On the Relevance of the Modularity Concept for Understanding Outsourcing Risk Factors

Philip Huysmans
University of Antwerp
philip.huysmans@uantwerp.be

Peter De Bruyn
University of Antwerp
peter.debruyn@uantwerp.be

Shazdada Benazeer
University of Antwerp
shazdadamd.benazeer@ua.ac.be

Alain De Beuckelaer
Radboud University Nijmegen
a.debeuckelaer@fm.ru.nl

Steven De Haes
University of Antwerp
steven.dehaes@uantwerp.be

Jan Verelst
University of Antwerp
jan.verelst@uantwerp.be

Abstract

Contemporary organizations frequently seek external and specialized partners to outsource some of their non-core, though highly complex tasks, including their information systems. In literature, a variety of outsourcing risk factors which lead to unsuccessful project outcomes have been identified. Despite many remedies suggested by practitioners and scholars to mitigate these risk factors, empirical studies continue to report frequent failures in outsourcing projects. This paper suggests an alternative perspective to analyze risks related to outsourcing projects, based on the concept of modularity. Such a new perspective can help in complementing existing outsourcing risk analyses with new or deeper insights as a basis to define a more exhaustive list of required mitigating actions, which in turn could lead to more succesful outsourcing projects. To illustrate this alternative perspective, a reanalysis of a failed outsourcing case is discussed. The paper shows in a detailed way how poorly designed modular structures at the technical and project level could have been identified ex-ante. This identification may explain the manifestation of ex-post outsourcing risk factors such as 'lack of required skills', 'managing user expectation' and 'communication problems'.

1. Problem statement

In the contemporary volatile global business environment, it is difficult for many organizations to adapt to the rapid pace of technological change, manage its complexity and (at the same time) achieve economies of scale. Consequently, many organizations seek external and specialized partners to hand over (i.e., outsource) some of their non-core, though highly complex tasks [14], including the outsourcing of their Information Systems (hereafter IS) development and maintenance. Many motivations and perceived benefits for this IS outsourcing have been identified, including increased flexibility, benefit from vendors' economies of scale, specialization in core business activities, reduced risk, and access to technical expertise [19].

An outsourcing project constitutes a complex undertaking, and an astonishing high number of IS outsourcing projects fail to achieve their intended targets. For instance, a study of Gay and Essinger [8], which examined 29 major outsourcing engagements over a period of eight years, concluded that more than 35 per cent of the engagements failed. Scholars and practitioners suggested many mitigation actions (e.g., a high quality service level agreement) that may, without any doubt, help in reducing the failure rate. Nevertheless, a large number and variety of failures continues to occur. This might indicate that sometimes, not all relevant mitigating actions to address outsourcing risks were taken, due to the fact that not all risks could be clearly identified ex-ante.

2. Research question

Building on the problem statement as discussed in previous section, this paper proposes an alternative approach, based on the concept of modularity, to identify outsourcing risks ex-ante.

The modularity concept states that a system can be decomposed into several interrelated subsystems (i.e., modules). This concept, originating from systems theory, turned out to be a successful means for controlling complex systems in several domains, including product design, software design, etcetera [1]. Because outsourcing projects are generally conceived as being complex, especially in dynamic environments, it seems reasonable to explore the possibilities of using the modularity concept in addressing outsourcing complexities. As a matter of fact, several scholars have recently suggested a possible link between the general

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.544

concepts of 'modularity' and 'outsourcing' (see e.g., [4,20]). More specifically, various authors link modularity to the explanation of outsourcing failures [9,24]. However, many of these scholars only identify this link while the call for more in-depth case analyses remains unresolved.

Given this lack of in-depth research, it seems plausible that such an alternative perspective can help in complementing existing outsourcing risk analyses with new and deeper insights, as such providing a more complete basis to define a more exhaustive list of mitigating actions. The research question of this paper can therefore be formulated as follows: *How can the concept of modularity help in obtaining a better ex-ante understanding of outsourcing risks?* To answer this question, this paper analyzes the BSkyB outsourcing case: a failed outsourcing project between EDS (the vendor) and BSkyB (the client), of which extensive and objective case details are publicly available in the form of court proceedings. Even though this outsourcing case has already been scrutinized by some scholars [21], to the best of our knowledge no profound case analysis has discussed this case in relation to the modularity concept.

3. Methodology and case introduction

The study relied on case study methodology because this qualitative methodology is particularly suited to investigate a phenomenon (here: a large-scale outsourcing project) in its natural environment, a contemporary business context [2,22]. Because of the selected methodology this study may be expected to provide actionable results, as well as in-depth insights as to how the concept of modularity can be successfully applied in order to better analyze failure risk of outsourcing activities. Since theory-building work on the importance of modularity in outsourcing projects is scarce, a descriptive (rather than an explanatory) one-case study design (i.e., relying on just one unit of analysis) was deemed to be a good starting point. While we acknowledge the limited generalizability of a one-case study design, a descriptive approach is required to gain sufficient in-depth insight before more confirmatory research projects are initiated.

Four main criteria underlie the selection of a suitable project for this paper. First, the outsourcing project should have been a failure as especially such project may provide insight into those elements that make a successful completion of an outsourcing project challenging. Second, the project had to be large enough so that a certain amount of complexity is involved. Third, sufficient documentation of the project needed to be available to allow detailed examination of the applicability of modularity. Fourth, only outsourcing projects that have already been presented and discussed in earlier studies were considered. Inclusion of this fourth criterion implies that key issues of the outsourcing project examined in this study correspond to those key issues which have been identified in earlier studies. Obviously, the perspective taken in the earlier studies differs from the one taken in this study, namely a perspective with a central focus on the modularity concept. Based on these four criteria the 'BSkyB outsourcing case' was selected. The project described in this case concerns an outsourcing deal on the development of a Customer Relationship Management (CRM) system by Electronic Data Systems (EDS) for the British Sky Broadcasting (BSkyB) group. BSkyB appointed EDS in 2000, estimating that the project would take 18 months and cost £48 million. However, the project was considered a failure, and the contract was terminated in December 2002 (cf. criterion one). By then, BSkyB had already spent £170 million. Eventually, the project would take six years to complete and cost £265 million. Conflicts regarding the proper project execution resulted in the filing of a case at the London Technology and Construction Court. The case ended in July 2008, but the judgment was only finalized in January 2010 because of the complexity of the case (cf. criterion two). BSkyB was awarded £318 million. The court proceedings are publicly available [10] (cf. criterion three). An earlier discussion of the risks involved in this outsourcing project was provided by Verner and Abdullah [21] (cf. criterion four).

Data analysis was based on the framework proposed by Miles and Huberman [13]. First, *data collection* only concerned the collection of secondary data: the court proceedings and additional information provided by the media. All data were primarily analyzed in relation to the concept of modularity. To facilitate *data reduction*, a list of modularity aspects (see Section 4) derived from key publications on modularity (e.g., [1,4]) was used to code the court proceedings. Next, codings made by two authors were compared, discussed and iteratively refined. Subsequently, excerpts from the coded data were displayed. Data triangulation was performed by comparing coded court proceedings with press releases and grouping the coded data excerpts using the outsourcing risk factors identified by Verner and Abdullah [21]. Finally, three risk factors were selected to discuss in this paper (i.e., lack of required skills, managing user expectation and communication problems). These risk factors were selected because, according to Verner and Abdullah, they were related to the two largest claims in the court case (the so-called 'proven technology representation' and the 'significant risk representation'). This selection guarantees the analysis of outsourcing risk factors which were highly relevant in this case, rather than presenting risk factors which provided the best results for our approach. Consequently, *conclusions* were drawn by interpreting the excerpts and making explicit links between the outsourcing risk factors and the modularity aspects.

4. Theoretical background

Even though originating from systems theory, the modularity concept has caught the attention of engineers, managers and researchers in a large variety of fields [1]. *Modularity* is defined as a property of a complex system, whereby the system is decomposed into several subsystems (i.e., modules). Obviously, each of these modules ultimately must cooperate with other modules in order to ensure the adequate functionality of the system as a whole. The interaction of a module with its external environment should therefore be exhaustively and unambiguously documented in its *interface*. The interface describes the inputs required by the module to perform its part of the functionality, and the output it will provide to its external environment (which includes other modules in the system). As soon as such an interface is designed, one may learn about the *intermodular dependencies*, i.e., what does a module require from the other modules to function and what is the impact of a change in the module design for other modules.

Modular dependencies can be understood better if their relation with the design of a system is elaborated upon. The design of a system module can be made explicit through its design parameters, which describe certain characteristics or design decisions [1]. Ideally, each design parameter value should be determined optimally for a given design. However, a certain design decision for a parameter can force certain design decisions for other parameters. For example, when limited resources such as space in a product design are attributed to a certain component, other components may not be able to claim these resources as well. The amount of dependent design parameters within a module determines its coherence. A good modular design requires high intramodular cohesion, which means that a high amount of dependent design parameters should be captured within a module. This means the complexity caused by a certain function of the system can be captured within a module. The amount of dependent design parameters between different modules determine their coupling. Good modular design requires low coupling, which means that design parameters should not be impacted by design parameters from other modules. Therefore, from a modularity perspective, good design is characterized by: (a) low intermodular coupling (i.e., few intermodular dependencies), and (b) high intramodular cohesion (i.e., strongly related and dependent elements within a module). From a practical point of view, this means that: (a) changes in the design of one module have no or only a limited impact on the design of other modules, and (b) the function of one module can be studied more or less in isolation from the rest of the system. Consequently, a well-designed modular system enhances the comprehensibility and decreases the complexity of the overall system.

While numerous advantages are associated with good modular design, increased flexibility is probably the most prevalent one. Baldwin and Clark [1] captured this notion of flexibility in *modular operators*, which can be applied to modular structures (i.e., splitting, substituting, augmenting, excluding, inverting, and porting). For instance, the substitution operator states that it should be relatively easy to replace an old version of a module by a newer (e.g., better) one and, this way, ameliorate the overall performance of the system. As a result, the system as a whole can be reconfigured easily by applying the substitution operator. Logically, many intermodular dependencies (i.e., high coupling) hinders the application of the operators. Actually, *all* intermodular dependencies should be made explicit (i.e., no 'hidden dependencies' are allowed) to confidently apply such operator. By explicitly prescribing a design decision for an intermodular dependent design parameter, the impact of this dependency can be removed. For example, it can be prescribed that a product component needs to be restricted to a certain spacial location. Other components can then be designed around this location, and no interference between the spatial locations of such components will occur. When such design decisions are made explicit for all dependencies, a set of prescriptive rules is obtained, which all modules of that system need to adhere to. This set of rules is called a *modular architecture*. Therefore, a modular product architecture describes the rules which need to be adhered to by modules in order to be able to include them in that particular product.

Unfortunately, designing a modular architecture is far from trivial because many decisions have to be taken. Hence, a theory which prescribes principles to guide the design of a good modular design is highly desirable. In this context, an often referred to principle is *separation of concerns*. This principle prescribes to separate elements which can change independently (i.e., concerns) in distinct modules. For instance, Normalized Systems theory [12] relies on this principle to create fine-grained modular software designs.

The concept of modularity has already been applied in a large amount of management studies. Campagnolo and Camuffo (2010) presented a literature review of 125 studies which applied modularity on the product, production system, or organizational level. Moreover, many authors have related the concepts modularity and outsourcing to each other. An overview of several conclusions of research regarding this relation is presented in Table 1. Nevertheless, we already argued that it is hard to design good modular architectures. The design of modular architectures on different levels is even harder in comparison with a single level. This difficulty is not adequately addressed in theoretical studies on organizational modularity. These studies focus, on the one hand, on the benefits of flexibility and complexity reduction, and on the other hand, on the risks of interdependencies and product imitation. An explicit application of the theories, which prescribe design guidelines for the different levels and for their integration, is lacking. Given the complexity of the design of modular architectures, we believe that a more precise analysis needs to be made of how the modularity concept impacts design on these levels, and how violations against these prescriptions can help understand outsourcing issues.

Table 1: The relation of outsourcing and modularity concepts.

"Overall, modularity is believed to help firms manage outsourcing efficiently and effectively thus facilitating the integration of external sources of innovation" [3].
"Interface standards and modularity, of course, facilitate outsourcing and thereby sharpen requirements for integration" [24:23].
"As the relationship between modularity and outsourcing exists although the direction is debated" [4:277].
"We conclude that understanding the true complexity of inter-firm relations may lead managers to refrain from outsourcing altogether. We suggest that managers may consider implementing a modular organization design to limit complexity and thus facilitate outsourcing" [9:2].
"In the past, 'modularity' and 'outsourcing' were investigated predominantly in separate research communities. More recently, however, a research stream has emerged that links these two topics together" [7:167].
"On the whole, however, outsourcing, task partitioning, standardization and knowledge encapsulation, although conceptually distinct, remain strictly intertwined in practice, since the evidence coming from the field shows that, especially within global strategies, modularization and outsourcing are becoming increasingly inseparable" [5:8].

5. Findings: interpreting outsourcing risk factors in terms of modularity aspects

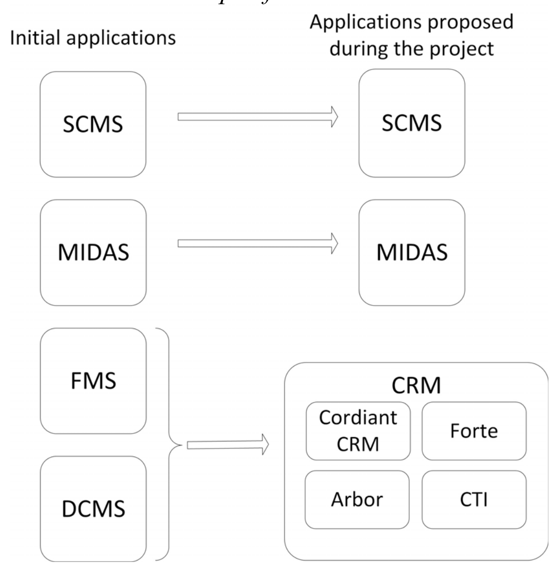
In the BSkyB outsourcing trial, a crucial accusation concerned the alleged fraudulent misrepresentation of the vendor's own capabilities [10:4]. More specifically, the vendor would have claimed to be able to complete the project according to specified functionalities and within budget and time constraints. In a legal context, this is a significant issue. However, based on insights from modularity reasoning, one may demonstrate that: (1) the vendor could have realized that this claim is both risky and hard to redeem, and (2) the client could have realized that the claim was not realistic. To illustrate this, two root causes underlying the project failure are elaborated upon, namely the technical complexity in the project and the project team composition. The analysis below shows that by describing modular structures at this level, some of the root causes of these ex-post identified risk factors can be uncovered. Therefore, we hypothesize that identifying these modular structures upfront, might help in a better ex-ante understanding of outsourcing risks.

For each of our selected root causes, we first briefly highlight some relevant parts of the available case study description. Next, we describe how this part of the case description could be interpreted in terms of modularity aspects and what would considered as a good modular design. Violations against these modularity requirements will be shown to provide an explanation of the manifestation of each of the risk factors.

5.1. Lack of required skills due to technical complexity

From a technical perspective, the context in which the project was embedded was complex. Before the project was initiated, several legacy applications handled the customer relationship management functionality at Sky [10:9–10]: (1) 'DCMS', the Digital Customer Management System, (2) 'SCMS', the Subscriber Card Management System, (3) 'MIDAS', the Management of Information for Digital and Analogue Systems, and (4) 'FMS', the Field Management System. First, the new CRM system needed to replace some of these legacy applications (i.e., DCMS and FMS) while another application remained operational and needed to interface with the new CRM application (i.e., SCMS) [10:11–12]. Second, as depicted in Figure 1, different software technologies were used by the vendor to build the new CRM system: 'Chordiant CRM', 'Arbor' as billing software, 'CTI' for the call centers and 'Forte' as a development framework and a middleware product [10:49–50]. The alleged failure to deliver a promised seamless integration of all these (new) technologies, was referred to as the '*proven technology representation*' in the court proceedings.

Figure 1: The modular structure of the application portfolio.



5.1.1. Identifying the modular structure and requirements

From a modularity point of view, the applications represent separate modules at a coarse-grained level. In Figure 1, we represent these application modules as rounded boxes. The modular operator 'substitution' is applied to a set of two modules as they are replaced by a new module. As discussed in Section 4, the successful application of a modular operator is conditional on the assumption that no hidden modular dependencies are present. Hence, besides dependencies listed in the interface specification of the SCMS system in the Invitation to Tender (ITT) [10:11], no other dependencies are allowed for. This requirement is referred to as *modularity requirement 1.1*.

Moreover, the internal structure of the new module (i.e., the new CRM system), constitutes a modular structure as well. Here, the different technologies (see Figure 1) represent modules, which – after integration – should deliver a well-functioning CRM system. As it is widely acknowledged that each technology is usually prone to many and different reasons for change, one should consider these modules as changing independently [12]. Indeed, each technology is maintained by a different vendor and has its own product life cycle [10:1507]. When, during the development of an application, these technologies become interwoven and are not properly separated from one another, changes in any technology (i.e., one module) may require adaptations to all other technologies (i.e., the other modules). Therefore, based on the 'separation of concerns' principle, modularity requires a systematic way (e.g., a project methodology) to integrate these technologies, without the introduction of unnecessary modular dependencies; this requirement is referred to as *modularity requirement 1.2*.

5.1.2. Assessing the modularity requirements

The court proceedings attested to the presence of numerous hidden dependencies. For example, a new log-on method from Chordiant resulted in instability issues in various legacy applications [10:1236]. Different components of the CRM system (such as Arbor and CTI) and legacy applications (such as SCMS), make log-on calls to Chordiant. Consider now a modification to how Chordiant requires a call to be made. Such modification is likely in projects like this, especially because the project's requirements were not well-defined upfront and labeled "unclear", "inadequate" and "ambiguous" [21]. A change in the way this call is made, resulted in (possibly multiple) changes in each application, implying that modular dependencies exist. Moreover, given the unconscious nature of these dependencies, the impact of this change is unknown upfront. Hence, the absence of an explicit specification of all modular dependencies in the systems structure shows that modularity requirement 1.1 was not met.

Once modular dependencies would be made explicit, it is still extremely unlikely that these modular dependencies can be adequately dealt with, unless a suitable project methodology is used. Normalized Systems theory (see Section 4), for instance, explicitly classifies the joint reliance of one module on two or more external technologies as a violation of the 'separation of concerns' principle [12]. The theory emphasizes the importance of a systematic approach to obtain fine-grained modular structures, by isolating each independently changing element (i.e., concern). Even though the vendor claimed that a suitable software development life cycle methodology was employed [10:1029–1061], later on it became obvious that a variety of methodologies were adopted, and that "*no overarching methodology was applied consistently*" [10:1316]. Consequently, no specific and thorough approach to deal with modular dependencies was relied upon and modularity requirement 1.2 was not met.

5.1.3. Understanding the outsourcing risk factor through modularity reasoning

The violation regarding modularity requirement 1.1 (i.e., an interface documenting all possible modular dependencies) explains why the technical complexity of the overall project was underestimated by both vendor and client. The overall technical complexity was present due to the many undocumented intermodular dependencies which were not taken into account and thus (at least partially) ignored. This resulted in unrealistic capability requirements and expectations during project implementation. This thought was also worded eloquently by an employee who stated that "*the problems facing the project had not been caused by an incompetent, under-resourced delivery team but that the original bid team had under-estimated project size and complexity, with the result that no delivery team, however competent, could have succeeded in the timeframe available*" [10:1143].

In addition, the violation regarding modularity requirement 1.2 (i.e., an overarching methodology resulting in a fine-grained modular structure adhering to separation of concerns) more specifically explains how the technical complexity for individual employees was underestimated as well and why they were expected to deal with these dependencies. For instance, due to the lack of a structured approach or methodology enforcing separation of concerns, an employee who needs to implement the CTI technology will also require in-depth knowledge of the implementation details of the others packages, such as Chordiant. However, the capabilities required to deal with such dependencies are unlikely to be possessed by a single employee, especially because knowledge regarding (the combination of) different technologies is needed and each technology is owned by a separate organization. Considering the modular structure of the applications, it then becomes clear that adequately skilled employees will not be available.

However, according to the client, the vendor claimed that its employees "*had the technical experience, knowledge and expertise to integrate and implement the proposed technical solution that they had recommended and that such experience, knowledge and expertise would be available and applied to the project*" [10:975]. From our modularity reasoning, we know that this statement seems to be both unfounded (cf. violation of modularity requirement 1.1) as well as highly unlikely to be realistic (cf. violation of modularity requirement 1.2). Consequently, the failure of the vendor to provide adequate resources was seen as an important factor contributing to project failure. Verner and Abdullah [21] categorized this phenomenon ex-post under the risk factor 'lack of required skills' to successfully complete the project and cope with its associated (technical) complexities.

In terms of our research question as defined in Section 2, we believe that this risk factor could have been better understood and anticipated upon ex-ante by analyzing the case upfront through the modularity reasoning (and its resulting requirements) we presented here. First, the lack of documented intermodular dependencies should have withheld the vendor to utter the straightforward feasibility of integrating the multiple technologies and should have alarmed the client for the credibility of this claim. Second, not adopting one consistent project methodology facilitating the systematic separation of different concerns into distinct modules, should at least have put serious question marks regarding the possibility to find employees able to cope with the subsequent complexity of handling implementation details of different technologies at the same time.

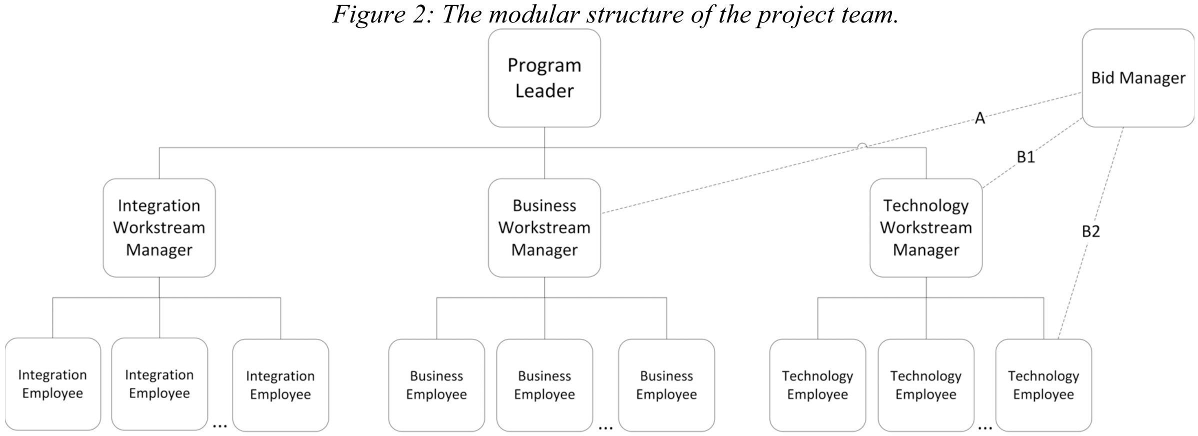
5.2 Project communication

This section shows that non-technical root causes, such as the project team composition, can be interpreted and explained in terms of modular structures as well. In the initial ITT, the project team structure was decomposed in different 'workstreams' (i.e., a business, technical, and transition workstream), with each workstream composed of "*competent and qualified staff*" and supervised by a workstream manager [10:66, 83]. In turn, these workstream managers had to be supervised by an overall program manager, who "*would have overall responsibility for all work within the scope of the ITT and associated contracts*" [10:66]. At the vendor's side, the main responsibility for project communication was attributed to the bid manager, since he "*was also the conduit for communications between EDS and Sky*" [10:1019]. The alleged failure of the vendor to adequately inform the client about the project risks, was coined as the '*significant risk representation'* in the court proceedings.

5.2.1. Identifying the modular structure and requirements

From a modularity perspective, one can interpret the different roles (each having their own responsibility, competence and authority) in the project team as modules. For example, a method to identify modular actor roles has been proposed by Dietz [6]. In Figure 2, we represent these role modules as rounded boxes. When examining project communication flows, one learns that all project information provided to the

Figure 2: The modular structure of the project team.



client is attributable to the bid manager. From a modularity perspective, two possible designs are available for the information exchange. In a first design, the bid manager should possess all required knowledge (i.e., from the business, technology and transition workstream) to handle all communication with the client adequately. This would result in a poor modular design, because the degree of cohesion of the 'bid manager module' would be very low. Additionally, as discussed earlier on, employees with such diverse knowledge base are virtually non-existent. Alternatively, in a second design, the 'bid manager module' could be regarded as an 'interface'. As an interface, the bid manager would then provide the required functionality (i.e., communication with the client), but would rely on other modules (i.e., workstreams) to react to requests. This is represented in Figure 2 by the dotted lines. These lines indicate that the bid manager can retrieve information from workstream managers (e.g., case A or B1), or from specialized employees (e.g., B2). Given the heterogeneity of relevant knowledge, this second design would be preferable from a modularity perspective because specialist knowledge is now embedded in specific modules (thereby increasing intramodular cohesion). Therefore, one should be able to detect a mechanism of how the adequate information is requested from (for example) members of the technology stream team, when a request for information concerning technology arrives at the bid manager. The presence of such mechanism is referred to as *modularity requirement 2.1*.

5.2.2. Assessing the modularity requirement

Content analysis of the court proceedings revealed that an interface mechanism as described above was not in place. When asked whether technical risks were involved in the CRM project, the vendor responded that it was "*believed* [that] *the solution could be delivered in the time and for the budget*" [10:1019]. However, when asked for a justification for this belief, no proper evaluation or justification by way of proof-of-concept or documentation could be provided [10:1020]. Instead, the bid manager argued: "*I was comfortable that the technical team were confident to make appropriate decisions and trust them accordingly*" [10:1021]. Therefore, one must conclude that the role of bid manager was not fulfilled as an adequate interface. Rather, the bid manager communicated based on his own (insufficient) knowledge concerning technological topics. He stated that: "*I certainly did not feel we had a risk here*" [10:1021]. In sum, the analysis supports the conclusion that (in any case) a poor modular structure was in place because either (1) a modular design with low cohesion was implemented, or (2) a dysfunctional interface was relied upon. Therefore, it is concluded that modularity requirement 2.1 was not met.

5.2.3. Understanding the outsourcing risk factor through modularity reasoning

The violation regarding modularity requirement 2.1 (i.e., a communication mechanism assuring factual correctness and hence, high intramodular cohesion by means of an interface) explains why, at the vendor side, the bid manager was not able to provide the correct information as requested by the project leader. This was due to two reasons. First, it could simply not be deemed reasonable to expect from the bid manager to master the extremely diverse knowledge base to provide the requested answers in a well-founded way. Second, no communication mechanism was in place to retrieve the required information from other actors (i.e., modules) in his organization. Consequently, at the client's side, the vague or simply incorrect messaging from the bid manager was later on perceived as a malicious intent from the vendor to mislead the client.

In the analysis by Verner and Abdullah [21], the 'significant risk representation' was ex-post associated with the outsourcing risk factors 'communication' and 'managing user expectations'. However, as demonstrated in the present reanalysis of the case, it may well be that such risk factors only refer to symptoms of underlying issues. With the modular structure as implemented, it became obvious that no adequate internal communication channels were in place to adequately inform the client. As a consequence, in terms of our research question as defined in Section 2, we believe that these risk factors could have been better understood and anticipated upon ex-ante by analyzing the case upfront through the modularity reasoning (and its resulting requirements) we presented here. The lack of a decent communication mechanism in the modular structure of the project team should have triggered the vendor to address the correctness of its external communication. From the client's side, questions could have been asked regarding the reliability and trustworthiness of the received information if they were aware of this project team configuration. Moreover, the present analysis indicates that one should be aware that some perceived 'risk factors' may actually represent a cover-up of an inadequately designed modular structure.

6. Discussion

The findings discussed in the previous section are summarized in Table 2. The columns of the table outline the level at which the modular structure was identified and the modularity aspects which were applied. Next, the resulting modularity requirements which could have been assessed in the project are listed, and the corresponding outsourcing risk factor is identified. This table shows how the modularity concept is related to outsourcing risk factors. It demonstrates how the identification of a modular structure (e.g., the application portfolio or the project team composition) and the selection of specific modularity aspects for this modular structure allows to formulate prescriptive modularity requirements. A violation of these requirements results in a lack of modularity characteristics, such as module substitution (i.e., applying modular operators) or reducing complexity. These characteristics are relevant in outsourcing projects, as certain organizational parts are substituted (i.e., replaced by a part offered by the outsourcing supplier), or project complexity needs to be dealt with. We showed in Section 5 how these four columns can be used to analyze how the modularity and outsourcing concepts are related. In our opinion, such in-depth relations need to be documented and aggregated to better understand this relation.

The four columns of Table 1 also indicate the approach which could be used in real-life projects to leverage the modularity concept to identify outsourcing risks ex-ante. The concrete way of working is discussed here in more detail. First, the modular structure of the problem domain needs to be made explicit. Most importantly, the identification of modules needs to be addressed. The application portfolio (which consists of applications as modules) and the project team composition (which consists of the roles of project members as modules) have been identified as modular structures. Both the application portfolio and the project team composition are frequently considered as modular structures [1,6,12]. Modular structures have already been described for different problem domains as well, such as marketing [17], legal [18], and telemedicine [15]. This indicates the wide applicability of this approach.

Second, the relevant modularity aspects need to be selected. Certain modularity aspects such as coupling, cohesion and interface have been mentioned in Section 4. A more detailed overview of the different aspects in modular systems can be found in [1]. Based on the kind of problem which needs to be studied, systems theory prescribes different modularity aspects (e.g., the module interface or the internal construction of the module) should be focused on. For example, by focusing on the integration of different modules, attention to the interfaces of the modules is required. For example, integration needed to be studied in the modular structure of the application portfolio. When the design of individual modules is being studied, attention to coherence is needed.

Third, the resulting modularity requirements are listed, which indicate ex-ante outsourcing risks. For the application portfolio, the absence of hidden dependencies between application interfaces and a specific development methodology for integrating different technologies (i.e., modularity requirements 1.1 and 1.2) have been identified. For the project team composition, the modularity requirement of high internal cohesion has been discussed (i.e., modularity requirement 2.1). In Table 2, a general description of these requirements is provided. In Section 5, a more in-depth discussion was provided concerning the specific indications of non-compliance regarding these requirements. A failure to comply with these requirements leads to the absence of modularity

Table 2: Summary of the results of the modularity analysis.

Modular Structure	Modularity Aspects	Modularity requirements (ex-ante)	Risk factors (ex-post) [10]
IT application portfolio	Interface	No hidden dependencies	Lack of required skills
	Separation of Concerns	Systematic method	
Project team composition	Cohesion	High intramodular cohesion	Communication problems Managing user expectations

characteristics. Therefore, we argue that in real-life projects, such analysis needs to be made to detect outsourcing risks ex-ante.

Fourth, the absence of modularity characteristics can be associated with specific outsourcing risks. The violations of modularity requirements resulted in a perceived lack of skills in this project. The design of the project team composition resulted in the observation of a lack of sufficient communication and management of user expectations.

The different steps of the approach discussed here provide a specific way to identify outsourcing risks ex-ante. As such, it can be positioned within the related research of outsourcing project failures. Several research projects have shown that significant early warning signs can be detected in outsourcing projects well before the project is actually regarded as a failure [11,16]. For instance, Philip et al. propose a four-stage process towards successful management of early warning signs: monitoring early warning signs, detecting early warnings signs, acknowledging issues, and addressing issues [16]. In the second stage, i.e., issue detection, outsourcing risk factors which are detected ex-post are employed (e.g., lack of trust and lack of required skills). In Section 5, the use of a modularity perspective (which results in clearly defined modularity requirements) was applied to identify such risks ex-ante, as violations against modularity requirements. Therefore, this lens contributes to a better ex-ante understanding, and hence detection, of outsourcing risks. In future research, the focus will be on how these outsourcing risks can be mitigated based on modularity insights as well. Consequently, future research will contribute more to the fourth step, i.e., addressing issues. This future research seems promising, based on the results of other researchers. For example, Zheng and Abbott argue that reconfiguration of organizational resources is vital to be successful in an outsourcing context [23]. As discussed in Section 4, the ability to reconfigure systems is an important characteristic of modular systems. Consequently, it is possible that the identification of designs to address issues based on modularity contributes to the ability to reconfigure, and hence, contributes to success in outsourcing.

7. Conclusion, contributions, limitations & future research

Many organizations tend to outsource IS functions in order to deal with the rapid pace of technological change. Unfortunately, outsourcing projects remain complex projects subject to a high failure rate. This paper showed in a detailed way that, by looking through the modularity lens, some risk factors could have been made transparent upfront. A reanalysis of the well-documented BSkyB case from a modularity perspective provided useful additional insights into risk factors related to both technical and non-technical issues. We focused on the technical complexity and project team composition to explain the risk factors 'lack of capabilities', 'communication', and 'managing user expectation'.

This paper has contributions for both academic researchers and practitioners. Regarding practitioners, our analysis provides a hands-on "tool" to assess the technical complexity and project team composition of outsourcing projects in the form of some ex-ante testable modularity requirements. In case violations regarding these requirements are identified, this might be seen as an ex-ante warning sign for the project manager in terms of a possible future project failure. Detecting such warning signs should urge project managers to take mediating actions on these domains accordingly, to prevent the corresponding risk factors from being realized. Moreover, the approach which was used to identify these testable modularity requirements can be used to identify additional modularity requirements. These additional requirements can indicate different early warning signs. Regarding the scientific knowledge base, it was described in a more detailed way than currently available, how outsourcing projects (and their failures) can be understood and interpreted from a modularity point of view. Additionally, it was shown that – to a certain extent – some of the traditionally identified risk factors in outsourcing projects (such as 'communication problems', 'managing user expectation' and 'user expectations management') might be considered as merely referring to symptoms of underlying issues. Such issues could be related to modularity aspects and provide additional insights into the causal mechanisms of project failures in an outsourcing context.

Currently, two main limitations are present in the study. The first limitation concerns the use of a single-case study research design. As the current findings are based on a single case, no valid generalizations can be made yet. In future research, additional cases might be carried out to further validate the extent to which the modularity perspective (and the identified modularity aspects) may help in getting a better ex-ante understanding of outsourcing project risks leading to project failure. As additional cases are analyzed, the research project could lead to a set of hypotheses, proposing modularity aspects and requirements as instruments to identify outsourcing project risk factors, of which the relation might ideally be tested quantitatively. The confirmation of (several of) these hypotheses might provide a sound basis to more confidently identify and understand outsourcing risk factors. The second limitation is the current lack of focus on mitigating actions to avoid the identified outsourcing risks. Therefore, as the ultimate goal of the presented future research, the identification of a set of mitigating measures should be aimed for, and tested in practice to validate whether they are indeed effective in preventing the concerning risk factors.

8. Acknowledgment

P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

9. References

[1] Baldwin, C.Y. and Clark, K.B. *Design Rules, Volume 1: The Power of Modularity*. The MIT Press, 2000.
[2] Benbasat, I., Goldstein, D.K., and Mead, M. The Case Research Strategy in Studies of Information Systems. *MIS Quarterly 11*, 3 (1987), 368–386.
[3] Cabigiosu, A., Zirpoli, F., and Camuffo, A. Modularity, interfaces definition and the integration of external sources of innovation in the automotive industry. *Research Policy 42*, 3 (2013), 662–675.
[4] Campagnolo, D. and Camuffo, A. The concept of modularity in management studies: A literature review. *International Journal of Management Reviews 12*, 3 (2010), 259–283.
[5] Camuffo, A. Rolling out a "world car": globalization, outsourcing and modularity in the auto industry. *Korean Journal of Political Economy 2*, (2004), 183–224.
[6] Dietz, J.L.G. *Enterprise Ontology: Theory and Methodology*. Springer, 2006.
[7] Fixson, S., Ro, Y., and Liker, J. Modularity and Outsourcing: Who drives whom? A Study of Generational Sequences in the U.S. Automotive Cockpit Industry. *International journal of Automotive Technology and Management 5*, 2 (2005), 166–183.
[8] Gay, C.L. *Inside outsourcing: the insider's guide to managing strategic sourcing*. N. Brealey Pub, London ; Naperville, IL, 2000.
[9] Hecker, A. and Kohleick, H. Explaining Outsourcing Failure. (2006).
[10] High Court of Justice, Queen's Bench Division, T. and Court, C. *BSkyB Limited, Sky Subscriber Services Limited, HP Enterprise Services UK Limited, Electronic Data Systems LLC*. 2010.
[11] Kappelman, L.A., McKeeman, R., and Zhang, L. Early Warning Signs of it Project Failure: The Dominant Dozen. *EDPACS 35*, 1 (2007), 1–10.
[12] Mannaert, H., Verelst, J., and Ven, K. Towards evolvable software architectures based on systems theoretic stability. *Software: Practice and Experience 42*, 1 (2012), 89–116.
[13] Miles, M.B. *Qualitative data analysis: a sourcebook of new methods*. Sage Publications, Beverly Hills, 1984.
[14] Miozzo, M. and Grimshaw, D. Modularity and innovation in knowledge-intensive business services: IT outsourcing in Germany and the UK. *Research Policy 34*, 9 (2005), 1419–1439.
[15] Peters, C. and Leimeister, J.M. TM^3 - A Modularization Method for Telemedical Services: Design and Evaluation. (2013).
[16] Philip, T., Wende, E., and Schwabe, G. Why Project Managers Fail to Act upon Early Warning Signs: Evidence from Failed Offshore-outsourced Software Projects. *Proceedings of the 21st European Conference on Information Systems*, (2013).
[17] Sanchez, R. Modular Architectures in the Marketing Process. *The Journal of Marketing 63*, (1999), 92–111.
[18] Smith, H.E. Modularity in Contracts: Boilerplate and Information Flow. 2006.
[19] Tiwana, A. and Bush, A. A Comparison of Transaction Cost, Agency, and Knowledge-Based Predictors of IT Outsourcing Decisions: A U.S.-Japan Cross-Cultural Field Study. *Journal of Management Information Systems 24*, 1 (2007), 259–300.
[20] Tiwana, A. Does interfirm modularity complement ignorance? A field study of software outsourcing alliances. *Strategic Management Journal 29*, 11 (2008), 1241–1252.
[21] Verner, J.M. and Abdullah, L.M. Exploratory case study research: Outsourced project failure. *Information & Software Technology 54*, 8 (2012), 866–886.
[22] Yin, R.K. *Case Study Research: Design and Methods*. Sage Publications, Newbury Park, California, 2003.
[23] Zheng, Y. and Abbott, P. Moving up the value chain or reconfiguring the value network? An organizational learning perspective on born global outsourcing vendors. *Proceedings of the 21st European Conference on Information Systems*, (2013).
[24] Zirpoli, P. and Becker, M.C. What Happens When You Outsource Too Much? *MIT Sloan Management review 52*, 2, 59–64.