Modelling Prioritisation Decision-making in Software Evolution

Denisse Muñante[1], Fitsum Meshesha Kifetew[1], and Oliver Albrecht[2]

[1] Fondazione Bruno Kessler, Italy
munante|kifetew@fbk.eu
[2] SEnerCon GmbH, Germany
oliver.albrecht@senercon.de

Abstract. Decisions concerning prioritisation occur in different moments during software development and can involve different stakeholders. Our research objective is to develop prioritisation processes that meet stakeholders' needs, and allow obtaining better quality decisions. In this paper we propose a structured approach to model decision-making in real setting with the purpose of eliciting from the stakeholders involved in the decision-making process their needs for improvements. We use the resulting models to derive a general model for prioritisation processes and outline how such processes could be tool-supported.

Keywords: prioritisation decision making process, software evolution, decision model notation (DMN), business process model notation (BPMN)

1 Introduction

Decision-making (DM) is a frequently occurring activity during software evolution. The decisions often take the form of *prioritisation* [3]. In particular, requirements prioritisation is a crucial DM activity which involves different stakeholders including clients, managers, and developers [6]. The impact of the decisions taken, in particular at the early stages of software development, could have consequences of strategic importance during the later stages of the development process. Consequently, the quality of the resulting decisions could be improved by supporting the DM with a tool-supported process, however designing the appropriate tool-supported process requires a deep understanding of the needs of the software development team and of the "as-is" DM processes, if any. In this paper we particularly focus on DM in requirements prioritisation [1], performed as part of the evolution and maintenance of a software system.

The work in this paper is based on the analyses of DM processes in the industrial use cases of SUPERSEDE [3], a European H2020 project under the theme "Tools and Methods for Software Development". SUPERSEDE aims at delivering methods and tools to support requirements prioritisation and enactment, to

[3] https://www.supersede.eu/

enable a continuous software evolution, driven by user-feedback and monitored data. SUPERSEDE has industrial use cases coming from three different companies: ATOS Spain, Siemens AG, and SEnerCon GmbH. In this paper, we focus on the use case from SEnerCon since it is compact enough to present in a paper while sharing most of the cross-cutting issues and challenges of the other uses cases. However, the methodology presented in this paper has been applied to the other use cases as well.

SEnerCon is a Small-Medium Enterprise with 25-years of experience in engineering and consultancy in the domain of energy efficiency management. The company has mainly software developers and engineers, who focus on software development and energy consulting. The software application *iESA* [4] (interactive Energy Saving Account) developed by SEnerCon is the subject of the case study. *iESA* is used in Germany by private households, public buildings and offices, as well as in several projects as a monitoring tool of energy consumption. For SEnerCon it is important to improve the quality of its services in order to satisfy its customers. Therefore, a monthly DM process takes place to evolve and maintain *iESA*.

To analyse DM processes for requirements prioritisation in the use cases, we defined a structured methodology, including a semi-structured survey to elicit information about the current practice, and modelled the resulting processes. The formulation of a reliable and representative model is an important step in the study of DM processes. Knowing, understanding, and being able to represent how decisions are adopted is a fundamental prerequisite for improving existing DM processes or designing new ones. We used the models to identify the main concepts involved in the prioritisation DM process, and consequently to derive a general model for a tool-supported prioritisation DM process. These results could help managers and designers to explore and implement their own prioritisation DM processes.

In the rest of the paper, we describe the methodology in Section 2 and illustrate how we applied it to the case study in Section 3. We discuss the analysis of the models and how they supported us in deriving a general, tool-supported prioritisation DM process in Section 4. Related works and conclusion are presented in Sections 5 and 6 respectively.

2 The methodology for modelling prioritisation DM processes

We introduce the methodology used to *elicit* and *model* prioritisation DM processes from SUPERSEDE use cases. This methodology is divided in parts:

1. eliciting and modelling the "as-is" prioritisation DM process from the use cases;
2. identifying the main concepts involved in the prioritisation DM processes;

[4] www.energiesparkonto.de

3. identifying potential improvements in the current (as-is) processes. Thus, we model the "to-be" prioritisation DM process considering the identified improvements.

The technique adopted to **elicit information** from stakeholders in the use cases about their relevant DM processes is questionnaire-based. The interviewed stakeholders should know very well their DM processes, they could be project managers, developer leaders, and so on. The objective of the questionnaire is to obtain a vision of the major aspects of the DM processes when software products are evolved. The following main questions are used:

Q1 *What are the inputs to the DM process?* the sources of information used in the DM process are described and discussed. For example: list of alternatives (features), company policies, or decisions from other DM processes.
Q2 *What is the output of the DM process?* the result of the DM process is described. For example: artefacts that are approved, rankings of features or allocation of development activities to developers.
Q3 *Who are the stakeholders involved in the DM process?* the stakeholders involved in the DM process are identified and described in order to have a clear vision of their roles, goals, expertises and perspectives.
Q4 *What are the methods/tools used for the DM process?* the methods/tools exploited in the DM activities are described. For example: face-to-face meetings, discussions, focus groups, automated methods already in place.
Q5 *How is the DM process structured and how is its flow of activities?* the set of activities and their sequence are detailed in the DM process.

The information elicited through the questionnaire are used **to model the *as-is prioritisation DM processes***. To this end, we use the Decision Model Notation (DMN) [5] proposed by OMG [5]. According to OMG, DM is addressed from three different perspectives by existing modelling standards: *Business process models* (e.g. *Business Process Model and Notation (BPMN)*), *Decision Requirements*, and *Decision Logic*. The last two are part of the DMN notation.

Business process models
BPMN provides multiple diagrams to analysts who design and manage business processes. BPMN provides businesses with the capability of understanding their internal business procedures in a graphical notation and will give organisations the ability to communicate these procedures in a standard manner. It facilitates the understanding of the performance of collaborations and business transactions within and between the organisations.

Decision Requirements Level
Figure 1 depicts the elements and dependencies involved in a DM. A *decision* denotes the act of determining an output from a number of inputs, using decision logic (see below) which may reference one or more business knowledge models.

[5] http://www.omg.org/

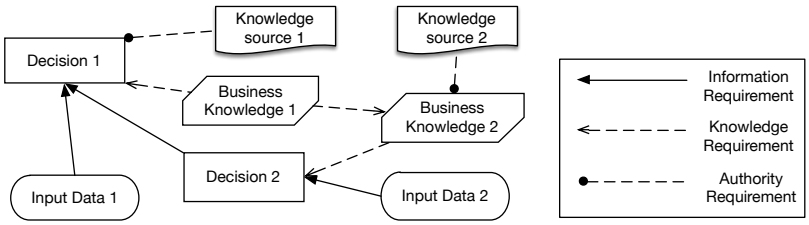

Fig. 1. An example of elements and dependencies of a domain of DM

Notice that the output of a decision could be the input of other decision. A *business knowledge model* denotes a function encapsulating business knowledge, e.g., as business rules, a decision table, or an analytic model. An *input data* denotes information used as input by one or more decisions. A *knowledge source* denotes an authority for a business knowledge model or decision. Concerning the dependencies, an *information requirement* denotes input data or decision output being used as input to a decision. A *knowledge requirement* denotes the invocation of a business knowledge model by the decision logic of a decision. Finally, an *authority requirement* denotes the dependence of an element on another element that acts as a source of guidance or knowledge.

Decision Logic level
At this level, the defined decision requirements elements (see above) may be specified in greater detail, to capture a complete set of business rules and calculations, and to allow the DM to be automated. Thus, every ***decision*** is defined using a ***value expression*** which is a table specifying how the decision's output is determined from its inputs. In the same way, a ***business knowledge model*** is defined using a ***value expression***, which can be expressed as ***functions*** invoked from decisions' value expressions.

The combined use of BPMN and DMN provides a graphical language for describing multiple levels of human DM within an organisation. In this context, DMN models describe collaborative organisational decisions, their governance, and the business knowledge required for them. These models help us in understanding the current practice and exploring new methods that can support the "as-is" processes, e.g. automated or semi-automated DM methods.

3 Applying the methodology in SEnerCon case

SEnerCon's *iESA* application enables end-users to monitor and analyse their energy consumption. It counts today tens of thousands of end-users. Most of the features of *iESA* are free, on condition that the data of registered end-users is used and analysed, upon anonymity. SEnerCon performs a monthly DM process to improve the quality of *iESA*'s services. The DM process is related to

Table 1. Example of bug reports and new features requests

Source	**Subject**	**Priority**	**Effort**	**Due date**
Ticket	Error in heating diagram	maximum	20h	Jan 2017
Ticket	Discovergy meter can not be created	high	2h	-
Ticket	Conversion factor for electricity	high	-	-
Ticket	Missing validation on gas consumption of cars	high	1h	-
Project	Enable payment system (PayPal)	-	-	Jan 2017
Project	Create a new household navigation	-	-	-
Project	Enable importation of "Itron" meter data	-	-	Feb 2017

product maintenance and aims at selecting which new features and bug fixes to implement in an upcoming release of the product. To elicit the current ("as-is") DM process of the *iESA*, we apply the first step of the methodology introduced in Section 2. In this section, we report the elicited information via the questionnaire introduced as part of the methodology.

Q1: inputs to the DM process: the main input to the DM process is a set of requests, *i.e.,* requests for new features and bug reports, to be implemented or resolved. They are collected from the *Ticket System*, from *project managers*, and from *advisors*.

From the "Ticket System": Requests for new features and bug reports are collected form telephone calls, emails and feedback gathering mechanism of iESA. The *help-desk* adds them as requests to the *Ticket System*. Table 1 shows examples of such requests. A *subject* and a *priority* should be established initially. The *effort* and *due date* could also be added, if not they are addressed later in the DM process. Priorities can be *maximum*, *high*, *normal*, and *low*.

From project managers: Other requests are collected from *project managers* of external projects which, in one way or another, are related to the iESA application. In most cases they have *due dates* based on the corresponding project plan of the related application. Table 1 shows examples of such types of requests. The collection of information is mainly done manually during *inter-project meetings*. However, some of the new feature requests may be created in the *Ticket System*.

From advisors of the iESA application: *Advisors*, who are stakeholders of SEnerCon , can suggest new features to implement. Their suggestions mainly focus on strategical aspects, *e.g.,* to improve the iESA's visibility.

Q2: outputs of the DM process: the goal of the DM process is the next release plan, *i.e.,* the list of new requests to implement in the upcoming release.

For this, the priorities should be assigned (if they are not set before), and clarifications on the due dates should be establish accordingly available resources.

Q3: stakeholders involved in the DM process: different stakeholders, with varying expertise and levels of decision power, are involved in the DM process. In particular, the following stakeholders were identified:

- **Help-desk**, s/he uses the Ticket System to collect new requests from end-users.
- **Product manager**, collects all requests, filters and merges them. S/he leads meetings with the developers and project managers to assign the importance, called attributes, to the requests. S/he decides the requests to be implemented for the next release.
- **Developers**, they maintain the software. They express the development effort of each request.
- **Project manager** is responsible for collecting requests of their applications (software applications that use iESA's services). These requests are expressed in the inter-project meetings.
- **Advisors**, who can be the CEO, editors and sales staff, inform new requests accordingly their impressions to make more useful and visible the application.

Q4: methods/tools used in the DM process: besides the Ticket System, there is no automated tool employed for supporting the DM process. The most important method used is the weekly stand-up meeting, which is conducted by the *Product manager* and attended by the *development team*, the *help desk* and *project manager*. The *Product manager* should collect and analyse the data to decide on a new release plan as soon as possible.

Q5: current DM process A representation of the DM process using BPM and DMN is depicted in Figure 2. The main steps are:

- *Requests* are collected form the Ticket System and project managers.
- Requests are filtered and merged by product manager with the whole development team, help-desk and project managers.
- *Attributes* are assigned to requests in order to indicate their importance. These attributes change the priorities of requests. They can be: *due date*, such a date may be fixed during press releases; *user impact*, this reflects, for instance, the number of users affected by the bug, on the value expected with the new feature; and, *development effort*, this estimation represents the number of hours to implement a request.
- Once the attributes are collected, *Product manager* decides the priorities of the requests. This list is consequently reviewed to determine its stability.
- Once the prioritised list is stable, the *Product manager* plans the next release considering available time and resources.

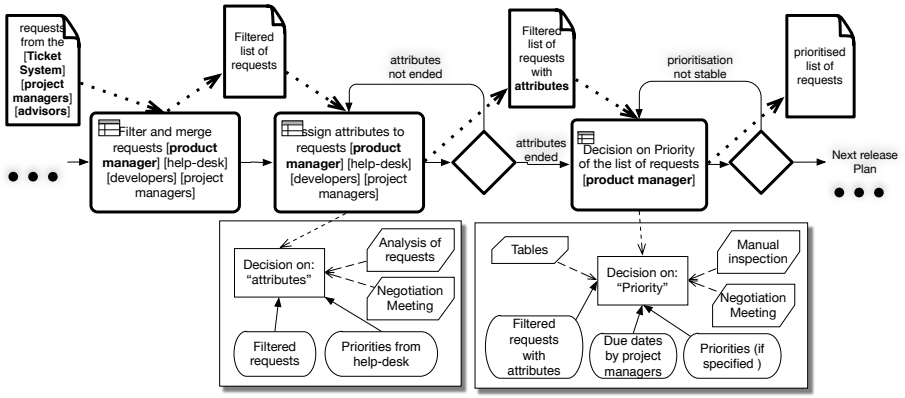

Fig. 2. The prioritisation DM process in SEnerCon

As discussed above, the *product manager* plays a central role in the prioritisation DM process of SEnerCon . While in a small organisation this may have its advantages, it could easily become a bottleneck as well as a single point of failure as the organisation grows. Furthermore, in an organisation where the input of every stakeholder int he DM process is to be considered, a more collaborative and transparent process involving all stakeholders would be beneficial.

4 Extracting the requirements for a general, tool-supported prioritisation DM process

In this section, we use the previous model to extract the *main general concepts* involved in prioritisation DM processes. We then introduce a general, tool-supported prioritisation DM process based on these concepts.

Main concepts in the prioritisation DM process

Figure 3 depicts a class diagram containing the main concepts involved in the prioritisation DM process. The top and bottom parts of the figure present the SEnerCon concepts, depicted as dotted rectangles, identified from the ("as-is") DM process. The middle-part of the figure contains the SW development and the prioritisation DM concepts, they are depicted as classes with associations.

The SEnerCon concepts are used to identify/match the *counterparts* in the prioritisation DM process. The prioritisation DM concepts allow designers to know the requirements to implement a tool that supports this DM process. We also present a matching of concepts between the software development and the prioritisation DM. This matching allows managers to know the sources of setting inputs to configure the prioritisation DM process.

In the SW development, on one side, an *organisation* (e.g. *SEnerCon*) develops *software products* (e.g. the *iESA* software application) using a *software*

development process. This process is composed of a set of *phases*, each development phase affects *software artefacts* (e.g. *requests, i.e., new features and bug reports*). On the other side, an organisation contains *roles* (e.g. *developers, projects managers, product manager*) which are played by *stakeholders* (concrete people) inside or outside the organisation, the stakeholders could have different *influences* (e.g. derived from their *expertises*). Finally, an organisation contains some *policies* or rules that manage/guide the decision processes (e.g. *attributes* to take into account when a decision is taken).

In the prioritisation DM, when a software artefact requires a *prioritisation*, it becomes *feature* in the prioritisation DM process. Then, a subset of stakeholders (see SW development), who are affected by the decision, become *decision makers*. A stakeholder becomes *negotiator* to solve the potential conflicts in the process. Decision makers express *preferences* on features according to some *criteria*, the criteria are derived from policies of the organisation (see SW development). Decision makers could have different *powers of decision per criterion* that can alter the final decision. These powers are derived from the influences and roles played by decision-makers (see SW development). Finally, *methods* are used to determine the list of *prioritised features*.

Concerning SEnerCon , three kind of stakeholders are involved in the prioritisation DM process: *developers*, *project managers* and the *product manager*. In case of conflicts concerning the priorities of features expressed by the various stakeholders, the product manager acts as negotiator to solve these conflicts. The criteria used in this process are provided as *types of attributes*. The preferences are expressed as *attributes to support* the features. Currently, attributes provided by decision makers depend on their roles, e.g. developers provided attributes re-

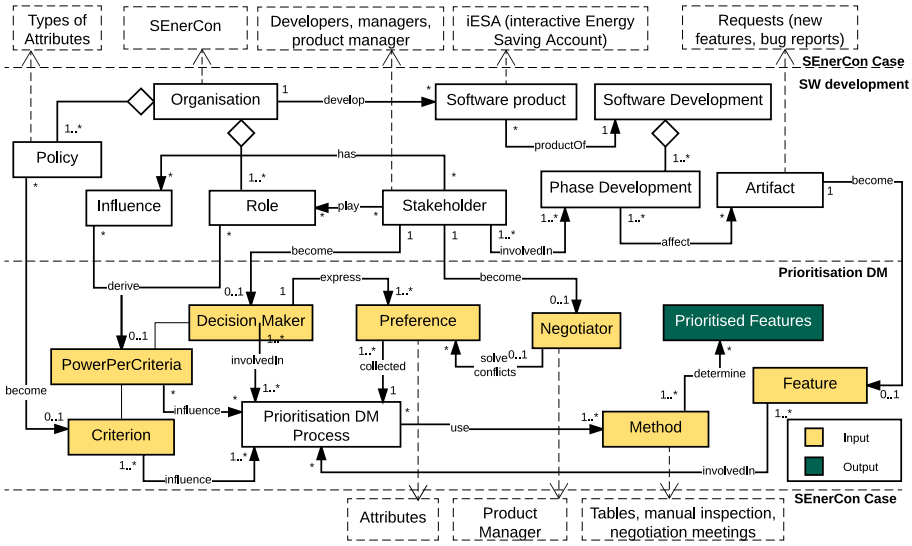

Fig. 3. The concepts involved in the prioritisation DM process

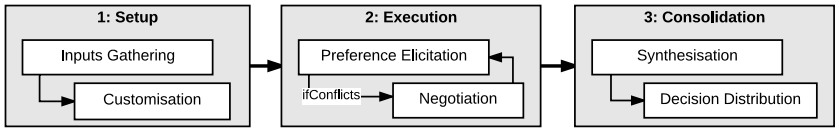

Fig. 4. The automated tool-supported prioritisation DM Process

lated to implementation whilst project managers related to the management. Thus, the *power of decision per criterion* is derived from the roles played by decision makers. Finally, the methods used to determine the list of prioritised features are *tables*, *manual inspection* and *negotiation meetings*.

Tool-supported prioritisation DM process
Based on the previously discussed prioritisation concepts and the proposal by Ruhe et al. [6], we present a tool-supported prioritisation DM process which is composed of three main steps, see Figure 4. First, a tool should support the *setup* of the DM process by gathering and customising the inputs. For example, collecting, filtering and selecting the features to be prioritised, and/or selecting the stakeholders to be involved in the process and assign responsibilities of decision-makers or negotiator. Second, the tool should support the *execution* of the process eliciting the preferences of decision makers, and being able to facilitate negotiations in case of conflicts. Third, the tool should support the *consolidation* of the process determining the final decision based on a synthesis method. Once the decision is taken, it is distributed to the stakeholders.

5 Related Work

Ruhe et al [7] outline a high-level DM process composed of three phases: the *modelling phase* to collect the setting information, the *exploration phase* to generate alternative solutions, and the *consolidation phase* to determine the decision. We build on this general outline and propose how to realise the details of each phase in practice. Firesmith [3] outlines fundamental issues to be taken into consideration when dealing with requirements prioritisation. In particular, Firesmith discusses the meaning of priority, it could be: prioritise *by implementation order* or *by importance*. While the ultimate outcome of the two meanings could be the same, the processes and methodologies followed to achieve them could widely differ depending on which meaning is chosen. Firesmith also outlines the following issues and challenges faced during requirements prioritisation: mandatory nature of requirements, large number of requirements, limited resources, changing requirements, stakeholder-developer collaboration, incompatible requirements, subjective prioritisation, and consequences of poor prioritisation. On the other hand, Carlshamre et al [2] outlined the following classes of dependencies among requirements that need to be taken into consideration during prioritisation: combination, implication, exclusion (conflict), revenue/cost-based, and time-related. Following the issues on dependencies outlined by Carshamre

et al, Li et al [4] analyse the influences of these dependencies on requirements selection/scheduling and use these dependencies in their proposed approach. Furthermore, a review of the issues related to requirements prioritisation and the solutions proposed to address them are reported by Achimugu et al [1].

The aforementioned works help us understand the main concepts, issues and challenges related to the prioritisation DM process. In this paper, we build on these concepts and propose a methodology to elicit the information involved in such processes in the context of real industrial use cases.

6 Conclusion

In this paper, we presented a structured approach to elicit and model prioritisation DM processes. The elicited information was used to derive the main concepts required for the implementation of a general, tool-supported prioritisation DM process. Specifically, we presented (i) a methodology to elicit prioritisation DM processes, (ii) the application of the methodology to industrial use cases, (iii) the derivation of the main general concepts involved in the prioritisation DM process, and (iv) the identification of a general, tool-supported DM process. In conclusion, this paper can be useful as a guideline for companies that wish to explore and implement their own prioritisation DM processes.

Acknowledgement. The research described in the paper was funded by the EU project SUPERSEDE (grant agreement no 644018).

References

1. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.N.: A systematic literature review of software requirements prioritization research. Inf. Softw. Technol. 56(6), 568–585 (Jun 2014), http://dx.doi.org/10.1016/j.infsof.2014.02.001
2. Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J.: An industrial survey of requirements interdependencies in software product release planning. In: Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on. pp. 84–91 (2001)
3. Firesmith, D.: Prioritizing requirements. Journal of Object Technology 3(8), 35–48 (2004), http://dx.doi.org/10.5381/jot.2004.3.8.c4
4. Li, C., Akker, M., Brinkkemper, S., Diepen, G.: An integrated approach for requirement selection and scheduling in software release planning. Requirements Engineering 15(4), 375–396 (2010), http://dx.doi.org/10.1007/s00766-010-0104-x
5. Object Management Group (OMG): Decision Model and Notation (DMN), Version 1.0 (September 2015)
6. Ruhe, G., Saliu, M.: The art and science of software release planning. Software, IEEE 22(6), 47–53 (Nov 2005)
7. Ruhe, G., Saliu, O., Bhawnani, P., Momoh, J., Ngo-The, A.: Decision support for software release planning - methods, tools, and practical experience. In: Software Engineering Workshop - Tutorial Notes, 2005. 29th Annual IEEE/NASA. pp. 217–250 (2005)