

The GRADE taxonomy for supporting decision-making of asset selection in software-intensive system development



Efi Papatheocharous^{a,b*}, Krzysztof Wnuk^a, Kai Petersen^a, Séverine Sentilles^c, Antonio Cicchetti^c, Tony Gorscak^a, Syed Muhammad Ali Shah^b

^a Blekinge Institute of Technology, Karlskrona, Sweden

^b SICS Swedish ICT AB, Kista, Sweden

^c Mälardalen University, Västerås, Sweden

ARTICLE INFO

Keywords:

Software engineering
Decision-making
Taxonomy
Knowledge management

ABSTRACT

Context: The development of software-intensive systems includes many decisions involving various stakeholders with often conflicting interests and viewpoints.

Objective: Decisions are rarely systematically documented and sporadically explored. This limits the opportunity for learning and improving on important decisions made in the development of software-intensive systems.

Method: In this work, we enable support for the systematic documentation of decisions, improve their traceability and contribute to potentially improved decision-making in strategic, tactical and operational contexts.

Results: We constructed a taxonomy for documentation supporting decision-making, called GRADE. GRADE was developed in a research project that required composition of a common dedicated language to make feasible the identification of new opportunities for better decision support and evaluation of multiple decision alternatives. The use of the taxonomy has been validated through thirty three decision cases from industry.

Conclusion: This paper occupies this important yet greatly unexplored research gap by developing the GRADE taxonomy that serves as a common vocabulary to describe and classify decision-making with respect to architectural assets.

1. Introduction

Software-intensive system development is a complex endeavor. Since software-intensive systems continue to grow in size and complexity [1], they often contain several components, sub-systems or other assets. An asset is here defined as a software artifact developed or obtained by a software development organisation that impacts software value [2]. These assets can range from software functionality exposed to the end-user (e.g., a navigation system in a vehicle) to software that controls physical embedded assets (e.g., fuel injection control in vehicle engines). A set of decisions need to be made on strategic, tactical and operational levels [3] before each asset can be considered and incorporated.

These decisions usually have an impact on software architecture and require the involvement of the software architects who make decisions related to the selection of the right components, connections, and the architectural style to be used [4]. Several authors focused on architectural decision-making, e.g. [4–9]. These decisions create architecture

knowledge [5] that should be managed to support system development and maintenance. However, architecture knowledge mainly concerns technical consequences of selecting a specific asset and is just a subset of the knowledge that is created and should be captured during software-intensive system development [10]. Additional knowledge concerns business: goals, roles other than architects, utilised decision methods, system and software development process knowledge, knowledge about assets considered but not selected and other environmental factors related to the decision. Only capturing this vibrant picture of relevant aspects creates the necessary comprehensive view on decision-making for software-intensive systems that can support the challenges associated with their development [10].

Software-intensive systems are often large and composed of many assets originating from various sources. Assets can be developed internally (within software organisations), purchased as software components, obtained from Open Source Software (OSS) communities or developed with the help of outsourcing. A recent literature review exploring factors that influence the selection of assets originating from in-

* Corresponding author. Fax: +004646131021.

E-mail addresses: efi.papatheocharous@ri.se (E. Papatheocharous), krzysztof.wnuk@bth.se (K. Wnuk), kai.petersen@bth.se (K. Petersen), severine.sentilles@mdh.se (S. Sentilles), antonio.cicchetti@mdh.se (A. Cicchetti), tony.gorscak@bth.se (T. Gorscak), syed.shah@sics.se (S.M.A. Shah).

house, outsourced, components off the shelf (COTS), OSS, or services highlights the lack of a systematic approach to decision documentation [11] that can support learning and retrospective analysis. Moreover, the review indicates that companies perform only partial analysis of possible options, mainly because they lack systematic analysis approaches. The first steps towards creating systematic analysis approaches are: 1) to create a common vocabulary with a shared understanding of decision components and contextual factors, and 2) to provide a structured instrument for collecting decision-making evidence.

These two issues prompted the development of the GRADE taxonomy [12] that provides a structured instrument for collecting and documenting decision-making evidence. This opens up for the potential for retrospective learning and improved comprehension of decision outcomes that can mitigate adverse effects of sub-optimal decision-making on strategic, tactical and operational levels. To the best of the authors' knowledge, no other work supports this endeavor.

This paper presents the GRADE taxonomy version 1.0 and the steps involved in its creation, refinement, and step-wise validation. This paper extends our previous work [12] with the following aspects: 1) a refined and validated GRADE taxonomy is presented, 2) the details about the taxonomy construction process are presented, and 3) the validation of GRADE is carried out and presented within a two-phase validation study where additional viewpoints and challenges are identified.

This paper is structured as follows: [Section 2](#) discusses related work, [Section 3](#) describes the research process, [Section 4](#) analyses the GRADE taxonomy elements, [Section 5](#) validates GRADE via thirty-three industrial decision cases, [Section 6](#) includes a discussion and the limitations of this work and finally [Section 7](#) concludes the work and summarises future research.

2. Related work

The related work to GRADE has been divided into two areas: (i) knowledge management and decision-making support (ii) documentation and taxonomies in developing software systems.

Knowledge management and structuring evidence to support architectural decisions gain importance in software engineering [13] since architecture is not isolated from decision-making and an architect is “a decision maker instead of someone drawing boxes and lines” [5]. The reasons underlying architectural design decisions, which result in corresponding software architecture, are gaining importance, even over the architecture specification itself [14]. One of the reasons for this increased emphasis is that software-intensive systems grow in size and complexity and often are designed by several architects during their development and maintenance phases which are also extended to long periods of time. Capturing, storing and managing architecture knowledge helps to minimise knowledge vaporisation and architectural drift [5].

Several authors studied how practitioners make architectural decisions, e.g. [4], suggested models for architectural decisions, e.g. [8], or proposed ontologies for architectural decisions, e.g. [9]. A suitable amount of solutions for storing architectural knowledge exists, and a complete review of them is beyond the scope of this article. The interested reader is referred to [5–8,15–17] for detailed surveys on the subject. For this work, it is essential to remark that automated knowledge reasoning is still scarcely supported because it typically requires a preliminary encoding of knowledge [15]. The support of such an initial encoding is the goal of ontologies and taxonomies efforts like the one described throughout this paper. However, ontologies of architectural decisions are often based on opinions rather than empirical evidence, e.g. [9]. Moreover, architecture knowledge is a subset of knowledge that should be captured during software-intensive system development.

Knowledge repositories are used in software engineering for a variety of purposes [10], e.g., for system modeling [18], for recording architectural decisions in the design of model and meta-data

repositories [19], and for tracing the originators of software and data artifacts in a project [20]. These works usually aim to record results of evaluations and decision processes rather than their rationale. In this respect, Capilla et al. [21] discuss a solution to record architectural decisions as linked to other factors involved, notably functional and non-functional properties. Such knowledge is stored using links between artifacts, whose semantics are defined as a meta-model.

Documentation and taxonomies in software engineering aim to increase common understanding between the different stakeholders and to create traceability between decisions. As such, in [22] a documentation framework for architectural decisions is presented using the conventions of ISO/IEC/IEEE 42010 [23] consolidating four different viewpoints. The four viewpoints, Decision Detail, Decision Relationship, Decision Chronology and Decision Stakeholder Involvement, satisfy several stakeholder concerns related to architecture decision management. In [24] the ADDRA approach is presented, where architects can use for recovering architectural decisions made retrospectively. Only a limited number of studies combine explicit description of design decisions with architectural design. As such work, in [25] a design map for recoding architectural decisions and a meta-model focusing on the relationships between non-functional properties and architectural styles are described.

Several related works support documentation in software engineering by developing taxonomies, outside however the context of architectural decision-making. These works are limited in only structuring knowledge areas in software engineering and make explicit use of the notion of taxonomies. Examples include the Guide to the Software Engineering Body of Knowledge (SWEBOK) [26], which describes the software engineering discipline in a structured way, the work of Glass et al. [27] which describes a taxonomy on software engineering research, Blum [28] which describes development methods, Smite et al. [29] which describes a taxonomy for global software engineering and Unterkalmsteiner et al. [30] which describes a taxonomy associating software requirements engineering and testing. Bayona-Ore et al. [31] defined an approach for the construction and evaluation of taxonomies. To the best of our knowledge, no taxonomy focuses on supporting decision-making of asset selection in software-intensive system development.

Based on the previous work, we have taken into account the following: 1) efforts in building taxonomies should be driven by clearly defined goals (as recommended in [26,31]), 2) a systematic process needs to be followed in the taxonomy construction (similar to [31]), 3) taxonomies can be built bottom-up in cases where relationships are not well understood (according to [30]), 4) experts should be involved in the taxonomy construction process (as in [29]) and 5) taxonomies can be validated against their purpose, either through classification based on the literature [29], or through industrial case studies [30]. The above points were used to define and refine the GRADE taxonomy (introduced in [12]) and in this work, evaluated to effectively document cases of various decision alternatives in the development of software-intensive industrial systems.

3. Research process and methodology

The GRADE taxonomy was created following a three-phase process and influenced by the design science methodology [32]. The taxonomy is the main artifact developed in three phases: 1) problem identification, 2) artifact design and 3) validation.

In the problem identification phase, we conducted a literature survey (published in a separated publication [11]) to outline the challenges in systematic decision-making and the first set of requirements that the taxonomy to be constructed should fulfill. We focused on the factors that influence the decision to choose among different asset origins and solutions for decision-making [11]. Using a snowball sampling literature review method we listed 24 studies and 11 factors affecting or influencing the decision to select an origin. The factors are

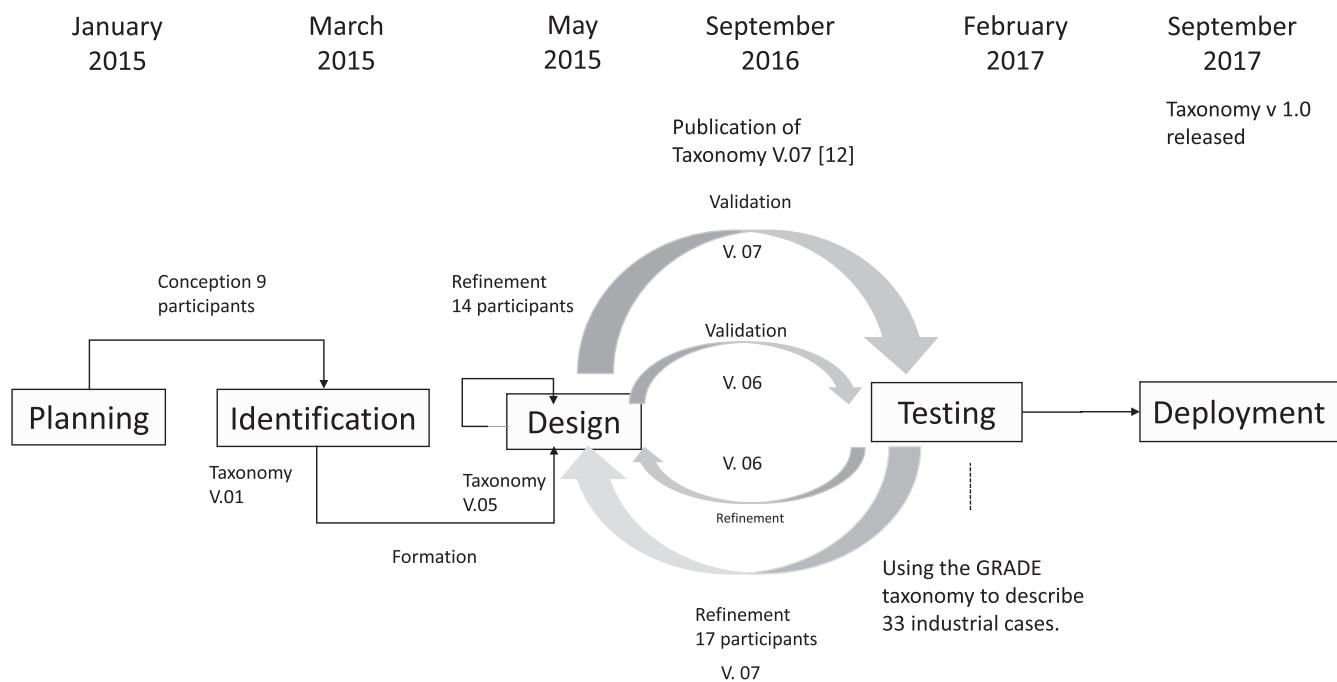


Fig. 1. The taxonomy construction timeline.

divided into *project metric factors* (time, cost, effort, quality), *external factors* (market trends, source code availability, technical support, license) and *software development activity factors* (integration, requirements, maintenance). Next, the positive or negative influence of each of the factors for each of the *sourcing strategies* (Open Source Software (OSS), COTS, in-house, outsourced) is discussed. This study provided foundations for an outline of the factors associated with decision-making and highlighted the need for a more systematised approach to defining them.

In the artifact (taxonomy) creation phase, we constructed the GRADE taxonomy inspired by the process guidelines provided by Bayona-Ore et al. [31]. The reason for selecting this process is that it was created with the focus on building taxonomies and it is based on an extensive literature review and summary of available taxonomy creation methodologies. The GRADE creation process is outlined in Fig. 1. The process contains the planning, identification, design, testing and deployment phases. The details of each phase are provided in Section 3.1.

In the validation phase, we conducted a number of case studies following the case survey methodology [33] and we summarise our findings in Section 5. We focused on investigating the completeness of the GRADE taxonomy by analysing thirty three industrial decision cases.

3.1. Taxonomy construction

We outline the GRADE taxonomy construction steps according to the guidelines suggested by Bayona-Ore et al. [31].

3.1.1. Phase 1 - Planning: Identify problem and motivate

Bayona-Ore et al. [31] break down the planning step into multiple activities, primarily focusing on positioning and scoping the taxonomy.

P1. Identification of the area of study: The need for creating a common vocabulary with a shared understanding of decision components and contextual factors and to provide a structured instrument for collecting decision-making evidence emerged directly after the ORION project set off. This need was amplified by an extensive industrial experience of the ORION project members and the performed literature review [11].

We carefully analysed the SWEBOk classification [26] with the purpose of positioning the taxonomy to be developed into the right knowledge area. We decided to place the taxonomy to be developed in the software design knowledge areas as these are most likely to be affected by choice of an asset (e.g., the effect on the life-cycle).

P2. Definition of the objectives: The following two main objectives were identified:

- Objective 1: The taxonomy should serve as a common vocabulary, or a way to reach to a common language, to describe and document the essential elements related to the decision. In that respect, it should make feasible identification and classification of decision-making aspects concerning architectural decisions in an intuitive way. Thus, it is important that the taxonomy is vibrant, clear and unambiguous when used for decision scenario documentation.
- Objective 2: The taxonomy should provide a structured instrument for collecting decision-making evidence in a common way for practitioners involved in architectural decisions. This structured instrument combined with the common vocabulary provides the necessary building blocks for knowledge databases. These knowledge databases can be then populated with past decisions and enable possibilities for systematically reflecting on recent decisions and comparing them for suggesting improvements in decision-making.

P3. Definition of the survey of user needs: The target user group for the taxonomy is researchers and practitioners. To better understand their needs and goals, we revisited definitions and terms from standards such as [34] that are relevant for researchers and practitioners. In this step, 17 researchers and three practitioners were involved. Out of the 17 researchers, 9 of them worked collaboratively and intensively on the taxonomy creation. The rest contributed as reviewers and using the taxonomy in the validation case studies (see Section 5).

P4. Definition of the scope of the taxonomy: The scope of our taxonomy is describing the important characteristics of decision-making for choosing suitable assets for developing software-intensive systems. In principle, this is illustrated in the upper part of Fig. 2. A system (pre-change) evolves, e.g., due to the need for additional functionality. This can be done by adding an asset or modifying an existing

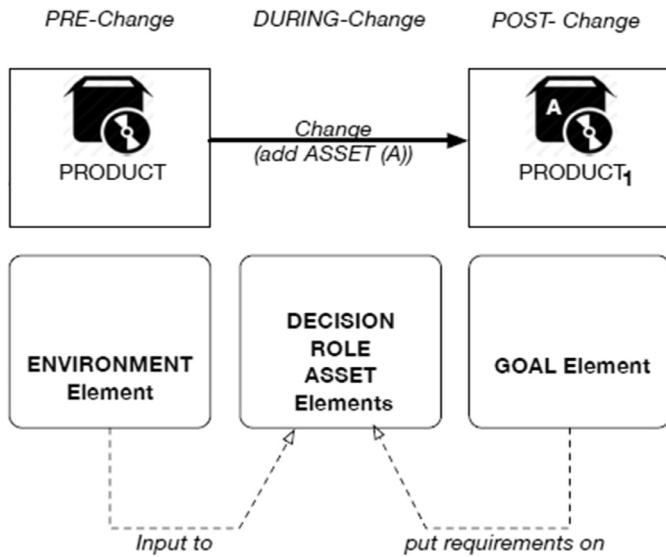


Fig. 2. Usage of GRADE in product development.

asset, which then leads to a system post-change. That is, the decision we are concerned with is which asset to choose. It is a decision about the asset origin (but also regarding functionality) which can be from various sources. Assets can, for example, be OSS, developed in-house, COTS, or obtained through outsourcing.

P5. Definition of the team in charge, the required resources and document the development plan: The taxonomy development is driven by the heterogeneous background and expertise in the ORION project team. The team members consist of senior researchers with on average of 8 years of research experience. Four members have over 20 years of research experience. Examples of relevant expertise of the team includes: (a) software-intensive system development and architecture, (b) decision support in software engineering, (c) empirical research methods, and (d) software development processes. The diverse background of the team and being from different institutions also facilitate mixed views on the taxonomy while it is constructed and ensures that the resources required for the taxonomy construction are secured. Finally, the schedule for the activities and the execution plan were created.

3.1.2. Phase 2 - Identification and extraction of information

11. Identification of the sources of information: The identified sources of information included: a literature summary [11], previously published taxonomies in software engineering that were considered relevant for this case and industrial experience of the team members. The identified information was extracted as a list of the most representative terms or attributes.

12. Extraction of terms and identification of candidate categories: In the first workshop, conducted in March 2015, see Fig. 1, 10 researchers were involved, and the groupings of candidate categories (called perspectives) were identified from the flat list of attributes generated in Phase 2 I1 with the help of a white-board and sticky-notes. Researchers clustered the essential perspectives needed to describe decision-making processes and identified suitable labels for these perspectives. The group also added further elements or refined the elements under them. This resulted in the creation of the GRADE structure. A multi-faceted taxonomy was found necessary because of the existence of multiple receivers and users of the taxonomy, i.e., researchers, developers, business people, architects, and customers. Having various facets satisfies the need of having dependencies, which are flexible enough to be extended in the future. The faceted taxonomy allows for more flexibility in classification and easy extensions. The classification is multi-dimensional as it includes goals, roles, assets, decision and

environment aspects. The hierarchical classification could not capture all these aspects since selecting one aspect would mean walking down the hierarchical tree [35].

3.1.3. Phase 3 - Design and construction: design/Develop and generate new ideas

D1. Checking the list of terms and defining the criteria for grouping: In the second workshop, conducted in May 2015, 14 researchers were involved, and the taxonomy was further improved and refined. The workshop was structured as follows: First, the taxonomy was revisited and presented based on the first workshop version (GRADE v.01) to define the criteria to group the elements. Thereafter, breakout groups worked with each part of the taxonomy (i.e., one group for assets, one for goals, etc.) to further improve the taxonomy definitions and descriptions. Reviewer roles were also assigned for reviewing and approving the result from each group. The reviewers did not take part in the breakout groups formed for each part. The resulting GRADE v.05 was further improved in the next phase.

D2. Defining the first level of taxonomy design: Each of the perspectives identified within the top-level elements of the taxonomy should be well-defined concerning their terminology. Whenever possible, the researchers aimed to identify published and well-established definitions for the perspectives. The following five top-level elements were identified in the core of GRADE:

- Goals: To classify the goals the software value map [2] was used. The map presents an exhaustive collection of what software-intensive product development organisations may strive for to satisfy various objectives and stakeholder perspectives. Given that the map has been systematically created and is based on an extensive literature study and the involvement of practitioners, it provides a well-founded structure to define goals.
- Roles: They were based on related work in decision-making, in particular, Herrmann et al. [36], Tan [37], and Morissat et al. [38], which together provided a comprehensive overview of decision-making functions.
- Asset: The assets and their properties were also defined to a large degree through standards. For example, the asset property of usability and the measures for the usability has been defined in the standards IEEE Std 1517–1999 (R2004), ISO/IEC/IEEE 24765:2010(E) and ISO/IEC 9126-1:2001. The origins for assets were identified through a systematic literature review of sourcing options when choosing components [11]. The four origin types considered in this taxonomy are in-house, OSS, outsourced and COTS.
- Decision: For the decision, a wide range of references was used for the construction. For example, a wide range of criteria is relevant when choosing an asset, such as dependability. For that purpose, SWEBOK [26] and existing standards (e.g., ISO/IEC CD 25010.2) were used.
- Environment: The environment element was inspired by the context model checklist proposed by Petersen and Wohlin [39].

The top-level elements (as mentioned above) are an abstraction of the underlying decision theory elements [40] and typically considered in decision structures like induction tasks of decision trees [41]. These elements refer to: Acts (the options considered by the decision maker - in our case the assets), Events (the facts affecting the decision - in our case we consider several factors to affect the decision, e.g., overall goal, decision criteria and external environmental factors), Outcomes (the result of choosing each and every one Event - in our case that we are defining a taxonomy this is not applicable), and Payoffs (the values that the decision maker at each instantiation of the decision process puts on the Options - in our case this is applicable in the GRADE validation stage as is reflected in the achievement of the overall goal of the decision).

Different alternatives exist for structuring a taxonomy defining how

the elements are grouped; common ones are hierarchical taxonomies and faceted taxonomies [31]. GRADE was structured as a faceted taxonomy for the reason that faceted taxonomies are more flexible and thus easily extendable by new goals, criteria and asset types. Flexibility is essential for the researchers and practitioners who are going to use this taxonomy as industrial cases may vary due to different contexts.

D3. Performing terminology control: Spreadsheets were used to manage the terms and perform continuous terminology control within each element of the taxonomy. Each worksheet (one per top-level element of GRADE) comprised of the perspectives' names under each top-level element, the definitions, and references where definitions were found. In groups of two, researchers were assigned to work extensively on each of the top-level elements, and within the same worksheet to document, categorise, and identify references. Thereafter, the definitions were driven by literature and standards as pointed out in the steps I1 and I2 (see Section 3.1.2).

D4. Defining the subset levels of the taxonomy: The elements categories were developed in an iterative fashion and were both designed using a hybrid approach: top-down (using the top-level elements to drive the identification of lower level elements) and bottom-up (using lower level list of elements to formulate and identify the needed categories on the top level covering the lower level elements).

D5. Reviewing and approving the taxonomy by stakeholders and experts: Two researchers, who were not involved in the specification of the top-level elements were assigned to carry out reviewing and approving that element. After that, each element was discussed and revised according to the feedback. Hence, the definitions of the perspectives and lists of elements of the taxonomy were thoroughly reviewed prior to validating them.

The updates were carried out in two large refinement cycles, resulting in GRADE v.06 and v.07. 17 researchers worked in smaller focus groups on parts of the taxonomy and for each of these smaller groups feedback was used from the researchers not working within the small groups, who reviewed and approved the work. The resulting GRADE version v.07 was published in a workshop publication [12].

3.1.4. Phase 4 - Testing and validation

The key to testing and validation is to assess the taxonomy against the objectives outlined for the taxonomy. The objectives of GRADE set for testing and validation were: 1) to serve as a common vocabulary to describe and classify the decision-making, and 2) to provide a structured instrument for collecting decision-making evidence and enable possibilities for systematically reflecting on past decisions and comparing them for suggesting improvements in decision-making.

For the validation of the GRADE taxonomy 33 industrial cases collected in two phases were used, see Section 5. In the first phase of the validation, we investigated four cases from two companies and coded their decisions with the help of GRADE. In the second phase, we applied GRADE to 29 cases collected from own experiences and interviews of past decision-making. Five experts were involved in the codification, i.e., used GRADE for describing those cases and questionnaires were used to collect their feedback.

3.1.5. Phase 5 - Deployment

The taxonomy deployment is a long-term objective as a taxonomy requires wide acceptance by the software engineering community and practitioners. Apart from publishing the initial version of the taxonomy in a workshop publication [12] and the current version in this publication, the deployment phase included creating a knowledge repository where GRADE-coded cases can be stored and analysed [42] to conduct meta-analyses of the cases to identify best practices for decision-making. The taxonomy is also planned to be integrated with the COACH¹ decision framework that is available as an open-source

project. The knowledge repository and the COACH framework integration ensures long-term maintenance of GRADE and enables documenting a substantial number of decision cases that allow for retrospective analysis and further reasoning. Finally, the maintenance and management documentation is to be created in this phase.

4. The GRADE Taxonomy version 1.0

The GRADE taxonomy provides a structured way of characterising decisions for software-intensive systems through the lenses of five top-level elements, namely *Goals (G)*, *Roles (R)*, *Assets (A)*, *Decision methods and criteria (D)* and *Environment (E)*. These five elements are further refined to provide a common terminology that can be used by researchers and practitioners to document a decision and identify additional elements that could affect decisions during the decision-making process. Intrinsically, the taxonomy aims at answering the following questions:

- **Why?** — Why do we make a decision? What do we want to achieve? What are the main reasons behind it?
- **Who?** — Who has been involved in the decision? Who do we need to involve in the decision-making process? What do the people involved in the decision bring into the process?
- **What?** — What are the different options that are being considered in the decision? What additional characteristics for the options do we need to consider?
- **How?** — How has the decision been made? What methods and criteria are available to us?
- **Where?** — Where has the decision been made? Which characteristics of the environment or the company were important to consider at the time of the decision?

In the GRADE taxonomy, the why question is answered by the goals element, the who by roles, the what by assets, the how by decision methods and criteria and the where by the environment.

Furthermore, when used in a decision-making process as illustrated in Fig. 2, the GRADE taxonomy allows capturing the evolution between the state before an architectural decision has been made in the development of a software-intensive system and the subsequent state after the decision has been taken. In this view, the environment element, on the left-hand side of Fig. 2, describes the current context in which the decision is made, i.e., the company, system or product pre-change. The goals, on the right-hand side, represent the objectives to be fulfilled post-change by the new system or product. Deciding on a goal or set of goals implicitly impose requirements on how the decision should be made (i.e., the decision methods and criteria), who should be involved in the decision (i.e., the roles) and what alternative options should be considered (i.e., the assets). Similarly, the environment imposes restrictions on what is known for the decision to be made. As a result, to transition from pre-change to post-change, the decision methods and criteria, the roles and assets need to be considered as they are affected by the environment and the goals (as visualised in the middle of the figure).

4.1. Top-level elements of the GRADE taxonomy

As illustrated in Fig. 3, GRADE consists of five top-level elements, each of them further refined into perspectives. Some of these perspectives are subsequently decomposed into a number of items. Elements in blue represent customisation points which can be extended through non-predetermined lists to better tailor different organisation practices. Links to the complete taxonomy extended with examples of customisation points can be accessed in the project's results section². The following subsections describe in detail the top-level elements.

¹ <https://github.com/orion-research/coach>

² <http://orion-research.se/results.html>

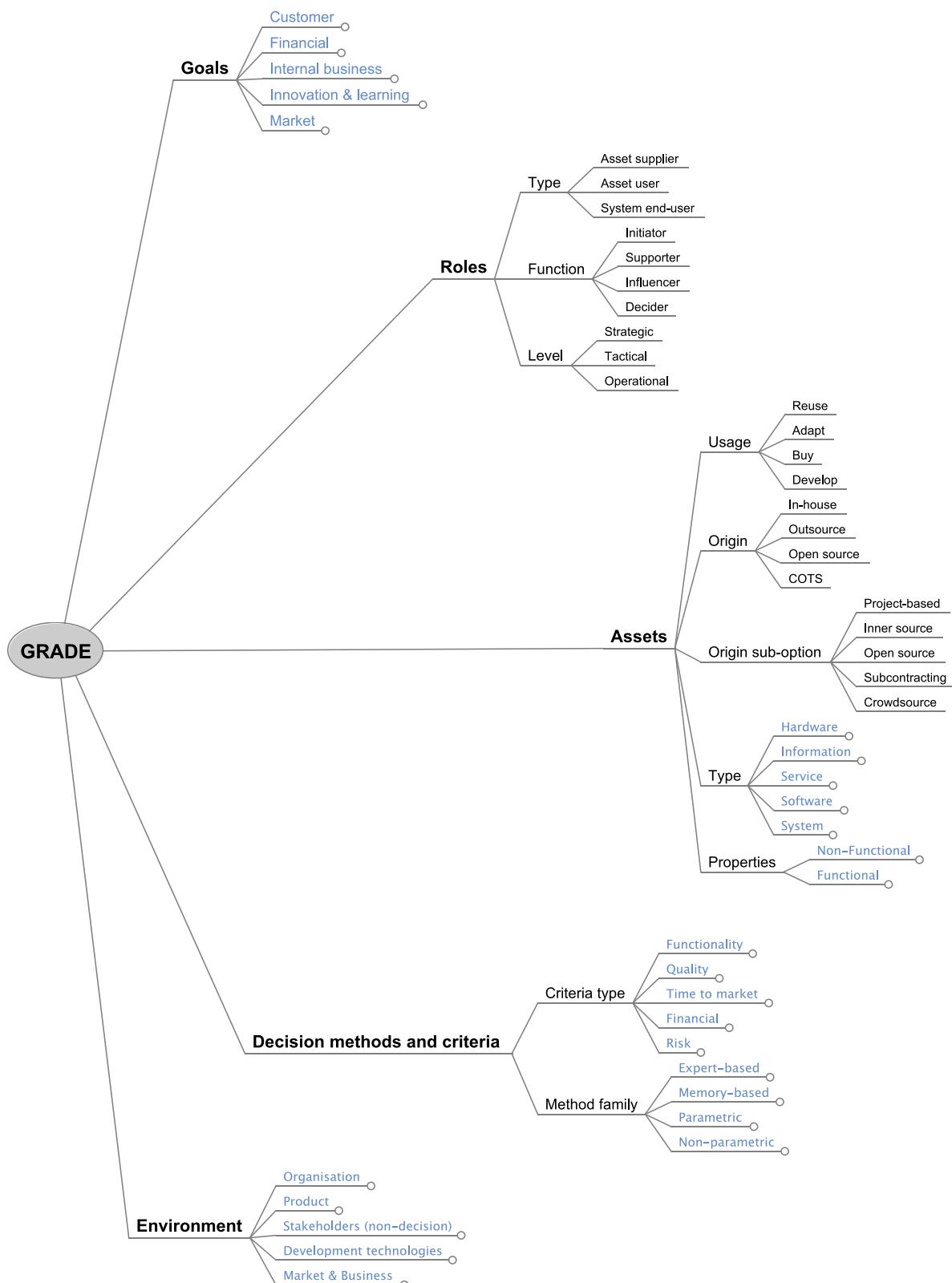


Fig. 3. Overview of the GRADE taxonomy. Elements in blue (also marked with a small circled end-point) can be further refined through open-ended list of elements to better tailor different organisation practices. (For interpretation of the coloured parts in this figure, the reader is referred to the web version of this article.)

4.1.1. Goals (G) — Why do we make a decision?

The top-level element *Goals* states the objective(s) that ought to be achieved with a decision. In other words, it characterises the expected values that will be delivered to a customer as the result of the

realisation of the decision. For example, goals encompass statements such as to “minimise the time to market of product X?”, “extend our market shares for product Y?”, and “extend our product Z with the new functionality F1 to satisfy our customer’s request”.

Table 1

Value perspectives of the top-level element Goals. The acronyms GVx correspond to the value perspective used in the validation cases.

Perspective	Description	Symbol
Customer	Value for the customer obtained from the software-intensive system delivered (such as perceived value of the product wrt. functionality, reliability, etc., quality of the delivery process, network and support, user experience).	GV1
Financial	Financial (monetary) value relating to or involving money, which can be obtained from the software-intensive system delivered. This can also be the value translated into, for instance, shareholder value.	GV2
Internal business	Contains value aspects, sub-value aspects and value components related to the development company's internal view of the product/service offered by themselves. Examples are time, cost, and quality. Quality here refers to the quality of the internal work products not visible to the customer directly (i.e., they are not the work product, but may of course influence the quality of the final work product).	GV3
Innovation & Learning	Innovation and learning value relates with the ability an organisation has to innovate and grow (related concepts are human capital, knowledge, learning, idea generation, creation of partnerships, etc.).	GV4
Market	Market value is the current value market valuation or price of an industry/product. Also known as Open Market Valuation (OMV) is the price at which an asset would trade in a competitive auction setting.	GV5

When considered in retrospect, goals are important elements to record in a decision-making process. They describe the reason why a decision was made and indirectly expose the constraints and requirements that were placed on other aspects of the decision-making process at the time of the decision.

As it can be seen from the above examples, the goals element can bring values from different perspectives. To support them, the Goals element is categorised according to the value perspectives identified for software-intensive product development in [2]. We have extended these categories with an additional one, called Market, to allow the consideration of goals related to the company's position in the market.

The five main value perspectives used in the Goals top-level element of GRADE are:

- *Customer* which includes objectives aimed at improving the overall customer's satisfaction. It can be improvement in terms of functionalities and inherent quality of the products, and it can also concern the view and the confidence the customer has of the company itself.
- *Financial* encompasses any objective linked to potential monetary gain for the company (e.g., profit, product margins, cash flow).
- *Internal business* which gathers objectives related to the internal activities and affairs of the company. It includes objectives dealing with processes, development strategies such as outsourcing and activities to cut costs.
- *Innovation & learning* targets all objectives related to improving the intangible assets of an organisation such as the skill sets, knowledge and competences of the employees, adapting the current environment to new methods, processes and practices, establishing trusted partnerships with other companies, etc.
- *Market* covers objectives that relates to the company's position in the current marketplace.

The five value perspectives for Goals are summarised in Table 1. For each perspective, a symbol (e.g., GV1) is given. These symbols are used in the validation cases.

Each value perspective can be further tailored through an open-ended list of detailed goals, specific for the current decision scenario or environment. These detailed goals include objectives such as to “*extend an existing system in terms of functionality*”, “*improve customer's satisfaction*”, “*optimise time to market*”, “*reinforce existing relations with a business partner*”, etc.

In a worksheet³, we have identified a list of 32 specific goals which can have an impact on the post-asset implementation of the system (or service).

4.1.2. Roles (R) — Who is involved?

The top-level element *Roles (R)* characterises the stakeholders involved in the decision-making process and allows exposing their contributions to the decision. In our context, the term stakeholders refers to any individual or group of individuals who directly participate in the decision-making process to either make the decision or provide input during the decision-making process. Additionally, it also includes people who will be affected by the outcomes of the decision such as the end-user of the product. The *senior developer*, *CEO of the organisation*, *product manager*, *lawyer specialist in copyright*, and *test manager* are a few examples of stakeholders.

As a result of this broad spectrum of persons involved, the stakeholders may have quite heterogeneous perspectives in mind during the decision-making process and may value the importance of various inputs differently. It is thus important to be able to document the different perspectives considered by each and every stakeholder. For this purpose, the Role element of the GRADE taxonomy revolves around the following perspectives:

- *Role Type* which gives a characterisation of the stakeholder's function in the decision-making process in relation to his viewpoint in the option selection. The role type describes whether the stakeholder contribute to the decision-making process from the perspectives of the *asset supplier* (i.e., the organisation providing one of the asset involved in the decision), the *asset user* (i.e., the organisation using the asset) or the *system end-user* (i.e., the user of the system in which the asset will be used).
- *Role Function* which indicates the responsibility of the decision maker in the process. In this perspective, the stakeholder can be an *initiator* who is the one who triggered the process, a *supporter* who supplies relevant inputs to decision, an *influencer* who affects the outcome of the decision or the decision-making process, or a *decider* who is the person responsible to make the final decision. Although the difference between a supporter and an influencer seem small, the effects on a decision are important. An influencer has generally more power to influence the outcomes of a decision. A supporter, on the other hand, has a more technical role used to provide concrete inputs.
- *Role Level* refers to the reference frame the stakeholder considers during the decision-making process. The level takes into consideration the complexity of the decision, the time frame the decision will affect and the frequency at which a decision can be taken. This perspective has three levels: strategic, tactical and operational. The *strategic level* corresponds to long-term, complex and non-routine decisions. The *tactical level* covers less complex decisions that will have an effect at a medium-term, and the *operational level* refers to the day-to-day, simple decisions that are made routinely.

Finally, a list of examples of actual positions of stakeholders within the organisation has been identified. It is an open-ended list that can be

³ <https://goo.gl/WwuEkB>

Table 2

Excerpt of the main elements of Roles that are used in the validation.

Perspective	Element	Description	Symbol
Role Type	Asset supplier	The organisation providing the asset.	RT1
	Asset user	The organisation using the asset.	RT2
	System end-user	The user of the system in which the asset is utilised.	RT3
Role Function	Initiator	Initiator is a person who starts the whole process of decision-making, combines diverging contributions by relating them to a summarising statement; if the discussion diverges, the person calls for an informal vote to reach consensus.	RF1
	Supporter	Supplies the information needed to make the decision.	RF2
	Influencer	Role influencing the outcome of the decision making and hence has influence on the decision maker.	RF3
Role Level	Decider	Role is responsible for taking the decision, actually reaches to the decision.	RF4
	Strategic	Policy, long-term, complex, non-routine decision making.	RL1
	Tactical	Implementation plan for policy, medium-term, less complex decision-making.	RL2
	Operational	Day-to-day, simple, routine decision making.	RL3

customised to match the structure of an organisation. For example, in a worksheet⁴, we identified a list of 22 positions (e.g., software designer, software tester, CEO) inspired from [26] and BAPO [43]. Table 2 provides an excerpt of the Roles perspectives that are used in the validation.

4.1.3. Assets (A) — What options do we have?

The top-level element Assets documents the options that are available during the decision-making process, together with their inherent characteristics. Assets are the components a decision-maker considers in order to satisfy the goals and generate the expected corresponding value. For example, a typical decision-making process for software-intensive systems could include the following scenario: “Is it most beneficial for our company to buy the software component CPT1 from vendor V or should we develop the functionality ourselves in-house?”. In this scenario, two options are weighed: a COTS component versus an in-house solution.

In practice, a large range of assets can be considered in decision-making processes. They span from small software components realising a dedicated functionality such as a device driver or a mathematical library, to big hardware-software systems such as an adaptive cruise control system in a car or web-service solutions, passing through components that can be accessed remotely such as services. Assets also include immaterial elements that are related to software or can lead to software development such as a patent, patterns or documentation. In this work, we therefore consider an asset as any artefact, developed, obtained or accessed, which contains software or is directly related to software development.

To better identify critical characteristics of the assets involved in a decision-making process and allow weighing the options rationally, we define the following perspectives to describe an asset:

- *Asset Usage* characterises the development option for the asset, i.e., how the asset will be used. Four options are available: reuse, adapt, buy and develop. An asset available for reuse means is already owned by the company and will be re-used as-is without any modifications. The option to adapt an asset means that an asset that is also already owned by the company but needs modifications to the functional or extra-functional properties. A buy option requires for the asset to be purchased and then used. Lastly, the develop option for an asset corresponds to the case that an asset does not exist yet but is planned to be developed.
- *Asset Origin* identifies the provenance of an asset by distinguishing whether the asset comes from, i.e., an internal or external source. Internal and external origins can be further extended with a non-predetermined list of sub-elements, thus allowing customising the available types of sources to the specific needs of an organisation by listing relevant options. As featured in Table 3, an internal origin

can be simply in-house, whereas multiple external origins can be included such as commercial off-the-shelf (COTS), open source, or outsource.

- *Asset Origin Sub-Option* allows to further characterise the origin by specifying the mode of development of the asset. We have identified five modes of development, namely project-based, inner-source, open source, subcontracting and crowdsourcing. In a project-based development mode, the asset’s source code is taken from a specific project. For inner-source, the source code is available for modification within the boundaries of the organisation. On the other hand, in open-source development mode, the source code is freely available for modification, including by anyone outside the organisation. In subcontracting, the development is delegated to a third-party (outside the company boundary) through a contractual agreement. Finally, in a crowdsourcing development mode the development and production is carried out in a distributed manner, in which networked people collaborate.
- *Asset Type* indicates the nature of the option. As mentioned before, many different types of assets can be considered in a decision-making process ranging from small components to big system of systems. To structure this perspective, we have used the categories proposed in the ISO/IEC 27002:2005 standard [44]. These are: *information* which encompasses any intangible assets and documents related to software such as patents, trademarks, copyrights and business methodologies; *software* which includes any asset corresponding to a modular piece of code such as software libraries, functional components, user interfaces; *system* that include any asset which functionality is realised by a combination of software and hardware such as a navigation system; *hardware* which covers any physical device containing some kind of software such as servers, sensors, actuators; and *service* which relates to assets which functionality is realised on-demand over the internet. Each element from the type perspective can be further refined through an open-ended list of sub-elements, thus providing further clarification concerning the types of assets involved in the decision. A list of 22 examples of sub-elements has been collected in a worksheet⁵.
- *Asset Properties (AP)* lists the asset’s functional and non-functional properties, which documents the technical characteristics relevant to the decision-making process. The functional and non-functional sub-categories can be further refined with an open-ended list of elements to accommodate the specifics of the assets. In a worksheet⁶, we exemplify a possible sub-characterisation of these elements by using the ISO 25010 standard [34] with the addition of an extra sub-category to cover financial aspects linked to the asset such as a licensing cost. Accordingly, 42 attributes of interest and their definitions were identified.

⁴ <https://goo.gl/Ur6UQY>

⁵ <https://goo.gl/pk97va>

Table 3

Excerpt of the Assets elements that are used in the validation.

Perspective	Element	Description	Symbol
Usage	Reuse	An asset is reused with no change.	AU1
	Adapt	An asset is used but tailored.	AU2
	Buy	An asset is used after it is owned (bought).	AU3
	Develop	An asset is developed by the company/individual.	AU4
	In-house	This is the process whereby the company uses their own workers (in-house team) to develop or implement an IT system that fits the specific needs of the company.	AO1
Origin	Outsource	Outsourcing is defined as a situation where a company (a client) contracts out all or part of its software development activities to another company (a vendor), who provides agreed services for remuneration.	AO2
	Open source	Open source development refers to the case where the source code is available for modification or enhancement by anyone outside the boundaries of an organisation or individual.	AO3
	COTS	Commercial items, including services, available in the commercial marketplace that can be bought and used under government contract.	AO4
Sub-option	Project-based	Source code is originated only from specific projects.	AS1
	Inner source	Source code is available for modification as in open source, but applied to internal software development.	AS2
	Open source	Open source origin is when source code is available for modification or enhancement by anyone.	AS3
	Sub-contracting	Subcontracting refers to when establishing a contractual agreement with an outside person or company to perform the work.	AS4
	Crowd-source	Crowdsourcing is a new on-line distributed problem solving and production model in which networked people collaborate to complete a task.	AS5

4.1.4. Decision methods and criteria (D) — How do we decide?

The top-level element *Decision methods and criteria* catalogues the methods that may be used to make a decision and it goes along with a set of criteria the decision is based upon. For example, a decision can be made by achieved a Pugh-Analysis, an expert judgement or case-based reasoning. As many decision methods and criteria exist we have listed some of them, as examples, in a worksheet⁷. Decision methods are ways to carry out decision-making and typically evaluate different criteria. They differ in how they are applied, how the evaluation is made and what kind of data they can use. They also provide different types of outputs, e.g., analysis, interpretations or values. Using different methods decision makers can reach to different results and based on these different decisions can be made. It is therefore important in a decision making process to document the methods and the criteria. For this reason, the Decision methods and criteria element of the GRADE taxonomy include the following perspectives:

- *Method Family* is associated to a collection of decision methods that have some common characteristics. The method families identified are: expert-based, memory-based (or analogy), parametric and non-parametric, based on [45]. Methods can be expert-based if the method is relying on the judgment and expertise of an expert. Memory-based (or analogy) means the method is relying on evidence from previous or similar situations and decisions are made in an analogous manner. In some cases analogous methods also use experts. Parametric methods use statistical relationships between historical data and other variables. Finally, non-parametric methods use statistics but do not require any assumptions about the distribution of the data to hold. We identified a total of 55 methods.
- *Criteria Type* The decision criteria are considered and used as input to the decision methods. The decision criteria have the following types: functionality, quality, time to market, financial and risk, see Table 4. Criteria related to functionality refer to cases when the decision method takes into account the capability of the asset to perform some function. Quality refers to types of criteria that adhere to non-functional requirements and these are used as inputs to the decision method. Time to market refers to criteria types related with when the product is delivered and is also a criterion used as input to the decision method. Financial type of criteria refer to cost and revenue. Risk related type of criteria refer to associated risks the decision process needs to consider, such as unforeseeable events causing problems. We identified a total of 67 such specific criteria.

4.1.5. Environment (E) — Where is the decision made?

As stated in the ISO/IEC/IEEE 42010 standard [23], “The environment of a system includes developmental, technological, business, operational, organisational, political, economic, legal, regulatory, ecological and social influences.”. Similarly, the top-level element *Environment* lists contextual aspects relevant to the decision-making process as the influence factors that might affect the outcomes of a decision. It represents a snapshot of the situation before the decision is made and pre-asset implementation and integration. It includes criteria, facts and occurrences that are, in many cases, outside the control of the decision-makers. Elements such as *the numbers of developers in a development team*, *the list of rules and regulations* that the organisation must follow, *the technical details on the development environment* are few examples of these influence factors but many more exist. For example, in a worksheet⁸, we have identified 80 factors which can be important in different decision cases. However, providing an exhaustive list of influence factors is hardly achievable.

Accordingly, the environment top-level element of the GRADE taxonomy is organised through the following perspectives inspired from [39]:

- *Organisation* which describes the company structure through the identification of its main characteristics. It includes elements such as its organisational and geographical structure, the rules, policies and regulations that are followed, and the overall available resources.
- *Product*, refers to the influence factors originated from the product in which the asset will be integrated. The type of the systems with its inherent constraint (e.g., vehicle with timing and safety-critical requirements), the required level of openness, its quality properties, are few examples of the product’s influence factors.
- *Stakeholders (non-decision)* gives information on the human resources that are available to the development of the asset such as the skills and knowledge of the employees or their number of years of experience. The notion of stakeholders must be here differentiated from the one used in the role perspective, in which a stakeholder described the persons directly involved in the decision-making process.
- *Development technologies* lists relevant information about processes, tools and methods that are available to the asset development and integration.
- *Market & Business* places the current decision-making process in a

⁷ <https://goo.gl/JXpUiI>

⁸ <https://goo.gl/F6bMpW>

Table 4

Excerpt of the main elements of the decision methods and criteria that are used in the validation cases.

Perspective	Element	Description	Symbol
Method Family	Expert-based	[Technique] judgement provided based upon expertise in an application area, knowledge area, discipline, industry, etc. as appropriate for the activity being performed. Such expertise may be provided by any group or person with specialised education, knowledge, skill, experience, or training.	DMF1
	Memory-based (or Analogy)	Analogous estimating means using the actual costs/performance figures/circumstances of previous, similar projects as the basis for estimating the cost/risk of the current project. For instance analogous cost estimating is frequently used to estimate costs when there is a limited amount of detailed information about the project (e.g., in the early phases). Analogous cost estimating uses expert judgement.	DMF2
	Parametric	Parametric. A technique that uses a statistical relationship between historical data and other variables (for example, square footage in construction, lines of code in software development) to calculate an estimate for activity parameters, such as scope, cost, budget, and duration.	DMF3
	Non-parametric	A method commonly used in statistics to model and analyze ordinal or nominal data with small sample sizes. Unlike parametric models, non-parametric models do not require the modeller to make any assumptions about the distribution of the population, and so are sometimes referred to as a distribution-free method.	DMF4
Criteria Type	Functionality	The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.	DT1
	Quality	Non-functional requirements (NFRs), sometimes termed quality, or quality of service, attributes or requirements, have been a topic of interest within systems engineering, software engineering, and requirements engineering for a considerable period of time.	DT2
	Time to market	Time to market is the length of time it takes from a product being conceived until its being available for sale.	DT3
	Financial	Financial criteria includes Cost and Revenue, where Cost is usually a monetary valuation of (1) effort, (2) material, (3) resources, (4) time and utilities consumed, (5) risks incurred, and (6) opportunity forgone in production and delivery of a good or service. All expenses are costs, but not all costs (such as those incurred in acquisition of an income-generating asset) are expenses. Revenues are the incoming costs a business has from conducting its normal business activities, usually from the sale of goods and services to customers.	DT4
Risk		Refers to the possibility of inadequate profits, unforeseeable events that cause even losses due to uncertainties.	DT5

Table 5

Main perspectives of the Environment top-level element.

Perspective	Description	Symbol
Organisation	Describes the contextual company information, structure, model of management, distribution etc. in which the product is implemented or intended for.	EA1
Product Stakeholder (non-decision)	The product, software system developed, and all contextual information related to the current product or service (prior the decision).	EA2
	Stakeholders that might affect the decision or are affected by the decision. They are not taking part in the decision making process directly but can be the end-users of the decision, and thus are differentiated to the decision-makers.	EA3
Development technologies	Systematic approaches that are used in the organisation and that are contextual elements to the current product (prior the decision).	EA4
Market & Business	Represents the current market, i.e., customers, competitors, partners and business contextual factors, prior the decision.	EA5

general context by detailing the current situation in the world (i.e., outside the organisation) from economical and financial points of view. This involves providing information on the current market (i.e., on the customers, competitors, and partners) as well as on the business and political prospects such as general market trends, business plans, global economies, political issues or upcoming regulations.

These perspectives, which can be extended through open-ended lists of influence factors tailored to a specific organisation, are summarised in Table 5. We have until now identified a total of 87 such specific environmental factors.

4.2. Embedding GRADE in a knowledge repository

An important part of GRADE deployment is to design and develop a knowledge repository. This is an integral part of objective 2 of the taxonomy creation. The repository has a twofold aim: 1) to promote the use of the terms and concepts introduced by GRADE, and, 2) to support the decision process by retrieving relevant information from previous decision scenarios. These goals are directly connected to the use and validation of the proposed taxonomy: i.e., if decision information cannot be documented in a satisfying way, it might indicate that the taxonomy is incomplete and/or ambiguous. In the same way, an ambiguous and/or incomplete taxonomy would make the decision history accumulated in the repository scarcely usable in future cases. For this article, the knowledge repository has to be considered as a proof-of-concept for objective 2. The potential support of future decision-making

activities is left as future investigation.

The knowledge repository developed for GRADE is based on graph databases technology with the help of Neo4J⁹ graph database. The implementation details are provided in a separated publication [42]. Graph databases offer advantages in terms of malleability with respect to data representation and performance with respect to information retrieval [46]. Moreover, they provide query languages operating at the level of abstraction of the data structures they contain. In this respect, while lower level of abstraction techniques (that is programming language coding) might still be used to create and maintain graph-based data structures, they would require considerable efforts to create an adequate data management layer comparable to the one provided by solutions as Neo4J.

Fig. 4 shows an excerpt of the graph database resulting from storing decision cases X1 (case 71), X2 (case 72), X3 (case 73) and Y1 (case 74). Violet nodes are decision cases, red nodes are decision roles, yellow nodes are asset origins, green nodes are properties, in blue there are decision methods, and in gray environment information. To interpret reference to coloured nodes in the figure, the reader is referred to the web version of this article. The edges between the nodes represent their relationships. It is very important to notice that both the colour code and the relationships used in the picture are intended to simplify its readability. In practical terms, nodes and relationships are characterised by a specific type and internal attributes, based on the proposed taxonomy. In particular, nodes are created for the decision

⁹ (<https://neo4j.com>)

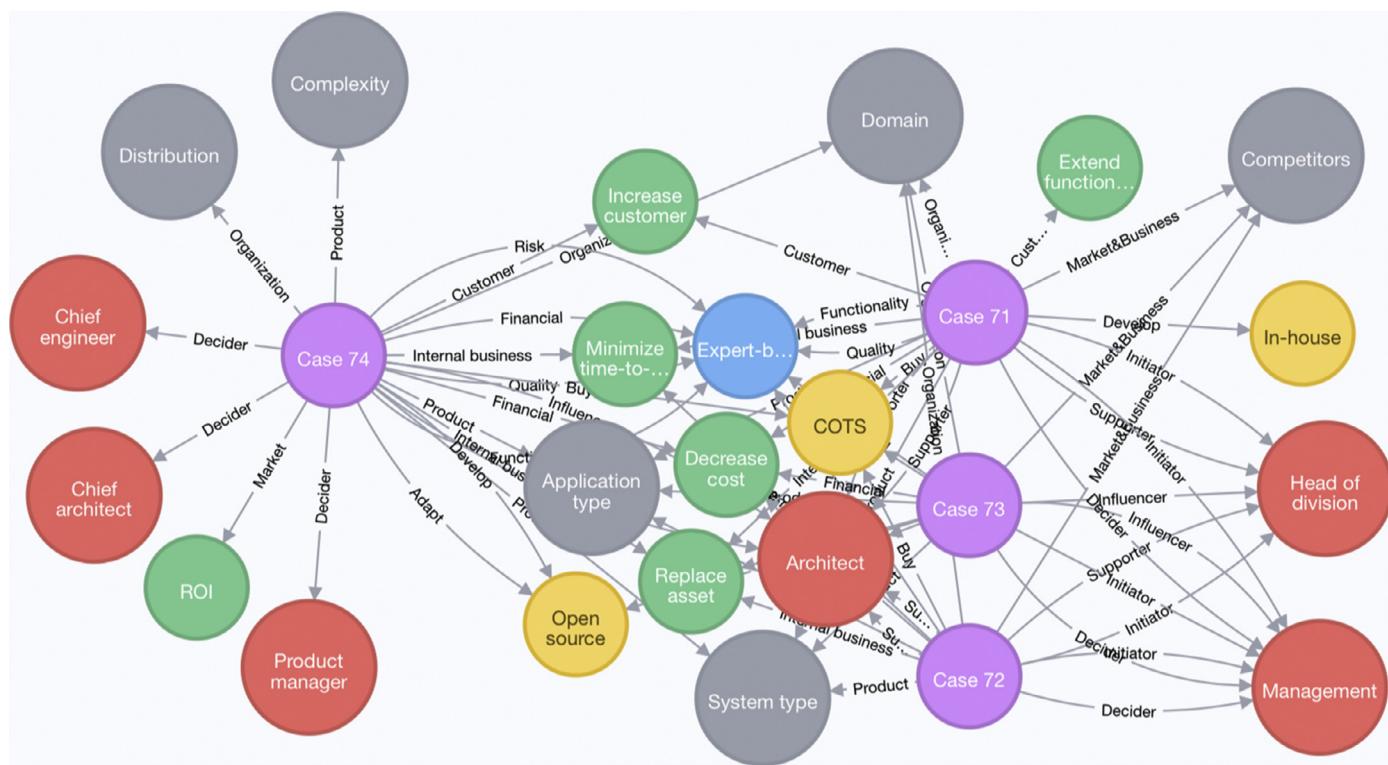


Fig. 4. An excerpt of entries in the graph database implementing the ORION knowledge repository. Cases 71, 72 and 73 represent cases X1, X2 and X3 while case 74 represents case Y1. The visualisation algorithm places similar decisions close to each other.

elements and their perspectives, i.e., Assets (A) and Asset Origins (AO) (Outsource, In-house, etc.); decision Roles (R) and Roles list (Technical Experts, Management, etc.); Decision methods and criteria (D) (Cost of Development, Cost of Maintenance, and Decision methods Expert-based, etc.), and so forth. Moreover, directed links between nodes depict relationships, notably COTS is the assets origin for case 74 and In-house is the origin for case 71. Management plays the Role Function of the Initiator (RF1 in [Table 2](#)) in both case 74, Y1 and case 73, X3, Cost of Development is a criterion of Financial decision Criteria Type (DT4 in [Table 4](#)) used in cases 71 and 74 in an Expert-based Decision Method Family (DMF1 in [Table 4](#)), and so on.

The most important advantage of graph databases is that they have no pre-defined data structure, so that nodes and edges can be arbitrarily defined. This aspect gives graph databases tremendous flexibility capabilities since newer and/or refined information can be stored in the database without the need for restructuring the data schema. However, this flexibility has to be kept under control to avoid having a multitude of unrelated decision cases, and it is here where the taxonomy plays a fundamental role. In fact, as also noticeable in Fig. 4, users exploit only the terms proposed by the taxonomy to document the decision cases, and this automatically creates meaningful correspondences between different decisions. The visualisation algorithm used places similar cases next to each other (cases 71, 72 and 73).

5. GRADE Validation

The GRADE taxonomy was validated in two phases. In the first phase, we investigated four decision cases from two companies and coded the decision scenarios with the help of GRADE. In the second phase, five researchers used GRADE to code 29 industrial decisions, and each decision was independently coded by two researchers. Finally, we collected the experiences from coding 29 cases in a questionnaire. The validation focuses on both objectives, excluding the database implementation which is exemplified in Section 4.3 and populated by the same cases.

5.1. Phase 1: Initial validation with four cases

GRADE was first validated by coding four decisions from two companies. Decisions X1, X2, and X3 come from Company X while decision Y1 come from Company Y, see Table 7.

Company X develops software-intensive products for a global market and operates in a Market-Driven Requirements Engineering (MDRE) context. The products consist of hardware that is partly internally developed and partly obtained from suppliers and subcontractors. The software is based on an OSS platform (for the commodity parts) with extensive extensions and adaptations for the differentiating parts. Components of strategic importance are either delivered by external partners (as COTS) or internally developed to ensure control over intellectual property rights and that the competitive advantage remains in the company.

The company was involved in three decisions (depicted as X1, X2, and X3 in Table 6) regarding the linguistic analysis and support for a text input software component that provides advanced user support features, e.g., next word prediction. This component was needed to be integrated with the user interface component and other parts of the operating system. The quality of this component was important from the aspect of the overall user satisfaction and thus influenced the human-computer interface.

The first decision (X1) was about an Asset Usage (AU) of buying a component and a choice of the Asset Origin (AO) between in-house and COTS. The primary decision driver for going for COTS was the pressure for acquiring some advanced functionality that could not be internally delivered, without spending a substantial amount of resources. The Asset Usage (AU) was described as a combination of buying a component and adapting it for achieving the intended user experience (AU3). The main properties of the assets considered was the Functionality offered and the asset Usability. Next, decision X2 was about changing the Asset Origin (AO) from one COTS supplier to a different one. The main goals for this decision were Cost and the Level of support provided by the COTS suppliers. The supplier was changed to a different one than

Table 6
Cases coverage of GRADE taxonomy.

Case	Goals	Roles	Assets	Decision	Environment
X1	1	0.83	1	0.3	0.75
X2	0.5	0.66	1	1	0.75
X3	1.0	0.83	1	0.3	0.75
Y1	0.4	0.71	0.28	1.0	1.0
1	0.33	0.57	1	1	0.5
2	0	0.375	0.43	0.33	0.66
3	1	0.57	0.75	0.33	0.5
4	0.66	0.375	0.66	0.5	1
5	0.25	0.55	0.5	0.75	0.66
6	1	0.375	0.8	0.8	1
7	1	0.4	0.83	0.5	0.5
8	0	0.875	0.71	1	1
9	0	0.5	0.71	0.33	1
10	0	0.33	0.82	0.4	0
11	0.33	0.71	0.83	0.5	0.5
12	0.67	0.44	0.5	0.4	1
13	0	0.57	0.6	0.4	1
14	0	0	0.67	0.5	0.5
15	0	0.71	0.625	0.4	0.67
16	0.33	0.75	0.5	0.6	1
17	0	0.5	0.67	0.6	1
18	0	0.57	0.5	0.8	0.75
19	1	0.55	0.43	0.2	1
20	0.5	0.55	0.29	0.29	1
21	0	0.5	0.27	0.25	0.5
22	1	0.43	0.66	0.4	0.25
23	0	0.6	0.17	0	0
24	0	0.5	1	0	0.5
25	1	0.375	0.67	0.33	0.67
26	0	0.28	0.6	0.67	0.75
27	0	0.17	0.5	0	0.67
28	1	0.5	0.67	1	1
29	0	0.2	0.4	1	0.5

the one considered in decision X1, despite the increased risks and uncertainties from changing supplier. The third decision (X3) concerned changing the Asset Origin (AO) from COTS (AO4) to open source (AO3). In this case, cost reductions were the main goal for the decision and had to be achieved by compromising some of the functionality that the COTS component provided but the OSS solution lacked. Despite the risks identified the company decided to switch to the OSS solution.

Company Y delivers ICT solutions to the global market and has offices around the globe. Decision Y1 was about replacing an existing component due to high license cost and insufficient support level. At the same time, developing a new component in-house or by outsourcing were excluded due to high costs and lack of required competence. An OSS solution was selected based on positive proof-of-concept results and positive recommendations of other companies that adapted the same OSS component. So the decision Y1 related with changing the Asset Usage (AU) from develop (AU4) to adapt (AU2).

The GRADE elements were associated with the information collected from the case studies using spreadsheets by the researchers conducting the interviews and collecting the information. The mapping between the GRADE elements was carried out individually by the researchers. The mapping and evidence was also reviewed by one more researcher.

The Jaccard index [47] has been utilised to compare the elements associated by each researcher. Thus, evaluation of the similarity of the documentation of evidence from the decision scenarios from each researcher is feasible.

According to its definition:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

the Jaccard index measures the similarity between two sets A and B. The closer the index J is to 0 indicates less similarity and the closer to 1 indicates more similarity. We assume that high index values indicate

that GRADE is clear and unambiguous and provide a structured instrument for collecting decision-making evidence. Low index values indicate that the researchers mapped the cases using different elements and could suggest ambiguity or necessary changes to the taxonomy. Thus, the Jaccard index is the primary measure to use when validating if objectives 1 and partly 2 are fulfilled.

Table 6 shows the Jaccard index values for all four cases (X1, X2, X3, and Y1) in the first four rows. To better explain the calculation of the Jaccard index value, an example taking the Goals element for Company Y is used: both researchers identified in their ranking that the decision scenario takes into account all respective aspects of the Goals, i.e., Customer value (GV1) as the aspects of customer support, fault tolerance, functionality, scalability, performance, replication, security and multi-tenancy were important. Also, Financial (monetary) value (GV2) was important, as the existing supplier license costs were high, as well as Internal Business (GV3) which was expected to be improved, as by changing to the alternative asset, Company Y would not need to depend on the external supplier for support anymore. The decision was also impacted by the Innovation & Learning value, as the Company was willing to invest resources in it, as a future investment, as well as the Market value (GV5) was also mentioned as important, as the alternative asset comes with some considerable gain in market value. As such, the Jaccard index value calculated from the perspective of the rankings from the two researchers (A and B) as follows: $(1 + 1 + 1 + 1 + 1) / ((1 + 1 + 1 + 1 + 1) = 1$.

The average Jaccard index for the Goals element is as high as 0.9 indicating that the two experts who coded the cases were in a great agreement for this element. The average Jaccard index for the Roles elements is 0.76 and for two cases (X1 and X3) is above 0.8. This is despite that each case in the first round was coded by at least 6 elements which confirms high agreements despite many elements selected and that experts can select multiple-elements without too much problem. The main discrepancies were between the Role Levels (RL) and Role Functions (RF). This appears to be logical as different roles may appear on different organisational levels, depending on the organisational structure of the companies. In response to that, we updated the descriptions in the Roles element of GRADE.

The average Jaccard index values for the Assets element is 0.82 and only in case Y1 the value is 0.28 while the remaining cases delivered full agreement between the experts. The main disagreements around case Y1 were about Asset Usage (AU) and Asset Origin (AO) and resulted in updated descriptions of the relevant elements of GRADE. Finally, the average Jaccard index for the Environment element for the initial four cases is 0.81 with a full agreement in case Y1.

5.2. Phase 2: Applying GRADE to code 29 cases

In the second phase, five researchers independently coded 29 decisions occurring in industrial settings. Each decision was coded independently by two researchers. Later we discuss the coding and the discrepancies. The results are summarised in **Tables 6** and **7**.

5.3. Analysis and experiences

We conducted a short survey to collect the experiences of the five participants who coded the decision cases summarised in **Table 7**. The questionnaire provided questions about the experiences and challenges of coding each of the GRADE elements. We also asked additional questions for specific cases based on the responses of each participant. The summary of the questionnaire answers is provided in this subsection.

Goals: the average Jaccard index value for this main element for all 29 cases was 0.37 which indicates several disagreements between the experts coding the cases. In particular, cases 2, 8, 9, 10, 13, 14, 15, 17, 18, 21, 23, 24, 26, 27 and 29 received zero values of the Jaccard index. The main reason for these discrepancies was that the value perspectives

Table 7
The results from coding 33 cases using GRADE. The initial four cases are marked with X1, X2, X3 and Y1. The GRADE elements marked bold text were selected by both researchers in a given case. The remaining elements (not highlighted) were selected only by one researcher. A star symbol indicates full agreement of both researchers for a given case and all elements.

Case	Goals	Roles	Assets	Decision	Environment
X1	*GV1, GV2, GV3	RT1, RT2, RF1, RF2, RL1, RF4	*AU3, A01, A04, AS4	EA1, EA2, EA5, EA4	EA1, EA2, EA4, EA5
X2	*GV3, GV2	RT1, RT2, RF2, RL1, RF1, RF4	*AU3, A04, AS4	EA1, EA4, EA5, EA2	EA1, EA4, EA5, EA2
X3	*GV2	RT1, RT1, RT2, RF1, RF2, RL1, RF3	*AU1, A03, A04, AS3, AS4, AS5	*DT4, DMFI	*DT4, DMFI
Y1	GV2, GV3, GV1, GV4, GV5	RF1, RF3, RF4, RT2, RL1, RL2, RL3	*AU3, A04, AO1, AO4, AS4, AS5	DMFI, DT1, DT4, DT5	DMFI, DT1, DT4, DT5
1	GV2, GV3, GV4	RT1-X, RF4, RT1, RF3, RL1, RL3	*AU3, A04, AU4, AU3, A01, A02, A04	DMFI, DT4, DT1, DT2, DT3, DT5	DMFI, DT4, DT1, DT2, DT3, DT5
2	GV1, GV2	RT1, RF1, RF4, RL1, RF2, RF3, RL3	AU4, A01, A01, AS1	DMFI, DT2, DT4, DMF3, DT3, DT5	DMFI, DT2, DT4, DMF3, DT3, DT5
3	*GV2, GV4	RF1, RF4, RT1-X, RT2-X, RF3, RL1, RL3	AU3, A04, A01, A02, AS1, AS4	DMFI, DT2, DT5, DT4	DMFI, DT2, DT5, DT4
4	GV2, GV3	RF1, RF4, RT1-X, RT2-X, RF3, RL1, RL3	AU1, A03, A04, AU2, AU3, A04, AS1	DMFI, DT1, DT2, DT4, EA5	DMFI, DT1, DT2, DT4, EA5
5	GV1, GV2, GV3, GV4	RF1, RF4, RT1-X, RT2-X, RF3, RL1, RL3	AU1, A03, A04, AS3	*EA1, EA2, EA4, EA5	*EA1, EA2, EA4, EA5
6	*GV2	RT1, RF1, RL1, RF4, RT2, RL2, RF2, RF4	AU1, A04, AU3, A04, AS3	DMFI, DT2, DT4, DT1, DT3, DMF3	DMFI, DT2, DT4, DT1, DT3, DMF3
7	*GV2	RT1-X, RF1, RF3, RF4, RL1, RL3, RL2	AU1, A04, AU3, A04, AS1, AS3	*DMFI, DT2, DT4	*DMFI, DT2, DT4
8	GV1, GV2, GV5	RF1, RF3, RF4, RL1, RL3, RL2	AU3, A04, A01, A02, A04, AS1, AS4	DMFI, DT2, DT3, DMF2, DT4, DT5	DMFI, DT2, DT3, DMF2, DT4, DT5
9	GV1, GV4, GV5	RT1, RT2, RT4, RF1, RF3, RL1, RL2, RL3	AU1, A03, A04, A01, A02, AS1, AU2	DMFI, DT1, DT4, DMF1, DT2, DT3	DMFI, DT1, DT4, DMF1, DT2, DT3
10	GV1, GV2, GV5	RT1, RT3, RF3, RF4, RL1, RL2, RL3	AU1, A03, A04, AS2, AS4, A01, A02, AS3	DMFI, DT4, DT2, DT1, DT3, DT4	DMFI, DT4, DT2, DT1, DT3, DT4
11	GV2, GV3, GV5	RT1, RF1, RF2, RF4, RL1, RL2, RL3, RT3	AU1, A02, A03, A04, AS3	*EA1, EA2, EA4	*EA1, EA2, EA4
12	GV4, GV3	RT1, RF1, RF3, RF4, RL1, RL2, RL3, RT3	AU1, A03, A04, AS3, A04, AS3	DMFI, DT1, DT4, DMF1, DT2, DT3	DMFI, DT1, DT4, DMF1, DT2, DT3
13	GV1, GV5	RT1, RF1, RF3, RF4, RL2, RL2, RF2	AU4, A01, A01, AS1, A02, A03, AS3	EA1, EA2, EA4, EA5	EA1, EA2, EA4, EA5
14	GV1, none	RT1, RT2, RF1, RF2, RF3, RF4, RL1, RL3	AU4, A01, AS1	EA1, EA2, EA4, EA5	EA1, EA2, EA4, EA5
15	GV1, GV3, GV4	RT1, RF3, RF4, RL1, RL3, RF1, RF2	AU2, AU4, A01, A03, AS1, AS3, AU1, A02	DMFI, DT2, DT1, DT3, DT4	DMFI, DT2, DT1, DT3, DT4
16	GV1, GV2, GV3	RT1, RF3, RF4, RL1, RL3, RL2, RT2,	AU3, A04, A01, A04, AS1, AS4, AU1, AU2,	DMF3, DT2, DT4, DMFI, DT3	DMF3, DT2, DT4, DMFI, DT3
17	GV1, none	RT1, RF3, RF4, RT1, RL1, RL3,	AU3, A04, A01, A04, AS1, AS4	*EA1, EA2, EA4	*EA1, EA2, EA4
18	GV1, GV3	RT1, RF1, RF4, RL1, RF2, RF3, RL3	AU3, A02, A04, A01, AS1	DMFI, DT1, DT2, DT4, DMF3	DMFI, DT1, DT2, DT4, DMF3
19	*GV1	RT2, RF1, RF2, RF4, RL2, RL3, RT3	AU1, A03, A04, A02, AU3, AS1, AS3	DT4, DMFI, DMF3, DT2, DT5	DT4, DMFI, DMF3, DT2, DT5
20	GV1, GV3	RT1, RL2, RF1, RF3, RF4, RL2, RT2, RL1, RF2	AU1, A04, A01, AU3, A02, A04, AS1	DT2, DT4, DMF3, DMF1, DT1, DT3, DT5	DT2, DT4, DMF3, DMF1, DT1, DT3, DT5
21	GV1, GV3	RT1, RF1, RF4, RF3, RL1, RL3	AU1, AS1, A01, AU2, AU3, AU4, A02, A03, AS3, AS4	DMFI, DT1, DT2, DT4	DMFI, DT1, DT2, DT4
22	*GV1	RT1, RF1, RF2, RF3, RL1, RL3	AU4, A01, AS1	DT3, DT4, DT1, DT2, DMF1	DT3, DT4, DT1, DT2, DMF1
23	GV1, GV3, none	RT1, RF3, RF4, RT1, RF2, RF3, RL1, RL3	AU2, AU1, AU4, A01, AS1, AS4	DT1, DT2, DT4, none	DT1, DT2, DT4, none
24	GV2, GV3	RT1, RF1, RF3, RL1, RL3, RF2	*AU4, A01, AS1	DMFI, DT2, DT4, none	DMFI, DT2, DT4, none
25	*GV1	RT1, RF1, RF4, RT3, RF2, RL1, RL3	AU4, A01, A02, AS1, AU3, AS4	EA1, EA2, EA4	EA1, EA2, EA4
26	GV1, GV2, GV4, GV5	RF1, RF4, RT2, RF3, RL1, RL2, RL3	AU4, A01, A02, AU2, AU3	EA1, EA2, EA4	EA1, EA2, EA4
27	GV2, GV3	RF1, RF4, RT1, RT2, RF2, RF3	AU4, A02, A01, AS1	DMFI, DT4, none	DMFI, DT4, none
28	*GV1	RT1, RF1, RF4, RL1, RF2, RF3, RL2, RL3	AU4, A01, AS1	*DMFI, DT2	*DMFI, DT2
29	GV1, GV3	RF4, RT1, RT2, RF1, RL1	AU1, AU4, AU3, AS4, AS5	EA1, EA2	EA1, EA2

are highly interconnected and when one perspective is explicitly mentioned, the second may be implicitly assumed by the experts, e.g., cases 2 and 8 include both customer (GV1) and financial (GV2) perspectives and these usually are connected and cases 24 and 27 include financial (GV2) and internal business (GV3) perspectives. Since the perspectives are interconnected their definitions also overlap, suggesting that more clear division aspects should be incorporated into the definitions.

What was surprising was that only 11 out of 29 cases were tagged with the internal business (GV3) value perspective (cases 1, 4, 5, 11, 12, 15, 16, 18, 21, 24, 27) providing strong indication that the perspective is not explicitly considered or documented in decisions. It might indicate that it is not relevant to the specific decisions, or that it may not be understood by the experts.

The main challenges and difficulties when coding the Goals element were:

- information about goals was often implicitly provided in the decision cases, something that resulted in the selection of 1–2 elements in most of the cases and these were described using rather high-level elements,
- in some cases, a specific goal could be related to more than one perspective of GRADE (e.g., return on investment (ROI) could be considered equally a financial (GV2), internal business (GV3), and market (GV5) related goal). The most challenging turned out to be to distinguish between decreased cost and improved ROI.

In cases 14, 17 and 23 one perspective was selected but it was not clear which one should have been selected. Therefore, there was a case where one subject did not select any. We considered the validation feedback and updated the descriptions of the goals value perspectives of GRADE in the cases that the descriptions of the elements were not clear enough to make the appropriate selection.

Roles: the average Jaccard index value for this element is 0.457 and three cases (14, 27, 29) have values under 0.2. In these cases, there was also discrepancy when coding the Goals element and therefore they may have impacted the discrepancies in coding Roles too. The discrepancies among the remaining cases were mostly equally distributed among Role Types (RT), Role Function (RF) and Role Levels (RL) perspectives. There was a relatively high number of selected elements in each Role perspective, which is related to the fact that roles were coded to capture as many decisions as possible. The combination of role levels, function and responsibility increased the number of elements that needed to be stored in the knowledge repository.

The five experts who coded the 29 cases highlighted the following challenges with regards the Roles coding:

- it was difficult to correctly identify the Role Level (this was often done based on the roles involved, e.g., executive roles would suggest a strategic decision),
- it was challenging to distinguish between strategic, tactical and operational decisions based on the provided case descriptions,
- detailed levels of information were not typically available in the collected information of the cases of decisions analysed,
- the *external consultant* or *external* roles were missing from GRADE and thus were suggested as an addition by the participants. These were added to the taxonomy.

Asset: the average Jaccard index value for Asset was 0.637. Cases 2, 19, 20, 21, 23, and 29 have values below 0.5 and only in three cases (20, 21 and 23) the values are below 0.3. In these cases, a lot of Asset main categories were selected (as many as 11 in case 21) which brings the partial explanation of the low values. The variation is present among Asset Usage (AU), Asset Origin (AO), and Asset Sub-options (AS) and none of them appear to be more prone to variation than others. The main challenges and difficulties when coding the Assets elements of GRADE were:

- lower level perspectives of the taxonomy were more difficult to be used and therefore were used only in a few cases,
- open source appeared in two places in the structure and should be perhaps eliminated,
- perspectives required improved clarifications as they were found initially unclear to the participants and,
- there was no clear link between the origin and asset management (e.g., open source is adapt or reuse, or both).

The high number of Asset elements can be attributed to the fact that assets are related to properties (criteria used typically for asset evaluation). As a result, each asset option gets multiplied by the number of properties/criteria evaluated in the decision (e.g., if cost and reliability are selected in Asset Properties (AP) and then software and system are selected in Asset Type (AT), then four entries will appear). Regarding completeness, the participants suggested that service asset types could have a clearer distinction and the Asset Properties should also include maintainability and source code quality aspects. The necessary adjustments to the taxonomy based on the feedback were carried out.

Decision: the average Jaccard index value for the Decision main element was 0.51. Four cases received full agreement (1, 8, 28, 29) but also three cases (23, 24, 27) received zero values because one of the experts did not select any decision perspectives. The discrepancies are distributed among the decision method family (DMF) and decision type (DT) criteria. The main challenges and difficulties were:

- it was difficult to identify the decision methods,
- it was challenging to encode the outcome of a decision, as the concrete result of the short-/long-term consequence,
- the experts highlighted several missing quality attributes from the list of specific criteria, e.g., Availability, Maintainability, Maintenance time/cost, Change rate, Novelty and IP, Documentation, Reusability, Access and Control, Cost (not only development cost, but e.g., cost for buying a COTS), Robustness, Code quality, Compatibility, Compliance to standards, Architecture impact, Acquisition cost, adaptation cost, change rate. These were included into the updated version of GRADE.

Environment: the average Jaccard index value for the Environment was 0.706. 10 cases received full agreement (4, 8, 9, 12, 13, 16, 17, 19, 20, 28) but also two cases (10, 23) received no agreement, because one of the experts did not select anything in the Environment element. The main challenges and difficulties when coding Environment were:

- the information about the Environment provided in the cases description was greatly incomplete to make the feasible mapping to GRADE,
- the product domain should not always be linked to the organisation and the necessary adjustments to the taxonomy based on the feedback were carried out,
- for most of the Environment elements, it was not a binary yes/no coding as it was in the other elements of GRADE, but rather providing some specific value or description (e.g., size = 40.000 employees, process = waterfall).

One expert emphasised lack of the link between the Environmental aspects and their impact on the decision. It was said that the link existed but not the semantics of the link. Another expert felt that crucial information about the product such as scalability, performance, component history, component usage, compatibility, expandability were missing and should be added to the Environment. Finally, another expert stressed that just picking an element such as maturity or system type does not tell you anything. Thus, in contrast to all other elements, they need to be further quantified, e.g., maturity = MEDIUM and system type = "EMBEDDED".

Most of the challenges identified from the evaluation related to

inadequate scenario descriptions which made the codification of the decisions look inconsistent and difficult. However, this indicates that important decision evidence is left undocumented or unstructured (i.e., textual descriptions, interviews) and thus highlights the need of a taxonomy. Other issues identified relate to unclear definitions and relations between elements of the taxonomy and even missing elements, and from the feedback we have revised GRADE to meet these issues. Another issue identified related to the granularity of the information being difficult to be found in the descriptions, something which is natural to occur, as information can be described in different ways from different people, and by structuring the information in at least some levels, as proposed in our taxonomy, may contribute in harmonising the view on which level evidence is needed to be collected in decision cases to facilitate decision-making. Some missing possibilities mentioned in the feedback regarding the taxonomy, like the outcome of the decision, the impact of the Environment to the decision and giving specific values (i.e., binary, categorical, numerical) to the elements, were intentionally not supported by GRADE. However, any tool using GRADE is planned to cover these issues.

6. Validity threats and limitations

6.1. Validity threats

Validity threats are discussed according to the guidelines proposed by Runeson et al. [48].

Construct validity concerns the relation between the studied concept and the theory behind it. The mono-source bias threat is substantial in this study, i.e., the study might be deemed to depend on the single source of information. To mitigate this threat, we have used multiple sources of information in formulating the solution, such as expert opinions, international standards, and scientific literature. Thus, the solution is formulated from a wide spectrum of theory and experiences [49]. Next, confounding constructs were identified early in this study and properly handled. All researchers that took part in this study were senior researchers, but their experiences vary. To overcome this threat, we have formed small groups (e.g., consisting of at least two researchers with the high level of experience and with one researcher with the medium level of experience) and the GRADE elements were developed within these groups with similar on average experience. Similarly, we have increased the generalisability across constructs by assigning reviewers, to evaluate the work produced by the groups, i.e., at least two researchers have reviewed the GRADE elements and categories. Finally, we involved five researchers in the validation of GRADE based on empirical evidence from 33 industrial cases.

Conclusion validity threats were minimised as the researchers worked in small groups, described and categorised the GRADE elements using literature, theories, and standards in order to indicate specification and generalisation of the studied relationships. Such collaborative environment reduces the error rate fishing risk based on subjective opinions. In addition, the collaborative work also adds to the reliability of the findings. On the other hand, a minor threat related to the reliability of the development of GRADE exists, as the implementation within the different groups could be different, if different groups of researchers were used. However, the development of the GRADE taxonomy is flexible, scalable and expandable, and if this is required, new entries can be easily entered or modified by its users. The validation with 33 industrial cases provided new ideas about changes and new elements that should be added and these were incorporated into GRADE. Another threat related to conclusion validity is the fact that the researchers interpreted the source documents when coding the 33 industrial cases. This creates a risk of bias for the evaluation that could be mitigated by involving practitioners instead of researchers. Albeit the five researchers involved in the GRADE validation have industrial experience, their understanding of the 33 cases is limited to studying the documentation than direct involvement. This could explain the

difficulties in coding the cases, e.g., low Jaccard index values when coding the Roles element could be the result of lack of organisational knowledge and direct involvement in the coded cases. Similarly, without direct involvement it can be hard to code the selected decision method and the reason for selecting it. Therefore, it remains future work to apply GRADE by practitioners who are directly involved in the decisions.

Internal validity threats are the potential confounding factors that can affect without the researchers' knowledge. Since this study is not an experiment, internal validity threats have limited scope and influence. Still, for each step in the research process we followed a strategy that reduces the potential confounding factors and we have ensured that these threats were not shifted to the next phase. Concerning the problem identification step, the first threat is related to not having an actual problem to solve or a problem that is worthy to investigate. Considering the researchers experience in tackling industry relevant problems, as the one tackled by GRADE in different industries, we assume that this threat is mitigated. Also, the preliminary review of literature conducted by the researchers has highlighted the lack of evidence in documenting strategies supporting decision-makers in the selection of architectural assets and has provided a baseline of problem description. These have both ruled out any additional threats related to problem formulation. In the solution formulation step, we created a structured brainstorming workshop among the researchers that took part in this study. Concerning the GRADE evaluation, we initiated it appropriately with generic and common scenarios that included most of the major elements considered in decision-making. Hence, following such structured research process we have mitigated related internal threats such as: mortality [49] (all senior researchers worked collaboratively together), selection [49] (experienced researchers took part and international standards and literature have been considered) and instrumentation [49] (data collection forms were used in this study – i.e., Google spreadsheets which were well designed and peer-reviewed to help make the categorisation of the elements possible).

External validity threats are conditions limiting generalisability. In this study, we have included different sources of information and we have followed a set of well-established guidelines for creating taxonomies by Bayona-Ore et al. [31]. We have also instantiated the taxonomy with different scenarios to test representativeness and generalisability. However, even after coding 33 industrial decisions with the help of GRADE, our confidence in saying how far it can go and satisfy the industrial setting of a real on-going decision-making process needs to be put into test in future work.

6.2. Limitations

In this section, the most important limitations of the GRADE taxonomy are discussed.

Creating a taxonomy is a complex task that requires to analyse large amount of data to identify key aspects that are essential to unambiguously summarise the subject of the classification. With the identification of relevant aspects comes problems such as deciding on the nomenclature (i.e., different persons often associate different meaning to a term), categorising the terms (i.e., terms could make sense in different categories), etc. One limitation thus introduced with the nature of taxonomies is the level of ambiguity associated with the elements used in the classification, as well as the definitions attached to each element.

The GRADE taxonomy has gone through several validity checks from several experts who have performed several iterations of validation and re-design of the taxonomy. We believe that its current design, especially concerning the top-level elements of GRADE (i.e., Goals, Roles, Assets, Decision, and Environment) includes the appropriate constructs to describe the main elements in any decision taken in the development of software-intensive systems. Also, through the specification of the sub-elements, called perspectives, a common and less

ambiguous vocabulary can be shared among the decision makers, which could enable fruitful discussions around decisions and could potentially increase the confidence that decision evidence is sufficiently documented and distributed to projects.

Another limitation of the taxonomy is that it can never be complete. We do not expect GRADE to be complete, as this will mean that new decision methods and criteria cannot be discovered and this would significantly compromise its flexibility and applicability. Also, it would require that all the product properties and quality aspects will need to be covered by the taxonomy, which is impossible. However, we do believe that the core concepts are coherent and orthogonal so that any new additions should fall under one of them. The GRADE taxonomy is also flexible and adaptable to cope with these changes. In particular, many leaves of the taxonomy (e.g., the properties of an asset) can either be used as proposed or can be replaced according to the needs. For example, in a given company, the categories corresponding to the properties of an asset could be replaced to match the list of properties used in the company, together with their definition. The same would apply to the list of roles or the list of methods among others.

With the current version of GRADE, it remains difficult to discover dependencies between the elements used to describe a decision. A similar impression was provided from the five subjects who coded the 29 cases with the help of GRADE as several inconsistencies between the selected codes are related to the dependencies between the GRADE elements. These dependencies may either involve relationships between elements or time-related considerations (i.e., a decision may consist of several sub-decisions and a decision might need to be revisited in some later point in time). One of the main reason why we adopted the graph database for the knowledge repository for GRADE is to be able to present and identify these dependencies and some examples are provided in Fig. 4. It remains a part of future work to thoroughly analyse dependencies between the collected cases and the potential benefits that dependency analysis can bring for these type of decision-making.

For now, visualising the relations between the top-level elements, and the perspectives to describe a decision is not feasible and therefore other methods will need to be investigated to support this vital aspect of documentation. Analysing decisions after their documentation is another limitation which will need to be addressed in some other way, i.e., we expect that tools using the GRADE taxonomy will need to consider the timing aspect (e.g., starting and ending time of the decision-making process, decision evolution, decision date, etc.).

The GRADE taxonomy has been specifically designed to document a decision, support retrospective learning and improve comprehension of decision outcomes. However, in its current form there is no way to document why a specific decision method was selected in a decision scenario. This limitation can be partially addressed when the GRADE decision canvas is used to document a decision. The canvas includes questions related to understanding why a specific decision method was selected [50].

Another significant limitation of this study is associated with the fact that researchers coded 33 industrial cases without direct involvement in these cases. This could explain why the five researchers experienced difficulties when using GRADE and why the Jaccard index values are low in some cases. For example, the difficulties in selecting between customer (GV1) and internal business perspectives (GV3) on the Goals element or precisely coding the Role Levels (RL) and Role Functions (RF) could have been mitigated if the participants have had more organisational knowledge or direct involvement in these cases. Similarly, without direct involvement, it can be hard to code the selected decision method and the reason for selecting it. At the same time, the validation results bring supporting evidence that the two top levels of GRADE are stable, necessary for documenting decisions and described in a clear way. Selecting elements on lower levels or GRADE will remain challenging due to specific nature of particular cases and subtle differences in details. Therefore, in future work, we plan to further explore the stability of lower level GRADE elements and

facilitate their adaptation to different organisational settings and documentation styles. We also plan to ask practitioners to use GRADE for coding decisions that they are or were directly involved in. This should help to validate if the GRADE usage difficulties expressed by the five researchers remain significant for practitioners.

7. Conclusion

Decision-making intensifies software-intensive system development and pervades it on numerous abstraction levels and during multiple development phases. The presence of various sourcing strategies for software components (e.g., COTS, OSS, in-house, outsourced), further fuels the intensity and complexity of decision-making scenarios [11]. The result is that more and more software-intensive companies drift away from in-house development and leverage on other mechanisms by incorporating externally sourced components. At the same time, these companies are left without systematic means to analyse, document and store decisions and associated important attributes.

This paper occupies this important yet significantly unexplored research gap by developing the GRADE taxonomy that serves as a common vocabulary to describe and classify decision-making concerning architectural assets. GRADE also provides a structured instrument for collecting decision-making evidence that can be realised in knowledge repositories. GRADE was systematically developed using the state-of-the-art recommendations and guidelines for taxonomy development [31].

Taxonomies are means to structure, advance and communicate knowledge, as well as offer a common base for research in a field [30]. The GRADE taxonomy is focused solely on selecting various architectural assets (as described in Section 4). The two-phase validation involving 33 industrial cases brings supporting evidence that the taxonomy fulfills the goals it was designed for. Moreover, we report moderate evidence that GRADE helps to sufficiently and consistently document the components of decision scenarios and suggest decision-making improvement when decision cases are documented in a knowledge repository structured with the help of GRADE. Moreover, the experts involved in GRADE validation confirmed that above, and that it captures important decision characteristics and its understandability is relatively high.

It is important to notice that GRADE, as any taxonomy, is under constant evolution, and the construction of the taxonomy itself has been subject to versions of refinement following the collection of empirical data from both peer-reviewed sources and industry cases. From a researchers perspective, GRADE can be used as a starting point for research into the field, enabling both finding current and characterising new research. Practitioners can use GRADE as a guide or even checklist when designing or selecting a decision process for a specific scenario. GRADE does not prescribe any one decision model or method, rather should be seen as a detailing of this type of decision scenario, its constituents with examples and links to related work in each part.

As future work we plan to apply the taxonomy in industrial settings and evaluate its usefulness in real on-going decision-making activities. We also plan to further fuel the development of the knowledge repository to document decision cases and the COACH decision support system. Our ambition is to provide as complete picture of decision-making as possible and ensure that GRADE's high-level elements remain stable and consistent and that GRADE offers flexibility on lower levels.

Visualising the decision cases with the help of a knowledge repository provided valuable initial steps towards dependency discovery and inconsistencies identification. We plan to further expand dependency analysis with the help of visualisations and the knowledge repository. Moreover, we plan to integrate GRADE with the decision canvas to offer greater ability to capture the rationale for decisions. Finally, we plan to involve practitioners in using GRADE and coding the cases that they are directly involved in.

Acknowledgements

The authors wish to thank for their input all the researchers, experts and the rest of the researchers working in the ORION project. This work was supported in part by a grant for the ORION project from The Knowledge Foundation reference number (20140218) in Sweden.

References

- [1] B. Regnell, R. B. Svensson, K. Wnuk, Can We Beat the Complexity of Very Large-Scale Requirements Engineering?, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 123–128. 10.1007/978-3-540-69062-711.
- [2] M. Khurum, T. Gorschek, M. Wilson, The software value map — an exhaustive collection of value aspects for the development of software intensive products, *J. Softw.* 25 (7) (2013) 711–741.
- [3] A. Aurum, C. Wohlin, The fundamental nature of requirements engineering activities as a decision-making process, *Inf. Softw. Technol.* 45 (14) (2003) 945–954. Eighth International Workshop on Requirements Engineering: Foundation for Software Quality. 10.1016/S0950-5849(03)00096-X .
- [4] V.S. Rekhav, H. Muccini, A study on group decision-making in software architecture, 2014 IEEE/IFIP Conference on Software Architecture, (2014), pp. 185–194, <http://dx.doi.org/10.1109/WICSA.2014.15>.
- [5] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, M.A. Babar, 10 years of software architecture knowledge management: Practice and future, *J. Syst. Softw.* 116 (2016) 191–205, <http://dx.doi.org/10.1016/j.jss.2015.08.054>.
- [6] D. Tofan, M. Galster, I. Lytra, P. Avgeriou, U. Zdun, M.-A. Fouche, R. de Boer, F. Solms, Empirical evaluation of a process to increase consensus in group architectural decision making, *Inf. Softw. Technol.* 72 (C) (2016) 31–47, <http://dx.doi.org/10.1016/j.infsof.2015.12.002>.
- [7] M. Nowak, C. Pautasso, Team situational awareness and architectural decision making with the software architecture warehouse, Proceedings of the 7th European Conference on Software Architecture, ECSA'13, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 146–161, <http://dx.doi.org/10.1007/978-3-642-39031-913>.
- [8] R.C. De Boer, R. Farenhorst, P. Lago, H. Van Vliet, V. Clerc, A. Jansen, Architectural knowledge: Getting to the core, International Conference on the Quality of Software Architectures, Springer, 2007, pp. 197–214.
- [9] P. Krutchen, An Ontology of Architectural Design Decisions in Software Intensive Systems, 2nd Groningen Workshop Software Variability, (2004), pp. 54–61.
- [10] A. Aurum, R. Jeffery, C. Wohlin, M. Handzic, Managing Software Engineering Knowledge, Springer Science & Business Media, 2013.
- [11] D. Badampudi, C. Wohlin, K. Petersen, Software component decision-making: in-house, OSS, COTS or outsourcing — a systematic literature review, *J. Syst. Softw.* 121 (2016) 105–124.
- [12] E. Papatheocharous, K. Petersen, A. Cicchetti, S. Sentilles, S.M.A. Shah, T. Gorschek, Decision support for choosing architectural assets in the development of software-intensive systems: The GRADE taxonomy, Proceedings of the 2015 European Conference on Software Architecture Workshops, ACM, 2015, p. 48.
- [13] P. Kruchten, R. Capilla, J.C. Dueñas, The decision view's role in software architecture practice, *IEEE Softw.* 26 (2) (2009) 36–42, <http://dx.doi.org/10.1109/MS.2009.52>.
- [14] P. Kruchten, P. Lago, H. Van Vliet, Building up and reasoning about architectural knowledge, International Conference on the Quality of Software Architectures, Springer, 2006, pp. 43–58.
- [15] Z. Li, P. Liang, P. Avgeriou, Application of knowledge-based approaches in software architecture: a systematic mapping study, *Inf. Softw. Technol.* 55 (5) (2013) 777–794.
- [16] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, M.A. Babar, A comparative study of architecture knowledge management tools, *J. Syst. Softw.* 83 (3) (2010) 352–370.
- [17] W. Ding, P. Liang, A. Tang, H. Van Vliet, Knowledge-based approaches in software documentation: a systematic literature review, *Inf. Softw. Technol.* 56 (6) (2014) 545–567.
- [18] R. France, J. Bieman, B.H. Cheng, Repository for model driven development (remodd), International Conference on Model Driven Engineering Languages and Systems, Springer, 2006, pp. 311–317.
- [19] C. Mayr, U. Zdun, S. Dustdar, Reusable architectural decision model for model and metadata repositories, International Symposium on Formal Methods for Components and Objects, Springer, 2008, pp. 1–20.
- [20] N. Milanovic, R. Kutsche, T. Baum, M. Cartsburg, H. Elmasgunes, M. Pohl, J. Widiker, Model&metamodel, metadata and document repository for software and data integration, International Conference on Model Driven Engineering Languages and Systems, Springer, 2008, pp. 416–430.
- [21] R. Capilla, O. Zimmermann, U. Zdun, P. Avgeriou, J.M. Küster, An enhanced architectural knowledge metamodel linking architectural design decisions to other artifacts in the software engineering lifecycle, European Conference on Software Architecture, Springer, 2011, pp. 303–318.
- [22] U. van Heesch, P. Avgeriou, R. Hilliard, A documentation framework for architecture decisions, *J. Syst. Softw.* 85 (4) (2012) 795–820.
- [23] ISO/IEC/IEEE 42010:2011, Systems and software engineering architecture description, the latest edition of the original IEEE std 1471:2000, recommended practice for architectural description of software-intensive systems. (2011).
- [24] A. Jansen, J. Bosch, P. Avgeriou, Documenting after the fact: recovering architectural design decisions, *J. Syst. Softw.* 81 (4) (2008) 536–557.
- [25] A. Sawada, M. Noro, H.-M. Chang, Y. Hachisu, A. Yoshida, A design map for recording precise architecture decisions, Software Engineering Conference (APSEC), 2011 18th Asia Pacific, IEEE, 2011, pp. 298–305.
- [26] I.C. Society, Swebok guide v3.0, 2009, (<https://www.computer.org/web/swebok/v3>) Accessed: 2017-03-30.
- [27] R.L. Glass, I. Vessey, V. Ramesh, Resres: the story behind the paper research in software engineering: an analysis of the literature, *Inf. Softw. Technol.* 51 (1) (2009) 68–70.
- [28] B.I. Blum, A taxonomy of software development methods, *Commun. ACM* 37 (11) (1994) 82–94.
- [29] D. Šmite, C. Wohlin, Z. Galvina, R. Prikladnicki, An empirically based terminology and taxonomy for global software engineering, *Emp. Softw. Eng.* 19 (1) (2014) 105–153.
- [30] M. Unterkalmsteiner, R. Feldt, T. Gorschek, A taxonomy for requirements engineering and software test alignment, *ACM Trans. Soft. Eng. Methodol.* 23 (2) (2014) 16.
- [31] S. Bayona-Oré, J.A. Calvo-Manzano, G. Cuevas, T. San-Feliu, Critical success factors taxonomy for software process deployment, *Soft. Qual. J.* 22 (1) (2014) 21–48.
- [32] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, *J. Manage. Inf. Syst.* 24 (3) (2007) 45–77.
- [33] R. Larsson, Case survey methodology: quantitative analysis of patterns across case studies, *Acad. Manage. J.* 36 (6) (1993) 1515–1546.
- [34] ISO/IEC JTC1/SC7 N4098CD 25010.2, Software engineering—Software product quality requirements and evaluation (SQuare) quality model, WG23, 25010, N3803, N4008, N4097, (2008).
- [35] M. Usman, R. Britto, J. BÄ¶rlster, E. Mendes, Taxonomies in software engineering: a systematic mapping study and a revised taxonomy development method, *Inf. Softw. Technol.* 85 (2017) 43–59, <http://dx.doi.org/10.1016/j.infsof.2017.01.006>.
- [36] T. Herrmann, I. Jahnke, K.-U. Loser, The role concept as a basis for designing community systems, COOP, (2004), pp. 163–178.
- [37] J.K. Tan, S.B. Sheps, Health decision support systems, Jones & Bartlett Learning, 1998.
- [38] C. Morisset, I. Yevseyeva, T. Groß, A. van Moorsel, A formal model for soft enforcement: influencing the decision-maker, Security and Trust Management, Springer, 2014, pp. 113–128.
- [39] K. Petersen, C. Wohlin, Context in industrial software engineering research, Proceedings of the Third International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, October 15–16, 2009, Lake Buena Vista, Florida, USA, (2009), pp. 401–404.
- [40] E. Vailati, A very fast intro to decision theory, 2017, (<http://www.siu.edu/~eavailat/decision.htm>) Accessed: 2017-03-30.
- [41] J.R. Quinlan, Induction of decision trees, *Mach. Learn.* 1 (1) (1986) 81–106.
- [42] A. Cicchetti, M. Borg, S. Sentilles, K. Wnuk, J. Carlson, E. Papatheocharous, Towards software assets origin selection supported by a knowledge repository, 2016 1st International Workshop on Decision Making in Software ARCHitecture (MARCh), (2016), pp. 22–29, <http://dx.doi.org/10.1109/MARCh.2016.11>.
- [43] F. van der Linden, J. Bosch, E. Kamsties, K. Känsälä, H. Obbink, Software product family evaluation, Proc. of the 3rd Software Product Line Conference, (2004), pp. 110–129.
- [44] ISO/IEC 27002:2013, Information technology-security techniques-code of practice for information security management, (2013).
- [45] A. Trendowicz, R. Jeffery, Software project effort estimation, Foundations and Best Practice Guidelines for Success, Constructive Cost Model—COCOMO pages (2014) 277–293.
- [46] R. Angles, A comparison of current graph database models, Proceedings of the IEEE 28th International Conference on Data Engineering Workshops (ICDEW), (2012), pp. 171–177, <http://dx.doi.org/10.1109/ICDEW.2012.31>.
- [47] W. McCormick, N. Lyons, K. Hutcheson, Distributional properties of Jaccards index of similarity, *Commun. Stat.* 21 (1) (1992) 51–68.
- [48] P. Runeson, M. Host, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, 1st, Wiley Publishing, 2012.
- [49] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Sci. Bus. Media, 2012.
- [50] E. Papatheocharous, K. Petersen, A. Cicchetti, S. Sentilles, S.M.A. Shah, T. Gorschek, Decision support for choosing architectural assets in the development of software-intensive systems: The grade taxonomy, Proceedings of the 2015 European Conference on Software Architecture Workshops, ACM, 2015, p. 48.