# Organizational Patterns between Developers and Testers
## *Investigating Testers' Autonomy and Role Identity*

Michal Doležel[1] and Michael Felderer[2,3]
[1]*Department of Information Technologies, University of Economics, Prague, Czech Republic*
[2]*Institute of Computer Science, University of Innsbruck, Innsbruck, Austria*
[3]*Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden*

Keywords: Software Testing, Agile Testing, Conflict, Profession, Role Identity, Social Identity Theory, Organizational Structure, Combined Software Engineering, Software Engineering Management.

Abstract: This paper deals with organizational patterns (configurations, set-ups) between developers/programmers and testers. We firstly discuss the key differences between these two Information Systems Development (ISD) occupations. Highlighting the origin of inevitable disagreements between them, we reflect on the nature of the software testing field that currently undergoes an essential change under the increasing influence of agile ISD approaches and methods. We also deal with the ongoing professionalization of software testing. More specifically, we propose that the concept of role identity anchored in (social) identity theory can be applied to the profession of software testers, and their activities studied accordingly. Furthermore, we conceptualize three organizational patterns (i.e. isolated testers, embedded testers, and eradicated testers) based on our selective literature review of research and practice sources in Information Systems (IS) and Software Engineering (SE) disciplines. After summarizing the key industrial challenges of these patterns, we conclude the paper by calling for more research evidence that would demonstrate the viability of the recently introduced novel organizational models. We also argue that especially the organizational model of "combined software engineering", where the roles of programmers and testers are reunited into a single role of "software engineer", deserves a closer attention of IS and SE researchers in the future.

## 1 INTRODUCTION

Information Systems Development (ISD) is a team activity. Not even mentioning varying needs and demanding expectations of future IS users, the nature of the interplay among programmers, testers, analysts, and other functional roles involved in the execution of ISD activities influences outcomes of ISD projects to a great extent (Walz et al., 1993). Agile ISD is seen as an important step towards improving this interplay by promoting values of flexibility, cooperation, learning, and leanness (Conboy, 2009). Compared to traditional (i.e. sequential or phase-oriented) ISD methodologies, the Agile ISD philosophy brings two principal changes regarding the "human element" in ISD.

First, it assumes a less fragmented and little formalized distribution of responsibilities across different functional roles active in ISD, including software testers (Cohn, 2010). Second, it puts forward the view that software development is a complex socio-technical process understandable through studying people and their interactions (Balijepally et al., 2006). These two changes motivate our paper.

Though systems/software testing is a vital part of ISD activities, it has received scant attention in previous Information Systems (IS) research (Hassan and Mathiassen, 2018). Similarly, Software Engineering (SE) research has focused on technical challenges of software testing, and remains mostly silent on the management ones (Garousi and Felderer, 2017). In particular, there is very little work available that discusses the evolving role of testers in agile teams from an organizational point of view based on social science theories. This perspective is what we aim for, but the present paper is just a first step in this direction.

This paper concentrates on software testing personnel (testers, test engineers, test analysts etc.) and the changing nature of their role at present day. Due to space constraints, however, we do not further expand on how various sub-professions in software

Doležel, M. and Felderer, M.
Organizational Patterns between Developers and Testers.
DOI: 10.5220/0006783703360344
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 336-344
ISBN: 978-989-758-298-1

testing (e.g., test managers) are exactly impacted. We take a simplistic view that software tester or test engineer has been the one *who carries out the majority of testing work in ISD*.

Our aim here is to review relevant Information Systems (IS) and Software Engineering (SE) literature, identify gaps in it, and prepare the grounds for presentation and interpretation of our results based on an ongoing research project. In so doing, this paper investigates the interconnected problems of testers' role identity and organizational independence in ISD activities. To understand the problem comprehensively, we use also literature from Management and Organization Studies (MOS).

This paper proceeds as follows. Section 2 lays conceptual foundations. Section 3 presents an overview of the organizational patterns suggested by us as distinct. Section 4 indicates the further direction of our research. Finally, Section 5 concludes the paper.

## 2 CONCEPTUAL BACKGROUND

The points of departure of our paper are discussed in this section. Section 2.1 briefly highlights certain core features of Agile ISD approaches. Section 2.2 discusses differences between programmers and testers. Section 2.3 conceptualizes the problem of testers' independence. Finally, Section 2.4 concludes by presenting some thoughts on the profession of software testers.

### 2.1 Agile ISD

The popularity of agile IS development methods (Agile) has been steadily growing over the last decade (Conboy, 2009). More specifically, aside from small teams and start-up businesses, Agile gradually penetrates also traditional enterprises. In some organizations, the observed growth of Agile development initiatives can be, at least partly, attributed to the general popularity of the concept. Managers and executives have been always paying attention to emerging management trends, and Agile ISD approaches may be seen as one among their present day favourites (Cram and Newell, 2016). Cautiously stated, true efforts to introduce a revolutionary, people-centric management philosophy into the world of corporate organizing may drive the remaining efforts (Laloux, 2014).

Considering the nature of the shift from traditional to Agile ISD methods, it is essential to recognize that agile software development consists of socio-technical activities. This understanding contrasts with the predominantly technical, mechanistic understanding of software processes at earlier times (Balijepally et al., 2006), even though the socio-technical nature of software processes obviously did not emerge over-night (Fuggetta and Nitto, 2014). Indeed, ISD personnel is in the centre of research on ISD and software processes nowadays.

Another important change introduced by Agile directly influences the terminology adopted in this paper: Instead of the previously common term "developer", we use a less-frequent term "programmer" to avoid confusion with the Agile terminology. Specifically, the concept of *cross-functional development team* promoted by Agile has a significant organizational impact: "the [development] team needs to include everyone [e.g., programmers, testers, analysts, and business representatives] necessary to go from idea to implementation" (Cohn, 2010, p. 152).

### 2.2 Two Different Software Species

The mind-set of programmers and testers is considered as different (Pettichord, 2000). A tester is the one who empirically proves the system under test to investigate whether the system is able to stand its future operational mission. Examining system's behaviour, he or she is driven by the ideal of protecting future end users and mitigation as many risks as possible. By *breaking* the system, the tester pursues to improve it.

By contrast, programmers are the *builders*. "[F]lexing their intellectual muscles" (Cohen et al., 2004, p. 78), they may look for creative, technically-sound solutions irrespectively of potential negative implications for end-users. Differently put, while many programmers quite narrowly focus on technical aspects of ISD by prioritizing solution efficiency in technical terms, testers look primarily for solution effectiveness and fit-for-future-use. These differences are summarized in Table 1. Naturally, this table presents a sort of black/white, simplistic perspective.

Table 1: Characteristic differences between programmers and testers. Adapted from Zhang et al (2018).

| Dimension | Programmers | Testers |
|---|---|---|
| Work mindset | Build | Break |
| Key value | Technical excellence | Customer advocacy |
| Thinking focus | Theory | Practice |
| Project goal | Efficiency | Effectiveness |
| Job knowledge | Depth | Breadth |

It should come as no surprise that this dichotomy frequently results in conflicts between the two groups. Importantly, previous research indicates that this conflict is inherent to software development activities (Cohen et al., 2004). In principle, there are many sources of the conflict. From their rich research data, Zhang et al. (2014) identified five major categories, focusing on three common elements: process, people, and communication. Conflict management strategies differ accordingly (Cohen et al., 2004).

However, our knowledge that the conflict is inevitable little helps with avoiding organizational misalignments between programmers and testers in real organizations (Onita and Dhaliwal, 2011). This also brings us to the point that the observed conflict has not been comprehensively studied in traditional ISD approaches with respect to the V-model. This ISD concept (Boehm, 1984) portrays segmented test levels (i.e., unit, integration, system, and user acceptance), and assumes different expectations and testing tasks distribution at each of the levels. But most importantly, the V-model indicates that there will be a specific dynamics between programmers and testers at each of the levels.

Another walkable approach might be to radically change the rules of the ISD game. Going this route, Agile practitioners argue that Agile ISD methods help to reduce the tensions between programmers and testers by redefining the role of testers (Cohn, 2010). Aside from other factors, this redefinition is associated with the elimination of testers' independence from programmers. Yet another group of Agile practitioners calls for removing testers from ISD processes entirely (Anderson, 2003). In general, Agile ISD does not conceptualize software testing using the V-model, because all necessary test activities must be executed iteratively (Cohn, 2010).

We discuss the mentioned organizational strategies in Section 3. Prior doing so, we explain the historical role of testers' independence in ISD, and the ongoing professionalization of software testing.

## 2.3 The Cost of Testers' Independence

In general, testers' independence is codified by various practitioner sources by postulating a "common wisdom" that

> *A certain degree of independence (avoiding the author bias) often makes the tester more effective at finding defects and failures.*
>
> (ISTQB, 2011, p. 18)

The above thesis says that to prevent the "contamination" of their perspective, testers need to enjoy a certain level of organizational autonomy. Two examples follow. A high level of independence is when testers work in isolation and use formal ISD documentation as the primary source of information about tested applications. In theory, the testers should be better prepared to discover programmers' lapses. In practice, they may find themselves isolated and disconnected from project activities. In addition, such organizational set-up may strengthen adversarial relationships between programmers and testers (Grechanik et al., 2010).

An extreme case of testers' independence typically occurs when contractual relationships are involved. First and foremost, an external test factory run by a third party may be contracted (Andrade et al., 2017). Second, as a tool of vendor management, a special client unit might be designated to perform *quality verification* of vendor's IS development and testing activities. In such case, another psychological factor may drive vendor personnel's behaviour. That is the angst of defects that escaped detection at vendor premises (Shah and Harrold, 2013).

Despite the fact that some organizations decided against the organizational set-up with a high level of independence for testers, practitioner literature frequently promotes it (McKay, 2007). In addition, the existence of an independent testing unit in organizations was previously institutionalized as an important criteria of test maturity; for example, it is suggested by a popular test management guideline (TMMi Foundation, 2012).

## 2.4 The Profession of Software Tester

In a broad sense, professions are vocations that carry out professional activities in a given area of practice (Hughes, 1958). The execution of professional activities might be conditioned by a previous formal training, length of practice, or entirely open to a loosely defined group of people who claim to belong among the professionals. The former two criteria apply, for example, to medical professions, whereas the latter one to software programmers and testers. Aside from formal regulations that might be in place, many professions informally or semi-formally (e.g., through optional certifications) postulate certain behavioural norms that are then expected to be followed by the profession's members.

Using the language of social sciences, this process is related to the social construction of "self-identity" of professions. Through the formation of shared meanings, members of a profession gradually reach consensus what the professional norms are. In the following, we use the term "professional identity" to label distinct "goals, values, beliefs, norms, [and] interaction styles" (Ashfort, 2001, p. 6) settled in a profession. In young fields like software development, the above formative processes are naturally different from the processes in well-established, formally regulated professions like law or medicine (Evetts, 2014).

For a number of years, testers were socialized in ISD environments where they were to become *quality advocates* (a rather soft version of the metaphor), *quality gatekeepers* (a mild version of the metaphor), or *quality police/enforcers* (an extreme version of the metaphor) (Charrett, 2012). Not long ago, managers of testing teams were instructed to build their unit as "The Perfect Beast" (McKay, 2007 n.p.), by metaphorically combining qualities of several animal predators to fight with software bugs (and possibly also with programmers). And Software Quality Assurance departments seen as *quality watchdogs* were encouraged to "bite if necessary" (Chemuturi, 2011, p. 65). Interestingly, people from industry routinely (but incorrectly) mix the role of *software quality assurance* and the one of *software testing* (Koch, 2000).

Often mentioned during trainings, conferences, or in books, all these metaphors and labels may be seen as part of testers' professional identity built through the past decades. The metaphors also somehow relate to the level of testers' independence and their main mission as explicitly formulated or tacitly expected by an organization. Sadly enough, little guidance grounded in empirical research is available to the practitioners who struggle with whether one of the mentioned modes is fit-for-purpose in their company. Differently put, the role models that describe expected or ideal professional behaviour in software testing are often anecdotal, based on the personal experience of trainers, mentors, or various testing school gurus. And as a profession, software testing heavily relies on personal experience, which is not always shared with a wider community (Beer and Ramler, 2008).

## 3 TYPICAL ORGANIZATIONAL CONFIGURATIONS

In this section, we review three typical organizational patterns that can be encountered in software companies and IT units/divisions nowadays. Organizational configuration implemented in a particular company results from perceptions held by the company regarding the role of software testers in the company's ISD processes (Charrett, 2012). Note that we do not discuss various sourcing options (e.g., offshore, onshore, nearshore), but we focus on programmers and testers in the sense of their *standings* and organizational *relationships.*

### 3.1 Traditional Testing: Testers as Gatekeepers

#### 3.1.1 Grounding

Originating in a late-1970s vision of Barry Boehm (1979), software testing has been traditionally perceived as a distinct, separate ISD phase. The concept of separate test levels with dedicated testing responsibilities codified by the V-model (i.e., unit, integration, system and user acceptance test levels) has been traditionally presented as a form of test maturity ideal. According the V-model, somewhat exaggeratedly put, the more test levels exist in the organization and the higher number of diverse groups involved in software testing, the more mature test process the organization exercises.

#### 3.1.2 Key Industrial Challenges

Though the dilemma *"What level of independence should testers enjoy?"* is one among "test management classics" for ISD managers, little research effort has been devoted to explore it scholarly so far (Garousi and Mäntylä, 2016; Sunyaev and Basten, 2015). Not surprisingly, the extreme cases when testers and programmers are geographically separated with no mechanisms to facilitate effective communication between them are typically found dysfunctional (Grechanik et al., 2010). However, quite little is known about the real effects of having testers reporting back to a manager who supervises both programmers and testers in a co-located environment. This single-point-of-responsibility configuration is established in many companies and supported by the way how a typical ISD project is managed (Atkinson, 1999). Though the problem of conflicting goals of the project manager or development manager is quite evident, there is no simple remedy.

This case is represented by the full independence scenario (Figure 1, type A). Importantly, the character of the metaphorical wall ("Who reports to whom?") and its "permeability" ("How testers interact with programmers?") should be of interest to further research efforts exploring this area. Similarly, the conflicting goals dilemma should be explored from the viewpoint of software testers and their everyday activities.

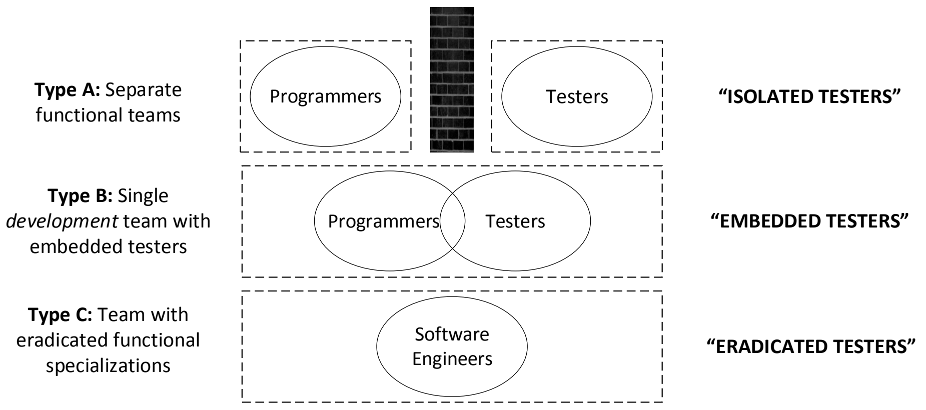
Type A: Separate functional teams
Programmers
Testers
"ISOLATED TESTERS"
Type B: Single development team with embedded testers
Programmers
Testers
"EMBEDDED TESTERS"
Type C: Team with eradicated functional specializations
Software Engineers
"ERADICATED TESTERS"

Figure 1: Typical organizational patterns between programmers and testers.

## 3.2 Agile Testing: Removing the Wall

### 3.2.1 Grounding

To solve the challenge of conflicting ISD priorities, Agile ISD approaches understand software testing as part of whole-team responsibility (Cohn, 2010; Crispin and Gregory, 2009). SCRUM, a well-known agile framework, explicitly states that testers are integral part of the development team. In other words, testers are "embedded" into the development team, and their responsibilities overlap with programmers to some extent (Figure 1, type B). Blending the responsibilities of programmers, testers, and analysts by creating a "cross-functional team", SCRUM aims to remove unnecessary organizational boundaries.

> *Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule; ...*
>
> (Scrum.org, n.d.)

In theory, the inherent conflict between programmers and testers should be solved. In practice, research shows that some testers still report problematic relationships with programmers (see below). This could be partly attributed to the fact that there is no single way of "doing Agile ISD"; the same label "Agile" may represent quite diverse ISD strategies in reality. Aside from *pure* Agile ISD approaches, many companies follow the way of *tailoring* or even *dabbling* the original Agile ISD philosophy (Cram and Newell, 2016). The latter two approaches indicate that in present days some companies tend to *hybridize* software processes rather than fundamentally transform their nature (Kuhrmann et al., 2017).

### 3.2.2 Key Industrial Challenges

Practitioner literature suggests that moving from waterfall to agile environment may be a challenging task for testers (Crispin and Gregory, 2009). The main reason behind this challenge is the nature of expected change in testers' mind-set towards frequent direct communication and participative behaviours (Cohn, 2010).

The fact that transitioning to Agile does not assure happiness of testers was explicated by Deak et al. (2016). From their work, however, it is not entirely clear why "more agile [than waterfall] testers were unhappy about their relationship with the developers". Their research indicates that "removing the wall" might be not enough if subsequent coaching strategies for both programmers and testers are not implemented in order to increase "group maturity" of the ISD team (Gren et al., 2017).

Based on these challenges, the essence of future research efforts could lay in (i) understanding effective coaching mechanisms to help programmers and testers transitioning from waterfall to agile, and (ii) the creation of guidelines to help these groups working in hybrid environments where not all agile principles are applicable in a pure form.

## 3.3 Combined Software Engineering: Eradication of Testers

### 3.3.1 Grounding

"Combined software engineering" is a term popularized by Microsoft (Dobrigkeit et al., 2016). The notion implies that there had been traditionally at least two broad functional responsibilities and career paths in ISD: programming and testing. (In this discussion, for the sake of simplicity, we ignore distinct career paths of software architects/analysts and development managers.) They may had been given titles such as "Software (Development) Engineer" and "Software (Development) Engineer in Test" (Page et al., 2009) . Note that Google, among others, is known to differentiate between "Software Engineers in Test" and "Test Engineers" more precisely (see Whittaker et al., 2012).

The original organizational situation of having *some* dedicated testing roles clearly differs from having *no* testers at all historically. The latter may be typical in smaller or "less mature" – according the traditional worldview – companies (Prechelt, 2016). Speaking about the former, some companies recently introduced certain organizational changes in order to stop differentiating between their programmers and testers in terms of their professional status. Simply put, these companies have *combined* the two previously independent ISD functions (Figure 1, type C). These organizational changes are implemented consistently with the companies' hiring, firing, and compensation & benefits policies. Recently the expert public paid quite a lot of attention to the case of Microsoft in which testers played an important role (Thonangi, 2014).

### 3.3.2 Key Industrial Challenges

The idea of "combined software engineering" is quite new and unproven. Though there are some interesting blog posts (e.g. Jensen, 2016), there is not a lot of information in printed literature to date. We see two important goals on which further research should concentrate: (i) to understand organizational enablers of combined software engineering models, and (ii) to help organizations with solving possible side effects and people problems in the area of motivation when such a model is introduced to the organization. In our opinion, the first area can be elegantly studied using cultural lens in order to understand nuances of organizational life *culturally* (Smircich, 1983). The latter one calls for more research on the motivation of programming and testing specialists under the mentioned organizational conditions (Beecham et al., 2008; Deak et al., 2016).

## 4 RESEARCH DIRECTION AND DISCUSSION

In this section, we briefly explain our open-ended research idea. Our overall goal is to understand which of the configurations explained above real organizations use, and what the reasons behind their decisions are. By exploring this problem, we hope to provide a conceptual guideline to organizations transitioning from traditional ISD approaches to Agile ISD. Specifically, we believe that this endeavour might help practitioners with forming and managing cross-functional agile teams in enterprise environment. Similarly, the new theory we aim to develop will hopefully contribute to a better theoretical understanding of this area. Overall, regarding the theorizing which follows, we are roughly guided by data from our ongoing research projects.

It is our argument that the body of knowledge on *role identity* (RI) accumulated in MOS can help with further directing our research. The RI work is informed predominantly by concepts originating in social psychology and microsociology (or *sociological* social psychology), in particular by Social Identity Theory (SIT, see Tajfel and Turner, 1986) and Identity Theory (IDT, see Stryker and Burke, 2000) respectively.

> *Role identities [or role-based identities] are socially constructed definitions of self-in-role (this is who a role occupant is). Role identities anchor or ground self-conceptions in social domains. To switch roles is to switch social identities.*
>
> (Ashfort, 2001, p. 27)

A specific type of role identity is professional role identity (or simply professional identity), which is the term we have introduced in Section 2.4 without providing much theoretical background. Differently from personal identity, social, role, and professional identities are based on group membership. With regards to the problem studied by the present paper, it is important to understand the concept of group very broadly. In our case, the first group (a *macro* group) might be that of the professional community of software testers (where exists and its influence is salient). The second group (a *micro* group) is that of a particular ISD team where the tester works. An additional group membership (a *meso* group) may be introduced when the company centralizes software testing activities under one tent of enterprise test organization or test centre. These centralized entities can form an additional organizational layer across the existing landscape of ISD teams that have been previously constituted in the company (Doležel, 2017). And naturally, these two or three social group memberships can collide.

We thus propose that one needs to understand not only the micro organizational context, but also more abstract, high-level layers that contribute to forming of the professional identity (i.e. the meso and macro levels). In this sense, one needs to identify the influence of "institutional forces" (a term borrowed from sociology). This need stems from the fact that "professionals act as bridges between the institutional forces of their professions and their respective organizations" (Daudigeos, 2013, p. 725).

Our key thesis is thus provocative. We argue that the macro social forces that drive the ongoing professionalization of software testers may significantly conflict with the core Agile principles implemented in a purist (i.e. *crusader*) way at the micro level. *Crusader organizations* are "employing a highly prescriptive adoption of agile techniques, alongside an avoidance of traditional approaches entirely" (Cram and Newell, 2016, p. 9). By contrast, dedicated software testers typically work in traditional, larger, and "more mature" organizations.

When such a traditional organization wants to *suddenly become a crusader*, software testers as a profession may feel jeopardized by the ideas presented by the Agile community, and react defensively. An excellent example supporting our view is presented by Larman and Vodde (2010) in their book. Book sections titled "*Avoid… Test department*", "*Avoid… TMM, TPI, and other [test] 'maturity' models*", and *"Avoid… ISTQB and other tester certification"* (!) speak for themselves. It seems not exaggerated to expect that if their advices are followed by ISD managers literally, the relevant organizational changes must result in a *professional identity crisis of software testers* working for those managers. *Everything the testers learned in the past is gone, and the world is not the same as it was.* A piece of anecdotal evidence describing testers' reactions during a training run by Larman and Vodde is presented in the same source.

Inversely, people in *crusader organizations* typically believe that Agile is "a better, more rational approach compared to traditional methods" (Cram and Newell, 2016, p. 6). In other words, relevant socio-psychological forces are aligned with the organizational culture of such companies and people happily work there while adoring the mentioned principles of agility. As indicated above, this balanced, positive state may significantly differ from a situation, when an organization had institutionalized different working patterns and interaction styles in the past years, and suddenly decided to change them overnight. In such cases, the psychological safety of members of certain professional groups may be significantly harmed. The previous work patterns and interaction styles are suddenly out-dated, and the new ones still to be created. More importantly, unless the impacted groups feel safe and comfortable with the new reality, the implemented change won't be successful (Burnes, 2004).

Though the above ideas are mostly speculative, we present them in this paper because we believe that they are quite important and promising for ISD practice. It is also our hope that they may guide further theory building efforts carried out by both IS and SE researchers. Interestingly, though similar interests were originally inherent mostly to research in the sociology of professions, it is argued that this discipline gradually "came to an intellectual standstill" (Gorman and Sandefur, 2011, p. 290). Instead of occupational sociologists, increasingly often scholars who educate knowledge workers dominate the field research on "their" professions, focusing on everyday activities of "their" professionals (ibid.).

## 5 CONCLUSION

This paper has discussed the topic of organizational patterns between programmers and testers, and how these patterns evolve in Agile ISD. Drawing on the problem of testers' independence (Sunyaev and Basten, 2015), and extending the problem to the Agile world, the paper has summarized existing knowledge and indicated the further research direction. We have taken a critical stance to pinpoint some problems we see when "Agile ideals" are blindly followed and used as a *rhetorical tool* and *salvation device* (Case, 1999).

Our position articulated in this paper is that the previous IS and SE research indicates that programmers and testers form distinct groups with distinct social and professional identities. These groups execute their work activities in a different manner. We support the view that agile ISD must be philosophically based on a new set of fundamental principles (Cohn, 2010). However, based only on anecdotal evidence, we remain undecided whether transformation to a "combined software engineering" model in a traditional company can be successful. If it can, it will be imperative to understand situational factors that contribute to this success (Clarke and O'Connor, 2012).

We indeed agree that ISD teams may function very well without dedicated testers (Prechelt, 2016). We, however, cautiously note that before going this route, a "traditional" company where software testers are a well-established profession should carefully analyse the impact of such decision, and propose sound risk mitigating strategies. The history teaches us that simply jumping on the bandwagon of the latest management fad hardly ever paves the way towards the success of the intended change (Case, 1999).

We are eager to hear about further independent research efforts that probe into organizations who deployed such models. We propose that success or failure of the change initiative should be understood in terms of the following criteria (though not necessarily all adopted in a single case study): (i) objective, measurable quality metrics demonstrate that software quality is not degraded under the new conditions; (ii) fulfilling all tasks of the former ISD actors, new ISD teams have a high level of group maturity (Gren et al., 2017); (iii) previous programming and testing personnel is still motivated and happily working in the new setting (Deak et al., 2016), proud of their newly acquired identity. Based on anecdotal evidence and our own work-in-progress research data, we are concerned about, especially with regards to the last criteria. We have noticed that effects on the morale and motivation of both the unique *software species* seem to devastating when the changes are introduced insensitively, and people do not understand reasons behind them (see also Jensen, 2016).

Our research continues to concentrate mostly on the standing of testers in hybridized ISD settings (Cram and Newell, 2016; Kuhrmann et al., 2017). We work on an empirical study that uses the lens of SIT and IDT to understand the challenges introduced by blending the roles and responsibilities of programmers and testers in agile ISD processes. In parallel, we also explore the growing influence of DevOps on software testing concepts, as exemplified by the concept of "testing in production".

## REFERENCES

Anderson, D. J. (2003), *Agile Management for Software Engineering*, Pearson Education, Upper Saddle River, NJ.

Andrade, R., Santos, I., Lelli, V., Oliveira, K. and Rocha, A. (2017), "Software Testing Process in a Test Factory: From Ad hoc Activities to an Organizational Standard", *International Conference on Enterprise Information Systems*, pp. 132–143.

Ashfort, B. E. (2001), *Role Transitions in Organizational Life: An Identity-Based Perspective*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Atkinson, R. (1999), "Project management: cost, time, and quality, two best guesses and a phenomenon, it's time to accept other success criteria", *International Journal of Project Management*, Vol. 17 No. 6, pp. 337–342.

Balijepally, V., Mahapatra, R. and Nerur, S. (2006), "Assessing Personality Profiles of Software Developers in Agile Development", *Communications of the Association for Information Systems*, Vol. 18 No. 1, p. Article 4.

Beecham, S., Baddoo, N., Hall, T., Robinson, H. and Sharp, H. (2008), "Motivation in Software Engineering: A systematic literature review", *Information and Software Technology*, Vol. 50 No. 9–10, pp. 860–878.

Beer, A. and Ramler, R. (2008), "The Role of Experience in Software Testing Practice", *Software Engineering and Advanced Applications*, pp. 258–265.

Boehm, B. (1984), "Verifying and validating software requirements and design specifications", *IEEE Software*, Vol. 1 No. 1, pp. 75–88.

Burnes, B. (2004), "Kurt Lewin and the planned approach to change: A re-appraisal", *Journal of Management Studies*, Vol. 41 No. 6, pp. 977–1002.

Case, P. (1999), "Remember re-engineering? The rhetorical appeal of a managerial salvation device", *Journal of Management Studies*, Vol. 36 No. 4, pp. 419–441.

Clarke, P. and O'Connor, R. V. (2012), "The situational factors that affect the software development process: Towards a comprehensive reference framework", *Information and Software Technology*, Vol. 54 No. 5, pp. 433–447.

Cohen, C. F., Birkin, S.J., Garfield, M.J. and Webb, H.W. (2004), "Managing conflict in software testing", *Communications of the ACM*, Vol. 47 No. 1, pp. 76–81.

Cohn, M. (2010), *Succeding with Agile*, Addison Wesley, Upper Saddle River, NJ.

Conboy, K. (2009), "Agility from first principles: Reconstructing the concept of agility in information systems development", *Information Systems Research*, Vol. 20 No. 3, pp. 329–354.

Cram, W.A. and Newell, S. (2016), "Mindful revolution or mindless trend? Examining agile development as a management fashion", *European Journal of Information Systems*, Vol. 25 No. 2, pp. 154–169.

Crispin, L. and Gregory, J. (2009), *Agile Testing: A Practical Guide for Testers and Agile Teams*, Addison-Wesley, Upper Saddle River.

Daudigeos, T. (2013), "In Their profession's service: How staff professionals exert influence in their organization", *Journal of Management Studies*, Vol. 50 No. 5, pp. 722–749.

Deak, A., Stålhane, T. and Sindre, G. (2016), "Challenges and strategies for motivating software testing personnel", *Information and Software Technology*, Vol. 73, pp. 1–15.

Dobrigkeit, F., Meyer, S. and Uflacker, M. (2016), "Case Studies on End-User Engagement and Prototyping during Software Development", *Design Thinking Research: Taking Breakthrough Innovation Home*, pp. 183–213.

Doležel, M. (2017), "Images of Enterprise Test Organizations: Factory, Center of Excellence, or Community?", *Software Quality Days*, Vienna.

Evetts, J. (2014), "The concept of professionalism: Professional work, professional practice and learning", *International Handbook of Research in Professional and Practice-Based Learning*, pp. 29–57.

Fuggetta, A. and Nitto, E. Di. (2014), "Software Process", *Proceedings of the on Future of Software Engineering*, pp. 1–12.

Garousi, V. and Felderer, M. (2017), "Worlds Apart: Industrial and Academic Focus Areas in Software Testing", *IEEE Software*, Vol. 34 No. 5, pp. 38–45.

Garousi, V. and Mäntylä, M. V. (2016), "A systematic literature review of literature reviews in software testing", *Information and Software Technology*, Vol. 80, pp. 1339–1351.

Gorman, E. H. and Sandefur, R. L. (2011), "'Golden Age,' Quiescence, and Revival", *Work and Occupations*, Vol. 38 No. 3, pp. 275–302.

Grechanik, M., Jones, J. A., Orso, A. and Van Der Hoek, A. (2010), "Bridging gaps between developers and testers in globally-distributed software development", *FoSER*, Santa Fe, New Mexico, pp. 149–153.

Gren, L., Torkar, R. and Feldt, R. (2017), "Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies", *Journal of Systems and Software*, Vol. 124, pp. 104–119.

Hassan, N. R. and Mathiassen, L. (2018), "Distilling a body of knowledge for information systems development", *Information Systems Journal*, Vol. 28 No. 1, pp. 175–226.

Hughes, E.C. (1958), *Men and Their Work*, Free Press, London.

Charrett, A.-M. (2012), "Appendix D: Cost of Starting Up a Test Team", *How to Reduce the Cost of Software Testing*, CRC Press, Boca Raton, FL.

Chemuturi, M. (2011), *Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers*, J.Ross Publishing, Fort Lauderdale, FL.

ISTQB. (2011), "ISTQB Foundation Level Syllabus", available at: https://www.istqb.org/downloads/.

Jensen, B. (2016), "The Combined Engineering Software Model", available at: https://testastic.wordpress.com/2016/01/03/the-combined-engineering-software-model/.

Koch, A. S. (2000), "'A' is for 'Assurance'– A Broad View of SQA", *Pacific Northwest Software Quality Conference*, Portland, OR, pp. 217–236.

Kuhrmann, M., Hanser, E., Prause, C.R., Diebold, P., Münch, J., Tell, P., Garousi, V., et al. (2017), "Hybrid software and system development in practice: waterfall, scrum, and beyond", *Proceedings of the 2017 International Conference on Software and System Process*, pp. 30–39.

Laloux, F. (2014), *Reinventing Organizations: A Guide to Creating Organizations Inspired by the Next Stage in Human Consciousness*, Nelson Parker, Brussels, Belgium.

Larman, C. and Vodde, B. (2010), *Practices for Scaling Lean & Agile Development*, Addison-Wesley, Upper Saddle River, NJ.

McKay, J. (2007), *Managing the Test People: A Guide to Practical Technical Management*, Rocky Nook, Santa Barbara, CA.

Onita, C. and Dhaliwal, J. (2011), "Alignment within the corporate IT unit : an analysis of software testing and development", *European Journal of Information Systems*, Vol. 20 No. 1, pp. 48–68.

Page, A., Johnston, K. and Rollison, B. (2009), *How We Test Software in Microsoft*, Microsoft Press, Redmond, WA.

Pettichord, B. (2000), "Testers and Developers Think Differently", *Software Testing and Quality Engineering Magazine*, Vol. Jan/Feb, pp. 42–46.

Prechelt, L. (2016), "Quality Experience : A Grounded Theory of Successful Agile Projects Without Dedicated Testers", *International Conference on Software Engineering*.

Scrum.org. (n.d.). "What is a Scrum Development Team?", available at: https://www.scrum.org/resources/what-is-a-scrum-development-team.

Shah, H. and Harrold, M.J. (2013), "Culture and Testing: What is the Relationship?", *8th International Conference on Global Software Engineering*, IEEE, pp. 51–60.

Smircich, L. (1983), "Concepts of Culture and Organizational Analysis", *Administrative Science Quarterly*, Vol. 28 No. 3, p. 339.

Stryker, S. and Burke, P.J. (2000), "The Past, Present, and Future of an Identity Theory", *Social Psychology Quarterly (Special Millenium Issue on the State of Sociological Social Psychology)*, Vol. 63 No. 4.

Sunyaev, A. and Basten, D. (2015), "Truth and myth of independent software testing", *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 1722–1728.

Tajfel, H. and Turner, J.C. (1986), "The Social Identity Theory of Intergroup Behavior", *Psychology of Intergroup Relations*, Nelson-Hall Publishers, Chicago, IL, pp. 7–24.

Thonangi, U. (2014), "Why did Microsoft lay off 'Programmatic testers'?", available at: https://www.linkedin.com/pulse/20140806183208-12100070-why-did-microsoft-lay-off-programmatic-testers/.

TMMi Foundation. (2012), "Test Maturity Model integration", Ireland, available at: www.tmmi.org.

Walz, D. B., Elam, J. J. and Curtis, B. (1993), "Inside a software design team: knowledge acquisition, sharing, and integration", *Communications of the ACM*, Vol. 36 No. 10, pp. 63–77.

Whittaker, J. A., Arbon, J. and Carollo, J. (2012), *How Google Tests Software*, Addison-Wesley, Upper Saddle River, NJ.

Zhang, X. "Paul", Nickels, D., Poston, R. and Dhaliwal, J. (2018), "One World, Two Realities: Perception Differences between Software Developers and Testers", *Journal of Computer Information Systems*, No. in press.

Zhang, X., Stafford, T. F., Dhaliwal, J. S., Gillenson, M. L. and Moeller, G. (2014), "Sources of conflict between developers and testers in software development", *Information and Management*, Vol. 51 No. 1, pp. 13–26.