Test Governance Framework for contracted IS development: Ethnographically informed action research

Michal Doležel [a,*], Alena Buchalcevová [a]

[a] Department of Information Technologies, University of Economics, Prague, Czech Republic

[*] Corresponding author.
E-mail address: michal.dolezel ( ) vse.cz (M. Doležel), Phone: +420 420 224 09 5460, Postal address: W. Churchill Sq. 4, 130 67 Prague, Czech Republic

Context: Over the past two decades, interest has increased in software development and testing outsourcing. Although the decision to outsource development or test processes is founded on various background motives, e.g., costs, capacity, time-to-market, etc., additional factors that influence the outsourcing relationships are frequently overlooked by client organizations.

Objective: The main objectives of this paper are to investigate the role of testing in contracted software development projects and examine the interactions between both parties during the process. A Test Governance Framework (TeGoF) was developed to propose an organization-wide but project-centred mechanism for control of the test process in contracted software development projects.

Method: The principles of design science were applied to develop the TeGoF. The principles of ethnographically informed action research were used to evaluate this management artefact in an industrial context and examine the impact of a particular organizational setting on the framework.

Results: Of a total of three projects, the TeGoF was applied smoothly in only one case. However, this observation should not be interpreted as a TeGoF deficiency because its primary goal is to define a powerful detection mechanism to cope with quality-related issues in contractual relationships. In this sense, the TeGoF proved itself as a feasible tool. Additionally, an analysis is presented that describes the factors that contributed to the implementation difficulties of the TeGoF principles in the two remaining projects. Finally, the relationships among trust, control and power are indicated as well as the potential influence of national, organizational and occupational culture.

Conclusion: The main contribution of this paper consists in the development of the TeGoF as a tool that pinpoints significant limitations in the current research related to control issues in the domain of contracted software systems testing. Additionally, the authors analysed key factors that influence the success of the TeGoF in client organizations.

Keywords: software testing; outsourcing; control; software engineering management; culture

Final accepted manuscript – please cite as:
DOLEŽEL, Michal; BUCHALCEVOVÁ, Alena. Test Governance Framework for contracted IS development: Ethnographically informed action research. *Information and Software Technology*, 2015, 65: 69-94.
http://dx.doi.org/10.1016/j.infsof.2015.03.003

1 Introduction

Over the past two decades, interest has increased in software development and testing outsourcing [1,2]. It has become fairly common for numerous business organizations to outsource development of their systems either completely or partially. In fact, software testing is an activity eligible for potential delimitation from core development activities to be conducted autonomously in an offshore setting [3]. Thus, a proposal for outsourcing software testing often results in contracted software projects or services.

Although the decision to outsource relevant processes is based on various background motives, e.g., costs, capacity, time-to-market, etc., additional factors that influence outsourcing relationships are frequently overlooked by client organizations in the initial phases of the decision-making process. Previous research has stressed a number of such factors, certain of them more tangible (e.g., contract model [4]) but others without even a theoretical possibility of formalization or standardization of these relationships (e.g., trust [5]). Although much of the research is also focused on the negative consequences of global software deliveries (i.e., the role of geographical distance, cultural clashes, etc. [6]), more concrete managerial challenges for business organizations are frequently ignored in the research literature. Such challenges include design of appropriate client project management and control mechanisms [7].

To design these mechanisms, it is necessary to first understand their background. This paper highlights the importance of contracted software development in the IS domain. In these cases, contracted software development typically involves tight cooperation and interactions between a business organization (referred to as the “client”) and one or more software-intensive organizations (referred to as “contractors”). A broad consensus exists that the client must manage an outsourcing relationship with its contractors [8]. However, the possibilities of direct client control over the development process are significantly reduced in the case of outsourcing as well as the necessary decision-making. Moreover, concrete opinions on the level of client involvement in software development and test management will vary.

To this end, we focus on contracted system testing in this paper. In general, we challenge an anecdotal idea of the understanding of User Acceptance Testing (UAT) [9] as the only phase that typically allows the client to control the contractor outputs related to development and testing. A similar view of this phase is common at least in the classic models of contracted software development projects [4]. However, our position is somewhat different. We believe that the client should be provided with early insight into the test process. Thus, the main objectives of this paper are to emphasize the role of client involvement in the contracted test process and to propose an appropriate management tool. In response to these challenges, we developed a Test Governance Framework (TeGoF) based on our earlier conceptual paper [10]. The TeGoF represents an organization-wide but project-centred mechanism for clients. To evaluate this mechanism, we conducted an action research study in a client organization. Primarily, we sought to understand the social interactions between the client and contractors relevant to such a tool that controls the entire contracted test process.

In our understanding, Test Governance is a *set of processes and responsibilities that enables client organizations to exercise activities related to control and coordination of the test process, typically in the context of contracted software development projects*. These high-level processes allow oversight of test management and test execution activities ranging from the organization level to the project level. Such activities should consider the client business alignment perspective, i.e., stress the future operational context of a software system. However, these activities should be clearly differentiated from day-to-day test management activities conducted by each of the contractors. Essentially, a balance between an inefficient day-to-day control of every particular detail and total omission of control must be found. In this sense, we attempt to identify an adequate level of control in the process of TeGoF design.

Overall, we contribute to the body of knowledge in two areas. First, we propose software engineering/information technology management guidelines for client organizations, provide analysis of evaluation results, and discuss novel methodological aspects of the study. Second, we also attempt to draw attention to a potentially common and overlapping agenda of Software Engineering/Information Systems (SE/IS) research with a significant inclination towards SE concepts. We believe that this work represents a novel and fruitful perspective from which to view such issues because we treat problem-driven client-based perspective as a neglected issue in the current computing research [11]. Similar research problems tend to be located in the disciplinary gap between theory-driven IS research [12] and the current relatively low interest in SE management concepts [11].

This paper is organized as follows. Following the introduction, Section 2 presents the conceptual background. Section 3 focuses on exercising control in contracted development and testing, and Section 4 introduces the research methods. Section 5 identifies the initial challenges resulting in the framework development, defines the guiding principles, depicts the basic structure of the TeGoF, and describes all of its processes. Evaluation of the framework performed on three real projects using an action research methodology is outlined in Section 6. Finally, the research results are discussed, and concluding remarks are presented in the final section.

2 Conceptual Background of Control

In this section, we first place our research into the broader context of computing disciplines and discuss our cross-disciplinary adoption. Next, we outline the merits of agency and control theory. Furthermore, we make a necessary conceptual connection between control theory and management praxis. Consequently, we illustrate certain of the practically focused concepts related to control of software development. Finally, we review the human-related factors with possible impacts on the proposed framework.

2.1 Disciplinary Boundaries

In this research, we attempt to merge both the theoretical and practical perspectives of control. In fact, our motivation for this paper lies in our belief that the above-mentioned disciplinary gap could be bridged and that such an approach would be salutary. To this end, we benefit from concepts of both the IS and SE disciplines. Nevertheless, the commonly held wisdom suggests that the IS discipline is client-oriented, whereas the SE discipline is contractor-oriented. Based on this hypothetical perception, one may question whether we have actually conducted SE or IS research. Nevertheless, we would dispute such a simplified understanding and differentiation.

Although control of systems development is an interesting research problem in both IS and SE, it is rather important to acknowledge that both disciplines currently represent different epistemic community tents. The foundation of this argument lies in the fact that the majority of IS research (as represented by top journals) is theory-driven, and the SE research is problem-driven [11–13]. Our research falls into the latter category, which means that we conducted SE/IT management research with strong practical implications and a lesser contribution to theory. However, based on the rather atypical organization context, the formal disciplinary differentiation may be elusive. We explored the management practices of the IS department within the business organization, different approaches by which IS personnel manage the contractors that supply the contracted software systems, how these contractors interact with IS personnel, and how these contractors manage their own software practices.

Based on our exploration, we argue together with Iivari that in similar cases:

> *"the border between Information Systems and Software Engineering in particular is like a line drawn on water, as exemplified by the distinctions information systems vs. software artefacts, IS development vs. software engineering, management of IS functions/department vs. management of software processes/organizations"* [13].

Still, the theory has its place in this research. Although more pragmatic approaches are typically pursued in SE research, this does not imply that agency and control theory are not applicable or useful for conceptualization of certain problems related to management of software development. In the next section, we provide such a conceptual background.

2.2 Theoretical Lens

From the theoretical point of view, information systems within organizations are complex socio-technical elements that exceed the vision of software purely as an IT artefact. In this sense, the crucial role of the organizational context is often stressed. Following this grounding, information system development represents a challenging set of activities that involves various actors and encounters numerous organization and process constraints [14]. Exercising control is a tool for coping with such constraints [15].

2.2.1 Control

The word *control* represents a clear ambiguity from the beginning of management research [16]. In this paper, we draw upon a broad view of control primarily in the behavioural sense. This *social perspective of control* is captured in a significantly large number of theoretical contributions to the management and IS research literature [14,15,17,18]. Nevertheless, our research objectives in the software realm are mainly pragmatic and are positioned towards the role of control in the area of contracted software development and testing. Thus, we attempt to place a *pragmatic view of control*[1] into a relevant context later on. From one standpoint, this latter view of control is appealing and graspable; however, it also can be perceived by social scientists as over simplified and *cybernetic* [18]. We seek to understand both views in our paper and relate them to the area of contracted test processes.

2.2.2 Agency and Control Theory

Turning to concrete theories, our research is based on agency theory [19] and control theory [15,17], which are two well-known though controversial theories originally rooted in organization research. In agency theory, an agency problem arises when different parties have different goals. Looking at contracting relationships within the business context, they do represent a typical example of such situation. In this case, work is delegated by a principal

[1] For example, the Project Management Body of Knowledge (PMBOK) Fifth Edition [28] defines control as "Comparing actual performance with planned performance, analyzing variances, assessing trends to effect process improvements, evaluating possible alternatives, and recommending appropriate corrective action as needed." A similar definition was also adopted for the term *test control* by the ISTQB glossary [9].

(client) to an agent (contractor), but the principal cannot verify what type of work the other party actually performs (because of costs or the factual impossibility of the current set-up).

Agency theory has been widely criticized for being over-simplistic and even misleading [e.g., 20]. However, in certain situations, it is reasonable to expect that conflict between the principal and the agent will inherently exist and that the theory is valid. An example of such a situation typically involves a traditional contract model in IS relationships [4] distinguished by a strong pressure on the lowest possible price from the principal (client) side. Many such cases can be found in the literature [e.g., 21]. However, even if this behaviour is not observed and a more "win-win" strategy for contract price is followed, the expectations among both parties can vary substantially and might result in critical issues as the project evolves.

A possible solution used to settle the agency problem involves the involvement of a control mechanism. We perceive control in the broad sense, i.e., "attempting to ensure individuals act in a manner that is consistent with achieving desired objectives" [17], which means that control represents a "regulation of behaviour" [18]. Two basic categories of control modes are defined: formal control and informal control. Formal control consists of (a) outcome control and (b) behavioural control and is exercised in a formalized manner. Although outcome control "focuses on the outputs … without regard to the process" [17], behavioural control "seeks to influence the process … by explicitly prescribing specific rules". Informal control consists of (a) clan control and (b) self-control. These two control modes "differ from formal controls in that they are based on social or people strategies". In general, clan control includes (in addition to other forms) use of such strategies as influence tactics that appeal to a common goal, and self-control includes individual responsibility and internal control processes.

2.3 Software Process as a Form of Control

In this section, we demonstrate the conceptual connection between the above-mentioned theories and a pragmatic approach to control in SE/IT management and relevant research. Referring to the SE literature, Kirsch [15,18] claims that an Information Systems Development (ISD) process is typically continuously controlled by forms of control that differ from outcome control. In Kirsch's understanding, control of the development process is vital as the software system evolves. In line with this argument, we suggest that several mechanisms (grounded in software engineering management, process improvement practice, and commonly known under the term Software Quality Assurance/Management) adopt the prescription of software processes as one form of behavioural control mechanism. In this sense, these approaches can be examined and empirically explored.

In the SE domain, this approach was shown by McBride [7] (as an example), whose empirical study is based on conceptual synthesis of project monitoring, control and coordination concepts. Similarly, the work in [22] provides a quantitative study that builds on control theory. In applying management and organizational theory, these papers make a clear contribution to SE praxis. However, we admit that a typical approach to issues of control in SE is less influenced by an explicit application of the theories introduced above. Thus, we proceed with our review by focusing on key well-known concepts, especially in the practitioner arena and those related to contracted software development.

2.4 Examples of Control Artefacts in the Practitioner Arena

In this section, we first provide a brief historical context to demonstrate the conceptual origins of our thought process. Although this short historical summary cannot capture the earlier and later development in its full scope, we believe it is important to a better understanding of the design element in our research. Therefore, we also briefly review two contemporary SE/IT management artefacts with a strong practitioner focus. We consider these works essential for our work (as demonstrated later on) because we use them as a starting point for our own artificial design. These management artefacts include: 1) the PMBOK process groups and 2) the ISTQB test process conceptualized as a portion of ISTQB activities. For the purpose of this study, we present only a brief introduction of key concepts from both areas and do not aim to provide a comprehensive review of either PMBOK or ISTQB. For a recent review of the PMBOK and its relationship to software development, see [23]. Similarly, Sánchez-Gordón and Moreno [24] provide a decent overview of the ISTQB and related research. Nevertheless, the evidence-based research studies related to these artefacts and frameworks are still quite scarce (cf. [25]).

2.4.1 Historical Background

Broadly speaking, one can consider the original paper by Royce [26], which questioned the waterfall model in 1970, as a key initial contribution to control artefacts. However, we believe that with respect to our interest in management of artefacts and contracted development, the later contributions mentioned below are of particular importance to the topic of our paper. Thus, we trace the historical roots of control-related activities in the software engineering domain to the 1980s and 1990s.

In 1994, the well-known U.S. military standard MIL-STD-498 replaced two previous standards, namely, the DOD-STD-2167A and DOD-STD-7935A. Although the DOD-STD-2167A was previously used in the mission-critical domain and the DOD-STD-7935A in the IS domain, the MIL-STD-498 unified the development process of both classes of software systems. Although the MIL-STD-498 was applicable in several contexts, we particularly stress its high-level guidance through the acquisition process, i.e., contractor management capabilities.

In closely examining the MIL-STD-498, one can find at least one key principle that this standard shares with our approach described below, which is the general *visibility* of the software and test process from the acquirer (i.e., client) perspective. The MIL-STD-498 provided a clear guideline to improve the visibility in this sense.

Similarly, in internal material titled the “COTS GUIDE” [27] dating back to 1993, we also learn how the COTS acquisition practice appeared in the U.S. Air Force premises during this period. In this domain, acquirers were instructed to obtain and assess “contractor test results” [27], and similarly, they were asked to benefit from “[o]bserving contractor testing at his facilities” to minimize their own testing.

It is interesting to note that ideas compatible with our design approach were considered as *best practices* in military computing three or so decades ago. However, such ideas have not fully filtered into today’s software engineering management reality in the business segment. This inconsistency might be due to different beliefs with respect to control. We further analyse this claim in Section 3 in which empirical research is reviewed.

2.4.2 PMBOK

The Guide to the Project Management Body of Knowledge (PMBOK) Fifth Edition [28] is a generic project management standard that provides guidelines for project management practices.

Role of the PMBOK in our research: For the purpose of TeGoF publication, we opted for the PMBOK framework to conceptualize the TeGoF's basic structure and demonstrate its benefits. This decision does not imply that the TeGoF is to be used only in the context in which the PMBOK is applied. However, because a tight conceptual connection is already established between the Software Engineering Body of Knowledge (SWEBOK) and PMBOK, we find this linkage to be appropriate. More specifically, the SWEBOK considers software project management as one component of the "Software Engineering Management" knowledge area. Thus, the SWEBOK also provides guidelines in this sense.

Key elements: A key component of the PMBOK standard consists of 47 processes that are logically grouped into five process groups. These process groups are not directly mapped to project lifecycle because the PMBOK explicitly states that "The Process Groups are not project lifecycle phases" [28]. However, a certain level of correlation will occur in reality. The following process groups are introduced by the PMBOK:

- *Initiating Process Group:* Activities related to the definition of a new project (or phase)
- *Planning Process Group:* Definition of scope and objectives
- *Executing Process Group:* Activities executed to fulfil objectives of a project
- *Monitoring and Controlling Process Group:* Activities that track progress and performance to keep a project running smoothly
- *Closing Process Group:* Activities related to guiding a project to its end

Further grouping of the processes is provided at the level of knowledge areas. The following knowledge areas are defined:

- Project Integration Management
- Project Scope Management
- Project Time Management
- Project Cost Management
- Project Quality Management
- Project Human Resource Management
- Project Communications Management
- Project Risk Management
- Project Procurement Management
- Project Stakeholder Management

In the end, the process core of the PMBOK can be visualized in the form of a 10x5 matrix. In this matrix, the rows represent respective knowledge areas, and the columns represent process groups.

Finally, we briefly turn our attention to the relationship between the PMBOK and control. The PMBOK introduces an important organization entity, i.e., the Project Management Office (PMO). The main function of this organizational entity can be characterized as process standardization and knowledge sharing but with different levels of involvement and prevailing activities taking place in different organizations. We are concerned with a particular configuration of the PMO labelled as Controlling. In this case, “PMOs provide support and require compliance through various means. Compliance may involve adopting project management frameworks or methodologies, using specific templates, forms and tools, or conformance to governance” [28], which means that the TeGoF can be used in this control mode either by client’s PMO or the individual client’s project managers to apply the principles of test governance in contracted software projects. However, we previously noted that the term “control” is formally defined by PMBOK in quite a narrow sense.

2.4.3 ISTQB Test Process

The International Software Testing Qualification Board (ISTQB) is an international organization that provides standardized certification schema in the area of software testing. The ISTQB can be considered as a body of accumulated knowledge relevant to testing. This body of knowledge is codified in a freely available syllabus and glossary [24].

Role of the ISTQB test process in our research: Similar to the PMBOK, we decided to follow well-recognized and practitioner-oriented knowledge to conceptualize the test process background related to the TeGoF. Although the ISTQB is not recognized by the current version of the SWEBOK, it is a widely known and appreciated *de-facto* standard related to the field of software testing, which means that the ISTQB is frequently used as a guideline for exercising control (in the broad sense) in the test domain.

The ISTQB Test Process: According to the ISTQB glossary [9], a generic test process “comprises test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities.” Sánchez-Gordon and Moreno [24] argue that this test process usually must be tailored to the specific needs of an organization. In our research, we use the generic form of the ISTQB test process described above and depicted in Figure 1.

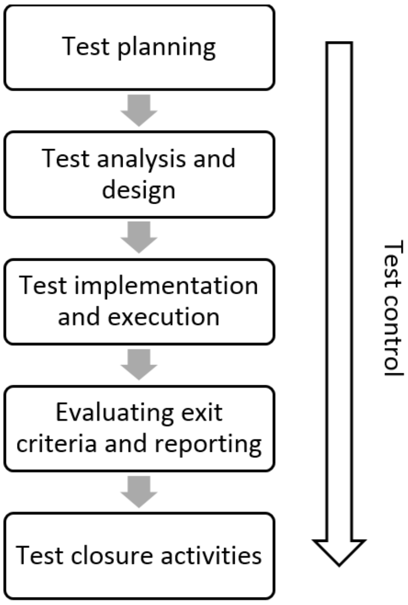

Fig. 1. The ISTQB test process

As mentioned in Section 2.2.1, the ISTQB understands *test control* in the narrow sense. Our own definition of test governance also encompasses a high-level test control process as one partial element. However, we fully acknowledge that an effective and efficient test governance (control) mechanism must stem from a broad understanding of control that includes utilization of “social or people strategies” [15]. Based on this argument, we proceed with a literature review that identifies such elements of control. Thus, the social perspective of effective control is stressed in the following section.

2.5 Human Factors Related to Control

In this section, we present the reasons for why people-related aspects were considered prior to artefact construction. Consequently, we underline those aspects that we consider vital for future deployment of the TeGoF in organizational reality.

2.5.1 People versus Management Artefacts

As further discusses in Section 3, the TeGoF is a type of SE/IT management artefact of similar designation as software process maturity models or the organizational structure of software development department [29]. These management artefacts are best described as a “future organizational reality” [30] and are tightly bound to the theoretical concept of behavioural control [18]. Because organisations are complex social systems, a simplistic *engineering view* of design and deployment of management artefacts is inappropriate [31]. In relating this view of the SE discipline, this argument was repeatedly proven, especially in the Software Process Improvement (SPI) area, in which a set of non-trivial people-related factors was identified to have an enormous impact on SPI implementations [32,33].

Considering the TeGoF, we were fully aware of the fact that an adoption process is typically influenced by a plethora of such factors, and no *simple* or *universal* solutions exist as represented by the process framework itself. Typically, similar management frameworks (control artefacts) are always subject to simplifications suffering from inadequate capture of real-world issues and subjects of potential organizational struggles [30]. In the initial phase of our research, i.e., prior to artefact design, we undertook a non-formalized literature review to identify key contextual factors in terms of their potential influence on the process framework adoption. This form of literature review together with a parallel collaboration with industry is claimed as common among design science methodologies [34]. In this study, we identified the concept of *culture* as the key contextual factor. Essentially, we believe that such artefacts must be always *compatible* with the organizational culture to become successful management tools.

2.5.2 Organizational Culture and Climate

Organizational culture is considered one of the primary people-related factors. Schein [35] views organizational culture as consisting of three mutually related levels. These levels are: “(1) visible artefacts; (2) espoused beliefs, values, rules, and behavioural norms; and (3) tacit, taken-for-granted, basic underlying assumptions” [35]. It is reasonable to expect that organizational culture will influence many contextual-sensitive variables that are important for the research, i.e., the client’s tendency to put enormous pressure on the lowest possible price, the level of control generally perceived by the client as adequate, etc. The importance of *quality culture* with reference to software development is stressed, for example, by Sommerville [36]. From our point of view and related to the TeGoF, quality culture in the test domain can be viewed as a *perceived level of an appropriate amount of testing conducted during the System Testing phase by a contractor*. In the case in which these beliefs greatly vary between both parties, this approach carries the possibility of a *quality culture clash* resulting from opposing perceptions of appropriate values.

Although clear evidence exists of “paradigm wars” [37] that can be found in the literature and question the difference between *organizational climate and organizational culture*, in line with Schein, we distinguish among these two vital concepts. In brief, **organizational climate** “is better thought of as the product of some of the underlying assumptions and is, therefore, a manifestation of the culture“ [35]. In this sense, the meaning of “quality culture“ in the form introduced by Sommerville intertwines with the understanding of the organizational or group climate.

With reference to this concept of culture and climate, we also stress the importance of an organization’s ability to cope with bad news emerging from troubled projects. In the literature, this situation is often referenced as “whistle blowing” [38], and the opposite behaviour is referred to as “keeping mum” [39]. “Keeping mum” is characterized as a situation in which a person involved in the project, from either the contractor or client side, refuses to transfer unwelcome or negative information related to the progress of the project [40]. Such information includes but is not limited to facts related to a project’s tendency to either completely fail or to experience a major delay, budget overrun, etc. Conversely, “blowing the whistle” is characterized as openly and completely providing information related to a troubled project. However, this situation can be associated with potential negative consequences for the whistler-blower [38]. We included an examination of these factors during the framework evaluation.

2.5.3 National Culture

We did not consider national culture in the design component of our research initially[2]. However, broadly speaking, national culture might become a potent and seminal factor in a particular environment in which the TeGoF is deployed, and thus, we stress its potential influence.

For our purposes, the construct of national culture can be understood in terms of the same inherent properties as the construct of organizational culture (see above). Cross-cultural differences have recently become a seminal research topic, especially in Global Software Development research [39,41]. Various quantitative models of national culture theoretically can be used for cross-cultural comparisons, but this approach is subject to critique from qualitatively oriented researchers who generally consider such models as “simplistic and … suffer[ing] from theoretical and methodological flaws” [42].

Related to testing, Shah and Harold [43] provided an interesting ethnographically informed probe into a client-contractor setting that enabled cooperation between an Indian contractor company and its clients located in Japan and the U.S.; their work stresses a significant research gap in this area and provides vital insights. However, these results should be interpreted with

[2] A client company operated within the realm of on-site development interactions with contractors and no other nationality was directly involved on-site. For accuracy, two points should be mentioned. First, two of our informants were actually Slovaks working in the Czech Republic. Due to the long-lasting and intertwined history of the Czech and Slovak nations and the common location of the work (Czech Republic), we do not consider them as members of a different national culture for purposes of differentiation in this research. This presumption is in line with data from the participant observation, which indicates no support for such differentiation inside the researched client organization. Second, a multinational consulting company represented by Czech employees was involved in one case. In the appropriate sections, we provide the relevant details.

care and supported by further in-depth research that examines Japanese versus U.S. client management practices related to contracted testing before articulating a possible generalization.

In general, we believe that numerous organizational routines might be a residuum of historical management interventions. Such interventions could be based originally on a rich spectrum of individual motivations of respective managers. Therefore, it is always necessary to understand the broader context of the respective organization to reach valid conclusions at the national culture level. In the Shah and Harold's case, missing access to the Japanese client team inside Japanese client organization appears to be the key to understanding the intrinsic motivation of Japanese employees with respect to their behaviour. We were concerned with understanding the employees' motivation in the case of the researched Czech client organization and provide additional thoughts on this topic in the Discussion (Section 7).

3 Exercising Control in Contracted Development and Testing

This paper has focused on rather abstract concepts related to control thus far. In Section 2, our aim was to provide a necessary conceptual understanding in three areas. Thus, the relevant theoretical background, relevant examples of control artefacts used by practitioners, and human factors that possibly influence control behaviour were introduced. The next section represents a synthesis of related empirical work relevant to control in contracted software development with special attention on contracted testing. Finally, we articulate the need for Test Governance in a contracted environment.

3.1 Problem of Unsystematic Control

Development of a custom-built software system as a component of a complex ISD process is performed either internally, i.e., directly by an IS department of a client, or externally, i.e., through contracted software development. In choosing the internal option, the major benefit for the client lies in significant oversight of the entire process. Despite this fact, contracted software development is quite common among businesses. Nevertheless, the control, governance and coordination practices focused on contracted software systems development appear to lack relevant consensus.

For example, Gopal and Gosain [44] examined 23 Indian software firms and their outsourcing relationships with clients and rejected the hypothesis of a positive expected effect of behavioural control on software quality and results. Furthermore, Tiwana and Keil [45] focused on distinct differences in control modes among internal and contracted projects and found similar paradoxical patterns. In contrast, Choudhury and Sabherwal [17] presented results based on research into contracted projects that went awry and suggested that IS client managers should pay an adequate attention to issues of control and *guard* contractors from the beginning; they additional claim that "it is a mistake to rely extensively on vendor self-control" [17].

Overall, theory-driven empirical studies conducted within the IS discipline and anecdotal evidence gathered in the initial phase of our research depict a hazy picture of an unsystematic control process that is currently exercised by certain business organizations that acquire custom-built software. For example, although appropriate and adequate system testing is generally considered as one of the key success factors in contracted software projects [46], it is likely to be among the first activities cut by a supplying contractor as soon as a strict project deadline approaches or when the project economics are jeopardized (at least in non-mission-critical applications) [47].

Client organizations appear to tolerate this situation because references to preventive management arrangements are rarely found in empirical material. We offer a possible explanation that reflects our opinion. Although the postulate that contractors attempt to maximize profit and clients aim to minimize costs is generally accepted based on the above-mentioned theoretical presumptions of agency theory, conclusions that address these approaches and respect the "win-win" principle have not yet been made clear [48]. Thus, both parties are vulnerable to the opportunistic behaviour of the other party [2].

Relying mostly on these sources, we can speculate that current organizations in the business segment typically use different concepts to manage their contractors rather than practitioner concepts similar to the MIL-STD-498 or its successors (e.g., ISO/IEC/IEEE 12207). Numerous business organizations do not appear to use behavioural forms of control of contractors but likely tend to rely on other forms of control, i.e., outcome control, contractor self-control (trust), and clan control (informal influence). Nevertheless, we speculate on the reason for the inconsistency between the presumably proven practices of the MIL-STD-498 or similar concepts and the observed praxis. This curiosity initially triggered our research in the contracted test domain.

3.2 Controlling Contracted Test Process

Generally, conclusions similar to those stated previously can be articulated for high-level management and control of contractor testing. Again, it is difficult to depict a complete picture when only few studies on the issue of control of contractors in the test domain are available. Although the SE and IS literature has surveyed outsourcing relationships for a significant period of time, only a small amount of information exists with respect to contracted testing.

At minimum, we define a starting point that does indicate contractor opportunistic behaviour within testing. For example, Sabherwal [2] briefly reports findings on a project in which "[a client] project manager believed that the vendor personnel shirked their responsibilities for software testing, especially *when they [i.e., contractor/vendor] realized that he [i.e., client manager] was quite thorough in testing*. The project manager felt that 'their programmers developed the sense of *let's program it and give it to him to test*'" (emphasis added). In fact, our own research was triggered by anecdotal reports that routinely described similar behaviour in the field.

As a component of our literature review, we identified subtle mentions of client organizations attempting to cope with such phenomena. The description of the Kensa Testing Group's typical activities in Japanese client organizations mentioned in [43] serves as a first example, and an empirical investigation of client manager responsibilities related to the test process published in [3] forms another example. In the former paper [43], the authors speculate that Kensa differs from the traditional concept of User Acceptance Testing because Kensa provides "thorough testing beyond just ensuring that the mutually agreed-upon requirements are met" [43]. The authors also place Kensa's practice into sharp contrast with the practice exercised by an U.S. client company explored in the same study. However, we have notably little information on the manner in which Kensa actually operates and lack further details of its routine management practices. In the second case, the authors demonstrate that the U.S. client company was not thoughtful in controlling the testing itself and that "it was relatively easy to gain the trust of the U.S. client team" [43].

In the latter case [3], the authors depict an interesting picture of U.S. client managers (in another U.S. client company) who maintain their day-to-day duties with offshore testing contractors. Again, our insight into an extant management mechanism is quite limited. Nevertheless, the authors confirm our finding when they agree that the literature "provides limited guidance on

CMs’ [client managers] project management responsibilities when projects are outsourced, whether vendor is located onshore or offshore” [3]. Interestingly, whereas Japan’s Kensa appeared to control contractors by emulating their testing work in an attempt to find blank spots and enforce correction (to *assure quality*), both U.S. client organizations worked with their offshore cousins in a type of *collegial mode*. For example, the U.S. client managers also report their struggles with internal organization issues that resulted in the contractor’s employees not being perceived and not benefitting in the same manner as members of their own U.S. team.

In terms of the level of involvement, Galin [47] explicitly perceives an external participation in contractor testing as one of the Software Quality Assurance (SQA) tools. He states that the “character of participation in the testing process is sufficiently comprehensive to enable the … [client’s] representative to *intervene,* if necessary, to *obtain assurance* of the quality demanded of the supplied software and the expected timetable for completion of the testing (and correction) process” (emphasis added) [47]. However, both of the abovementioned modes (i.e., Japanese and U.S.) appear to conform to this definition. Thus, the extant organization design and the formal and informal relationships between the client and contractor appear as variable but important factors. To sum up this dilemma, the issue of the necessity to balance *control versus trust* (i.e., control versus self-control) exists in the contracted test process, and there are likely multiple ways to do so.

For the guidelines and models of management actions, we reference two contributions focused on process models of test contracting. In the industry, these models are also known as concepts of *test factories* or *managed services* and roughly conform to the above-mentioned modes of operation. Sanz et al. [49] defines a test factory as an independent organization entity that provides testing services. However, this model lacks a clear distribution of responsibilities between the client and contractor in its current form. Lu and Käkölä [50] do not directly mention this term but similarly provide a life-cycle model for contracted software testing based on similar initial assumptions as those of Sanz et al. but with a clear responsibility boundary. Lu and Käkölä primarily present the contractor’s perspective but also propose active client engagement in the process.

To conclude the previous discussion, the current research only infrequently addresses contracted testing in general. To this end, our paper primarily covers the *inadequacies* of the test process performed by contractors. We are particularly concerned with the setting in which a contracted (i.e., custom-built) software system is developed within the frame of a fixed-price project. However, our current knowledge related to application of a control mechanism to address such issues is quite limited. In general, because high-level contractor management in the test management domain (together with a proven management framework) is not covered by current research, a gap exists in the research literature.

In the Introduction, we proposed a solution to address this issue, i.e., *Test Governance*. In the following section, we support our claim that the Test Governance Framework (TeGoF) is a management artefact that contributes to bridging such a gap.

3.3 Concept of Test Governance

Software testing can be divided into several levels with unique responsibilities. The test levels consist of component, integration, system and acceptance testing [9]. Bound to the traditional form of contractual relationships between clients and contractors, contracted software is often developed using sequential (waterfall-like) models, and contractors are often responsible for all aspects of a delivery except for acceptance [4]. Thus, this mode of contracted software development will typically involve testing solely by the client's employees at the end of the project, i.e., in the user acceptance testing phase. All other test levels are frequently perceived in industrial practice as the sole responsibility of the contractor, i.e., without a necessity for client involvement or the need for a continual inside view of the client [4]. This situation means that the mechanism of behavioural control is not frequently implemented [17].

As we have argued previously, the extant form of control mechanism over contractor testing is tightly connected to the perception of the role of Software Quality Assurance (SQA). Thus, we estimate that an SQA mechanism of good practice is either unclear or ignored in contractual relationships or, at least in the business IS domain, is generally considered as non-critical (in terms of potential threats to public safety and health). Among possible factors of influence distinct for business software systems that result in such a behaviour appear to be specific qualification of IS project managers in certain business organizations. It is possible that a subset of these client project managers are arguably aware of the SQA concept (see, e.g., [51]). However, adequate comparative research that probes into distinct differences between management of software development in software and business organizations is not available to date.

In sum, we argue together with the above-mentioned principles of behavioural control that the *blind trust* approach is (at least in a certain context) fundamentally flawed and such information asymmetry [52] between clients and contractors contributes to the high probability of project escalation [53]. The reason for this presumption is twofold. First, our presumption conforms to the theory that a testing organization entity should be separated from a programming organization entity [54], i.e., an independent authority should be involved. Second, data gathered in the initial (design) phase of our research show that the outcome-based mechanism represents one of the primary sources of problems that occur in the studied organization (see Section 5). In this paper, we present our results based on this presumption, i.e., in the specific context. Nevertheless, we are aware of a potential role of this context as well as the need to balance the costs versus benefits of control. Therefore, other scenarios should be considered by practitioners. For example, Black [55] suggests the application of a "due diligence" concept from the area of mergers and acquisitions to the evaluation of contractor quality on a one-time basis.

Based on our arguments and supported by practical implications of control theory research, we find that contractor management related to testing of contracted software in sequential models is vital. If this topic not handled otherwise (previous due diligence process, notably high level of trust, etc.), it occurs as the only method for assuring the quality of the contracted software system. Therefore, we decided to propose the Test Governance Framework (TeGoF) to address this issue. Using the term *governance,* we make a reference to the IT Governance concepts [56] that are well known in business organizations. In its broadest sense, IT governance could be understood as "patterns of authority" [57] in a client organization related to decision-making in the IT domain, and test governance can be understood in the same way.

4 Research Methods

This section provides details of the research design adopted in our study. First, we provide a definition of the research questions (Section 4.1) and briefly introduce key characteristics of design science and action research (Section 4.2). Furthermore, we clarify information on the research site and the dual role of the researcher (Section 4.3). Based on this information, we summarize the benefits and limitations of our research design (Section 4.4). In addition, further details of the research process are provided (Section 4.5). To conclude, techniques of data collection and analysis are explained in Sections 4.6 and 4.7, respectively.

4.1 Research Questions

The objective of this research is to investigate testing as a subject of control in contracted software development. The research also covers the design of a relevant management artefact. This artefact is intended to improve control behaviour and result in particular benefits for client organizations. Moreover, the results of this artefact evaluation are examined in an industrial setting, primarily from a social perspective.

The research questions are defined as follows:

- *How should client organizations manage or influence the test process conducted in contracted custom-built software development projects by respective contractors?*
- *What are the organizational determinants and consequences of introducing such a mechanism?*

This paper summarizes the results of a longitudinal two-year study conducted in the form of an action research project with design science principles. Broadly speaking, this research project was focused on software and test process improvement in which contracting of custom-built software systems development constituted the predominant acquisition model of the client.

4.2 Combining Design Science and Action Research

In this research project, we used two well-established research methodologies with practical implications, namely, design science and action research. First, we applied the principles of design science to develop a process framework. Next, we used the principles of action research to examine this framework in the industrial setting.

Design science is a research methodology particularly known in the field of Information Systems [58] and also partially used in Management and Organization Studies [59]. The aim of this methodology is construction of IT and management artefacts (with methodological rigor) that deliver clear benefits for the industry. The designed artefact can be generally evaluated using artificial evaluation (simulations and experiments) or naturalistic evaluation (case studies and action research) [60]. Due to its pragmatic concerns, this methodology is considered to have strong engineering roots [58].

Surprisingly, even in the IS discipline, probing into technological artefacts has a much stronger tradition than probing into IT management artefacts [31]. Whereas IS research concerning technological artefacts is gaining greater attention, IT management research in the form of design science is currently rather scarce [30]. Due to its strong focus on practitioners, software engineering appears to be more open to the design approach. Nevertheless, SE management research was previously stamped as “[p]robably the least accepted work in software engineering” [61]. Currently, relatively little management research has been conducted in this discipline [11]. Moreover, whereas design-oriented contributions (e.g., in the area of software processes or maturity models) are noticeable among SE publications, they are not frequently declared as “design science” explicitly [29].

Action research (AR) is a research methodology with strong roots in social science, especially the post-war work of Kurt Lewin. Action research can be viewed as “a highly unstructured field experiment on themselves together with others" [62] due to its nature, which is tied to interpretive philosophical presuppositions. In general, action research is oriented towards institutional and organizational change. In the field of computing disciplines, this approach has been used primarily in the IS discipline and Software Process Improvement areas. In SE, action research was formerly characterized as "closely related to case study" [63]. In their recent paper [64], Wohlin and Aurum claim that case study methodology, design science and action research constitute three basic research methodologies of Empirical Software Engineering, and we understand these methodologies in this sense in our paper.

4.3 Research Site and the Dual Role of the Researcher

This research was undertaken in a large organization listed in the CE TOP 500 (http://www.deloitte.com/cetop500) in the Czech Republic. To describe the company in additional detail, the FC (Financial Company, a pseudonym) is a business (i.e., non-IT) organization that provides financial services in the Czech market. The FC has a long and rich organization history. The research project originally emerged from real organizational issues and was motivated by the CIO's intent to raise the quality of software development outsourced to respective contractors.

In our case, the first author had the opportunity to occupy an insider position [65] in the researched client organization based on a particular type of contractual relationship. Thus, the first author represented both researcher and practitioner perspectives, and the perspective presented in this paper is that of the first author.[3] To frame this position, design science and action research were used as key research methodologies. Interestingly, one can compare this approach with Heiskanen's et al. [66] concept of a *reflective IS practitioner*. Our approach and Heiskanen's concept might be perceived as similar. However, two key differences are evident after taking a closer look. First, a clear intent to shape organizational reality by implementing a management and process improvement framework is quite obvious in our case. This fact is manifested by explicit adoption of a design science approach resulting in a creation of the TeGoF. Second, our position is strongly influenced by anthropological concepts, tools, and traditions and is thus built upon *ethnography* as the key source of inspiration in the social domain.

Although traditional ethnographies do not aspire to change the researched reality, culturally sensitive action research can be considered as an option for probing into organizational culture and organizational phenomena that are referred to by Kunda [67] as *backstage realities*; this term means access to people's "hidden everyday culturally derived realities" [68]. Although ethnographies conducted in (software) organizations are akin to the activities of anthropologists living with exotic tribes for a prolonged period of time to study the native habits and rituals [69], our action research approach can be metaphorically described as a form of applied anthropology used by missionaries for interventional purposes (see [70]).

Interestingly, our decision to follow this inductive form of AR practice was originally triggered by a key informant at the beginning of the research project. During the initial weeks at the organization, the informant stated in a private conversation: "It's all here about politics, … don't openly declare your intentions [regarding strict control of contractors]; you must *sense*

[3] To clarify the potential pre-understanding, we additionally note that the first author's original professional background is in software quality management of mission-critical embedded software development in a particular segment of technology. In this sense, the environment of the business organization provided him with many opportunities to compare.

the situation, people, and act *according it*" (client IS manager). Thus, the first author was made aware that the organization was apparently understood as a specific culture in which power and personal relationships play a central role in success or failure of the IT change initiatives (our aim was to put the TeGoF into action in all contracted software projects and improve software quality, but we lacked strong upper management support (see [71]).

Finally, because the first author was generally perceived as a member of the FC, it was necessary to assure the credibility of our research in this sense. Thus, the first author strove to develop a form of cognitive mechanism to cope with the pre-understanding [65] and view the reality as critically as possible. This vital goal was endorsed by the fact that the author initiated the research in parallel with entering the organization. Thus, his level of the FC's pre-understanding was rather low, and he shared many aspects with *real* ethnographers entering a *foreign* culture [72]. To further endorse the above stated aim, the author also maintained field notes on a regular basis [73].

To conclude, this research design enable clear benefits as well as negative consequences, which are summarized and further elaborated below.

4.4 Benefits and Limitations of the Research Design

First, one of the clear benefits of our study is the long-term immersion with the studied phenomena and a high level of control over the situation. As an insider in the community [74], the first author benefited from full access to the key informants with significant decision rights and certain social status. Thus, the key benefit lies in the fact that this research design offered flexibility and the possibility of broadly observing and influencing decisions. Unobtrusive ethnographers, who are typically outsiders at first, often encounter serious problems in attempting to become accepted by members of the researched organizations, especially if they aim to uncover backstage realities [35]. Second, our research design had practical implications for the FC. Broadly speaking, design science and action research methodologies are vehicles of first choice for researchers with pracademic (pragmatic-academic) orientations. The ability to bring benefits to the industry as well as academia and attempt to "bridge the practice-research gap" is a strong motivation for many pracademics [12].

However, our research design does imply negatives. First, the above-mentioned fact (i.e., long-term immersion connected with action participation) can also be perceived as a negative aspect because unobtrusive ethnographers are usually warned against the danger of *going native*, which is a situation in which the ethnographer becomes too deeply integrated into the community under study and thus loses distance and objectivity [75]. Due the nature of these projects, action researchers in culturally immersive AR projects are generally even more strongly exposed to issue of *going native* because they are convinced that the actions that they take are the appropriate ones and feel a particular level of commitment to offer the organization a sort of improvement.

Second, action research projects are often claimed as highly context dependent, which means that the reader must understand a particular context to capture the broader implications [62]. Third, it is quite challenging to make a strong theoretical contribution in an AR or DR setting due to the primary focus of both methodologies on practical implications. This aspect might be considered as a disadvantage of this research design because theoretical contribution is ranked highly among certain academic fields, e.g., IS and other academic disciplines in management sciences in general [76].

Fourth, research projects with similar goals, i.e., to explore (and in our case, partially change) a particular culture by introducing management artefacts, are typically quite demanding because they require the above-mentioned long-term immersion in the field. When conducted properly and in line with their anthropological roots, these projects can be highly time demanding and thus are rarely a preferred option for well-established scholars due to other academia-related pressures [73].

All of the above-mentioned issues are tightly bound to the validity and reliability of the research. For our research design rooted in anthropological tradition as well as interventional approaches (i.e., design science and action research), which are clearly non-detached from an explored reality in objective terms, it is not easy to directly address these matters. Validity and reliability are terms generally considered as a heritage of the positivist research tradition [75]. In the broadest sense, validity in design science and action research is tightly bound to pragmatic effects (i.e., benefits for practitioners) of the intervention process [58], which means that it is evidently necessary to "balance rigor and relevance" [31]. Moreover, a significant portion of our research project has the nature of an interpretive qualitative inquiry in which these terms also do not apply in their originally intended meaning [75]. Thus, we use the term *credibility* in this paper (as akin to validity).

4.5 Research Process

Our methodological approach can be further characterized as abductive because it is an overall combination of inductive and deductive logic [77]. First, this fact is demonstrated by the manner in which we combine design science and action research. Second, this fact is manifested by the method of analytic induction we used in the AR phase (see below). With respect to the combination of both methods, i.e., design science and action research, the research process consists of two phases: (1) design of the management artefact and (2) naturalistic evaluation of the management artefact.

The design phase, which is distinguished by the application of design science principles, was based primarily on a previous literature review and initial anecdotal organization rumours with such key themes as "Delivered software quality is poor" and "Contractors are not under control". For the logic, this phase was mainly deductive. A key component of the TeGoF was created in the form of a detailed "test methodology" document (20 pages in A4 format excl. appendixes) within three months after the first author had entered the organization. The form of the document was different from the one presented in this paper, but all key principles and activities were equivalent. In this phase, we were guided by the seminal IS work of Hevner et al. [58] and the seven guidelines suggested in their work. These guidelines have been recently proposed to SE researchers conducting research on maturity models [29]. Nevertheless, it is important to understand that Hevner's et al. work largely leans towards IT (technological) artefacts, and its direct applicability in the management domain is therefore limited [30,31].

In compliance with the commonly accepted design science canon, it is necessary to perform an evaluation of the proposed artefacts. In essence, several approaches are available for conducting a TeGoF evaluation. Hevner et al. [58] suggest Observational, Analytical, Experimental, Black-box and White-box Testing (!), and Descriptive methods as different means of evaluation. Iivari and Venable [60] fundamentally distinguish between artificial (simulations and experiments) and naturalistic (case studies and action research) evaluations. Unfortunately, as previously noted, little consensus exists on the proper evaluation of management artefacts at present [30,78]; especially because management artefacts fundamentally interact with the social world of organizational actors, it is not easy to address the precise quantitative criteria that conform to the engineering view of design that currently prevails in the computing disciplines[4] (see [31,79]).

In the case of the TeGoF, one might argue that similar *hard* quantitative evaluation is needed. However, it is not feasible in a real industrial setting (at least in the FC) to conduct a comprehensive analysis that differentiates between defects captured by the TeGoF and defects captured without the contribution of the TeGoF due to the factual impossibility of finding two comparable software development projects. In principle, a carefully designed artificial experiment might constitute an evaluation of certain components of the TeGoF mechanism (e.g., contractor's creation and client's review of the test plan), but it is rather difficult to imagine such an experiment with the TeGoF as a whole in its full scope.

In our research, we did not follow the path of evaluating the TeGoF in a directly measurable manner. Instead, we turned our attention to the rich spectrum of research methods in the social science domain and focused on subjective interpretation of TeGoF contribution as perceived by the respective organizational actors. Thus, the design phase was immediately followed by the *phase of naturalistic evaluation of the management artefact* using non-positivist philosophical presuppositions. In contrast to the design research phase, the AR took place over a prolonged period of time (1 year and 9 months). In this phase, the TeGoF was dispersed (instantiated) among multiple software projects, and its effects were carefully examined from the *human-side*. Based on the reason described, we selected a specific inductive form of action research with a strong qualitative research orientation and theory-generating approach [68]. Consequently, we focused on "thick descriptions of the patterns of subjective meanings that organizational actors use to make sense of their worlds, rather than entailing the testing of hypotheses deduced from a priory theory" [68].

Thus, our aim in the AR phase was not to seek a sort of "universal" or "objective" truth in the positivist sense. In contrast, we took an interpretive stance [64,80] to understand the world of the organization's actors and their interactions with the TeGoF. Because our primary aim in this paper is to introduce the TeGoF and its evaluation, our ambition of introducing "thick descriptions" and wider theoretical implications is therefore quite limited due to space and conciseness constraints, and the focus of this paper is more pragmatic.

4.6 Data Collection

In the phase of naturalistic evaluation, the first author conducted initial rounds of semi-structured interviews with each client project manager at the time when the TeGoF was originally put into action, and the project manager's cooperation was necessary. However, a key portion of data (due to the nature of the project) stems from repeated interviewing and day-to-day interactions with project managers and other organizational members during TeGoF implementation. The interviews were not tape recorded due to the sensitive nature of certain conversations and our primary interest in real reasons and backstage realities, which are often quite different from the "official" versions of the story [80]. Most of the interviews were transcribed immediately after the interview was finished. All interviews were carried out in Czech/Slovak, and therefore, all quotes presented in this paper are translated from Czech/Slovak to English. The central topic of the interviews was the following: "What are the

[4] This debate does not actually represent a new discussion topic. We consider the history of management thought by contrasting *Management Science* and *Operations Research* with *Organizational Behaviour* and *Organization Theory*. Although the first two stem from engineering/nature science orientation, the last two benefit from the research tradition of psychology and sociology, respectively [106].

reasons for success/failure of the implemented management mechanism [i.e., TeGoF] from your perspective?" and "How useful are principles implemented [by TeGoF] for you?"

One round of unstructured interviews was also conducted with each contractor project manager in introducing the TeGoF principles to the three contractors. Numerous additional short and repeated interviews with different organizational actors occurred in the framework of participant observation. For example, participant observation resulting in field notes allowed gathering of insights from informal conversations (among other outcomes) with upper managers from contractor organizations as well as client organization (FC). The upper managers held roles as key account managers or managing directors (all three contractors) and the CIO (FC).

Although the data collection method might potentially appear biased in terms of positivist science, it is not extraordinary in terms of ethnographic practice [75]. For example, in his seminal and widely cited book on field methods, van Maanen [72] describes the following excerpt as an example of a field practice in a *realist tale* type of ethnography, including ethnographically informed research in organizations. This quotation was excerpted from his paper on control in a police organization:

> "Again, my methods of study were largely those of the cultural anthropologist, emphasizing direct, sustained participant observation and the repeated interviewing of key informants. Most of the data reported here stem from informal interaction with members of the police world as they attended to their ordinary work activities. Since I rarely used a tape recorder, the conversational data are only as accurate as memory and ear allow." [72]

To increase credibility, we also used data triangulation. More specifically, we benefited from access to a rich document collection. The key documents within the research project included test and project documentation prepared by the contractors (test plans, test scripts, project charts, defect reports, status reports, meeting minutes, project management methodology, historical versions of the test methodology, and others). Because these documents rarely describe the *backstage realities*, we were interested in the most frequent but rather depict the *officially acceptable version of reality*[5], we also carefully focused on comparison of participants' behaviour in different situations (formal/informal, different audiences etc.) as they evolved over time. To further increase credibility, we investigated three different software projects with distinct characteristics in our research.

4.7 Data Analysis

In qualitative inquiries, data collection and data analysis are often inseparable from one another [75,81]. This situation is particularly manifested by the method of analytic induction [81] that we used in the AR phase of our research. Analytic induction is a research technique that was originally closely associated with the work of Znaniecki [82] and Lindesmith's later study of opiate addiction [83]. Broadly speaking, analytic induction is based on intentional search for non-conforming evidence and negative cases. The application of this method is thus guided by collected data and typically takes advantage of the constant comparative method developed by Glasser and Strauss in the 1960s as a component of the grounded theory research repertoire [84]. Analytic induction is abductive in essence, i.e., it combines a theory-generation approach with a theory-testing approach. van Maanen et al. [85] claim that "abduction begins with an unmet expectation and works backward to invent a plausible world or a theory that

[5] For example, a Test Plan document may contain a detailed description of the good-practice, but a real test execution may vary greatly.

would make the surprise meaningful". As we demonstrate further in the case description, we encountered several "unmet expectations" related to the TeGoF.

In our research, we used analytic induction in the following manner. All three cases described were objects of examination and fully or partially overlapped in the sense of their beginning and end. In the field, the first author was constantly evolving theories related to these three cases, and these theories were consequently accepted or rejected, which means that the first author constantly iterated between data and theory-in-use until plausible explanations were discovered. These explanations became a component of the field notes in the form of analytic memos [73] and brief summaries. Later, data were systematically re-examined and open coded. Categories and central themes were identified, again using a constant comparative method.

After introducing both phases of the research process in this section, we further elaborate on the design phase by focusing on the starting points and later development of the framework that resulted in the construction of the TeGoF.

5 Framework Design Process and Structure

This section provides details on the TeGoF origin, development and core concepts. Furthermore, the proposed role of a client Test/Quality manager is examined. Finally, the TeGoF structure itself is depicted.

5.1 Initial Challenges

Table 1 summarizes the initial challenges that became key reasons to advocate for the TeGoF construction. The data were gathered within the phase of design research and summarized into the following points. We present these points as concrete findings in the form of propositions suggesting their generalizability but stress that the main source of data was represented by the FC's practice, which means that these propositions are possibly context dependent and generally should be treated as a subject of possible future validation in other contexts.

Table 1
Initial key challenges that motivate framework design

	Key issue	Description
Challenge no. 1.	Limited dedicated resources	Certain business organizations cannot simply afford powerful and dedicated testing teams because of their tight IT budgets. Such organizations partially or fully rely on their contractors in terms of software quality concerns, or alternatively, they must involve core-business experts (hereafter "domain testers") in the software test process during user acceptance testing.
Challenge no. 2	Domain testers' allocation conflicts	Involving domain testers is usually accompanied by multiple problems, e.g., inability to allocate them fulltime because of their regular tasks, which often results in jeopardizing the project schedule when commitments and staffing are improperly managed or result in incomplete user acceptance testing.

Challenge no. 3	Domain testers' limited expertise	Domain testers are not well trained as testing experts; their knowledge of testing is strictly based on (and is restricted to) domain expertise, and for such reasons, they can easily omit important components of functional testing. Their involvement also does not address the need for special testing (e.g., performance testing, maintainability testing etc.).
Challenge no. 4	Missing control mechanism	Rigorous and extensive software testing at the end of software system development lifecycle is inefficient for both involved parties (client and contractor) because it lacks efficient control aimed towards project finalization on time. Therefore, testing must be handled properly by the contractor during the system-testing phase together with sufficient client involvement and a necessary level of supervision from the client side.
Challenge no. 5	Agency problem	Omitting the contractor governance principles from the client side can lead to unpredictable results and *quality catastrophe* scenarios in the user acceptance phase of contracted software system development projects when these projects go awry; certain contractors appear to prefer short-term gains and tend to partially or fully get around client rules and standards, which might lead to many shortcomings when *nobody is watching*. This situation can be understood as an adequate reaction to the pressure associated with the tendency to minimize prices in fixed-price deliveries but definitely should not be used as a general excuse.
Challenge no. 6	Unique profile of client project managers	Client project managers in business organizations often possess more general skills and competencies than their colleagues in software businesses who come from a technical background. Their (*non-software* project manager) knowledge of software development and quality management is definitely far less reliable. The tendency to fully rely only on information gained from the contractor (contractor's self-control) appears to be fatal, especially in the case in which a contractor project manager chooses to *keep silent* about poor project progress.

5.2 Sources of Testing Debt in Contracted Development

In Section 3.2, we presented a conceptual background related to the consequences of inadequacies in the testing provided by contractors. In this section, we aim to draw a link between such issues observed in the FC and the TeGoF because these issues have become a key driver for TeGoF construction. We use the term *testing debt* as a conceptual metaphor to present our findings in a more illustrative manner.

Testing debt can be generally characterized as a metaphorical expression of inconsistency between planned or rationally expected test activities and the test activities actually performed [86,87]. In the FC, sources of testing debt were detected with varying intensity in one or more examined software projects. The issues that create testing debt specifically included: (1) omission of a proper test planning phase and stakeholder involvement, causing a number of serious flaws in consecutive test execution process; (2) an incorrect or inadequate test basis (requirements) analysis or test coverage during the test design process; (3) incorrect or inadequate test execution strategy in the context of a concrete IS development project, including (a) an inadequate number of test runs that does not respect the development dynamics, (b) ignorance of release management principles during testing that causes test result invalidity, (c) lack of regression testing, (d) defect closure without re-testing, (e) underestimation of the severity of identified defects, and (f) overall marginalization of the potential impact of less severe defects; (4) intentional affirmation of inadequately tested software handed over to user acceptance testing with the goal of transferring responsibility and costs of testing to the customer and/or avoidance of contractual penalties associated with late delivery.

At the time of the TeGoF design, we believed that all of the issues presented above that contribute to testing debt would be repaid later, usually during user acceptance testing performed by the client organization. From this perspective, the proposed framework should contribute to an elimination of testing debt by introducing a control mechanism. Thus, we believed that client organizations must actively control the sources of testing debt to eliminate the necessity of later repayment of the debt from the client's own funds. The relationship between the testing debt sources relative to the proposed framework (TeGoF processes) is further portrayed in Table 4.

5.3 Guiding Principles of the TeGoF design

Based on the above-mentioned findings, the TeGoF design is grounded in six key design principles:

Principle 1: Reduction of information asymmetry; early client involvement
Principle 2: Reduction of testing debt
Principle 3: Regular quality management reviews that increase client insight into the test process
Principle 4: Formal technical reviews that serve as quality gates of the project
Principle 5: Alignment with *human factors* of a client organization
Principle 6: Assignment of a competent client test/quality manager

The first principle describes an existing demand for reduction of information asymmetry, i.e., evokes the need for the full scope of relevant information as well as full and early visibility from the client perspective. Together with the second principle, this point evokes conformity to the consequential principles (Principle 3, 4, 5, 6). The third principle covers management reviews that consist of contractor-client test/quality status meetings on a weekly basis. The following key actors are invited to participate in these meetings: project managers on both sides, the contractor test manager, the client Test/Quality manager, and in later phases, the client business representatives. A regular set of measurement evaluations based on client demands should be presented during these meetings.

The fourth principle underpins the importance of formal technical reviews supported by the contractor and aimed at software testing artefacts. These technical reviews should be considered as crucial to the entire process of test governance during the contracted software development project lifecycle because they enable empowerment of the client and increase decision-making power. Formal technical reviews include but are not limited to review of test planning documentation and test design documentation (test conditions, test scripts etc.) as well as output artefacts (test logs, test reports) used as a foundation for client evaluation of contractor test process completion and client user acceptance testing start. The contractor is also expected to internally implement a peer review process, and outputs of internal peer reviews should be checked on the request of the client.

The fifth principle covers the conceptual connection with a notation of management artefacts that are subjects of social interactions and interpretations of social reality by key project stakeholders, as shown in Section 2.5. Thus, the aim of the TeGoF is not to prescribe solutions to the issues belonging to the social domain but rather to underpin the need for other appropriate management actions where necessary.

With respect to the sixth principle, this paper promotes the role of an independent Test/Quality manager from the client organization acting (apart from other aspects) as a client project manager mentor in the field of software testing and quality assurance (see below).

Additionally, formal support for the proposed principles is necessary to ensure proper test governance. Typically, this process will be embedded in a form of contract obligations that define general rules in the area of software development and testing and possibly reference to binding and detailed regulations issued by the client.

5.4 Role of the Client Test/Quality Manager and Responsibilities

One of the interesting points in this research project is the role of the Test/Quality manager. We strongly believe that this position and its influence are vital to TeGoF success. In the following, we briefly sketch two possible models of the client Test/Quality manager’s (hereafter referred to as **“client T/Q manager”**) participation, a basic set of responsibilities, and key elements of his/her qualification.

From a project management point of view, we generally suggest two modes of operation of client T/Q managers; they can either (1) participate in project management as a client project co-manager (*the guiding role*) or (2) act in a more formal position, i.e., an SQA auditor (*the enforcing role*) [cf. 38]. In the FC, option no. 1 was adopted among the described cases. In any case, the role of the client T/Q manager during an IS development project greatly differs from the standard test management goals and activities during software development projects. Client T/Q managers are much more closely tied to quality objectives, high-level oversight of quality, and overall evaluation as well as participation in software system development project coordination (on the client side). In contrast, contractor test managers are engaged in test planning, evaluation, and immediate test team management.

At the same time, the presented approach does not preclude the possibility of making a client T/Q manager responsible for user acceptance test (UAT) management and coordination in parallel. In this case, the client T/Q manager would occupy the *standard* role in the process of UAT management and clearly belong to an IS or business unit. Alternatively, and related to a close cooperation with multiple project managers, a T/Q manager may benefit from an independent standing inside the Project Management Office in a guiding role. Based on our empirical findings (see related case descriptions), we strongly believe that a client T/Q manager that observes multiple IS development projects and governs multiple contractors’ performance (together with an individual client project manager) can contribute to solving the whistler-blower dilemma [38] of these managers and, at the same time, help to reduce information asymmetry between the client and respective contractors. Thus, we prefer the vision of an independent standing for this role (analogy of an independent Quality Assurance function; see, e.g., [88]) and suggest a lower involvement in the operational management of UAT testing.

With respect to the qualification of T/Q manager, we consider the following set of soft and hard skills as crucial. Basic soft skills should include customer advocacy attitude, leadership skills (quality leadership across client and contractors organizations), emotional intelligence (ability to articulate bad news in an appropriate manner), and structured thinking. Basic hard skills should include knowledge of testing and quality assurance related concepts (ISTQB Advanced Level qualification may be a reasonable demand) and knowledge of project management concepts.

In addressing the above-mentioned soft skills, client T/Q managers must be able to articulate and discuss their vision of quality culture with the contractors because a shared vision in this area must be perceived as an important trust-building factor. Moreover, in the case in which a new contractor is entering the client environment, a long-term insider view of this environment is typically missing. Consequently, a lack of contractor effort devoted to building a quality culture can later reduce trust between both parties. Finally, both the client T/Q manager and the project manager should be aware of the possible influence of cognitive and psychosocial factors. This influence should not be underestimated because information asymmetry reduction and proper detection of symptoms of the mum effect play a crucial role in successful implementation of the proposed mechanism.

5.5 TeGoF Structure

This section describes key elements of the proposed framework. To build this framework, we considered the structure of a standard test process as defined by the International Software Testing Qualification Board (ISTQB) and mapped it to the PMBOK structure (as described in Section 2). However, we are only concerned with the contractor's system testing phase, and thus, we consider issues of client user acceptance testing to lie beyond the scope of the TeGoF.

We treat test governance tasks as a separate knowledge area in the sense of the PMBOK concept, which means that we divided the respective processes into five process groups, i.e., Initiating, Planning, Executing, Monitoring and Controlling, and Closing. This concept is portrayed in Table 2 and Table 3. The first row represents the organization-wide component and illustrates that the approach should not be isolated in terms of single projects but aligned and coordinated across the entire organization. The columns indicate a relationships among phases of the test process (ISTQB) and process groups (PMBOK). The last row provides a summary of the designed processes. Each process is further described in a structured form following a brief description of the individual process groups:

1. *Initiating Process Group* – The processes in this group generally aim to define a new project. With respect to test governance, the key process provides contractual support for the proposed principles because different views on the contractor's obligations in testing might exist.
2. *Planning Process Group* – Project planning represents a vital phase that influences project success. From a test governance perspective, it is crucial to review and acknowledge a Test Plan document. The Test Plan document is prepared by the contractor based on a common methodological grounding provided by the client, e.g., Test Policy and/or Test Strategy.

3. *Executing Process Group* – This group provides a guideline for a concrete task to meet the project goals. The test scripts prepared by contractors and their form and content represent one of the key success factors and quality determinants in test governance. In this sense, an in-depth review of test scripts is necessary to achieve alignment between both sides. The same logic applies to test data as well as to test artefacts that are produced. Although stressed, involvement of users can be considered as a redundant and self-evident task. Usually, as previously noted in the discussion about the traditional contract model, users are often involved as late as at the end of the entire development process, i.e., during user acceptance testing and this might result in project failure.
4. *Monitoring and Controlling Process Group* – This group of processes should be considered as a vital component of the entire framework because monitoring and control (in the narrow sense) allow the client to overcome difficulties with the black-box character of the developed system. For example, introducing various control meetings (e.g., including additional milestones in the project) aids in achieving the control goals. These processes help to transform an outcome-based control into behavioural one.
5. *Closing Process Group* – At this stage, the framework role is primarily supportive and provides feedback information on previous test activities; its inclusion is important for a future cooperation between the subjects.

The fields indicated in the template are defined as follows:

ID – Unique identification of the process as a component of the TeGoF
Main goal – Short annotation of the process
Further details – Specification of key process goals and further guidance
Responsible roles – Roles responsible and accountable for the process execution on the client side
Inputs – Work products required to perform the process
Outputs – Work products generated by the process

Mapping of testing debt sources to the Test Governance Framework processes is provided in Table 4.

Table 2
Core structure of the Governance Framework and mapping of the PMBOK and ISTQB components to the framework

Organization-wide Test Governance component

G.1 Establish coherent Test Policy and Test Strategy

Underlying principle: Govern, lead and improve maturity of test processes across organisation and contracted software projects

Project level component of Test Governance (focused on contracted software development)

Process group view (PMBOK)	**Initiating**	**Planning**	**Executing**		**Monitoring and Controlling**	**Closing Process Group**
Test process view (ISTQB)		**Test Planning**	**Test Analysis & Design**	**Test Implementation & Execution**	**Reporting and Evaluating Exit Criteria**	**Test Closure**
Process name	1.1 Establish contractual adherence	2.1 Review test plan and planned test effort	3.1 Review test scripts	4.1 Review test data 4.2 Review output test artefacts 4.3 Involve users	5.1 Organize test meetings and review test progress 5.2 Control regression testing strategy 5.3 Prioritize persisting defects 5.4 Organize “End of System Testing” meeting	6.1 Support final GO/NOGO decision 6.2 Review “Lessons Learned” document

Table 3

Description of the Test Governance Framework's processes

Organization-wide – 1 process

ID	**G.1 Establish coherent Test Policy and Test Strategy**
Main goal	Prepare and put into action the Test Policy and Test Strategy documents
Further details	These documents contain general guiding principles as well as binding rules. Other naming may be used when appropriate, e.g. Test Methodology. The adoption process may include interrogations with business and other stakeholders and a usage of influence tactic. Multiple versions of the Test Strategy may be created when appropriate (e.g. for agile and traditional development respectively).
Responsible roles	Client T/Q manager, Client Quality Assurance manager (when applicable)
Inputs	Formal and informal stakeholders' expectations Organization Quality Policy (when applicable)
Outputs	Test Policy Test Strategy

Initiating Process Group – 1 process

ID	**1.1 Establish contractual adherence**
Main goal	Provide a contractual support which allows a future enforcement of test governance principles
Further details	Adherence to client's Test Policy and Test Strategy is to be stated, entry and exit criteria for System Testing and User Acceptance Testing (UAT) is to be defined altogether with the process of UAT suspension and redemption procedure.
Responsible roles	Client project manager, Client T/Q manager
Inputs	Contract templates Contract proposal Test Policy and Test Strategy
Outputs	Binding contract

Planning Process Group – 1 process

ID	**2.1 Review Test Plan and planned test effort**
Main goal	Organize a series of project activities in order to reach a mutual agreement as well as responsibility determination within phases of System Testing and User Acceptance Testing
Further details	A conceptual Test Plan is presented by a contractor during a formal meeting. Stakeholders from the client's side are invited to share their suggestions or requirements. The content of the Test Plan document should be based on good practice in this area (e.g. IEEE 829, TMMi). Key parts of the Test Plan document typically include scope, test approach, risks, resources, test schedule, test suspension/redemption criteria, regression strategy etc. It is recommended to also include a section containing planned test effort needed for individual functionality, at least in a high-level form and mutually acknowledged. The structure and content of a "typical test case" should be defined and mutually agreed. The location of defect reports should be agreed. The client should either get an access to a contractor's defect tracking system, or preferably provide his own defect tracking system to allow for possible screening.
Responsible roles	Client project manager, Client T/Q manager
Inputs	Test Plan proposal Test Policy and Test Strategy
Outputs	Approved Test Plan stressing out possible differences in comparison with the Test Policy and Test Strategy

Executing Process Group – 4 processes

ID	**3.1 Review test scripts**
Main goal	Execute a formal review activity in order to gain an adequate level of confidence about the content of future System Testing performed by a contractor
Further details	The review depth depends on many factors, i.e. mutual trust, history and results of previous cooperation between the client and contractor. The expected cost of the control process should be evaluated against potential harm caused by an improper test analysis or inadequate test case design.
Responsible roles	Client T/Q manager, Client test or system analyst
Inputs	Test scripts Test basis/Requirements Architecture/Design Example of "typical test case" (2.1)

Outputs	Reviewed and improved test scripts

ID	**4.1 Review test data**
Main goal	Assure System Testing phase to be provided with valid test data
Further details	The client usually provides or specifies test data. If not, quality of test data provided by the contractor must be verified.
Responsible roles	Client T/Q manager, Client test or system analyst
Inputs	Test data Test basis/Requirements
Outputs	Reviewed and improved test data

ID	**4.2 Review output test artifacts**
Main goal	Assure that the quality of test process output artifacts is adequate and complies to prior agreements
Further details	As a minimum, quality of defect reports, and test execution logs should be checked in a test management tool/information system. If appropriate, these checks can be conducted in accordance with random patterns.
Responsible roles	Client T/Q manager, Client test or system analyst
Inputs	Test Plan Data records in a test management tool/information system
Outputs	Typically, no specific work item (adjustments and test process corrections – when applicable)

ID	**4.3 Involve users**
Main goal	Allow participation of end-users in the test process
Further details	The character of participation may be either formal, or informal. However, even in traditional development methodologies, users should be involved in testing as soon as possible. That means, at least during the second half of the System Testing phase. The main effects include a lower user resistance as well as a lower number of major issues identified during User Acceptance Testing.

	Adherence to client’s Test Policy and Test Strategy is to be stated, entry and exit criteria for System Testing and User Acceptance Testing is to be defined altogether with the process of UAT suspension and redemption procedure. (UAT is not further covered by the TeGoF.)
Responsible roles	Client project manager, Client T/Q manager
Inputs	Test Plan
Outputs	Formal or informal plan of user involvement in the test process

Monitoring and Controlling Process Group – 4 processes

ID	**5.1 Organize test meetings and review test progress**
Main goal	Organize periodical management sessions to keep status of the test process up to date
Further details	Upon the mutual agreement, periodical meetings focused on current progress in the test process should be held. The contractor should give a presentation regarding current status of test progress. Typically, a set of pre-defined metrics will be presented (e.g. number of defects found, number of removed defects, number of persisting defects etc.).
Responsible roles	Client project manager, Client T/Q manager
Inputs	Data records in a test management tool/information system Set of pre-defined metrics
Outputs	Test metrics representation

ID	**5.2 Control regression testing strategy**
Main goal	Keep control over the volume of changes implemented during the test process
Further details	When traditional development methodologies are followed, test cycle should basically serve as a frame guaranteeing a stable software version. From the viewpoint of test result validity, a consistent test execution strategy needs to be agreed and carried out by the contractor. Unplanned and spontaneous deployment of new versions should be suppressed. Version control tools may be used to monitor the scope of source code changes in final test rounds. A mutual agreement on concrete principles and conclusions should be reached (see also 2.1).
Responsible roles	Client T/Q manager, Client test or system analyst

Inputs	Version control tool reports
Outputs	Metrics representation

ID	**5.3 Prioritize persisting defects**
Main goal	Keep control over the priorities of persisting defects
Further details	Under normal circumstances, not all defects can be solved until a software system hand-over. Based on that, it is necessary to arrange a mechanism with the contractor that allows determining which defects are critical in terms of their priority. This decision can be based e.g. on defect severity, or on business priorities. Therefore, a defect prioritization meeting should be organized and persisting defects reviewed one by one, or by other appropriate criteria.
Responsible roles	Client project manager, Client T/Q manager, Client test or system analyst
Inputs	Data records in a test management tool/information system Set of pre-defined metrics
Outputs	Prioritized defects

ID	**5.4 Organize "End of System Testing" meeting**
Main goal	Keep control over the software system quality before the hand-over to acceptance testing
Further details	End of System Testing (followed by the start of User Acceptance Testing) is a crucial moment in a contractual relationship between a contractor and client. At this point, the main responsibility for system testing is transferred to the client. Thus, from that moment on the contractor expects that the client will prove the system contains defects not vice-versa. Consequently, prior to accepting this role by the client, the contractor must demonstrate that the developed software system provides an adequate level of confidence in terms of software quality. This level of confidence can be demonstrated e.g. by examining trends of new defects found over time, random checks, code review etc. The main aim is to save client's resources and costs.
Responsible roles	Client project manager, Client T/Q manager
Inputs	Data records in a test management tool/information system Set of pre-defined metrics

Outputs	Acknowledged End of System Testing protocol

Closing Process Group – 2 processes

ID	**6.1 Support final GO/NOGO decision**
Main goal	Provide a support for the decision making process related to the acceptation of a developed software
Further details	As a part of the final set of activities and project closure, client stakeholders must decide whether the developed system is ready for a production use. This decision should be evidence-based. Test/Quality manager guidance and expertise will likely be demanded to check data records in quality management software and/or other form of evidence. Also an expert judgment regarding persisting defects may be requested by the client project manager or stakeholder.
Responsible roles	Client T/Q manager
Inputs	Data records in a test management tool/information system Metrics representation
Outputs	Signed acceptance or GO/NOGO protocol

ID	**6.2 Review "Lessons Learned" document**
Main goal	Initiate a continuous improvement in the area of testing in cooperation with the contractor
Further details	Lessons Learned document focused on testing and quality issues should be prepared by the contractor and reviewed by the client. A series of discussions should lead to concrete steps allowing future improvement in mutual cooperation and quality of the test process. Note: Lessons Learned can also be considered as a part of standard project management activities.
Responsible roles	Client project manager, Client T/Q manager
Inputs	Data records in a test management tool/information system

	Metrics representation
Outputs	Lessons learned regarding mutual cooperation between the contractor and the client Internal lessons learned on the client's side

Table 4

Mapping of testing debt sources to the Test Governance Framework's processes

	Source of testing debt	Corrective action
1.	**Omission of a proper test planning phase and stakeholder involvement**	2.1, 4.3
2.	**Incorrect or inadequate test basis analysis or test coverage**	2.1, 3.1
3.	**Incorrect or inadequate test execution strategy**	5.1
3a.	* inadequate number of test runs	2.1, 4.2
3b.	* ignorance of release management principles	5.2
3c.	* lack of regression testing	5.2
3d.	* defect closure without re-testing	4.2
3e.	* underestimation of severity of identified defects	5.3, 4.2
3f.	* overall marginalization of potential impact of less severe defects	5.3
4.	I**ntentional affirmation of an inadequately tested software handed over to user acceptance testing**	4.2, 4.3, 5.4

6 Framework Evaluation

To evaluate our framework, we applied a naturalistic (observational) type of evaluation in the form of a qualitative interpretive field study that involved three different projects, as explained in Section 4 at full length. The key characteristics are presented below (Table 5).

Table 5
Key characteristics of software systems projects involved in the framework evaluation

	Case 1	**Case 2**	**Case 3**
Project manager	Member of business unit	Member of Project Management Office	Member of Project Management Office
T/Q manager	Member of IS department	Member of IS department	Member of IS department
Contractor	Contractor 1	Contractor 2	Contractor 3
Aim	New system development	New system development	Maintenance, feature development
Type of SS (custom/COTS)	Custom software system	Custom software system	Highly customized off-the-shelf software for financial enterprises
Characteristic	Special-use ERP	Financial services – contracting (customer-faced)	Financial services - support processes (back office)
Architecture	Client-server Web-based interface	Client-server Thick client	Client-server Thick client & multiple web-based interfaces
Contractual model	Fixed price	Fixed price	Fixed priced per new feature
Project scope estimation	~700 MD	~1000 MD	~500 MD
Organization history	None	0.5 year	10 years

6.1 Case 1

6.1.1 Specific Context

Case 1 represents the first project that allowed evaluation of the entire framework within the FC. This project could be considered as a typical custom-built software project aimed at development of a new software system to support a special-use functionality focused on the FC's interactions with external business partners. Thus, the application was built from scratch with relatively low demands on application integration with other software systems in the FC's application portfolio. Interestingly, business unit members participating in this project were highly enthusiastic from the beginning because this was the first software system in the FC that responded to their unique needs (previously, they had fulfilled their working duties either manually or with a use of off-the-shelf personal office automation applications perceived as inadequate for the task purpose).

6.1.2 Roles Involved in Test Governance[6]

The TeGoF served as a guideline for cooperation among the client project manager, client T/Q manager, contractor project manager and contractor test manager. Particularly on the client side, a participative atmosphere existed between the project manager and T/Q manager. The client project manager appreciated the proposed control mechanism because he had little experience with project management in the field of software systems development (although he had rich IT project management experience in general). Interestingly, this client project manager was not a member of a corporate Project Management Office but rather a representative of the business unit responsible for the project from business point of view. However, the concept of two project managers, i.e., business project manager and IT project manager, was not routinely used in this organization. In this case, the IT component was formally coordinated by an IT coordinator in a process role assigned to a senior IT analyst.

In this project, as in the other two projects, the specific role of a business representative was also involved. In short, this role was the main contact point on the business side of the FC for software system development (e.g., related to requirements elicitation). This role (at the B-3 level, a senior management role in the FC) did not participate in the day-to-day duties related to software development but played a central function in management reviews and passage of important project milestones. Although the business representative was not formally considered as the project sponsor (a different role, typically at the B-1 level in the FC), his formal and informal decision-making power was quite strong.

In the end, the business representative took full advantage of the TeGoF results and became quite strict on the start of User Acceptance Testing of the developed software system. Under normal circumstances (i.e., without the TeGoF), the contractor would simply hand over the software system, without the possibility of the client performing pre-screening, and User Acceptance Testing would start. In contrast, the TeGoF provided a strong mechanism that put a significant pressure on the contractor because new defects were still in the process of identification by contractor's testers and by several of the client's employees selected to perform prototype testing.

Interestingly, the TeGoF was not originally designed for use by a business representative (but rather a client T/Q manager and client project manager). However, due to the implemented behavioural control mechanism and the business representative's previous experience with software development, he was aware of the existing situation and could benefit from the TeGoF as well. The business representative appeared to follow "unrealistic expectations" (contractor project manager) because he was willing to start User Acceptance Testing only when no persisting defects of Critical, Major or Minor severities remained. (Previously, the FC had dictated that User Acceptance Testing would begin if no critical defects and only "some" occurred, meaning a relatively low amount of major and minor defects. In the majority of past cases, the need to keep the project inside its timeframe was perceived to be stronger than the formal criteria. After the TeGoF introduction, this habit changed, and the process was also formalized in quantitative terms as a component of test planning).

[6] To protect identity of our informants and other organizational actors, we further reference all of the FC's employees as "he" irrespective of the real situation. In this text, we do not intentionally provide further details on the roles and responsibilities of client IS managers who did not occupy the roles described in the TeGoF design. Similarly, we do not differentiate among concrete client IS managers in presenting direct quotations. Such information could lead to a disclosure of our informants' identity. We also altered the emic term originally used in the FC for referring to the role of a business representative.

6.1.3 Contractor Reaction to TeGoF

Although the contractor had significant experience in previous cooperation with the FC, this was his first project in which the TeGoF principles were introduced and applied. Thus, these principles were entirely novel for this contractor. Initially, the contractor expressed concerns related to "changing the rules in the middle of the game" (contractor project manager) because the formal acquisition process had already been completed at that time. However, binding agreements included in the contract were too general to decide whether the TeGoF principles could be enforced by the FC. Nonetheless, the contractor project manager accepted the TeGoF principles after participating in two initial meetings.

At this point, the TeGoF contribution looked quite promising. Initially, a serious project issue emerged during a review of the Test Plan document (Process 2.1 – Review test plan and planned test effort). In fact, performance testing was not included in the test plan because it was not planned or even considered by the contractor. After a series of mutual interrogations, this type of testing was supplemented in the project plan without an impact on the contract price. As the project progressed, the mechanism captured in the TeGoF (Process 5.1 – Organize test meeting and review test progress) further assisted in early detection of problems related to a low volume of resources allocated to known defect resolution. Thus, the project indicators (e.g., number of defects found versus removed) revealed a seriously jeopardized project deadline. Following several mutual interrogations and negotiations, the contractor boosted the manpower dedicated to the software system development on explicit request from the client project manager and T/Q manager.

Overall, the TeGoF contributed to an identification of the above-mentioned inadequacies in the software system development process. However, the TeGoF also became a socio-technical tool that allowed non-IT client roles to enter into and influence the control process as described above. In this situation, the contractor repeatedly appealed to the client project manager and T/Q manager to "help to finalize the project [and prevent another interventions of the business representative]" (contractor project manager). This situation represents a reversal of control tactics grounded in clan control and self-control mechanism. We further discuss this issue in the following section.

6.1.4 Discussion of the Case and Lessons Learned

In this case, we perceive the implementation of the TeGoF as a clear success. First, all processes were successfully put into action. Second, client intervention was enabled in the area of missing performance testing as well as inadequate level of resources available to address a growing number of known defects. Overall, the TeGoF principles and processes were deeply appreciated by the client project manager and steering board in addition to other lessons learned. These actors emphasized that "in-depth [pre]testing of the application by end users and a [continual] involvement of FC's test[/quality] manager" (lessons learned presentation) contributed to the project's success.[7]

However, in examining the lessons learned, we identified a crucial point, i.e., *who should the TeGoF primarily serve in a business organization?* Should this framework serve as a vehicle enabling control over the testing and quality process and also to qualified non-IT (business) stakeholders? Because the TeGoF was not originally designed to fulfil this purpose, we do not

[7] With respect to potential bias stemming from the first author's participation at the project, it must be clarified at this point that the "lessons learned" was prepared independently, without any inputs or recommendations from the first author of the paper.

have a clear-cut answer to this question. More generally, we suggest that an IS department should be perceived as a *supplier* of software systems in a business organization. Thus, the business units would be considered as *customers* and principles similar to those in the TeGoF could be applied. Nevertheless, the level of involvement and *patterns of authority* between the IS department and business units will vary across business organizations [14,57]. The extant form of these arrangements can be related to *IT culture* [89] and thus also to the espoused beliefs and norms of behaviour in Schein's (organizational) culture construct. However, further research in this field is needed to fully understand how different types of these arrangements influence the control mechanism in contracted software development including the test governance process.

Summary: All of the processes described in the TeGoF were implemented successfully and contributed to early detection of problems in the software quality domain. The client project manager and client T/Q Manager acted via mutual interplay. In addition, the business representative participated in the process of behavioural control supported by the TeGoF. This fact brought up a fundamental question as to whether the TeGoF should also serve non-IT (business) actors.

Broader implications of the case: The project was finalized with a significant delay (two months) but was later perceived by a broader audience in the FC as a successful.

6.2 Case 2

6.2.1 Specific Context

Case 2 represents the second examined project in which the management principles of the TeGoF were implemented. The main aim of this project was to develop a complex front-end application that facilitated support of the FC's core business for the purpose of face-to-face interactions with potential customers (i.e., customer financial calculations). It is crucial to note that this application was distinguished by tight integration with a core client-server system that was already working well. Aside from technical factors that we omit in this study, this observation resulted in a significant organizational implication that we examine later.

The development of this application experienced numerous quality drawbacks at the earlier development stage. Thus, the project stage that we describe in this study was perceived beforehand by the FC's IT management as a potential source of problems. The FC's CIO was actively encouraging application of the principles of "quality improvement", especially in the test domain. As we learned later on, an inadequately low contracting price and elusion of the standard organizational approval procedure (which any other new project must routinely undergo before its realization) also became vital characteristics of this project.

6.2.2 Roles Involved in Test Governance

On the client side, the roles included the client project manager, client T/Q manager and business representative, as in Case 1. It is worth mentioning that in the initial project phase, the client project manager was replaced with another client project manager who claimed to be more experienced. However, even with experience in systems integration projects of packaged software, software systems development posed a challenging activity even for the second client project manager.

Historically and irrespective of this project manager swap, the business representative held great informal power within the client organization to make decisions instead of the actual project manager. This situation was manifested by the fact that the business representative organized frequent informal one-to-one meetings with the contractor project managers and account managers with the aim of maintaining his power. Additionally, during project status meetings related to contractor and client team members, we found his behaviour to be strictly directive. This informal power was tightly connected with the heritage of developed applications that were dependent on the core system (a back-end data processing application deployed to maintain core business functionality) in which a similar organizational set-up was established long ago. In addition, the business representative was also involved in the acquisition process (especially in the selection of potential contractors).

Thus, the decision-making authority of the project was not generally clear, and serious (previously hidden) disagreements between the project manager and business representative frequently emerged. Both of these actors used formal and informal control power extensively, and numerous internal and external conflicts arose. In the midst of this scenario, the TeGoF principles were introduced to the contractor managers.

6.2.3 Contractor Reaction to TeGoF

The contractor managers were reluctant at first to use the control process embodied in the TeGoF and, in fact, refused its adoption. This approach changed later on due to the support of the client project manager. Soon after the principles had been introduced, a hidden organizational agenda was discovered. The reluctance of the contractor project managers increased as the implementation of the TeGoF principles supported by the client project manager and T/Q manager progressed. It was quite apparent that the basic issue was a previously negotiated low contract price and that the contractor was reluctant to "supply a BMW when you have ordered a Trabant" (contractor project manager). During an informal managerial meeting connected with the preparation of the future meeting (Process 2.1 – Review Test Plan and planned test effort), the client project manager noted that "This is the worst project I have ever managed in FC" and brought up hidden historic agendas of strong price pressures. Eventually, the price issue was escalated to the B-2 level. However, from this client management level, any debates proposing an approach corresponding with the previously bargained low price were denied. Thus, all potential negative consequences were fully transferred to the contractor.

It is worth mentioning that the contractor managers were fully aware of the intriguing behaviour of the business representative. Considering this as a *backstage reality,* they declined to openly convey further details because his informal organizational authority posed a threat for them. However, one of the contract managers reframed the factual state into the metaphorical expression, "He is a good player". With this subtle reference to games (e.g., poker, in which the players must put on a false face irrespective of the actual cards that they play during the game), he metaphorically explained the nature of the business representative's interventions.

6.2.4 Discussion of the Case and Lessons Learned

The TeGoF can be considered one component of a broader client control mechanism in contracted software system development. In this case, the main deficiency of the mechanism as a whole was caused by the fact that no client manager involved in this project (including T/Q manager) was able to effectively force the contractor to develop the software system within the allocated timeframe, with the demanded quality, and with all of the requested features.

Irrespective of numerous requests from the client managers to the contractor project managers (the second "crisis" contractor project manager was later appointed), the contractor appeared reluctant to make major changes to the organization of this project. From the TeGoF viewpoint, the amount of necessary test work was identified by the client T/Q manager as significantly exceeding the given timeframe. However, no significant boost in testing manpower was enacted, although it was repeatedly requested in advance. The probable explanation lies in sunk costs, relatively low contractual penalties, and the already bad reputation of this contractor within the client organization. Therefore, pragmatically speaking, there was *nothing more to lose*.

During a status meeting, the client project manager noted: "If somebody from the board [of directors] sees this, they will fire us all", referencing (in his own word) the "amateurism" of the contractor and the hopelessness of the client managers. As our analysis later showed, one of the key causes that resulted in such a situation was the client project manager's reluctance to pass on bad news to the CIO. In the literature review, we characterized this situation of passing on bad news as *whistle blowing*. However, the client project manager decided *to keep mum* instead.

According to Schein's cultural model presented in the literature review, visible artefacts and observable behaviour are just one component of the culture construct. Schein argues that although these artefacts can be observed, they cannot be deciphered correctly without interpretation of the two remaining layers. The nature of factual client project management mechanism, which can be characterized in this case as *We can get it under control ourselves (without letting CIO know),* can be thus viewed as a manifestation of the two deeper layers: (a) espoused beliefs, values, rules, and behavioural norms and (b) tacit, taken-for-granted, and basic underlying assumptions. In this case, we are convinced that the second layer, i.e., rules and behavioural norms, particularly contributed to the above outlined situation. Arguably, we believe that the main role responsible for such situation is the client project manager who is responsible for the project overall. Although it is evident from the presented data that the project manager was familiar with the situation, he failed to begin the escalation process. Later, he noted: "Now, everybody is saying we failed because we did not approach [CIO's name]. However, when [previous client project manager's name] complained about a similar situation, he was suspended [from this project]". Thus, one of the rooted behavioural norms in this client organization can be characterized as *Project manager must handle all project-related situations by himself,* even if situations go awry.

The third layer likely contributed to this project manager's behaviour as well. Schein claims that certain organizational subcultures could be derivatives of broader occupational cultures in industries or professions. It is reasonable to expect that the occupational culture of project managers can be characterized by a tacit presumption similar to the one we describe above and thus subject to similar implicit behaviour of members of the group (when not actively encouraged by their employer otherwise, thus *overridden*). However, this estimate is not adequately supported by the current data and should be primarily understood as an impulse for future research.

Summary: All of the processes (paradoxically, except process 6.2 – Review Lessons Learned document) described in the TeGoF were implemented successfully and contributed to early detection of problems in the software quality domain. The client and T/Q manager were gradually accepted by the contractor as important players in the Test Governance process exercised in terms of behavioural control. Nevertheless, the TeGoF did not cover the situation that occurred when a client project manager attempted to cope with all project-related bad news by himself as a result of an unhealthy organizational culture and climate. In addition, the broader ethical implications of a *low price – extreme expectations* pattern are of a particular interest and are related to potential TeGoF deployment elsewhere.

Broader implications of the case: The project was later abandoned irrespective of sunk costs. The contractor received a penalty, and cooperation between both companies was temporarily terminated from the FC's side. Moreover, the business representative was forced to leave the FC.

6.3 Case 3

6.3.1 Specific Context

Case 3 represents the last research setting, which again is quite unique because it was represented by a software program [28], i.e., a chain of software projects with a long organizational history of circa 10 years. Thus, the behaviour of the actors was influenced by long-term relationships. The developed software system was considered as a core system (see above). If one can metaphorically speak of a *social stratum* of software systems in this financial organization, this application would be likely be considered as that of highest concern.

Aside from the internal IS department, a multinational and well-known external consulting company managed the software system development. Thus, a combination of internal and external resources required rather demanding coordination. The long-term involvement of this external consulting company appeared to polarize the internal stakeholders in the FC. We further reference employees of this contractor as *consultants* because this was the prevailing image that they sought to build. From one point of view, many employees as well as members of IT management openly or privately declared their negative judgments of this consulting company. We recorded quotations, e.g., "I don't like them" (client IS manager) and "They are like a sect" [all consultants are dressed uniformly according to strict dress code and they presented a sort of prefabricated customer-centric attitude] (another client IS manager) into the field notes.

From a different viewpoint, the consulting company was supported by the FC's parent company. So, many of the FC's local IS managers believed that it was impossible to control this consulting company effectively because they can "do anything they want" and "In the end, it is like they always blame somebody else for being responsible for an issue [technical problem, defect, etc.]" (two client IS managers). In addition, many internal FC employees (including IS managers) intuitively deprecated the fact that the consultants were widely trained in the area of impression management, whereas the soft skills of the FC's employees were not as developed. Only one IS client manager explicitly stated that he "[likes the situation because he] can learn something [i.e., impression management skills] from them". During our research, we had not observed a similar strongly prevailing negative attitude towards any other contractors.

With respect to the test management, crucial components of system testing (especially regression testing) were performed by client employees in various business units, i.e., sales, operations and customer services. This situation posed an interesting challenge for the test process management conducted by the contractor.

6.3.2 Roles Involved in Test Governance

On the client side, the roles included the client project manager, client T/Q manager and business representative, as in the previous two cases. A common sign shared by this software program and the software project described in the previous case represented the informal power of the business representative. Again, numerous informal one-to-one meetings were held, especially in the composition of business representative/contractor project manager. Although the factual power of the client project manager (both formal and informal) was quite weak, the power of the business representative was clearly dominant. Thus, in contrast with the previous case, this situation was characterized by almost no internal conflicts among the client project manager and business representative because the client project manager did not have any aspiration to *beat his head against the wall* and accepted his role. In fact, the business representative frequently attempted to ridicule the client project manager in the presence of the contractor as well as client staff. This behaviour can be viewed as a demonstration and consolidation of informal control power.

In the previous case, the contractor manager used the emic term "player" to describe a certain type of opportunistic behaviour of the business representative. Surprisingly, this metaphor was independently used by a client IS manager (in the initial phase of the research) with a subtle but significant difference. He characterized the client project manager in the manner that "[project manager's name] is *not a player*", meaning *He is not the one who decides*. The client IS manager was in fact suggesting that it did not make any sense to engage with *non-players* with respect to issues related to the TeGoF. Initially, not involving the client project manager in the TeGoF implementation seemed silly, but after prolonged time spent within the organization, it had become clear that his real influence was minor, and thus the key informant's advice was reasonable.

6.3.3 Contractor Reaction to TeGoF

Whereas the contractor consultants tried quite strenuously to build a positive image of the managed test process, this process suffered from several deficiencies. Such deficiencies included an enormous reliance on free testing without formal test documentation (impact on processes 3.1 and 4.1 – Review test scripts, Review test data) and the factual impossibility of re-checking the validity of testing due to missing evidence (Process 4.2 – Review output test artefacts). One of the key defensive tactics exhibited by the contractor was based on the argument that the contractor cannot entirely guarantee the quality of system testing process because its level of control over internal FC's employees is limited (see note above on the nature of their involvement). Therefore, an initial effort of the client T/Q manager was focused on this issue. However, due to strong resistance to all forms of test process improvement related to testing conducted by the listed business units, these activities were postponed because several internal organisational conflicts arose.

In general, dependence on the business units in system testing was presented within the FC as a favour by these units with the proclamation to "save costs" that would be otherwise consumed by the contractor. In reality, business unit testing was observed as inefficient and costly (the low price per time unit was countered by the enormous amount of time units claimed as consumed). Further, the defect reports often claimed false defects, thus resulting in enormous costs invested in wasted analysis. A number of different stakeholders declared the need for change, but such change never occurred in reality. The reasons for this situation were twofold.

First, this situation complied with the contractor's declarations that they could not fully guarantee the quality of system testing process, and thus the situation was generally profitable for the contractor. Second, the internal costs of testing were used as a hidden motivation source for the business units, irrespective of factual results and test process quality. Thus, strong advocacy of business unit managers towards preserving such configuration was in place. Due to the weak authority of the client project manager and the inability to differentiate between responsibility for comprehensive system testing and self-imposed and potentially useful informal involvement of business units in the system testing process ("But we are *happy* to have them [business units] in [system] testing, [because only they *know* the application]" – client project manager), one can understand this paradoxical situation.

6.3.4 Discussion of the Case and Lessons Learned

In this case, we perceive the implementation of the TeGoF as a partial success. From one point of view, all key processes were put into action (except for process 6.2 – "Review Lessons Learned document"). From another point of view, the real contribution was rather limited and at times even formal. The inadequacies of the FC's organizational capabilities related to this software program were demonstrated by the fact that evidence gathered by the TeGoF detection capabilities was not adequately used for decision-making. For example, the process 5.4 – Organize "End of System Testing" meeting did not fulfil proposed expectations because the formal criteria required to finish system testing (especially the number of defects distributed according to severity) were not met. Despite this fact, User Acceptance Testing was initiated. We can relate this behaviour to three origins.

First, the contractor's consultants were quite successful in creating good impressions and drove progress in this manner. Thus, even when the formal criteria were not met, the client project manager and especially the business representative were both relatively easily convinced by the contractor that the phase of User Acceptance Testing allows for plenty of opportunities to fix all remaining defects. While observing this behaviour in the field, we also identified a significant research gap in this area. Whereas studies concerning social desirability and impression management are an important component of social psychology research and more recently management research as well [90], little is known about the influence of social desirability on control of contractor-client relationships in a software engineering management context.

Second, due to the particular level of participation of the FC's internal IS department in this software program, it was often difficult to decipher who exactly was in charge of an issue and thus who was responsible. The FC's stakeholders generally perceived that it was incorrect to either formally or informally penalize the contractor when the problem lay in the client sphere, even when the contractor had the status of main system integrator and the client IS department served only as a subcontractor. This situation could be interpreted as an espoused belief or value in terms of Schein's culture construct.

Third, the factor of "keeping mum" played its role in this case as well. During numerous conversations, the client project manager repeatedly stressed his opinion with the rhetoric, "I cannot do anything", "This is a change management, not a project [even the project manager is appointed], "It rolls along, and we cannot stop it" and even "We are factually driven by [Contractor's name]. *They have us in their pocket*." From this data, it is evident that project management feared to address an issue and introduce serious change, primarily due to strong links between the contractor managers and the management of the FC's parent company. This situation resulted in a tacit norm of behaviour eventually.

Again, we can relate this project manager's behaviour to Schein's culture construct. We sum up the mentioned behavioural norm as *Do not start a fight in which you can bleed to death*, which was in fact the real metaphor used by the project manager during one of the interviews. Interestingly, only this contractor was perceived as so powerful, i.e., as potentially ready to cause "bleeding to death" of FC's employees based on informal power of his consultants.

An interesting behavioural pattern was evidenced by one of the IS client managers who expressed his concerns in the following way: "They [contractor's employees] are *not incompetent*. They simply do *just as much* as necessary." Thus, this manager suggested that the quality level of the contractor's work outputs (e.g., the quality of the test management process performed by contractor's employees) greatly varies based on the level of the FC's employees' pressure and control-related activities. We speculate that certain contractors might be more open to control if they have the potential to impress the client's employees and fulfil their expectations. This situation seems to confirm the proposition of Choudhury and Sabherwal that "it is a mistake to rely extensively on vendor self-control" [17].

In this case, we also demonstrated that visible artefacts were explicitly perceived by the FC's employees, for example, when one informant used the metaphor of the *sect*. Although the FC is a financial organization, the contractor's consultants were perceived as acting above the average level of the FC's not fully formalized IS department and/or above other contractors' employees', as displayed by their respective dress codes. At this point, we can speculate whether this artefact actually drives creation of good impressions (as likely originally intended) or triggers a form of a defensive reflex for certain of the FC's IS employees ("They are like a *sect*, [so be aware of potential danger when you encounter them]").

The final matter of interest is the strong dependence on testing conducted by the business unit's members in the system-testing phase. It is believed that mature test organizations demand full-time testers [91]. The described setup, i.e., system testing executed by multiple different roles, is not a typical pattern but is also not rare, even in software organizations [92,93]. However, in our case, we conclude that based on the presented data, the organization set-up involving business testers was arguably managed. Thus, during the system testing phase, mixing the modes of *business users claimed as qualified system testers* with *contractor's full-time testers* brought neither the demanded quality of test process nor clear responsibility for poor results. However, the client project manager seemed to believe that this organizational set-up is the ideal one. To recapitulate, a possible explanation could lie in the sphere of the intentional building of such an impression, initiated either by the contractor's employees or by the business unit managers.

Summary: All of the processes (except process 6.2 – "Review Lessons Learned document") described in the TeGoF were implemented successfully and contributed to early detection of minor problems in the software quality domain. The roles of the client project manager and client T/Q manager suffered from weak power. The main negative consequence of the TeGoF implementation consisted in the observation that the TeGoF's outputs were not used for factual decision-making because project management continued to be driven by clan-control and self-control mechanisms.

Broader implications of the case: The FC's internal tensions later resulted in withdrawal of the contractor from the software program, which was fully back-sourced to the FC.

6.4 Summary of Evaluation Process

In Table 6, we present a recapitulation of the TeGoF deployment in the FC.

Table 6
Summary of evaluation results

	General Project perception in FC	**Claimed TeGoF contribution**
Case 1	Success	Detection of problems, early warnings, source of corrective actions
Case 2	Failure	Detection of major problems, early warnings. Corrective actions demanded but rarely adopted
Case 3	Partial success	Detection of minor problems. TeGoF's results rarely used in decision-making related to project progress

7 Discussion

The designed management artefact and empirical findings presented in this research provide certain interesting insights. Generally, the issues of IS/SE managers related to the test process, as viewed from the stance of client organizations, are the locus of our analysis, solution proposal, and solution evaluation. We focus on a discussion of our findings in this section. First, we propose modes in which the TeGoF can be used within industry, i.e., how practitioners can benefit from our research. Second, we discuss certain of our findings and relate them to research reported by others.

7.1 Practical Implications of TeGoF

There are three basic options of organization design patterns in which client organizations could benefit from the TeGoF introduction, and they are described as follows:

- **The TeGoF as a tool of control in traditional contractual relationships (business clients)**

  In this case, a traditional contract model (fixed-price model) is followed. The TeGoF is employed in its full length as described in this paper. The basic aim is to assure a particular level of independence that complies with the classic idea of testing as an independent activity separated from development [54], i.e., from the contractor.

- **The TeGoF as a tool of control in global software development (software-intensive clients)**

  Essentially, the TeGoF was particularly designed to cover business client needs. However, we believe that this tool can also address software-intensive client needs, e.g., as a component of Global Software Development management as well. From pragmatic point of view, it is reasonable to expect that software-intensive clients employ well-qualified project managers who tacitly perform these activities to a certain extent already. Thus, the potential effects may not be as evident for practitioners in this mode.

- **The TeGoF as a tool of control of testing service quality (business or software-intensive clients)**

  In this case, only the testing is contracted as one particular stage of software development activities. In the literature review, we reference this mode of relationship as the *test factory* or *managed services*. The other components (i.e., software design and construction) are performed either internally or contracted to different contractors. Because the level of independence is already assured, the main aim of the TeGoF is to control the quality of this independent test process. The TeGoF can also provide a foundation for a definition and common understanding of a similar model of contracted testing services, i.e., to conceptualize the interactions between clients and testing service providers.

Moreover, the TeGoF might also serve as a tool of control in an internal-only setting, i.e., outside of its contractual context. In this case, a central organization unit (e.g., test process improvement group [91]) might benefit from its adoption and use it to control decentralized testing activities in the respective project contexts.

7.2 TeGoF's Implementation in the Realm of Social Interactions

Given that the TeGoF represents a SE/IT management (social) artefact, we focused on social interactions that stemmed from TeGoF introduction to the FC and fully relied on the intuitive perceptions of key FC stakeholders who were expected to most benefit from the TeGoF. Nonetheless, it is necessary to keep in mind that by following this approach, we linked our research to interpretive research tradition, which is idiographic in nature. Therefore, this work exists in sharp contrast to the hypothetico-deductive reasoning of traditional positivist science [64,80]. In this sense, our main aim was to provide interesting insights (as ethnographers typically do), but unfortunately, we were not able *to prove* the TeGoF's efficiency in a directly quantifiable manner. Nevertheless, we believe that our findings improve the understanding of SE/IT management in contractual relationships.

In Section 6, we demonstrated how key actors attempted to make sense of their world in matters of the TeGoF and contractor management in general using several examples. At this point, we stress one interesting aspect of our cases: use of metaphors. When ethnographically informed research is conducted and written, it is common to take advantage of these stylistic means of communication [72]. A metaphor is a form of analogy or "ways of seeing things as if they were something else" [94]. As we have repeatedly demonstrated, metaphors are a powerful source of subjective meanings related to organization actor perception.

The information technology and software engineering domain can particularly benefit from metaphors as we explore the social side of computing (e.g., management of SE/IT). For example, Kaarst-Brown and Robey [89] sketch a portrait of five IT cultures of business organizations. These IT cultures are pictured in the form of mythical dragons (technology) and wizards (IS personnel) in five different standings. The dragon can be depicted on pile of gold, as caged, as a pet, as a member of the team with business users or even as dead. The wizards may be present or missing. Without a doubt, contractor managers and client managers who interact during contracted software system development are greatly influenced by real conditions in particular business organizations represented by the form of similar archetypes.

In addition, the TeGoF implementation was subject to the influence of distinct contextual factors present in the FC, and we particularly stress the following one. In two independent cases, our informants used the metaphor of a "player" in connection with the contractor management mechanism. According to this perception, the *player* is a person who holds significant formal or informal power and who has the right to decide on the contractor's destiny (Case 3). The player can also be someone who conforms to this definition but uses his/her power in an opportunistic manner without broader ethical implications as a component of the *game*, i.e., only to win the entire pot for himself/herself (Case 2). Unfortunately, the behaviour of *players* also influences implementation of management artefacts similar to the TeGoF to a large extent. Thus, it is important to recognize these organizational *players* because they can be a potential key to success or failure. For example, Ngwenyama and Nørbjerg [71] show how a Software Process Improvement (SPI) group used organization influence processes to work with similar *players* towards successful SPI implementation when top management support was weak. In our research project, we were not as successful in overcoming the resistance of all *players* that we encountered within the dedicated timeframe. However, we learned important lessons summarized in the case descriptions presented.

7.3 Trust, Control and Power

From the beginning, we were aware of the important role of trust. In the FC, the original mechanism of contractor management heavily relied on trust, i.e., self-control of contractors. Originally, behavioural control mechanisms were not implemented by the FC managers. In this mode of contracted software development, software quality was perceived as inadequately low by the FC's upper managers, which was the reason why the first author entered the FC initially. Our initial motivation to design and implement the TeGoF was primarily driven by organization rumours and the research literature, as explained in Section 4. Based on these sources, we identified the need to balance trust versus control as a key driver of our research, which means that we realized quite soon that control is related to trust. In fact, "[c]ontrol is the other side of the [same] coin" [66]. To reach a balance, we focused on the test processes conducted within the framework of contracted fixed-price software development projects. As a consequence, we designed the TeGoF, an SE/IT management artefact for the purpose of client organizations.

The story of the TeGoF began as a story of the need to balance trust versus control. The story ends as one that offers a rich picture of several key events that led to the following conclusion related to power. As described in Section 6, the FC did not benefit from the TeGoF to its full extant in Cases Nos. 2 and 3. In both cases, identification of testing weak points via the TeGoF was not followed up. We believe that one important factor played a crucial role in this phenomenon and that is power.

However, we did not initially consider power in the TeGoF design or deployment. Many other research pieces concerned with trust and control actually did not consider power either, as summarised in [66]. Nevertheless, in our opinion, power *should be* understood as the third key player in the interplay among control-trust-power. As [66] states: "It is possible to omit one or two of power, trust, and control, but the analysis of the client–vendor relationship suffers consequently." Our research fully confirmed this proposition. In Case No. 3, the role of power was evident. Because the T/Q Manager held *no internal formal power* based on a certain type of the FC's formal organization rule of "heavy calibre",[8] he was not able to prevent this situation because upper management support was weak. Thus, this manager relied only on personal relationships and influence tactics. Moreover, the contractor used *backstage power* to navigate

[8] The Test Methodology document was not considered as a heavy-calibre organization artefact from the perspective of the FC's employees. From the perspective of the contractors, this document was widely accepted as a quality enforcement tool that should be followed in a certain manner. When these two perspectives clashed, the consequences were unpredictable.

the choppy waters of the FC. In Case No. 2, the role of power was not as evident initially. In this case, the contractor likely used *bargaining power* later on to bargain with the upper managers and chief executives backstage (hidden from our direct observations). Moreover, the observable behaviour of the business representative represented a demonstration of both internal and external power *par excellence*.

Aside from acting a tool of control, we realized that in certain cases, the TeGoF might be used as a tool of power. This setting is likely to occur under particular circumstances, i.e., a large client company factually dictating contractual conditions that involve a high level of control but low-cost pricing. Under unfavourable circumstances related to the current market supply and demand for software systems development, it appears reasonable to expect that certain contractors would be eventually forced (under the influence of relevant microeconomics factors) to accept such a contract. In such cases, mechanisms akin to the TeGoF might potentially serve as a tool of power over afflicted contractors. Because we do not intend to address similar ethical issues in depth in this paper, we simply state this dilemma for further exploration.

7.4 Potential Influence of Culture on Trust and Control

In our research, we are also concerned with additional possible factors that could influence how control, trust and power are exercised or perceived. We believe that a strong cultural dependence of software development and software engineering management practices (also including the software test domain) exist and that issues related to control and trust can be considered a vital component. For example, Scandinavian IS/SE research is claimed as widely reluctant to the basic presumptions of models similar to the one we developed. Instead of control, Scandinavian IS/SE research historically placed a large emphasis on end-user involvement [95]. In addition, from the analysis of Japanese and U.S. client practices in the domain of testing [43], one might conclude that Japanese client organizations exhibit a high level of control and low level of trust, whereas U.S. client organizations exhibit the exact opposite. However, all of these generalizations currently should be handled with care.

At the moment, we perceive issues of control, trust and power as possibly emerging from values, behavioural norms and taken-for-granted presumptions embedded in Schein’s culture construct and manifesting in the behaviour of organization members or creation of distinct organization artefacts as results of this behaviour. We agree with Schein and other authors [e.g., 96] that exploration of culture is a complex endeavour because respective types of culture (national, organizational, occupational, IT, etc.) are difficult to imagine as bordered entities that do not interfere with one another. However, these suggestions are not fully reflected by the current research, unfortunately [97]. In a number of current research contributions, only one particular level is examined in a form that evokes an over-simplistic approach. In general, we propose to examine these issues in a more holistic manner. A layered vision of culture viewed as a “spinning top” [96] and Schein’s construction of culture might provide interesting means for such future efforts.

For example, considering the Kensa test quality group from [43], we could perceive the existence of Kensa (or any SQA unit, more generally) as a form of visible organization artefact (Level 3). In a particular organization, the decision to establish a group similar to Kensa might be a result of an organizational culture that steadily prioritizes software quality. However, this organizational culture could have been heavily influenced by the national culture because most members of the organization are likely Japanese with a historically heavy emphasis on quality in general [98]. Moreover, members of particular professions bring their own beliefs and behavioural norms (Level 2) obtained from their occupational cultures. Thus, quality management personnel working in Kensa would likely employ different beliefs than people working in marketing, for example, which render the situation even more complicated. Because a commonly held stereotype says that the Japanese tend to stay in their jobs for their entire lives, the influence from an outside environment might be heavily impaired. Thus, the influence of occupational culture might be relatively low in Japan but quite huge in the U.S. In Japan, the influence of occupational culture might be rather expressed by the organization subculture [35] of Kensa's quality management personnel, as an example.

In our case of the FC, the Test Methodology representing principles of the TeGoF could be considered as a visible and directly observable organization artefact (Level 3). The decision to involve the first author of this paper in the FC, and thus to create the TeGoF in the form of the Test Methodology document, was tightly bound to the CIO's subjective perception of low software quality in the contracted software systems, as we explain elsewhere. It is believed that managers heavily contribute to the process of formation of the organizational culture [35]. Moreover, the first author of this paper brought his own pre-understanding and beliefs of how to address software quality in contractual relationships (see Section 4). Because the first author is an active member of a community of practice and is responsible for university teaching of quality management topics, he could be considered as a member of the quality managers occupation group who is well informed on quality management global trends and concepts. This situation can eventually result in a distinct form of his beliefs (impact to Level 2).

Thus, we find the key determinant that created the initial presumptions for the TeGoF design, i.e., low trust and a perceived need for higher control, was not the national culture *per se* but the organizational culture (or possibly the contemporary organization *setup including also associated power relations*). Moreover, at least in one case (Case 2), the FC seemingly omitted long-term relationships with contractors and steadily preferred to apply only decision criteria based on the lowest price (this behaviour is more or less related to the exhibition of power). Under these circumstances, the introduction of a control mechanism was clearly the only immediate solution to increase quality. However, was FC's behaviour directly driven by national culture? We are not sure whether the national culture also played a role in our case. Generally, we believe that post-socialist societies [99] might exhibit a greater affiliation with control-related behaviour (in the cybernetic sense of control) due to a form of taken-for-granted presumptions (Level 1) formed during past decades. However, a huge research gap currently exists in this area, and thus our assumptions would rather be speculative at this point.

Similarly, we suggest distinguishing among issues of control, trust and power at the micro-level (individuals), meso-levels (working groups and organizational subcultures) and macro-level (organizations). For example, our own encounter with power occurred at the micro- or meso-level, although members of both client and contractors were involved. In contrast, the TeGoF was intended to perform test process control at the macro-level and not to directly control individual relationships between the client and contractors' employees but rather organization relationships. To conclude the discussion of the FC's case, we suggest that trust and control are mutually coherent and further influenced by power. Micro-level power caused certain elements of the TeGoF to become forceless within the FC. It is likely that organizational culture had a strong influence on the fact that the FC tended to exhibit strong bargaining macro-level power (demanding low prices from potential contractors), which resulted in low macro-level trust and the consequent necessity of raising macro-level control. Interestingly, in his memos, the first author documented that micro-level trust between him and all contractor managers was high overall, except for in Case 2.

8 Conclusion and Outlook

This paper focuses on issues related to control of the test process in the context of contracted software development. We understand this process as a component of a wider acquisition process for custom-built software. Aimed at client organizations, we propose the TeGoF as an innovative tool of software engineering and information technology management. As a form of TeGoF evaluation, we present the results of three action research cases.

8.1 Main Conclusions

We summarize the main conclusions related to this piece of research as follows:

- **The TeGoF increases client insight into the test process performed by contractors.**

When custom-built software is developed, the test process is often an integral component of contracted activities covered by a single contract. Thus, this process is conducted by the same party as that involved in the software construction process. However, this set-up involves potential threats to client organizations as suggested by the agency theory. As we show in the empirical section, the TeGoF contributed to early detection of several flaws related to the test process and was perceived as a useful tool by client managers. Other client organizations can benefit from the TeGoF in a similar manner.

- **The TeGoF is especially useful in an environment in which the level of trust is low.**

In the previous section, we explained that the control mechanism is only one portion of the concept of the interplay of client-contractors interactions. The remaining two components, i.e., trust and power, must be considered if the principles captured in the TeGoF are to be implemented in a meaningful way. In a high-trust environment, implementation of the TeGoF may not be reasonable at all. Moreover, the potential influence of trust and power should be examined not only from the perspective of future effects on mutual relationships between the client and contractor (macro-level) but also internally, i.e., from the perspective of social interactions among key client stakeholders who might be formal or informal controllers (micro-level; see also below).

- **Success or failure of TeGoF implementation is highly context-dependent.**

The result of TeGoF's implementation, i.e., the concrete effects of the prescription on how to interact with the contractors, is the subject of sense making and a result of numerous interventions of particular actors across both organizations. This process stems from the individuals and cannot be prescribed or formalized. In this sense, the TeGoF should be understood as a guideline for client managers and not as a key to universal success in management of contracted software development. Implementation of the TeGoF might require other change-management activities consisting of strong leadership or coaching skills, utilization of organization influence tactics [71] or organization development activities [35], as examples.

8.2 Contributions

The contributions of our paper are threefold. First, we introduce the Test Governance Framework (TeGoF) itself. In the existing literature, it is not possible to find the exact formula for how client organizations exercise their control mechanism relative to the test process, at least in the form of empirically grounded research. Our paper enables conceptualization and improves understanding of interactions between clients and contractors in this area. Thus, the main contribution of this paper consists of the Test Governance Framework as a tool for pinpointing the significant limits of current research concerned with control issues in the context of client-contractor relationships and especially stressing the perspective of an inadequate scope of testing performed by a contractor. We find that a similar discussion is vital because we have anecdotally noted a great variation of opinions on the issue in the industry praxis, metaphorically said as *whether a client has the right to look into the kitchen during cooking*. Of course, this question makes sense only in contractual development in which custom-built software system is developed and not in product line development.

Second, we present our results in the form of a naturalistic evaluation of the TeGoF in the field. We do believe in practical and theoretical relevance of the extant form of the TeGoF, but at the same time, we are aware of the role of context that implies the uniqueness of every client. As we have argued, stressing a potential contextual factor is also in line with credibility and the underlying philosophical assumptions of action research. Related to this proposition, we showed that if software is acquired by business organizations, distinct factors influence the processes captured in the TeGoF. Among these major factors, we outlined the factor of multiple control exhibited by client stakeholders, which was a potent source of power-related conflicts stemming from interactions among the FC's members. Our three action research cases increased understanding of the ways in which SE/IT management artefact implementation should be performed in different types of organizations.

Third, we also demonstrated how methods of ethnographically informed action research could be used to examine a designed management (social) artefact. Similar to recent appreciation of unobtrusive ethnographically informed case studies focusing on software practice, this active form of an inquiry holds a great potential. We argue that this rough equivalent of Schein's clinical research model [100], which is well known in the area of organization development and (cultural) change, can provide fresh and fruitful insights into the studies of software practices, and important practical implications can be offered to the industry simultaneously. To the best of our knowledge, this is a novel approach in the arena of software engineering and information systems development. Our research can serve as a guideline in this sense, which is considered as the third contribution.

8.3 Future research

During our research, which was explorative in its nature, we also identified three key themes that can serve as a basis for future research. We generally encourage further research focusing on contractor-client relationships within the frame of software acquisitions, irrespective of client organizations as software developers or businesses in nature. Historically, management and social issues have been highly undervalued areas in the discipline of software engineering [61]. Currently, this research area has become more well established, particularly due to great efforts in the area of Global Software Development research related to offshore development [1]. However, we argue that onshore acquisitions of packaged or custom-built software also create new, important, and highly challenging social and technical issues [101] only partially covered by the current Information Systems research [13]. We believe that the following research streams could provide fruitful insights into the important areas of SE/IT management practice.

- **Client IT culture and *patterns of authority* in contracted software deliveries**

First, in Case No. 1, we questioned the idea of involving business stakeholders in the control process implemented by the TeGoF. From the viewpoint of software organizations, it may be viewed as groundless to discuss such issue. However, IS departments in business organizations are generally perceived as a supportive functional area and not as the value-earning core. From this perspective, *patterns of authority* (responsibilities for decision making) are often formally or informally set up in a manner that heavily involves core-business stakeholders and chief executives as the key decision-makers [57].

In business organizations, specific IT cultures might exist [89]. Moreover, IT management and governance frameworks, e.g., COBIT [25], might be in place. Both of these factors potentially influence contracted software deliveries, their management and their success. For example, client managers' *patterns of authority* would likely drive client project mechanisms quite heavily. Research studies that uncover these issues are thus desirable. Similarly, it may be interesting to explore how custom-built software system deliveries and packaged software acquisitions are influenced by particular software and test management principles captured in the "Build" domain of the COBIT 5 when this framework is applied by a client organization.

Currently, most of the research focusing on the context of client organizations has been conducted in the Information Systems discipline and is basically theory-driven [102] and rarely interested in software processes and other pragmatic management concepts. Software Engineering can potentially benefit from an improved understanding of contextual factors specific to business organization in the case in which a more straightforward approach is adopted by SE researchers.

- **IS/SE managers' occupational culture and individual cognitive elements**

Second, in Case No. 2, we noted the fact that Schein believes that broader occupational cultures have a significant influence on the behaviour of members of particular organizations by adopting certain occupational cultures' values, rules and norms. To our best knowledge, research contributions in the current computing research are scarce. Nevertheless, certain promising initial contributions already exist [96,103]. We believe that this area represents a notably vital stream of research and that we should focus on exploration of the differences between IS (client) and SE (contractor) project managers from this perspective. Knowledge of these differences appears to be the key to improving our understanding of failed projects and trust, control, power and motivation. As a sort of parallel, we note the research in the field of medicine that explores differences in behavioural norms and values between surgeons and intensivists (critical care specialists) [104] and how these two occupational groups address decision making in end-of-life situations. Ethnographically informed research design appears to be an ideal means for such research.

Moreover, although we are aware of the phenomena, we consider a proper exploration of cognitive biases to be beyond the scope of our study. Nevertheless, we strongly believe that cognitive dissonance [105] and confirmation bias greatly influence management of contracted software deliveries and especially the related decision making, particularly on the client side. Similarly, one of the client project managers informed us (in a situation unrelated to any of the described cases) that he attempts to modify perceptions of the client team members on purpose to increase their motivation in a form of "positive thinking". Thus, this manager underestimates certain "bad news" intentionally. In addition, the relationships of all of these phenomena with the deeper levels of Schein's culture construct should be considered. However, similar studies should build upon rigorous research design in a descriptive or experimental set-up because our own set-up was fundamentally formulative (i.e., design-oriented and explorative).

- **Contractors' social desirability and impression management**

Finally, in Case No. 3, we described the impression management efforts put forth by employees (consultants) of the contractor. Social desirability and impression management [90] represent an relatively unexplored area in the context of software development and contractor-client interactions. During prolonged time in the field, we noted that other contractors are generally aware of the significantly higher *art of impression management* exhibited by employees of the multinational consulting company (contractor 3) and perceived their own deficiencies in this area as a disadvantage. Similarly, certain of the client IS managers perceived similar deficiencies. Thus, further exploration in this direction would contribute to a better understanding of the applied portfolio of impression-building techniques and its consequences on client organizations that acquire custom-built software and interact with socially skilled contractors. Similarly, the art of building good impressions can aid contractors in improving their competitive capabilities. This area evidently calls for a qualitative and possibly ethnographically informed research design.

Acknowledgement

This research was supported by the Internal Grant Agency Project No. IG406013 and the Internal Project IP 400040.

References

[1] D. Damian, D. Moitra, Global Software Development: How Far Have We Come?, IEEE Software. 23 (2006) 17–19.

[2] R. Sabherwal, The evolution of coordination in outsourced software development projects: a comparison of client and vendor perspectives, Information and Organization. 13 (2003) 153–202.

[3] R.P. Jain, R.S. Poston, J.C. Simon, An Empirical Investigation of Client Managers' Responsibilities in Managing Offshore Outsourcing of Software-Testing Projects, IEEE Transactions on Engineering Management. 58 (2011) 743–757.

[4] S. Atkinson, G. Benefield, Software development: Why the traditional contract model is not fit for purpose, in: Proceedings of the Annual Hawaii International Conference on System Sciences, 2013: pp. 4842–4851.

[5] M.A. Babar, J.M. Verner, P.T. Nguyen, Establishing and maintaining trust in software outsourcing relationships: An empirical investigation, Journal of Systems and Software. 80 (2007) 1438–1449.

[6] R. Heeks, S. Krishna, B. Nicholsen, S. Sahay, Synching or sinking: global software outsourcing relationships, IEEE Software. 18 (2001) 54–60.

[7] T. McBride, The mechanisms of project management of software development, Journal of Systems and Software. 81 (2008) 2386–2395.

[8] R. Kishore, H.R. Rao, K. Nam, S. Rajagopalan, A. Chaudhury, A relationship perspective on IT outsourcing, Communications of the ACM. 46 (2003) 86–92.

[9] International Software Testing Qualifications Board, Standard glossary of terms used in Software Testing v. 2.2, (2012). http://www.istqb.org/downloads/glossary.html.

[10] M. Doležel, Towards elimination of testing debt in outsourced IS development projects: a view from the client side, in: Information System Development: Improving Enterprise Communication, Springer International Publishing, 2014: pp. 135–146.

[11] R. Glass, V. Ramesh, I. Vessey, An analysis of research in computing disciplines, Communications of the ACM. 47 (2004) 89–94.

[12] D. Avison, J. Malaurent, Is theory king?: questioning the theory fetish in information systems, Journal of Information Technology. 29 (2014) 327–336.

[13] J. Iivari, The IS Core – VII: Towards Information Systems as a Science of Meta-Artifacts, Journal of the Association for Information Systems. 12 (2003) 568–581.

[14] C. Soh, C.E.H. Chua, H. Singh, Managing diverse stakeholders in enterprise systems projects: a control portfolio approach, Journal of Information Technology. 26 (2011) 16–31.

[15] L. Kirsch, The management of complex tasks in organizations: Controlling the systems development process, Organization Science. 7 (1996) 1–21.

[16] G.B. Giglioni, A.G. Bedeian, A Conspectus of Management Control Theory: 1900-1972., Academy of Management Journal. 17 (1974) 292–305.

[17] V. Choudhury, R. Sabherwal, Portfolios of control in outsourced software development projects, Information Systems Research. 14 (2003) 291–314.

[18] L. Kirsch, Portfolios of control modes and IS project management, Information Systems Research. 8 (1997) 215–239.

[19] K.M. Eisenhardt, Agency Theory: An Assessment and Review, The Academy of Management Review. 14 (1989) 57–74.

[20] S. Ghoshal, Bad management theories are destroying good management practices, Academy of Management Learning & Education. 4 (2005) 75–91.

[21] M. Jorgensen, How to avoid selecting bids based on overoptimistic cost estimates, IEEE Software. 26 (2009) 79–84.

[22] J.Y.-C. Liu, V.J. Chen, C.-L. Chan, T. Lie, The impact of software process standardization on software flexibility and project management performance: Control theory perspective, Information and Software Technology. 50 (2008) 889–896.

[23] C.G. von Wangenheim, D.A. Da Silva, L. Buglione, R. Scheidt, R. Prikladnicki, Best practice fusion of CMMI-DEV v1.2 (PP, PMC, SAM) and PMBOK 2008, Information and Software Technology. 52 (2010) 749–757.

[24] M.-L. Sánchez-Gordón, L. Moreno, Toward an Integration of Web Accessibility into Testing Processes, in: 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion, 2014: pp. 281–291.

[25] S. De Haes, W. Van Grembergen, R.S. Debreceny, COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities, Journal of Information Systems. 27 (2013) 307–324.

[26] W.W. Royce, Managing the development of large software systems, in: Proceedings of IEEE WESCON, 1970: pp. 1–9.

[27] COTS (COMMERCIAL-OFF-THE SHELF) GUIDE, 1993. https://nepp.nasa.gov/DocUploads/1219C61B-7337-48C4-8760E6456F861839/COTS guide.pdf.

[28] PMI, A Guide to the Project Management Body of Knowledge, 5th ed., (2013).

[29] R. Wendler, The maturity of maturity model research: A systematic mapping study, Information and Software Technology. 54 (2012) 1317–1339.

[30] A. Drechsler, Design Science as Design of Social Systems–Implications for Information Systems Research, Journal of Information Technology Theory and Application. 14 (2013) 5–26.

[31] W. Kuechler, V. Vaishnavi, The emergence of design research in information systems in North America, Journal of Design Research. 7 (2008) 1–16.

[32] T. Dyba, An empirical investigation of the key factors for success in software process improvement, IEEE Transactions on Software Engineering. 31 (2005) 410–424.

[33] M. Niazi, M.A. Babar, Identifying high perceived value practices of CMMI level 2: An empirical study, Information and Software Technology. 51 (2009) 1231–1243.

[34] L. Ostrowski, M. Helfert, Commonality in various design science methodologies, in: Proceedings of the Federated Conference on Computer Science and Information Systems, 2011: pp. 317–320.

[35] E.H. Schein, Organizational Culture and Leadership, 4th ed., Jossey-Bass, San Francisco, CA, 2010.

[36] I. Sommerville, Software Engineering, 8th ed., Addison-Wesley, Harlow, England, 2007.

[37] D.R. Denison, What is the difference between organizational culture and organizational climate? A native's point of view on a decade of paradigm wars, Academy of Management Review. 21 (1996) 619–654.

[38] D. Robey, M. Keil, Blowing the whistle on troubled software projects, Communications of the ACM. 44 (2001) 87–93.

[39] S. Ramingwong, A.S.M. Sajeev, Offshore outsourcing: the risk of keeping mum, Communications of the ACM. 50 (2007) 101–103.

[40] C. Park, G. Im, M. Keil, Overcoming the mum effect in IT project reporting: Impacts of fault responsibility and time urgency, Journal of the Association for Information Systems. 9 (2008) 409–431.

[41] M. D'Mello, T.H. Eriksen, Software, sports day and sheera, Information and Organization. 20 (2010) 81–110.

[42] M.D. Myers, F.B. Tan, Beyond models of national culture in information systems research, Journal of Global Information Management. 10 (2002) 24–32.

[43] H. Shah, M.J. Harrold, Culture and Testing: What is the Relationship?, in: 8th International Conference on Global Software Engineering, IEEE, 2013: pp. 51–60.

[44] A. Gopal, S. Gosain, The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance, Information Systems Research. 21 (2010) 960–982.

[45] A. Tiwana, M. Keil, Control in Internal and Outsourced Software Projects, Journal of Management Information Systems. 26 (2009) 9–44.

[46] A. Chua, Exhuming IT projects From their graves: An analysis of eight failure cases and their risk factors, Journal of Computer Information Systems. 49 (2009) 31–39.

[47] D. Galin, Software Quality Assurance. From theory to implementation, Pearson Education Limited, Harlow, England, 2004.

[48] P. Savolainen, J.J. Ahonen, I. Richardson, Software development project success and failure from the supplier's perspective: A systematic literature review, International Journal of Project Management. 30 (2012) 458–469.

[49] A. Sanz, J. García, J. Saldaña, A. Amescua, A proposal of a process model to create a Test Factory, in: ICSE Workshop on Software Quality, Vancouver, Canada, 2009: pp. 65–70.

[50] Y. Lu, T. Käkölä, A dynamic life-cycle model for the provisioning of software testing services: Experiences from a case study in the Chinese ICT sourcing market, in: 19th European Conference on Information Systems, 2011.

[51] S.M. Lee, C.K. Lee, IT managers' requisite skills, Communications of the ACM. 49 (2006) 111–114.

[52] M. Keil, H.J. Smith, S. Pawlowski, L. Jin, "Why didn"t somebody tell me?': climate, information asymmetry, and bad news about troubled projects, Data Base for Advances in Information Systems. 35 (2004) 65–84.

[53] A. Kataja, T. Tuunanen, Information systems development methods and reducing information asymmetry: a way to decrease project escalation in outsourcing?, in: Proceedings of the 14th European Conference on Information Systems, 2006: pp. 1–8.

[54] G.J. Myers, The art of software testing, Wiley, New York, 1979.

[55] R. Black, Managing the testing process: practical tools and techniques for managing hardware and software testing, 3rd ed., Wiley, New York, 2009.

[56] R.R. Moeller, Executive's Guide to IT Governance: Improving Systems Processes with Service Management, COBIT, and ITIL, John Wiley & Sons, 2013.

[57] V. Sambamurthy, R.W. Zmud, Arrangements for information technology governance: A theory of multiple contingencies, MIS Quarterly. 23 (1999) 261–290.

[58] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems research, MIS Quarterly. 28 (2004) 75–105.

[59] J.E. van Aken, Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules, Journal of Management Studies. 41 (2004) 219–246.

[60] J. Iivari, J. Venable, Action research and design science research - seemingly similar but decisively dissimilar, in: European Conference on Information Systems, 2009: pp. 1–13.

[61] D.M. Berry, Academic Legitimacy of the Software Engineering Discipline, CMU/SEI-92-TR-34, 1992. http://www.cc.gatech.edu/sparc/Resources/readings/berry.pdf.

[62] R. Baskerville, A.T. Wood-Harper, Diversity in information systems action research methods, European Journal of Information Systems. 7 (1998) 90–107.

[63] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering. 14 (2009) 131–164.

[64] C. Wohlin, A. Aurum, Towards a decision-making structure for selecting a research design in empirical software engineering, Empirical Software Engineering. (2014). doi:10.1007/s10664-014-9319-7.

[65] D. Coghlan, Insider action research: opportunities and challenges, Management Research News. 30 (2007) 335–343.

[66] A. Heiskanen, M. Newman, M. Eklin, Control, trust, power, and the dynamics of information system outsourcing relationships: A process study of contractual software development, The Journal of Strategic Information Systems. 17 (2008) 268–286.

[67] G. Kunda, Engineering Culture: Control and Commitment in a High-Tech Corporation, Temple University Press, 2009.

[68] C. Cassell, P. Johnson, Action research: Explaining the diversity, Human Relations. 59 (2006) 783–814.

[69] H. Robinson, J. Segal, H. Sharp, Ethnographically-informed empirical studies of software practice, Information and Software Technology. 49 (2007) 540–551.

[70] S. Van der Geest, Anthropologists and missionaries: brothers under the skin, Man. 25 (1990) 588–601.

[71] O. Ngwenyama, J. Nørbjerg, Software process improvement with weak management support: an analysis of the dynamics of intra-organizational alliances in IS change initiatives, European Journal of Information Systems. 19 (2010) 303–319.

[72] J. van Maanen, Tales of the field: On writing ethnography, The University of Chicago Press, Chicago, 2011.

[73] M. Myers, Investigating information systems with ethnographic research, Communications of the AIS. 2 (1999).

[74] S. French, Action research for practising managers, Journal of Management Development. 28 (2009) 187–204.

[75] M.D. LeCompte, J.P. Goetz, Problems of reliability and validity in ethnographic research, Review of Educational Research. 52 (1982) 31–60.

[76] L. Mathiassen, A. Sandberg, How a professionally qualified doctoral student bridged the practice-research gap: a confessional account of Collaborative Practice Research, European Journal of Information Systems. 22 (2013) 475–492.

[77] A. Dubois, L.-E. Gadde, Systematic combining: an abductive approach to case research, Journal of Business Research. 55 (2002) 553–560.

[78] A. Drechsler, Specializing a Design Science Research Methodology for the Domains of IS/IT Management and IT Project Management, in: Proceedings of the 22nd European Conference on Information Systems, Tel Aviv, Israel, 2014: pp. 1–10.

[79] K. Pandza, R. Thorpe, Management as Design, but What Kind of Design? An Appraisal of the Design Science Analogy for Management, British Journal of Management. 21 (2010) 171–186.

[80] G. Walsham, Doing interpretive research, European Journal of Information Systems. 15 (2006) 320–330.

[81] L.M. Given, The Sage encyclopedia of qualitative research methods, SAGE Publications Ltd, London, 2008.

[82] F. Znaniecki, The method of sociology, Farrar and Rinehart, New York, 1934.

[83] A.R. Lindesmith, Addiction and Opiates, Aldine, Chicago, 1968.

[84] B. Glaser, A. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine de Gruyter, Hawthorne, NY, USA, 1976.

[85] J. van Maanen, J. Sørensen, T. Mitchell, The interplay between theory and method, Academy of Management Review. 32 (2007) 1145–1154.

[86] Y. Guo, C. Seaman, A portfolio approach to technical debt management, in: Proceedings of the 2nd Workshop on Managing Technical Debt, 2011: pp. 31–34.

[87] E. Tom, A. Aurum, R. Vidgen, An exploration of technical debt, Journal of Systems and Software. 86 (2013) 1498–1516.

[88] M. Chemuturi, Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers, J.Ross Publishing, Fort Lauderdale, FL, 2011.

[89] M.L. Kaarst-Brown, D. Robey, More on myth, magic and metaphor: Cultural insights into the management of information technology in organizations, Information Technology & People. 12 (1999) 192–218.

[90] M.C. Bolino, Citizenship and impression management: good soldiers or good actors?, Academy of Management Review. 24 (1999) 82–98.

[91] TMMi Foundation, Test Maturity Model integration, (2012). www.tmmi.org.

[92] J.J. Ahonen, T. Junttila, M. Sakkinen, Impacts of the organizational model on testing: Three industrial cases, Empirical Software Engineering. 9 (2004) 275–296.

[93] M. V. Mäntylä, J. Itkonen, J. Iivonen, Who tested my software? Testing as an organizationally cross-cutting activity, Software Quality Journal. 20 (2012) 145–172.

[94] P.K. Manning, Metaphors of the Field: Varieties of Organizational Discourse, Administrative Science Quarterly. 24 (1979) 660–671.

[95] J. Iivari, K. Lyytinen, Research on information systems development in Scandinavia—unity in plurality, Scandinavian Journal of Information Systems. 10 (1998) 135–185.

[96] T. Jacks, P. Palvia, A Cultural Sociology Perspective on IT Occupational Culture, in: Proceedings of the Seventeenth Americas Conference on Information Systems, Detroit, Michigan, 2011: pp. 1–10.

[97] D. Leidner, T. Kayworth, Review: a review of culture in information systems research: toward a theory of information technology culture conflict, MIS Quarterly. 30 (2006) 357–399.

[98] D. Dobosz-Bourne, M. Kostera, The quest for quality: translation of a mythical idea, Journal of Organizational Change Management. 20 (2007) 60–73.

[99] A. Soulsby, E. Clark, Organization theory and the post-socialist transformation: Contributions to organizational knowledge, Human Relations. 60 (2007) 1419–1442.

[100] E.H. Schein, Clinical Inquiry/Research, in: Handbook of Action Research, Sage Press, Thousand Oaks, CA, 2006: pp. 228–237.

[101] Y. Dittrich, S. Vaucouleur, S. Giff, ERP customization as software engineering: knowledge sharing and cooperation, IEEE Software. 26 (2009) 41–47.

[102] J. Iivari, R. Hirschheim, H. Klein, Towards a distinctive body of knowledge for Information Systems experts: coding ISD process knowledge in two IS journals, Information Systems Journal. 14 (2004) 313–342.

[103] S. Ramachandran, S. V. Rao, An effort towards identifying occupational culture among information systems professionals, in: Proceedings of the ACM SIGMIS CPR Conference on Computer Personnel Research, ACM Press, New York, NY, 2006: pp. 198–204.

[104] J. Cassell, T.G. Buchman, S. Streat, R.M. Stewart, Surgeons, intensivists, and the covenant of care: administrative models and values affecting care at the end of life, Critical Care Medicine. 31 (2003) 1263–70.

[105] L. Festinger, A Theory of Cognitive Dissonance, Stanford University Press, Stanford, CA, 1957.

[106] S. Barley, G. Kunda, Design and devotion: Surges of rational and normative ideologies of control in managerial discourse, Administrative Science Quarterly. 37 (1992) 363–399.