Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

A PROPOSITION OF A SET OF DESIGN PRINCIPLES FOR REQUIREMENT PRIORITIZATION ARTIFACTS

RAHUL THAKURTA
XAVIER INSTITUTE OF MANAGEMENT BHUBANESWAR
rahul@ximb.ac.in

ABSTRACT

Prioritization of requirements towards finalizing the intended requirement set has considerable received attention from academicians and practitioners. Articles proposing requirement prioritization artifacts have focused mostly on describing these artifacts. There is no attempt to uncover the principles that guide the design of these artifacts towards holistic understanding of the prioritization process. We address the gap by proposing a set of design principles that explains the form and function of a requirement prioritization artifact and in the process address the existing limitations. Subsequently, we carry out an evaluation in two rounds to demonstrate the validity and utility of the design principles.

Keywords: Requirement Prioritization, Design Principles, Content Analysis, Systematic Review

INTRODUCTION

The importance of prioritizing project requirements has been acknowledged by the academic and the practice community over decades [42]. Requirements are generally described as what the system is required to do along with the environment it is intended to operate in. Requirements provide the description of the system, its behaviour, application domain information, system constraints, specifications and attributes [54]. The importance of prioritizing requirements stems from the fact that not all requirements can usually be met with available time and resource constraints. Efficient and trustworthy methods for prioritizing requirements are therefore strongly demanded by practitioners. The increased attention behind prioritization has led the academic community to explore different mechanisms by which prioritization of requirements can be achieved. The results of the same are different types of requirement prioritization artifacts disseminated in various scholarly communications. Review articles on requirement prioritization also discuss the characteristics, strengths, and limitation of these prioritization artifacts so as to offer suggestions for usage by the practitioners. These documentations of the different requirement prioritization artifacts are indeed valuable contributions to the domain of requirements engineering. However, the respective articles are mainly descriptive, and these mostly discuss structural properties (i.e., components and their interplay) of these artifacts. The articles give little insights into the design principles (DPs) – that is, the principles of form and function, the requirement prioritization artifacts should meet.

Our research raises the question of what are the principles governing the design of a requirement prioritization artifact. The work is motivated by the limitations of the existing requirement prioritization artifacts [2], and awareness that literature on requirement prioritization has not yet attempted to arrive at the set of DPs and related classification of requirement prioritization artifacts. Hence, we carry out a systematic review of requirement prioritization artifacts reported upto 2013 in order to identify the common structures characterising the form and function of the existing artifacts in the process of deriving the set of eight DPs. These are then subsequently validated in two rounds based on expert reviews in the first and analysis of the applicability of the DPs on an artifact under construction in the second. By presenting the DPs, we provide a design science contribution, extending the current knowledge, and assisting practitioners to make better use of artifacts in their work sphere. To maintain consistency in terminology, we use the term "requirement" in this paper to refer to software requirement.

The remainder of the article is organized as follows. In the next section, we present the background where we discuss the contributions and limitations of requirement prioritization artifacts to explain the research gap. Subsequently, we describe our research process in detail. Next, we introduce a framework for categorizing DPs which are then subsequently elaborated. In the following section, we carry out an evaluation of the DPs. The final section summarizes our contributions, discusses limitations of our work, and provides topics for future research.

BACKGROUND

Benefits of Requirement Prioritization

There are several advantages of prioritizing requirements. Prioritization facilitates the implementation of a software system based on preferential requirements of stakeholders [3]. Berander et al. [14] indicate that prioritization of requirement facilitates elicitation of requirements in various software releases as it might not be possible to incorporate all the requirements in one single release. Perini et al. [77] suggest that requirement prioritization is critical as it enhances software release planning, budget control and scheduling. Furthermore, software products that are developed based on prioritized requirements are expected to have a lower probability of being rejected.

Requirement Prioritization Artifacts

A number of requirement prioritization artifacts have been proposed by various authors till date. Accepting that design is a creative process, there has been recognition of the fact that a general design approach cannot be formalized [45]. This has been the foundation behind design of the different prioritization artifacts with each having various capabilities and limitations in order to realize the intended benefits. Among the initial artifacts on requirement prioritization, the Analytic Hierarchy Process (AHP) [96] defines a structured approach where a decision problem is broken down into a hierarchy of sub-problems. Evaluation then takes place through pair-wise comparison of the elements constituting the hierarchy. The use of AHP is limited by the assumption that all the comparisons can be quantified and the assessment is based on a common ratio scale. The numerical assignment artefact is based on grouping requirements into different categories, viz. high, medium, and low [92]. A criticism of the approach is that use of categories like high, medium, and low may confuse the stakeholders since different stakeholders may have different views of what, for example, high and medium mean. Some of the recent artifacts like the MoSCoW [102] approximate the numeral assignment technique where the requirements are roughly classified in priority groups depending on importance. The letters stand for: M - MUST have this, S – SHOULD have this if at all possible, C - COULD have this if it does not effect anything else, W - WON'T have this time but would like in the future. The importance of this method is that when prioritizing the words mean something and can be used to discuss what is important. A limitation of the artifact is that the classification of these requirements into groups could be very subjective and differ across stakeholders, and mutual comparison of requirements is also not considered. Trienekens et al. [99] have proposed a step wise approach based on the ISO9126 standards and the AHP to include considerations of software quality in the process of mission-critical requirements engineering. A limitation of this artifact is that requirement dependencies are not explicitly included in the artifact design. Kukreja [56] presents the TOPSIS (Technique of Ordered Preference by Similarity to Ideal Solution) framework in which requirements are prioritized on a weighting scale towards satisfaction of business goals. The limitations of this artifact are the inability to handle hierarchical requirements and the inability to update requirement ranks in cases of requirement evolution.

The above evidences relate to some of the artifacts on requirement prioritization that has been proposed in the literature and the limitations of these artifacts. Achimugu et al. [2] indicate that the major limitations facing the existing requirement prioritization artifacts are lack of scalability, methods of dealing with rank updates during requirements evolution, stakeholder considerations, and requirement dependency issues. The evidences also indicate some of the considerations that have been used for designing requirement prioritization approaches. The referenced requirement prioritization artifacts are governed by specific objectives and specify different methods for prioritizing the intended requirement set. The evidences also point at the future design possibilities to address the existing limitations. However in the literature, there is absence of specific guidelines that can be used to structure the form and function of the intended requirement prioritization artifact, which we address in this research.

RESEARCH METHOD

Design Science Paradigm

Design science research (DSR) as a problem-solving paradigm [44] strives to create innovative artifacts as a solution to problems faced by stakeholders in different domains [40]. The basic principles of design science research (DSR) can be traced back to sciences of the artificial [89]. The DSR stream in contrast to behavioral science does not seek to understand the world as is exists and works; rather it strives to develop solutions to improve the current state of affairs. DSR intends to provide information technology (IT) artifacts that extend the current state of the art and serves human purpose. The deliverables of a DSR project can be at three levels with the levels indicating various maturities of contributions [39]. Level-1 represents specific instantiations in the form of products or processes. Level-2 represents more abstract contributions in the form of nascent design theory (e.g., constructs, design principles, models, methods, technological rules, etc.). Finally level-3 represents well-developed design theories about the phenomena under study. The contribution of our study is at level-2 where design principles are formalized to guide construction of requirement prioritization artifacts.

Research Overview

We resorted to a non-experimental approach using content analysis to identify and analyze articles on requirement prioritization in order to arrive at the set of DPs presented in this article. A systematic review of the identified articles was carried out by two research assistants whom we had engaged, based on a codebook which we had developed comprising of categories and sub-categories. These were refined and new categories and sub-categories were developed to suitably classify evidences emerging out of the articles under review. At the end of the classification process the sub-categories were analyzed for patterns. The creation of the codebook and the derivation of the patterns were facilitated by a third researcher (i.e., the author itself). The final results reflecting the essentials of form and function are presented in the form of DPs in this paper. Deriving DPs in this manner based on evidences from the literature has been used in some existing studies, for example, Bitzer et al. [17], Pöppelbuß and Röglinger [78].

Data Collection and Description

We base our findings on a systematic review of literature on requirement prioritization in order to formalize the DPs. We used the following search strings: (1) (requirement OR requirements), (2) (prioritization OR prioritize OR prioritizing OR selection OR dependency OR management OR negotiation OR conflict), and these were concatenated using the Boolean AND operator in the search query. We had to proceed like this in absence of any standardized, consistent terminology with respect to requirements prioritization. We also carried out a reference check within articles which presented some kind of discussion on prevailing requirement prioritization artifacts so as to ensure we don't miss out on potential articles matching the research objective. In addition, we reviewed works on requirement engineering as these may include prioritization aspects without mentioning the same in the search fields. We applied the search query on the fields: metadata, title, abstract, and keywords as per the search specifications allowed by the data sources listed in Table 1. These sources were finalized based on a review of some of the existing systematic review and systematic mapping studies in the domain of requirement prioritization, for example, Achimugu et al. [2], Herrmann and Daneva [42], and Pergher and Rossi [76]. The data sources mentioned in these articles were checked for appropriateness before their inclusion in the list given below. Additionally, we also included two distinguished conference on information systems viz. European Conference on Information Systems (ECIS), and the International Conference on Information Systems (ICIS). Design Science Research in Information Systems and Technologies (DESRIST) conference was included given the research contribution to the design science research domain. In Table 1, alongside the journal entries, the journal impact factors (IF) pertaining to the year 2014/2015 are listed.

The outlets mentioned in Table 1 were searched for relevant articles up to 2013. In the search, we excluded editorials, prefaces, summaries of articles and tutorials, workshops, panels and poster sessions. In certain cases, where the description of a requirement prioritization artifact in concern was traced to some other source like books and websites, we referred to those works. We carried out the search individually in the identified sources between February and July, 2014. Our search strategy resulted in identification of over 1000 articles (up to 2013), and these were further screened to ascertain if the articles were addressing construction or evaluation of requirement prioritization artifact. Based on the screening results, we were able to shortlist 91 articles for full-text review. These articles are written in English and published in 2013 or earlier. In the Appendix, we include a list of the artifacts on requirement prioritization which is classified in terms of the type of artifact like methods, models, etc.

Table 1: Selected Data Sources

Journal Publications:

- ACM Transactions on Software Engineering and Methodology (1.170)
- Communications of the ACM (3.621)
- Decision Support Systems (2.313)
- Empirical Software Engineering (2.161)
- Expert Systems with Applications (2.240)
- IEEE Software (1.053)
- IEEE Transactions on Software Engineering (1.614)
- IET Software (0.595)
- Information and Software Technology (1.046)
- International Journal of Software Engineering and Knowledge Engineering (0.362)
- Journal of Systems and Software (1.352)
- Journal of Systems Architecture (0.440)
- Requirements Engineering (0.882)
- Software Process: Improvement and Practice (1.32)*
- Software Quality Journal (1.143)

Conference/Symposium Proceedings:

- Agile Conference
- Design Science Research in Information Systems and Technologies
- Empirical Software Engineering and Measurement
- European Conference on Software Maintenance and Reengineering
- European Conference on Information Systems
- Euromicro Conference on Software Engineering and Advanced Applications
- Conference on Software Engineering Education and Training
- Conference on Systems Engineering Research
- IEEE/ACM International Conference on Automated Software Engineering
- IEEE International Conference and Workshops on Engineering of Computer Based Systems
- IEEE International Requirements Engineering Conference
- International Conference on Software Engineering
- International Conference on Information Systems
- Requirements Engineering – Foundation of Software Quality

** The corresponding IF pertains to the year 2013. 2014/2015 data is not available*.

Data Analysis

The content analysis commences by creating and defining categories and continues by pre-testing each category's definition, revising categories (if necessary) and eventually categorizing all the data [27]. We used the Ms Excel spreadsheet software to carry out content analysis. We followed the guidance offered in Wolfswinkel et al. [104] in order to develop the codes and carry out the review. We first demarcated various sub-areas following the general structure of a research paper. These represented the objective, motivation, research methodology, prioritization description, prioritization results, method evaluation and work contribution. Open coding technique was then applied in order to generate the codes to capture the themes represented in each article and pertaining to these sub-areas. Codes were generated from article keywords, analysis of the article abstract and, relevant content.

To facilitate the coding process, an excel template was created with individual rows assigned to the papers under classification. The two research assistants used the template to code the papers independently. The level of agreement between the two coders was assessed based on the Cohen's kappa coefficient [21] which measures the inter-rates reliability. A couple of iterations involving revision of the codes were required until the final value of Cohen's k (0.82) was found to be in the acceptable range [30]. At the end of each iteration the cases of disagreements were discussed in presence of the third researcher so that either an agreement was achieved with respect to coding of the data, or a new subcategory was developed that satisfied the research objective. This process took approximately nine weeks.

In the next stage which spanned approximately four weeks, the categories which emerged from the first stage were selectively merged to arrive at the set of thirteen subcategories related to the research objective. These subcategories were revised iteratively to make sure it was not only parsimonious but also represented the diversity of the initial coding. The last stage of the analysis involved derivation of the patterns constituting our DPs based on the subcategories' which emerged from the data. The work at this stage was carried out jointly (i.e. the author and the two research assistants) in a workshop format and required approximately five weeks for completion. The process was iterative and it involved revisiting the meaning of the thirteen subcategories which has been defined at the previous stage, going back to the articles to identify how the subcategories informed the prioritization artifact, identifying conceptual coherence, etc. To give an example, the occurrence of the subcategories "prioritization motivation" and "participating stakeholders" derived out of the set of codes {value focused, combining techniques, comparing techniques, process improvement, requirement satisfaction, decision analysis, etc.} and {product manager, requirement specialist, customer, developer, etc.} respectively were examined in the concerned articles, and the pattern "stakeholders' prioritization perspective" was defined. The other key patterns identified in this way were: objectives guiding prioritization, representation: prioritization requirements, representation: participating entities, prioritization process dependencies, description: prioritization mechanism, representation: prioritization artifact, and representation: prioritization results. These patterns are finally presented as DPs that we address in this paper.

A FRAMEWORK FOR CLASSIFYING DESIGN PRINCIPLES

Based on the analysis of existing literature dealing with requirement prioritization artifacts, we propose the following framework (Figure 1) to structure our design principles. The figure indicates that the prioritization process is guided or motivated by the prioritization concerns. The end result of the design manifests as an artifact towards prioritization of the candidate requirements. The contextual factors can influence the prioritization concerns, process, or the outcome.

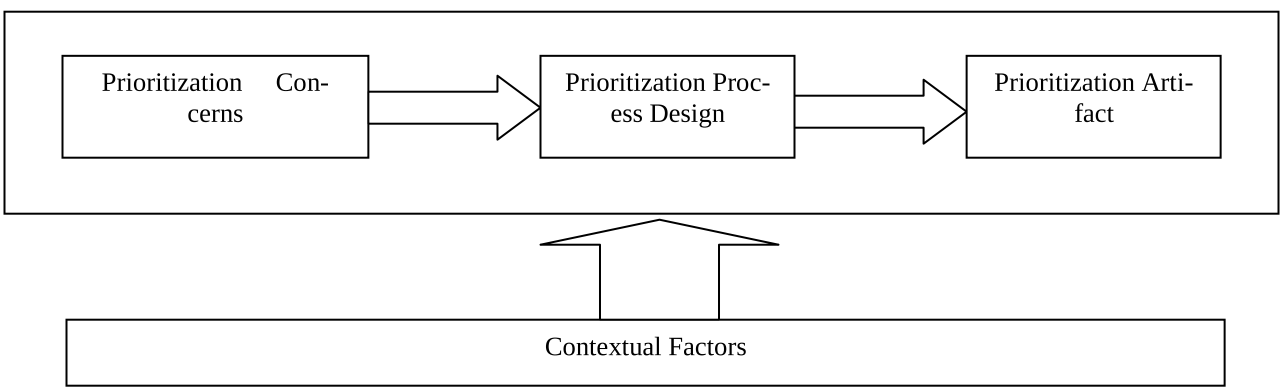

Figure 1: Framework for Classifying Design Principles

Prioritization concerns relate to the stakeholder objectives the prioritization is expected to fulfill. Explicating these concerns in the study is important, since the usefulness of a prioritization artifact is dependent on the purpose for what it is needed. Prioritization process design relate to the investigation of the processes and mechanisms to carry out prioritization of the candidate requirements. Under prioritization artifact, we explore the different types of artifacts that have been proposed towards prioritization of the requirements, and examine the nature of prioritization outcome from usage of these artifacts. Based on the results of the review, seven types of artifacts on requirement prioritization can be identified viz. (1) Constructs, (2) Model, (3) Method, (4) Methodology, (5) Framework, (6) System, and (7) Tool. The contextual factors are the project related or external factors that have an influence on the prioritization of requirements.

DESIGN PRINCIPLES FOR REQUIREMENT PRIORITIZATION ARTIFACTS

Synthesis of the coding results based on the classification framework (Figure 1) lead us to a set of eight DPs (Table 2) which we discuss below. We also do not make any claim for completeness as requirement prioritization artifacts in future may include constructs, methods, models, etc. that are hitherto not considered relevant. Instead the DPs intend to assist practitioners and researchers in comparing amongst prevalent requirement prioritization artifacts. It also serves as a checklist when designing new artifacts. We see our DPs as being necessary, but not sufficient in this regard.

Table 2: Design Principles

Related To	DP #	Design Principles (DP)
Prioritization Concerns	DP 1	Determine the prioritization perspective
	DP 2	Determine the over-arching objectives guiding the prioritization process
	DP 3	Identify candidate requirements for prioritization
Contextual Factors	DP 4	Identify the participating entities and relevant attributes
Prioritization Process Design	DP 5	Establish procedures to handle associations among interacting entities
	DP 6	Establish a method to carry out prioritization
Prioritization Artifact	DP 7	Specify the nature of the artifact
	DP 8	Develop a representation scheme to communicate the final results

Related to - prioritization concerns:

DP 1: Determine the prioritization perspective

The primary users of a requirement prioritization artifact can be customers, requirement specialists, project manager, product manager, etc. The perspectives (i.e. priorities and concerns) of these stakeholders are likely to be different, thereby influencing the design and the usage of the prioritization artifact under consideration. DP 1 specifies the need for identifying these perspectives influencing the design of the prioritization artifact. These perspectives might have implications on the procedural considerations of the artifact, the nature of representation of the artifact, the representation scheme for communicating the final results, etc. By including stakeholder perspective, we also address a design limitation (e.g. stakeholder considerations) mentioned in the section “background” above.

DP 2: Determine the over-arching objectives guiding the prioritization process

The second step towards designing a requirement prioritization approach is concerned with specifying the objectives guiding the prioritization process. The requirement prioritization might be targeted at ranking the given set of requirements on specified attributes, identifying the requirements to be considered for implementation at different phases of the project, finding out the association among project requirements and the overall project goals, etc. This implies that there can be different objectives that can inform the prioritization process. It is also possible to have multiple objectives simultaneously informing the prioritization. For example, the prioritization of requirements for an MIS project may have dual objective: (1) maximization of value of requirements; and (2) minimization of cost of implementing these requirements. Being an MIS project, the organization in concern may assign more importance to objective-1 in comparison to objective-2 as the costs may have to be borne internally (assumed). Likewise, it is also possible to have three or more objectives guiding the prioritization process. The level of importance of the identified objectives can also be ascertained at this stage. Both the objective and the level of importance can be specified as qualitative statements for comprehension and communication.

DP 3: Identify candidate requirements for prioritization

DP 3 specifies identifying the requirements that are candidate for prioritization. These form the master list of requirements considered for prioritization. The identification of these requirements may be implied through the objective statements, for example, an artifact for prioritizing functional requirements would necessarily include the functional requirements as its candidate set. The requirements can be specified in terms of statements, and identifiers can be used in order to uniquely identify these requirement statements. Subsequent to identification, it is possible to categorize the requirements in one or more dimensions such as requirement classes (i.e. based on requirement type, for example, functional requirements, non-function requirements-NFRs, database requirements, etc.), requirement hierarchy (i.e. based on parent-child relationship existing among the identified requirements), requirement importance (i.e. based on requirement preferences that may be specified by stakeholders at the onset), etc. This again could be governed by specifications laid down in other DPs, i.e. prioritization objectives (DP 2).

Related to - contextual factors:

DP 4: Identify the participating entities and relevant attributes

The design of a requirement prioritization artifact might include representations of different entities based on the considerations adopted during the design. These can be internal (e.g. those pertaining to the project, process, product or service utilizing the requirements considered for prioritization) or external (e.g. other considerations influencing the prioritization process). Relevant attributes of these participating entities may influence the design of the requirement prioritization artifact under consideration. For example, the design of a requirement prioritization artifact may require inputs on requirement costs, requirement quality, requirement usability, and requirement performance, which are different attributes of the participating entity: project requirements. Based on the review of existing literature, a number of attributes influencing requirement prioritization were identified and is listed in the Appendix. These attributes were grouped in the following dimensions: requirement attributes, project attributes, process attributes, product-service attributes, subject attributes, and prioritization environment. Table 3 presents a description of these dimensions. DP 4 prescribes that the entities that inform the design of the concern requirement prioritization artifact should be identified and represented suitably so that the intended design objectives can be successfully met.

Table 3: Description of the Identified Dimensions

Dimensions	Description
Requirement Attributes	This relate to the specific requirement characteristics that might influence the prioritization process
Project Attributes	This relate to the description of the project context influencing the prioritization process
Process Attributes	These are specific process related characteristics having an influence on the prioritization process
Product-Service Attributes	These are attributes related to the product or the service under consideration and to be rendered by the project implementing the concerned requirements
Subject Attributes	These relate to the stakeholder characteristics influencing the prioritization process
Prioritization Environment	These relate to the organization and market factors influencing requirement prioritization

Related to - prioritization process design:

DP 5: Establish procedures to handle associations among interacting entities

It is possible to have associations among the entities participating in the prioritization process. The nature of such association can be positive or negative with different degrees (extent). Considering a specific attribute of an entity, a positive association between two entities (say X and Y) implies that presence of X results in some kind of improvement in the concerned attribute of Y. Conversely, a negative association between two entities (say

X and Y) with respect to an attribute implies that presence of X causes to decrease the attribute of Y. The extent of association indicates the magnitude of improvement or degradation, expressed in suitable format. The type of associations can be intra-class (i.e. association among entities belonging to the same class, for example, association among NFRs [54] or inter-class (i.e. association among entities of different classes, for example, association between requirements belonging to class NFRs and class functional requirements). To elaborate the inter-class association, it is possible that incorporating security (a NFR) have a negative influence on the design features of a page in terms of effort needed to implement the page. Additionally these associations (intra-class or inter-class) are not restricted to only requirement classes but can also include all participating entities influencing the prioritization process as observed by an expert during the evaluation process (Table 5). DP 5 prescribes the need to consider possibility of these associations in its prioritization process and thereby address a design limitation (e.g. requirement dependency considerations) mentioned in the section “background”. The procedure may not specify anything (i.e. in case associations are all ignored), or may specify the rules to handle such associations (i.e. in case associations among the participating entities need to be addressed).

DP 6: Establish a method to carry out prioritization

This principle specifies the design of the method to be employed in order to arrive at the prioritization results. Drawing from design science literature [44], a design science contribution at this level is the novelty of the method employed to accomplish the prioritization, and achieve the intended objective. The method integrates relevant information captured till this point (i.e. as specified by the other DPs) and hence depending upon the extent of information can be of varied complexity levels. The prioritization method may specify grouping of requirements [68], introduce pair-wise comparison of requirements [50], specify actions in order to handle requirement addition or update or deletion during prioritization process [38], specify the nature of the computation process (i.e. single-step, iterative) [85], address scalability considerations etc. The implementation of the method is accomplished within the prioritization artifact under consideration.

Related to – prioritization artifact:

DP 7: Specify the nature of the artifact

DP 7 focuses on specifying the nature of the planned artifact. The different artifacts can be any of the seven types of artifacts listed above. A description of these artifacts is provided in Table 4. The representation of these artifacts can be at a conceptual level or at a physical level. The first six types of artifacts (i.e. constructs, model, method, methodology, framework, and system) are representations of artifacts at a conceptual level implying that the concerned artifact has been presented in an abstract manner. The last type of artifact (i.e. tool) indicates a physical instantiation of the artifact in concern. The physical instantiation may be achieved by implementing the tool as an application in a suitable platform.

Table 4: Artifact Description

Artifact Type	Description
Construct	Set of vocabulary or symbols that informs the design of a requirement prioritization artifact
Model	A specification (abstraction / representation) of some sort of relationships among established or identified constructs that informs the design of a requirement prioritization artifact.
Method	A specification of algorithms, practices, approaches, or techniques towards designing the requirement prioritization artifact.
Methodology	A specification of a set of methods, principles and rules towards designing the requirement prioritization artifact.
Framework	A specification of the structure (real or conceptual) towards supporting or guiding the design of the requirement prioritization artifact.
System	A specification of an integrated set of components that describes a requirement prioritization artifact.
Tool	A suitable representation of the physical instant of a requirement prioritization artifact.

DP 8: Develop a representation scheme to communicate the final results

The last design principle is on how to represent the results of prioritization. The prioritization may be governed by different objectives and may be of varied complexity levels depending upon the extent of entity participation and representation. The prioritization results may be reproduced in a quantitative format, may be sorted on appropriate scales in specific order (i.e. ascending, descending), or may be qualitatively represented and interpreted. To include some examples, the numerical assignment method [47] presents results by classifying prioritization output into three predefined categories viz. high, medium, and low. The cumulative voting technique [80] results in a ranked list of requirements from most important to least important. Top ten requirements [42] also provide a ranked list of requirements in terms of requirement importance to the stakeholders.

EVALUATION OF THE DESIGN PRINCIPLES

We adopted two rounds of evaluation in order to establish the validity and the utility of the DPs. The first round involved participant's feedback on the DPs. The results of the feedback led to refinement of the meaning and scope of some of the DPs. The applicability of the revised set of DPs on a design science artifact under construction was evaluated next. We describe both of them below.

Evaluation Based on Participant's Feedback

We gathered practitioner's feedback on the DPs to ensure that our recommendations would be valuable in practice. We chose to involve practitioners in the evaluation because we consider their view on the DP especially valuable. Their involvement ensured that we did not omit important DPs. Further, we believe that this evaluation has the potential to increase the relevance and utility of our findings which has been extensively discussed in IS research [35]. The evaluation was carried out by sharing the DPs and their detailed descriptions with the practitioners and gathering written feedbacks on the same. The descriptions of the DPs along with a specific example (i.e. how the design principles relate to the design of the "top ten requirements approach"), and a feedback form (included in the Appendix) were emailed to the practitioners. We chose this specific example as the design of the concerned artifact (top ten requirements) is very lucid and can be easily explained. There were five questions in the feedback form related to evaluation of the DPs, and these mostly related to their perception on the formulation of the design principles, aspects with which they disagree, and suggestions on extensions and modifications of the DPs and the descriptions. Six practitioners agreed to carry out the evaluation, all being males aged 45-55, and working at senior management positions in IT organizations. Involving only six experts can still be considered reasonable as related studies have used similar number of experts in the research process i.e. use of seven interviews in a mobile user interface validation study [100], and use of six experts in evaluation of design principles for learning service delivery [17]. These practitioners had expertise in various capacities in system analysis and design, and all of them have dealt with project requirement issues (e.g. requirement identification, negotiation, etc.) in previous engagements. Overall, the initial set of DPs, which we had constructed obtained quite positive results. We provide in Table 5, some exemplary data on the feedbacks received during this evaluation stage, and the implication on the DPs.

Evaluation Based on Artifact under Construction

Round two of the evaluation of the refined DPs was carried out based on a design science project that is currently underway. The project aims at developing a framework to determine the priority of a list of NFRs for implementation during software development. The partial results of the work are provided in Thakurta [95], where the framework along with validation and working details has been presented. The present efforts are focused on instantiating the framework using spreadsheet application software.

The objective of the design science project to develop and instantiate a framework in order to arrive at a ranked list of NFRs for final implementation corresponds to DP 2 (objectives) and DP 7 (artefact nature) specified in Table 2. The design science project introduces a six step framework (Table 6) in order to arrive at a set of NFRs that catered to the above mentioned objectives. We now provide a brief description of these steps and relate how these steps inform the list of DPs which we have proposed earlier.

Table 5: Selected Expert Feedbacks on the Design Principles

DP	Practitioner Feedback	Implications
DP 1	"*It is great that it [design principles] also considers the participating stakeholders"* [Software Quality Analyst]	The comment supports DP 1. No changes were required.
	"*Organizational actors from different communities probably have different prioritization schemes (e.g., a sales engineer vs. a platform/core engineer), resulting in incompatible priority orders for implementing software requirements ...*" [Business Analyst]	The observation explicates the need of designing mechanisms to balance different prioritization perspectives. Future artifacts on requirement prioritization may incorporate this consideration in their design.
DP 2	"*Generally the focus is on ranking of requirements (functional)*" [Senior Business Architect] "*Presence of multiple objectives governing the prioritization may be explored* " <example provided> [Product Manager]	The observation and the suggestion are related to the overarching objective governing the prioritization process. The formulation of DP 2 includes both these considerations and hence no changes were required.
DP 5	"*I may also want to include associations among other objects"* <example provided> [Senior Consultant]	The comment specifies the need to include other type of associations. DP 5 was refined to include additional possibilities as its description indicates.
DP 6	"*Scope creep is a problem for us, and good so see the aspect included in the design principles*" [Product Manager]	The comment acknowledges the reference to requirement addition and update in the description of DP 6. No changes were required.
DP 7	*"If we want to come up with a new prioritization mechanism, first we need to develop a conceptual representation ..."* [Project Manager]	The comment emphasised the need to specify the level of abstraction of the proposed artifact. We have taken this into consideration in DP 7. Hence no changes were required.
DP 8	"*The results should be easily interpreted*" [Business Analyst]	The comment reinforces the need to adopt suitable representation schemes to present the final results. This is as per DP 8 specifications.

Table 6: Steps in the Project & DP Correspondence

Step #	Description	Corresponding DPs
1	Identification of NFRs	DP 1, DP 3
2	Creation of a project level scenario	DP 4
3	Linking scenario to business objective	DP 5
4	Assessment of the scenario relevant NFRs	DP 5
5	Adjustment of scenario NFR scores	DP 6
6	A heuristic for deciding which NFRs to be dropped from a scenario	DP 6, DP 8

Step 1: Identification of NFRs

Step 1 of the NFR prioritization framework involves identification of the NFRs that are perceived to be important by the representatives of the business organization. A couple of business representatives identified the NFRs from a master list that was presented to them ([95], pp. 584). The inclusion of business representative's preferences relate to DP 1 (i.e. determine the prioritization perspective) and NFR identification by the stakeholder groups relate to DP 3 (i.e. identify candidate requirements for prioritization).

Step 2: Creation of a project level scenario

The paper introduces a scenario as a conceptual system with specific capabilities (i.e. NFRs), and providing some functionality to the business. The objective of a scenario is to assess if the set of functional requirements and NFRs contained within the scenario is good enough to satisfy the business objectives, and in the process identify improvement alternatives. Figure 2 given below (reproduced from reference [95], pp. 580) presents a component level view of a typical scenario consisting of a business function (indicating what an organization does) at the top, hierarchy of business processes (indicating how the work identified by the business function is accomplished), and the lowest level implementing the use cases. The NFRs represent the attributes of the scenario as a whole (i.e. system level attributes).

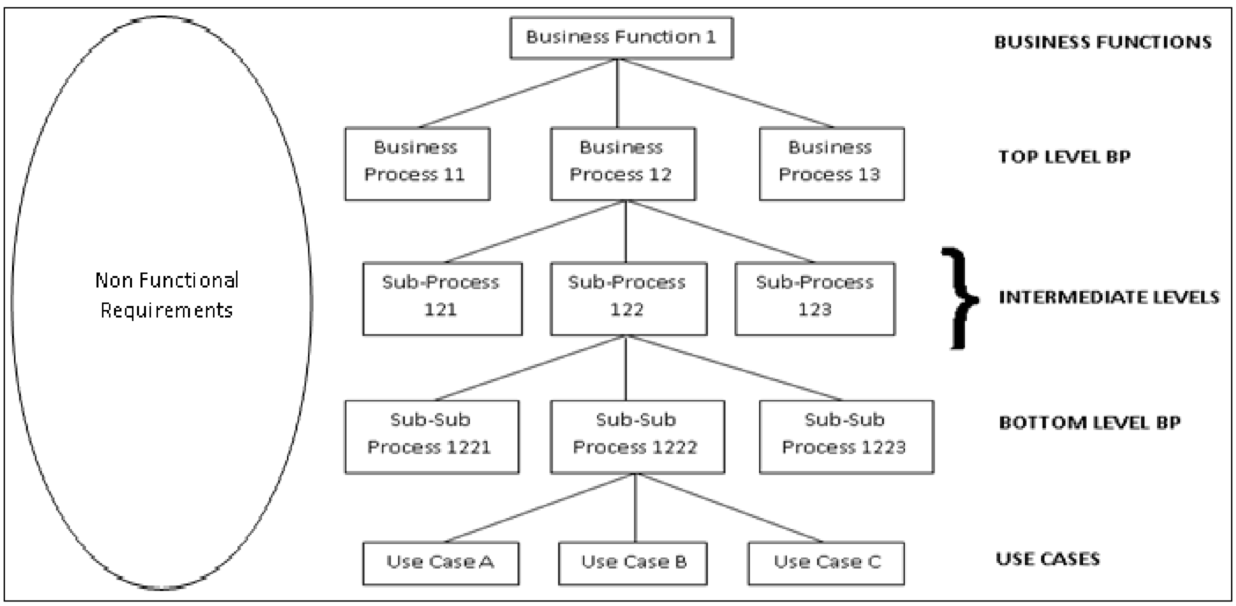

Figure 2: Components of a Scenario [95]

The inclusion of business components (i.e. business processes and business functions) in the NFR prioritization framework relate to DP 4 (i.e. identify the participating entities and relevant attributes). The construction of the scenario hints that the NFR prioritization framework could be part of a broader framework that relates business measures to project requirements. Further use cases and NFRs represent two requirement classes with possibility of inter-class interaction effects. However, these possibilities have not been investigated or reported in the concerned paper.

Step 3: Linking scenario to business objective

Step 3 in the NFR prioritization framework addresses the business contribution of a constructed scenario on specified parameters. This is a kind of inter-entity association (DP 5) where the entity scenario relates in some way to an entity representing business value. The author represents business value as a multi-dimensional construct using 24 business value dimensions (BVDs) (BVDs are related categories of business value which determines the health and well-being of an organization in the long-run [23]). The association between the entity scenario (Figure 2), and the 24 BVDs was captured on a 0 (indicating absence of association) to 9 (maximum) scale based on a questionnaire having question items as follows: '*In the context of the proposed scenario, how much importance would you ascribe to the scenario's impact on <chosen business value dimension>*'

Step 4: Assessment of the scenario relevant NFRs

In step 4 of the NFR prioritization framework, two aspects were addressed. The first aspect relates to assessment of the perceived value of NFRs under consideration. The framework inherits steps from Wiegers framework [103] and the NFRs were assessed from the following four dimensions of value (indicating importance of the NFR in the scenario), penalty (a measure of loss if the NFR is not included in the scenario), cost (cost of implementing the NFR in the scenario) and risk (business and technical risks associated with implementing the NFR in the scenario). The second aspect relates to inclusion and quantification of mutual association among NFRs in the framework. Considering the NFR class, presence of capabilities was found to enhance or degrade certain other capabilities. This was captured using an NFR association matrix shown in Table 7.

Table 7: NFR Association Matrix Pertaining to the Constructed Scenario [95]

	X1	X2	X3	X4	X5	X6
X1						**-m13**
X2			**-m6**		**-m11**	
X3		**-m4**		**-m8**		
X4			**-m5**		**-m10**	
X5		**-m3**				**-m12**
X6	**-m1**	**-m2**		**+m7**	**-m9**	

In Table 7, X1, .., X7 represents the capabilities (NFRs). The indicators mi's in the table denotes the % improvement (if +ve)/degradation (if –ve) of capability identified by NFRs (column-wise) because of each NFR present row-wise. For example, +m7 signifies there is m7% improvement in the capability identified by X4 because of the presence of the capability X6 in the scenario. Conversely, -m1 signifies that there is m1% degradation in the capability identified by X1 because of the presence of the capability X6 in the scenario. Hence as we see, the association among the NFRs has been represented both in terms of sign and extent, which is as per DP 5 presented above.

Step 5: Adjustment of scenario NFR scores

Step 5 of the NFR prioritization framework indicates a computation step wherein an intermediate parameter (i.e. adjusted importance) is computed, initially at the NFR level, and finally at the scenario level. The parameter adjusted importance is interpreted in the framework as a measure of importance adjusted in presence of mutual association among NFRs (discussed in Step 4). This step is as per DP 6 specifications where the method to be used for prioritization is described.

Step 6: A heuristic for deciding which NFRs to be dropped from a scenario

The final step of the NFR prioritization framework specifies an algorithm for identifying NFRs to be dropped from a scenario. The algorithm on the basis of adjusted importance scores, first computes a candidate set consisting of all requirements which can be removed from the scenario. Then among the candidate NFRs, the NFRs with negative adjusted importance scores, and NFRs with positive adjusted importance scores along with six or more rating (on a 1-9 scale) on the 'Cost' parameter are dropped. Specification of the algorithm for removal of NFRs is in accordance with DP 6 descriptions. The final results of the NFR prioritization framework is a ranking of the retained NFRs on adjusted importance scores, and depicts a ranked representation of results (DP 8).

We present the design contributions of this NFR prioritization framework in Table 8 where we indicate how some of the design limitations mentioned earlier in the paper has been addressed through this artifact.

Table 8: Design contributions of the NFR prioritization artifact

Design Limitations	How Addressed
Methods of dealing with rank updates during requirements evolution	Step 6 of the prioritization framework presents an algorithm to remove requirements from the prioritized list based on the adjusted importance scores and the cost parameter ratings. In case of requirements addition or update to the scenario, the recomputed value of the adjusted importance scores and the associated cost parameter ratings of the changed requirements will decide if the prioritized list needs to be retained or updated.
Stakeholder considerations	The identification of the NRS (Step 1) was based on the preferences of the business representatives. Further, the association of the constructed scenario with the business objectives (Step 3) was based on the ratings provided by these stakeholders on the questionnaire items.
Requirement dependency issues	The mutual association of the NFRs was assessed based on a NFR association matrix in Step 4 of the framework. The association matrix captured the mutual dependency of the NFRs in the proposed scenario.

CONCLUSIONS

We set out to identify general DPs – that is, principles of form and function – which govern the design of a requirement prioritization artifact. We propose a framework to group the set of eight DPs representing a well-founded “checklist” into four categories viz. prioritization concerns, contextual factors, prioritization process design, and prioritization artifact. Although we cannot yet provide an expository instantiation for the DPs [40], we consider these principles to be a valuable contribution to the nascent theoretical body of knowledge on software requirement prioritization so as to facilitate comparison of existing artifacts and design of new artifacts. We are also convinced that the practical applicability of the requirement prioritization artifacts will benefit if the DPs are taken into account in the course of their design.

Our research provides the scope of extending existing requirement prioritization artifacts. The DPs enhances existing prescriptive research in the following manner: It is possible to check existing artifacts to identify DPs that are not included. These then become suggestive areas by which the existing artifact can be extended. This on one hand may contribute towards addressing the limitations of the concerned artifacts, and on the other may contribute towards designing newer artifacts combining the strengths of artifacts already existing. However, the inclusion of additional DPs may increase the level of complexity of the resultant design.

Our work is not without limitations. First, the proposed DPs are justified on the foundation of existing literature on requirement prioritization only. Its content may thus be biased with respect to those requirement prioritization artifacts that have been documented and published. In order to enhance the validity of the DPs proposed in this paper, there is a need to discuss this extensively with users and developers from both industry and academia. The Delphi technique could be used, for instance, in order to provide valuable insights into whether the set of principles that have been proposed here is complete, and which DPs are generally considered mandatory or optional. Second, it is still possible that certain requirement prioritization artifact has been accidentally overlooked, and hence not included in the coding and analysis stage. The artifacts that are accidentally omitted in this manner could serve as a basis to evaluate the DPS, and refine the same if necessary. Third, we cannot yet provide an expository instantiation to evaluate our DPs. A realistic implementation could demonstrate that the design is worth considering [40]. Finally, our research is not free from methodological limitations related to the sole use of content analysis of articles on requirement prioritization.

Despite these limitations, we believe that the DPs and the analysis presented in this paper constitute a valuable starting point for future research directed at developing a “design theory” on requirement prioritization to guide, inform and justify artifact design. Future research should collect data from additional sources like documentation or observations [105] to corroborate the results of our investigation. We hope this research note may encourage and motivate other scholars to join us in this area of academic inquiry and application.

REFERENCES

[1] Aasem, M., M. Ramzan, and A. Jaffar. "Analysis and optimization of software requirements prioritization techniques," in *International Conference on Information and Emerging Technologies (ICIET)*, 2010, pp. 1-6

[2] Achimugu, P., et al., "A systematic literature review of software requirements prioritization research," *Information and Software Technology*, Volume 56, Number 6, 2014, pp. 568-585

[3] Ahl, V., "An experimental comparison of five prioritization methods," *Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden*, 2005.

[4] Aho, A.V., *Data structures and algorithms*. 1983, Pearson Education India.

[5] Asghar, M.W., et al. "Maintainability-based requirements prioritization by using artifacts traceability and code metrics," in *17th European Conference on Software Maintenance and Reengineering (CSMR)* 2013, pp. 417-420

[6] Azar, J., R.K. Smith, and D. Cordes, "Value-oriented requirements prioritization in a small development organization," *Software, IEEE*, Volume 24, Number 1, 2007, pp. 32-37

[7] Bagheri, E., et al., *Stratified analytic hierarchy process: Prioritization and selection of software features*, in *Software Product Lines: Going Beyond*. 2010, Springer. p. 300-315

[8] Bebensee, T., I. van de Weerd, and S. Brinkkemper, *Binary priority list for prioritizing software requirements*, in *Requirements Engineering: Foundation for Software Quality*. 2010, Springer. p. 67-78

[9] Beck, K. and C. Andres, *Extreme programming explained: embrace change*. 2004, Addison-Wesley Professional.

[10] Beg, M.R., R.P. Verma, and A. Joshi. "Reduction in number of comparisons for requirement prioritization using B-Tree," in *IEEE International Advance Computing Conference, IACC 2009.*, 2009, pp. 340-344

[11] Benestad, H.C. and J.E. Hannay. "Does the prioritization technique affect stakeholders' selection of essential software product features?," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, 2012, pp. 261-270.

[12] Berander, P. and A. Andrews, *Requirements prioritization*, in *Engineering and managing software requirements*. 2005, Springer. p. 69-94.

[13] Berander, P. and P. Johansson, "Hierarchical cumulative voting (hcv) prioritization of requirements in hierarchies," *International Journal of Software Engineering and Knowledge Engineering*, Volume 16, Number 6, 2006, pp. 819-849

[14] Berander, P., K.A. Khan, and L. Lehtola. "Towards a research framework on requirements prioritization," in *Proceedings of Sixth Conference on Software Engineering Research and Practice in Sweden (SERPS'06)*, 2006, pp. 18-19.

[15] Berander, P. and M. Svahnberg, "Evaluating two ways of calculating priorities in requirements hierarchies - An experiment on hierarchical cumulative voting," *Journal of Systems and Software*, Volume 82, Number 5, 2009, pp. 836-850

[16] Berteig, M., *Methods of Prioritization, March 20, 2006 in Agile Advice online practitioners forum*. 2006.

[17] Bitzer, P., M. Söllner, and J.M. Leimeister, "Design Principles for High-Performance Blended Learning Services Delivery," *Business & Information Systems Engineering*, 2015, pp. 1-15.

[18] Brodie, L. and M. Woodman, *Prioritization of Stakeholder Value Using Metrics*, in *Evaluation of Novel Approaches to Software Engineering*. 2011, Springer. p. 74-88

[19] Büyüközkan, G., "Determining the mobile commerce user requirements using an analytic approach," *Computer Standards & Interfaces*, Volume 31, Number 1, 2009, pp. 144-152.

[20] Carlshamre, P., "Release planning in market-driven software product development: Provoking an understanding," *Requirements Engineering*, Volume 7, Number 3, 2002, pp. 139-151

[21] Cavanagh, S., "Content analysis: concepts, methods and applications," *Nurse researcher*, Volume 4, Number 3, 1997, pp. 5-13

[22] Cleland-Huang, J. and M. Denne. "Financially informed requirements prioritization," in *27th International Conference on ICSE 2005.* , 2005, pp. 710-711.

[23] Cronk, M.C. and E.P. Fitzgerald, "Understanding "IS business value": derivation of dimensions," *Logistics Information Management*, Volume 12, Number 1/2, 1999, pp. 40-49

[24] Crow, K. "Customer-focused development with QFD," 1994, pp. 839-839.

[25] Danesh, A.S., S.M. Mortazavi, and S.Y.S. Danesh. "Requirements prioritization in on-line banking systems: using value-oriented framework," in *International Conference on Computer Technology and Development*, 2009, pp. 158-161

[26] Doerr, J., et al. "Built-in user satisfaction-Feature appraisal and prioritization with AMUSE," in *15th*

IEEE International Requirements Engineering Conference, RE'07., 2007, pp. 101-110

[27] Downe-Wamboldt, B., "Content analysis: method, applications, and issues," *Health care for women international*, Volume 13, Number 3, 1992, pp. 313-321.

[28] Duan, C., et al., "Towards automated requirements prioritization and triage," *Requirements engineering*, Volume 14, Number 2, 2009, pp. 73-89

[29] Elahi, G. and E. Yu, "Comparing alternatives for analyzing requirements trade-offs-In the absence of numerical data," *Information and Software Technology*, Volume 54, Number 6, 2012, pp. 517-530

[30] Everitt, B.S., *Making sense of statistics in psychology: A second-level course*. 1996, Oxford University Press.

[31] Feather, M.S. and T. Menzies. "Converging on the optimal attainment of requirements," in *IEEE Joint International Conference on Requirements Engineering*, 2002, pp. 263-270

[32] Fraser, J. *Setting Priorities*. 2002 [cited; Available from: http://www.adaptivepath.com/ideas/essays/archives/000018.php.

[33] Galster, M. and A. Eberlein. "Facilitating software architecting by ranking requirements based on their impact on the architecture process," in *IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, 2011, pp. 232-240

[34] Gaur, V., A. Soni, and P. Bedi. "An application of multi-person decision-making model for negotiating and prioritizing requirements in agent-oriented paradigm," in *International Conference on Data Storage and Data Engineering (DSDE)*, 2010, pp. 164-168

[35] Gill, G. and A. Bhattacherjee, "Whom Are We Informing? Issues and Recommendations for MIS Research from an Informing Science Perspective," *MIS Quarterly*, Volume 33, Number 2, 2009, pp. 217-235.

[36] Gottesdiener, E. *At a Glance: Other Prioritization Methods*. 2007 [cited; Available from: http://www.ebgconsulting.com/Pubs/Articles/At%20a%20Glance-Other%20Prioritization%20Methods-supplement-EBG%20Consulting.pdf.

[37] Greer, D., D.W. Bustard, and T. Sunazuka. "Prioritisation of system changes using cost-benefit and risk assessments," 1999, pp. 180-187.

[38] Greer, D. and G. Ruhe, "Software release planning: an evolutionary and iterative approach," *Information and Software Technology*, Volume 46, Number 4, 2004, pp. 243-253

[39] Gregor, S. and A.R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS quarterly*, Volume 37, Number 2, 2013, pp. 337-356

[40] Gregor, S. and D. Jones, "The Anatomy of a Design Theory," *Journal of the Association for Information Systems*, Volume 8, Number 5, 2007, pp. 313-335.

[41] Gupta, V., D.S. Chauhan, and K. Dutta, "Incremental development & revolutions of E-learning software systems in education sector: a case study approach," *Human-centric Computing and Information Sciences*, Volume 3, Number 1, 2013, pp. 1-14.

[42] Herrmann, A. and M. Daneva. "Requirements prioritization based on benefit and cost prediction: An agenda for future research," in *16th IEEE International Requirements Engineering, RE'08.* , 2008, pp. 125-134

[43] Herrmann, A. and B. Paech. "Quality misuse," in *In Proceedings of 11th Int. Workshop on Requirements Engineering: Foundation of Software Quality*, 2005, pp. 193-199.

[44] Hevner, A.R., et al., "Design Science in Information Systems Research," *MIS Quarterly*, Volume 28, Number 1, 2004, pp. 75-105.

[45] Hooker, J.N., "Is design theory possible?," *Journal of Information Technology Theory and Application (JITTA)*, Volume 6, Number 2, 2004, pp. 8.

[46] Iqbal, M.A., A.M. Zaidi, and S. Murtaza. "A New Requirement Prioritization Model for Market Driven Products Using Analytical Hierarchical Process," in *International Conference on Data Storage and Data Engineering (DSDE)*, 2010, pp. 142-149

[47] Karlsson, J. "Software requirements prioritizing," in *Proceedings of the Second International Conference on Requirements Engineering*, 1996, pp. 110-116

[48] Karlsson, J., S. Olsson, and K. Ryan, "Improved practical support for large-scale requirements prioritising," *Requirements Engineering*, Volume 2, Number 1, 1997, pp. 51-60

[49] Karlsson, J. and K. Ryan, "A cost-value approach for prioritizing requirements," *Software, IEEE*, Volume 14, Number 5, 1997, pp. 67-74

[50] Karlsson, J., C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements," *Information and Software Technology*, Volume 39, Number 14, 1998, pp. 939-947.

[51] Keeney, R.L., *Decisions with multiple objectives: preferences and value trade-offs*. 1993, Cambridge University Press.

[52] Keeney, R.L., "Foundations for making smart decisions," *IIE Solutions*, Volume 31, Number 5, 1999, pp. 24-30.

[53] Kerkow, D., et al., "Elicitation and documentation of non-functional requirements for sociotechnical systems," *Requirements Engineering for Sociotechnical Systems*, *Idea Group, Inc*, 2004

[54] Kotonya, G. and I. Sommerville, *Requirements Engineering: Processes and Techniques.* . 1998, Chichester, UK: John Wiley and Sons.

[55] Koziolek, A. "Architecture-driven quality requirements prioritization," in *Requirements Engineering: Foundation for Software Quality*, 2012, pp. 15-19

[56] Kukreja, N. "Decision theoretic requirements prioritization: a two-step approach for sliding towards value realization," in *35th International Conference on Software Engineering (ICSE)*, , 2013, pp. 1465-1467

[57] Laurent, P., J. Cleland-Huang, and C. Duan. "Towards automated requirements triage," in *15th IEEE International Requirements Engineering Conference, RE'07.* , 2007, pp. 131-140

[58] Li, C., et al., *Integrated requirement selection and scheduling for the release planning of a software product*, in *Requirements Engineering: Foundation for Software Quality*. 2007, Springer. p. 93-108

[59] Li, Y.-L., K.-S. Chin, and X.-G. Luo, "Determining the final priority ratings of customer requirements in product planning by MDBM and BSC," *Expert systems with Applications*, Volume 39, Number 1, 2012, pp. 1243-1255

[60] Li, Y., et al., "An integrated method of rough set, Kano's model and AHP for rating customer requirements' final importance," *Expert Systems with Applications*, Volume 36, Number 3, 2009, pp. 7045-7053

[61] Liaskos, S., et al., "Representing and reasoning about preferences in requirements engineering," *Requirements Engineering*, Volume 16, Number 3, 2011, pp. 227-249

[62] Lim, S.L. and A. Finkelstein, "StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE Transactions on Software Engineering*, Volume 38, Number 3, 2012, pp. 707-735.

[63] Lima, D.C., et al., *A fuzzy approach to requirements prioritization*, in *Search Based Software Engineering*. 2011, Springer. p. 64-69

[64] Liu, X., et al. "Priority assessment of software requirements from multiple perspectives," 2004, pp. 410-415

[65] Logue, K. and K. McDaid. "Handling uncertainty in agile requirement prioritization and scheduling using statistical simulation," in *AGILE'08. Conference*, 2008, pp. 73-82

[66] Martinez, A.B.B., et al., "Composing requirements specifications from multiple prioritized sources," *Requirements Engineering*, Volume 13, Number 3, 2008, pp. 187-206

[67] Mendizabal, O.M., M. Spier, and R. Saad. "Log-based approach for performance requirements elicitation and prioritization," in *20th IEEE International Requirements Engineering Conference (RE)*, 2012, pp. 297-302

[68] Moisiadis, F. "The fundamentals of prioritising requirements," in *Proceedings of the systems engineering, test and evaluation conference (SETE'2002)*, 2002, pp. 108-119.

[69] Mu, K., et al., "Handling inconsistency in distributed software requirements specifications based on prioritized merging," *Fundamenta Informaticae*, Volume 91, Number 3, 2009, pp. 631-670.

[70] Nas, T.F., *Cost-benefit analysis: Theory and application*. 1996, Sage.

[71] Negräo, E.C. and E.M. Guerra. "A case study for prioritizing features in environments with multiple stakeholders," in *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, 2011, pp. 131-140

[72] Otero, C.E., et al. "A quality-based requirement prioritization framework using binary inputs," in *Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation (AMS)*, 2010, pp. 187-192

[73] Palma, F., A. Susi, and P. Tonella. "Using an SMT solver for interactive requirements prioritization," in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, 2011, pp. 48-58

[74] Park, J.-W., D. Port, and B. Boehm. "Supporting distributed collaborative prioritization," in *Sixth Asia Pacific Software Engineering Conference, (APSEC'99).* , 1999, pp. 560-563

[75] Patton, J. "Finding the forest in the trees," in *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 2005, pp. 266-274

[76] Pergher, M. and B. Rossi. "Requirements prioritization in software engineering: A systematic mapping study," in *3rd International Workshop on Empirical Requirements Engineering*, 2013, pp. 40-44.

[77] Perini, A., A. Susi, and P. Avesani, "A machine learning approach to software requirements prioritization," *Software Engineering, IEEE Transactions on*, Volume 39, Number 4, 2013, pp. 445-461

[78] Pöppelbuß, J. and M. Röglinger. "What makes a useful maturity model? A framework of general design principles for maturity models and its demonstration in business process management," in *European Conference on Information Systems*, 2011,
[79] Ramzan, M., et al. "Value based fuzzy requirement prioritization and its evaluation framework," in *Fourth International Conference on Innovative Computing, Information and Control (ICICIC),*, 2009, pp. 1464-1468
[80] Regnell, B., et al., "An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software," *Requirements Engineering*, Volume 6, Number 1, 2001, pp. 51-62.
[81] Regnell, B., M. Höst, and R.B. Svensson, *A quality performance model for cost-benefit analysis of non-functional requirements applied to the mobile handset domain*, in *Requirements Engineering: Foundation for Software Quality*. 2007, Springer. p. 277-291
[82] Regnell, B.r., et al., "An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software," *Requirements Engineering*, Volume 6, Number 1, 2001, pp. 51-62.
[83] Ribeiro, R.A., et al., "Hybrid assessment method for software engineering decisions," *Decision Support Systems*, Volume 51, Number 1, 2011, pp. 208-219
[84] Roy, B., *Multicriteria methodology for decision aiding*. Volume 12, 1996, Springer.
[85] Ruhe, G., A. Eberlein, and D. Pfahl, "Trade-off analysis for requirements selection," *International Journal of Software Engineering and Knowledge Engineering*, Volume 13, Number 4, 2003, pp. 345-366
[86] Schwaber, K.X., *Agile project management with Scrum*. 2004, O'Reilly Media, Inc.
[87] Sen, C.G. and H. Baraçlı, "Fuzzy quality function deployment based methodology for acquiring enterprise software selection requirements," *Expert Systems with Applications*, Volume 37, Number 4, 2010, pp. 3415-3426
[88] Shao, P. "Sample selection: an algorithm for requirements prioritization," in *46th Annual Southeast Regional Conference on Software Engineering*, 2008, pp. 525-526
[89] Simon, H.A., *The sciences of the artificial*. 1996, MIT press, Cambridge.
[90] Smiley, K., et al. "Architectural requirements prioritization and analysis applied to software technology evaluation," in *2009 ACM symposium on Applied Computing*, 2009, pp. 397-398
[91] Sobczak, A. and D.M. Berry, "Distributed priority ranking of strategic preliminary requirements for management information systems in economic organizations," *Information and Software technology*, Volume 49, Number 9, 2007, pp. 960-984
[92] Sommerville, I. and P. Sawyer, "Requirements engineering: a good practice guide John Wiley & Sons," *Chichester, England*, 1997.
[93] Sun, Y. and X.F. Liu, "Business-oriented software process improvement based on CMMI using QFD," *Information and software technology*, Volume 52, Number 1, 2010, pp. 79-91
[94] Tabaka, J.X., *Collaboration explained: facilitation skills for software project leaders*. 2006, Pearson Education.
[95] Thakurta, R., "A framework for prioritization of quality requirements for inclusion in a software project," *Software Quality Journal*, 2013, pp. 573-597.
[96] Tl, S., *The analytic hierarchy process*. 1980, McGraw-Hill, New York.
[97] Tonella, P., A. Susi, and F. Palma, "Interactive requirements prioritization using a genetic algorithm," *Information and Software Technology*, Volume 55, Number 1, 2013, pp. 173-187
[98] Tonella, P., A. Susi, and F. Palma. "Using interactive GA for requirements prioritization," in *Second International Symposium on Search Based Software Engineering (SSBSE)*, 2010, pp. 57-66
[99] Trienekens, J.J.M., R.J. Kusters, and D.C. Brussel, "Quality specification and metrication, results from a case-study in a mission-critical software domain," *Software Quality Journal*, Volume 18, Number 4, 2010, pp. 469-490
[100] Veldhuis, M.M.O., *Defining and validating user interface design guidelines for professional mobile applications*. 2012, University of Twente, Netherlands.
[101] Voola, P. and A.V. Babu. "Interval evidential reasoning algorithm for requirements prioritization," in *International Conference on Information Systems Design and Intelligent Applications*, Visakhapatnam, India, 2012, pp. 915-922
[102] Waters, K. *Prioritization using MoSCoW*. Agile Planning 2009 January 12, 2009 [cited; Available from: http://www.allaboutagile.com/prioritization-using-moscow/.
[103] Wiegers, K., "First things first: prioritizing requirements," *Software Development*, Volume 7, Number 9, 1999, pp. 24-30.

[104] Wolfswinkel, J.F., E. Furtmueller, and C.P.M. Wilderom, "Using grounded theory as a method for rigorously reviewing literature," *European Journal of Information Systems*, Volume 22, Number 1, 2011, pp. 45-55.

[105] Yin, R.K., *Case study research: Design and methods*. Vol. 5. 2009, Sage publications, INC.

[106] Yu, D., et al. "An Algorithm for Priority Ranking of Individualized Functional Requirements in Networked Software," in *Seventh International Conference on Grid and Cooperative Computing, GCC'08.* , 2008, pp. 701-706

[107] Zhang, H., L. Liu, and T. Li, "Designing IT systems according to environmental settings: A strategic analysis framework," *The Journal of Strategic Information Systems*, Volume 20, Number 1, 2011, pp. 80-95

AUTHOR BIOGRAPHY

Rahul Thakurta is an Associate Professor of Information Systems at Xavier Institute of Management Bhubaneswar (XIMB), India. He received his Fellowship in Management Information Systems from Indian Institute of Management Calcutta. He teaches courses pertaining to Information Systems (IS) and Operations Management disciplines. His broad research interests include areas like planning and implementation of IS infrastructures, management of software ventures, design science, and technology adoption and diffusion. He has been the recipient of Infosys Research Fellowship and DAAD Short- Term Research Scholarship. He is also the Managing Editor of *Research World* and *Vilakshan* journals published by XIMB.

APPENDIX

Table 9: Artifacts on Requirement Prioritization

Artifact Type	Contributions
Constructs	<Reasoning notation on optional and preference requirements>* [61]
Model	Mathematical programming techniques [58]; Multi-Person Decision-Making Model [34]; Quality Performance Model [81]; <Model for requirement prioritization using AHP>* [46]; <Integer linear programming models>* [58]
Method	Analytic Hierarchy Process (AHP) [96]; Attribute Hierarchy-based Evaluation of Architectural Designs (AHEAD) method [90]; Requirements Triage [57]; Binary search tree [3]; Bubble sort [4]; Bucketing requirements [75]; CCR technique [41]; Cost benefit analysis [70]; Cost-Value Approach (CVA) [49]; Cumulative Voting (Hundred-dollar test) [82]; Evolutionary and Iterative (EVOLVE) approach [38]; Hierarchy AHP [50]; Hierarchical Cumulative Voting (HCV) [13]; Hybrid Assessment Method (HAM) [83]; Incremental funding method (IFM) [22]; Interactive Genetic Algorithm (IGA) [97]; Interval Evidential Reasoning (IER) Algorithm [101]; Minimal spanning tree matrix [50]; MoSCoW [102]; Multi-Attribute Utility Theory (MAUT) [51]; Multi-voting system [94]; Numerical Assignment (Grouping) [47]; Outranking [84]; [Pair-wise analysis, Weighted criteria analysis, Dot voting] [36]; Ping Pong Balls [86]; Planning game [9]; Prioritization Matrix [103]; Priority groups (also called grouping/numeral) [50]; Quality Functional Deployment (QFD) [24]; WinWin [85]; Ranking [12]; Ranking based on product definition [32]; Round-the-group prioritization [16]; Sample Selection [88]; StakeRare [62]; Top-ten requirements [42]; Weighting methods [52]; < AHP based approach>* [47]; <Agile Technique>* [71]; <Algorithm for ranking priority of individualized functional requirements dynamically>* [106]; <Approach based on ISO9126 and AHP>* [99]; <Approach that Augments HCV>* [15]; <Approach using B-Tree>* [10]; <Architecture driven method>* [55]; <Data-mining and machine learning based technique>* [28]; <Extension of impact estimation prioritization method>* [18]; <Heuristic algorithm>* [29]; <Integrated method of rough set, Kano's model and AHP>* [60]; <Approach integrates minimal deviation based method (MDBM), balanced scorecard (BSC), analytic hierarchy process (AHP), scale method>* [59]; <Intelligent fuzzy logic based technique>* [79]; <Interactive Approach based on Satisfiability Modulo Theory (SMT) techniques and pairwise comparisons>* [73]; <Interactive Optimization of Requirements>* [98]; <Log based approach>* [67]; <Method for ranking requirements>* [33]; <Multi-objective evolutionary optimisation algorithms>* [107]; <Prioritization approach based on use cases and quality models>* [53]; <Priority-based approach> [66]; <Risk based Requirement Prioritization>* [43]; <Using standard hierarchical clustering algorithm>* [57]
Method, Tool	AMUSE (appraisal and measurement of user satisfaction) [26]; Case-Based Ranking (CBRank) [77]; SERUM (Software Engineering Risk: Understanding and Management) [37]; Stratified Analytic Hierarchy process (S-AHP) [7]; Binary Priority List (BPL) [8]; [Pairwise comparisons & ranking, Simple Dropdown, Drag into bins, Sortable table] [11]; Release Planner Prototype (RPP) [20]; <Improvement of an existing cost-value approach>* [48]; <Requirement optimization approach>* [31]
Methodology	Fuzzy AHP [19]; <Fuzzy quality function deployment approach>* [87]; <Statistical methodology that allows for uncertainty>* [65]
Methodology, Tool	<Distributed priority ranking instrument>* [91]
Framework	Prioritized merging-based framework [69]; Requirement Prioritization Tool (RPT) [68]; Value-oriented prioritization (VOP) [6]; <Formal framework using fuzzy logic>* [63]; <Framework incorporating interperspective relationships>* [64]; <Framework combining existing techniques and approaches>* [1]; <Quality-Based Requirement Prioritization Framework>* [72]; <SPI framework based on the CMMI using QFD >* [93]; <NFR Prioritization Framework>* [95]; <Value oriented Framework>* [25]
Framework, Tool	Technique of Ordered Preference by Similarity to Ideal Solution (TOPSIS) [56]; Requirement Prioritization Tool (RPT) [68]
System, Tool	Distributed Collaboration Priorities Tool (DCPT) [74]
Tool	<Based on artifact traceability information>* [5]

* Exact names of these prioritization artifacts have not been specified by the authors

Table 10: List of Attributes

Grouping	Sub-Grouping	List of Factors
Internal	Requirement Attributes	Requirement Type, Requirement Risk, Requirement Complexity, Requirement Importance/Priority/Preference, Requirement Volatility / Stability, Requirement Performance, Requirement Value, Requirement Cost, Requirement Criticality, Requirement Desirability, Requirement Benefit, Requirement Hierarchy, Quality of Requirements, Importance of Views.
	Project Attributes	Development Cost, Development Benefit, Project Duration (Delivery Date), Effort, Project Size, Resources (Competencies), Difficulty, Project Costs, Project Expectations, Project Influence, Project Constraints, Project Contexts, Application Domain
	Process Attributes	Process Model, Development Phase, Requirement Dependencies, Requirement Trade-offs, Implementation Dependencies, Artifact Traceability, Repetitiveness, Legal Complexity / Mandate, Response Time, Regression Count, Software Architecture & Relevant Attributes, Requirement Baselining.
	Product-Service Attributes	Technical debt, System Impact, Requirement Usability, Frequency of Use, Re-use Frequency, Quality Attributes / Characteristics, Penalty, Harm Avoidance
External	Subject Attributes	Requirement's Issuer, Requirement Hits, Number of Unique Users, Multiple Groups, Stakeholder Roles, Stakeholder's Relative Importance, Stakeholder Preferences, Stakeholder Perspectives
	Prioritization Environment	Business Values, Business Goals, Business Processes, Business Functions, Organization Size, Type of Market, Time to Market, Competitive Benchmarking

Table 11: Participant Feedback Form used in Evaluation

Please refer to the description of the design principles and how these relate to designing the prioritization method: Top ten requirements provided to you in a separate attachment. Considering the design principles that have been listed, please provide your views with respect to the following questions (*you can refer to a design principle by its identifier, i.e. DP 1 if you are referring to the first design principle and so on*)

1. Please review the individual design principle statements. Considering each, please provide your observations on the clarity, accuracy, and completeness of the statements along with suggestions for revising the principles as applicable.

2. Please let us know if you want to restructure (i.e. merge, split, extend, update, etc.) any subset of the design principles and please provide your arguments for the same.

3. Please let us know if you agree or disagree with the explanation of the design principles. In case you disagree please let us know your arguments

4. Please relate the design principles to any experiences on prioritization you had in the past.

5. Any other comments or suggestions