Towards a Taxonomy of Requirements for Hybrid Products

Alexander Herzfeldt\(^1\) Robert O. Briggs\(^2\) Aaron Read\(^2\) Helmut Krcmar\(^1\)

\(^1\)Technische Universität München, Germany; \(^2\)University of Nebraska at Omaha, USA

\{alexander.herzfeldt|krcmar@in.tum.de\}; \{rbriggs|aread@mail.unomaha.edu\}

Abstract

In order to differentiate from competitors and to respond to new customer expectations, many organizations develop hybrid products, composed of hardware, software and service elements. Determining the requirements for a hybrid product, however, can be complex. Designers must address the requirements for each of the product elements, as well as the interfaces and interdependencies among them and the service organization. Complexity increases with stakeholder interests associated with each element.

As a first step towards reducing this complexity, we derive a taxonomy of requirements for hybrid products. We begin by analyzing requirements literature in the three disciplines: hardware, software, and service requirements and synthesize requirements categories from each discipline. Next, we synthesize a taxonomy of requirements for hybrid products, defining and describing each category. We conclude with limitations of our work and directions for future research to refine and utilize the taxonomy.

1. Introduction

In order to differentiate from competitors in the face of high market pressure and to respond to challenges created by customer expectations for services and applications, many organizations now develop, market, and use hybrid products [1]. A hybrid product is composed of hardware, software and service elements bundled together as a single experience [2]. Hybrid products are designed to meet customer-specific business requirements [2,3]. An example of a hybrid product is a pay-per-copy solution: The contractor has to provide the photocopier (hardware), a control system (software) and maintenance and repair services. The customer does not receive the elements separately but as a bundle.

Determining requirements for hybrid products can be a complex process [1,2]. The requirements must address not only the characteristics of three disciplines: hardware engineering, software engineering, and service engineering [3], but must also address the interfaces and interdependencies, e.g., service use and level service aspects, for example. Complexity increases because there are stakeholders for each of the product elements, and for the hybrid product as a whole whose interests must be accommodated in order for the hybrid product to succeed. It would, therefore, be useful to develop a well-structured work practice for determining requirements of hybrid products.

Authors have advanced several useful requirements methods for Software Requirements Engineering (SRE) e.g. [4-7]. These approaches have been shown to be effective in hundreds of software development projects [4]. Although they focus solely on software, they could be adapted for hybrid products. Each of these approaches has, as a crucial artifact at its core, a taxonomy of requirements.

A taxonomy is an artifact used to classify sets of objects [8]. Stakeholders in these work practices use a taxonomy throughout the requirements elicitation process to stimulate thinking, focus effort, and to check thoroughness as they work towards a comprehensive collection of requirements [5].

As a starting point toward reducing the complexity of developing hybrid products it would, therefore, be useful to develop a requirements taxonomy that addresses the needs of stakeholders in a hybrid product project. In this paper, we derive such a taxonomy, and validate it with experts and stakeholders in hybrid products development projects. We then show some limitations of our work and suggest future directions for research to refine the taxonomy.

2. Methods

To derive a taxonomy of hybrid product requirements, we analyze requirements taxonomies from literatures in the three relevant disciplines: hardware requirements, software requirements, and service requirements. We synthesize separate requirements categories from each of these literatures. We then synthesize our findings into a taxonomy of requirements for hybrid products. We argue for the addition of categories not found in the component literatures, and for the elimination some categories that are less useful. We define and describe the categories of the proposed taxonomy in detail.

For an initial verification of the taxonomy derived from literature we then conducted seven structured interviews with experts in the field of the hybrid...
product domain. The experts span the IT, telecommunications, medical, and engineering industry. Four of the experts work in academia and are authors of hybrid product or hybrid product related publications for many years. Three of the experts are stakeholder in hybrid product development projects and have broad experience in managing hybrid product projects, designing hybrid products or procuring hybrid products. Consequently, all experts are potential users of the taxonomy.

The structured interview consisted of ten open-ended questions and focused on the interviewee’s impressions of the taxonomy. These questions are available in Appendix 2. Five of the seven interviews were conducted face-by-face; two of the interviews were conducted via telephone. The interviews were carried out in Germany and the US and lasted between 25 and 50 minutes.

3. Requirements taxonomies in three literatures

The field of RE developed differently in each of the three disciplines involved in hybrid products [3]. While RE for hardware engineering is generally restricted to early phases of the development process, RE in software engineering is well-elaborated throughout the software life cycle [3]. As service engineering is still a young discipline [9], RE in this discipline is nascent [3].

Some research has also been done on complex systems [8,10-12], which are defined as having both hardware and software components. We analyzed the RE literature in this domain along with literature from hardware engineering.

3.1 Requirements in hardware engineering and complex systems

Several taxonomies and checklists of requirements have been advanced in the hardware engineering literature [13-15] and in literature on complex systems [8,10-12]. There are many points of congruence among these models, which we synthesized into a seven-category classification scheme for hardware requirements (cf. Table 1).

3.1.1 Project requirements Most RE researchers in the field of hardware design propose requirements for cost and schedule [8,11-15]. Others include categories for political constraints [12], marketing, business environment, strategic management, finance and accounting [14]. One addresses suppliers, experience, and social sanctions [13]. We coded all these requirements as Project Requirements, as they refer to the way development projects are to be set up and conducted.

3.1.2 Functions Many researchers [8,11-15,10] propose categories related to system functions [8,10,11,13], system features [14], functional requirements [12], and system capabilities [8]. Some divide this category into subcategories, such as functions, behavior, data and physical items and non-functional requirements, e.g. [8]. We coded these categories as Functions, which refer to the aspects of the hardware people will use to achieve their goals.

3.1.3 Development requirements A number of authors identified requirements related to technical constraints [14], physical constraints [11,15], regulatory like policies, procedures, standards [8,12,14], equipment [15], production, finishing and assembly constraints [14]. As all these are important aspects when developing a hardware, we summarized these requirements as Development Requirements, which pertain to the tools and practices of the development process.

3.1.4 Operational requirements Many authors [8,11-15,10] refer to requirements related maintenance [8]

<table>
<thead>
<tr>
<th></th>
<th>Project Requirements</th>
<th>Functions</th>
<th>Development Requirements</th>
<th>Operational Requirements</th>
<th>System Growth Requirements</th>
<th>Level of Service Requirements</th>
<th>Interface Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roman (1985)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hughes et al. (1994)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ullman (2002)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Marks (1991)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Pahl/ Beitz (1984)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Henninger (1980)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1 Comparison of requirement categories in hardware engineering and complex systems
[8,11-15,10], service [13], and support [14]. Some authors also suggest constraints related to the operational environment [8,12,13], operating constraints in terms of accessibility to maintenance [12], and operational steps and sequence [15]. These requirements are to be fulfilled during the operational phase of the hardware and we therefore coded them as Operational Requirements, which pertain to the requirements and assumptions about the resources and procedures needed to run the hardware.

3.1.5 System growth requirements Many authors [8,11-13] propose requirements related to expected system changes [8], enhancement [12], upgradeability and standardization [13]. We coded requirements related to improvement and to the hardware’s adaptability to new environments as System Growth Requirements, which pertain to future plans for changing the system.

3.1.6 Level of service requirements Most researches [8,11-15,10] suggest requirements categories like performance [11,13,15], non-functional requirements [8,12], flexibility [11], and usability [14]. Availability is named by most researchers as part of the non-functional requirements [10-15]. Other non-functional requirements named are reliability [11-13], survivability [10-12,15], overall performance [10-15], and security and safety [11-14]. Especially [13] offers a very comprehensive collection of these requirements. We synthesize requirements related to how well a product fulfills its purposes into a Level of Service category.

3.1.7 Interface requirements Several authors [8,10-13] suggest requirement categories related to interoperability [10,11], connections [8], interface constraints [12] and interfaces [13]. We coded all requirements defining the way the hardware must interact with other systems, with people, and with its environment as Interface Requirements, which pertain to ways the system must interact with people and with other systems.

3.2 Requirements in software engineering

We analyzed several general taxonomies of software requirements from the software engineering literature, among them the IEEE Recommend Practice for Software Requirements Specification [16] and the Model-Based Architecture and Software Engineering (MBASE) process [17]. We also examined taxonomies focused on software performance requirements [18], software security requirements [19], and risk [20]. We synthesized the points of congruence from the literature into a high-level classification scheme (Table 2).

3.2.1 Project requirements Several software requirements researchers propose categories for cost, budget, schedule, and staffing [17,20,16], facilities, contract constraints, reporting, customers and vendors [20]. We coded all requirements related to the way software projects are set up and conducted as Project Requirements.

3.2.2 Functions Several researchers for software requirements suggest requirements categories related to application capabilities [17], functional requirements [16] and functionality [17,20]. We coded such categories under the heading Functions, because they refer to the software’s features people use to achieve their goals.

3.2.3 Design requirements Researchers in software engineering [16,17,20] state requirements related to hardware constraints [16], standards [16,17], tools, programming language, and computer resources [17] in the design phase of a software. Because these aspects pertain to the tools, procedures, and constraints of the design process, we coded them as Design Requirements.

3.2.4 Level of service requirements Many taxonomies of software requirements [16-20] refer to categories related to level of service [17], performance [20], quality of service [18], quality requirements [19], and performance requirements [16]. Some address levels of service requirements consisting of general

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carr et al. (1993)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Sabata et al. (1997)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firesmith (2005)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE Recommended Practice</td>
<td>(X)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2 Comparison of requirement categories in software engineering
qualities, coherence, dependability, security, confidentiality, safety, interoperability, usability, performance, adaptability, and reusability [17]. Others identify reliability, availability, security, maintainability, and portability as important level of service requirements [16]. Some add categories for safety, security, and survivability [18], as well as timeliness, precision, and accuracy [19]. We coded all requirements related to software quality as Level of Service Requirements.

3.3 Requirements in service engineering

Although the service engineering discipline is young, one can, nonetheless, infer a number of requirements for services from the literature. We synthesized congruencies from literature not only from core contributions on service engineering [9,21], but also on service-oriented RE [22], service design [23], service marketing [24], and general service literature [25,26] into seven high-level categories.

3.3.1 Project requirements Authors mention issues related to organizational constraints [9,23,24], the organization pattern [25], and organizational fit [23]. Some authors also stress the importance of marketing requirements [9,23-26]. Furthermore, project requirements for services are suggested in [21]. As many of the project requirements for hardware and software engineering would apply equally to service projects, we argue for a Project Requirements category.

3.3.2 Functionality requirements Many researchers [9,23,21,24,22,25,26] discuss requirements related to core functions [21], core elements [24], functions [25], capabilities [22], how the outcomes are achieved [9], and service activities [23,26]. Therefore we propose a Functionality Requirements category to account for all requirements related to the ways value is created by a service for a customer.

3.3.3 Development requirements A number of authors [9,23,21,22,24] refer to reference models, procedures, tools [9], and service blueprints [24], or suggest methods for the development of a new service [9,23,21,22,24]. As these concepts play an important role in the developing of a service, we suggest a Development Requirements category.

3.3.4 Operational requirements Many authors [9,23,21,24,22,25,26] account for requirements related to people and IT during the operational phase of a service. Some address requirements for service facilities [23,24,26], service delivery personnel [9,23,24], and monitoring requirements [9,21]. Several also refer to continuous operating finance [25] and cost [9,24,26] requirements. We therefore derive an Operational Requirements category, as all these requirements play an important role in the operational phase of a service.

3.3.5 Evolution requirements A number of authors [23,21,24] refer to service change and improvement issues [23], the modification and extension of services in the future [21], and service reengineering [24]. We therefore derive an Evolution Requirements category that accounts for all foreseeable change and enhancement requirements of a service.

3.3.6 Level of service requirements Researchers [9,23,21,24,22,25,26] frequently identified issues related to quality [9,24-26], service level [23], performance [21,22], availability and reliability [21], profit and cost [23,25], and productivity [9,24,25] as service quality requirements. Moreover, a comprehensive list of general service qualities can be found in [27]. We therefore derive a Level of Service Requirements category to account for all requirements to how well a service meets the stakeholders’ expectations of quality and performance.

3.3.7 Interface requirements Services researchers [9,23,21,24,22,25,26] frequently discuss the ways that services and people interact. A number of authors [9,21,26] also account for requirements related to interrelationships [9], and service interfaces [9,21]. We therefore derive an Interface Requirements category, to
account for requirements that refer to the way people interact with services and service processes interact with each other.

4. A Taxonomy of requirements for hybrid products

Having classified requirements categories from hardware engineering [13-15], complex systems [8,10-12], software engineering [16-20] and the services literature [9,23,21,24,22,25,26], we synthesized a unified taxonomy of requirements for hybrid products.

In the following section we present the high level categories as section headings and list the low level categories within each section. The reader should note that the complete taxonomy, including definitions and examples for each category, can be found in Figure 1.

4.1 Project requirements

As most authors from hardware engineering [8,11-15], software engineering [16,17,20], and service engineering [9,23-26] account for project requirements, we argue for Project Requirements as a top-level category in our taxonomy on hybrid products. “Project Requirements are general constraints and mandates placed upon the design team.” [17] If Project Requirements were left unmet, the proposed hybrid product would not be acceptable or would not satisfy mutually acceptable agreements for success-critical stakeholders [17].

Similar to [8,11-15,17,20], we group Schedule and Budget in the Project Requirements category and define Schedule requirements as all the important dates, milestones, and timelines to meet during the project time. We define Budget requirements as all the cash flows related to the project.

We also group Project Staffing and Project Resources in the Project Requirements category, as the importance of these aspects was especially stressed in software engineering [17,20]. Project Staffing refers to all personnel involved in the project in terms of calendar months [17] and Project Resources are all the equipment needed during the project time [17], e.g. IS/IT, facilities, collaboration software.

As proposed in [12-14,17,20,9,23,24,22,25] we summarized several named project-relevant requirements under a new subcategory Organizational Project Requirements, accounting for requirements related to Strategic Fit [13-15], Marketing [9,23-25,13,14], and Norms, Policies and Legal [12,13,20,18,8,14,17]. Strategic Fit groups all these requirements to advance the strategic goals of the organization. Marketing Requirements are those that enable a transfer of the hybrid product from producers to customers, which could include support for concepts related to pricing models, delivery channels, promotion and communication, branding, and selling the hybrid product. The Norms, Policies, and Legal sub subcategory pertains to social, organizational and legal constraints on the design and operational phase of the hybrid product. For reasons of completeness we also argue for a Partner subcategory, to account for all requirements related to sourcing and supply chain management. We see this as an important subcategory as hybrid products are highly complex [1,2] and as outsourcing has become a common phenomenon [28].

4.2 Functionality requirements

Functional requirements are deemed important by all three disciplines: hardware engineering [8,11-15,10], software engineering [16-20], and service engineering [9,23,21,24,22,25,26]. The second top-level category we therefore find important is Functionality Requirements. We define Functionality Requirements as purposes the hybrid product should serve and the services it should provide [17] as well as its behavior [8]. To reduce complexity, this category is subdivided in Hardware Functionality, Software Functionality, Service Functionality and Hybrid Product Functionality. We argue for the latter as some functional requirements define the whole product (e.g. mobile TV on cell phones) and cannot easily be attributed to a specific part of the hybrid product but fall in two or three of the categories.

4.3 Lifecycle requirements

Requirements differ in lifecycle phases of a product [12]. Furthermore, many authors suggest requirements related to an operational phase [8,10-15] and requirements related to system growth and evolution [8,12-14,16,17]. The Lifecycle Requirements category groups four subcategories, each of them named after a lifecycle phase and accounting for the requirements in this phase: Development Requirements, Operational Requirements, Evolution Requirements, and Retirement Requirements.

4.3.1 Development requirements Similar to [8,10-17,20], Development Requirements account for all requirements and constraints in the development phase of the hybrid product: To account for Development requirements from hardware engineering [8,10-15], we propose a Production and Manufacturing Requirements sub subcategory. Production and Manufacturing Requirements shall contain all requirements related to material, physical constraints, production, finishing, assembly [14], procedure,
Figure 1 A Taxonomy of Requirements for Hybrid Products

<table>
<thead>
<tr>
<th>1. Project Requirements:</th>
<th>General constraints and mandates placed upon the design team. If project requirements were left unmet, the proposed hybrid product would not be acceptable or satisfy success-critical stakeholders.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. Schedule:</td>
<td>Refers to all important dates, milestones, and deadlines to meet.</td>
</tr>
<tr>
<td>1.2. Budget:</td>
<td>Refers to all cash flows related to the project.</td>
</tr>
<tr>
<td>1.3. Project Staffing:</td>
<td>Refers to all personnel involved in the project in terms of competencies, roles, and calendar months.</td>
</tr>
<tr>
<td>1.4. Project Resources:</td>
<td>Refers to all the equipment needed for the project team, e.g. IS/IT, facilities, and collaboration software.</td>
</tr>
<tr>
<td>1.5. Organizational Project Requirements:</td>
<td>Accounts for strategic fit, marketing, norms, policies, legal and partners.</td>
</tr>
<tr>
<td>1.5.1. Strategic Fit:</td>
<td>Groups all those requirements to advance the strategic goals of the organization.</td>
</tr>
<tr>
<td>1.5.2. Marketing Requirements:</td>
<td>Groups those requirements that enable a transfer of the hybrid product from producers to customers; e.g. pricing models, delivery channels, promotion and communication, branding, and selling the hybrid product.</td>
</tr>
<tr>
<td>1.5.3. Norms, Policies, and Legal:</td>
<td>Pertains to social, organizational and legal constraints on the design and operational phase.</td>
</tr>
<tr>
<td>1.5.4. Partners:</td>
<td>Accounts for all requirements related to sourcing and supply chain management.</td>
</tr>
<tr>
<td>2. Functionality Requirements:</td>
<td>Refers to the purposes the hybrid product should serve and the services it should provide and its behavior.</td>
</tr>
<tr>
<td>2.1. Hardware Functionality:</td>
<td>Accounts for all functional requirements related to the tangible part associated with the hybrid product.</td>
</tr>
<tr>
<td>2.2. Software Functionality:</td>
<td>Accounts for all functional requirements which relate to the software associated with the hybrid product.</td>
</tr>
<tr>
<td>2.3. Service Functionality:</td>
<td>Accounts for all functionality requirements which relate to the services provided by the hybrid product.</td>
</tr>
<tr>
<td>2.4. Hybrid Product Functionality:</td>
<td>Accounts for all functions which define the whole hybrid product as a whole and cannot easily be attributed to a specific part, e.g. TV on cell phones.</td>
</tr>
<tr>
<td>3. Lifecycle Requirements:</td>
<td>Accounts for the different requirements in each of the phases in the lifecycle of the hybrid product.</td>
</tr>
<tr>
<td>3.1. Development Requirements:</td>
<td>Requirements and constraints that have to be considered during the developing phase.</td>
</tr>
<tr>
<td>3.1.1. Production and Manufacturing Requirements:</td>
<td>Relates to all hardware requirements in the development phase, e.g. material, physical constraints, production, finishing, assembly, procedure, standards, and equipment.</td>
</tr>
<tr>
<td>3.1.2. Implementation and Software Requirements:</td>
<td>Relates to all software requirements in the development phase, e.g. computer hardware, computer communications, developing language, software tools, standards, and implementation.</td>
</tr>
<tr>
<td>3.1.3. Service Development Requirements:</td>
<td>Relates to service requirements in the development phase, e.g. tools, methods, and reference models.</td>
</tr>
<tr>
<td>3.1.4. Overall Development Requirements:</td>
<td>Relates to requirements in the development stage not attributable to a single category mentioned above, e.g. distribution, logistics, verification, packaging, warranty, deployment, change and central management, setting up an adequate organization, and controlling.</td>
</tr>
<tr>
<td>3.2. Operational Requirements:</td>
<td>Accounts for all requirements that have to be fulfilled during the operational phase of the hybrid product and for resources needed to run the hybrid product.</td>
</tr>
<tr>
<td>3.2.1. Environment Requirements:</td>
<td>Refers to all requirements related to environmental conditions, e.g. the country the hybrid product can be used in, electricity infrastructure, and working conditions.</td>
</tr>
<tr>
<td>3.2.2. Ongoing Staffing Requirements:</td>
<td>Refers to all requirements related to personnel issues during the operational phase.</td>
</tr>
<tr>
<td>3.2.3. Ongoing Resource Requirements:</td>
<td>Refers to all requirements related to ongoing resource needs, e.g. IS/IT, facilities.</td>
</tr>
<tr>
<td>3.3. Evolution Requirements:</td>
<td>Accounts for all the foreseeable directions of growth, change or enhancement of the hybrid product.</td>
</tr>
<tr>
<td>3.3.1. Functionality Evolution Requirements:</td>
<td>Accounts for functionalities that might become important in the future.</td>
</tr>
<tr>
<td>3.3.2. Interface Evolution Requirements:</td>
<td>Accounts for how the hybrid product must adapt to interface changes.</td>
</tr>
<tr>
<td>3.3.3. Level of Service Evolution Requirements:</td>
<td>Accounts for expectations related to the improvement of qualities of service.</td>
</tr>
<tr>
<td>3.3.4. Technology Evolution Requirements:</td>
<td>Addresses how the system should react to evolutions of underlying technologies.</td>
</tr>
<tr>
<td>3.3.5. Environmentally Driven and Workload Evolution:</td>
<td>Addresses expected growth, system usage, and changing social and legal requirements.</td>
</tr>
<tr>
<td>3.4. Retirement Requirements:</td>
<td>Captures all requirements related to removing the hybrid product from the market, dismantling and disposing of its infrastructure as well as reassigning its operational personnel and resources to other purposes.</td>
</tr>
<tr>
<td>4. Interfaces Requirements:</td>
<td>Accounts for all the applicable requirements on how the hybrid product should interact with users, hardware, software, and services for input and output.</td>
</tr>
<tr>
<td>4.1. Human Interface Requirements:</td>
<td>Accounts for all interfaces, the hybrid product presents to the user, e.g. GUI, access by telephone.</td>
</tr>
<tr>
<td>4.2. Hardware Interface Requirements:</td>
<td>Accounts all hardware equipment the hybrid product interacts with, e.g. scanner, bar code reader, and communication devices.</td>
</tr>
<tr>
<td>4.3. Software Interface Requirements:</td>
<td>Accounts for all interoperations with software programs, tools, databases, and external systems.</td>
</tr>
<tr>
<td>4.4. Service Interface Requirements:</td>
<td>Accounts for how service processes interact, e.g. for technical and organizational compatibility.</td>
</tr>
<tr>
<td>5. Level of Service Requirements:</td>
<td>Accounts for all requirements related to how well given requirements should be performed.</td>
</tr>
<tr>
<td>5.1. Hardware Level of Service Requirements:</td>
<td>Accounts for hardware quality requirements, e.g. performance, features, reliability, conformance, durability, serviceability, aesthetics, and perceived quality.</td>
</tr>
<tr>
<td>5.2. Software Level of Service Requirements:</td>
<td>Accounts for software quality requirements: e.g. general qualities, coherence, dependability, security, confidentiality, safety, interoperability, usability, performance, adaptability, and reusability.</td>
</tr>
<tr>
<td>5.3. Service Level Requirements:</td>
<td>Accounts for service quality requirements: e.g. reliability, responsiveness, competence, access, courtesy, communication, credibility, security, understanding, and tangibles.</td>
</tr>
</tbody>
</table>

* See [32] for definitions; ** See [17] for definitions; *** See [27] for definitions
4.3.3 Evolution requirements Most authors from the three disciplines involved in hybrid products: hardware engineering [8,10-15], software engineering [16,17], and service engineering [23,21,24] argue for requirements related to evolution and system growth. Therefore, we suggest a subcategory Evolution Requirements which shall refer to all the foreseeable directions of growth, change or enhancement [17] of the hybrid product. As a subcategory we suggest Functionality Evolution Requirements. This category accounts for design choices in functionality that might become important in the future [17]. Interface Evolution Requirements shall account for how the product must adapt to interface changes [17]. Level of Service Evolution Requirements shall describe expectations related to improvements of qualities [17]. Technology Evolution Requirements shall address how the system should react to evolutions of the underlying technology [17], and Environmentally Driven and Workload Evolution Requirements shall account for projected growth of system usage and scalability [17]. These subcategories are the same as in the taxonomy proposed by [17]. We deem the structure proposed in [17] also suitable for hybrid products, as it is similar to the top-level categories of the taxonomy and as it also takes general Evolution Requirements (Technology, Environment, and Workload) requirements into consideration.

4.3.4 Retirement requirements Some authors suggest requirements related to retirement or disposal [13,14]. Therefore, we argue for Retirement Requirements as these capture all requirements related to removing the hybrid product from the market and dismantling and disposing of its infrastructure as well as reassigning its operational personnel and resources to other purposes.

4.4 Interface requirements

The importance of interfaces is especially stressed by authors in software engineering [16,17,20] and service engineering [9,23,21,24,22,25,26]. We therefore adopted Interface Requirements as a top-level category. We adapted Boehm’s definition [17] of interfaces for hybrid products and define Interface Requirements as any applicable requirements on how the hybrid product should interact with users, hardware, software or services for input and output.

To reduce complexity we subdivided this top-level category into Human Interface Requirements, Hardware Interface Requirements, Software Interface Requirements, and Service Interface Requirements. We suggest a Human Interface Requirements subcategory to account for all user interfaces the hybrid product presents to the user [17], e. g. graphical user interfaces and access by telephone. We further define a Hardware Interface Requirements subcategory to account for all hardware equipment the hybrid product interacts with, e. g. other hardware components [8], scanner, bar code reader, communication devices [17], bridges, and power supply. We also suggest a Software Interface Requirements subcategory accounting for the
hybrid products’ interoperations with software programs, tools, databases, and external systems. A Service Interface Requirements subcategory shall account requirements related to how service processes interact, e. g. technical and organizational compatibility among services [21].

4.5 Level of service requirements

All Authors from each of the three disciplines identify requirements related to quality. These requirements are akin to those called “the -ilities” in the software engineering literature – e. g. reliability, maintainability, flexibility [30,31]. Therefore, we suggest Level of Service Requirements as the fifth top-level category in our taxonomy. Following [17], we define Level of Service Requirements as how well the hybrid product should perform the given requirements.

To reduce cognition overload we subdivided this category into Hardware Level of Service Requirements, Software Level of Service Requirements and Service Level Requirements. Literature on hardware engineering [8,10-15] doesn’t offer a consistent set of quality of service requirements. In our taxonomy, we refer to the qualities suggested by [32] (cf. Appendix 1) as [32] proposes general hardware qualities applicable to a wide range of products. For software qualities we adopted the complete set from [17] in our taxonomy, as these qualities proposed were useful in hundreds of facilitated workshops [4]. For Service Level Requirements we refer to [27] who offer a set of ten determinants of service quality (cf. Appendix 1). This selection might be very useful as it includes general service qualities; however, it might be adapted for specific hybrid product projects as performance measures for services can vary widely [23].

5. Findings from the interviews

The experts stated that they found the taxonomy “complete”, “holistic”, “intuitive” and “exhaustive” at first sight. One interviewee mentioned that due to the exhaustiveness “both the buyer and the seller can use the taxonomy for hybrid product projects”. Moreover, several of the experts explained that they appreciate the separation of project and ongoing requirements, the emphasis on functionality over technology, and the inclusion of level of service requirements. One expert suggested changing the order of Project Requirements and Functionality Requirements for her specific domain. The experts find the category definitions clear. However, some of them propose minor changes and additions to improve the practical usability. For example, the Environment and Workload Evolution Requirements category was suggested to be renamed to Environmentally Driven and Workload Evolution Requirements as it is mostly influenced by external conditions. Also, two experts pointed out the importance to account for changing legal situations in the definition of Evolution Requirements. Another two experts suggested accounting for change and release management in the taxonomy.

The experts could identify categories that are especially important in their respective industries. Experts with an IT and telecommunications background deemed Level of Service Requirements and Functionality Requirements especially important. Experts with an engineering or medical background stress the importance of Project Requirements, Lifecycle Requirements and Interface Requirements. Superfluous requirements and requirement categories were mentioned by none of the experts. Furthermore, the experts did not see any missing categories to be included. However, in addition to the corrections of the definitions, one expert mentioned that she was searching for replacement requirements first, but finally wanted the Retirement Requirements definition to also account for replacing requirements. Three experts had difficulties in discerning Functionality Requirements and Interface Requirements at first, but reported that the definitions solved the ambiguities.

The experts were able to find several uses for the taxonomy. One export asked to use the taxonomy as a completeness check for the requirements analysis in a project to develop a personal health manager. Another expert wanted to use the structure for brainstorming requirements when introducing a new IT system to her company. Two experts reported that they could think of the taxonomy as a project management guide. Moreover, the taxonomy was deemed useful in multiple stakeholder negotiations, as, due to its exhaustiveness, it might allow stakeholders to better see conflicting requirements.

6. Discussion

The proposed taxonomy is derived from the three disciplines involved in hybrid products: hardware engineering, software engineering, and service engineering. People coming to a hybrid product project will often have expertise in one of the three disciplines, but may be unfamiliar with the other two perspectives. This taxonomy is intended to make practitioners aware of the larger scope and the complexity of hybrid product projects. It might also be useful as a completeness check for people who might otherwise solely focus on their own domain.

Moreover, the proposed taxonomy follows a general approach towards RE for hybrid products as it is designed to be neither industry- nor stakeholder-
specific. Therefore it should be applicable in a wide range of hybrid product projects.

Practitioners can gain value from the taxonomy immediately by embedding it in existing collaborative requirements methods from the software field. There are already approaches existing [4-7], which might simply be adapted by replacing software taxonomies with the proposed hybrid product taxonomy.

While embedding the taxonomy in existing collaborative requirements methods might be helpful, these methods are software-centric. Therefore, more research will be required to adapt and optimize the taxonomy for the more the complex challenges of hybrid product requirements negotiations.

In order to be regarded as a taxonomy, a requirements negotiating scheme provides placement of requirements in one and only one requirement. More research will be required to determine the extent to which the proposed structure is taxonomic.

Furthermore, while the expert’s insights provided are useful, more extensive field work is required here to further validate, extend, and define the taxonomy. In a next step, the acceptance of the taxonomy should be more rigorously explored. It might be useful to apply the taxonomy in real hybrid product development projects. Here, case study research or Action research might be adequate.

That said, there might also be a downside to the general approach: The taxonomy might deliver useful, but not optimal results. Further research may reveal that industry-specific taxonomies would be useful for better results.

In order to determine quality of hybrid products, further research is also needed on level of service measures. Elaboration of this topic in the literature is sparse. Additional “-ilities” may be needed. It may also be useful to combine “-ilities” to derive higher-level metrics for hybrid products as a whole.

7. Conclusion

We drew from a wide range of related literature to synthesize this taxonomy, combining essential parts of existing taxonomies and new concepts from additional literature. Five top-level categories were designated for the hybrid product taxonomy, including Project Requirements, Functionality Requirements, Lifecycle Requirements, Interface Requirements, and Level of Service Requirements. In a first step to validate the taxonomy, we also presented it to experts in the field of hybrid products.

This paper contributes a novel, single expression of requirements for hybrid products. We combined overlapping concepts from each of the contributing disciplines (e.g., service use and level of service requirements), but also maintained domain specific concepts. We conclude that it is likely that there are requirements that are more than just the sum of three parts and define the hybrid product as a whole.

This contribution is intended to fill the gap where a unified taxonomy suitable for hybrid products has been lacking. We hope that practitioners find our taxonomy useful in collaborative requirements workshops and other settings. However, we also point out that more rigorous research might be useful, to further explore the taxonomy’s acceptability in the field. Researchers might use this taxonomy as a starting point for the development of hybrid product RE models.

8. References


References:


Appendix 1:

1) What is your first impression of the taxonomy?
2) Is there something unclear? Is the structure intuitive for you?
3) Do you have questions about the definitions?
4) Can you identify requirement categories that you find important for your discipline?
5) Are there requirements/ requirement categories missing?
6) Do you think there are superfluous requirement categories?
7) Can you think of requirements that fit into more than one category?
8) Would this taxonomy be useful for your projects/ research?
9) Why would this taxonomy be useful in a setting where requirements are being generated and negotiated by multiple stakeholders?
10) Is there something else you want us to know?