

Decision support for choosing architectural assets in the development of software-intensive systems: The GRADE taxonomy

Efi Papatheocharous

SICS Swedish ICT AB

Kista, Stockholm, Sweden

+46 705 28 63 76

efi.papatheocharous@sics.se

Kai Petersen

Blekinge Institute of Technology

Karlskrona, Sweden

+46 455 38 58 77

kai.petersen@bth.se

Antonio Cicchetti

Mälardalen University

Västerås, Sweden

+46 739 60 72 15

antonio.cicchetti@mdh.se

Séverine Sentilles

Mälardalen University

Västerås, Sweden

+46 21 10 70 38

severine.sentilles@mdh.se

Syed Muhammad Ali Shah

SICS Swedish ICT AB

Kista, Stockholm, Sweden

+46 725 03 94 72

syed.shah@sics.se

Tony Gorscak

Blekinge Institute of Technology

Karlskrona, Sweden

+46 455 38 58 17

tony.gorscak@bth.se

ABSTRACT

Engineering software-intensive systems is a complex process that typically involves making many critical decisions. A continuous challenge during system design, analysis and development is deciding on the reference architecture that could reduce risks and deliver the expected functionality and quality of a product or a service to its users. The lack of evidence in documenting strategies supporting decision-making in the selection of architectural assets in systems and software engineering creates an impediment in learning, improving and also reducing the risks involved. In order to fill this gap, ten experienced researchers in the field of decision support for the selection of architectural assets in engineering software-intensive systems conducted a workshop to reduce traceability of strategies and define a dedicated taxonomy. The result was the GRADE taxonomy, whose key elements can be used to support decision-making as exemplified through a real case instantiation for validation purposes. The overall aim is to support future work of researchers and practitioners in decision-making in the context of architectural assets in the development of software-intensive systems. The taxonomy may be used in three ways: (i) identify new opportunities in structuring decisions; (ii) support the review of alternatives and enable informed decisions; and (iii) evaluate decisions by describing in a retrospective fashion decisions, factors impacting the decision and the outcome.

Categories and Subject Descriptors

D.2.2 [Design Tools and Techniques]: Taxonomy

General Terms

Documentation, Design.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '15, September 8, 2015, Dubrovnik, Croatia.
Copyright 2015 ACM 1-58113-000-0/00/0010 ...\$15.00.

DOI: 10.1145/2797433.2797483

Keywords

Software-intensive Systems, Strategic Decision Support, Software Architectures, Taxonomy.

1. INTRODUCTION

Modern industrial systems are typically complex and software-intensive. Software-intensive systems refer to any product or service where software in an essential way contributes to the design, construction, deployment and evolution of a system [1]. Systems refer to man-made systems, including software products and services and software-intensive systems [1]. Making sure that efficient and cost-effective ways of development are followed is a vital issue for organisational longevity and typically involves critical decision-making in many points along the engineering process. It frequently faces the selection between more than one development options and integrating a plethora of architectural assets. An architectural asset is a valuable vehicle for capitalising on work previously done by providing well-defined reusable entities, from fine-grained programming idioms to large grained off-the-shelf packaged solutions [2]. In this work, we regard an architectural asset as any option for a decision-maker that contains or is related to software. These options can be of any level of granularity, such as a specification document, a software component, a system, a system-of-system or a part of a system.

Constructing software-intensive systems from reusable architectural assets is a form of development which is not newly discovered [3]. There are many benefits of automating and optimising architectural assets' mass production, but these benefits are not yet exploited fully by today's industries of software-intensive systems, as many issues related to the technical aspects of the development have not been entirely solved. For instance, continuous integration of architectural assets, which could reduce significantly time-to-market, is not simple, mainly due to the complexity and uniqueness of the software-intensive systems. Most products are synthesised from a plethora of components into complex interlaced systems and dealing with integration along the whole product lifecycle is complicated. Organisational issues are often impediments to improving

decision-making during the development. An illustrative example of organisational complexity and challenges faced in a global organisation are described by Crnkovic et al. [4] where the introduction of a reuse approach into the organisations' development process required considerable changes, that took a period of almost 10 years to be implemented successfully (by using their own technology "KOALA") [5].

The aim of this work is towards supporting decision-making for architectural assets in the development of software-intensive systems. Decision support in our context is any combination of means to support the decision-making process, where "means" could be interpreted as any process, method, model, framework, tool etc. that may support decision-makers. We recognise that the above challenge has been sporadically explored until today and will continue to be a wicked problem, as long as no documentation, traceability and support exists on how to structure decisions, review available alternatives, support informed decisions, and evaluate decisions. Collecting evidence in a structured way can be considered a starting point in enabling retrospective learning from past architectural decisions and their outcomes. In particular, it may reduce the severity of the technical issues emerging during the construction process, and other risks related to the decision. It could also enable transparent traceable holistic view of the decisions made and identify new opportunities for better decision support.

The lack of evidence, traceability and support in documenting strategies supporting decision-makers in the selection of architectural assets in software-intensive systems drives the motivation for conducting the work described in this paper. To fill-in this gap we have conducted a workshop to propose the definition of a taxonomy, called GRADE that contains the key elements of decision making. Although we expect in the future the taxonomy to evolve, it represents a starting point to better understand and consciously document the concrete factors involved in decisions made in the development of software-intensive systems.

The contribution of this paper is a taxonomy that may be used in three ways: (i) identify new opportunities in structuring decisions; (ii) support the review of alternatives and enable informed decisions; and (iii) evaluate decisions by describing in a retrospective fashion decisions, factors impacting the decision and the outcome.

The rest of this paper is structured as follows: Section 2 briefly associates our work to previous research, Section 3 describes the research process followed. Section 4 presents the GRADE taxonomy and its use is demonstrated through one case, where the taxonomy is instantiated and validated. Section 4 summarises the contribution of the paper, describes the main conclusions, and future research steps.

2. RELATED WORK

A brief summary of related work on the documentation of decisions in designing software-intensive systems and utilising the notion of taxonomies are presented in this section.

In [6] a documentation framework for architectural decisions is presented using the conventions of ISO/IEC/IEEE 42010 [1] consolidating different views. The four viewpoints, Decision Detail, Decision Relationship, Decision Chronology and Decision Stakeholder Involvement, satisfy several stakeholder concerns related to architecture decision management. In [7] the ADDRA

approach is presented, that architects can use for recovering architectural decisions made retrospectively. Very few works combine explicit description of design decisions together with architectural design [8]. In [8] a design map for recoding architectural decisions and a meta-model focusing on the relationships between non-functional properties and architectural styles are described.

To support documentation other works in developing taxonomies exist, they however do not explicitly serve the architectural decision documentation literature. The work is limited in only structuring knowledge areas in software engineering and using the notion of taxonomies. Examples include the Guide to the Software Engineering Body of Knowledge (SWEBOK) [9], which describes the software engineering discipline in a structured way, Glass et al. [10] describes a taxonomy on software engineering research, Blum [11] describes development methods, Smite et al. [12] describe a taxonomy for global software engineering and Unterkalmsteiner et al. [13] describe a taxonomy associating software requirements engineering and testing.

Based on the previous work and concerns raised regarding how to effectively document architectural decisions, we have taken into account the following: (i) Efforts in building taxonomies should be driven by clearly defined goals (as recommended in [9]). (ii) A systematic process needs to be followed in the taxonomy construction. (iii) Taxonomies can be built bottom-up in cases where relationships are not well understood (according to [13]). (iv) Experts can be involved in the process of taxonomy construction (as in [12]). Finally, (v) Taxonomies can be validated against their purpose, either through classification based on the literature [12], or through industrial case studies [13]. The strategy used for validation highly depends on the purpose of the taxonomy.

3. RESEARCH PROCESS

In our research we adopted the steps described in [14] to work in a structured and iterative manner, taking into account grounded theories for analysis. For example a number of standards and taxonomies were used to derive the analysis of categories and sub-categories for the different elements of the proposed solution. The steps included: (a) identify problem and motivate, (b) define objectives of a solution using brainstorming and theories, (c) design/develop and generate new ideas, (d) describe and demonstrate, (e) evaluate and review, which includes selecting and describing a scenario, instantiate the solution and then iteratively work to improve the suggested solution by performing the sequence of steps (b)-(e).

The work was performed between January-May 2015. It involved a targeted set of workshop of a focus group that provided their perceptions, opinions and beliefs towards concrete factors involved in decisions made in the development of software-intensive systems and contributions of ten international experienced researchers in software engineering were documented. The work was coordinated by one of the first authors of this paper. The participants first discussed the lack of documentation of strategies supporting decision-making in the selection of architectural assets in software-intensive systems engineering. The need for improvements in terminology and definitions for the elements of the decision making process was identified. The participants worked collaboratively to define a language for the key elements of a taxonomy that would help making informed decisions in the selection of architectural assets using scenarios from their personal experiences. Multiple views

on the elements were discussed openly, and in smaller groups, to reach consensus. The researchers also worked in small groups, described and categorised the elements using theories and standards in order to indicate specification and generalisation relationships. The final artefact of the process was a taxonomy (described in the following section), called GRADE, which identifies the key elements in architectural decision-making process). An evaluation stage followed for the definitions of the elements in the taxonomy: individuals' and smaller groups' views were reviewed by members of the working group that did not participate in the activity conducted in the smaller groups in the previous stage. The GRADE key concepts were lastly validated with a case to instantiate decision elements in a specific scenario.

4. THE GRADE TAXONOMY

The GRADE taxonomy is made up of the following elements: i) Goals, ii) Roles, iii) Assets, iv) Decision methods and criteria, and v) Environment, as explained in this subsection and illustrated in Figure 1. They are an abstraction of basic decision theory elements [15] and considered typically in decision structures like induction tasks of decision trees [16]. These elements refer to: Acts (the options considered by the decision-maker - in our case the architectural 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 validation stage of GRADE and is reflected in the achievement of the overall goal of the decision).

Goals: Constitute an important type of architectural knowledge. Goals refer to a set of useful and valuable targets that stakeholders identify when making an architectural decision. Goals can deliver value from different perspectives, such as to the customer (in terms of functionality and other non-functional qualities), financial, innovation and learning, benefits to the internal business, market and external production. This key element is similar to Payoffs based on the description given above.

Roles: Characterises decision-makers based on different attributes and clarifies their contribution in relation to the decision made. The role is defined based on the lifecycle-perspective of the product (i.e., where within the lifecycle of the product does the decision-maker stand), decision level perspective (i.e., what kind of decisions can the specific role take), functional role (i.e., what is the functional role of the decision-maker in a given organisation), and function of the role (i.e., what is the decision-making function or contribution of the decision-maker).

Assets: Represents artefacts developed or obtained at any level of the engineering process containing software. They are characterised by three attributes, namely: origin (i.e., where assets come from and how are they developed), type (i.e., what kind of assets they are), attributes (i.e., what are the assets' most important functional and non-functional properties). This key element is similar to Acts based on the description given above.

Decision methods and criteria: Documents the methods used to make a decision with the set of criteria used to establish this decision. This key element is similar to Events based on the description given above.

Environment: Includes contextual criteria, facts or occurrences that take place in most cases outside the control of the decision-maker and are more external in relation to the rest of decision criteria described above.

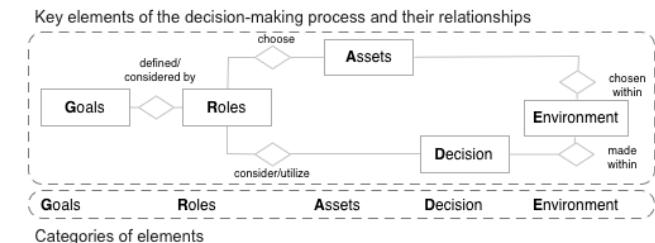


Figure 1. Key elements of GRADE taxonomy.

4.1 Categories of GRADE Key Elements

The GRADE elements have been analysed into categories explained in this section. Figure 2 presents an overview of the elements of GRADE.

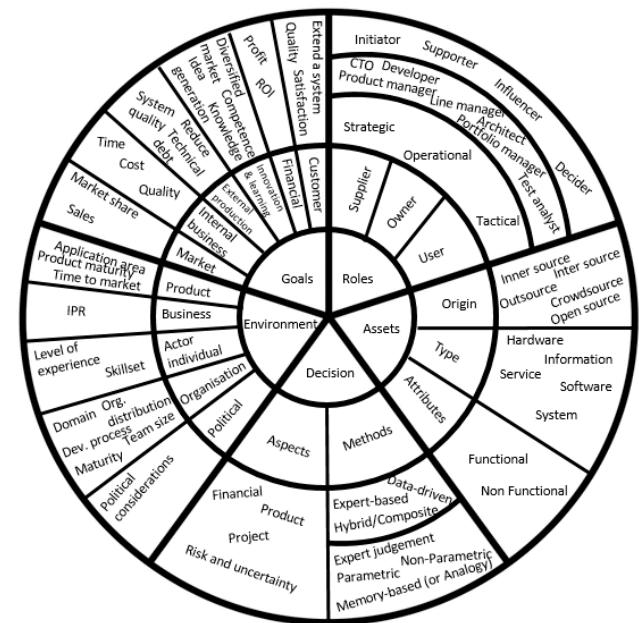


Figure 2. Overview of GRADE elements.

The detailed categories of the GRADE taxonomy are provided in an online mind map¹ due to its large size and to enable scalability. The rest of this subsection provides the description of the main categories.

4.1.1 Goals

Goals are categorised according to the value perspectives identified for software-intensive product development [17]. These value perspectives provide categorisations for the individual goals a decision-maker might wish to achieve. They are particularly suitable for classification as they are based on an exhaustive assessment of literature- and industry-focused research. Six main

¹ <https://app.wisemapping.com/c/maps/306265/public>

categories are represented, namely the customer-perspective (perceived value provided to the customer with the software product), the financial-perspective (monetary or financial benefits such as shareholder value), internal business value (increasing efficiency and quality of internal work products), innovation and learning value (building intellectual capital in terms of human knowledge) and market-perspective (ability to create brand identity). In the initial version of the taxonomy we have identified 34 goals. The main categories of goal element and some examples are shown in Figure 3.

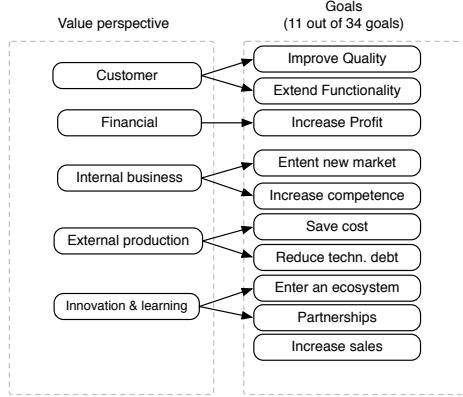


Figure 3. Main categories of Goals element of GRADE.

4.1.2 Roles

The roles of the individuals involved in the decision-making process can be viewed from different perspectives. These perspectives have been identified as lifecycle, decision-level perspective, decision-maker role and decision-making function. The lifecycle perspective determines whether decision-makers were involved from the supplier (provider of the asset), owner (user or organisation owning the asset), or user (consuming the asset) side. The decision-level perspective determines whether the roles execute strategic (long-term, complex, non-routine), tactical (strategic implementation, medium-term), or operational (running the day-to-day business, more routine-like) decision-making. Roles are also defined on the perspective of their position in an organised sum, as described in [9] (e.g. software requirements engineer, software designer, software tester, etc.). Finally, the decision-maker function determines the purpose of the role, i.e., whether the person was the initiator, supported the decision in some way, influenced the decision, or was the person finalising the decision. The main categories of Roles are shown in Figure 4.

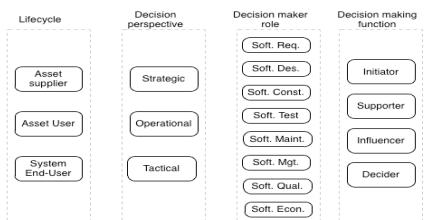


Figure 4. Main categories of Roles element of GRADE.

4.1.3 Assets

Assets are characterised based on their origin, type, and attributes. The origin of an Asset is defined according to its base use (i.e., option to reuse an asset, buy, develop or adapt) and detailed option of development. The combination of these two aspects of

Asset origin define where the asset originates from. For example two assets with origins buy and develop would mean the decision-maker is confronted with a make-or-buy decision. On the detailed option of development option the taxonomy defines the following origin options inner sourced (development is carried out in-house), outsourced (a supplier was contracted to develop the asset for an owner), crowdsourced (distributed development of the asset by a crowd of networked people), open sourced (source code of the asset originates from open source resources), inter sourced (development is happening from a set of partners connected in an ecosystem).

To facilitate the characterisation of asset further, we used the categories proposed in [18] to define the type of an asset. In that work, the type of an asset is distinguished as information assets (any document that describes/specifies an asset), software assets (any asset that is a modular piece of code, from a small functionality to application software, e.g. library, functional components, user interface), system assets (any asset which functionality is realised by a combination of software and hardware, e.g. navigation system), hardware assets (any asset that is a physical device containing some kind of software, e.g. server, sensor, actuator), and service asset (web-based service providing functionality through an interface over a network).

The attributes of an asset, i.e. its functional and non-functional properties, are according to the ISO 25010 standard [19]. An additional category has been added to cover economical aspects that need to be considered for the assets, such as price, level of support, and different types of costs (cost to learn, integrate, use, etc.). Assets main elements are illustrated in Figure 5.

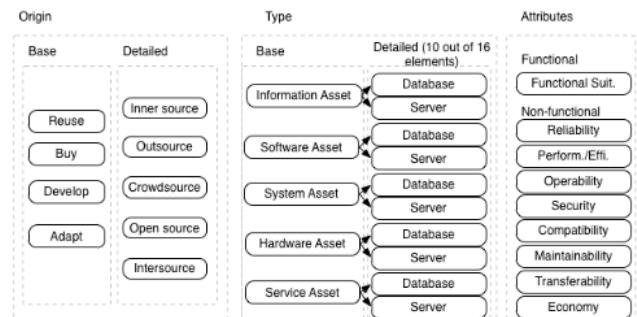


Figure 5. Main categories of Assets element of GRADE.

4.1.4 Decision methods and criteria

For supporting decisions two types of elements are used in GRADE, the decision methods and the decision criteria, as illustrated in Figure 6. The decision criteria are categorised in financial, product, project, risk/uncertainty and business, and in the first version of the taxonomy in total 54 elements were identified. The decision criteria are considered and used as input to the decision methods. Decision methods are categorised, using [20] in three types based on the type of support needed, i.e., Expert-based, Data-driven and Hybrid/Composite. Expert-based methods rely on the expert opinions or judgments of one or more (a group) of expert. Data-driven methods depend on large amounts of data and they are distinguished in memory-based (e.g., Analogy), Parametric (e.g., Regressions), Non-parametric (e.g., Machine Learning), or Semi-parametric (combinations of the two last types, such as Regression and Artificial Neural Networks). Finally, Hybrid or Composite methods emerge when any of the

above methods are combined, cross-cutting the main types (e.g., Expert judgement and a Parametric method are combined). A typical instance of method is the analogous method of estimating a property of an asset, and is frequently used to estimate the value when there is a limited amount of detailed information about the project (e.g., in the early phases). Analogous estimating uses expert judgment. For decision methods a total of 52 methods were identified in the first version of the taxonomy.

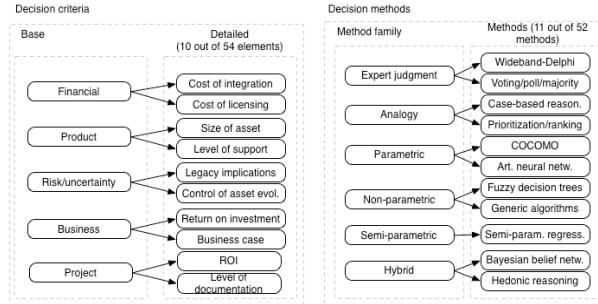


Figure 6. Main categories of Decision methods GRADE.

4.1.5 Environment

The environment characterises the context in which the asset is chosen and relates to criteria, facts or occurrences that might take place outside the control of the decision-maker, but affect the outcome of decisions. Five categories have been identified: actor (or individual), organisation, product, business and political types of influences. The context mentioned in [1] was also taken into consideration in making the categorisation. Figure 7 shows how Environment element is structured. In the first version of the taxonomy a total of 41 elements were identified. Actors comprise of people involved in the decision-making process and several factors related to their individual characteristics were listed here. Organisational aspects are important and consist of factors affecting the decision (e.g., maturity level of the organisation). Product factors refer to aspects of the product not listed in the previous elements of the taxonomy that refer to external properties (e.g., maturity of a product, system type, etc.). Business and political considerations are facts that need to be taken into consideration when making a decision and might include for example agreements, partnerships and Intellectual Property Rights of the product, or other political considerations.

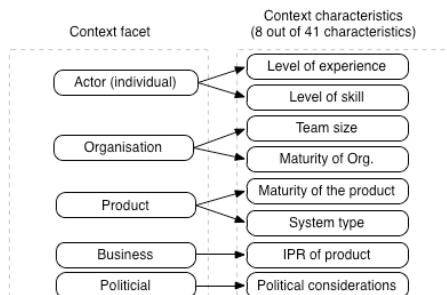


Figure 7. Main categories of Environment of GRADE.

4.2 GRADE Application Example

Above we provided descriptions of the main categories of GRADE. These were meant to be generic-enough so that their content can be customised according to the situation and context. For example, under a specific scenario and depending on how a

company defines Goals, Roles, Assets, Decision methods and criteria and Environment within the decision scenario, these could be instantiated differently. To better illustrate GRADE's applicability or feasibility we have instantiated GRADE through areal case scenario.

The GRADE taxonomy has been instantiated as follows: we first described a decision-making scenario based on a real industrial case from a consultancy activity of one of the co-authors of the paper and contributors of the work. Thereafter, we illustrate how the scenario can be described using an instantiation of the categories of the GRADE taxonomy.

The main purpose of the instantiation with a case study was to exemplify and illustrate the use of the taxonomy. Thereby, other researchers and practitioners should be enabled to classify their own decision processes. As the instantiation was based on one industrial case, it is at this point not feasible to infer the generalisability of the taxonomy. Consequently, when adding further cases extensions to the taxonomy are likely to occur. For example, in this case study a new element had to be added to the "Lifecycle" category (see Figure 8). Though, from a taxonomy point of view no main categories were missed.

Scenario description: A company investigated if an open source asset (MySQL) or a component off-the-shelf (Microsoft) should be chosen for developing a high-performance human resource management system. Following is a narrative description of the key elements of the taxonomy; the instantiation using GRADE is shown in Figure 8. In the figure, the grey elements are present in the scenario, and the orange element had to be added as it was not possible to classify the case based on the existing elements.

- **Goals:** The goal-prerequisites to be taken into account were reduced cost, long-term scalability and feature capability. All these were directly related to the selection of the architectural asset option that would enable improvement in internal business value, and no other goal-related statement beyond that was made. Hence, the goals for the scenario were described under elements found in the internal business value element of GRADE.
- **Roles:** The Chief Technical Officer (CTO) was the person who initiated the decision-making process. The stakeholders in the decision-making were architects/technical experts, developers, project managers, and external consultants. The CTO was also the person who finally took the decision. The Chief Financial Officer (CFO) was not involved in the decision-making process, but it was mentioned that he should have been. The roles described can be directly mapped using GRADE. The type of roles within the organisation described allows to implicitly determine whether the roles would be able to make strategic, operational, or tactical decisions. For instance, the CTO typically would operate on a more strategic level, the architect on a more operational as well as tactical level, and the developer on a tactical level. All decision-making functions could be clearly described with GRADE. The consultancy work of external consultants was not part of GRADE taxonomy and hence a new element representing "Support" in the lifecycle of roles was needed to be added. Thus GRADE was extended to include it.
- **Assets:** The two asset options considered in the scenario were development either using Component-off-the-Shelf (COTS) or Open Source Software (OSS). The asset type considered was software. The actual selection of the two asset options reflects

the attributes of the selected asset (i.e., functional suitability and economy).

- *Decision methods and criteria:* With regard to the decision methods, a feature list comparison of the alternatives was used in the scenario. A discussion took place, and estimations of pros and cons for the alternative options have been made. The decision criteria taken into consideration were financial- and product-related, and specifically were performance, reliability, security, cost, and scalability.
- *Environment:* The environment in which the decision took place was defined based on organisation and product constraints. The domain was Human Resource Management (HRM). The requirements engineering process was considered

bespoke initially, and later was market-driven. The size of the company was 400 people, the size of the development unit was 32 people. The development methodology was hybrid (combination of agile and plan-driven methodology).

The decision made from the company was to use MySQL, but some of the desired features (i.e., distributed syncing of the database) were not available in the solution taken. However, this could have been solved in other ways by the company (i.e., using smart hardware/software backbone). The cost of selecting the other alternative COTS option (Microsoft) was much higher and this made a difference in the decision.

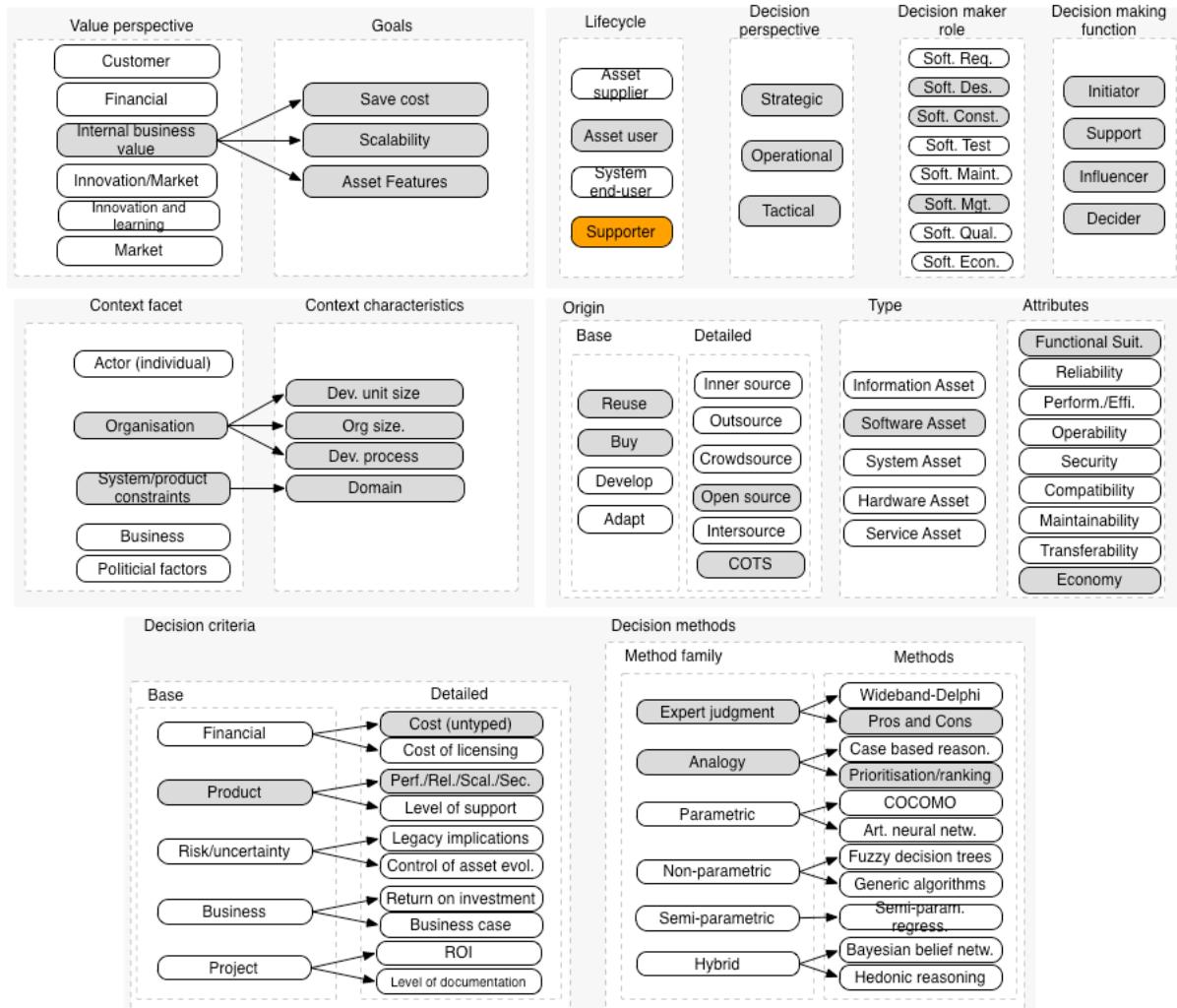


Figure 8. Case study instantiation using GRADE.

5. CONCLUSIONS

In summary, the work we conducted is towards improving and documenting decision-making and minimise risk in the topic of selecting architectural assets in systems and software engineering. The result of this study is a taxonomy, called GRADE. GRADE contains the key concepts of the elements of decision-making

which were also categorised to indicate generalisation relationships. The taxonomy was validated through a real case that instantiated the concepts and provided support for structuring a decision scenario. The taxonomy can be further used for individual cases classification (observations of individual case studies in a particular context) and also for future extensions of

the taxonomy (adding elements that were not identified yet). One using the taxonomy should have in mind that while many possible combinations can be realised, not all may be possible. Also some categories may correlate to more than one elements at the same time. In addition, some of the elements may appear in more than one of the categories analysed. The categories assist in advancing or improving of our understanding on defining aspects of strategic, operational and tactical decision-making in the selection of architectural assets in software-intensive systems engineering.

Two following main contributions were shown to be supported by the taxonomy:

- 1) Support practitioners in structuring decisions. Practitioners often make decisions on which architectural asset to choose among many alternatives. However, how they structure their decision-making process may not be conscious-enough or well-reflecting the decision. GRADE is a taxonomy offering alternatives on how to structure the decision-making process and minimise the risk of overlooking some beneficiary options.
- 2) Document current and hypothesised decision-making scenarios: In order to improve existing decision-making processes in practice we need to describe the facts in specific cases to both keep the consistency of the decision rationale in upcoming projects (e.g., taking into account the same goals and same constraints) and to reflect on the outcome of a decision. The GRADE taxonomy can support classification of existing processes and comparison of their outcomes. This enables (a) improving decisions within a specific context or company; (b) collecting empirical data on a large number of case studies irrespective of the output (successful or not).

Future work will focus on using the taxonomy for defining decisions in various domains to support research and practice in constructing software-intensive systems. Extended validation of the taxonomy will also be included through interviews of practitioners to compare the existing structure of GRADE and its applicability in more real-world cases.

6. ACKNOWLEDGMENTS

Our special thanks to the rest of the ORION project members than the authors that took part in this research. This work is partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.

7. REFERENCES

- [1] 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.
- [2] Eeles, P.: Understanding Architectural Assets. In: Software Architecture. Seventh Working IEEE/IFIP Conference on Software Architecture, pp. 267-270 (2008)
- [3] Mcilroy, D.: Mass-produced Software Components. In: Proceedings of Software Engineering Concepts and Techniques, pp. 138–155 (1969)
- [4] Crnkovic, I., Chaudron, M., Larsson, S.: Component-Based Development Process and Component Lifecycle. Int. Conf. Softw. Eng. Adv. (2006)
- [5] van Ommering, R., van der Linder, F., Kramer, J., Magee, J.: The Koala Component Model for Consumer Electronics Software. Computer (Long. Beach. Calif), 33 (3) 3, 78–85 (2000)
- [6] van Heesch, U., Avgeriou, P., Hilliard, R.: A documentation framework for architecture decisions. J. Syst. Software, 85 (4), 795-820 (2012)
- [7] Jansen, A., Bosch, J., Avgeriou, P.: Documenting after the fact: Recovering architectural design decisions. J. Syst. Softw., 81 (4), 536-557 (2008)
- [8] Sawada, A., Noro, M., Chang, H.M., Hachisu, Y., Yoshida, A. 2011. A design map for recording precise architecture decisions. In: 18th Asia Pacific Software Engineering Conference (APSEC'11). IEEE.
- [9] SWEBOK Guide V3.0, <http://www.computer.org/web/swebok/v3>
- [10] Glass, R.L., Vessey, I. Ramesh, V.: Resres: The story behind the paper “research in software engineering: An analysis of the literature”. Information & Software Technology, 51 (1), 68–70 (2009)
- [11] Blum, B. I.: A taxonomy of software development methods. Communications of the ACM, 37 (11), 82–94 (1994)
- [12] Smite, D., Wohlin, C., Galvina, Z., Prikladnicki, R.: An empirically based terminology and taxonomy for global software engineering. Empirical Software Engineering, 19 (1), 105–153 (2014)
- [13] Unterkalmsteiner, M., Feldt, R., Gorschek, T.: A taxonomy for requirements engineering and software test alignment. ACM Trans. Softw. Eng. Methodol., 23 (2), 16-54 (2014)
- [14] Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.:A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24 (3), 45-77 (2007)
- [15] A very fast intro to decision theory, <http://www.siue.edu/~availat/decision.htm>
- [16] Quinlan, J.R.: Induction of decision trees. Machine learning 1 (1), 81-106 (1986)
- [17] Khurum, M., Gorschek, T., Wilson, M.: The software value map - an exhaustive collection of value aspects for the development of software intensive products. Journal of Software: Evolution and Process 25 (7), 711-741 (2013)
- [18] ISO/IEC 27002:2005, Information technology-security techniques—code of practice for information security management (2005)
- [19] ISO/IEC JTC1/SC7 N4098, CD 25010.2, Software engineering—Software product Quality Requirements and Evaluation (SQuaRE) Quality model, WG23, 25010, N3803, N4008, N4097 (2008)
- [20] Trendowicz, A., Jeferry, R.: Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success. Springer Publishing Company, Incorporated (2014)