexibility required by agile projects (e.g., continuously changing customer needs, especially when the model is intended to plan more than one iteration as [M3, M12] do). Other approaches do also propose a solution in an agile context (e.g. [M14]) but the problems they face are not inherent to this paradigm (e.g., different type of requirement precedencies).

A second group of new models seems more concerned in proposing solutions that scale up in the presence of large sets of requirements ([M4, M6, M7]).

Another group [M10, M13] emphasise the contradictory nature of some SRP objectives: robustness (defined as a solution that satisfies expectations of project manager) vs. completion time [M10] and development cost vs. customer satisfaction [M13].

For the remaining new models, the motivations differ: grouping related features into themes to be scheduled preferable together [M15], considering different sorts of dependencies between requirements [M14], taking into account the requirements inherent risks [M16] or tightly integrating time [M14] into the solution.

For the 6 models that extend EVOLVE II/ReleasePlanner, the motivations are also different in each case. The resulting extensions, thus, are orthogonal and complementary with respect to the others. In [M1], for example, there is the need of dealing with requirements selection constraints that are more complex and richer than the ones that ReleasePlanner accepted as input. In [M5], the original motivation is to apply ReleasePlanner in an industrial case study and, as a result, a new extension is proposed to address the problem of feature generation. In [M8, M9] the proposed extension of EVOLVE II is driven by the need of dealing with quality aspects and features grouped into themes, respectively. In [M11] the main aim is to promote and support the active participation of the stakeholders in the planning process. Finally, in [M17] the motivation is to define a generic framework to solve different sorts of release planning problems by applying analytical methods on a diversity of data available from internal and external sources of information.

RQ1.2. *What are the inputs processed by the models?*

Here we replicate the analysis of input factors done in [14] by applying the same taxonomy of requirements selection factors. The taxonomy is as follows:

- Hard Constraints: include those factors that may restrict the order and time when certain features or requirements can be implemented. They are classified as:
  - Technical Constraints: Requirements Dependencies (RD), Quality Constraints (QC), and Other Technical Constraints (TeC).
  - Other hard constraints: Budget & Cost Constraints (BCC), Resource Constraints (RC), Effort Constraints (EC), Time Constraints (TiC).
- Soft Factors: include those factors that are more difficult to estimate and provide exact numbers on. They are classified as: Stakeholders' Influence Factors (SiF), Value Factors (VF), Risk Factors (RF), and Resource Consumption Factors (RCF).

In Table 7 the factors appear in their original formulation and, between parenthesis, their mapping to the taxonomy of requirements selection factors.

Table 7. Requirements selection factors per SRP model

Ref	Factors in model
New models	
M2	Technical precedencies (RD) • Story sizes+Velocity estimate (EC) • Preference (customers wish) precedencies (SiF) • Story & theme values (VF)
M3	Correlation & Precedence among stories (RD) • Story efforts (EC) • Story values (VF) • Story risks (RF)
M4	Requirements dependencies (RD) • Requirement costs (B&CC) • Customer satisfactions (SiF)
M6	Feasibilities (B&CC) • Profitabilities (VF) • Risks (RF)
M7	Keyword prioritization by user (SiF) • Pairwise comparisons between requirements by user (SiF)
M10	Task dependencies (RD) • Available resources/staff (RC) • Different skills (RC).
M12	Feature dependencies (RD) • Resource capacities (RC) • Business priorities and efforts (EC) • Iteration-delivery times (TiC) • Requirement priorities (SiF)
M13	Requirement costs (B&CC) • Client values (SiF) • Client weight factors (SiF)
M14	Task dependencies (RD) • Available employees (RC) • Employee salaries and dedications (RC) • Deadline of the project (TiC)
M15	Task dependencies (RD) • Available resources (RC) • Task efforts (EC) • Stakeholder values (SiF) • Theme & feature values (VF)
M16	Requirement costs (B&CC) • Stakeholder values (SiF)
EVOLVE-based models	
M1	Coupling and precedence dependencies (RD) • Constraints solvable by CP solvers (TeC) • Resource constraints (RC) • Stakeholder scores to features (SiF)

(*continued*)

Table 7. (*continued*)

EVOLVE-based models	
M5	Coupling and precedence dependencies (RD) • Resource constraints (RC) • Stakeholder scores to features (SiF)
M8	Coupling and precedence dependencies (RD) • Quality constraints (QC) • Resource constraints (RC) • Stakeholder scores to features (SiF)
M9	Coupling and precedence dependencies (RD) • Resource constraints (RC) • Stakeholder scores to features (SiF)
M11	Resource constraints (RC) • Stakeholder scores to features (SiF) • Stakeholder pre-selected features (SiF)
M17	Advanced feature dependencies (RD) • Budget capacities (B&CC) • Stakeholders' cost predictions (SiF) • Stakeholders' willingness to pay (SiF)

The input factors considered in EVOLVE II/ReleasePlanner are coupling and precedence constraints between features (RD), resource constraints (RC), as well as the stakeholder scores to features, given for a flexible number of criteria (SiF) [M1]. Some of the models that extend EVOLVE II/ReleasePlanner do not require extra input [M5, M9]. In addition, [M1] accepts as input any constraint that can be also expressed as an input of a Constraint Programming Solver (e.g., a constraint expressing mutual exclusion between features, or productivity investments). In the case of [M8], some quality aspects are added to the model. In [M11], the model adds a pre-selection of candidate features from stakeholders. Finally, [M17] presents a case study where the decision of what to release considers advanced feature dependencies, stakeholders' predictions on feature costs, budget capacities over different time periods, and the price that stakeholders would pay for each feature.

RQ1.3. *What are the outputs generated by the models?*

In the case of the new SRP models, we report three main categories of outputs. Eight approaches [M2, M4, M6, M7, M13–M16] produce as output the list of stories/requirements/features to be included in the next release. Moreover, in [M2] this list is divided into three groups with different priority: must-have, should-have and could-have stories. Going further, [M7] produces a list of prioritized requirements for the next release. In [M14] the output combines the list of next-released requirements with a schedule of the relative time at which these tasks should be performed by the development teams. Two approaches [M3, M12] produce a multi-release plan, more specifically an assignment of user stories or features to consecutive sprints. One approach [M10] does not produce release plans but an assignment of different tasks to different developers taking into account the specified constraints (i.e., the focus is only on the operation release planning).

The models that extend EVOLVE II/ReleasePlanner produce the same output than the original model/tool: an assignment of features to the releases in which they have to be implemented.

RQ1.4. *What are the algorithms or techniques applied by the models?*

In spite of their diversity, most of the approaches build on top of a very simple formulation of the SRP problem: calculate an assignment from a feature set {$f(1), \ldots, f(N)$} to a release plan $x = (x(1), \ldots, x(N))$ such $x(j) = k$ means that $f(j)$ is offered at release k. The solution is required to maximize the value of some utility or objective function. Except for naïve assumptions, the problem becomes a multi-objective problem and the approaches differ mainly in the techniques proposed to solve this NP problem.

Some approaches [M2, M3, M12, M14] formulate the problem as an instance of the knapsack problem, using some kind of branch and bound algorithm to solve it. The number of releases to plan and the type of constraints managed configure the exact type of knapsack (nested, generalized, …). [M14] combines this solution with additional techniques from resource-constrained project scheduling problem solving.

Other family of approaches use optimization-related techniques. Satisfiability modulo theory is among the preferred ones [M7, M16]. In addition, [M7] uses natural language processing (NLP) for extracting information from requirements, and the analytic hierarchy process (AHP) to fine-tune the prioritization of requirements. Evolutionary algorithms [M13, M15] and genetic algorithms [M10] are also used to implement the multi-objective problem. Finally, a backbone-based multilevel algorithm was tune to the SRP problem in [M4].

On a completely different setting, [M6], which is based on a large-scale industrial case at Ericsson for agile processes, presents a model which does not use any particular model. In fact, one of the motivations of the case study is to overcome the limitations posed by model-based approaches, especially in relation with the assumptions for their application. Therefore, all the release planning is integrated in the traditional agile lightweight agile process.

On the other hand, the EVOLVE-based models are implemented using the ReleasePlanner tool. From these approaches:

- [M5, M8] use the tool as it is.
- [M9, M11] complement the tool with some extra functionalities. For instance, [M11] includes a weighting-based technique to consider the influences that every stakeholder has in each iteration requirements.
- [M1, M9] integrate ReleasePlanner with some other software component. In [M1], ReleasePlanner is used to generate a first solution that feeds a constraint programming solver to find the best solution with an enlarged set of constraints. Conversely, [M9] uses the output of a graph clustering algorithm to feed ReleasePlanner.
- [M17] embeds ReleasePlanner into a more complex system which seeks maximizing a utility function.

3.2 RQ2. *To What Extent Have the SRP Models Surveyed in RQ1 Been Validated?*

To answer RQ2, following we summarize our findings for the three sub-RQs.

RQ2.1. *Are the models supported by tools?*

Nearly half of the works found in this state of the art mention some kind of tool, but is worth to differentiate those works that use a tool just to validate their approach (i.e. a prototype or just an ad-hoc solution specific for the paper) from those that are presenting a ready-to-use tool; in this second case, the most remarkable case is ReleasePlanner, mentioned in [M1, M5, M8, M9, M11, M17] (the models based on EVOLVE-II).

The papers that use a prototype or ad-hoc solution mention the following technologies: CP-Solver [M1], LP-Solve (an OSS linear programing tool) [M2], and CPLEX [M3]. In general, we can see that all the academic contributions use problem solvers to determine what features will be implemented in the next release.

The rest of papers ([M4, M6, M10, M12–M16]) did not report any kind of tool.

RQ2.2. *How has been industry involved in the models?*

All selected papers are academic works (i.e., all or most authors have an academic affiliation). In 4 cases [M5, M6, M8, M10] there was one author from the industry. It is worth noting that these works were the only ones that provide real case studies as part of their contribution. The rest of works were validated using experiments with the exception of [M9], which had a case study (using students).

All the proposed models except one were originated in academic research. The exception is an approach proposed by Ericsson [M6], which is also the only one that is being adopted by the industry (by the same company).

RQ2.3. *What are the major threats identified on the models?*

We found 5 papers [M5, M6, M7, M8, M10] with a wide analysis of the threats to validity (i.e., including internal, external, construct and conclusion validity threats). In 5 cases [M1, M4, M9, M11, M15] there were some threats explained but without any kind of structure and in 7 cases [M2, M3, M12–M14, M16, M17] there was no mention of threats to validity.

From those that mention the threats to validity, the most recurrent ones are: lack of testing in an industrial setting, the difficulty to generalize the results (e.g., due to the different skills of the testing participants, variability, and different domain applications), and that some formalizations do not contemplate all the possible dependencies.

4 Discussion

In this section discuss the most remarkable observations coming from the analysis of the results.

4.1 RQ1. *What SRP Models Have Been Presented Since 2009?*

One family of models prevails in the field, namely those coming from the EVOLVE-II/ReleasePlanner proposal. This fact is not surprising due to the high prevalence of the EVOLVE family in SRP before 2010 as reported in [14].

RQ1.1. *What are the main motivations for the models?*

Pursuing scale. In contrast with the results reported in [14], we have found more proposals aimed to scale in presence of large sets of requirements, which is a necessary step towards a full industry application. Unfortunately, the theoretical approaches have not been complemented with a proper validation in true industrial settings (see below).

RQ1.2. *What are the inputs processed by the models?*

Incomplete input factors. In a recent survey conducted through interviews with the three companies participating in the SUPERSEDE project [13], any SRP model should be able to deal with the following input factors in order to fit industry needs:

- For each requirement: the required effort to implement it (measured in person hours or similar), the developer skills required to implement it, its deadline (optional), its dependencies with respect to other requirements and its priority or its business value assessed by its stakeholders.
- For each release: its deadline and the list of available developers, for whom it is necessary to know:
  - his/her skills (to be matched with the ones required to implement requirements)
  - the amount of effort (measured in hours per week or similar) that s/he can invest in the release.

While requirements dependencies are taken into account by almost all the studies, other constraints like time (deadlines) are only addressed by two papers [M12, M14], and only one other paper [M10] take into account the different (developer) skills required and available to implement the requirements. On the contrary, soft factors like stakeholder consideration have been considered by a great share of approaches.

RQ1.3. *What are the outputs generated by the models?*

Simple outputs. Most of the SRP surveyed models produce a "binary" yes/no result that simply tells which requirements should be implemented for the next release (s). Only in the case of the models presented in [M10, M14], a richer type of results, in terms of requirement implementation scheduling and resource (developers) allocation, is provided. Clearly, this latter type of results is the one that matches better with the needs of software companies, in accordance with the input factors that they think that should be taken into account as reported above.

RQ1.4. *What are the algorithms or techniques applied by the models?*

SRP as a multi-objective problem. Most approaches recognize the existence of different and often conflicting objectives that need to be reconciled in the planning of releases. Solutions aim at reducing the inherent NP nature of the algorithms into linear-time implementations still able to find an optimal release plan. Experimentation is a key instrument for these works in order to demonstrate the accuracy and efficiency of the proposed technique.

4.2 RQ2. *To What Extent Have the SRP Models Surveyed in RQ1 Been Validated?*

RQ2.1. *Are the models supported by tools?*

Lack of ready-to-market tools. Only one tool (among the mentioned in the selected papers) can be considered a ready-to-market tool, namely ReleasePlanner. In fact, the product, while created in an academic context, is the core business of a spin-off company created around it. On the positive side, this commercial nature may be an indicator that there is a market share for this kind of tools. However, it is true that other tools are available in the market (e.g., Tempo Planner, a plug-in for JIRA) but none of them got their way into the selected papers. Another important observation is that 8 of the selected papers (out of 17) that did not mention any tool support; this circumstance makes very difficult to assess the suitability of these approaches.

RQ2.2. *How has been industry involved in the models?*

Scarce industrial contributions. Very few authors from the surveyed papers came from industry, in most cases as providers of a case study. Only in one single case the industrial authors were providing the SRP model. On the other hand, as commented before, there are other commercialized tools without representation in the academic literature. This situation may imply that the observations made in this paper are slightly deviated towards the academic perspective because there is a part of the big picture underrepresented in the academic literature.

RQ2.3. *What are the major threats identified on the models?*

Non-optimal consideration of threats to validity. For a literature survey focused on journals and main conferences, it is a bit surprising to find as much as 7 papers (from 17) with no mention at all at threats to validity. The absence of their analysis hampers the applicability of the presented models.

4.3 SRP Field Evolution

It is interesting to analyse the evolution of the SRP field by comparing the results from [14] and ours. As a first observation already stated in Sect. 4, the EVOLVE family of SRP models keeps its prevalence. In our literature review, we have found 6 papers out of 17 (35.3 %), less than the 16/28 = 57.1 % found by [14] but still the biggest share by far. No other method has shown prevalence in the field since 2009. In particular, in the case of papers not belonging to the EVOLVE family, none of the authors of models reviewed by [14] appear in the new models that we have studied, except for the case of [M14]. This means that although these researchers have made interesting proposals from a research point of view, the transferability to industry is not reported.

Svahnberg et al. [14] found that most models focused on a limited set of input factors, mainly hard constraints. Only a 57.1 % of the reviewed models considered soft factors. In our study, 15 out of the 17 models (88.2 %) do include soft factors. Noteworthy, soft-constraints that implied some kind of stakeholder involvement (SiF constraints) are considered in 13 out of the 17 models (76.5 %) that we have found, whereas that in [14] only 7 out of 28 (25 %) took them into account. An interpretation to this observation is that the proposals until 2008 needed to focus on algorithms able to solve such hard constraints in a comprehensive and efficient manner, while newer proposals could build on top of these results and focus on the business-related issues pointed out by soft factors.

As for model validation, we have already stated in Sect. 4 that industry involvement in the validation of the SRP models has decreased in the proposals found by our study compared to those in [14]: while in [14] the amount of models validated in the industry was 56 %, in the last years the participation of the industry has decreased to 23.5 % (4/17). This can be somewhat compensated by the fact that we have found more models, in proportion, considering larger sets of requirements, which is a factor supporting transferability of the models to industry.

5 Conclusions

In this paper, we have presented the state of the art on SRP models in the period 2009-2016. We have investigated two research questions analysing the characteristics of these models and their validation state. We have used the results of a previous systematic literature review published in 2010 [14] as main reference to our research methodology. The main results (detailed in Sect. 4) show some progress with respect the previous proposals in the period 1997–2008 surveyed in [14], in particular:

- Special attention to the scalability of the models (cf. RQ1.1).
- Increasing emphasis on soft factors like consideration of stakeholders and business value (RQ1.4).

On the contrary, some other observations make evident that SRP scientific proposals have not yet reached the maturity required by industrial contexts:

- Incomplete input factors considered (RQ1.2) and simple output produced (RQ1.3).
- Proof-of-concept tool support, except for the case of the EVOLVE-ReleasePlanner family of proposals (RQ2.1).
- Poor validation due to scarce industry validation (RQ2.2) and non-optimal consideration of threats to validity (RQ2.3).

We may conclude that the current state of the art claims for an increasing effort in making SRP models closer to industry requirements like those surveyed in [13].

Acknowledgements. This work is a result of the SUPERSEDE project, funded by the EU's H2020 Programme under the agreement number 644018.

References

1. Aydemir, F.B., Mekuria, D.N., Giorgini, P., Mylopoulos, J.: Next Release Tool. ER 2015
2. Basili, V., Caldiera, G., Rombach, D.: Goal/question/metric paradigm. In: Encyclopedia of Software Engineering, vol. 1, Wiley, New York (1994)

3. Didar-Al-Alam, S,M., Shahnewaz, S,M., Pfahl, D., Ruhe, G.: Analysis and improvement of release readiness – a genetic optimization approach. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) PROFES 2014. LNCS, vol. 8892, pp. 164–177. Springer, Heidelberg (2014). doi:10.1007/978-3-319-13835-0_12
4. Durillo, J.J., Zhang, Y., Alba, E., Harman, M., Nebro, A.J.: A study of the bi-objective next release problem. Empir. Softw. Eng. **16**(1), 29–60 (2011)
5. Grubb P, Takang AA. *Software maintenance: concepts and practice*. World Scientific, 2003
6. Lehman, M.M.: On understanding laws, evolution, and conservation in the large-program life cycle. J. Syst. Softw. **1**, 213–221 (1979)
7. Lehtola, L., Kauppinen, M., Kujala, S.: Requirements prioritization challenges in practice. In: Bomarius, F., Iida, H. (eds.) PROFES 2004. LNCS, vol. 3009, pp. 497–508. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24659-6_36
8. Lindgren, M., Land, R., Norström, C., Wall, A.: Towards a capability model for the software release planning process — based on a multiple industrial case study. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 117–132. Springer, Heidelberg (2008)
9. Luna, F., González-Álvarez, D.L., Chicano, F., Vega-Rodríguez, M.A.: The software project scheduling problem: a scalability analysis of multi-objective metaheuristics. Appl. Softw. Comput. **15**, 136–148 (2014)
10. Ruhe, G.: Product Release Planning: Methods, Tools and Applications. CRC Press, Boca Raton (2010)
11. Ruhe, G., Saliu, M.O.: The art and science of software release planning. IEEE Softw. **22** (6), 47–53 (2005)
12. Sommerville, I.: Software Engineering, 10th edn. Pearson (2015)
13. Stade, M., Seyff, N., Perini, A., Marco, J., Nadal, S., Franch, X.: D3.1: Requirements for Methods and Tools. SUPERSEDE EU project deliverables (2015)
14. Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., Shafique, M.U.: A systematic review on strategic release planning models. Inf. Softw. Technol. **52**(3), 237–248 (2010)
15. Szőke, Á.: Optimized feature distribution in distributed agile environments. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 62–76. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13792-1_7
16. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: EASE 2014

Primary Studies Found in the Literature Survey

M1. Przepiora, M., Karimpour, R., Ruhe, G.: A hybrid release planning method and its empirical justification. In: ESEM 2012

M2. van Valkenhoef, G., Tervonen, T., de Brock, B., Postmus, D.: Quantitative release planning in extreme programming. Inf. Softw. Technol. **53**(11), 1227–1235 (2011)

M3. Golfarelli, M., Rizzi, S., Turricchia, E.: Multi-sprint planning and smooth replanning: an optimization model. J. Syst. Softw. **86**(9), 2357–2370 (2013)

M4. Xuan, J., Jiang, H., Ren, Z., Luo, Z.: Solving the large scale next release problem with a backbone-based multilevel algorithm. IEEE Trans. Softw. Eng. **38**(5), 1195–1212 (2012)

M5. Zorn-Pauli, G., Paech, B., Beck, T., Karey, H., Ruhe, G.: Analyzing an industrial strategic release planning process–a case study at Roche diagnostics. In: REFSQ 2013

M6. Heikkilä, V.T., Paasivaara, M., Lassenius, C., Engblom, C.: Continuous release planning in a large-scale scrum development organization at ericsson. In: Baumeister, H., Weber, B. (eds.) XP 2013. LNBIP, vol. 149, pp. 195–209. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38314-4_14

M7. McZara, J., Sarkani, S., Holzer, T., Eveleigh, T.: Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment. Empir. Softw. Eng. **20**(6), 1721–1761 (2015)

M8. Felderer, M., Beer, A., Ho, J., Ruhe, G.: Industrial evaluation of the impact of quality-driven release planning. In: ESEM 2014

M9. Agarwal, N., Karimpour, R., Ruhe, G.: Theme-based product release planning: an analytical approach. In: HICSS 2014

M10. Gueorguiev, S., Harman, M., Antoniol, G.: Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering. In: GECCO 2009

M11. Heikkilä, V., Jadallah, A., Rautiainen, K., Ruhe, G.: Rigorous support for flexible planning of product releases - A stakeholder-centric approach and its initial evaluation. In: HICSS 2010

M12. Szõke, A.: Conceptual scheduling model and optimized release scheduling for agile environments. Inf. Softw. Technol. **53**(6), 574–591 (2011)

M13. Chaves-González, J.M., Pérez-Toledano, M.A.: Differential evolution with Pareto tournament for the multi-objective next release problem. Appl. Math. Comp. **252**, 1–13 (2015)

M14. Li, C., van den Akker, M., Brinkkemper, S., Diepen, G.: An integrated approach for requirement selection and scheduling in software release planning. Requir. Eng. **15**(4), 375–396 (2010)

M15. Karim, M.R., Ruhe, G.: Bi-objective genetic search for release planning in support of themes. In: Goues, C., Yoo, S. (eds.) SSBSE 2014. LNCS, vol. 8636, pp. 123–137. Springer, Heidelberg (2014). doi:10.1007/978-3-319-09940-8_9

M16. Pitangueira, A.M., Tonella, P., Susi, A., Maciel, R.S., Barros, M.: Risk-aware multi-stakeholder next release planning using multi-objective optimization. In: Daneva, M., Pastor, O. (eds.) REFSQ 2016. LNCS, vol. 9619, pp. 3–18. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30282-9_1

M17. Nayebi, M., Ruhe, G.: Analytical product release planning. In: The Art and Science of Analyzing Software Data. Morgan Kaufmann (2015)