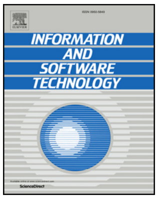
ARTICLE IN PRESS

Information and Software Technology xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



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

Efi Papatheocharous*,[b], Krzysztof Wnuk[a], Kai Petersen[a], Séverine Sentilles[c], Antonio Cicchetti[c], Tony Gorschek[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 knowle.g. [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, utilized 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.gorschek@bth.se (T. Gorschek), syed.shah@sics.se (S.M.A. Shah).

https://doi.org/10.1016/j.infsof.2018.02.007
Received 9 August 2017; Received in revised form 25 November 2017; Accepted 20 February 2018
0950-5849/

Please cite this article as: Papatheocharous, E., Information and Software Technology (2018), https://doi.org/10.1016/j.infsof.2018.02.007

house, outsourced, components of 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 minimize 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 knowle.g. [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 regarding 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