Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts

Jon Iden, Bjørnar Tessem, Tero Päiväranta

Article info

Article history:
Received 23 March 2010
Received in revised form 1 December 2010
Accepted 1 December 2010
Available online 14 December 2010

Keywords:
System development
IT operations
Interplay
Problems
The Delphi method

Abstract

Context: The assumption of the presented work is that the ability of system developers and IT operations personnel to cooperate effectively in system development projects has great impact on the quality of the final system solution, as well as on the service level of its subsequent operation.

Objective: The present research explores the interplay of system development and IT operations and the challenges they are meeting. We are especially interested in identifying problems encountered between these two parties in system development projects.

Method: We identify and rank problems by using a ranking-type Delphi study. We involved 42 Norwegian IT experts and split them into three expert panels: system developers, IT operations personnel and system owners. We then guided them through the three phases of the Delphi method – brainstorming, reduction and ranking.

Results: A comprehensive list of 66 problems, organized into seven groups, is compiled. Through a selection and ranking procedure, the panels found the following to be the six most serious problems in the interplay of system development and IT operations: (1) IT operations not being involved in the requirements specification; (2) poor communication and information flow; (3) unsatisfactory test environments; (4) lack of knowledge transfer; (5) systems being put into production before they are complete; and (6) operational routines not being established prior to deployment.

Conclusion: The sheer amount and variety of problems mentioned and the respondents’ explanations confirm that this interplay needs attention; the parties agree that they do not cooperate effectively in development projects. The results imply that IT operations should be regarded as an important stakeholder throughout several systems development activities, especially requirements analysis, testing and deployment. Moreover, such involvement should be facilitated by an increased focus on enhancing cooperation and communication.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The development as well as the subsequent operation of information systems (IS) may involve a number of failures. Many have studied the reasons for this unfortunate situation, and lack of user involvement, incomplete requirements specifications and lack of resources are among the explanations frequently mentioned [1,2]. With a more general perspective, Lyytinen and Robey have argued that the incapability of the system development organization to learn from previous failures is a root cause of system development failures [3].

A few approaches to improve the likelihood of success of IS implementations have been suggested, such as software process improvement initiatives [4]: information, system and service quality [5]; IT governance [6]; IT-business alignment [7]; and the ability to take the social context [8] and organizational learning [3] into consideration. Furthermore, Peppard and Ward [9] argue that competence in delivery (including system development and IT operations) of solutions is essential.

Studies have found that well-managed relationships between the stakeholders are among the enabling factors in system development success. For example, several studies address that the strength of the IT–business relationship has a direct effect on the outcome of a new information system [7,10,11]. On a different scale, as part of system development practice, the relationship between the system developer and the user is found to be important. This recognition has spawned important topics like customer participation, human–computer interaction and usability [12–15]. Although academics recognize the importance of business and user
involvement in various IS development practices, empirical evidence on the nature of the interplay of system development and IT operations in system development projects remains scarce.

There are several reasons why this area needs attention. First, system development and IT operations often reside, due to specialization, in different organizational units. This arrangement may be complicated by the fact that many companies have outsourced their IT operations, leading to a number of risk factors to be managed in their own right [16]. Outsourcing has also become common in system development projects [17], providing, again, challenges related to stakeholders’ interplay [18].

Second, although system development, in character, is an activity performed under control of the system development organization, the responsibility for running the new system in production is transferred to IT operations at the end of the project. Some of the activities related to this transfer are planning, documenting, infrastructure configuring, testing, training and deployment, as well as the overall management of these activities [19]. Performing these activities involves integrating and coordinating knowledge from many individuals from different fields and backgrounds. It demands a close partnership between system development and IT operations staff, both in formal project activities and processes, as well as in the more informal and daily working relationships. This applies regardless of the form of organizing the employment of the system developers and IT operators – whether outsourced or not.

Third, an information system is not an isolated entity. A system will operate in an often complex integrated operational environment, and it must comply with the rules, standards, architectures, infrastructures and operational routines defined by IT operations. Further, hardware, networks, operating systems and databases are often shared by many systems [9].

Fourth, deploying a new information system brings about changes in the operational environment, e.g. infrastructure, configuration, software, hardware and network capabilities, security, procedures, documentation, service catalogues, service level agreements, pricing models, personnel, competence and vendors. It could be argued that IT operations must be involved early in the project in order to make the necessary changes prior to deployment. Non-functional requirements is a commonly used term for many of these issues [20]. Successful system development and subsequent operation rest on the ability of the system developers to inform IT operations about the characteristics and requirements of the new system, and on IT operations’ ability to implement the necessary operational changes. Put another way, new systems should be designed and built for operability, reliability, performance and manageability and tested for compliance [21].

In this paper, we intend to explore the interplay of system development and IT operations and the challenges met in this interplay. We are especially interested in identifying problems encountered between these two parties in system development projects. We focus on systems which are built specifically to fulfill the target organization’s strategic or otherwise crucial business requirements. Thus, we leave out projects for the acquisition, implementation and operation of “standard” software packages, such as commercial ERP, office, or other “off-the-shelf” or even easily configurable products. We define a problem as a factor that may reduce the qualities of the final system solution, as well as the service level of its subsequent operation. For exploring the interplay of system development and IT operations, this study poses three research questions:

RQ1. What problems are experienced in the interplay of system development and IT operations in system development projects?

RQ2. Which problems do the stakeholders consider most severe?

RQ3. What may be the causes of these problems?

We approach these questions empirically, after recognizing a lack of investigation in previous literature, and a ranking-type Delphi method is chosen as the research method. Forty-two experts, grouped into three panels – system developers, IT operations and system owners – participate in this study in order to ensure a broad view of the problems and their prioritization. Based on the suggestions of the experts, a comprehensive list of 66 problems, organized into seven groups, is compiled. Through a selection and ranking procedure, three independent ranked lists, with one from each panel, are derived, thus pointing out the most severe problems.

The paper proceeds as follows. First, it presents the conceptual basis for this study. Next, it discusses our research methodology, followed by a presentation of the complete list of identified problems and the results from the selection and ranking procedure. The paper concludes by discussing the six highest-ranked problems and their causes, provides possible explanations and limitations and suggests implications for practice and future research.

2. Conceptual basis

System development is about building new information systems that support the needs of the business, and is commonly organized in projects with fixed goals, budgets and time limits, normally under the responsibility of a system development organization [20]. Several methodological approaches to system development exist, such as the Waterfall Method and its more iterative elaborations [22], the Spiral Model [23], the Unified Software Development Process [24] and the more recent approaches of so-called agile software development, such as Extreme Programming [25] and Scrum [12]. IT operations, on the other hand, are about running and controlling the various information systems in production, maintaining the technological IT infrastructure and the handling of system incidents and user requests [26]. IT operations normally reside in the computer centre, including such functions as service desk, operation systems and storage management, data base and transaction management, networks and telecommunication and the centre for the control and monitoring of online and batch production and networks. Taking an IT service management perspective, the tasks of IT operations can be divided into areas such as service planning, service design, service transition and service operation, each with their specific processes and functions [26].

Development of information systems may involve a number of failures; the project may exceed the budget, the system may be delivered behind schedule and it may fail to meet the users’ requirements [1]. The factors that cause projects to fail or their results to be challenged may include: lack of effective project management skills, lack of adequate user involvement, incomplete or changing requirements specifications, lack of executive support, lack of training and immature technology [2,27]. Lyytinen and Robey [3] state that a general root cause of system development failures is the incapability of the system development organization to learn from previous failures due to limited organizational intelligence, disincentives for learning, organization design and educational barriers.

Although similar studies are harder to find for IT operations, this area is not without its failures either. A Gartner Group report, for example, says that nearly 80% of production outages ( unplanned downtime) occur as a result of what they call “operator error” and “application failures,” while the remaining 20% are a result of technology (referenced in [21]).
Within system development and IT operations, initiatives are recurrently taken to improve task effectiveness and outcome quality. New system development methods are explored and introduced [12,23–25], and the software process itself is consciously being improved [4]. Under the name of IT Service Management, IT operations worldwide are on a course to change their practices from the traditional technical orientation to a more customer- and service-oriented style [28]. However, studies show that improvement initiatives in system development and IT operations are, in general, limited to challenges within each respective domain and that they seldom address topics that concern both domains and their respective interplay [19].

The recent concept of strategic IS capability [9] regards system development and IT operations as one collective competence: “deliver solutions”. According to Peppard and Ward [9], the concept of IS capability consists of six major IS competences, and they argue for organizations “to understand, develop and nurture this capacity if they are to deliver value form investments made in IT on an on-going basis” (p. 170). All organizations have an IS capability, weak or strong, that is embedded in organizational processes, routines and structure. Competencies are the abilities of the organization to develop, mobilize and use its resources; capability is what a firm can achieve through focused investments and deployment of competencies (p. 182). Peppard and Ward argue that fusing system knowledge and business knowledge is paramount to ensure the conception of strategies involving technological innovation, to make appropriate choices for the opportunities available and to implement these strategies quickly (p. 183). They also name, as one of the other macro competencies, “deliver solutions”, which is made up of (1) applications development, which concerns the development of solutions that satisfy business needs, and (2) service management, which concerns operational and performance criteria matching business requirements (p. 178). Analogically to the fusion of system and business knowledge, one could envision that, within the IT function, fusing development knowledge and operations knowledge is vital to ensure that new systems fulfill business needs. Although the framework by Peppard and Ward identifies operations as a part of “deliver solutions”, the relationship between operations and development remains to be discussed.

The lack of focus on the relationship between system development and IT operations in the literature is notable, as these two functions, in practice, have to cooperate effectively on a number of topics and activities. Cooperation is essential given the increasing complexity of modern information systems and their underlying technical configurations. A new system may, for example, introduce new operational technology, affect the capacity of existing computing and network resources and require specific security solutions. In addition, when developing a system, developers must take into consideration the standards, architectures and operational routines set by IT operations.

Although problems of interplay of system development and IT operations may exist with all kinds of software systems, our research examines the development and the subsequent operation of information systems for a particular organization. We are not considering the implementation of standardized software packages. Given that this interplay must be managed, we find that within the research literature, neither the methodological, nor project-related nor technical factors of the interplay of system development and IT operations are investigated in as detailed a manner as one would expect. It seems as though this area remains poorly formulated in the literatures of both system development and IT operations. Our approach is therefore to address these issues empirically in order to contribute to the aforementioned research gap.

3. Research method

The overall aim of this study is to develop an overview and understanding of what problems occur in the interplay of developers and operations personnel in development projects. Problems in this context denote factors that may have negative effects on the final qualities of the system, as well as for the subsequent daily operation of it. In order to identify, select and rank important problems, a ranking-type Delphi method was chosen [29]. The Delphi method is a widely employed and accepted group decision methodology. Delphi is being used in many complex areas in which pure model-based statistical methods are not practical or possible, where human judgmental input is necessary and when experts are geographically dispersed [30,31], but not necessarily to elicit a single answer to a problem [30,32].

The Delphi method is a highly formalized method of communication between researchers and a panel of experts, and is designed to extract the maximum amount of unbiased information from the experts. Four key features characterize the Delphi method and distinguish it from other group interaction methods: anonymity, multiple iterations, controlled feedbacks and the statistical aggregation of the group response [30]. The main disadvantages of a Delphi study are seen as the length of the process, researchers’ influence on the responses due to particular question formulations and difficulty in assessing and fully utilizing the expertise of the experts because they never meet face-to-face [33]. Readers interested in detailed descriptions and discussions of the Delphi method are referred to other studies [30,31,34–36].

3.1. Composition of the panels

The success of a study based on the Delphi method depends on highly qualified experts with thorough understanding and knowledge of the topic in question. A Delphi study does not, however, depend on a statistical sample that attempts to be representative for a certain population [34].

What kind of experts could inform our research? We found three groups of experts to be especially relevant: system developers, IT operations personnel and system owners. We chose developers and operators because they have direct experience with the subject in question. System owners were chosen because they, in addition to being involved throughout the development project from initiation to production, also have a position that allows them, as customers, to view the practices and the interplay from a reflective distance. We anticipated, however, that these three groups of experts have different views on the subject, and to be able to poll any differences in opinion, three panels – one for each group – were established. Comparison of perspectives between three organizational groups has been conducted by others [37,38]. The selection of experts was inspired by the procedure presented by [34], and the main selection criteria were as follows: (1) a long-established career in IT and (2) currently holding a position that clearly relates them to one of the three groups.

We contacted experts we know by e-mail and by phone, and invited them to participate by explaining the research area of interest, the purpose of the research and the research method to be used. We also asked them to nominate others that satisfied the criteria. Finally, from a list of 52 candidates, 42 experts agreed to contribute and ten declined; comprising 20 system developers, 12 persons from IT operations and 10 system owners. Thirty-four experts came from the private sector, representing business areas such as banking and insurance, engineering and construction, information technology, oil and gas and retail. Eight experts came from the public sector, representing large state offices of public administration and public education institutions.
From some companies we managed to recruit experts for more than one role, which is why we have fewer companies than panel members. In total, the experts represented 18 different companies from various sectors and of different sizes (cf. Table 1).

A few panel members informed us that they had worked as both system developers and IT operators during their career. It was also typical for the system owners to have previous experience in either one or both of these roles. However, we located the experts in the panels according to their current position. The employment relation of the experts varied. In the system development panel, ten panel members represented in-house IT departments, whereas the other ten represented IT vendors and consultancies offering system development expertise to the market. Among the IT operations panel, five experts represented in-house IT departments, whereas the other seven represented companies that provide IT operations as a service. With regard to the system owner panel, nine experts represented an in-house view, whereas the tenth was employed by a service provider external from the company. The companies from where we had in-house experts had more than 250 employees, whereas the IT vendors and consultancies varied greatly with regard to their size. For example, six of the twenty experts from the system development panel were employed by small consulting firms, whereas others represented multi-national consultancies, leading big system development projects and providing IT operations services. All in all, we believe that our ensemble of the panel members (Table 1) represents a satisfactory variation to bring up issues related to the problem area from many viewpoints, and avoids a bias of particular modes of organizing the development projects and IT operations. Non-response from panel members is a major problem in questionnaires. In this study, the non-response was exceptionally low. Of those who did not respond, only one expert, a system owner, left during the later phases of the study.

3.2. Data collection and analysis method

Data collection and analysis were based on the method provided by [29,27], and were divided into three distinct phases: brainstorming, reduction and ranking.

Table 1

<table>
<thead>
<tr>
<th>Company size (number of employees)</th>
<th>Number of companies represented</th>
<th>Number of panel members</th>
<th>Developers, IT operators, system owners (in-house/outsourced)</th>
<th>Areas of business</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;250</td>
<td>5</td>
<td>7</td>
<td>6 (0/6), 1 (0/1), 0 (0/0), 2 (2/0), 3 (3/0), 2 (2/0)</td>
<td>Information technology, Public administration, Education, Banking &amp; insurance, Retail Education, Information technology, Public administration, Banking &amp; insurance, Engineering &amp; construction, Information technology, Oil &amp; gas</td>
</tr>
<tr>
<td>250–1000</td>
<td>5</td>
<td>7</td>
<td>6 (0/6), 1 (0/1), 0 (0/0), 2 (2/0), 3 (3/0), 2 (2/0)</td>
<td>Information technology, Public administration, Education, Banking &amp; insurance, Retail Education, Information technology, Public administration, Banking &amp; insurance, Engineering &amp; construction, Information technology, Oil &amp; gas</td>
</tr>
<tr>
<td>1000–10,000</td>
<td>4</td>
<td>14</td>
<td>4 (1/3), 7 (1/6), 3 (2/1)</td>
<td>Information technology, Public administration, Education, Banking &amp; insurance, Retail Education, Information technology, Public administration, Banking &amp; insurance, Engineering &amp; construction, Information technology, Oil &amp; gas</td>
</tr>
<tr>
<td>&gt;10,000</td>
<td>4</td>
<td>14</td>
<td>8 (7/1), 1 (1/0), 5 (5/0)</td>
<td>Information technology, Public administration, Banking &amp; insurance, Retail Education, Information technology, Oil &amp; gas</td>
</tr>
<tr>
<td>Total:</td>
<td>18</td>
<td>42</td>
<td>20 (10/10), 12 (5/7), 10 (9/1)</td>
<td></td>
</tr>
</tbody>
</table>

3.2.1. Phase 1: Brainstorming

A welcome letter was first sent to the participants by e-mail, in which they were asked to identify, at a minimum, six problems that they have encountered or are currently aware of in the interplay of system development and IT operations in their respective environments. They were also asked to explain and comment on each problem. A plain document template was attached for the experts to fill in. The purpose was to allow the individual experts the freedom to identify and explain the important issues from their points of view.

Two researchers worked independently to bring together and collate the responses from the experts. Then, the two independently constructed lists were compared and reconciled by the researchers. Duplicates were removed, reducing the total number of items proposed to a final compiled list of 66 problems (Table 2).

3.2.2. Phase 2: Reduction

In the second phase, the experts were divided into three panels – system developers, IT operations and system owners – allowing each panel to independently pare down the number of problems. The purpose of this process was to narrow the list to a manageable number of items, so that the problems, in the next phase, could be meaningfully ranked. Each panellist received an e-mail with an attached randomized list of the 66 problems and was requested to select at least 10 problems that he or she considered to be most important.

A goal of this phase was to identify not more than 15–20 problems to be ranked in the next phase. A target of about 20 items has been suggested [29]. A good strategy in other Delphi studies [27,34] has been to pick the factors that were identified among at least 50% of the participants. We soon recognized that we would have to use a different strategy for reducing the problem list.

First of all, we had to consider that we had three different panels, and there was not a high correlation between the rankings of these three panels. The correlation computation was done as follows: for each of the three groups we made a ranked list of problems based on the number of participants selecting that particular list. We then computed Kendall’s \( r \) rank correlation of each pair of these three rankings. All the three ranking correlations turned out to be less than 0.4, which showed us that the different groups were quite different in their rankings.

We continued by using a 30% threshold in each panel, achieving 15 problems in the developers group, 17 in the operations group and 20 in the system owner group. But only four problems were common for all groups, and with the three lists combined, we achieved a total of 34 problems. Since we wanted to stick to one list of problems for the next phase, we chose, in order to further reduce the list, to select 4–6 problems from each group (six from development, six from operations and four from the system owners; the number varied due to ties in the rankings).

There were some overlaps, so we now possessed a list of 13 problems, and all panels had their four most important problems in the combined list. Next, we created a weighted ranking where we weighted the number for each item that was selected in each panel according to panel size, and picked the four not yet selected problems from the 15 highest-ranked in this weighted list. In total, we ended up with 17 problems that we wanted the panellists to rank (see Tables 2–4). With this approach, we achieved problems that were scored high overall, but also ensured that we included problems important within each panel.

3.2.3. Phase 3: Ranking

The ranking of the selected problems was done in the third phase. Here, panellists were asked to rank the 17 problems in order of priority; that is, the top-ranked problems should be those that

\[ Kendall's \ r = \text{correlation coefficient} \]

\[ \text{Threshold} = 30\% \]
Table 2
Full list of problems grouped by theme.

1. Stakeholder perspectives and priorities
   - System development and operations have different needs and priorities
   - System development does not sense responsibility for the system after it has been put into production
   - System development does not prioritize tasks related to system operation
   - System development projects can focus solely on the individual system, while operations must take care of the entire production environment
   - There are diverse views on what the objective for the new system implementation is
   - Personnel in operations do not possess the same level of ownership of a system as developers do

2. Organizing
   a. General-level organizing issues
      - System development and operations are functionally separate working environments prioritizing their own tasks
      - Operations are not formally involved in the system development project
      - Operations are involved in the system development project too late
      - Roles and task responsibilities in both development and operations are vague
      - There is no clear distinction between operations and development
      - Personnel turnover is high for both system development and operations
      - IT personnel who have obligations in both system development and operations find it difficult to commit time to system development work
      - System development and operations are geographically distributed
      - System development's access rights (authorization) to the production environment are too restricted
   b. Outsourcing
      - When system development is outsourced, operations become less involved in development projects
      - Outsourcing of operations makes communication and information exchange more difficult
      - Outsourcing of operations leads to more bureaucracy for system development
   c. In an outsourcing process, vendors are competing on price, and they tend to overlook operational functionality in their offers

3. Standardization and improvement
   a. Standardization of methods, tools and technologies
      - Manual routines and cumbersome tools are used for application installations in test and production environments
      - System development and operations have divergent interests concerning the use of modern development tools
      - System development projects make use of standardized methods and processes only to a small extent
      - Methods for determining the value of a system that is efficient to operate are not available. This makes it hard to prioritize operations in the development project
      - Availability and stability standards defined by operations are hampering system development work
      - Operations choose to standardize a few technologies and platforms. This minimizes potential solutions and may lead to non-optimal systems
   b. Standardization and improvement of work processes
      - Operational routines are too bureaucratic and do not match the needs of system development
      - System development is not following the change management processes set by operations when new systems are put into production
      - Initiatives for improving the processes in the interplay of development and operations are not being prioritized

4. Cooperation and communication
   a. System development and operations are protecting their own areas and interests
   b. System development finds it hard to engage people from operations for system development projects
      - Communication and the flow of information between system development and operations are poor
      - Operations do not inform sufficiently about operational changes, which may have consequences for the development project
      - Operations receive insufficient information about how the new system works because of missing knowledge transfer

5. Development project activities
   a. Planning and scheduling
      - System development projects do not make plans for deployment and operation
      - The needs for scheduling are divergent. Operations need predictability and a long-range planning perspective, while system development seeks flexibility
      - Training is not planned for enough
   b. Requirements specification
      - System development has too little focus on how the system should function in the production environment
      - IT operations are not involved in the requirements specification, consequently technological and operational aspects are thus not sufficiently taken care of
      - System development accepts new requirements from the sponsor after the system is transferred to the test phase
   c. Testing
      - Testing and approval of new systems are not prioritized
      - Systems are being tested on a small scale only, and not in relation to the whole configuration
      - Operations are not involved when tests are planned for and performed
      - Too little emphasis is placed on establishing satisfactory test environments for volume, performance and capacity testing
      - System development projects produce systems with quality that is too low
   d. Deployment
      - The process for handing over a new system from system development to operations is not well structured
      - Responsibilities in accordance with handovers and production settings are not well defined
      - New systems are put into production too early, before they are complete
      - Operational routines for the new system are not established prior to deployment
      - Too many new systems are put into production within a short timeframe. The consequence is reduced time for testing and further poorer system quality
      - The documentation that operations receive with a new system is not adequate

6. Competence
   a. Developer competence
      - System development has too little understanding of security requirements, which results in insecure and vulnerable solutions
      - System development has insufficient ability to address the effects a new system may have on operational performance
      - System developers are not well informed about how the entire production environment works; how the different systems work together
      - System development does not know how operations work or what routines apply to deployment
   b. Operator competence
      - Operations do not have sufficient understanding of the business
      - Operations do not get the chance to acquire adequate competence in new technologies introduced with a new system
      - Operations do not have insight into the development deliverables. Operations do not know what must be tested, what is to be implemented, what dependencies there are and what consequences the new system brings for the entire production
would most likely reduce the quality of the system, as well as the subsequent operation of it.

Two rounds of ranking were conducted. In a Delphi study the number of ranking rounds should be determined by each panel either reaching an acceptable level of consensus or reaching a state where we are not able to increase the consensus level any further. To measure the degree of consensus among the panellists, Kendall's coefficient of concordance (W) was used. Kendall's W is a statistic used on a collection of rankings on the possible values of a particular variable. In this case the variable is “Problem in the interplay of development and IT operations”, and the ranking is a respondent's ordering of the 17 problems selected in phase 2. Using Kendall’s W, one can measure the relative strength of the consensus within the group of respondents, make a realistic determination of whether any consensus has been reached and determine whether the consensus is increasing as the panelists receive information about how others rank the problems. A normal procedure is to ask the respondents for a ranking, making an average ranking, computing Kendall's W and deciding if it is necessary to run one more round of ranking with the respondents. If a new round is run, they are asked to take into consideration the last average ranking when giving their new ranking. In many studies, a concordance level of W = 0.7 is considered a goal, as this indicates a quite high level of agreement among respondents on how to rank the values.

One of the experts in the system owner panel left the study before this phase, reducing the total to 41 participants. From the first ranking round, we could see that there is a very small, but significant, degree of agreement among all 41 participants (Kendall’s W = 0.084, sign = 0.000). The significance-value (sign) here denotes the probability that there is no concordance given the rankings. The W-value was highest among operations personnel (N = 12, W = 0.183, sign = 0.004) and among system owners (N = 9, W = 0.216, sign = 0.013). Among the developers, however, we did not get significant concordance (N = 20, W = 0.05, sign = 0.447).

In the second ranking round, the panelists were once again asked to rank the problems, but now with knowledge of the average rankings from the first round. There was some correlation between the average rankings from the three groups, but it was not high (Kendall's τ ranging from 0.382 to 0.442), so we chose to present the participant's group average ranking as the input to the participants instead of the overall ranking. After this round of ranking, we saw increased and significant agreement in all three groups (developers: N = 20, W = 0.206, sign = 0.000; operations: N = 12, W = 0.316, sign = 0.000; system owners: N = 9, W = 0.371, sign = 0.000).

At this stage, we had to decide whether we should go on. We had not yet reached the W = 0.7 level that is often applied in Delphi studies, and we had a fair increment in the W-values from the first round. However, from the first to the second round, 30% of the participants chose not to change their ranking. In addition, many were slow in their responses, and we had to run several reminder rounds. The distance to W = 0.7 is also large, so we considered the chances of reaching that level of consensus to be very improbable. Thus, we decided to stop the ranking rounds at this point.

4. Results

4.1. The problems

The complete list of the 66 important problems identified in phase 1 is presented in Table 2. The first objective of this research is to identify the problems that are encountered in the interplay of system development and IT operations, and as this area is little researched; such a list may be of value in itself. Thus, we list all the problems here.

As may be expected, the list contains a blended mix of problems. The problems were grouped using a bottom-up approach, where the individual problems were analyzed and then grouped by the researchers according to theme. No pre-authorized grouping or categorization scheme for these problems has been found in the literature. Seven main groups and eleven subgroups were formed: stakeholder perspectives and priorities, organizing (in general and related to outsourcing), standardization and improvement (methods, tools, technologies and work processes), cooperation and communication, development project activities (planning and scheduling, requirements specification, testing and deployment), competence (developers, operators and in general) and economy and contracts.

4.2. Ranking of problems

This chapter presents the results of the rankings by the panel lists. Two principles guided the ranking process. First, as explained in the methodology chapter, we did not want our panellists to rank more than 20 problems, and second, we wanted the three panels to rank the same problems. Through the procedure accounted for in chapter 3, we obtained one list of 17 problems to rank. Tables 3–5 present the selected problems and the rankings, round by round, by each panel.

In Table 6, the rankings of the three panels have been combined, based on the rankings each problem has achieved from the panels. Most likely, we would get a somewhat different ranking if we had run the Delphi method without separating the panels. However, the presented ranking gives an indication as to which problems are the most important overall, based on the assumption that all three panels should be given equal weight on the issue.

5. Discussion and implications

The aim of this study is to develop an overview and understanding of what problems occur in the interplay of system development and IT operations in system development projects. The Delphi method is used in order to form the most reliable consensus of
opinion of three expert groups: system developers, IT operators and system owners. Two pertinent questions in relation to this consensus-forming process are as follows: what is the level of agreement within each panel (intra-panel agreement) and what is the agreement across the three panels (inter-panel agreement)? To understand to what extent the expert within a panel agreed with the relative importance of various problems, we compared the experts’ ranking within each panel using Kendall’s coefficient of concordance (W). As can be seen from Tables 3–5, the final levels of agreement for all panels are weak according to the scale given by [29], with W = 0.206 for the system developers, W = 0.316 for the IT operations personnel and W = 0.371 for the system owners. The difference in the value of W indicates a somewhat stronger agreement among the IT operators and among the system owners than among the system developers. Several factors may explain these variations. One evident reason resides in the variety of development projects, system types and the operational arrangements from where the experts have their experience. Further, the profile of the organizations in which the experts work may also vary; different approaches for organizing the IT function exist [39]. The lack of high concordance within each panel indicates that a strong within-profession consensus on the most important problems remains to be formed.

The lack of consensus is also clear between the domains. As anticipated, the three panels perceive the problems in the interplay differently. We saw from the results after the reduction phase, as

### Table 3

<table>
<thead>
<tr>
<th>Problems</th>
<th>Mean ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Round 1</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Kendall’s W</td>
<td>0.050</td>
</tr>
</tbody>
</table>

### Table 4

<table>
<thead>
<tr>
<th>Problems</th>
<th>Mean ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Round 1</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Kendall’s W</td>
<td>0.183</td>
</tr>
</tbody>
</table>
nevertheless interesting. For example, the problem reciprocal Kendall’s W-values lie around 0.4. This is anticipated, but sufficiently taken care of.

Table 5
Ranking results, round by round: system owners.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Problem</th>
<th>Mean ranks</th>
<th>Mean ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Communication and the flow of information between system development and operations are poor</td>
<td>5.00</td>
<td>3.78</td>
</tr>
<tr>
<td></td>
<td>System developers are not well informed about how the entire production environment works; how the different systems work together</td>
<td>6.56</td>
<td>4.78</td>
</tr>
<tr>
<td></td>
<td>IT operations are not involved in the requirements specification, consequently technological and operational aspects are thus not sufficiently taken care of</td>
<td>6.44</td>
<td>5.00</td>
</tr>
<tr>
<td>4</td>
<td>The process for handing over a new system from system development to operations is not well structured</td>
<td>7.33</td>
<td>6.44</td>
</tr>
<tr>
<td>5</td>
<td>New systems are put into production too early, before they are complete</td>
<td>8.00</td>
<td>7.11</td>
</tr>
<tr>
<td>6</td>
<td>Too little emphasis is placed on establishing satisfactory test environments for volume, performance and capacity testing</td>
<td>7.33</td>
<td>7.44</td>
</tr>
<tr>
<td>7</td>
<td>Operations receive insufficient information about how the new system works because of missing knowledge transfer</td>
<td>7.56</td>
<td>7.89</td>
</tr>
<tr>
<td>8</td>
<td>Operational routines for the new system are not established prior to deployment</td>
<td>8.33</td>
<td>9.00</td>
</tr>
<tr>
<td>9</td>
<td>Initiatives for improving the processes in the interplay of development and operations are not being prioritized</td>
<td>9.56</td>
<td>9.33</td>
</tr>
<tr>
<td>10</td>
<td>IT personnel who have obligations in both system development and operations find it difficult to commit time to system development work</td>
<td>9.44</td>
<td>9.56</td>
</tr>
<tr>
<td>11</td>
<td>Personnel in operations do not possess the same level of ownership of a system as developers do</td>
<td>9.11</td>
<td>9.67</td>
</tr>
<tr>
<td>12</td>
<td>Operations are involved in the system development project too late</td>
<td>9.56</td>
<td>10.56</td>
</tr>
<tr>
<td>13</td>
<td>Operations are not formally involved in the system development project</td>
<td>10.67</td>
<td>10.56</td>
</tr>
<tr>
<td>14</td>
<td>The documentation that operations receive with a new system is not adequate</td>
<td>10.22</td>
<td>11.33</td>
</tr>
<tr>
<td>15</td>
<td>System development does not know how operations work, or what routines apply to deployment</td>
<td>10.78</td>
<td>12.22</td>
</tr>
<tr>
<td>16</td>
<td>System development and operations are functionally separate working environments prioritizing their own tasks</td>
<td>13.00</td>
<td>13.22</td>
</tr>
<tr>
<td>17</td>
<td>The needs for scheduling are divergent. Operations need predictability and a long-range planning perspective, while system development seeks flexibility</td>
<td>14.11</td>
<td>15.11</td>
</tr>
</tbody>
</table>

Kendall’s W

0.216 0.371

Table 6
Ranking results from combining panels’ ranking.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Problem</th>
<th>System developers’ rank</th>
<th>IT operators’ rank</th>
<th>System owners’ rank</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IT operations are not involved in the requirements specification, consequently technological and operational aspects are thus not sufficiently taken care of</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>Development project activity</td>
</tr>
<tr>
<td>2</td>
<td>Communication and the flow of information between system development and operations are poor</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>Cooperation and communication</td>
</tr>
<tr>
<td>3</td>
<td>Too little emphasis is placed on establishing satisfactory test environments for volume, performance and capacity testing</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>Development project activity</td>
</tr>
<tr>
<td>4</td>
<td>Operations receive insufficient information about how the new system works because of missing knowledge transfer</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>Cooperation and communication</td>
</tr>
<tr>
<td>5</td>
<td>New systems are put into production too early, before they are complete</td>
<td>11</td>
<td>2</td>
<td>5</td>
<td>Development project activity</td>
</tr>
<tr>
<td>6</td>
<td>Operational routines for the new system are not established prior to deployment</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>Development project activity</td>
</tr>
<tr>
<td>7</td>
<td>System developers are not well informed about how the entire production environment works; how the different systems work together</td>
<td>10</td>
<td>11</td>
<td>2</td>
<td>Competence</td>
</tr>
<tr>
<td>8</td>
<td>The process for handing over a new system from system development to operations is not well structured</td>
<td>8</td>
<td>12</td>
<td>4</td>
<td>Development project activity</td>
</tr>
<tr>
<td>9</td>
<td>Operations are involved in the system development project too late</td>
<td>9</td>
<td>4</td>
<td>12</td>
<td>Organizing</td>
</tr>
<tr>
<td>10</td>
<td>Initiatives for improving the processes in the interplay of development and operations are not being prioritized</td>
<td>5</td>
<td>14</td>
<td>9</td>
<td>Standardization and improvement</td>
</tr>
<tr>
<td>11</td>
<td>The documentation that operations receive with a new system is not adequate</td>
<td>12</td>
<td>7</td>
<td>14</td>
<td>Development project activity</td>
</tr>
<tr>
<td>12</td>
<td>IT personnel who have obligations in both system development and operations find it difficult to commit time to system development work</td>
<td>15</td>
<td>9</td>
<td>10</td>
<td>Organizing</td>
</tr>
<tr>
<td>13</td>
<td>System development and operations are functionally separate working environments prioritizing their own tasks</td>
<td>6</td>
<td>15</td>
<td>16</td>
<td>Organizing</td>
</tr>
<tr>
<td>14</td>
<td>Operations are not formally involved in the system development project</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>Competence</td>
</tr>
<tr>
<td>15</td>
<td>System development does not know how operations work, or what routines apply to deployment</td>
<td>14</td>
<td>10</td>
<td>15</td>
<td>Organizing</td>
</tr>
<tr>
<td>16</td>
<td>Personnel in operations do not possess the same level of ownership of a system as developers do</td>
<td>16</td>
<td>17</td>
<td>11</td>
<td>Stakeholder perspectives and priorities</td>
</tr>
<tr>
<td>17</td>
<td>The needs for scheduling are divergent. Operations need predictability and a long-range planning perspective, while system development seeks flexibility</td>
<td>17</td>
<td>16</td>
<td>17</td>
<td>Development project activity</td>
</tr>
</tbody>
</table>
operations must handle the errors that an incomplete system encounters in deployment and production, and the system owners on their side experience the commercial consequences of a system not capable of fulfilling business needs. The system developers may be farther away from these consequences. Moreover, the problem “Operations receive insufficient information about how the new system works because of missing knowledge transfer,” is ranked high by both system developers (3) and IT operations (3), while system owners ranked it lower (7). Lack of knowledge about how a new system works will undoubtedly be experienced by the personnel in IT operations, and it may be straightforward for system developers to recognize this. System owners, since they do not deal with the internal details of a system in operation, do not experience this problem. The results from the three individual rankings indicate that work needs to be done in order to achieve a common understanding among all involved parties about what the challenges and problems in this interplay are.

By analyzing the three rankings of the three panels, we do see that some of the problems are considered by all panels as more important than others. Except for problem number five, these problems have gained a rank no lower than eight by all panels, and an average panel rank of seven or higher. In the following section, we will discuss each of the six highest-ranked problems of Table 6 by including the experts’ own comments on them.

1. **IT operations are not involved in the requirements specification, consequently technological and operational aspects are not sufficiently taken care of**

   The software engineering literature has long acknowledged the importance of including the viewpoints of the stakeholders in requirements engineering [40]. However, stakeholders are, in the literature, mainly assumed to denote the various roles within the user domain. The involvement of IT operations is seldom discussed. As this highly ranked problem demonstrates, IT operations are not satisfactorily involved in this activity in practice either. Overall, the experts perceive this as the most severe problem. This observation is in line with two previous case studies which support the existence and relevance of this issue [41,42].

   The experts’ qualitative comments from the brainstorming phase enlighten the matter: “those who write the requirements specification do not talk with IT operations,” “developers identify and specify customer requirements, but not operational requirements,” “functional requirements are regarded more importantly than the non-functional ones.” Some experts pointed out that know-how on this topic is scarce: “developers do not have competence in non-functional requirements,” while others claimed that this goes both ways: “each of the two areas has too little knowledge about each other’s needs and requirements.” System developers were also concerned about the fact that IT operations fail to specify areas where non-functional requirements are needed: “IT operations have not formally defined their existing requirements.” This makes it hard for developers to satisfy operational demands. Studies have concluded that non-functional requirements are not sufficiently addressed in the software engineering literature [43,44], which may explain the lack of competence among system developers and why this area does not receive adequate attention in development projects.

   An overall explanation for this highest-ranked problem is the fact that IT operations are either not engaged in the development project at all or that they enter the project too late. Either way, they are not able to influence the requirements specifications. As one system developer puts it, “If we are to design systems with operational qualities, IT operations and their competence must be involved early in the development project.”

2. **Communication and the flow of information between system development and operations are poor**

   In the early days of computing, it was normal for all IT professionals to fulfil development tasks as well as operational activities. However, as the complexity and number of systems grew, we have witnessed a gradual increase in specialization, and as a consequence, system development and IT operations have become two autonomous functional areas within the IT department [45]. Members of the two functions may not meet on a regular basis. Further, due to outsourcing, developers and IT operators may reside in different firms and may even be geographically separated [45].

   By analyzing the experts’ comments, we find several explanations as to why this problem exists: “personnel do not meet, they communicate only through e-mail and documents,” “system developers are requested to use an electronic service-desk system when requesting services from IT operations,” “the two groups use different terms (for the same things), they do not use the same language,” “there is a formal bureaucracy between the members of the two groups,” “following the routines is more important than solving a problem together,” “we do not inform each other about change of plans or changes in the solutions;” and “it is not clear who to contact in the other area about a certain issue.” There may be several reasons for this situation. In particular, it is a consequence of increased bureaucratization and specialization. In addition, the two groups do not have enough insight into each other’s areas, activities and standards. Many experts expressed their concern about this.

   It is noteworthy that system owners have ranked this the highest. This indicates that poor communication is highly recognized by those who are observing the interplay from a distance.

3. **Too little emphasis is placed on establishing satisfactory test environments for volume, performance and capacity testing**

   This is about vital non-functional requirements that are not tested properly before deploying the system into an operational environment, and it is thus related to problem number one. If the non-functional requirements are not defined, as a result of problem one, this is to be expected. However, even though requirements should be known, this problem states that an adequate test environment is seldom available for testing the requirements properly.

   Many experts commented on testing, and various aspects of this issue were raised. Overall, it is evident that testing is not prioritized. This may be due to lack of understanding and knowledge, lack of test requirements, lack of time and money and lack of a satisfactory production-like test environment, in which a realistic test of the functional and non-functional aspects can be carried out. Some of the comments given by the experts are as follows: “testing is not considered important, since errors can be fixed later on,” “developers are not focused on the challenges a new system brings to the production environment, and the need for testing a new system before it is deployed,” “testing is a burden for developers, it steals valuable time,” “IT operations do not have formal routines and requirements for testing,” “IT operations are not involved
when tests are planned for and carried out,” “the test and production environments have huge differences, tests are not real when it comes to non-functional requirements,” “testing is done only on a small scale, we do not carry out volume-testing,” and “we only test for functionality, aspects such as stability, response time, stress testing, recovery/restart, monitoring and surveillance, and scalability are not tested for.”

Although testing is a topic that is receiving more and more attention today, as witnessed from industry seminars’ agendas, it is obvious that in practice, testing, especially the testing of non-functional requirements is still not handled properly. Furthermore, it seems that building a production-like test environment is something that few can afford. As a consequence, system failures are identified after deployment, which may lead to system downtime, error-handling, severe security breakage, and often, the need to bring in additional and unplanned hardware, software and personnel resources, which is obviously not cost-effective.

4. Operations receive insufficient information about how the new system works because of missing knowledge transfer

Information systems developed today are often very complex, consisting of many different technologies, and may be redesigned several times during the project. Finally developed, the system is deployed and handed over to IT operations, and will have consequences for the existing operational environment where it will run. A new system may, for example, affect the configuration and capacity of existing computing and network resources, and it may also require and thus introduce a new technological platform to the operational environment. IT operations must plan for and implement the changes as necessary. In addition, IT operations will have the daily responsibility for operating the system and for handling errors and incidents that may occur. It is mandatory for IT operations to know system characteristics well.

Many experts commented that there is a lack of knowledge transfer to IT operations. The main reason for this, as many experts see it, is that IT operations are not involved in the project. Therefore, they do not learn about the system as it develops. Seldom is time set aside for training IT operations on the new system. Sometimes, it was explained, IT operations may learn that a new system is about to be deployed only days or hours in advance, and consequently, there is no time for them to study and learn the system characteristics. Since documentation is often scarce – another problem that was identified – IT operations have to operate a system they do not know enough about.

As the developers put it, “IT operations are not involved in the project, and do not learn about the new system beforehand,” “due to organizational distance, and since IT operations are not following the project, they do not gain information about how the new systems works,” “IT operations are the last link in the chain, and since projects frequently are behind schedule, there is no time to inform IT operations about how the system works,” “development does not prioritize documenting the system, they want the shortest way to production,” “the documents describing how a system is to be operated are defective,” “too often, no or little documentation follows the system,” and “if the new system runs on a new technical platform, IT operations may not acquire the required knowledge about this technology before the system is put into production.”

These statements are verified by the experts from IT operations: “we do not get information about the various details of a new system,” “the new system is insufficiently documented, which leads to misunderstandings and errors during deployment,” “we do not receive the training we need to control and operate a system, this leads to downtime and re-runs of batch routines,” and “we need information about all the relationships that a system has with the production environment, not only on the system functions themselves. Such information is scarce.”

The lack of suitable documentation is central to this problem, a deficiency frequently mentioned by experts from all panels. IT operations need documentation about how the system is built, its relationships with other systems, which parameters are set, how it should be monitored and logged and how incidents should be handled. Another interesting issue raised by system developers is that they often do not know who to inform. As one expert puts it, “Who does what in IT operations, who possesses the various roles, who are they?” It seems that such information is scarce among the developers, even for those with long experience in the organization. Another system developer mentioned that training and knowledge transfer are especially important since, overall, personnel in IT operations have a low formal educational level.

5. New systems are put into production too early, before they are complete

When is a new system ready for deployment? It seems as if the view on what is a deployable system differs among the panels. This problem was ranked very highly among both IT operators and system owners, but below average by the system developers. As one expert from IT operations commented, “Systems are frequently deployed too early, and IT operations are stuck with a system which is not 100% complete, while developers on their side regard the projects as finished.” And further, “a non-complete system leads to instability, error-handling, and bug fixes while in production, which is counterproductive.” Developers admitted to this: “in relation to deployment, it may happen that we decide to take a ‘short cut’ in order to save resources.”

The two groups – system development and IT operations – obviously have a different view on what is required for a system to be ready for deployment. For developers, readiness means that the required functionality is fulfilled. But in the view of the operators, this is not sufficient. They focus on what they call “operational readiness,” which means that all necessary operational technology, solutions and routines should be in place before deployment. In the eyes of IT operations, the lack of testing, documentation, information and training, amplify this problem. Overall, as one operator explained, “It seems that developers may have a ‘try and fail’ approach to deployment, and have greater tolerance for the unexpected, while we, the operators, have a more rigid perspective; the system must prove itself to be ready before it can be put in production.”

The experts gave several reasons for the existence of this problem. One dilemma that was mentioned is that “projects and project managers are evaluated on their ability to deliver on time, not on the operational quality of the system, as experienced in production.” It was also pointed to that for the customer, the business side, it may be so important to get the system in production on time that they choose to accept a system with operational shortcomings and even errors. This also causes stress for developers, as they have to handle bug fixes and adding new functionality in frequent, unplanned new releases. “An incomplete system means that developers are regularly involved in operational tasks.” This may continue during the system’s lifetime.

6. Operational routines for the new system are not established prior to deployment

Operational routine is a term that may have different meanings to different systems and in different contexts. Overall, for a system in production, this term may include practices for batch jobs and job scheduling, backup and recovery, console and monitoring management, logging, configuration and change management, access
management, user support, error management, database management, print and output management [46]. For a system to be fully operational, such routines must be planned for, implemented and delegated to personnel in IT operations [46].

This problem was ranked relatively high by all panels; highest by IT operations. Many IT operators commented that development projects have too little focus on how a system should be operated. “Again and again, we see that operational demands such as monitoring, logging and recovery are not defined by the project.” And the consequences, as they see it, are the lack of appropriate operational routines and thus unstable production. Without satisfactory operational routines, a system will not fulfill business needs or meet defined service levels, a consequence that IT operations are often blamed for. It is also worth noting that the lack of appropriate operational routines for one system may have consequences for other systems as well, thus reducing the reliability and availability for the system portfolio as a whole. The solution, as all panels see it, is to include personnel from IT operations early in the project in order to make operational routines ready at deployment. In addition, some experts noted that it is necessary to build up developers’ competence in this area, as they lack knowledge about the technological environment and its operational standards.

The cause of this problem can obviously be explained by the existence of some of the other problems. If IT operations are not involved in requirements specification or do not receive sufficient information about the new system, or if documentation is scarce and if the communication and information flow between the two areas are poor, it comes as no surprise that operational routines are not established prior to deployment.

5.1. Summing up

The six issues discussed above indicate some of the most important problems concerning the interplay of system development and IT operations as perceived by the experts. The sheer amount and variety of problems mentioned and the respondents’ explanations confirm that this interplay needs attention. For balance, it should be mentioned that two of our respondents consider these problems to be minor. One system developer, for example, commented that he had very good and professional cooperation with IT operations. Still, the majority of experts agree that this interplay has a lot of challenges that do need to be handled seriously.

Further, our observations contribute to the recent discussion about what important competencies are needed [9] to form an IS capability. With all the facets to the different problems found among the experts’ perceptions in the brainstorming phase, our data suggest that interplay of system development and IT operations, or worded differently, alignment of system development and IT operations, should be regarded as a crucial ingredient in IS competence, and this competence should be explicitly recognized in more detail as contributing to the overall IS capability under Peppard and Ward’s fifth macro competence, “deliver solutions.”

5.2. Implications for practice

It is evident that many of the identified problems continue from project to project, along the lines of how Lyytinen and Robey [3] demonstrate the limits of organizational intelligence to learn from previous experiences, i.e., failure to learn. The comments from the experts imply that the “failure to learn” situation may exist in the interplay between development and operations. That is, several IT organizations may have institutionalized incomplete practices hindering successful interaction. Further, in our discussion, we concluded that we need to look upon such interplay as part of strategic competence. Thus, any continuing problem in this domain can be seen as a failure to learn how to enact this competence.

The causes for the learning failure in the interplay could be many, but first of all, it seems as if a few problems originate due to a lack of resources, which may further be a symptom of lack of organization design, educational barriers and the incentive structure for facilitating learning [3]. The customer may not want to pay for IT operations participation in the development project. His focus is on getting as much functionality as possible. The consequences of this approach could be a system that is not able to support business needs, although all the functionality is there. Perhaps a reduction of functionality for higher operational quality would be a better strategy for the customer? Even if IT operations are invited to participate, personnel resources are often scarce, and IT operations are not able, although they may wish to set aside personnel for the development project. This is a senior management issue. The economic consequences of a poor system development/IT operations relationship have not been empirically quantified, but it is well accepted that system downtime has caused severe economic losses for many businesses. Models for customers’ participation, as we see in the agile methodologies [12,25], can be applied as a reference for how to involve personnel from IT operations personnel into development projects. Similarly, a different suggestion could be that developers should spend time within IT operations to get an understanding of their perspectives and practices.

Taking an overall view of the six highest-ranked problems (cf. also Table 6), we find that four of them (1, 3, 5 and 6) relate to three central activities in a system development project according to our grouping: requirements specification, testing and deployment. Two of the six highest-ranked problems relate to (the lack of) cooperation and communication between these stakeholders in general. For practice, this observation implies that IT operations are an important stakeholder at least in some, if not all, system development activities. Moreover, their participation and involvement should be ensured by an increased focus on cooperation and communication in general and during requirements engineering, testing and deployment in particular.

A wider look over the 17 ranked problems (Table 6) reveals a few additional trends of interest. A need for an increased focus on the deployment phase becomes even more visible with two additional deployment issues running alongside the two already identified among the top six issues. Another task-specific focus relates to the planning and scheduling phase so that the diverging foci of IT operations and system development could be united. Three issues relate to the general-level organizing of development work to enhance the interplay. This indicates that, beyond the focus on developers and operators as such, the management should, in many cases, take more responsibility in providing organizational capabilities of better cooperation between operations and development. Two issues among the top 17 related to general-level competence on operations among system developers. This may imply that the viewpoint of IT operations should be included to a greater degree in the programs educating future IS professionals, especially for large-scale and more tailored system implementations (which were in the focus of our study). See also Pollard and colleagues’ [47] proposal for integrating system development and IT operations in education. Finally, problem number 10 refers explicitly to the lack of focus on continuous improvement. Hence, the “failure to learn” was identified at least by a good number of our experts, and it should also be paid more attention to in practice, accordingly.

From the original consolidated list, no outsourcing-related topics reached the top 17 items in the list, which may imply either that outsourcing is not much used for strategic Norwegian IS projects or that it is not a significant problem in general, beyond particular projects. Additionally, no issues related to the economy and contracting processes, or IT operations personnel competence per se, were prioritized among the top 17 items.
5.3. Limitations of the study

This study is based on the Delphi method and a limited number of experts. Subjects were chosen for their knowledge and experience in system development projects from the perspectives of system development, IT operations, and system owners. Our sample represents a variety of experts working for in-house IT departments and for providers of system development and IT operations services. The Delphi method is a popular method for extracting information from domain experts and for obtaining the most reliable consensus of opinion, and has been applied in various disciplines, including information technology [32]. The method, however, is not without limitations and has also been criticized for its methodological inadequacies [31]. One problem is how to measure “expertise”. Although we have been careful to establish panels with experts who have thorough understanding and knowledge of the topic in question, their absolute expertise level has not been measured beyond our subjective evaluation. Another debated question is whether a group opinion is superior to an individual opinion, and how to combine individual opinions is also problematic. Last, a methodological issue is the mode of communication between the experts and the researchers. E-mail, as applied in this study, is a resource- and time-effective communication channel, but also a narrow one. Since our experts are geographically dispersed and difficult to reach with synchronous communication because of their job characteristics, we could see no other way of communicating with them other than via e-mail. Also, it should be mentioned that many felt that participating in this research was time-consuming and we consequently had difficulties collecting answers during the second round of ranking.

The study has been conducted within a Norwegian context. The experts reside in this region and their experiences of the studied area are based on this. This may influence the experts’ perceptions and suggestions. From cultural studies, we know that organizational styles and practices may vary according to geographical region [48].

As previously mentioned, the system development panel did not reach a significant level of agreement in the first round. Still, we chose to continue with the first-round rankings and presented these for the developers as the input for the second round. We achieved clearly significant results in this round, but maybe as a consequence of agreement with a more or less random list. This is clearly problematic, but the choice can be defended, as the highest-ranked problems have been identified and considered as important by system developers participating in an earlier study on this issue [44].

Finally, we will repeat that none of the panels reached the strong concordance level prescribed (0.7). The results indicate that there is some agreement in all panels, but the lack of concordance also suggests that problems are diverse and will vary from one context to another and from one project to the next.

6. Conclusions

In this study, we have approached issues relating to the interplay of system development and IT operations in system development projects, and have answered questions about what problems are viewed as the most severe in the given context and their causes. First, we used a systematic procedure to elicit and define a list of 66 problems organized into seven main groups. Since the list is based on input from 42 experts with long and relevant industry experience, we are fairly confident that the list is comprehensive and well grounded. Three panels, twenty system developers, 12 IT operators and ten system owners were then organized, and these panels reduced and ranked the list of problems using a rigorous data collection method known as a ranking-type Delphi survey [29]. Three independent ranked lists, one from each panel, were produced. The panels found the following to be the six most serious problems in the interplay of system development and IT operations: (1) IT operations not being involved in the requirements specification, (2) poor communication and information flow, (3) unsatisfactory test environments, (4) lack of knowledge transfer, (5) systems being put into production before they are complete, and (6) operational routines not being established prior to deployment. All in all, the results imply that IT operations should be regarded as an important stakeholder throughout several systems development activities, especially requirements analysis, testing and deployment. Moreover, such involvement should be facilitated by an increased focus on enhancing cooperation and communication.

Further, we identified a few causes of these problems by analyzing the experts’ qualitative comments from the brainstorming phase. In line with Lyttinen and Robey [3], we conclude that learning failures may be given as one of the general-level reasons for these problems, which should be tackled with increased managerial attention to improving and cultivating more collaborative and communicative projects among system development and IT operations. In addition, we propose the competence of aligning development and IT operations of new systems in a detailed manner as an extension to Peppard and Ward’s IS capability model in order to establish the importance of this area.

In this study we have investigated each of the problems separately. A natural path for future work is to delve into how the different problems relate to each other in terms of their potential cause–effect relationships.

References


[43] O. Stadheim, Fokus på og innhold til ikke-funksjonelle krav innan systemutvikling (The focus on and handling of non-functional requirements in system development literature), Department of Information Science and Media Studies, University of Bergen, Bergen, 2007, p. 76.


