A Systematic Literature Review of Requirements Prioritization Criteria

Norman Riegel(✉) and Joerg Doerr

Fraunhofer IESE, Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
{norman.riegel,joerg.doerr}@iese.fraunhofer.de

Abstract. [Context & motivation] Requirements prioritization is typically applied in order to determine which requirements or features should be included in a certain release or implemented first. While most requirements prioritization approaches prescribe a fixed set of prioritization criteria that have to be assessed during the prioritization process, there is often a need for criteria that are customized to the specific project situation. **[Question/problem]** However, determining customized prioritization criteria is a time-consuming and laborious task. Instead of an in-depth analysis, criteria are often identified by gut feeling, which is error-prone and bears the risk of choosing misleading criteria. **[Principal ideas/results]** This paper aims at identifying and categorizing prioritization criteria discussed in the vast body of prioritization literature for software development. We describe a systematic literature review and, as a result, present a consolidated prioritization criteria model. **[Contribution]** Besides a comprehensive overview of prioritization criteria discussed in the literature, this paper introduces a classification schema that allows researchers and practitioners to identify prioritization criteria and related literature in a time-saving manner.

Keywords: Requirements prioritization criteria · Systematic literature review

1 Introduction

Requirements prioritization is an important aspect of any software development process. Requirements prioritization approaches are typically applied in order to determine which requirements or features should be included in a certain release or which should be implemented first [1]. Beyond that, recent approaches even utilize prioritization in order to identify the requirements that shall be refined next during a requirements elicitation process [2] [3].

In all prioritization approaches, one or more criteria are taken into consideration in order to determine the value of the requirements. Criteria may be, for example, “business value”, “implementation cost” or “risk”. Depending on the concrete prioritization approach used, the criteria are rated explicitly by different stakeholders (e.g., on Likert scales) or determined automatically based on other information (e.g., by analyzing system usage protocols).

S.A. Fricker and K. Schneider (Eds.): REFSQ 2015, LNCS 9013, pp. 300–317, 2015.
DOI: 10.1007/978-3-319-16101-3_22

Most requirements prioritization approaches prescribe a fixed set of criteria that have to be assessed during the prioritization process. Wiegers, for example, proposes in his method the use of the criteria "value", "cost", "risk", and "penalty" [1]. Only a handful of approaches do not prescribe the use of any criteria: in the value-oriented prioritization (VOP) approach [4], for example, core business values have to be determined first in order to assess the requirements against them.

The benefit of predefined criteria is that they can be used quickly out-of-the box. However, they often do not fit well into the given context [5]. Hence, there is a need for criteria that are customized to the specific project situation, but determining customized prioritization criteria is a time-consuming and laborious task. Instead of an in-depth analysis, criteria are often identified by gut feeling, which is error-prone and bears the risk of choosing misleading criteria. Berander [5] already remarked that research should focus on finding efficient methods for the determination of criteria (and developing prioritization approaches that can be used flexibly with different criteria), rather than spending effort on optimizing the calculations of prioritization approaches.

This paper aims at identifying and categorizing prioritization criteria discussed in the vast body of prioritization literature for software development. We describe a systematic literature review and, as a result, present a consolidated prioritization criteria model which consists of around 280 criteria. Several systematic literature reviews have already been performed in the context of requirements prioritization (e.g., [6][7][8][9]). In contrast to these studies, the goal of our work is not to identify certain prioritization approaches but to identify the criteria that are discussed in the prioritization literature. Our study is also not limited to any particular application domain, specific prioritization techniques, or special types of criteria (e.g., benefits). Furthermore, the goal of our study is not to merely collect these criteria, but to consolidate them and integrate them into a complete criteria model. We introduce a classification schema that allows researchers and practitioners to identify suitable prioritization criteria and related literature references in a time-saving manner. Thus, it is a first step towards the efficient selection of prioritization criteria for more flexible prioritization approaches.

The remainder of this paper is structured as follows: Section 2 describes our reseearch approach, i.e. the systematic literature survey and the creation of the prioritization criteria model, section 3 discusses the details of the criteria model, and section 4 finally concludes the paper and gives an outlook on future work.

2 Research Approach

In order to identify requirements prioritization criteria from the literature, we conducted a systematic literature review according to the guidelines of Kitchenham [10]. Thus, the first step (1) was the definition of a review protocol that defines the rationale for

the survey, the research question, the search strategy, and the selection and assessment criteria (see Figure 1). The research question to be answered was: “Which prioritization criteria are discussed in the requirements prioritization literature?” In order to answer this question, a search string (i.e., “requirements AND (value OR criteria OR metrics OR attributes OR measures OR factors) AND (prioritization OR negotiation OR "release planning" OR "decision making")”) was developed, tested, and applied to Scopus[1] as well as to the ACM digital library[2] in title, abstract and keywords. The reason for using these two libraries was that they cover most of the content from relevant publishers or databases [11] (e.g., from IEEE Xplore [12], Springer, Elsevier) and thus include most of the relevant work in the requirements engineering area (e.g., proceedings of the International Requirements Engineering Conference, proceedings of the REFSQ conference, the Requirements Engineering Journal, ICSE conference proceedings), as well as, for instance, the LNCS proceedings, where many related papers are published. Thus, a direct search at IEEE Xplore, Springer Link or other sources was not deemed necessary. The search was limited to computer science and related fields and included all available publication years, publication channels (e.g., conference and workshop proceedings, dissertations, journals, etc.), and publication types (e.g., method papers, experience reports, case studys, etc.). Although the tested search strings were used, it was found that many hits were not relevant for the research question. Those that do not deal with software requirements prioritization (e.g., construction material selection, prioritization in networks, etc.) or do not fit the search terms (i.e., listed in the search result, but does not include the search terms) were excluded. By reading the publications’ titles and abstracts and comparing them with the exclusion criteria defined, we were able to dismiss many papers directly (2).

In the third step (3), the remaining publications were accessed and classified into two categories in order to get an overview of the prevailing publication types:

- Category 1: (Software) Requirements prioritization and negotiation approaches and release planning models (i.e., publications where a concrete approach is described)
- Category 2: Non-methodological publications in the context of requirements prioritization (e.g., empirical studies and literature reviews)

Then the publications were analyzed in detail in order to identify prioritization criteria mentioned in the text. If necessary (i.e., if references to prioritization criteria from other resources were mentioned), secondary resources from the references were also accessed (4) and analyzed according to the defined review criteria.

[1] http://www.scopus.com/
[2] http://dl.acm.org/

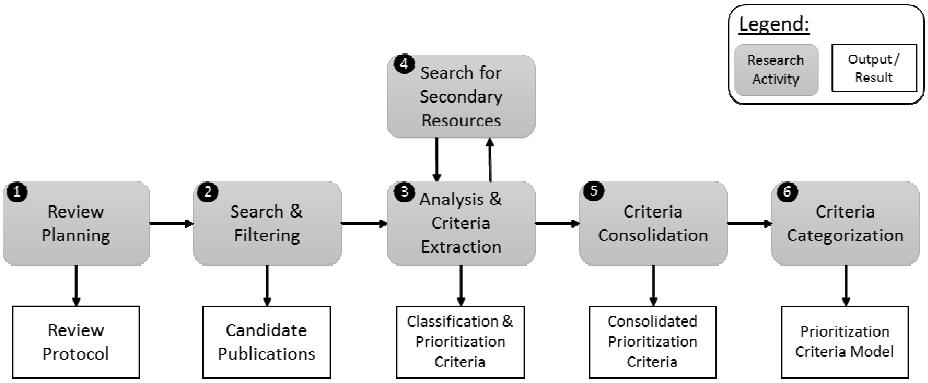

Fig. 1. Research Approach for Criteria Collection

The systematic literature review resulted in a set of 83 publications (narrowed down from around 1750 initial search hits) from which prioritization criteria could be extracted (the complete list of literature sources can be found in [13]). We classified 61 publications in category 1 and 22 publications in category 2. From the 83 publications, we extracted around 760 criteria in total.

In the next step (5), these criteria were then analyzed in order to remove duplicates and identify synonyms and homonyms. During the consolidation of the criteria, we experienced the following challenges:

- Most of the publications we analyzed do not describe the contained criteria in detail; rather, they just list them in the text without any description or further discussion. Thus, it is often not clear for the reader what exactly is meant by the criterion. Only a few papers (e.g., [14], [15]) describe the mentioned criteria in more detail. As a consequence, identifying synonyms or homonyms across the different publications was difficult and not always possible.
- In general, there is no consensus regarding the meaning of a criterion across publications and among different authors. For example, in some publications the criterion "implementation cost" is synonymous with "development cost", i.e., the authors do not distinguish between these two terms. However, in other publications, "development cost" (and further cost types) is seen as merely a part of "implementation cost". Thus, "implementation cost" is rather used as an umbrella term in the latter. As a consequence, the classification of a criterion in the hierarchy of our criteria model (see step (6)) might not accurately reflect the original meaning of this criterion intended by certain authors. Thus, the classification of a criterion in the criteria model reflects our interpretation of its meaning, which is also defined in the model. Furthermore, due to the similar nature of some criteria, not all criteria are independent of each other.

- Due to the nature of the different publications (e.g., concrete prioritization method descriptions, empirical surveys, etc.), the criteria that we found were on very different abstraction levels. Whereas the criteria applied in method descriptions are typically on a level that can be assessed on a scale by a stakeholder, the criteria discussed in surveys are on a level where an assessment scale is hard to apply. Furthermore, some criteria are on a very abstract business level (e.g., "market value"), while others are on a very detailed technical level, depending on particular requirements artifacts (e.g., "number of times use case appears in model").
- Often, publications use generic terms for the criteria, e.g. "risk". In this case, we tried to find out from the surrounding text what exactly the authors meant with this term. Often it was possible to find a more concrete interpretation, such as "technical risk". However, in some of the publications, it was not possible to find a concrete interpretation. Thus, only the generic term could be referenced.

Finally, we were able to consolidate the vast number of criteria into a smaller set of around 280 criteria. Similar to a thematic analysis approach [16], we categorized them into several abstract categories during this consolidation. The clusters were built by browsing the criteria and identifying and naming themes to which the criteria belong. As a starting point, we named the major categories according to the structuring in [1].

In the last step (6), this categorization was iteratively refined, creating further subcategories. In addition, a definition was provided for each criterion in order to foster a common understanding of the criteria. Finally, the model was discussed with several requirements engineering experts. Based on their feedback, the model was slightly restructured into its final structure, which is described in section 3.

Concerning the threats to validity, we consider two threats to be the most prominent ones. First of all, despite the rigorous search strategy, there exists the possibility that we may have missed important publications. Publications might exist that are not indexed in the databases we used. Also, we might not have extracted all relevant publications using the search terms in their titles, abstracts, or keywords. However, we noticed during the extraction of the criteria that after a certain number of publications, no new criteria could be identified anymore. Thus, the integration of missing publications might not have a large impact on the model, as only few new criteria might exist. The second threat arises due to the challenges descibed above. As we had to consolidate the vast number of criteria into a smaller set, the possibility exists that some criteria were seen as synonyms, even if their meaning might actually be different. Thus, some criteria might have been discarded inadvertently. This is due to the fact that mostly no definitions for criteria were provided in the literature. However, we tried to minimize this threat by not eliminating questionable criteria, rather trying to integrate them into our taxonomy. Furthermore, due to the structure of the model and the definitions we created, it should be easy to integrate other criteria into the model.

3 Prioritization Criteria Model

The prioritization criteria model consists of six major categories, which are further refined into different subcategories. The major categories are: (1) benefits, (2) costs, (3) risks, (4) penalties & penalty avoidance, (5) business context, and (6) technical context & requirements characteristics. The largest group is the benefits category, containing 93 criteria, which represent 33% of all criteria contained in the model (see Figure 2).

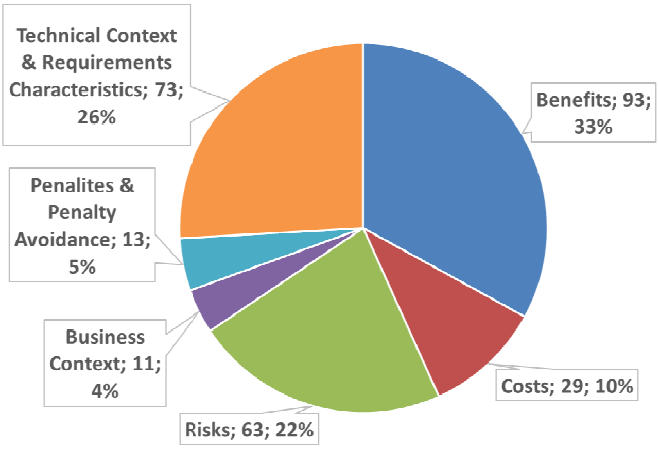

Fig. 2. Major Categories in the Prioritization Criteria Model and the included Number of Criteria

Figure 3 shows the Top Ten of the most frequently mentioned criteria in the literature. The most frequent criterion is "implementation effort" with 54 occurrences. Three of the Top Ten criteria are from the benefits category ("business value", "customer satisfaction", and "stakeholder satisfaction"), two criteria are from the costs category ("implementation effort" and "development effort"), four criteria are from the technical context & requirements characteristics category ("resource availability", "implementation dependencies", "requirements dependencies", and "staff competence"), and one criterion is from the risks category ("requirements volatility").

In the next sections, the prioritization criteria model will be described in more detail. Section 3.1 describes the structure of the model and basic terminology necessary to understand its structure. Section 3.2-3.7 describe the different categories and subcategories of the model, as well as some examples of criteria included in them.

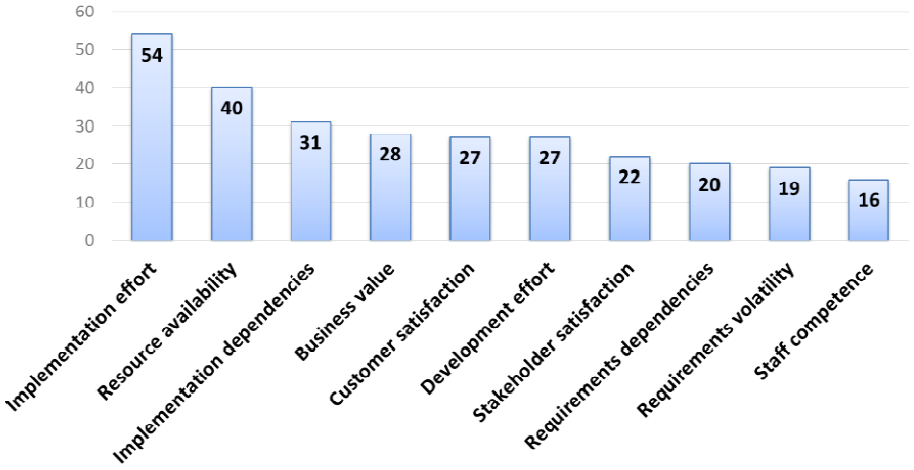

Fig. 3. Top Ten Prioritization Criteria mentioned in the Literature

3.1 Structure and Terminology

During the creation of the criteria model we recognized that in order to create a meaningful categorization, a common understanding is also necessary of the basic terms needed to describe the different categories and subcategories of the model. In the following, an overview of the most important terms used in the model and their conceptual relationships is given (see Figure 4).

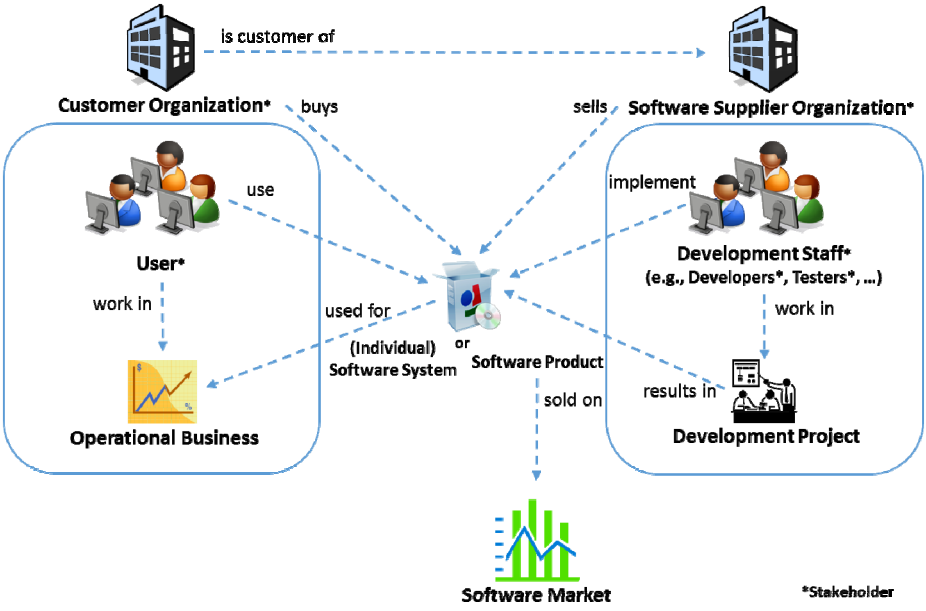

Fig. 4. Terminology used in the Model

Typically, the requirements are prioritized either by a software supplier organization (software development organization) or by a customer organization (receiver/beneficiary of the software system to be developed). Thus, both are also the main stakeholder groups of the criteria model. The prioritized requirements serve as a basis for developing either a software product/system to be sold on a software market or an (individual) software system for a certain customer organization. Whereas in the former case, the software supplier prioritizes the requirements based on market needs, in the latter the customer organization does so based on their operational business needs. The software product is implemented in development projects by development staff (e.g., developers, testers, user interface designers) of the software supplier. The software product (either bought on the market or individually developed) is finally used by the end users in the customer organization in order to support the operational business of the organization. Organizations, and mainly particular roles like user or developer, which are typically involved in or affected by the prioritization, are summarized under the term stakeholder.

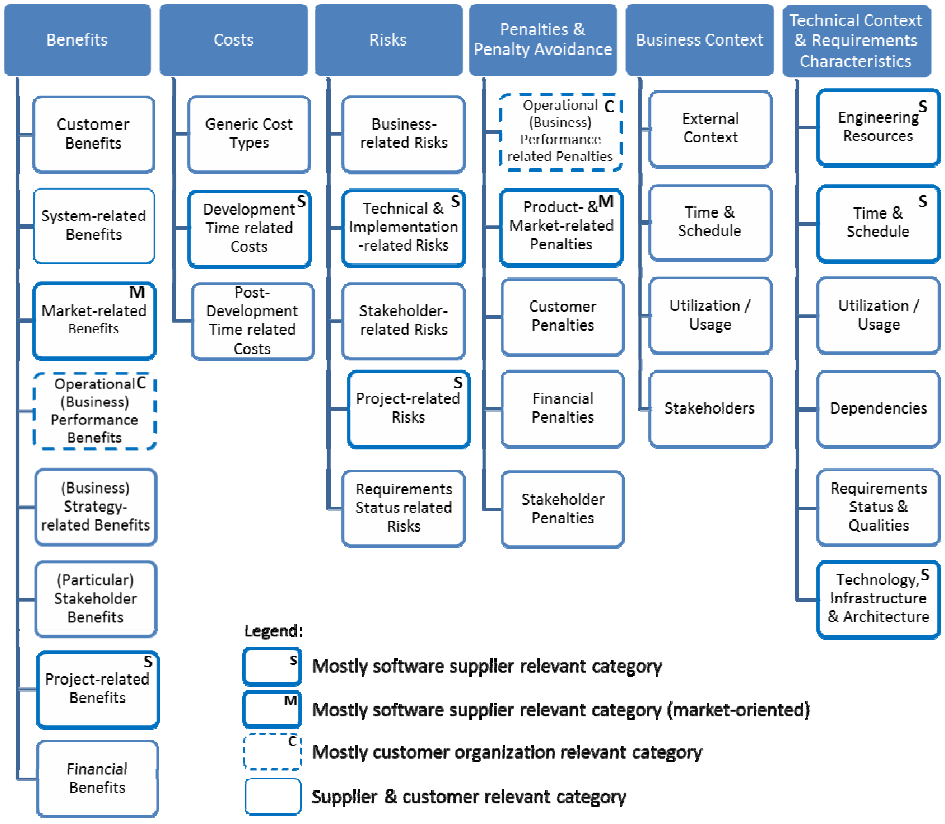

Fig. 5. The Structure of the Prioritization Criteria Model

The model contains criteria that are useful in different requirements prioritization contexts. Thus, it is not customized for any specific point of view. Depending on the situation and the perspective of the decision maker (i.e., software supplier or customer organization), particular subcategories (and the included criteria) are more appropriate for the concrete prioritization than others. An overview of the structure of the model is given in Figure 5. In order to provide guidance, we included some indicators to facilitate navigation in the model based on the prioritization perspective. As it can be seen, the number of subcategories is not balanced between the different major categories. The reason for this is that the type and number of subcategories depend on the criteria found in the literature.

In part, the criteria are also structured hierarchically, i.e., more specific criteria are subordinated under more generic criteria, if appropriate[3]. Furthermore, the criteria themselves are described in detail in a tabular form; an example is shown in Table 1. In the following sections, we will present the criteria that are included in the different categories. However, due to their vast number, we are not able to provide all their details here.

Table 1. Criterion Description Scheme

Criterion	References	Interpretation	Exemplary Metric
Example: Implementation cost / effort	Example: [Wie99] [Moh08] [Fir04]	Example: Costs of any kind (development, testing, integration, etc.) that are incurred if the requirement is implemented.	Example: E.g., effort in [person-days]

3.2 Benefits

In this category, criteria are included that express or are related to benefits that are realized if the respective requirement[4] is implemented (see overview in Table 2). We further divided this category into the following subcategories:

- Customer Benefits: contains criteria that express or are related to benefits for a customer (e.g., “efficiency gains for customer, “competitive gains for customer”).
- System-related Benefits: contains criteria that express or are related to benefits with regard to a software system/product (e.g., “product quality”).
- Market-related Benefits: contains criteria that express or are related to benefits with regard to the market to which a software product is related (e.g., “customer loyalty”).

[3] However, the subordinated criteria typically do not completely describe the superordinate criterion.

[4] Although the term *feature* is sometimes used in the literature instead of the term *requirement*, we do not distinguish between these two terms in our model.

- Organizational (Business) Performance Benefits: contains criteria that express or are related to operational performance benefits with regard to the operational business of an organization (e.g., "cost saving", "process efficiency").
- (Business) Strategy-related Benefits: contains criteria that express or are related to business-strategy related benefits (e.g., "contribution to business goals", "long-term strategic value").
- (Particular) Stakeholder Benefits: contains criteria that express or are related to benefits for stakeholders in general (e.g., "stakeholder satisfaction") or particular stakeholder groups, e.g., users of the software product or system (e.g., "end user satisfaction").
- Project-related Benefits: contains criteria that express or are related to benefits for a software development project (e.g., "relevance to project success", "contribution to overall release goal").
- Financial Benefits: contains criteria that express or are related to financial benefits (e.g., "ROI", "NPV").

Table 2. Overview of the Criteria in the Benefits Category

Benefits

- Business value / business importance / gain for organization

System-related Benefits

- Product / system value
  - Linkage to overall system goals
- Product / system quality
  - Ease of use / convenience
  - Scalability
  - Sustainability of solution
  - Changeable solution
  - Uniform solution
  - Performance
  - Stability
  - Security
  - Integrity
  - Availability
  - Testability
  - Accuracy

Market-related Benefits

- (Product) Market value
- Customer loyalty / retention
- Marketability / ability to sell
- New business potential / product and service enhancement
  - Additional customer sales
  - Extra cost customer will spend
- Market percentage
- Competitiveness
  - Creation of competitive advantage
  - Status of competitors with respect to the requirement
  - Innovativeness
- Market technology trends
- Brand protection
- (Feature) Influence on buying decision
- Resalable solution
- (Long term) Product strategy
- Fit with / effects on other products

Table 2. *(continued)*

Project-related Benefits

- Project value
  - Relevance to project success
  - Importance wrt. / contribution to overall release goal
  - Release theme
  - Feature contribution to project vision
- Synergy effects by combining tasks

(Particular) Stakeholder Benefits

- Personal preference & stakeholder priority / preference / value / satisfaction / desire
  - End user value / satisfaction
  - Value creation for developer
  - Relevance to stakeholders' goals
- Fit with skills / training

Financial Benefits

- Financial benefit / revenue
  - Return on investment (ROI)
  - Net present value (NPV)
  - Internal rate of return (IRR)
  - Payback period

(Business) Strategy-related Benefits

- Strategic alignment / suitability to business strategy
  - Importance / contribution to business goals
  - Criticality to mission success
- Long term strategic value / strategic benefit
- Tactical usefulness

Customer Benefits

- Customer value / satisfaction / preference
  - Efficiency gains for customer
  - Competitive gains for customer
  - Intermediary satisfaction

Operational (Business) Performance Benefits

- Organizational effectiveness
  - Support of work
    - Fit with business processes
    - Contribution to user task
    - Feature / requirement support for main (usage) scenario
- Organizational efficiency / productivity improvement
  - Cost saving / reduction
  - Economies of production
  - Cost reduction of IT operation
  - Process / workflow efficiency
  - Speed
  - Throughput
- Operational risk reduction
  - Countermeasure benefit
- Inbound logistics
- Supplier relations
- Customer relations
- Competitor relations
- Business innovation
- Deliveries
- Third party relations
- Marketing support
- Decision making
- Learning and knowledge
- Organization culture
- Information
- Technology / tools
- Strategy formulation and planning
- Communication
- Flow of products / services
- Control and follow-up (improved reporting)
- Change management
- Integration and coordination
- Flexibility

3.3 Costs

In this category, criteria are included that express or are related to costs that are incurred if the respective requirement is implemented (see overview in Table 3). The category is divided into the following subcategories:

- Generic Cost Types: contains criteria that express or are related to generic cost types (e.g., "life-cycle costs", infrastructure").
- Development Time related Costs: contains criteria that express or are related to costs related to the development phase of a system or software product (e.g., "development cost", "testing cost").
- Post-Development Time related Costs: contains criteria that express or are related to costs that are incurred after the development phase of a system or software product (e.g., "maintenance cost", operational costs").

Table 3. Overview of the Criteria in the Costs Category

Costs

Generic Cost Types
- Life-cycle costs
- Total ownership cost
- Money / finances / budget
- Time
- Labor
- Overhead
- Infrastructure
  - Hardware unit costs
- Travel
- Material

Post-development Time related Costs
- Post-development cost
  - Maintenance cost
  - Support costs
  - Operational costs
  - Shipping

Development Time related Costs
- Implementation cost / effort
  - Development cost / effort
    - Task size
  - Documentation cost
  - Functional implementation effort
  - Technical implementation effort
  - Quality cost
    - Testing cost
      - Integration testing cost
      - User acceptance testing cost
    - Cost for fixing defects detected during testing
  - Quality (attribute) implementation effort
  - Risk mitigation effort / cost
- Product costs

3.4 Risks

In this category, criteria are included that express or are related to risks regarding the implementation of a requirement (see overview in Table 4). The category is divided into the following subcategories:

- Business-related Risks: contains criteria that express or are related to risks with respect to the business (e.g., "loss of customers", "sales barriers").
- Technical & Implementation-related Risks: contains criteria that express or are related to risks with respect to technology and implementation (e.g., "architecture conflict", "implementation difficulty").
- Stakeholder-related Risks: contains criteria that express or are related to risks for and induced by stakeholders (e.g., "personnel risks", "risks of acceptance").
- Project-related Risks: contains criteria that express or are related to risks in a software development project (e.g., "overrun risk", "over budget risk").
- Requirements Status related Risks: contains criteria that express or are related to risks due to the changes and imprecision of a requirement (e.g., "market changes", "changes from technical perspective").

Table 3. Overview of the Criteria in the Risks Category

Risks

Technical & Implementation-related Risks

- Technical risks
  - Architecture conflict
    - Severe redesign of architecture
    - Impact on essential non-functional requirements
  - Database risks
  - Product quality loss
    - Performance risks
    - Risk of maintenance
    - Loss of confidential data
  - Technical risk in current system
  - Technical risk in proposed system
  - (Technical) Complexity
- Implementation risk
  - Implementation technology risks / technical uncertainty
  - Scope loss
  - Risk of buggy implementation
  - Implementation difficulty
    - Development risk
    - Difficulty of programming language used
    - Large Size
    - Tedium
    - Uncertainty
    - Novelty
    - Number of people involved
    - Organizational constraints
  - Implementation feasibility
    - Ease of realization (technical feasibility)
  - External risks

Requirements Status related Risks

- Vague story
- Requirements volatility / stability
  - Changes from business perspective
  - Changes from technical perspective
  - Market changes
  - Legislative changes
  - Users change
  - Requirements become more clear during software life cycle

Business-related Risks

- Business risks
  - Productivity loss
  - Loss of reputation
  - Loss of customers
  - Negative value of undesired event inherent in a feature
    - Security related risks / misuse case risk
    - Safety risks
  - Ease of realization (economic feasibility)
  - Ease of realization (social feasibility)
  - Ease of realization (political feasibility)
  - Market uncertainty
  - Sales barriers
  - Commercial concerns

Project-related Risks

- Overrun / schedule risk / loss
- Over budget risk
  - Fixing cost (losses)
- (Implementation) Process risk
- Environmental factor (development context)
- Project duration
- Project dependencies
- Impediment of attaining the requirement in the project

Stakeholder-related Risks

- Personnel risks
- Estimation risks (in size & team productivity)
- Risks of acceptance
- (Decision) uncertainty
- Part time team member use

3.5 Penalties & Penalty Avoidance

In this category, criteria are included that express or are related to penalties that occur if the respective requirement is not implemented (e.g., “negative value”), as well as criteria that express or are related to penalty avoidance (e.g., “harm avoidance”) if the respective requirement is implemented (see overview in Table 5). The category is divided into the following subcategories:

- Operational (Business) Performance related Penalties: contains criteria that express or are related to penalties for the operational business of an organization (e.g., "penalty in operation").
- Product- & Market-related Penalties: contains criteria that express or are related to penalties for a software product or with regard to the market to which the software product is related (e.g., "damage to product").
- Customer Penalties: contains criteria that express or are related to penalties for customers (e.g., "customer dissatisfaction").
- Financial Penalties: contains criteria that express or are related to financial penalties (e.g., "contractual commitment").
- Stakeholder Penalties: contains criteria that express or are related to penalties for stakeholders (e.g., "stakeholder dissatisfaction").

Table 4. Overview of the Criteria in the Penalties & Penalty Avoidance Category

Penalties & Penalty Avoidance
❖ Negative value / loss / damage / penalty to business / loss of value
❖ Harm avoidance

Operational (Business) Performance related Penalties	**Financial Penalties**
❖ How complicated would workaround be	❖ Cost of not implementing
❖ Penalty in operation / performance impact	❖ Financial penalty / profit aspect (penalty)
	❖ Legal mandate / regulations
	❖ Promised / contractual commitment
Product- & Market-related Penalties	**Stakeholder Penalties**
❖ Detraction from product's value / damage to product	❖ (Stakeholder) Dissatisfaction
❖ Market share aspect (penalty)	
Customer Penalties	
❖ Negative value / damage / penalty for / loss to customer	
❖ Customer dissatisfaction	

3.6 Business Context

In this category, criteria are included that are related to the business context of a requirement (see overview in Table 6). These criteria do not express any concrete benefit or loss that is realized when implementing a requirement, but rather describe context factors that have a positive or negative influence on the requirement's value. For example, if two requirements are equally beneficial, a context factor may help to distinguish them and help to decide which one to prefer. An example of such a criterion is "urgency". Two requirements might be of the same benefit, but one is more urgent than the other. Thus, the more urgent one is preferred. The category is divided into the following subcategories:

- External Context: contains criteria that express or are related to the external context of an organization (e.g., "external dependencies", "customer demand").
- Time & Schedule: contains criteria that express or are related to business-relevant time & schedule contexts (e.g., "urgency", "time to market").
- Utilization/Usage: contains criteria that express or are related to the business usage context (e.g., "frequency of use").

- Stakeholders: contains criteria that express or are related to the stakeholder context (e.g., “originator of requirement”, “stakeholder agreement”).

Table 5. Overview of the Criteria in the Business Context Category

Business Context	
External Context ❖ After sale support ❖ External dependencies ➢ Customer demand ➢ Formal governmental demand	**Utilization / Usage** ❖ Frequency of use ❖ Actor priority / weight
Time & Schedule ❖ Urgency ➢ Time to market	**Stakeholders** ❖ Originator of requirement ❖ Stakeholder agreement ❖ Likelihood of success

3.7 Technical Context & Requirements Characteristics

In this category, criteria are included that are related to or based on the technical context of a requirement (e.g., “architecture impact”), as well as particular requirements characteristics (e.g., “readiness for implementation”). Just like the business context criteria, these criteria do not express any concrete benefit or loss that is realized when implementing a requirement, but rather describe context factors that have a positive or negative influence on the decision in favor of or against a requirement. An overview is shown in Table 7. The category is divided into the following subcategories:

- Engineering Resources: contains criteria that express or are related to engineering resources, i.e., basically human resources (e.g., “staff competence”, “resource availability”).
- Time & Schedule: contains criteria that express or are related to implementation time and schedule (e.g., “development lead time”, “delivery date”).
- Utilization/Usage: contains criteria that express technical usage requirements characteristics (e.g., “reuse potential”).
- Dependencies: contains criteria that express or are related to dependencies of and between requirements (e.g., “implementation dependencies”, “cost dependencies”).
- Requirements Status & Qualities: contains criteria that express or are related to the status of requirements and the (specification) quality of the requirements (e.g., “traceability”, “completeness”).
- Technology, Infrastructure & Architecture: contains criteria that express or are related to the technological, infrastructural, and architectural context of a requirement (e.g., “importance for product architecture”, “infrastructure criticality”).

Table 6. Overview of the Criteria in the Technical Context & Requirements Characteristics Category

Technical Context & Requirements Characteristics

Engineering Resources

- Staff competence / skills
  - Familiarity of the life-cycle model during the project
  - Experience on the area
  - Experience on development methodology used
  - Ability of analyzer
- Motivation of the team
- Balanced workload
- Resource availability / capacity
  - Developer productivity
    - Velocity
  - Key resources
    - Resources for specification
    - Functional resources
    - Analysis and design
    - Implementation / Development
    - Testing
    - User interface
    - Research

Time & Schedule

- (Implementation) time / schedule
  - (Development) Lead time
- Delivery date / release date
- Project deadline / temporality

Utilization / Usage

- Object usage for a particular scenario of the use case
- Actor usage for a particular scenario of the use case
- Reuse potential / reuse frequency
- Number of times use case appears in model

Requirements Status & Qualities

- Readiness for implementation
- Adequate / Sufficient detail in specification
- Requirements quality / requirements specification factors
  - Modifiability
  - Traceability
  - Testability
  - Completeness
  - Consistency
  - Understandability
  - Within Scope
  - Non-redundant

Dependencies

- Requirements dependencies
  - Technical & functional dependencies
  - Number of requirements that depend on requirement
    - Number of use cases this use case includes
    - Number of use cases that includes this use case
    - Number of use cases this use case extends
    - Number of use cases that extend this use case
    - Number of use cases inherited by this use case
  - Feature weight from use cases
  - Revenue dependencies
  - Cost / effort dependencies
  - Inter-domain dependencies
  - Dependencies among user stories
  - Dependencies among delivery stories (between non-functional requirements and architectural choices)
- Implementation dependencies
  - Intra-domain dependencies
  - Dependencies due to downstream activities
  - Team-based dependencies

Technology, Infrastructure & Architecture

- System impact (changes to existing system)
  - Keep legacy system alive
  - (Impact on) Maintenance (of current system)
  - (Long term) architecture Impact
    - Importance for product architecture
    - Technical debt
    - (Impact on long-term) Evolution (of system)
  - (Short term) Architectural / development impact
  - Infrastructure (criticality)
  - Preferred operating architecture
  - Technology opportunities
  - Technology should support current functionality
  - Integration to external systems
- System value of a feature (impacted components)
- Use case weight (transactions)
- Adherence to corporate software design parameters
  - IT departments technical guidelines
- Technical priority

4 Conclusion and Future Work

In this paper, we presented a prioritization criteria model as the result of a systematic literature review. It consists of about 280 prioritization criteria extracted from the literature and allows identifying prioritization criteria in a time-saving manner. It is a first step towards the efficient selection of prioritization criteria for more flexible prioritization approaches. It can be used as a basis for the further development of domain-specific criteria models. Future work on the model should include customization for certain application domains to facilitate its usage. Also it cannot be ruled out that criteria might exist that were not identified during the survey and that must be integrated in the future. As a next step, we will customize the model for the business application domain in order to apply it in our prioritization framework [2].

References

1. Wiegers, K.E.: First Things First: Prioritizing Requirements. Software Development **7**(9), 48–53 (1999)
2. Riegel, N.: Guiding requirements elicitation using a prioritization framework. In: REFSQ 2013 Workshop Proceedings, pp. 133–144 (2013)
3. Riegel, N., Doerr, J.: An analysis of priority-based decision heuristics for optimizing elicitation efficiency. In: Salinesi, C., van de Weerd, I. (eds.) REFSQ 2014. LNCS, vol. 8396, pp. 268–284. Springer, Heidelberg (2014)
4. Azar, J., Smith, R.K., Cordes, D.: Value-oriented requirements prioritization in a small development organization. IEEE Software **24**(1), 32–37 (2007)
5. Berander, P.: Evolving Prioritization for Software Product Management. Blekinge Institute of Technology. Doctoral Dissertation Series (2007)
6. Daneva, M., Herrmann, A.: Requirements prioritization based on benefit and cost prediction: An agenda for future research. In: Proc. of RE 2009, pp. 125–134 (2009)
7. Pitangueira, A.M., Maciel, R.S.P., de Oliveira Barros, M., Andrade, A.S.: A systematic review of software requirements selection and prioritization using SBSE approaches. In: Ruhe, G., Zhang, Y. (eds.) SSBSE 2013. LNCS, vol. 8084, pp. 188–208. Springer, Heidelberg (2013)
8. Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., Shafique, M.U.: A systematic review on strategic release planning models. Information and Software Technology **52**, 237–248 (2010)
9. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.N.: A systematic literature review of software requirements prioritization research. Information and Software Technology **56**(6), 568–585 (2014)
10. Kitchenham, B.: Procedures for Performing Systematic Reviews. Keele University Technical Report TR/SE-0401. Keele University (2004)
11. Elsevier: http://www.elsevier.com/online-tools/scopus/content-overview. (last accessed on October 16, 2014)
12. IEEE: https://supportcenter.ieee.org/app/answers/detail/a_id/510/~/is-ieeexplore-digital-library-content-indexed-in-scopus%3F. (last accessed on October 16, 2014)

13. Riegel, N.: Prioritization Criteria Collection and Literature Sources. IESE-Report, 048.14/E (2014)
14. Wohlin, C., Aurum, A.: What is important when deciding to include a software requirement in a project or release? In: Int. Symp. on Empirical SE, pp. 246–255 (2005)
15. Barney, S., Aurum, A., Wohlin, C.: Quest for a silver bullet: Creating software product value through requirements selection. In: Proc. of SEAA 2006, pp. 274–281 (2006)
16. Cruzes, D.S., Dybå, T.: Research synthesis in software engineering: A tertiary study. Information & Software Technology **53**(5), 440–455 (2011)