A SPECIFICATION FOR DESIGNING REQUIREMENT PRIORITIZATION ARTIFACTS

Rahul Thakurta, Xavier Institute of Management, Xavier University, Bhubaneswar, India.
rahul@ximb.ac.in

Abstract

The importance of prioritizing requirements stems from the fact that not all requirements can usually be met with available time and resource constraints. Although several papers have been published in this domain, they mainly focus on descriptive research endeavors to suggest different requirement prioritization approaches. Prescriptive research dealing with design science for a systematic and holistic understanding of the prioritization process is still scarce. The gap motivates our research, which aims at arriving at a set of design principles that explains the form and function of software requirement prioritization artifacts. We resort to a non-experimental approach using content analysis to identify and analyze articles on requirement prioritization published up to 2009 in order to arrive at the set of initial design principles. This subsequently is evaluated based on expert feedbacks. We close the paper by indicating our research continuation plans, and highlighting issues for future considerations.

Keywords: Requirement prioritization, Design principles, Content analysis.

1 INTRODUCTION

The importance of requirement prioritization in software engineering discipline has been very well acknowledged (Herrmann and Daneva 2008). Requirements provide the description of the system, its behaviour, application domain information, system constraints, specifications and attributes (Kotonya and Sommerville 1998). The importance of prioritizing requirements stems from the fact that not all requirements can usually be met with available time and resource constraints. The emphasis on 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. Some of the design limitations of these existing requirement prioritization artifacts are scalability restrictions, stakeholder considerations, requirement dependency considerations, etc. (Achimugu et al. 2014).

The different requirement prioritization artifacts are indeed valuable contributions to the domain of research under investigation. 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) (Gregor and Jones 2007) that govern the design of the requirement prioritization artifacts. Therefore, we argue that research must focus on gaining better insights on the fundamentals that govern the design of these requirement prioritization artifacts in order to apply them successfully in appropriate contexts.

Our research raises the question of what are the principles governing the design of a requirement prioritisation artifact. The work is motivated by the identified design limitations, and awareness that existing literature on requirement prioritisation has not yet attempted to arrive at the set of DPs and related classification of requirement prioritisation artifacts. Hence, we carry out a systematic review of requirement prioritisation artifacts in order to identify the common structures characterising the form and function of the existing artifacts in the process of deriving the set of seven DPs. These are subsequently validated based on expert reviews by six practitioners, ensuring their validity and applicability. 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.

2 BACKGROUND

Design science research (DSR) as a problem-solving paradigm (Hevner et al. 2004) strives to create innovative artefacts as a solution to problems faced by stakeholders in different domains (Gregor and Jones 2007). March and Smith (1995) introduced four types of IT artifacts viz. constructs, methods, models and instantiations, as the outcome of a DSR endeavour. DSR contributions in the domain of requirement prioritization have been specifications of different artifacts towards prioritization of the candidate requirement set. Accepting that design is a creative process, there has been recognition of the fact that a general design method cannot be formalised (Hooker 2004). This has been the foundation behind design of these different artifacts with each having various capabilities and limitations (Achimugu et al. 2014). For example, as part of the initial attempts, the numerical assignment (Karlsson 1996) specifies a method of grouping requirements in specified categories based on stakeholder ratings. This artifact is limited by the qualitative interpretation of the categories to the participating stakeholders. Among the later attempts, Doerr et al (2007) present the AMUSE (appraisal and measurement of user satisfaction) method and a tool support (instantiation) towards prioritizing project’s features based on user satisfaction. This artifact differs from the former in its use of quantitative assessment procedure in the prioritization process. These evidences point at the current considerations adopted in designing requirement prioritisation artifacts (e.g. each driven by a defined objective, etc.) and 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 prioritisation artifact, which we address in this research.

3 RESEARCH PROCESS

3.1 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 proposed in the 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 manuscript.

3.2 Data Collection

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 channels listed in Table 1.

Journal Publications	**Conference/Symposium Proceedings**
ACM Transactions on Software Engineering Methodologies	Agile Conference
Communications of the ACM	Design Science Research in Information Systems and Technologies
Decision Support Systems	Empirical Software Engineering and Measurement
Empirical Software Engineering	European Conference on Software Maintenance and Reengineering
Expert Systems with Applications	Euromicro Conference on Software Engineering and Advanced Applications
IEEE Software	Conference on Software Engineering Education and Training
IEEE Transactions on Software Engineering	Conference on Systems Engineering Research
IET Software	IEEE/ACM International Conference on Automated Software Engineering
Information and Software Technology	IEEE International Conference and Workshops on Engineering of Computer Based Systems
International Journal of Software Engineering and Knowledge Engineering	IEEE International Requirements Engineering Conference
Journal of Systems and Software	International Conference on Software Engineering
Journal of Systems Architecture	International/European Conference on Information Systems
Requirements Engineering	Requirements Engineering – Foundation of Software Quality
Software Process Improvement and Practice	
Software Quality Journal	

Table 1. Search Outlets

In our 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. In this research-in-progress paper, we present the synthesis of search results up to 2009 given our progress status in this research.

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 70 articles for full-text review. These articles were written in English only and included both qualitative and quantitative research published in 2009 or earlier. In Table 2, we include a list of some of the common artifacts on requirement prioritization that we could cover in our systematic review. We classified the list in terms of the type of artifact like methods, models, etc. We are unable to describe each of these artifact because of space constraint, and hence advice the reader to look into the relevant references for further information on these artifacts.

Artifact Type	**Contributions**
Method	Numerical Assignment (Grouping) (Karlsson 1996); Top-ten requirements (Herrmann and Daneva 2008); Round-the-group prioritization (Berteig 2006); Ping Pong Balls (Schwaber 2004); Cumulative Voting (Hundred-dollar test) (Regnell et al. 2001); Weighting methods (Keeney 1999); Outranking (Roy 1996); Minimal spanning tree matrix (Karlsson et al. 1998); Bubblesort (Aho 1983); Binary search tree (Ahl 2005); Cost benefit analysis (Nas 1996); Priority groups (also called grouping/numeral) (Karlsson et al. 1998); Planning game (Beck and Andres 2004); Ranking based on product definition (Fraser 2002); Automated Requirements Triage (Laurent et al. 2007); Prioritization Matrix (Wiegers 1999); Quality Functional Deployment (QFD) (Crow 1994); Multi-Attribute Utility Theory (MAUT) (Keeney 1993); Analytic Hierarchy Process (AHP) (Tl 1980); EVOLVE (Greer and Ruhe 2004); <Priority-based approach> (Martinez et al. 2008); MoSCoW (Waters 2009); Ranking (Berander and Andrews 2005); Multi-voting system (Tabaka 2006); Hierarchy AHP (Karlsson et al. 1998); Cost-Value Approach (CVA) (Karlsson and Ryan 1997); WinWin (Ruhe et al. 2003); Hierarchical Cumulative Voting (HCV) (Berander and Johansson 2006)
Method, Tool	Software Engineering Risk: Understanding and Management (SERUM) (Greer et al. 1999) AMUSE (Doerr et al. 2007)
Model	Mathematical programming techniques (Li et al. 2007); Quality Performance Model (Regnell et al. 2007)
Framework	Value-oriented prioritization (VOP) (Azar et al. 2007); Prioritized merging-based framework (Mu et al. 2009)
Framework, Tool	Requirement Prioritization Tool (RPT) (Moisiadis 2002)

Table 2. Artifacts on Requirement Prioritization

3.3 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 (Downe-Wamboldt 1992). We followed the guidance offered in Wolfswinkel et al. (2011) 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, prioritisation description, prioritisation 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 signifies the measure of shared, rather than individual, understanding of the content and this is referred in the literature as inter-coder reliability (Cavanagh 1997). A couple of iterations involving revision of the codes was required until the final value of Cohen's k (0.82) was found to be in the acceptable range (Everitt 1996). 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.

In the next stage, the categories which emerged from the first stage were selectively merged to arrive at the set of 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 in a workshop format. The process was iterative as it needed revisiting the meaning of the sub-categories which we had defined, going back to the articles to identify how the sub-categories informed the prioritization artifact, identifying conceptual coherence, etc. These patterns are finally presented as prescriptive state-ments in the DPs that we address in the next section.

4 DESIGN PRINCIPLES FOR REQUIREMENT PRIORITIZATION ARTIFACTS

A synthesis of the coding results leads us to the following seven DPs presented in Table 3. We do not require every requirement prioritization artifact presently available to meet all DPs. We also do not make any claim for completeness, as the results are based on synthesis of findings up to the year 2009. Hence artifacts published or proposed beyond 2009 might include constructs, methods, and models that are not included in this review. 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.

Identifier	Design Principle (DP)
DP 1	Specify the nature of the artifact
DP 2	Determine the prioritization perspective
DP 3	Determine the over-arching objectives guiding the prioritization process
DP 4	Identify candidate requirements for prioritization
DP 5	Establish a procedure to handle associations among interacting entities
DP 6	Establish a method to carry out prioritization
DP 7	Develop a representation scheme to communicate the final results

Table 3. DPs for Requirement Prioritization Artifact

DP 1: Specify the nature of the artifact

The first step towards designing a requirement prioritization artifact is to decide on the nature of the planned artifact. The typical artifacts can be designing a requirement prioritization model, method or framework. These artifacts may be represented conceptually, for example, a blueprint of the artifact is presented in a suitable format. These artifacts may also be represented physically, for example, as an instantiation within an application or a tool.

DP 2: 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 2 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 perspctive, we also address a design limitation (e.g. stakeholder considerations) in DP 2.

DP 3: Determine the over-arching objectives guiding the prioritization process

The third step towards designing a requirement prioritization artifact is concerned with specifying the objectives guiding the prioritization process. The prioritization process might be linked to identifying the requirements to be considered for implementation in different phases of the project, arriving at a measure of benefit of the requirements considered for implementation, deriving 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. 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.

DP 4: Identify candidate requirements for prioritization

DP 4 specifies identifying the requirements that are candidate for prioritization. These form the master list of requirements considered for prioritization. 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 3).

DP 5: Establish a procedure 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. DP 5 prescribes the need to consider possibility of associations in its prioritization process and thereby address a design limitation (e.g. requirement dependency considerations). 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 (Hevner et al. 2004), a design science contribution at this level is the novelty of the method employed to accomplish the prioritization, and achieve the intended objective. The prioritization method may specify grouping of requirements (Moisiadis 2002), introduce pair-wise comparison of requirements (Karlsson et al. 1998), specify actions in order to handle requirement addition and update during prioritization process (Greer and Ruhe 2004), specify the nature of the computation process (i.e. single-step, iterative) (Ruhe et al. 2003), address scalability considerations etc. The results of DP 6 are prioritized indicators of the candidate requirement set as per the specified objectives.

DP 7: Develop a representation scheme to communicate the final results

The last DP is about 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 as per the results obtained from the prioritization artifact, may be sorted in ordinal scales in specific order (i.e. ascending, descending), or may be qualitatively represented and interpreted. To include some examples, the numerical assignment approach (Karlsson 1996) presents results by classifying prioritization output into three predefined categories viz. high, medium, and low. Top ten requirements (Herrmann and Daneva 2008) provide a ranked list of requirements in terms of requirement importance to the stakeholders.

5 PRELIMINARY EVALUATION OF THE DESIGN PRINCIPLES

The preliminary set of DPs that are described above were subjected to expert evaluation 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 (Gill and Bhattacherjee 2009). Any refinement of DPs at this stage can be checked against the findings from the systematic review of articles published since 2009 as we continue with our research.

The evaluation was carried out by sharing the DPs and their descriptions and a feedback form with the practitioners and gathering their views. 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 DPs, aspects with which they disagree, and suggestions on extensions and modifications of the DPs and the descriptions. Overall, six practitioners were involved who were males, aged 45–55 and working at senior management positions in IT organisations. 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 4, some illustrative data on the feedbacks received during this evaluation stage, and the implication on the initial set of DPs.

DP	Practitioner Feedback	Implications
DP 1	*"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 1. Hence no changes were required.
DP 2	*"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 handle conflicts between different prioritization perspectives. Future artifacts on requirement prioritization may incorporate this consideration in their design.
DP 3	"*Generally the focus is on ranking of requirements (functional)*" [Senior Business Architect] "*Presence of multiple objectives governing the*	The observation and the suggestion are related to the overarching objective governing the prioritization process. The formulation of DP 3 includes both these considerations and hence no

	prioritization may be explored ” <example provided> [Product Manager]	changes were required.
DP 6	“*Scope creep is a problem for us, and good so see the aspect included in the design principles*” [Product Manager]	The comment relates to the consideration of requirement addition and update included as part of DP 6. No changes required.
DP 7	“*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 7 specifications.

Table 4. Expert Feedbacks on the Design Principles

6 RESEARCH CONTINUATION

We have presented and described a set of validated DPs for requirement prioritization artifacts based on a systematic review of relevant literature up to 2009. In continuation, we intend to finalize the set of DPs based on synthesis of literature from 2010 to 2013. We plan to evaluate the final set of DPs in two rounds. In the first round, we will again involve expert feedbacks wherein any changes and additions to the first set of DPs (Table 3) will be discussed. In the second round, any further revision to the DPs will be validated based on comparisons with requirement prioritization artifacts that have been published in 2014. This will ensure the practical utility of the DPs towards guiding the design of requirement prioritization artifacts. We will be using Google scholar (http://scholar.google.co.in/) to identify a couple of such artifacts suitable for the purpose.

7 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 set of seven DPs representing a well-founded “checklist” based on review of requirement prioritization artifacts included in our research. The set of DPs allows one to compare existing artifacts on requirement prioritization and facilitate design of new artifacts. Although we cannot yet provide an expository instantiation for it (Gregor and Jones 2007), we consider the initial set of DPs to be a valuable contribution to the nascent theoretical body of knowledge on requirement prioritization by providing starting points for further exploration. 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.

The work presented here offers scope of extending requirement prioritization artifacts that have been covered in our review. It is possible to check these artifacts to identify DPs that are not included. These can then become suggestive areas by which the concerned 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.

To correctly interpret the results of our work, some limitations need to be taken into account. First, the DPs are justified on the foundation of approaches that has been discussed above. Its content may thus be biased with respect to those requirement prioritization artifacts that have been published. In order to enhance the validity of the DPs proposed in this article, there is also a need to discuss this extensively with users and developers from both industry and academia. Second, given the nature of literature inclusion for the study purpose, it is still possible to miss out artifacts that have been published in some other channels in the concerned timeframe. These artifacts could also be used evaluate the final set of DPs, and refine the same if necessary. Third, we cannot yet provide an expository instantiation to evaluate our DPs. A realistic imple-mentation could demonstrate that the design is worth considering (Gregor and Jones 2007). Finally, methodological limitations might arise from criticism of the qualitative method (Committee et al.; Flyvbjerg 2006). While we sought to address some concerns by relying on an established approach to qualitative analysis, our research has methodological limitations related to the sole use of content analysis of articles on requirement prioritization.

After completion of our research, the final set of DPs can constitute a valuable starting point for future inquiry directed at developing a "design theory" on requirement prioritization to guide, inform and justify artifact design. We hope this article may encourage and motivate academicians and practitioners to join us in this area of scientific inquiry and application.

References

Achimugu, P., Selamat, A., Ibrahim, R., and Mahrin, M.N.r. (2014). A Systematic Literature Review of Software Requirements Prioritization Research, Information and Software Technology (56:6), pp. 568-585.

Ahl, V. (2005). An Experimental Comparison of Five Prioritization Methods, Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden).

Aho, A.V. (1983). Data Structures and Algorithms. Pearson Education India.

Azar, J., Smith, R.K., and Cordes, D. (2007). Value-Oriented Requirements Prioritization in a Small Development Organization, Software, IEEE (24:1), pp. 32-37.

Baskerville, R. (2008). What Design Science Is Not, European Journal of Information Systems (17:5), pp. 441-443.

Beck, K., and Andres, C. (2004). Extreme Programming Explained: Embrace Change. Addison-Wesley Professional.

Berander, P., and Andrews, A. (2005). Requirements Prioritization, in Engineering and Managing Software Requirements. Springer, pp. 69-94.

Berander, P., and Johansson, P. (2006). Hierarchical Cumulative Voting (Hcv) Prioritization of Requirements in Hierarchies, International Journal of Software Engineering and Knowledge Engineering (16:06), pp. 819-849

Berteig, M. (2006). Methods of Prioritization, March 20, 2006 in Agile Advice Online Practitioners Forum.

Cavanagh, S. (1997). Content Analysis: Concepts, Methods and Applications, Nurse researcher (4:3), pp. 5-13.

Committee, I.C.S.S.E.S., Electronics Engineers, I., and Board, I.-S.S. (1998). IEEE Recommended Practice for Software Requirements Specifications: Approved 25 June 1998. IEEE.

Crow, K. (1994). Customer-Focused Development with Qfd, pp. 839-839.

Doerr, J., Hartkopf, S., Kerkow, D., Landmann, D., and Amthor, P. (2007). Built-in User Satisfaction-Feature Appraisal and Prioritization with Amuse, IEEE, pp. 101-110.

Downe-Wamboldt, B. (1992). Content Analysis: Method, Applications, and Issues, Health care for women international (13:3), pp. 313-321.

Downeâ-Wamboldt, B. (1992). Content Analysis: Method, Applications, and Issues, Health care for women international (13:3), pp. 313-321.

Everitt, B.S. (1996). Making Sense of Statistics in Psychology: A Second-Level Course. Oxford University Press.

Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research, Qualitative inquiry (12:2), pp. 219-245.

Fraser, J. (2002). Setting Priorities. [online] http://www.adaptivepath.net/ideas/e000018/ (accessed 6 May 2014)

Gill, G., and Bhattacherjee, A. (2009). Whom Are We Informing? Issues and Recommendations for Mis Research from an Informing Science Perspective, MIS Quarterly (33:2), pp. 217-235.

Greer, D., Bustard, D.W., and Sunazuka, T. (1999). Prioritisation of System Changes Using Cost-Benefit and Risk Assessments, IEEE, pp. 180-187.

Greer, D., and Ruhe, G. (2004). Software Release Planning: An Evolutionary and Iterative Approach, Information and Software Technology (46:4), pp. 243-253.

Gregor, S., and Jones, D. (2007). The Anatomy of a Design Theory, Journal of the Association for Information Systems (8:5), pp. 313-335.
Herrmann, A., and Daneva, M. (2008). Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research, IEEE, pp. 125-134.
Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research, MIS Quarterly (28:1), pp. 75-105.
Hooker, J.N. (2004). Is Design Theory Possible? Journal of Information Technology Theory and Application (JITTA) (6:2), pp. 8.
Karlsson, J. (1996). Software Requirements Prioritizing, Proceedings of the Second International Conference on Requirements Engineering: IEEE, pp. 110-116
Karlsson, J., and Ryan, K. (1997). A Cost-Value Approach for Prioritizing Requirements, Software, IEEE (14:5), pp. 67-74.
Karlsson, J., Wohlin, C., and Regnell, B. (1998). An Evaluation of Methods for Prioritizing Software Requirements, Information and Software Technology (39:14), pp. 939-947.
Keeney, R.L. (1993). Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Cambridge University Press.
Keeney, R.L. (1999). Foundations for Making Smart Decisions, IIE Solutions (31:5), May, pp. 24-30.
Kotonya, G., and Sommerville, I. (1998). Requirements Engineering: Processes and Techniques. . Chichester, UK: John Wiley and Sons.
Laurent, P., Cleland-Huang, J., and Duan, C. (2007). Towards Automated Requirements Triage, IEEE, pp. 131-140.
Li, C., Van Den Akker, J.M., Brinkkemper, S., and Diepen, G. (2007). Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product, in Requirements Engineering: Foundation for Software Quality. Springer, pp. 93-108.
Martinez, A.B.B., Arias, J.J.P., Vilas, A.F., Duque, J.G., Nores, M.L., Redondo, R.P.D., and Fernandez, Y.B. (2008). Composing Requirements Specifications from Multiple Prioritized Sources, Requirements Engineering (13:3), pp. 187-206.
Moisiadis, F. (2002). The Fundamentals of Prioritising Requirements, Citeseer, pp. 108-119.
Mu, K., Liu, W., Jin, Z., Lu, R., Yue, A., and Bell, D. (2009). Handling Inconsistency in Distributed Software Requirements Specifications Based on Prioritized Merging, Fundamenta Informaticae (91:3), pp. 631-670.
Nas, T.F. (1996). Cost-Benefit Analysis: Theory and Application. Sage.
Regnell, B.r., Höst, M., och Dag, J.N., Beremark, P., and Hjelm, T. (2001). An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software, Requirements Engineering (6:1), pp. 51-62.
Regnell, B.r., Höst, M., and Svensson, R.B. (2007). 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. Springer, pp. 277-291.
Roy, B. (1996). Multicriteria Methodology for Decision Aiding. Springer.
Ruhe, G., Eberlein, A., and Pfahl, D. (2003). Trade-Off Analysis for Requirements Selection, International Journal of Software Engineering and Knowledge Engineering (13:4), pp. 345-366
Schwaber, K.X. (2004). Agile Project Management with Scrum. O'Reilly Media, Inc.
Tabaka, J.X. (2006). Collaboration Explained: Facilitation Skills for Software Project Leaders. Pearson Education.
Tl, S. (1980). The Analytic Hierarchy Process. McGraw-Hill, New York.
Waters, K. (2009). Prioritization using Moscow, Agile Planning [online] http://www.allaboutagile.com/prioritization-using-moscow/ (accessed 11 April 2014).
Wiegers, K. (1999). First Things First: Prioritizing Requirements, Software Development (7:9), pp. 24-30.
Wolfswinkel, J.F., Furtmueller, E., and Wilderom, C.P.M. (2011). Using Grounded Theory as a Method for Rigorously Reviewing Literature, European Journal of Information Systems (22:1), pp. 45-55.